

ellucian.

# **Banner General Data Security Handbook**

Release 8.6.3  
March 2014



Banner®, Colleague®, PowerCampus™, and Luminis® are trademarks of Ellucian Company L.P. or its affiliates and are registered in the U.S. and other countries. Ellucian®, Ellucian Advance™, Ellucian Degree Works™, Ellucian Course Signals™, Ellucian SmartCall™, and Ellucian Recruiter™ are trademarks of Ellucian Company L.P. or its affiliates. Other names may be trademarks of their respective owners.

© 2004-2014 Ellucian Company L.P. and its affiliates.

Contains confidential and proprietary information of Ellucian and its subsidiaries. Use of these materials is limited to Ellucian licensees, and is subject to the terms and conditions of one or more written license agreements between Ellucian and the licensee in question.

In preparing and providing this publication, Ellucian is not rendering legal, accounting, or other similar professional services. Ellucian makes no claims that an institution's use of this publication or the software for which it is provided will guarantee compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization's own choosing.

Prepared by: Ellucian  
4375 Fair Lakes Court  
Fairfax, Virginia 22033  
United States of America

## Revision History

<b>Publication Date</b>	<b>Summary</b>
March 2014	New version that supports Banner General 8.6.3 software.

# Contents

---

<b>Overview</b> .....	<b>11</b>
<b>VBS and PII</b> .....	11
<b>Oracle Fine-Grained Access Control</b> .....	11
<b>Value-Based Security</b> .....	12
Examples of VBS .....	12
Banner Student .....	13
Banner Finance .....	13
Banner Advancement .....	13
Banner Financial Aid .....	13
Banner Human Resources .....	14
<b>Personally Identifiable Information Security</b> .....	14
PII Security Features .....	14
Example of PII .....	15
<b>Masking, Hiding, and Concealing Fields</b> .....	15
<b>FGAC Exemptions</b> .....	16
Cross Domain Search Forms .....	16
<b>FGAC Information for End Users</b> .....	17
<b>FGAC in Integrated SunGard Higher Education Products</b> .....	17
Self-Service Products .....	17
Virtual Private Database (VPD) .....	18
Banner Workflow and Banner Document Management Suite .....	18
Operational Data Store and Enterprise Data Warehouse .....	18
e~Print Reports .....	19
Banner Voice Response .....	19
Luminis Channels for Banner and Luminis Data Integration (LDI) .....	19
<b>Banner Forms for FGAC</b> .....	19
<i>Fine-Grained Access Control (*GENFGAC)</i> .....	20
<i>FGAC Value-Based Security Maintenance (*GENFGBS)</i> .....	20

<i>FGAC Personal Identification Information Maintenance (*GENFGPII)</i> . . . . .	20
<i>Protection of Sensitive Data (*GENDMSK)</i> . . . . .	20
<b>FGAC Glossary</b> . . . . .	21
<i>Business Profile</i> . . . . .	21
<i>Context Variables</i> . . . . .	21
<i>Domain</i> . . . . .	21
<i>FGAC</i> . . . . .	21
<i>Field-Level Security</i> . . . . .	21
<i>GOKFGAC</i> . . . . .	21
<i>PII</i> . . . . .	21
<i>Predicate Functions</i> . . . . .	22
<i>Row-Level Security</i> . . . . .	22
<i>Banner Object-Level Security</i> . . . . .	22
<i>VBS</i> . . . . .	22
<i>VPD</i> . . . . .	22
<b>Value-Based Security</b> . . . . .	23
<b>Business Need Analysis for VBS</b> . . . . .	23
Analysis Worksheet Template . . . . .	23
Worksheet Example 1 . . . . .	24
Worksheet Example 2 . . . . .	26
<b>Setup Steps for VBS</b> . . . . .	27
Step 1—Define the Domain and Its Driver Table . . . . .	27
Driver Tables with Sequence Numbers . . . . .	27
Step 2—Define Policy Tables for the Domain . . . . .	27
Example—Admissions Domain . . . . .	28
Step 3—Create Oracle Policies . . . . .	29
Using BANINST1 to run gfvbsaddpol.sql . . . . .	29
Viewing Policy Data from SQLPLUS . . . . .	30
<i>Policy names</i> . . . . .	30
Dropping a Policy . . . . .	30
Step 4—Set Up VBS Rules . . . . .	31
Adding a Group Code . . . . .	31
Setting up the Predicate for the Rule on the GOAFGAC Form . . . . .	31
Defining Access to the Predicate . . . . .	31
Testing the Predicate on the FGAC Predicate Inquiry Form (GOIFGAC) . . . . .	32
<b>Building Predicate Logic</b> . . . . .	32
Worksheet Example . . . . .	33
<b>VBS Domains</b> . . . . .	34

<b>VBS Domain Driver Tables</b> .....	35
<b>VBS Domain Policy Tables</b> .....	36
<b>VBS Domain Implementation Details</b> .....	54
FGAC VBS Predicate Access .....	54
FGAC VBS Access .....	55
International .....	56
Addresses .....	58
Telephones .....	60
Medical .....	61
Catalog .....	62
Schedule .....	64
Curriculum .....	66
Field of Study .....	69
Recruit .....	72
Admissions .....	75
Learner (General Student) .....	78
Test Scores .....	81
Student Test Scores .....	82
Test Codes .....	84
Other GPAs .....	85
Student Other GPAs .....	86
Other GPA Type Codes .....	87
<b>PII Security</b> .....	<b>89</b>
<b>PII Setup Steps</b> .....	89
Step 1—Define Domains .....	89
Naming Conventions for PII Domains .....	89
Creating a Domain .....	89
Step 2—Define the Table PII is Restricting .....	90
Examples of PII on SPBPERS .....	90
Entering PII table on the FGAC PII Table Rules Form (GORFDPI) .....	90
Step 3—Run GFPIIADDPOL.SQL to Create the FGAC Policy .....	91
Steps for Starting (Turning on) PII .....	91
Viewing policy data from SQLPlus .....	92
Dropping a Policy .....	92
Step 4—Manage User Access to PII Domains .....	92
Assigning Domains to Users with the GOAFPUD Form .....	93

Assigning Domains to Business Profiles . . . . .	93
<b>Creating Exemptions from PII . . . . .</b>	<b>93</b>
Exempting Individuals from PII . . . . .	93
Cross Domain Search . . . . .	94
Banner Search Forms—SOAIDEN, FOIIDEN, GOAMTCH, GUIALTI. . . . .	94
Cross Domain Search Indicator . . . . .	94
Exempting Objects from PII . . . . .	94
Self-Service . . . . .	95
<b>The PII Predicate . . . . .</b>	<b>95</b>
Viewing the PII Predicate . . . . .	95
The PII Predicate’s Logical Structure . . . . .	95
<b>Suggested PII Implementation . . . . .</b>	<b>96</b>
<i>Decide if the GTVSDAX PIIPROMPT is set to Y or N . . . . .</i>	<i>97</i>
<i>Exempt all users from PII . . . . .</i>	<i>97</i>
<i>Turn PII on . . . . .</i>	<i>97</i>
<i>Define Business Profiles . . . . .</i>	<i>97</i>
<b>PII Domains . . . . .</b>	<b>98</b>
<b>PII Domain Driver Tables . . . . .</b>	<b>99</b>
<b>Masking, Hiding, and Concealing Fields . . . . .</b>	<b>101</b>
<b>Setup for Protecting Sensitive Data . . . . .</b>	<b>101</b>
List Items in the Data Display Mask Columns Data Dictionary. . . . .	101
Setting up Formatting Rules on the GORDMSK Form . . . . .	102
Hiding . . . . .	102
Concealing . . . . .	102
Masking a Number or Date . . . . .	102
Partial Character Masking . . . . .	103
SSN Field on the Extended ID/Name Search Window. . . . .	103
Masking “International” Person Information Fields . . . . .	104
Using gssmask.sql . . . . .	104
Using GORDMSK to Hide the “International” Fields . . . . .	105
Hiding the “International” Fields in Self-Service . . . . .	105
Including Last Name Prefix in Full Name . . . . .	106
Priority of Rules . . . . .	106
Implementation Issues with Masking, Hiding, and Concealing. . . . .	107
The First Field in a Block. . . . .	107

Hiding All Fields on a Block or Form . . . . .	107
Incorrect or Mistyped Item Names . . . . .	108
Masking a Required Data Entry Field . . . . .	108
Masking All Items on a Search Form . . . . .	108
Masking or Concealing Banner ID. . . . .	108
Dropdown Lists . . . . .	109

## **Controlling and Troubleshooting FGAC. . . . . 110**

<b>Debugging and Oracle Trace Output . . . . .</b>	<b>110</b>
--	------------

<b>Policy Errors . . . . .</b>	<b>111</b>
--------------------------------	------------

SELECT . . . . .	111
------------------	-----

INSERT. . . . .	111
-----------------	-----

UPDATE . . . . .	112
------------------	-----

DELETE . . . . .	112
------------------	-----

<b>Best Practices and Guidelines for FGAC Implementation . . . . .</b>	<b>113</b>
--	------------

Oracle Parameter <code>_DYNAMIC_RLS_POLICIES</code> . . . . .	113
---	-----

Business Profiles . . . . .	113
-----------------------------	-----

Creating Rules . . . . .	114
--------------------------	-----

Building Restrictions . . . . .	114
---------------------------------	-----

Checking Access . . . . .	114
---------------------------	-----

Verifying Rules . . . . .	114
---------------------------	-----

Reviewing GOIFGAC . . . . .	115
-----------------------------	-----

Where Not to Use VBS. . . . .	115
-------------------------------	-----

Recursive FGAC Processing . . . . .	116
-------------------------------------	-----

Working with Different Domains . . . . .	116
--	-----

Predicate Errors . . . . .	116
----------------------------	-----

Syntax for the Join Condition on GORFDPL . . . . .	116
--	-----

VBS and Sequence Number Fields in Tables. . . . .	118
---	-----

Using Select Restrictions . . . . .	118
-------------------------------------	-----

Restrictions and Validation Table Codes . . . . .	118
---	-----

## **FGAC Reference. . . . . 119**

<b>Tables. . . . .</b>	<b>119</b>
------------------------	------------

FGAC Domain Table (GTVFDMN) . . . . .	119
---------------------------------------	-----

FGAC Business Profile Validation Table (GTVFBPR) . . . . .	119
--	-----

FGAC Group Code Validation Table (GTVFGAC) . . . . .	119
--	-----

FGAC Domain Type Code Validation Table (GTVFDTP) . . . . .	120
FGAC Domain Driver Rules Table (GOBFDMN) . . . . .	120
FGAC Domain Type Driver Rule Table (GOBFDTP) . . . . .	120
FGAC User Business Profile Assignments Table (GORFBPR) . . . . .	121
FGAC PII Business Profile Assignments (GORFBPI) . . . . .	121
FGAC User Profile Table (GOBFPUD) . . . . .	121
PII Domain Maintenance Table (GORFPPII) . . . . .	122
FGAC Group Definition Table (GOBFGAC) . . . . .	122
User Access for Domain and Group Table (GORFGUS) . . . . .	122
Business Profile Access to Domain and Group Table (GORFGPR) . . . . .	123
FGAC Predicates Table (GORFPRD) . . . . .	123
FGAC VBS Table Rules Table (GORFDPL) . . . . .	124
FGAC PII Table Rules Table (GORFDPI) . . . . .	124
FGAC Excluded Objects Table (GOBFEOB) . . . . .	125
Data Display Mask Columns Table (GORDMCL) . . . . .	125
Data Display Mask Rules Table (GORDMSK) . . . . .	126
Person Identification/Name Repeating Table (SPRIDEN) . . . . .	126
<b>Views</b> . . . . .	126
FGAC Restricted Users View (GOVFGAC) . . . . .	126
FGAC Restricted Users View (GOVFUSR) . . . . .	127
<b>Procedures</b> . . . . .	127
GOKFGAC Database Package Procedures to Support VBS and PII Processing . . . . .	127
P_ADD_POLICY . . . . .	127
P_DROP_POLICY . . . . .	128
P_REQUIRE_PII_PROMPT . . . . .	128
P_ENABLE_FDMN . . . . .	128
P_DOMAIN_ENABLED . . . . .	128
P_GET_USER_INFO . . . . .	128
P_FIND_USER_PII_DOMAINS . . . . .	128
P_SET_HOME_DOMAIN . . . . .	128
P_DISABLE_FGAC . . . . .	128
P_UPDATE_TEMP_PIDM . . . . .	128
P_OBJECT_EXCLUDED . . . . .	128
P_TURN_FGAC_OFF . . . . .	129
P_TURN_FGAC_ON . . . . .	129
P_SEARCH_FORM_UPDATE_TEMP_PIDM . . . . .	129
<b>Functions</b> . . . . .	129
GOKFGAC Database Package Functions to Support VBS and PII Processing . . . . .	129
f_get_pidm . . . . .	129

f_get_temp_pidm . . . . .	129
f_get_home_domain . . . . .	129
f_is_user_exempt . . . . .	129
f_find_pii_domain . . . . .	129
f_spriden_pii_active . . . . .	130
f_predicate_header . . . . .	130
f_find_predicate . . . . .	130
f_common_predicate . . . . .	130
f_query_policy . . . . .	130
f_query_predicate . . . . .	130
f_select_fnc . . . . .	130
f_insert_fnc . . . . .	130
f_update_fnc . . . . .	130
f_delete_fnc . . . . .	131
f_validate_predicate . . . . .	131
f_validate_vbs_driver . . . . .	131
f_format_data . . . . .	131
<b>Scripts</b> . . . . .	131
GENROLE.SQL . . . . .	131
<b>APIs</b> . . . . .	131
GORFDPL API . . . . .	132
GOBFDMN API . . . . .	132
GOBFDTP API . . . . .	133
GOBFGAC API . . . . .	133
GORFPRD API . . . . .	134
GOBFEOB API . . . . .	134
GORFBPR API . . . . .	135
GORFGPR API . . . . .	136
GORFGUS API . . . . .	136
GORFPPII API . . . . .	137
GOBFPUD API . . . . .	137
GORDMCL API . . . . .	138
GORDMSK API . . . . .	138
GORFDPI API . . . . .	139
GORFBPI API . . . . .	139
Login Script GOTVBSIO . . . . .	140
<b>Context Variables</b> . . . . .	140
CTX_FDMN_CODE_HOME . . . . .	140
CTX_FG_PIDM . . . . .	141

CTX_FG_TABLENAME_ACTIVITY .....	141
CTX_FG_USER_EXEMPT_IND .....	141
CTX_FG_PII_PROMPT .....	141
CTX_FG_EXCLUDE_OBJECT .....	141
<b>Database Objects</b> .....	142
gspsecr.g\$_verify_password2_prd .....	142
GOQRPLS.PLL .....	142
How Form Item Properties Are Applied .....	142
G\$_STARTUP_MASKS .....	142
G\$_QUERY_STRING_MASKS .....	143
GOAMTCH Form .....	143
SOAIDEN and GUIALTI Forms .....	143
<b>Seed Data</b> .....	143
FGAC Domain Type Validation Table (GTVFTYP) .....	143
FGAC Domain Type Predicate Rule Table (GOBFDTP) .....	144
FGAC PII Table Rules Table (GORFDPI) .....	144
Crosswalk Validation Table (GTVSDAX) .....	144
Data Display Mask Columns Table (GORDMCL) .....	144
FGAC Excluded Objects Table (GOBFEOB) .....	145

# Overview

---

Fine-Grained Access Control (FGAC) is an Oracle feature that can be used to provide row-level security for Oracle tables. Banner® offers two optional data security features that take advantage of the capabilities of Oracle FGAC. These features are:

- The new Value-Based Security (VBS)
- Security for Personally Identifiable Information (PII)

This handbook describes VBS and PII, explains how to set up and manage them, and provides some advice on how to plan your implementation. Throughout the handbook, the term *FGAC* will be used to refer both to the Oracle feature and to Banner's full set of VBS and PII functionality.

In addition to these FGAC-based features, this handbook also describes how to accomplish the protection of sensitive data at the field level, using a set of techniques called masking, hiding, and concealing.

## VBS and PII

---

Banner's VBS and PII both use the same technology—Oracle FGAC. They both selectively limit the user's access to rows in the Oracle database. The difference between the two is based on their intended purposes. VBS is a flexible tool for restricting user's access to data. PII is designed for a specific purpose: to prevent unnecessary exposure of sensitive person data.

VBS restrictions are coded as needed for individual users. VBS applies row-level security based on column values. The FGAC predicate defines the specific columns and values for a VBS restriction.

PII security is designed to keep users from viewing person information outside their own functional processing area. The processing area is typically affiliated with a department, such as a payroll department or a registrar's office. Because PII's focus is protecting person information, all seed data delivered for PII is keyed to the Banner Person Identification Table (SPRIDEN).

## Oracle Fine-Grained Access Control

---

VBS and PII both rely on Oracle's Fine-Grained Access Control (FGAC). FGAC transparently modifies the code that retrieves, updates, inserts, or deletes information from the database by dynamically adding business rules established by your institution. SQL statements are created in the Banner rule forms to specify restriction criteria.

Security checking takes place when data rows are accessed for select, insert, update, or delete. FGAC dynamically and transparently modifies the `WHERE` clause by adding the institutionally defined criteria.

As a simple example, you can apply VBS rules to users in one department so that they can only see student records relevant to that department. When a user in that department is using Banner to view student records, FGAC will transparently filter out other records, returning only the student records that the user is allowed to see.

## Value-Based Security

---

Prior to Release 7.0, Banner had a limited Value-Based Security feature that could be used to restrict certain Student and Financial Aid data. This “old VBS” was removed and replaced in Release 7.0 with the much more versatile “new VBS” based on FGAC.

By taking advantage of FGAC, Banner’s VBS offers the following features:

- Streamlined design; easier to maintain rules that control row level security
- Security is implemented once and enforced by all products and tools accessing the database
- Flexible group rules can be developed and maintained for business profiles and users. You can set up and change FGAC security definitions as business practices change.
- FGAC security carries over to all Banner processing including Banner job submission reports and processes, Oracle Forms, Oracle Reports, Workflow, Job Submission, and Self-Service.
- Forms are provided for establishing FGAC policy rules on a table, table relationships, and user access rules.
- Users, individual Banner processes, and jobs can be excluded from FGAC security.
- FGAC security rules (predicates) can be defined to include a single column or multiple columns. Oracle table rows can also be secured based on column values or other data elements.

The setup of VBS rules is similar to the setup of Banner population selection and letter generation rules. SQL statements are created in Banner forms to specify the restriction criteria. Using SQL-based rules means you aren’t limited to a small number of data elements in your rules and you can use them in any Banner module.

## Examples of VBS

Suppose you want to restrict access to addresses by address type. You create a VBS rule that says staff in the Admissions, Registration, Student Billing and the Student Dean’s offices can only select, insert, update and delete addresses if the type is mailing, permanent or parents. The staff in Payroll and Central Accounting can insert, update and delete addresses only if it’s a W2 or permanent type but they can view any address.

When the admissions staff creates a new applicant, they can only enter a permanent, mailing or parent address for the new person. Their mailings can only select mailing, permanent or the parents address types. When HR needs to select addresses for an upcoming mailing, they can select any address for the individual but they can only maintain the W2 and permanent types.

Users in Alumni Development are not included on any of the VBS rules, therefore not restricted to any specific address type.

Following are additional examples of how VBS can be used in Banner.

## Banner Student

Most institutions have separate processes for undergraduate and graduate admissions. A crucial step in the admissions process is tagging received “checklist” items for an application. Typically, institutions hire student workers to enter the “received date” on checklist items. With the new VBS, you can restrict student workers to handle only those applications that belong to a specific school/college and department. Furthermore, student workers can be restricted to handle only those checklist items that are affiliated with a specific school/college and department.

## Banner Finance

When institutions distribute requisition entry using self-service, they want to limit purchasing to certain expenditure account codes. With VBS, they can assign those users specific account codes to use in the requisition process, but exempt other users so they have full use of other accounts such as inventory accounts.



**Note:** VBS does not replace fund/organization security in Banner Finance. These security features, though similar in purpose, work independently of VBS.

## Banner Advancement

Class year giving volunteers may need to see gifts made to their class year fund but not any gifts from classmates for other years. VBS can restrict these users' access to gifts from a specific year only.

## Banner Financial Aid

Institutions may have separate financial aid offices for their graduate and undergraduate divisions. VBS can be used to restrict users to the appropriate division.

## Banner Human Resources

Institutions may want to restrict specific users to see or process employees in certain position classes in Position Control. VBS can be set up to accomplish this restriction. Likewise, VBS can be used to restrict users to specific bargaining units.



**Note:** VBS does not replace employer, organization, salary, and employee class security in Banner Human Resources. These security features, though similar in purpose, work independently of VBS.

## Personally Identifiable Information Security

---

Banner's PII feature enables you to secure the sensitive personal information of applicants, students, employees, and alumni maintained within your Banner database. With PII, you can protect private information such as birth dates, social security/social insurance numbers, and addresses.

The philosophy behind the protection of PII is that Banner users should be permitted to access personal information for only those people they need to see as part of their job responsibilities. For example, a user should only be able to view personal information for student applicants if he/she works in the Admissions Office. Workers in the Admissions Office should not have access to personal information for an institutional employee unless the employee also applies for admissions.

PII is implemented system-wide. All users of the system are restricted by PII except when specifically exempted. PII restrictions are PIDM-based (that is, keyed to Banner's internal person identifier code) and based on only one or two tables in the database.

## PII Security Features

- PII is secured according to a user's access to business processing domains (similar to the System Identification Form (GUASYST)).
- PII security is set to protect the SPRIDEN table as delivered seed data. PII security on other tables is possible.
- Each user can have a default home domain and other accessible domains.
- A new person search form is provided with data display masking and hiding capabilities.
- PII security can be turned on or off for an individual Banner user or Banner object.

The Banner Person Identification table (SPRIDEN) functions as a "gateway" to all personal information in Banner. Each time an ID is entered in a key block that has an ID and name, the form looks up the associated name in SPRIDEN. This name lookup invokes the rules that control PII. If the PII rules prevent a successful name lookup, the form assumes the person does not exist in Banner and prevents any other queries on the form for that ID. By using Oracle's Fine-Grained Access Control (FGAC) to access this "gateway", Banner 7.0 restricts users' access to the records of those individuals they need

to see to perform their job responsibilities. Together, FGAC and the “gateway” concept allow Banner to implement this new security feature without modifying each individual object within Banner.

Domains define the processing areas within Banner such as student admissions, student recruiting, student registration, accounts receivable, and HR employment. Each domain has a key or driving table that identifies the people belonging to that group. For example, the driving table for student admissions applicants is the SARADAP table. In addition to the students, employees, and alumni that reside in a domain, Banner users can be assigned to PII domains depending on their job responsibilities. For a user to have access to a person’s data, the user must be assigned to a PII domain that includes the person’s record.

## Example of PII

Assume you want to restrict access to ID and name information based on the work location of the user and functional location of data for the ID. Staff in Admissions, Registration, Student Billing and Student Dean’s offices are assigned functional assignments of Admissions, Enrollment, General Student. Staff in payroll are assigned functional assignments of Applicant, Benefits and Beneficiary. When Staff in Admissions opens a Banner form and enters an ID that does not have a record in Admissions, Enrollment, or General Student, it will look as if the ID does not exist in the system. When staff in the Payroll area open a Banner page and enter an ID that does not have a record in any of the HR areas, it will again look as if the ID does not exist for the system.

Typically, the student administrative offices do not have access to Payroll type forms, and payroll may not have access to student administrative forms. That alone prevents access to the data in those products. PII goes one step farther and prevents these offices from viewing the lists of people that exist for the other products.

## Masking, Hiding, and Concealing Fields

---

The Protection of Sensitive Data menu offers another approach to selectively limit users’ access to data. This feature is designed to partially or fully obscure data in specific fields on Banner forms, using the techniques of masking, hiding, and concealing fields.

For a specified subset of users, all users, or an individual user, a data field can be:

- hidden, so that it is not visible on the form at all
- concealed, replaced by (for example) \*\*\*\*\*
- masked, so that only part of the data appears, for example, \*\*\*\*\*520

The Protection of Sensitive Data features are not based on Oracle’s FGAC feature, but are designed to help accomplish some of the same goals as VBS and PII: limiting specific users’ access to specific data. In this case, the limitation is applied field by field at the Banner form.

# FGAC Exemptions

---

FGAC can be turned off for batch processes and reports. Many summary reports must be able to “see” all records in a table to provide the necessary information regardless of who is actually running the report.

All baseline batch processes and reports are set up in seed data (originally delivered with Banner 7.0) as exempt from FGAC processing. It is important that these processes be exempt to ensure that the data presented on these reports/processes is complete and can be used for its intended purposes. Individual jobs can have FGAC turned back on where appropriate. FGAC restrictions will apply to institutionally developed ad hoc reports. These reports can be exempted where necessary.

Some processes in Banner need to have FGAC turned off temporarily while their work is being done, and then have FGAC re-established when their work is finished. Common Matching is one example. Because the overall objective of Common Matching is the prevention of multiple PIDMs, users running this process must be able to “see” all the existing identification records regardless of specific FGAC rules that may otherwise restrict what the user can “see.” Another example is the registration process so enrollment counts are not impacted. FGAC is also temporarily turned off for application status calculations, concurrent curricula conversions, and calculations of new record sequence numbers in admissions, recruiting, and accounting.

## Cross Domain Search Forms

There are five forms that are exempt from VBS and PII FGAC. Users have access to view all PII data on these forms. Additional access is available to the user if the **Cross Domain Search Indicator for Personal Identification Information** checkbox is checked for their ID on the FGAC Person User Defaults Form (GOAFPUD).

If the user calls one of these forms from another form, all available records are visible, regardless of the user's PII restrictions. The user can select a record that they would normally not have had access to and bring it back to the original form.

The five forms that provide temporary access to an ID are:

- Finance Person Search Form (FOIIDEN)
- Common Matching Entry Form (GOAMTCH)
- SSN/SIN Alternate Search Form (GUIALTI)
- Person Search Form (SOAIDEN)
- System Identification (GUASYST)

Applying PII restrictions to these forms could have undesirable consequences. For example, a user might create a duplicate person record because the existing record for that person was not visible to the user.

You can turn off the **Cross Domain Search Indicator** checkbox for individual users who do not need full access to information on these five forms.

In addition, you can use the Protection of Sensitive Data module to mask part of the information that appears on these forms. For example, you could mask all but the day and month of the birthdate.

## FGAC Information for End Users

---

FGAC is designed to be transparent to end users. Users will simply not see any data that VBS and PII rules are preventing them from seeing. From the user's point of view, it will appear as though the restricted data is not in the database to begin with. If the VBS and PII rules are carefully set up according to each user's responsibilities, this shouldn't present a problem.

But there are times when the effects of FGAC may mislead users and lead to incorrect conclusions. (For example, a user who didn't realize their data view is limited by FGAC rules might say, "That's strange—all of the college's newly enrolled students are in my department!")

Banner provides a few tools to help users be aware of how FGAC is limiting the data they can see.

- A new form displays all the policies that are in effect for the user. The FGAC Predicate Inquiry Form (GOIFGAC) displays the status of a policy and the user's entire predicate for a table.
- A new toolbar icon reflects the status of active policies. If the icon shows up as active (instead of grayed out), FGAC security is in place for you as a user. You can select this icon to navigate directly to the GOIFGAC form for the details.
- When using self-service pages with VBS activated, a warning message is visible to the end user if VBS policies are in effect for the web session.

## FGAC in Integrated SunGard Higher Education Products

---

Because FGAC restrictions are enforced within Banner's Oracle database, they are automatically applied to any standard or customized tool or application when it accesses the Banner database.

### Self-Service Products

Because FGAC is implemented at the database level, it can be used in the Self-Service products when the user queries data about other people, such as with Faculty and Advisor Self-Service and Advancement. FGAC is not intended for self-service pages where users look at their own data, such as Student Self-Service and Employee Self-Service.

## Virtual Private Database (VPD)

Oracle's Virtual Private Database (VPD) is the partitioning of a database by some overriding code, such as a code for campus or institution. VPD, just like VBS and PII, is implemented through FGAC, but there are important differences. When a database table is partitioned with VPD, the data is sliced across an entire population or product line. Users do not know the other user groups exist.

Some security goals could be implemented by either VBS or VPD. The choice of which to use should be made after a thorough analysis of your institutional needs.

Consider the following comparisons:

- VPD uses a new column to secure data. VBS uses existing columns to secure data.
- VPD works for any group of related tables. VBS is appropriate for sets of tables that contain the data necessary for security.
- VPD is best for large-scale segmentation of data. VBS is best for securing small sets of data.
- An example of a VPD is a shared Student database with Financial Aid virtual private databases. An example of VBS is in-state and out-of-state admissions officers.

VBS and VPD can co-exist comfortably. Each generates its own FGAC predicate statement. Oracle combines multiple predicates at runtime with **AND** logic. This means that both sets of restrictions are applied before data is returned to the user.

In Banner, VPD is used to implement Multi-Entity Processing (MEP). For more information on VPD and MEP, refer to the *Banner Security Administration Handbook*.

## Banner Workflow and Banner Document Management Suite

Banner Workflow and Banner Document Management Suite (BDMS) store their own representation of data outside the core Banner tables, such as Workflow context data and BDMS metadata. Because of this, these products do not automatically enforce the FGAC data restrictions set up in Banner.

## Operational Data Store and Enterprise Data Warehouse

Both Operational Data Store (ODS) and Enterprise Data Warehouse (EDW) use Oracle's FGAC to implement Value-Based Security, but their implementation is separate from the Banner implementation. These products use a separate database with different data structures, so there is no easy way for ODS/EDW to implement the FGAC rules that are set up in Banner.

## e~Print Reports

Reports are run in Banner and then the output is sent to e~Print. Because reports are initially delivered as exempt from FGAC, there is no impact to e~Print.

The e~Print Virtual Bursting feature for Finance Fund/Orgn security and HR security remains intact. For example, e~Print uses login data to retrieve a user's Fund/Orgn access data from Banner and dynamically create a report containing only the pages the user is allowed to view. You can keep one copy of a monthly report and let e~Print manage the pages a user is allowed to view.

## Banner Voice Response

Because Voice Response deals purely with the user's own personal data, there is no impact.

## Luminis Channels for Banner and Luminis Data Integration (LDI)

Banner, since Release 7.0, uses the mid-tier proxy authentication feature, allowing interfaces such as Luminis® Channels for Banner and LDI to connect to the database as the “real” end-user. This allows all FGAC-based security features, such as VBS and VPD, to work for these interfaces the same way they work for direct user logins. Mid-tier proxy authentication creates light-weight database sessions, with a proxy user connecting on behalf of the actual user. This authentication is available to any interface that can connect to the database through OCI (Oracle Call Interface) or JDBC.

An example is a Luminis portal user who also has a database user ID. The Luminis LDAP server already provides secure authentication. Once a user logs into Luminis, any channels on the user's home page can safely connect to the database through the proxy user as the end-user without requiring additional authentication. The VBS policies are applied while displaying information for the channel.

## Banner Forms for FGAC

---

Information on Banner forms and fields can be found in Banner Online Help. The help system runs in a web browser, and is available by choosing the **Online Help** button on the toolbar or the Online Help option from the Help pull-down menu. Additionally, you can access online help by selecting the **Help Center** link on the main menu.

Beginning with Release 8.0, Online Help includes a PDF version of form and field reference, which you can print or save to your local computer.

Banner's FGAC menus appear on the System Functions/Administration Menu. The menus and forms are organized as indicated below.

## Fine-Grained Access Control (\*GENFGAC)

- FGAC Group Validation (GTVFGAC)
- FGAC Group Rules (GOAFGAC)
- FGAC Predicate Inquiry (GOIFGAC)
- FGAC Excluded Objects (GORFEOB)
- FGAC Business Profile Validation (GTVFBPR)
- FGAC Business Profile Assignments (GOAFBPR)
- FGAC Business Profile Inquiry (GOIFBPR)
- Enterprise Access Controls (GOAEACC)

## FGAC Value-Based Security Maintenance (\*GENFGBS)

- FGAC Domain Validation (GTVFDMN)
- FGAC Domain Driver Rules (GORFDMN)
- FGAC VBS Table Rules (GORFDPL)
- FGAC Domain Type Validation (GTVFDTP)
- FGAC Domain Type Predicate Rules (GORFDTP)

## FGAC Personal Identification Information Maintenance (\*GENFGPII)

- FGAC Domain Validation (GTVFDMN)
- FGAC Domain Driver Rules (GORFDMN)
- FGAC PII Table Rules (GORFDPI)
- FGAC Person User Defaults (GOAFPUD)
- FGAC PII Domain Business Profile Assignments (GOAFBPI)
- PII Domain Maintenance (GOAFPPII)

## Protection of Sensitive Data (\*GENDMSK)

- Data Display Mask Columns (GORDMCL)
- Data Display Mask Rules (GORDMSK)

# FGAC Glossary

---

## Business Profile

Grouping system introduced to help reduce data entry. Allows users who have the same access restrictions to be grouped together.

## Context Variables

Oracle global variables that are stored in the database context area. Context variables can be created dynamically, and are active throughout a user's Oracle session. The ones used for VBS and PII processing are defined for usage in the GOKFGAC package and are used to store information to eliminate repeat database activity.

## Domain

Area in Banner that has a common driving table. An example of a VBS domain is Student Admissions, where the driver is the Admissions Application Table (SARADAP). All lesser tables in Admissions are part of the domain and will follow restrictions based on the rules defined for the Admissions domain. The PII domains define where the PIDM must exist for the user to have access to that person and are similar to what the System Identification Form (GUASYST) defines.

## FGAC

Fine-grained access control. Oracle functionality that provides the ability to define access policies that control the access of individuals or groups of users to views and tables.

FGAC dynamically modifies an SQL statement in a transparent way to users by adding a condition (*WHERE* clause) which restricts the rows shown to the user.

## Field-Level Security

Hides fields and/or masks the display of fields on a form from a specific user.

## GOKFGAC

The General package that dynamically constructs the VBS or PII predicate. The GOKFGAC functions are executed by the Oracle policy and return the predicate to Oracle.

## PII

Personally Identifiable Information. PII domains secure General Person identifiable information.

A user can access PII for records in their processing area (for example, a user can only view the PII for student applicants when in Admissions).

## Predicate Functions

Predicate functions are database functions that establish the policy statement for a table and database activity. In Banner, these database packages organize the predicate that is used by FGAC for row-level security. The return from the predicate is the addition to the SQL `WHERE` clause that will restrict row level access.

## Row-Level Security

Restricts rows shown to the user. For example, row level security for SARADAP could restrict a user to only viewing rows where `SARADAP_COLL_CODE = AS`.

## Banner Object-Level Security

Restriction of the Banner objects a user can access, or a restriction of the access mode (Query or Update). Maintenance is performed on the GSASECR Form.

## VBS

Value-Based Security. An optional security feature that allows installations to secure information based on specific data values in database columns. For example, an institution may offer undergraduate and graduate programs with separate admissions offices for undergraduate and graduate processing. VBS allows the institution to restrict access to applications (which are stored in a single application table) so that personnel in the Undergraduate Admissions Office can access only undergraduate applications and personnel in the Graduate Admissions Office can access only graduate applications.

## VPD

Virtual Private Database. An Oracle feature that involves partitioning a database by some overriding code, such as a campus or institution code. VPD uses FGAC in a way that slices the database across an entire population or product line. When VPD is in place, users do not know the other user groups exist.

# Value-Based Security

---

This chapter details the setup process for Value-Based Security (VBS). For each VBS domain, you will follow four basic steps:

- [“Step 1—Define the Domain and Its Driver Table” on page 27](#)
- [“Step 2—Define Policy Tables for the Domain” on page 27](#)
- [“Step 3—Create Oracle Policies” on page 29](#)
- [“Step 4—Set Up VBS Rules” on page 31](#)

Before VBS setup is begun, thorough planning and analysis should be done to understand the business need for VBS, propose a set of rules, and make sure the rules will satisfy the business need with no unintended side effects. Careful testing should be done on a new set of VBS rules before rolling them out to your users.



**Note:** For the Banner® forms that support VBS setup, see [“Banner Forms for FGAC” on page 19](#).

## Business Need Analysis for VBS

---

Careful analysis (functional and technical) needs to be done, together with intensive testing, before rolling out a new VBS rule. Because the restriction is placed on a table, adequate analysis needs to be done on the programs (forms, Web pages, etc.) that use the table, to understand the full impact of the restriction on end users.

Use the template and examples in this chapter to help you analyze your business case, objects impacted, and required SQL to implement a VBS restriction. SunGard® services will also be available to help you with your analysis.

You should also make your end users aware of the implications of the restrictions that are created for them.

## Analysis Worksheet Template

A Business Need Analysis Worksheet has been created to help guide the analysis process that should precede setting up a new VBS rule. The worksheet is a tool to help organize your thinking and complete a thorough analysis.

The examples that follow will illustrate how the worksheet is used.

In this template, [] indicates required information, while <> is optional)

<b>Domain: GTVFDMM, GORFDMM</b>	<b>Driver: GORFDMM</b>
[product abbrv + B_entity name_VBS]	[driver table]

<b>Business Case: GOAFGAC</b>	<b>SQL Predicate</b>

<b>Table: GORFDPL</b>	<b>Join</b>
[driver table]	
<dependent table>	<join between dependent table and driver table>

<b>Banner Object</b>	<b>Restriction checked on GOAFGAC</b>	<b>Possible Impacts</b>

## Worksheet Example 1

The following worksheet illustrates the components required for example VBS business cases. This example will be followed in the first VBS setup steps below. First the domain and its driver table are defined, the domain tables are listed, and the necessary SQL statements are identified to join each domain table to the domain's driver table.

The last part of the worksheet charts all places where the rule may impact an end user's processing. JOBS type objects are not included. The following examples provide only a portion of all objects that may possibly be impacted by the restriction. It is recommended that the Banner Enlighten product be used to find all places the tables are used.

<b>Domain:</b>	<b>Driver:</b>
SB_ADMISSIONS_VBS	SARADAP

Table	Join
SARADAP	
SARCHKL	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARCHKL_PIDM AND SARADAP_TERM_CODE_ENTRY = SARCHKL_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SARCHKL_APPL_NO)
SARQUAN	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARQUAN_PIDM AND SARADAP_TERM_CODE_ENTRY = SARQUAN_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SARQUAN_APPL_NO)

Business Case	SQL Predicate
Application data entry clerks in Admissions file readers in Admissions are assigned by applicants residency code	SARADAP_RESID_CODE = 'xxx'
Application reviewers assigned applications by admissions status	SARADAP_ADMT_CODE = 'xxx'

Banner Object	Restriction checked on GOAFGAC	Possible Impacts
SAAADMS	Insert	Will not be able to insert applications
SAAADMS	Delete	Will not be able to insert applications
SAAADMS	Update	Will not be able to insert applications
SAAADMS	Select	Will not be able to insert applications
SAAACKL	Insert	Will not be able to insert application checklist items
SAAACKL	Update	Will not be able to update application checklist items

## Worksheet Example 2

This second example shows the worksheet being used to set up VBS rules based on address types.

Domain:	Driver:
GB_SPRADDR_VBS	SPRADDR

Table	Join
SPRADDR	

Business Case	SQL Predicate
Payroll can insert and update W2, mailing and permanent addresses.	SPRADDR_ATYP_CODE in ('W2', 'PR', 'MA')
Student Offices can insert and update parents, mailing and permanent addresses.	SPRADDR_ATYP_CODE in ('PA', 'PR', 'MA')
Student Bursar Office can insert and update billing, permanent and mailing addressES.	SPRADDR_ATYP_CODE in ('BI', 'PR', 'MA')

Banner Object	Restriction checked on GOAFGAC	Possible Impacts
SPAIDEN	Select	Cannot select addresses on an Address tab
SPAIDEN	Insert	Cannot insert addresses with certain types on an Address tab

# Setup Steps for VBS

---

## Step 1—Define the Domain and Its Driver Table

A series of domain codes are delivered as seed data, but if you need to add or modify a domain code, use the FGAC Domain Validation Form (GTVFDMN).

Domains define the driver for your VBS rules. Each domain has a driver table associated with a VBS rule. The domain is usually the parent table of a module.

The naming convention for a domain code is as follows: *SB\_ADMISSIONS\_VBS*, where *S* represents the Banner product, which in this case is Student, *B* is for Banner, *ADMISSIONS* is the entity name, and *VBS* is the domain type.

1. Define domain codes and descriptions on the FGAC Domain Validation Form (GTVFDMN).
2. Enter the domain driver table definition on the FGAC Domain Code Rules Form (GORFDMN). The domain driver includes the driver table and the type of VBS.

## Driver Tables with Sequence Numbers

Use caution if the domain driver table has a sequence number as part of the primary key and is incremented using addition based on maximum number of existing rows. This may present key constraint issues.

Consider an example where the domain driver is SARADAP. The primary key includes SARADAP\_APPL\_NO, which is incremented by 1 based on the number of existing applications for the person. A user can only select, insert, update, and delete applications with college code AS. The applicant already has an application number 1 for college ED. The user tries to insert a new application for college AS but the process does not find existing applications and thinks the application number is 1. This causes a foreign key constraint because the application number 1 already exists.

In Banner 7.0 the SARADAP processing is modified so the application increment process turns FGAC off; calculates the new application number, and then turn FGAC back on. Other tables in Banner Student that follow this pattern are SRBRECR, SORLCUR and SORLFOS.

## Step 2—Define Policy Tables for the Domain

Domain policy tables define the parameters for creating an Oracle Policy. Each domain has at least one policy table, which is its driver table. Additional domain policy tables for the domain control additional areas where the VBS restriction should execute. The policy tables defined for a domain carry the same restrictions that are defined for the driver table.

## Example—Admissions Domain

The Admissions domain uses SARADAP as the driver table. Additional child tables for Admissions are SARCHKL and SARQUAN. When a VBS restriction is built for the Admissions domain, access to SARCHKL and SARQUAN are defined by the restriction on SARADAP.

1. List the tables that are either children of domain driver or will follow the same restrictions as the domain driver.
2. Write Join SQL between Domain Driver and Policy Tables. For each table, write the partial SQL statement that will join the table to the driver table.

For example, if the table is SARCHKL and the domain driver is SARADAP, the VBS rule is: SARADAP\_RESD\_CODE = 'M'.

A normal query to join SARCHKL and SARADAP is the following subquery:

```
SELECT 'X' FROM SARCHKL
WHERE EXISTS
  ( SELECT 'X' FROM SARADAP
    WHERE SARADAP_PIDM = SARCHKL_PIDM
    AND SARADAP_TERM_CODE_ENTRY = SARCHKL_TERM_CODE_ENTRY
    AND SARADAP_APPL_NO = SARCHKL_APPL_NO)
```

The partial driver SQL to enter for this driver SQL join is:

```
EXISTS
  ( SELECT 'X' FROM SARADAP
    WHERE SARADAP_PIDM = SARCHKL_PIDM
    AND SARADAP_TERM_CODE_ENTRY = SARCHKL_TERM_CODE_ENTRY
    AND SARADAP_APPL_NO = SARCHKL_APPL_NO)
```

Eventually, when the predicate is built, the partial driver SQL is appended to the executing SQL statement along with the SARADAP\_RESD\_CODE = 'M' VBS rule.

The code required for setup is as follows

SQL Predicate	Source
Select sarchkl_pidm, sarchkl_admr_code	
From sarchkl	
Where sarchkl_pidm = :pidm	
And sarchkl_term_code_entry = :term_in	User executes this SQL query
AND	Oracle automatically connects the SQL query to the FGAC predicate with an AND

## SQL Predicate

## Source

```
Exists ( select 'x' from
saradap Where saradap_pidm =
sarchkl_pidm And
saradap_term_code_entry =
sarchkl_term_code_entry and
saradap_appl_no =
sarchkl_appl_no
```

```
And saradap_resd_code = 'M'
```

```
)
```

The join statement you defined for the domain table is appended

The VBS rule coded on GOAFGAC. Note that the **AND** is automatically added and should not be coded on the predicate.

GOKFGAC concludes the predicate with `)` if necessary



**Note:** This does not include the embedded comments that GOKFGAC will add.

3. Enter all domain policy tables on the FGAC VBS Tables Rule Form (GORFDPL).

Note the function of the **Active Indicator**. You can create the Oracle policy and uncheck this checkbox to inactivate all rules for this domain and table. The GOKFGAC process does not pick up rules for a predicate if the **Active Indicator** on the table is unchecked.

The **Driver SQL** is the partial SQL that joins the table to the domain driver.

## Step 3—Create Oracle Policies

This part of the process creates the Oracle policies for all tables defined for the domain.

### Using BANINST1 to run gfvbsaddpol.sql

You must use the BANINST1 User ID when you run the script to create the policies.

The predicate package owner must equal the policy owner. In this case, GOKFGAC is owned by BANINST1.

From SQLPlus run the `gfvbsaddpol.sql` script while logged in with the BANINST1 User ID. You are prompted for a table name (you can use wild cards).

The `gfvbsaddpol.sql` script is located in the Banner General Plus directory.

This step has to be done only once per table. This step does not need to be repeated if the `gorfdpl_driver.sql` script changes, if the table is added to another domain, or if there is any change to the domain and table.



**Note:** The `gfvbsaddpol.sql` script is rerunnable and restartable. The process will not try to recreate a policy. If the policy exists, the step to add the policy is skipped.

## Viewing Policy Data from SQLPlus

GORFDPL creates 4 policies per table, separate ones for insert, update, delete and select. All have the package defined as GOKFGAC and must be owned by BANINST1.

### Policy names

Function, `gokfgac.f_get_policy_name`, ensures that the policy name created is less than 30 characters:

- If the table name is less than 19 characters, the policy name will use the following pattern:

```
GOKFGAC_tablename_INS (INS, UPD, DEL, SEL)
```

- If the table name length is 19 or greater characters, the policy name will use the following pattern:

```
tablename_I (I, U, D, S)
```

```
SQL> select * from dba_policies where
2 object_name = 'SSBSECT';
```

OBJECT_OWNER	OBJECT_NAME	POLICY_GROUP
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_DEL	BANINST1	GOKFGAC
F_DELETE_FNC	NO NO NO YES YES YES NO	
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_INS	BANINST1	GOKFGAC
F_INSERT_FNC	NO YES NO NO YES YES NO	
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_SEL	BANINST1	GOKFGAC
F_SELECT_FNC	YES NO NO NO YES YES NO	
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_UPD	BANINST1	GOKFGAC
F_UPDATE_FNC	NO NO YES NO YES YES NO	

## Dropping a Policy

To drop a policy, run the `gfgacdroppol.sql` script from SQLPlus using the BANINST1 ID. This script accepts wild cards for the table name prompt.

## Step 4—Set Up VBS Rules

This requires three steps:

1. Establish the group code on GTVFGAC.
2. Enter the rule code, domains, and the predicate and users on GOAFGAC.
3. View the resulting predicate for the table and User ID on GOIFGAC.

The rule code defined on GTVFGAC is designed to resemble a header for common business cases. After a thorough examination of the relationship between the FGAC rules and the domains, you can start to organize your business cases into rules.

### Adding a Group Code

Use the FGAC Group Validation Form (GTVFGAC) to add a group code. A group is a way to group a set of rules together.

**Example:** An institution has two colleges: College of Arts and College of Education. Create a group code for each college.

### Setting up the Predicate for the Rule on the GOAFGAC Form

A predicate is partial SQL. The validate SQL button creates a SQL statement such as: `Select 'x' from <domain driver> where <predicate>` and executes the code dynamically.

You can perform the following functions at this point.

1. Inactive all predicates defined in the rule, or future date a rule
2. Enter different predicates for the same domain. This may be necessary if different populations have differing access requirements.
3. Enter multiple domains with different predicates on the same rule.
4. Test your predicate using the **Validate SQL** button.
5. Use the pull-down fields to list tables, columns and operators.

### Defining Access to the Predicate

On the Business Predicate tab, enter Business Profile access to the predicate, or specific user access to the predicate. You must check select, insert, update or delete to indicate the access restriction. The policy for each of these CRUD is always created by `gfvbsaddpol.sql`. Checking these fields establishes the restriction for the particular rule and predicate.

## Testing the Predicate on the FGAC Predicate Inquiry Form (GOIFGAC)

The GOIFGAC form displays the status of the policy and gives you the entire predicate for a table. Enter a user ID and table name in the Key Block. The GOKFGAC package embeds comments in the predicate to tell you the domain name and the rule names so you can easily trace the source of the SQL.

## Building Predicate Logic

---

This section explains some of the relationship between the FGAC rules and domains. Keep in mind that the configuration of the rules and domains is extremely flexible and carries inherent risks.

Here are a few points to consider.

1. The parsed outcome from multiple rules and domains for a single table is combined with ORs and ANDs depending on GOAFGAC rules and GORFDMN domain definitions.

All rules for a domain use ANDs to combine a domain predicate. All domain predicates use ORs.

2. The GOKFGAC parser embeds comments about the domain name and the FGAC rules name. This is meant to aid in debugging and in locating the predicate on GOAFGAC.
3. ORACLE always appends a parsed predicate with an AND to the executing SQL statement.

The following example shows the template of the parsed predicate and illustrates the placement of comments, ANDs, and ORs:

```
<executing SQL statement>
AND (/*Domain 1 */
  (/*Group 1*/ (Predicate 1 AND Predicate 2 AND Predicate
3))
OR (/*Group 2*/ (Predicate 4 AND Predicate 5)))
OR (/*Domain 2*/ (/*Group 3*/...))
```

Predicates parsed from different domains use ORs, as shown in this example:

Example: SARCHKL is in domains SB\_ADMISSIONS\_VBS and SB\_CHECKLIST\_VBS

GOAFGAC Rule 1: Predicate for domain SB\_ADMISSIONS\_VBS is:

SARADAP\_RESD\_CODE = 'M'

GOAFGAC Rule 2: Predicate for domain SB\_CHECKLIST\_VBS is:

SARCHKL\_ADMR\_CODE = 'TAX'

The parser uses a logical OR to tie these two rules together:

( SARADAP\_RESD\_CODE = 'M' ) OR ( SARACHKL\_ADMR\_CODE = 'TAX' )

Predicates from different rules for the same domain use OR, as the next example illustrates.

GOAFGAC Rule 1: Predicate for domain SB\_ADMISSIONS\_VBS is:

SARADAP\_RESD\_CODE = 'M'

GOAFGAC Rule 2: Predicate for domain SB\_ADMISSIONS\_VBS is:

SARADAP\_APST\_CODE = 'C'

The parser uses OR to tie these two together.

SARADAP\_RESD\_CODE = 'M' OR SARADAP\_APST\_CODE = 'C'

Predicates within a single Rule for the same predicate are tied together with AND, as shown in this example:

GOAFGAC Rule 1: Predicate for domain SB\_ADMISSIONS\_VBS is:

SARADAP\_RESD\_CODE = 'M'

Predicate 2 for domain SB\_ADMISSIONS\_VBS is:

SARADAP\_APST\_CODE = 'C'

The parser uses AND to tie the two predicates together:

SARADAP\_RESD\_CODE = 'M' AND SARADAP\_APST\_CODE = 'C'

## Worksheet Example

The following worksheet example illustrates how to identify new rules and make assignments to predicates. Careful thought needs to be given to how populations are assigned to predicates within rules because of the OR/AND connector logic.

Create population lists on the FGAC Business Profile page GOAFBPR:.

Business Reason	Table	Domain	Business Case	VBS Rule	User Population
Secure Addresses by Type	SPRADDR	GB_SPRADDR_V BS	Payroll can insert and update W2, mailing and permanent addresses	SPRADDR_ATYP_CODE in ('W2', 'PR', 'MA')	HR_Data - entry
Secure Addresses by Type	SPRADDR	GB_SPRADDR_V BS	Student Offices can insert and update parents, mailing and permanent addresses.	SPRADDR_ATYP_CODE in ('PA', 'PR', 'MA')	Student - Admin

<b>Business Reason</b>	<b>Table</b>	<b>Domain</b>	<b>Business Case</b>	<b>VBS Rule</b>	<b>User Population</b>
Secure Addresses by Type	SPRADDR	GB_SPRADDR_V BS	Student Bursar Office can insert and update billing, permanent and mailing address	SPRADDR_ATYP_CODE in ('BI', 'PR', 'MA')	Bursar_Admin

## VBS Domains

The following domains are included as seed data in the FGAC Domain Validation Table (GTVFDMN).

The Product Dependency column indicates the products that you must have licensed in order for seed data to be inserted.

<b>VBS Domain</b>	<b>Description</b>	<b>Product Dependency</b>
GB_FGAC_PREDICATE_VBS	FGAC Rules Predicate VBS	General
GB_FGACACCESS_VBS	FGAC Rules Access VBS	General
GB_INTERNATIONAL_VBS	Person International VBS	General
GB_SPRADDR_VBS	Address VBS Security	General
GB_SPRMEDI_VBS	Person Medical VBS Security	General
GB_SPRTELE_VBS	Telephone VBS Security	General
RB_FINAID_STUDENT_VBS	Financial Aid Student VBS Security	Financial Aid and Student
RB_FINAID_VBS	Financial Aid VBS Security	Financial Aid
SB_ADMISSIONS_VBS	Student Admissions VBS	Student
SB_CATALOG_VBS	Student Catalog VBS Security	Student
SB_CURRICULUM_VBS	Student Curriculum VBS Security	Student
SB_FIELDOFSTUDY_VBS	Student Field of Study VBS	Student
SB_LEARNER_VBS	Student Learner VBS Security	Student

VBS Domain	Description	Product Dependency
SB_OTHERGPA_CODES_VBS	Student Other GPA Code VBS Security	Student
SB_OTHERGPA_STUDENT_VBS	Student Other GPA with Student Secured VBS Security	Student
SB_OTHERGPA_VBS	Student Other GPA VBS Security	Student
SB_RECRUIT_VBS	Student Recruit VBS Security	Student
SB_SCHEDULE_VBS	Schedule VBS Security	Student
SB_TESTCODES_VBS	Student Test Code VBS Security	Student
SB_TESTSCORE_STUDENT_VBS	Student Test Score with Student Secured VBS Security	Student
SB_TESTSCORE_VBS	Student Test Score VBS Security	Student

## VBS Domain Driver Tables

The following VBS driver tables were delivered as seed data in the FGAC Domain Driver Rules Table (GOBFDMN). A driver table is included for each VBS domain delivered with the FGAC Domain Validation (GTVFDMN). All domains are delivered as disabled.

These entries are inserted if the domain exists in GTVFDNMN. Please refer to the domains seed data table above for product dependencies.

Domain Code	Domain Type	Driver Table
GB_FGACACCESS_VBS	VBS	GOBFGAC
GB_FGAC_PREDICATE_VBS	VBS	GORFPRD
GB_INTERNATIONAL_VBS	VBS	GOBINTL
GB_SPRADDR_VBS	VBS	SPRADDR
GB_SPRMEDI_VBS	VBS	SPRMEDI
GB_SPRTELE_VBS	VBS	SPRTELE
RB_FINAID_STUDENT_VBS	VBS	RORSTAT
RB_FINAID_VBS	VBS	RORSTAT
SB_ADMISSIONS_VBS	VBS	SARADAP

Domain Code	Domain Type	Driver Table
SB_CATALOG_VBS	VBS	SCBCRSE
SB_CURRICULUM_VBS	VBS	SORLCUR
SB_FIELDOFSTUDY_VBS	VBS	SORLFOS
SB_LEARNER_VBS	VBS	SGBSTDN
SB_OTHERGPA_CODES_VBS	VBS	STVGPAT
SB_OTHERGPA_STUDENT_VBS	VBS	SORGPAT
SB_OTHERGPA_VBS	VBS	SORGPAT
SB_RECRUIT_VBS	VBS	SRBRECR
SB_SCHEDULE_VBS	VBS	SSBSECT
SB_TESTCODES_VBS	VBS	STVTESC
SB_TESTSCORE_STUDENT_VBS	VBS	SORTEST
SB_TESTSCORE_VBS	VBS	SORTEST

## VBS Domain Policy Tables

---

The following is delivered as seed data with an Active indicator of *N* in the FGAC VBS Table Rules Table (GORFDPL).

These entries are inserted if the domain exists in GTVFDMN and the driver exists in GOBFDMN. Please refer to the domains seed data table above for product dependencies.

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
GB_FGACACCESS_VBS	GOBFGAC	(none)
GB_FGAC_PREDICATE_VBS	GORFPRD	(none)
GB_INTERNATIONAL_VBS	GOBINTL	(none)
GB_SPRADDR_VBS	SPRADDR	(none)
GB_SPRMEDI_VBS	SPRMEDI	(none)
GB_SPRTELE_VBS	SPRTELE	(none)

<b>Domain Name</b>	<b>Policy Table</b>	<b>Code that Links Policy Table to Domain Driver Table (GOBFDMN)</b>
RB_FINAID_STUDENT_VBS	RORSTAT	SOKFGAC.F_FIND_IF_STUDENT_FINAID (RORSTAT_PIDM, RORSTAT_AIDY_CODE) = 'Y'
RB_FINAID_VBS	RORSTAT	(none)
SB_ADMISSIONS_VBS	SABSUPL	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SABSUPL_PIDM AND SARADAP_TERM_CODE_ENTRY = SABSUPL_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SABSUPL_APPL_NO
SB_ADMISSIONS_VBS	SARAATT	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARAATT_PIDM AND SARADAP_TERM_CODE_ENTRY = SARAATT_TERM_CODE AND SARADAP_APPL_NO = SARAATT_APPL_NO
SB_ADMISSIONS_VBS	SARACMT	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARACMT_PIDM AND SARADAP_TERM_CODE_ENTRY = SARACMT_TERM_CODE AND SARADAP_APPL_NO = SARACMT_APPL_NO
SB_ADMISSIONS_VBS	SARADAP	(none)
SB_ADMISSIONS_VBS	SARAPPD	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARAPPD_PIDM AND SARADAP_TERM_CODE_ENTRY = SARAPPD_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SARAPPD_APPL_NO
SB_ADMISSIONS_VBS	SARCHKL	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARCHKL_PIDM AND SARADAP_TERM_CODE_ENTRY = SARCHKL_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SARCHKL_APPL_NO
SB_ADMISSIONS_VBS	SARCHRT	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARCHRT_PIDM AND SARADAP_TERM_CODE_ENTRY = SARCHRT_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SARCHRT_APPL_NO
SB_ADMISSIONS_VBS	SARDSCL	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARDSCL_PIDM AND SARADAP_TERM_CODE_ENTRY = SARDSCL_TERM_CODE AND SARADAP_APPL_NO = SARDSCL_APPL_NO

<b>Domain Name</b>	<b>Policy Table</b>	<b>Code that Links Policy Table to Domain Driver Table (GOBFDMN)</b>
SB_ADMISSIONS_VBS	SARQUAN	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARQUAN_PIDM AND SARADAP_TERM_CODE_ENTRY = SARQUAN_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SARQUAN_APPL_NO
SB_ADMISSIONS_VBS	SARRRAT	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARRRAT_PIDM AND SARADAP_TERM_CODE_ENTRY = SARRRAT_TERM_CODE AND SARADAP_APPL_NO = SARRRAT_APPL_NO
SB_ADMISSIONS_VBS	SARRSRC	EXISTS (SELECT 'X' FROM SARADAP WHERE SARADAP_PIDM = SARRSRC_PIDM AND SARADAP_TERM_CODE_ENTRY = SARRSRC_TERM_CODE_ENTRY AND SARADAP_APPL_NO = SARRSRC_APPL_NO
SB_CATALOG_VBS	SCBCRSE	(none)
SB_CURRICULUM_VBS	SORLCUR	(none)
SB_FIELDOFSTUDY_VBS	SORLFOS	(none)
SB_LEARNER_VBS	SGBEOPS	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGBEOPS_PIDM AND SGBSTDN_TERM_CODE_EFF = SGBEOPS_TERM_CODE_EFF
SB_LEARNER_VBS	SGBOEDU	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGBOEDU_PIDM AND SGBSTDN_TERM_CODE_EFF = SGBOEDU_TERM_CODE_EFF
SB_LEARNER_VBS	SGBSTDN	(none)
SB_LEARNER_VBS	SGBUSER	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGBUSER_PIDM AND SGBSTDN_TERM_CODE_EFF = SGBUSER_TERM_CODE
SB_LEARNER_VBS	SGRACMT	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRACMT_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRACMT_TERM_CODE_EFF
SB_LEARNER_VBS	SGRADVR	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRADVR_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRADVR_TERM_CODE_EFF

<b>Domain Name</b>	<b>Policy Table</b>	<b>Code that Links Policy Table to Domain Driver Table (GOBFDMN)</b>
SB_LEARNER_VBS	SGRASSI	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRASSI_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRASSI_TERM_CODE_EFF
SB_LEARNER_VBS	SGRASST	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRASST_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRASST_TERM_CODE_EFF
SB_LEARNER_VBS	SGRCHRT	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRCHRT_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRCHRT_TERM_CODE_EFF
SB_LEARNER_VBS	SGRCMNT	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRCMNT_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRCMNT_TERM_CODE
SB_LEARNER_VBS	SGRCOOP	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRCOOP_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRCOOP_TERM_CODE
SB_LEARNER_VBS	SGRDISA	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRDISA_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRDISA_TERM_CODE
SB_LEARNER_VBS	SGRDSER	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRDSER_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRDSER_TERM_CODE
SB_LEARNER_VBS	SGRDUTY	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRDUTY_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRDUTY_TERM_CODE
SB_LEARNER_VBS	SGRESEL	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRESEL_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRESEL_TERM_CODE_EFF
SB_LEARNER_VBS	SGRSACT	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRSACT_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRSACT_TERM_CODE

<b>Domain Name</b>	<b>Policy Table</b>	<b>Code that Links Policy Table to Domain Driver Table (GOBFDMN)</b>
SB_LEARNER_VBS	SGRSATT	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRSATT_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRSATT_TERM_CODE_EFF
SB_LEARNER_VBS	SGRSCMT	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRSCMT_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRSCMT_TERM_CODE
SB_LEARNER_VBS	SGRSPRT	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRSPRT_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRSPRT_TERM_CODE
SB_LEARNER_VBS	SGRVETN	EXISTS (SELECT 'X' FROM SGBSTDN WHERE SGBSTDN_PIDM = SGRVETN_PIDM AND SGBSTDN_TERM_CODE_EFF = SGRVETN_TERM_CODE_VA
SB_OTHERGPA_CODES_VBS	STVGPAT	(none)
SB_OTHERGPA_STUDENT_VBS	SORGPAT	SOKFGAC.F_FIND_IF_STUDENT(SORGPAT_PIDM) = 'Y'
SB_OTHERGPA_VBS	SORGPAT	(none)
SB_RECRUIT_VBS	SRBRECR	(none)
SB_RECRUIT_VBS	SRRCHRT	EXISTS (SELECT 'X' FROM SRBRECR WHERE SRBRECR_PIDM = SRRCHRT_PIDM AND SRBRECR_TERM_CODE = SRRCHRT_TERM_CODE AND SRBRECR_ADMIN_SEQNO = SRRCHRT_ADMIN_SEQNO
SB_RECRUIT_VBS	SRRLEND	EXISTS (SELECT 'X' FROM SRBRECR WHERE SRBRECR_PIDM = SRRLEND_PIDM AND SRBRECR_TERM_CODE = SRRLEND_TERM_CODE AND SRBRECR_ADMIN_SEQNO = SRRLEND_ADMIN_SEQNO
SB_RECRUIT_VBS	SRRRATT	EXISTS (SELECT 'X' FROM SRBRECR WHERE SRBRECR_PIDM = SRRRATT_PIDM AND SRBRECR_TERM_CODE = SRRRATT_TERM_CODE AND SRBRECR_ADMIN_SEQNO = SRRRATT_ADMIN_SEQNO

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_RECRUIT_VBS	SRRRSRC	EXISTS (SELECT 'X' FROM SRBRECR WHERE SRBRECR_PIDM = SRRRSRC_PIDM AND SRBRECR_TERM_CODE = SRRRSRC_TERM_CODE AND SRBRECR_ADMIN_SEQNO = SRRRSRC_ADMIN_SEQNO)
SB_SCHEDULE_VBS	SSBDESC	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSBDESC_CRN AND SSBSECT_TERM_CODE = SSBDESC_TERM_CODE)
SB_SCHEDULE_VBS	SSBFSEC	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSBFSEC_CRN AND SSBSECT_TERM_CODE = SSBFSEC_TERM_CODE)
SB_SCHEDULE_VBS	SSBOVRR	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSBOVRR_CRN AND SSBSECT_TERM_CODE = SSBOVRR_TERM_CODE)
SB_SCHEDULE_VBS	SSBSECT	(none)
SB_SCHEDULE_VBS	SSBSSEC	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSBSSEC_CRN AND SSBSECT_TERM_CODE = SSBSSEC_TERM_CODE)
SB_SCHEDULE_VBS	SSRATTR	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRATTR_CRN AND SSBSECT_TERM_CODE = SSRATTR_TERM_CODE)
SB_SCHEDULE_VBS	SSRBLCK	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRBLCK_CRN AND SSBSECT_TERM_CODE = SSRBLCK_TERM_CODE)
SB_SCHEDULE_VBS	SSRCORQ	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRCORQ_CRN AND SSBSECT_TERM_CODE = SSRCORQ_TERM_CODE)
SB_SCHEDULE_VBS	SSREVAL	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSREVAL_CRN AND SSBSECT_TERM_CODE = SSREVAL_TERM_CODE)

<b>Domain Name</b>	<b>Policy Table</b>	<b>Code that Links Policy Table to Domain Driver Table (GOBFDMN)</b>
SB_SCHEDULE_VBS	SSREXTN	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSREXTN_CRN AND SSBSECT_TERM_CODE = SSREXTN_TERM_CODE)
SB_SCHEDULE_VBS	SSRFEEES	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRFEEES_CRN AND SSBSECT_TERM_CODE = SSRFEEES_TERM_CODE)
SB_SCHEDULE_VBS	SSRLINK	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRLINK_CRN AND SSBSECT_TERM_CODE = SSRLINK_TERM_CODE)
SB_SCHEDULE_VBS	SSRMEET	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRMEET_CRN AND SSBSECT_TERM_CODE = SSRMEET_TERM_CODE)
SB_SCHEDULE_VBS	SSRMPRT	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRMPRT_CRN AND SSBSECT_TERM_CODE = SSRMPRT_TERM_CODE)
SB_SCHEDULE_VBS	SSRMRDF	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRMRDF_CRN AND SSBSECT_TERM_CODE = SSRMRDF_TERM_CODE)
SB_SCHEDULE_VBS	SSRRARE	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRARE_CRN AND SSBSECT_TERM_CODE = SSRRARE_TERM_CODE)
SB_SCHEDULE_VBS	SSRRCLS	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRCLS_CRN AND SSBSECT_TERM_CODE = SSRRCLS_TERM_CODE)
SB_SCHEDULE_VBS	SSRRCMP	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRCMP_CRN AND SSBSECT_TERM_CODE = SSRRCMP_TERM_CODE)
SB_SCHEDULE_VBS	SSRRCOL	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRCOL_CRN AND SSBSECT_TERM_CODE = SSRRCOL_TERM_CODE)

<b>Domain Name</b>	<b>Policy Table</b>	<b>Code that Links Policy Table to Domain Driver Table (GOBFDMN)</b>
SB_SCHEDULE_VBS	SSRRDEG	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRDEG_CRN AND SSBSECT_TERM_CODE = SSRRDEG_TERM_CODE)
SB_SCHEDULE_VBS	SSRRESV	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRESV_CRN AND SSBSECT_TERM_CODE = SSRRESV_TERM_CODE)
SB_SCHEDULE_VBS	SSRRFND	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRFND_CRN AND SSBSECT_TERM_CODE = SSRRFND_TERM_CODE)
SB_SCHEDULE_VBS	SSRRLVL	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRLVL_CRN AND SSBSECT_TERM_CODE = SSRRLVL_TERM_CODE)
SB_SCHEDULE_VBS	SSRRMAJ	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRMAJ_CRN AND SSBSECT_TERM_CODE = SSRRMAJ_TERM_CODE)
SB_SCHEDULE_VBS	SSRRPRG	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRPRG_CRN AND SSBSECT_TERM_CODE = SSRRPRG_TERM_CODE)
SB_SCHEDULE_VBS	SSRRSTS	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRSTS_CRN AND SSBSECT_TERM_CODE = SSRRSTS_TERM_CODE)
SB_SCHEDULE_VBS	SSRRTST	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRRTST_CRN AND SSBSECT_TERM_CODE = SSRRTST_TERM_CODE)
SB_SCHEDULE_VBS	SSRSCCD	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRSCCD_CRN AND SSBSECT_TERM_CODE = SSRSCCD_TERM_CODE)
SB_SCHEDULE_VBS	SSRSPRT	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRSPRT_CRN AND SSBSECT_TERM_CODE = SSRSPRT_TERM_CODE)

<b>Domain Name</b>	<b>Policy Table</b>	<b>Code that Links Policy Table to Domain Driver Table (GOBFDMN)</b>
SB_SCHEDULE_VBS	SSRSRDF	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRSRDF_CRN AND SSBSECT_TERM_CODE = SSRSRDF_TERM_CODE)
SB_SCHEDULE_VBS	SSRSYLN	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRSYLN_CRN AND SSBSECT_TERM_CODE = SSRSYLN_TERM_CODE)
SB_SCHEDULE_VBS	SSRSYLO	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRSYLO_CRN AND SSBSECT_TERM_CODE = SSRSYLO_TERM_CODE)
SB_SCHEDULE_VBS	SSRSYRM	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRSYRM_CRN AND SSBSECT_TERM_CODE = SSRSYRM_TERM_CODE)
SB_SCHEDULE_VBS	SSRSYTR	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRSYTR_CRN AND SSBSECT_TERM_CODE = SSRSYTR_TERM_CODE)
SB_SCHEDULE_VBS	SSRTEXT	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRTEXT_CRN AND SSBSECT_TERM_CODE = SSRTEXT_TERM_CODE)
SB_SCHEDULE_VBS	SSRXLST	EXISTS (SELECT 'X' FROM SSBSECT WHERE SSBSECT_CRN = SSRXLST_CRN AND SSBSECT_TERM_CODE = SSRXLST_TERM_CODE)
SB_TESTCODES_VBS	STVTESC	(none)
SB_TESTSCORE_ STUDENT_VBS	SORTEST	SOKFGAC.F_FIND_IF_STUDENT(SORTEST_P IDM) = 'Y'
SB_TESTSCORE_VBS	SORTEST	(none)

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCBSUPP	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCBDESC_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCBDESC_SUBJ_CODE AND SCBCRSE_EFF_TERM =( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCBDESC_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCBDESC_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCBDESC_TERM_CODE_EFF ) </pre>
SB_CATALOG_VBS	SCRATTR	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCBSUPP_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCBSUPP_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCBSUPP_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCBSUPP_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCBSUPP_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRCORQ	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRATTR_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRATTR_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRATTR_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRATTR_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRATTR_EFF_TERM ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRCPRT	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRCORQ_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRCORQ_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRCORQ_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRCORQ_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRCORQ_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRCRDF	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRCPRT_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRCPRT_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRCPRT_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRCPRT_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRCPRT_TERM_CODE_EFF ) </pre>
SB_CATALOG_VBS	SCREQIV	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRCRDF_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRCRDF_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRCRDF_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRCRDF_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRCRDF_TERM_CODE_EFF ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRFEES	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCREQIV_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCREQIV_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCREQIV_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCREQIV_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCREQIV_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRGMOD	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRFEES_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRFEES_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRFEES_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRFEES_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRFEES_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRINTG	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRINTG_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRINTG_SUBJ_CODE AND SCBCRSE_EFF_TERM = (SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRINTG_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRINTG_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRINTG_TERM_CODE_EFF) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRLEVL	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRGMOD_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRGMOD_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRGMOD_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRGMOD_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRGMOD_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRREARE	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRLEVL_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRLEVL_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRLEVL_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRLEVL_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRLEVL_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRRCAM	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRREARE_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRREARE_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRREARE_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRREARE_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRREARE_TERM_CODE_EFFECTIVE ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRRCLS	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRCAM_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRCAM_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRCAM_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRCAM_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRCAM_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRRCMP	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRCLS_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRCLS_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRCLS_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRCLS_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRCLS_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRRCOL	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRCMP_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRCMP_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRCMP_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRCMP_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRCMP_EFF_TERM ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRRDEG	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRCOL_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRCOL_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRCOL_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRCOL_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRCOL_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRRLVL	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CR SE_NUMB = SCRRDEG_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRDEG_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRDEG_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRDEG_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRDEG_TERM_CODE_EFFECTIVE ) </pre>
SB_CATALOG_VBS	SCRMAJ	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRLVL_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRLVL_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRLVL_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRLVL_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRLVL_EFF_TERM ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRRPRG	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRMAJ_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRMAJ_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRMAJ_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRMAJ_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRMAJ_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCRRTM	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRPRG_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRPRG_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRPRG_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRPRG_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRPRG_TERM_CODE_EFFECTIVE ) </pre>
SB_CATALOG_VBS	SCRRTST	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRTM_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRTM_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRTM_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRTM_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRTM_EFF_TERM ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCR SBGI	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRRTST_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRRTST_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRRTST_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRRTST_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRRTST_TERM_CODE_EFF ) </pre>
SB_CATALOG_VBS	SCR SCHED	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCR SBGI_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCR SBGI_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCR SBGI_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCR SBGI_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCR SBGI_EFF_TERM ) </pre>
SB_CATALOG_VBS	SCR SYLN	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCR SCHED_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCR SCHED_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCR SCHED_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCR SCHED_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCR SCHED_EFF_TERM ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRSYLO	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRSYLN_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRSYLN_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRSYLN_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRSYLN_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRSYLN_TERM_CODE_EFF ) </pre>
SB_CATALOG_VBS	SCRSYRM	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRSYLO_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRSYLO_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRSYLO_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRSYLO_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRSYLO_TERM_CODE_EFF ) </pre>
SB_CATALOG_VBS	SCRSYTR	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRSYRM_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRSYRM_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRSYRM_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRSYRM_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRSYRM_TERM_CODE_EFF ) </pre>

Domain Name	Policy Table	Code that Links Policy Table to Domain Driver Table (GOBFDMN)
SB_CATALOG_VBS	SCRTEXT	<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRSYTR_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRSYTR_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRSYTR_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRSYTR_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRSYTR_TERM_CODE_EFF ) </pre>
SB_CATALOG_VBS		<pre> EXISTS (SELECT 'X' FROM SCBCRSE WHERE SCBCRSE_CRSE_NUMB = SCRTEXT_CRSE_NUMB AND SCBCRSE_SUBJ_CODE = SCRTEXT_SUBJ_CODE AND SCBCRSE_EFF_TERM = ( SELECT MAX(A.SCBCRSE_EFF_TERM) FROM SCBCRSE A WHERE A.SCBCRSE_SUBJ_CODE = SCRTEXT_SUBJ_CODE AND A.SCBCRSE_CRSE_NUMB = SCRTEXT_CRSE_NUMB AND A.SCBCRSE_EFF_TERM &lt;= SCRTEXT_EFF_TERM ) </pre>

## VBS Domain Implementation Details

This section uses the Business Need Analysis Worksheet to describe the seed data for specific VBS domains. See [“Analysis Worksheet Template” on page 23](#) for an introduction to the worksheet.

## FGAC VBS Predicate Access

Domain	Description	Driver	Tables
GB_FGAC_PREDICATE_VBS	VBS using FGAC Predicates Rules	GORFPRD	GORFPRD

<b>Columns for use in Select, Insert, Update, Delete Restriction Mode</b>	<b>Description</b>
GORFPRD_FGAC_CODE	FGAC CODE: FGAC group code.
GORFPRD_FDMN_CODE	FGAC DOMAIN CODE: FGAC domain code

<b>Examples of Select, Insert, Update, Delete Restrictions using GB_FGAC_PREDICATE_VBS</b>	<b>Business Case Issues</b>
Restrict maintenance based on the FGAC group code GORFPRD_FGAC_CODE LIKE 'SB_%'	Restrict users in the Student Security Administrative group to the maintenance of only VBS group rules for the Student users. Set up naming convention for GTVFGAC_CODESs that have the product as a prefix to the group name.
GORFPRD_FDMN_CODE LIKE 'SB%'	Restrict Student Security Administrators group to creating VBS Group rules on student product domains.

<b>Objects that may be impacted by insert, update and delete restrictions</b>	<b>Possible Outcomes/Impacts</b>
GOAFGAC	Maintenance of FGAC group rules is restricted.

## FGAC VBS Access

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
GB_FGACACCESS_VBS	VBS using FGAC Group Rules	GOBFGAC	GOBFGAC

<b>Columns for use in Select, Insert, Update, Delete Restriction Mode</b>	<b>Description</b>
GOBFGAC_FGAC_CODE	FGAC CODE: FGAC group code.

<b>Examples of Select, Insert, Update, Delete Restrictions using GB_FGACACCESS_VBS</b>	<b>Business Case Issues</b>
Restrict maintenance based on the FGAC group code  GOBFGAC_FGAC_CODE LIKE 'SB_%'	Allow users in the Student Security Administrative group to the maintenance of only VBS group rules for the Student users. Set up naming convention for GTVFGAC_CODES that have the product as a prefix to the group name.

<b>Objects that may be impacted by insert, update and delete restrictions</b>	<b>Possible Outcomes/Impacts</b>
GOAFGAC	Maintenance of FGAC group rules is restricted.

## International

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
GB_INTERNATIONAL_VBS	Person International Information	GOBINTL	GOBINTL

<b>Columns for use in Select, Insert, Update, Delete Restriction Mode</b>	<b>Description</b>
GOBINTL_ADMR_CODE	This field indicates Admission Request Code.
GOBINTL_SPON_CODE	This field indicates the Sponsor Code.

<b>Columns for use in Select, Insert, Update, Delete Restriction Mode</b>	<b>Description</b>
GOBINTL_EMPT_CODE	This field indicates the person's type of Employment.
GOBINTL_LANG_CODE	This field indicates the person's native Language Code.
GOBINTL_CELG_CODE	This field indicates Passport Certification Eligibility Code
GOBINTL_NATN_CODE_LEGAL	This field indicates the Country Code of the nation of the person's Citizenship.
GOBINTL_NATN_CODE_BIRTH	This field indicates the Country Code of the nation of the person's Birth.

<b>Examples of Select, Insert, Update, Delete Restrictions using GB_INTERNATIONAL_VBS</b>	<b>Business Case Issues</b>
Restrict maintenance based on the admissions code.  GOBINTL_ADMR_CODE = 'F1'	Student admissions offices can maintain international records only if they are for a specific application.

<b>Objects that may be impacted by insert, update and delete restrictions</b>	<b>Possible Outcomes/Impacts</b>
GOAINTL	Maintenance of international information
SAAEAPS	Push application with international information
SRIPREL	Create recruit from electronic prospect with international information
SAAADMS	Maintain application checklist for international information
SAAACKL	Maintain application checklist for international information
Student view SAVADM2	Selection of international information
Payroll view PEVPRF1	Selection of international information
International Person View SPVINTL	Selection of international information
Person View AS_PERSON	Selection of international information

## Addresses

Restrictions on the address type code (STVATYP) can be used to limit the list of values to codes applicable to data in their area. If a select restriction is placed on the address type code validation table, the same restriction needs to be added using the GB\_SPRADDR\_VBS. The impact of not doing this is missing descriptions and form error messages during the query process on all other forms that display the address type

Domain	Description	Driver	Tables
GB_SPRADDR_VBS	Person Addresses	SPRADDR	SPRADDR

Restriction Mode	Columns Used in Restrictions	Column Description
Select, Insert, Update, Delete	SPRADDR_ASRC_CODE	Address source code associated with the person
Select, Insert, Update, Delete	SPRADDR_ATYP_CODE	Type of address associated with the person
Select, Insert, Update, Delete	SPRADDR_STAT_CODE	State/province associated with the person's address
Select, Insert, Update, Delete	SPRADDR_ZIP	ZIP code associated with the person's address
Select, Insert, Update, Delete	SPRADDR_NATN_CODE	Nation/country associated with the person's address

Business Cases Using GB_SPRADDR_VBS	Business Case Issues
Restrict by address type (using Select, Insert, Update, Delete)  SPRADDR_ATYP_CODE IN ('MA', 'W2')	

<b>Other Business Case Examples Using GB_SPRADDR_VBS</b>
By address maintenance by the address source SPRADDR_ASRC_CODE IN ( 'SELF' , 'POST' )
By maintenance of addresses by state, ZIP, or nation SPRADDR_STAT_CODE IN ( 'MA' , 'VT' , 'NH' , 'ME' , 'NY' , 'RI' )

<b>Student objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SPAIDEN	Maintenance of addresses
SAAQUIK	Maintenance of addresses
SRAQUIK	Maintenance of addresses

<b>Other product objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
APAIDEN	Maintenance of addresses
FOAIDEN	Maintenance of addresses
FTMAGCY	Maintenance of addresses
FTMCUST	Maintenance of addresses
FTMFMGR	Maintenance of addresses
FTMVEND	Maintenance of addresses
GOAADDR	Maintenance of addresses
PEA1PAY	Maintenance of addresses
PPAIDEN	Maintenance of addresses

## Telephones

Domain	Description	Driver	Tables
GB_SPRTELE_VBS	Person Telephone Numbers	SPRTELE E	SPRTELE

Restriction Mode	Columns Used in Restrictions	Column Description
Select, Insert, Update, Delete	SPRTELE_TELE_CODE	Telephone type code for the person's telephone number
Select, Insert, Update, Delete	SPRTELE_ATYP_CODE	Optional address type code associated with telephone number

Business Cases Using GB_SPRTELE_VBS	Business Case Issues
Restrict by telephone type (using Select, Insert, Update, Delete)  SPRTELE_TELE_CODE IN ( 'HOME' , 'FAX' )	

Other Business Case Examples Using GB_SPRTELE_VBS
N/A

Objects that may be impacted by restrictions	Possible Impacts
SPAIDEN	Maintenance of telephone number
SAAQUIK	Maintenance of telephone number
SRAQUIK	Maintenance of telephone number

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SPATELE	Maintenance of telephone number

## Medical

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
GB_SPRMEDI_VBS	Person Medical Information	SPRMEDI I	SPRMEDI

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Insert, Update, Delete	SPRMEDI_MEDI_CODE	Medical code associated with the person
Select, Insert, Update, Delete	SPRMEDI_MDEQ_CODE	Medical equipment code associated with the person
Select, Insert, Update, Delete	SPRMEDI_DISA_CODE	Disability type code associated with the person
Select, Insert, Update, Delete	SPRMEDI_SPSR_CODE	Disability service code available to the person

<b>Business Cases Using GB_SPRMEDI_VBS</b>	<b>Business Case Issues</b>
Restrict by medical code (using Select, Insert, Update, Delete)  SPRMEDI_MEDI_CODE = 'DE'	Different health professionals are used for different medical issues

<b>Other Business Case Examples Using GB_SPRMEDI_VBS</b>
N/A

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SAAEAPS	Push application with medical information
Student views	Selection of medical information

<b>Other product objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
GOAMEDI	Maintenance of medical information
PEAEACC	Maintenance of employee accommodations
Payroll views	Selection of medical information

## Catalog

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
SB_CATALOG_ VBS	Course Catalog	SCBCRSE	SCRSYLO, SCRSYRM, SCRSYTR, SCRTEXT, SCRSYLN, SCRFEEES, SCRGMOD, SCRLEVEL, SCRRARE, SCRRCAM, SCRRCLS, SCRRCMP, SCRRCOL, SCRRDEG, SCRRLVL, SCRRMAJR, SCRRPRG, SCRRTRM, SCRRSTST, SCRSBGI, SCBCDTL, SCBCRSE, SCBDESC, SCBSUPP

The Catalog domain predicates should be used to control the insertion, updating, and deletion of data. The impact of having predicates to control selection of the catalog is far reaching in Banner Student and may cause corrupt data. For example, the course title is derived from the catalog if not it is not available from the schedule. The course title appears on many pages and reports.

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Insert, Update, Delete	SCBCRSE_SUBJ_CODE	Subject area of the course

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Insert, Update, Delete	SCBCRSE_CRSE_NUMB	Course number associated with the subject for the course
Insert, Update, Delete	SCBCRSE_EFF_TERM	Term this version of the course becomes effective
Insert, Update, Delete	SCBCRSE_COLL_CODE	College which offers the course
Insert, Update, Delete	SCBCRSE_DIVS_CODE	Division which offers the course
Insert, Update, Delete	SCBCRSE_DEPT_CODE	Department which offers the course
Insert, Update, Delete	SCBCRSE_CSTA_CODE	Status code for the course
Insert, Update, Delete	SCBCRSE_CIPC_CODE	CIP code (primary subject matter) for the course

<b>Business Cases Using SB_CATALOG_VBS</b>	<b>Business Case Issues</b>
Restrict by subject or department or college (using Insert, Update, Delete)  SCBCRSE_SUBJ_CODE IN ('1', '2') AND SCBCRSE_DEPT_CODE IN ('1', '2') AND SCBCRSE_COLL_CODE IN ('1', '2')	Restrict maintenance by college, department, or subject

<b>Other Business Case Examples Using SB_CATALOG_VBS</b>
N/A

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SCACRSE	Maintenance of course catalog information

Objects that may be impacted by restrictions	Possible Impacts
SCAETL	Maintenance of course catalog information
SCARRES	Maintenance of course catalog information
SCAPREQ	Maintenance of course catalog information
SCASRES	Maintenance of course catalog information
SCABASE	Maintenance of base catalog information
SCASYLB	Maintenance of course catalog syllabus
Faculty Self-Service	Maintenance of course catalog information

## Schedule

Domain	Description	Driver	Tables
SB_SCHEDULE_VBS	Class Schedule	SSBSECT	SSBDESC, SSBFSEC, SSBOVRR, SSBSECT, SSRATTR, SSRBLCK, SSRCORQ, SSREVAL, SSREXTN, SSRFEES, SSRLINK, SSRMEET, SSRMPRT, SSRMRDF, SSRRARE, SSRRCLS, SSRRCMP, SSRRDEG, SCRRLVL, SSRMAJ, SSRPRG, SSRSTS, SSRRTST, SSRSCCD, SSRSPRT, SSRSRDF, SSRSYLN, SSRSYLO, SSRSYRM, SSRSYTR, SSRTEXT, SSRXLST, SSRRESV, SSRRFND, SSRRLVL, SSRMAJ, SSRPRG

The Schedule domain predicates should be used to control the insertion, updating, and deletion of data. It is recommended that the selection restrictions not be applied to the Schedule tables. This could impact viewing student course data on transcripts and other online course reports.



**Note:** Special processing has been added to the learner registration validation and update package SFKEDIT to turn FGAC off so that enrollment counts will not be compromised by VBS restrictions.

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Insert, Update, Delete	SSBSECT_INSM_CODE	Instructional method code assigned to the section
Insert, Update, Delete	SSBSECT_TERM_CODE	Term code for the course section information
Insert, Update, Delete	SSBSECT_PTRM_CODE	Part-of-term in which the section is offered
Insert, Update, Delete	SSBSECT_SUBJ_CODE	Subject code for the section
Insert, Update, Delete	SSBSECT_SSTS_CODE	Status of the section
Insert, Update, Delete	SSBSECT_SCHD_CODE	Instructional type of the section being scheduled
Insert, Update, Delete	SSBSECT_CAMP_CODE	Campus on which the section is scheduled
Insert, Update, Delete	SSBSECT_GMOD_CODE	Grading mode for the section
Insert, Update, Delete	SSBSECT_SESS_CODE	Session in which the section is scheduled

<b>Business Cases Using SB_SCHEDULE_VBS</b>	<b>Business Case Issues</b>
Restrict by subject or campus or session or schedule type (using Insert, Update, Delete)  SSBSECT_SUBJ_CODE IN ( '1', '2' ) AND SSBSECT_CAMP_CODE IN ( '1', '2' ) AND SSBSECT_GMOD_CODE IN ( '1', '2' )	Restrict maintenance of the schedule to subject, campus, or grade mode types

<b>Other Business Case Examples Using SB_SCHEDULE_VBS</b>
N/A

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SSASECT	Maintenance of schedule information
SSADETL	Maintenance of schedule information
SSATEXT	Maintenance of schedule information
SSAPREQ	Maintenance of schedule information
SSASYLB	Maintenance of schedule information
SSARULE	Maintenance of schedule information
SSAACCL	Maintenance of schedule information
SSAASEC	Maintenance of schedule information
SSAEVAL	Maintenance of schedule information
SSAOVRR	Maintenance of schedule information
SSASECQ	Maintenance of schedule information
Faculty Self-Service	Maintenance of schedule information

## Curriculum

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
SB_CURRICULUM_VBS	Recruiting, Admissions, Learner, Outcome, Base Curriculum	SORLCUR	SORLCUR

Starting with Banner 7.0, all curriculum data is maintained in the new base curriculum and field of study tables SORLCUR and SORLFOS. Restrictions on the insertion and deletion of curriculum data should now be written against these new tables and not against SRBRECR, SARADAP, and SGASTDN. Also starting with Banner 7.0, curriculum data can no longer be updated. The process used to change data includes copying the curriculum record using the key insert/key duplicate and making changes to the new record.

If the user wishes to restrict selection against SRBRECR, SARADAP, and SGASTDN based on the primary or secondary curriculum, they can continue to use those tables' curriculum columns in building selection, updating, and deletion rules against those tables. The primary and secondary curriculum data are now backfilled from SORLCUR and SORLFOS to SARADAP, SRBRECR, and SGASTDN after all inserts have occurred.

If selection restrictions are built using the SRBRECR, SARADAP, and SGASTDN curriculum data columns, an alternative check needs to be added for the data to be null. The Curriculum API validates the existence of the host record before allowing an insert. For example, you cannot insert an application module curriculum if the application does not exist. The application will appear to not exist if the rule is only for non-null values.

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Insert, Delete	SORLCUR_LMOD_CODE	Learner module code for the curriculum
Insert, Delete	SORLCUR_TERM_CODE	Term code for the curriculum
Insert, Delete	SORLCUR_LEVL_CODE	Level code for the curriculum
Insert, Delete	SORLCUR_COLL_CODE	College code for the curriculum
Insert, Delete	SORLCUR_DEGC_CODE	Degree code for the curriculum
Insert, Delete	SORLCUR_TERM_CODE_CTLG	Catalog term code for the learner curriculum
Insert, Delete	SORLCUR_ADMT_CODE	Admit code for the learner's admission to the curriculum
Insert, Delete	SORLCUR_CAMP_CODE	Campus code for the curriculum
Insert, Delete	SORLCUR_PROGRAM	Program code for the curriculum

<b>Business Cases Using SB_CURRICULUM_VBS</b>	<b>Business Case Issues</b>
Restrict recruits by curriculum SORLCUR_LMOD_CODE = sb_curriculum.f_recruit AND SORLCUR_LEVL_CODE = 'X' AND SORLCUR_COLL_CODE = 'XX'	Need to restrict various campus offices that recruit different populations for different programs

<b>Other Business Case Examples Using SB_CURRICULUM_VBS</b>
<p>Applicants by curriculum</p> <p>SORLCUR_LMOD_CODE = sb_curriculum.f_admissions AND</p> <p>SORLCUR_LEVL_CODE = 'X' AND</p> <p>SORLCUR_COLL_CODE = 'XX'</p>
<p>Learners by curriculum</p> <p>SORLCUR_LMOD_CODE = sb_curriculum.f_learner AND</p> <p>SORLCUR_LEVL_CODE = 'X' AND</p> <p>SORLCUR_COLL_CODE = 'XX'</p>
<p>Outcome by curriculum</p> <p>SORLCUR_LMOD_CODE = sb_curriculum.f_outcome AND</p> <p>SORLCUR_LEVL_CODE = 'X' AND</p> <p>SORLCUR_COLL_CODE = 'XX'</p>

<b>Objects that may be impacted by restrictions * Possible Impacts **</b>	
SRAQUIK	Cannot save the recruit, application, or learner record if primary curriculum cannot be saved
SAAADMS	Cannot save the application if primary curriculum cannot be saved
SGASTDN	Maintenance of the learner's curriculum
SAAQUIK	Cannot create the learner record, application, or recruit record since the curriculum cannot be saved
SRARECR	Cannot save the recruit record if primary curriculum cannot be saved
SFAREGS	Maintenance of the learner's curriculum
SHADEGR	Maintenance of the learner's outcome curriculum
SRIPREL	Cannot create the recruit or application for prospect
SAADCRV	Cannot add a decision to applicant that will result in a learner record being created

<b>Objects that may be impacted by restrictions *</b>	<b>Possible Impacts **</b>
SAADCBT	Cannot add a decision to applicant that will result in a learner record being created
SAAEAPS	Cannot push the application for an electronic applicant
Online Grade Roll	Cannot create the outcome record or add unrolled curriculum during the online grade rolls
SOVLCUR	Selection of curriculum information

\* Impacted objects do not include Banner batch objects. All job type objects are listed in the Banner General FGAC GORFEOB object. Banner security turns VBS using FGAC off when an object exists in the GORFEOB table.

\*\* This list may not include all possible impacts. This needs to be reviewed by each institution based on the type of restrictive rule coded on GOAFGAC.

## Field of Study

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
SB_FIELDOFSTUDY_VBS	Recruiting, Admissions, Learner, Outcome, Base Fields of Study	SORLFOS	SORLFOS

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Insert, Delete	SORLFOS_MAJR_CODE_ATTACH	Major code to which the concentration is attached
Insert, Delete	SORLFOS_MAJR_CODE	Major code for the field of study
Insert, Delete	SORLFOS_DEPT_CODE	Department code associated with the major field of study
Insert, Delete	SORLFOS_TMST_CODE	Time status code used to indicate the intent of the student's perusal of the field of study

<b>Business Cases Using SB_FIELDOFSTUDY_VBS</b>	<b>Business Case Issues</b>
<p>Restrict any learner module by major</p> <pre>SORLFOS_DEPT_CODE = 'XXX' AND SORLFOS_MAJR_CODE = 'XXX' AND SORLFOS_LFST_CODE = sb_fieldofstudy_str.f_major</pre>	<p>Need to restrict major changes to certain departments by department</p>

<b>Other Business Case Examples Using SB_FIELDOFSTUDY_VBS</b>
<p>Recruits by department or major</p> <pre>SORLFOS_DEPT_CODE = 'XXX' AND SORLFOS_MAJR_CODE = 'XXX' AND EXISTS (SELECT 'X' FROM SORLCUR WHERE SORLCUR_PIDM = SORLFOS_PIDM AND SORLCUR_SEQNO = SORLFOS_LCUR_SEQNO AND SORLCUR_LMOD_CODE = sb_curriculum_str.f_recruit)</pre>
<p>Applicants by department or major</p> <pre>SORLFOS_DEPT_CODE = 'XXX' AND SORLFOS_MAJR_CODE = 'XXX' AND EXISTS (SELECT 'X' FROM SORLCUR WHERE SORLCUR_PIDM = SORLFOS_PIDM AND SORLCUR_SEQNO = SORLFOS_LCUR_SEQNO AND SORLCUR_LMOD_CODE = sb_curriculum_str.f_admissions)</pre>
<p>Learners by department or major</p> <pre>SORLFOS_DEPT_CODE = 'XXX' AND SORLFOS_MAJR_CODE = 'XXX' AND EXISTS (SELECT 'X' FROM SORLCUR WHERE SORLCUR_PIDM = SORLFOS_PIDM AND SORLCUR_SEQNO = SORLFOS_LCUR_SEQNO AND SORLCUR_LMOD_CODE = sb_curriculum_str.f_learner)</pre>

### Other Business Case Examples Using SB\_FIELDOFSTUDY\_VBS

Outcome by department or major

```
SORLFOS_DEPT_CODE = 'XXX' AND
```

```
SORLFOS_MAJR_CODE = 'XXX' AND
```

```
EXISTS (SELECT 'X' FROM SORLCUR WHERE SORLCUR_PIDM =  
SORLFOS_PIDM AND
```

```
SORLCUR_SEQNO = SORLFOS_LCUR_SEQNO AND SORLCUR_LMOD_CODE =  
sb_curriculum_str.f_outcome)
```

### Objects that may be impacted by restrictions

### Possible Impacts

SRAQUIK	Cannot save recruit, application, or learner record if primary curriculum cannot be saved
SAAADMS	Cannot save application if primary curriculum cannot be saved
SGASTDN	Maintenance of learner's curriculum
SAAQUIK	Cannot create the recruit, application, or learner record since the curriculum cannot be saved
SRARECR	Cannot save recruit record if primary curriculum cannot be saved
SFAREGS	Maintenance of learner's curriculum
SHADEGR	Maintenance of learner's outcome curriculum
SRIPREL	Cannot create the recruit or application for prospect
SAADCRV	Cannot add a decision to an applicant that will result in a learner record being created
SAADCBT	Cannot add a decision to an applicant that will result in a learner record being created
SAAEAPS	Cannot push the application for an electronic applicant
Online Grade Roll	Cannot create the outcome record or add unrolled curriculum during the online grade roll
SOVLFOS	Selection of field of study information

## Recruit

Domain	Description	Driver	Tables
SB_RECRUIT_VBS	Recruiting	SRBRECR	SRBRECR, SRRLEND, SRRRATT, SRRRSRC, SRRCHRT

The Recruit domain predicates built on non-curriculum items are meant to control the selection, insertion, updating, and deletion of the recruit data. Predicates that include the primary or secondary curriculum items on SRBRECR can only be used to control selection of the recruit data.

Restriction Mode	Columns Used in Restrictions	Column Description
Select, Insert, Update, Delete	SRBRECR_TERM_CODE	Effective term where recruiting records will be viewed or created.
Select, Insert, Update, Delete	SRBRECR_RECR_CODE	Recruiter assigned to the prospect recruiting record
Select, Insert, Update, Delete	SRBRECR_RSTA_CODE	Status of the prospect associated with the recruiting record
Select, Insert, Update, Delete	SRBRECR_ADMT_CODE	Admission type code for the prospect recruiting record
Select, Insert, Update, Delete	SRBRECR_FULL_PART_IND	Indicates if prospect will be full-time or part-time student
Select, Insert, Update, Delete	SRBRECR_RTYP_CODE	Recruit type code for the prospect recruiting record
Select, Insert, Update, Delete	SRBRECR_RESD_CODE	Residency code for the prospect recruiting record
Select, Insert, Update, Delete	SRBRECR_SESS_CODE	Session code for the prospect recruiting record
Select, Insert, Update, Delete	SRBRECR_SITE_CODE	Site code for the prospect recruiting record
Select, Insert, Update, Delete	SRBRECR_STYP_CODE	Student type code for the prospect recruiting record
Select, Update, Delete	SRBRECR_TERM_CODE	Effective term where recruiting records will be viewed or created

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Update, Delete	SRBRECR_DEPT_CODE	Department associated with the prospect recruiting record
Select, Update, Delete	SRBRECR_LEVL_CODE	Level code associated with the prospect recruiting record
Select, Update, Delete	SRBRECR_DEGC_CODE	Degree code associated with the prospect recruiting record
Select, Update, Delete	SRBRECR_MAJR_CODE	Major code associated with the prospect recruiting record
Select, Update, Delete	SRBRECR_CAMP_CODE	Campus code associated with the prospect recruiting record
Select, Update, Delete	SRBRECR_COLL_CODE	College code associated with the prospect recruiting record
Select, Update, Delete	SRBRECR_PROGRAM_1	Curriculum 1 program code associated with the prospect recruiting record
Select, Update, Delete	SRBRECR_TERM_CODE_CTLG_1	Curriculum 1 catalog term code associated with the prospect recruiting record

<b>Business Cases Using SB_RECRUIT_VBS</b>	<b>Business Case Issues</b>
Restrict by instate recruits (using Select, Insert, Update, Delete)  SRBRECR_RESD_CODE = 'R'	Need to restrict creation of the recruit based on if applicant is instate or out of state
Restrict selection of recruit records based on the primary curriculum (using Select, Update, Delete)  (SRBRECR_LEVL_CODE = 'X' AND SRBRECR_MAJR_CODE = 'XXXX' ) OR (SRBRECR_LEVL_CODE IS NULL AND SRBRECR_MAJR_CODE IS NULL)	Need to restrict selection, updating, and deletion of the recruit record based on the primary curriculum level and major

Other Business Case Examples Using SB_RECRUIT_VBS
By recruiter type SRBRECR_RTYP_CODE = 'GR'
By full-time recruits SRBRECR_FULL_PART_TIME = 'F'
By recruiter SRBRECR_RECR_CODE IN ('1', '2')

Objects that may be impacted by restrictions	Possible Impacts
SRARECR	Recruit record maintenance
SRAQUIK	Recruit record quick entry
SRIPREL	Create recruit record during create process for an electronic prospect
SAAQUIK	Create recruit record during quick admissions process
SRARINF	View recruiting information
SRASRCE	View recruiting information
SOILCUR	View recruiting information
SRASUMI	View recruiting information
SOASBSM	View recruiting information
SOAPCSM	View recruiting information
Student and Financial Aid product views with policy tables for SB_RECRUIT_VBS	View recruiting information

## Admissions

Domain	Description	Driver	Tables
SB_ADMISSIONS_ VBS	Admissions Application	SARADAP	SARADAP, SARAPPD, SARCHKL, SARCHRT, SARDSCL, SARQUAN, SARRRAT, SARRSRC

The Admissions domain predicates built on non-curriculum items are meant to control the selection, insertion, updating, and deletion of the application data. Predicates that include the primary or secondary curriculum items should only be used to control selection, updating, or deletion of the application.

In addition, the curriculum data is null until the backfill has been run, and any predicate that checks the application curriculum should also check alternatively if the curriculum fields are null. This is required for the curriculum APIs to complete, because one of the checks in the curriculum insert is that the host record must exist.

Restriction Mode	Columns Used in Restrictions	Column Description
Select, Insert, Update, Delete	SARADAP_TERM_CODE_ENTRY	Term associated with the information on applicant record
Select, Insert, Update, Delete	SARADAP_APST_CODE	Status of the application
Select, Insert, Update, Delete	SARADAP_ADMT_CODE	Admission type of the applicant
Select, Insert, Update, Delete	SARADAP_STYP_CODE	Type of student the applicant will become if admitted
Select, Insert, Update, Delete	SARADAP_SITE_CODE	Site to which applicant is applying
Select, Insert, Update, Delete	SARADAP_RESD_CODE	Residency classification of the applicant
Select, Insert, Update, Delete	SARADAP_FULL_PART_IND	Identifies if applicant intention is for full or part time enrollment
Select, Insert, Update, Delete	SARADAP_SESS_CODE	Session applicant is applying for
Select, Insert, Update, Delete	SARADAP_RATE_CODE	Specific assessment rate assigned to the student
Select, Insert, Update, Delete	SARADAP_RECR_CODE	Recruiter code for the applicant

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Insert, Update, Delete	SARADAP_RTYP_CODE	Recruit type code for the applicant
Select, Update, Delete	SARADAP_LEVL_CODE	Level for which the applicant is applying
Select, Update, Delete	SARADAP_CAMP_CODE	Campus to which applicant is applying
Select, Update, Delete	SARADAP_COLL_CODE_1	Primary college to which applicant is applying
Select, Update, Delete	SARADAP_DEGC_CODE_1	Primary degree for which applicant is applying
Select, Update, Delete	SARADAP_MAJR_CODE_1	Primary major for which applicant is applying
Select, Update, Delete	SARADAP_DEPT_CODE	Department to which applicant is applying
Select, Update, Delete	SARADAP_PROGRAM_1	Curriculum 1 program code for which applicant is applying
Select, Update, Delete	SARADAP_TERM_CODE_CTLG_1	Curriculum 1 catalog term code for which applicant is applying

<b>Business Cases Using SB_ADMISSIONS_VBS</b>	<b>Business Case Issues</b>
Restrict by instate applicants (using Select, Insert, Update, Delete)  SARADAP_RESD_CODE = '<value for instate>'	Need to restrict creation of the application based on if applicant is instate or out of state
Restrict applications by primary level and college (using Select, Update, Delete)  (SARADAP_LEVL_CODE = 'UG' AND SARADAP_COLL_CODE_1 = 'AS') OR (SARADAP_LEVL_CODE IS NULL AND SARADAP_COLL_CODE_1 IS NULL)	Need to restrict selection, updating, and deletion of the application based on the primary curriculum level and college

<b>Other Business Case Examples Using SB_ADMISSIONS_VBS</b>
By instate applicants SARADAP_RESD_CODE = 'R'
By full-time applicants SARADAP_FULLL_PART_TIME = 'F'
By admit type SARADAP_ADMT_CODE IN ('1', '2')

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SAAACKL	Maintenance of checklist items
SAAQUAN	Maintenance of application questions and answers
SAAADMS	Maintenance of applications
SAAQUIK	Create application function on the quick admissions process
SAARRAT	Maintenance of admissions ratings
SAADCRV	Maintenance of decisions
SAADCBT	Maintenance of decisions and admissions ratings
SRIPREL	Create application in the create recruit function for the electronic prospects
SOASUPL	Maintenance of supplemental application information
SOAGPAT	Maintenance of other GPAs for applicant
SOILCUR	View application information
SAASUMI	View application information
SOASBSM	View application information
SOAPCSM	View application information
Student and Financial Aid product views with policy tables for SB_ADMISSIONS_VBS	View application information

## Learner (General Student)

Domain	Description	Driver	Tables
SB_LEARNER_VBS	General Student	SGBSTDN	SGBEOPS, SGBOEDU, SGBSTDN, SGRACMT, SGBUSER, SGRADVR, SGRASSI, SGRASST, SGRCHRT, SGRCMNT, SGRCOOP, SGRDSER, SGRDISA, SGRDUTY, SGRESEL, SGRSACT, SGRSATT, SGRSCMT, SGRSPRT, SGRVETN

The Learner domain predicates built on non-curriculum items are meant to control the insertion, updating, and deletion of the learner data. Predicates that include the primary or secondary curriculum items on SGBSTDN can only be used to control selection of the learner data.

Restriction Mode	Columns Used in Restrictions	Column Description
Insert, Update, Delete	SGBSTDN_TERM_CODE_GRAD	Term student intends to graduate
Insert, Update, Delete	SGBSTDN_ACYR_CODE	Year student intends to graduate
Insert, Update, Delete	SGBSTDN_SITE_CODE	Site code for the student record
Insert, Update, Delete	SGBSTDN_TERM_CODE_EFF	Effective term associated with student record
Insert, Update, Delete	SGBSTDN_STST_CODE	Student status for the effective term
Insert, Update, Delete	SGBSTDN_STYP_CODE	Student type for the effective term
Insert, Update, Delete	SGBSTDN_FULL_PART_IND	Indicates whether the student is full-time or part-time
Insert, Update, Delete	SGBSTDN_SESS_CODE	Session student is attending for the effective term
Insert, Update, Delete	SGBSTDN_RESID_CODE	Residency status of the student for the effective term

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Update, Delete	SGBSTDN_ADMT_CODE	Admissions type from the admissions application
Select, Update, Delete	SGBSTDN_DEPT_CODE	Department code for the effective term
Select, Update, Delete	SGBSTDN_PROGRAM_1	Curriculum 1 program code for the effective term
Select, Update, Delete	SGBSTDN_TERM_CODE_CTLG_1	Curriculum 1 catalog term code for the effective term
Select, Update, Delete	SGBSTDN_LEVL_CODE	Level of the student for the effective term
Select, Update, Delete	SGBSTDN_TERM_CODE_ADMIT	Term student was first admitted to institution
Select, Update, Delete	SGBSTDN_CAMP_CODE	Campus location associated with the student for the effective term
Select, Update, Delete	SGBSTDN_COLL_CODE_1	College associated with the primary curriculum for the effective term
Select, Update, Delete	SGBSTDN_DEGC_CODE_1	Degree within the primary curriculum for the effective term
Select, Update, Delete	SGBSTDN_MAJR_CODE_1	Primary major within the primary curriculum for the effective term

<b>Business Cases Using SB_LEARNER_VBS</b>	<b>Business Case Issues</b>
Restrict by instate students (using Insert, Update, Delete)  SGBSTDN_RESD_CODE = 'R'	Need to restrict creation of the learner based on if person is instate or out of state
Restrict selection of learner records based on the primary curriculum (using Select, Update, Delete)  (SRBRECR_LEVL_CODE = 'X' AND SRBRECR_MAJR_CODE = 'XXXX') OR (SGBSTDN_LEVL_CODE IS NULL AND SGBSTDN_MAJR_CODE IS NULL)	

<b>Other Business Case Examples Using SB_LEARNER_VBS</b>
<p>By full-time learners</p> <p>SGBSTDN _FULL_PART_TIME = 'F'</p>
<p>By student type</p> <p>SGBSTDN_STYP_CODE IN ( 'AS' , 'NS' )</p>
<p>Selection by primary curriculum</p> <p>SGBSTDN_LEVL_CODE = 'XX' AND</p> <p>SGBSTDN_COLL_CODE = 'XX' OR</p> <p>( SGBSTDN_LEVL_CODE IS NULL AND</p> <p>SGBSTGDN_COLL_CODE_1 IS NULL )</p>
<p>Selection by admissions code</p> <p>SGBSTDN_ADMT_CODE = 'XX'</p>

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SGASTDN	Maintain general learner information
SFAREGS	Maintain general learner information
SAADCRV	Admit application and create learner
SAADCBT	Admit application and create learner
SOILCUR	View general learner information
SGASTDQ	View general learner information
SGACOOOP	Maintain general learner information
SGAEOPS	Maintain general learner information
SGASPRT	Maintain general learner information
SGAADVR	Maintain general learner information
SGADISA	Maintain general learner information
SGAUSDF	Maintain general learner information

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
Student and Financial Aid product views with policy tables for SB_LEARNER_VBS	View application information

## Test Scores

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
SB_TESTSCORES_VBS	Test Scores	SORTEST	SORTEST

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Insert, Update, Delete	SORTEST_TESC_CODE	Test score code associated with various tests, i.e., SAT, ACT
Select, Insert, Update, Delete	SORTEST_TSRC_CODE	Test source or how the test score was entered, i.e., data load or self-reported
Select, Insert, Update, Delete	SORTEST_TADM_CODE	Administrative type of a particular test, i.e., locally or nationally administered
Select, Insert, Update, Delete	SORTEST_ADMR_CODE	Admissions checklist item code which is cross-referenced

<b>Business Cases Using SB_TESTSCORES_VBS</b>	<b>Business Case Issues</b>
Restrict by test code (using Insert, Update, Delete) SORTEST_TESC_CODE in ('1', '2', '3')	Restrict maintenance to specific test codes

### Other Business Case Examples Using SB\_TESTSCORES\_VBS

By admission request checklist code  
 SORTEST\_ADMR\_CODE in ('1','2','3')

Objects that may be impacted by restrictions	Possible Impacts
SOATEST	Selection of test scores
SAADCRV	Decision calculator
SRIPREL	Insertion of test scores during creation or updating of recruit and application from electronic prospect
SAAEAPS	Insertion of test scores during creation or updating of application from electronic applicant
Student and Financial Aid views with queries against SORTEST	Selection of records

## Student Test Scores

Domain	Description	Driver	Tables
SB_TESTSCORES_STUDENT_VBS	Test Scores for Learner for Which User has Access	SORTEST	SORTEST

This domain comes with a built-in function that limits access to test scores based on the existence of recruit, applicant, or learner records. If the learner has one of these module records and the user has access to one of them, then they will have access to the associated test scores for the student. If the learner does not have one of these module records, then the user has access to the test scores.

Restriction Mode	Columns Used in Restrictions	Column Description
Select, Insert, Update, Delete	SORTEST_TESC_CODE	Test score code associated with various tests, i.e., SAT, ACT

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Insert, Update, Delete	SORTEST_TSRC_CODE	Test source or how the test score was entered, i.e., data load or self-reported
Select, Insert, Update, Delete	SORTEST_TADM_CODE	Administrative type of a particular test, i.e., locally or nationally administered
Select, Insert, Update, Delete	SORTEST_ADMR_CODE	Admissions checklist item code which is cross-referenced

<b>Business Cases Using SB_TESTSCORES_STUDENT_VBS</b>	<b>Business Case Issues</b>
<p>Restrict access to test scores (using Insert, Update, Delete)</p> <p>Limit user access to test scores if the learner has a record in their processing area (recruiting, application, or general student), and they have access to that record. If the learner does not have one of these records, the user then has access to all test scores.</p> <p>No restrictive predicate is required for the above.</p>	

<b>Other Business Case Examples Using SB_TESTSCORES_STUDENT_VBS</b>
<p>By admission request checklist code</p> <p><code>SORTEST_ADMR_CODE in ('1','2','3')</code></p>

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SOATEST	Selection of test scores
SAADCRV	Decision calculator
SRIPREL	Insertion of test scores during creation or updating of recruit and application from electronic prospect
SAAEAPS	Insertion of test scores during creation or updating of application from electronic applicant

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
Student and Financial Aid views with queries against SORTEST	Selection of records

## Test Codes

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
SB_TESTCODES_VBS	Test Codes	STVTES	STVTESC
S		C	

Test code predicates can be used to limit the List of Values to only the test codes the user is restricted to for insertion. If a Select restriction is placed on the Test Code Validation Table (STVTESC), the same restriction needs to be added using the SB\_TESTSCORES\_VBS or SB\_TESTSCORES\_STUDENT\_VBS domains. If this is not done, you may have to deal with missing descriptions and form error messages during the query process on SOATEST and all other forms that display test scores.

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Insert, Update, Delete	STVTESC_CODE	Test code for the test
Select, Insert, Update, Delete	STVTESC_ADMR_CODE	Admissions materials required with the associated test

<b>Business Cases Using SB_TESTCODES_VBS</b>	<b>Business Case Issues</b>
Restrict by test code (using Insert, Update, Delete) STVTESC_CODE in ('1','2','3')	Restrict to specific test codes

<b>Other Business Case Examples Using SB_TESTCODES_VBS</b>
N/A

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
STVTESE	Selection and maintenance of test codes
SOATEST	Selection of test scores
SAADCRV	Decision calculator
SRIPREL	Insertion of test scores during creation or updating of recruit and application from electronic prospect
SAAEAPS	Insertion of test scores during creation or updating of application from electronic applicant
SOABGIY	Selection and maintenance of test types for institution
SAADCSN	Decision calculator rules
Student and Financial Aid views with queries against STVTESE	Selection of records

## Other GPAs

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
SB_OTHERGPA_V BS	Other GPAs	SORGPAT	SORGPAT

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Select, Insert, Update, Delete	SORGPAT_GPAT_CODE	Other GPA type code
Select, Insert, Update, Delete	SORGPAT_SBGI_CODE	Source from which the other GPA originated

<b>Business Cases Using SB_OTHERGPA_VBS</b>	<b>Business Case Issues</b>
Restrict by GPA type (using Insert, Update, Delete) SORGPAT_GPAT_CODE IN ('1', '2')	

### Other Business Case Examples Using SB\_OTHERGPA\_VBS

By source or background institution code, limiting access based on source

```
SORGPAT_GPAT_CODE IN ( '1' , '2' )
```

Objects that may be impacted by restrictions	Possible Impacts
SOAGPAT	Selection and maintenance of other GPA scores for learner
Student views with queries against SORGPAT	Selection of records

## Student Other GPAs

Domain	Description	Driver	Tables
SB_OTHERGPA_STUDENT_VBS	Other GPAs for learners, for which user has access	SORGPAT	SORGPAT

This domain comes with a built-in function that limits access to other student GPAs based on the existence of recruiting, applicant, or learner records. If the learner has one of these module records, and the user has access to one of them, then they will have access to the other student GPAs.

Restriction Mode	Columns Used in Restrictions	Column Description
Select, Insert, Update, Delete	SORGPAT_GPAT_CODE	Other GPA type code
Select, Insert, Update, Delete	SORGPAT_SBGI_CODE	Source from which the other GPA originated

<b>Business Cases Using SB_OTHERGPA_STUDENT_VBS</b>	<b>Business Case Issues</b>
<p>Restrict access to GPAs (using Insert, Update, Delete)</p> <p>Limit user access to GPAs if the learner has a record in their processing area (recruiting, application, or general student), and they have access to that record. If the learner does not have one of these records, the user then has access to all GPAs.</p> <p>No restrictive predicate is required for the above.</p>	

<b>Other Business Case Examples Using SB_OTHERGPA_STUDENT_VBS</b>
<p>By source or background institution code, limiting access based on GPA type</p> <p>SORGPAT_GPAT_CODE IN ( '1' , '2' )</p>

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
SOAGPAT	Selection and maintenance of other GPA scores for learner
Student views with queries against SORGPAT	Selection of records

## Other GPA Type Codes

<b>Domain</b>	<b>Description</b>	<b>Driver</b>	<b>Tables</b>
SB_OTHERGPACODES_VBS	Other GPA Codes	STVGPAT	STVGPAT

GPA type code predicates can be used to limit the List Of Values to only the GPA codes applicable to data in their area. If a Select restriction is placed on the GPA Type Code Validation Table (STVGPAT), the same restriction needs to be added using the SB\_OTHERGPA\_VBS or SB\_OTHERGPA\_STUDENT\_VBS domains. If this is not done,

you may have to deal with missing descriptions and form error messages during the query process on SOAGPAT and all other forms that display other GPAs.

<b>Restriction Mode</b>	<b>Columns Used in Restrictions</b>	<b>Column Description</b>
Insert, Update, Delete	STVGPAT_CODE	GPA code for the student

<b>Business Cases Using SB_OTHERGPACODES_VBS</b>	<b>Business Case Issues</b>
Restrict by GPA code (using Insert, Update, Delete) STVGPAT_CODE in ('1','2','3')	

<b>Other Business Case Examples Using SB_OTHERGPACODES_VBS</b>
N/A

<b>Objects that may be impacted by restrictions</b>	<b>Possible Impacts</b>
STVGPAT	Selection and maintenance of other GPA type codes
SOAGPAT	Selection and maintenance of other GPA scores for learner

# PII Security

---

The philosophy of Personally Identifiable Information (PII) security is to restrict a person from viewing information that is not related to their own functional processing area. The processing area is most often affiliated with a department, such as a payroll department or a registrar's office.



**Note:** For the Banner® forms that support PII setup, see [“Banner Forms for FGAC” on page 19](#).

## PII Setup Steps

---

### Step 1—Define Domains

Domains define the driver table for your PII processing area, which is where a PIDM must exist for access to be granted. Often, functional locations, such as payroll department and registrar's office, are PII domains. The PII Domain seed data delivered with Banner 7.0 parallels the functional areas on the System Identification Form (GUASYST).

The VBS domains define the driver for a VBS rules. Each domain has a driver table for which the VBS rule is written. The domain is usually the parent table of a module. In this address example, the driver of the domain is SPRADDR.

### Naming Conventions for PII Domains

Preface each domain with the product abbreviation and *B*, such as that for the API package names, followed by the entity name and then *PII*.

Examples:

- SB\_ADMISSIONS\_PII
- GB\_ADDRESS\_PII



**Note:** VBS domains have the ending VBS (SB\_ADMISSIONS\_VBS).

### Creating a Domain

1. Enter your domain into Banner.

Enter the domain code and description on the FGAC Domain Validation Form (GTVFDMN).

2. Enter the Domain Driver Table Definition on the FGAC Domain Code Rules Form (GORFDMN).

The domain driver includes the driver table (where the PIDM must exist) and the domain type of PII. The enable indicator is for turning a particular PII domain 'OFF' while not shutting down all of PII. You also enter the PIDM column name on the driver table. Internal processing creates dynamic SQL that verifies the existence of the PIDM in the PII Domain table.

## Step 2—Define the Table PII is Restricting

You can build PII on multiple tables. Only the SPRIDEN table is included in Banner 7.0. The premise is that the PIDM the user is accessing on the table must exist in one of the PII domain driver tables for them to have access to SPRIDEN.

Recall the example on page 15 where staff in the Admissions, Registration, Student Billing and Student Dean's offices is assigned functional locations of Admissions, Enrollment, General Student. When staff in Admissions opens a form and enters an ID, the process behind the scenes checks to see if the PIDM for the ID entered on the form exists in the driver table for Admissions, Enrollment or General Student. The 'exists' clause that is created for the PII domains is appended to the query against SPRIDEN by FGAC.

The function `GOKFGAC.F_FIND_PII_DOMAIN` is executed behind the scenes to create the code (predicate) that is sent to FGAC. It creates code that checks if the PIDM is in any of the PII domains. This function can be reused for the other tables the user wishes to implement for PII. Other possible tables are SPBPERS, SPRADDR, SPREMRG and basically all of the IDEN tables.

## Examples of PII on SPBPERS

Assume a site wants to restrict access to person data based on work location of the user, and functional location of data for the ID. Staff in the Admissions, Registration, Student Billing and Student Dean's offices is assigned functional assignments of Admissions, Enrollment, General Student. Staff in payroll is assigned functional assignments of Applicant, Benefit and Beneficiary. When Staff in Admissions opens SPAPERS, SAAQUIK or SRAQUIK, the SPBPERS data is blank if the ID does not have a record in Admissions, Enrollment or General Student. If the user tries to insert a record onto the SPAPERS form when one already exists, a duplicate key error will most likely occur.

When staff in the payroll area opens similar Banner forms that have queries against SPBPERS, it will again look as if that data does not exist in the system unless the ID has a record in the HR PII domains.

## Entering PII table on the FGAC PII Table Rules Form (GORFDPI)

This form is required for the definition of the table that PII exists on. This table defines the PII table and the PII function used to create the code that determines if the PIDM exists in the PII domains.

Note the significance of the **Active Indicator**. You can create the policy and uncheck this checkbox to inactivate all PII on this table. The GOKFGAC process does not pick up the table if the active indicator on the table is unchecked.

## Step 3—Run GFPIIADDPOL.SQL to Create the FGAC Policy

Use the BANINST1 User ID to run GFPIIADDPOL.SQL. This step is run by your DBA to create the FGAC policy on the PII table. Once the FGAC policy has been created, PII is active and begins to restrict users. From SQLPlus, run the GFPIIADDPOL.SQL script using the BANINST1 ID.

This script creates one select policy on all tables entered in GORFDPI. If a VBS policy exists on the same table, a second policy will not be created. Only one policy can exist per table and activity for the GOKFGAC package.

## Steps for Starting (Turning on) PII

The DBA needs to complete the following steps:

1. Execute the BANINST1 GOTVBSI0.SQL database trigger. This is a login database trigger that sets global variables, such as your exemption status.
2. Create the policy on the PII table that has been entered on GORFDPI.
3. Check the active indicator for the PII table on GORFDPI.
4. Check the enable indicator for all PII domains that will be used and are listed on GORFDMN.
5. Please refer to the subsequent section describing an implementation plan.

## Viewing policy data from SQLPlus

```
SQL> select * from dba_policies where
2 object_name = 'SSBSECT';
```

OBJECT_OWNER	OBJECT_NAME	POLICY_GROUP
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_DEL	BANINST1	GOKFGAC
F_DELETE_FNC	NO NO NO YES YES YES NO	
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_INS	BANINST1	GOKFGAC
F_INSERT_FNC	NO YES NO NO YES YES NO	
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_SEL	BANINST1	GOKFGAC
F_SELECT_FNC	YES NO NO NO YES YES NO	
SATURN	SSBSECT	SYS_DEFAULT
GOKFGAC_SSBSECT_UPD	BANINST1	GOKFGAC
F_UPDATE_FNC	NO NO YES NO YES YES NO	

You may see four policies per table, separate ones for insert, update, delete, and select. All have the package defined as GOKFGAC and must be owned by BANINST1. For more information on policy names, please refer to [“Policy names” on page 30](#). The GFPIIADDPOL . SQL only creates one, which is for select.



**Note:** You can make a single table be a member of multiple domains. There will always be only one policy for GOKFGAC on the table. When the table is accessed each policy’s package function is executed. We want GOKFGAC to execute once because the functions pick up rules for all domains, group rules for the table.

Another way is to query for your User ID on the GOIFGAC form (see the section below).

## Dropping a Policy

To drop a policy, run the GFGACDROPPOL . SQL script from SQLPlus. This script accepts wild cards for the table name prompt.

## Step 4—Manage User Access to PII Domains

This section explains how to assign user access to PII domains.

## Assigning Domains to Users with the GOAFPUD Form

Use the FGAC Person User Defaults Form (GOAFPUD) to assign PII domains directly to a User. Banner 7.0 PII domain seed data corresponds to the system identification areas on the GUASYST form.

Use the PII Domain Assignments block on the PII User Profile tab to enter assignments directly to a user.

Use the PII User Profile block to assign the home domain and to check the exempt or cross domain search indicators, if applicable. The home domain is the main domain to which the person is associated. When the user adds a new SPRIDEN row, their home domain fills the SPRIDEN\_CREATE\_FDMN\_CODE. If you do not populate the **Home PII Domain** field, the SPRIDEN\_CREATE\_FDMN\_CODE will be null. The user has access to SPRIDEN rows where their home domain equals the create domain on the SPRIDEN record.

## Assigning Domains to Business Profiles

An easier way to assign PII domains to individuals is to first create business profile lists and assign the PII domains to the business profile.

1. Create the business profile on the FGAC Business Profile Validation Form (GTVFBPR) and assign members on the FGAC Business Profile Assignments Form (GOAFBPR).
2. Assign PII Domains to the business profile on the FGAC PII Domain Business Profile Assignments Form (GOAFBPI).

The Business Profile tab on the GOAFPUD form is display-only and lists the business profiles for the user, and shows the PII domains assigned to the business profile.

## Creating Exemptions from PII

---

### Exempting Individuals from PII

On the PII User Profile tab of the GOAFPUD form, check the **Exempt from Personal Identification Information** to exempt the user will be exempt from all PII checks.

#### *Who should be exempt from PII?*

1. Generic user IDs used in Self-Service.
2. Distributed Security users, the BANSECR% users who maintain user accounts and manage Banner security.
3. All batch processes are automatically exempt unless you remove the batch job from the GORFEOB form's list of objects excluded from FGAC processing.

4. You will need to review issues and alternatives, but it may be a requirement that users who create new SPRIDEN rows are exempt.

## Cross Domain Search

### Banner Search Forms—SOAIDEN, FOIIDEN, GOAMTCH, GUIALTI

The SOAIDEN, GOAMTCH, GUIALTI, and FOIIDEN (if Finance is installed) forms are delivered as exempt from PII and VBS. These forms are included in the GORFEOB list of forms excluded from FGAC processing. The *Exit with Value* and *Save* functions for these forms also utilize a procedure to check if the user has cross search permission. If the user has cross search permissions, the PIDM is saved in a temporary access variable. The user can view PII data for this person throughout the current login session.

### Cross Domain Search Indicator

Use the **Cross Domain Search Indicator for Personal Identification Information** field on the GOAFPUD form to indicate that an individual can select an ID and carry permission to the next form for a PIDM the user does not normally have access.

For example, an Admissions application data entry clerk is restricted to PII domains for Recruiting, Admissions, Enrollment and General Student. The user is not allowed to perform a cross domain search. An application is received for an employee. The clerk does a name search on SOAIDEN, which is exempt from all PII and VBS restrictions. They find the employee and select the ID and navigate back to the Admissions page. The ID they selected is in the Key Block with an error that the ID does not exist. The name is also not filled in.

An HR staff -person who processes employee data receives paperwork that a student is a new employee. The HR staff is restricted to only HR PII domains, but they have cross domain searching. They open the Banner HR form and type in the name of the student, and find the student does not exist in the system. They next open the SOAIDEN page (because they know that PII is turned off on that page) to search for the student, and find the match. They select the student and navigate back to the HR page. The ID and name appear in the Key Block.

## Exempting Objects from PII

1. Add the Banner object to the GORFEOB Form and all GOKFGAC VBS and PII functions are effectively turned off.

GSPSECR security checks for the existence of the object in GORFEOB and sets a flag that makes GOKFGAC ignore all VBS and PII restrictions.

- The following objects are delivered in GORFEOB to support PII.
- GOAFPPII
- SOAIDEN

- GOAMTCH
  - GOAEACC
  - FOIIDEN (if Finance is installed)
  - GUIALTI
  - GUASYST
2. DBProcs are not included in GORFEOB, but you can easily control GOKFGAC VBS and PII by adding the following lines:

At the start of your dbprocs:

```
gokfgac.p_turn_fgac_off;
```

At the end of the your dbproc:

```
gokfgac.p_turn_fgac_on;
```

This may be necessary for calculations that required data outside of the user's normal business processing area.

## Self-Service

FGAC functions on the Banner Self-Service products. To create the cross reference between a self-service login ID and a Banner ID, you must code the person on the GOAEACC form.

## The PII Predicate

---

### Viewing the PII Predicate

The FGAC Predicate Inquiry Form (GOIFGAC) displays the status of a policy and gives you the entire predicate for a table.

Enter a user ID and table name in the Key Block in order to the predicate that is applied. The GOKFGAC package embeds comments in the predicate to tell you the domain name and rule names so you can easily trace where the SQL is coming from.

### The PII Predicate's Logical Structure

This section provides some technical background on the function that parses PII access, `GOKFGAC.f_find_pii_domain`.

This function creates the predicate for the PII table. It creates a series of 'exists in' sub-queries for the driver tables of the PII domains the user is assigned and connects them

with an OR. PII Domains assigned to the user are not included if the gobfdmn\_enable\_ind = N.

```
Exists ( select 'x' from <gobdmn_table_name>
Where <gorfdmn_pii_column_name> = <gorfdpi_column_name>
OR ..
```

The process also connects a clause to compare the PIDM to the cross domain temporary PIDM.

```
sys_context('g$_vbsi_context','ctx_fg_pidm_' || to_char(
<gorfdmn_pii_column_name> ) ) = <gorfdpi_column_name>
```

If the PII table is SPRIDEN, the create home domain on SPRIDEN is compared to the user's home domain coded on GUAFPUD.

```
nvl(spriden_create_fdmn_code,'X') = <value from
gobfpud_fdmn_code>
```

If the User has an entry in GOBEACC, the following is added to give the User access to their own record:

```
<gorfdpi_column_name> = <value of gobeacc_pidm where
gobeacc_username = user>
```

The predicate will eventually resemble:

```
(( ( /* pii domain assignments */
exists ( select 'x' from FTVCUST where FTVCUST_PIDM =
SPRIDEN_PIDM ) OR
exists ( select 'x' from FCBEMPL where FCBEMPL_PIDM =
SPRIDEN_PIDM ) OR
exists ( select 'x' from SARADAP where SARADAP_PIDM =
SPRIDEN_PIDM ) ) OR
( /* match gobeacc pidm for self service */ SPRIDEN_PIDM =
11111)
OR ( /* create home domain equals users home domain */
nvl(spriden_create_fdmn_code,'X') = 'SB_TRANSFER_PII')
OR ( /* temp access */
sys_context('g$_vbsi_context','ctx_fg_pidm_' || to_char(
SPRIDEN_PIDM ) ) = SPRIDEN_PIDM )
))
```

## Suggested PII Implementation

---

The following is a suggested phased approach to implementing PII.

## Decide if the GTVSDAX PII PROMPT is set to Y or N

You can optionally require that users with multiple PII domain assignments select one of them as their current PII domain. If the *PIIPROMPT* function on the Crosswalk Validation Form (GTVSDAX) is set to Y, the user must select one domain they wish to access from a list of their PII Domain assignments on the PII Domain Maintenance Form (GOAFPPII).

## Exempt all users from PII

You can run the following script globally to exempt all users. This script should be re-run daily to catch all new users. This must be run by the DBA group for the Banner 7.0 implementation.



**Note:** This is a sample script. Your site may need help from Professional Services to customize and run the script correctly.

```
Declare
Lv_user  varchar2(30);
Lv_rowid varchar2(18);
Cursor users_c is
  Select username from dba_users
Where not exists
  (select 'x' from gobfpud
where gobfpud_fgac_user_id = username);
begin
open users_c;
loop
  fetch users_c into lv_user;
  exit when users_c%notfound;
  gb_userdefault.p_create
  ( p_fgac_user_id => lv_user,
    p_exempt_ind => 'Y',
    p_cross_domain_ind => 'N',
    p_user_id => gb_common.f_sct_user,
    p_fdmn_code => null,
    p_rowid_out => lv_rowid);
end loop;
end;
```

## Turn PII on

Please refer to [“Steps for Starting \(Turning on\) PII” on page 91](#).

## Define Business Profiles

Functional resources on the project need to identify logical business groupings for your site’s population for the purposes of creating business profiles.

For example, group users together if they work in the central Admissions Office; create another group for the Registrar’s Office.

For each logical grouping of users identified:

1. Create a business profile for the group of users in a logical area
  - Add business profile code to GTVFBPR
  - Add users to business profile on GOAFBPR or on the GSASECR Users window (Banner Rules option) during user creation or maintenance

- Assign PII Domains to the business profile on GOAFBPI
2. Reset exemption indicator for users in the business profile, and assign their home domain. This can be done for each user on GOAFPUD, or by having the DBA run the following script.



**Note:** This is a sample script. Your site may need help from Professional Services to customize and run the script correctly.

```

Declare
Lv_user varchar2(30);
Lv_profile_found varchar2(1);
Lv_rowid varchar2(18);
Cursor users_c is
  Select gorfbr_fgac_user_id from gorfbr
  Where gorfbr_fbpr_code = '<business profile code>';
Cursor profile_found_c is
  Select 'x' from gobfpud
  Where gobfpud_fgac_user_id = lv_user;
begin
open users_c;
loop
  fetch users_c into lv_user;
  exit when users_c%notfound;
open profile_found_c;
  fetch profile_found_c into lv_profile_found;
  if profile_found_c%notfound then
  gb_userdefault.p_create
  ( p_fgac_user_id => lv_user,
    p_exempt_ind => 'N',
    p_cross_domain_ind => 'N',
    p_user_id => gb_common.f_sct_user,
    p_fdmn_code => '<home domain>',
    p_rowid_out => lv_rowid);
  else
  gb_userdefault.p_update
  ( p_fgac_user_id => lv_user,
    p_exempt_ind => 'N',
    p_user_id => gb_common.f_sct_user,
    p_fdmn_code => '<home domain>');
  end if;
close profile_found_c;
end loop;
end;

```

## PII Domains

The PII domains listed below are included as seed data in the FGAC Domain Validation Table (GTVFDMN).

The Product Dependency column indicates the products that you must have licensed in order for seed data to be inserted.

PII Domain	Description	Product Dependency
AB_CONSTITUENT_PII	Alumni Constituent PII	Alumni

<b>PII Domain</b>	<b>Description</b>	<b>Product Dependency</b>
AB_ORG_PII	Alumni Organization PII	Alumni
FB_AGENCY_PII	Finance Agency PII	Finance
FB_CUSTOMER_PII	Finance Customer PII	Finance
FB_EMPLOYEE_PII	Finance Employee PII	Finance
FB_MANAGER_PII	Finance Manager PII	Finance
FB_VENDOR_PII	Finance Vendor PII	Finance
PB_APPLICANT_PII	HR Applicant PII Security	Payroll
PB_BENEFITS_PII	HR Benefits PII Security	Payroll
PB_COBRA_PII	HR Cobra PII Security	Payroll
PB_EMPLOYMENT_PII	HR Employment PII Security	Payroll
RB_FINAID_PII	Financial Aid PII	Financial Aid
SB_ADMISSIONS_PII	Student Admissions PII Security	Student
SB_FACULTY_PII	Student Faculty PII Security	Student
SB_GENSTUDENT_PII	General Student PII Security	Student
SB_HOUSING_PII	Student Housing PII Security	Student
SB_RECRUIT_PII	Student Recruit PII Security	Student
SB_REGISTRATION_PII	Student Registration PII Security	Student
SB_TRANSFER_PII	Student Transfer Credit PII Security	Student
TB_ACCOUNT_PII	Accounts Receivable PII	Student or Finance

## PII Domain Driver Tables

---

The following driver tables were delivered as seed data in the FGAC Domain Driver Rules Table (GOBFDMN). A driver table is listed for each PII domain delivered with the FGAC Domain Validation (GTVFDMN). All domains are delivered as disabled.

These entries are inserted if the domain exists in GTVFDNMN. Please refer to the domains seed data table above for product dependencies.

<b>Domain Code</b>	<b>Domain Type</b>	<b>Driver Table</b>	<b>PII Column Name</b>	<b>PII Enabled</b>
AB_CONSTITUENT_PII	PII	APBCONS	APBCONS_PIDM	N
AB_ORG_PII	PII	AOBORGN	AOBORGN_PIDM	N
FB_AGENCY_PII	PII	FTVAGCY	FTVAGCY_AGCY_CODE_PIDM	N
FB_CUSTOMER_PII	PII	FTVCUST	FTVCUST_PIDM	N
FB_EMPLOYEE_PII	PII	FCBEMPL	FCBEMPL_PIDM	N
FB_MANAGER_PII	PII	FTVFMGR	FTVFMGR_FMGR_CODE_PIDM	N
FB_VENDOR_PII	PII	FTVVEND	FTVVEND_PIDM	N
PB_APPLICANT_PII	PII	PABAPPL	PABAPPL_PIDM	N
PB_BENEFITS_PII	PII	PDRBENE	PDRBENE_PIDM	N
PB_COBRA_PII	PII	PCBPERS	PCBPERS_PIDM	N
PB_EMPLOYMENT_PII	PII	PEBEMPL	PEBEMPL_PIDM	N
RB_FINAID_PII	PII	RORSTAT	RORSTAT_PIDM	N
SB_ADMISSIONS_PII	PII	SARADAP	SARADAP_PIDM	N
SB_FACULTY_PII	PII	SIBINST	SIBINST_PIDM	N
SB_GENSTUDENT_PII	PII	SGBSTDN	SGBSTDN_PIDM	N
SB_HOUSING_PII	PII	SLBRMAP	SLBRMAP_PIDM	N
SB_RECRUIT_PII	PII	SRBRECR	SRBRECR_PIDM	N
SB_REGISTRATION_PII	PII	SFBETRM	SFBETRM_PIDM	N
SB_TRANSFER_PII	PII	SHRTTRM	SHRTTRM_PIDM	N
TB_ACCOUNT_PII	PII	TBRACCD	TBRACCD_PIDM	N

# Masking, Hiding, and Concealing Fields

---

Protecting sensitive data in Banner® can be achieved at the field level by the selective use of three techniques:

- Hiding: removing an item from a form by setting its visibility to `False`.
- Concealing: changing the display to show the conceal character instead of the field's data.
- Masking: changing the display mask to partially hide some of the data. This is also referred to as modifying the format of the item on the form.

These three techniques are sometimes collectively referred to as *masking*, but in this document the term *masking* is used specifically to mean changing the field's display mask.

The following Banner forms help set up protecting sensitive data. These are found on the Protection of Sensitive Data menu (\*GENDMSK).

- Data Display Mask Columns (GORDMCL)
- Data Display Mask Rules (GORDMSK)

## Setup for Protecting Sensitive Data

---

### List Items in the Data Display Mask Columns Data Dictionary

The first step in protecting sensitive data at the field level is to define the items that need protection and enter them on the Data Display Mask Columns (GORDMCL) data dictionary form.



**Note:** The data entry task is meant for a technical resource. The technical resource needs to know the block name and the item name to enter the data on GORDMCL. This can be determined by using the Search options to view item properties. In some cases, the form must be opened using Oracle Developer to find the block and item names.

You can apply masks on the Data Display Mask Rules Form (GORDMSK) once the items that need protection are identified and entered on GORDMCL.

Enter all items that you want to secure in the data dictionary on GORDMCL. The items that have the system required indicator checked are the seed data delivered by

SunGard®. The seed data includes items SunGard supports for enhanced character masking.

The data types are character, date, numeric, and button. If you wish to remove a character item from a form and the item has a search button, remember to also remove the search button from the form.

## Setting up Formatting Rules on the GORDMSK Form

A formatting rule can be applied either to all users, a specific user, or a group of users through the use of a business profile. Business profiles are set up on the FGAC Business Profile Validation form (GTVFBPR) and the FGAC Business Profile Assignments form (GOAFBPR).

You can create multiple rules for the same form, block and item. The sequence number creates uniqueness. The data type, length and numeric precision are display only on this form and provide limited validation on the format masks entered.

There are three basic options available for the protection of sensitive data. These options are available on all Banner forms and do not require changes to the form. The three basic options are hiding, concealing, and masking a number or date.

You cannot select a format mask on GORDMSK and apply it to a character field, so a different method has been provided to apply partial character masking to a character field.

### Hiding

Hiding means setting the visibility property to *FALSE*

*FALSE* translates to making the item invisible on the form. The prompt for the item is also removed from the form. To set an item's visibility property to *FALSE*, uncheck the **Visible Indicator** checkbox on GORDMSK.

### Concealing

Concealing means setting the conceal property to *TRUE*

*TRUE* translates to concealing the item. This means the item is visible but the value is replaced with all asterisks (\*). Example: The SSN is 123456789. After concealing the item the value, it appears as *\*\*\*\*\**.

To set an item's conceal property to *TRUE*, check the **Conceal Indicator** checkbox on GORDMSK.

### Masking a Number or Date

You can select a format mask for a date or numeric item. Enter the format mask code in the **Format Mask** field on the GORDMSK form. There are literally thousands of possible format masks that can be created using the standard format mask elements provided by

Oracle, but not all of the combinations are valid. See Online Help for the GORDMSK form for

- a list of examples of valid format masks
- guidelines for constructing format masks
- precautions and tips for avoiding problems with format masks



**Warning!** Before using a format mask in production, you must test it to make sure that it works as expected. This is particularly important with date format masks. See “Testing Date Format Masks” in the Online Help for GORDMSK.

## Partial Character Masking

You can partially mask a character field. For example, you could choose to mask most of an ID number, while leaving just a few characters readable to allow a user to verify that the number in the field is the correct one.

Partial character masking allows a specified number of characters at the right side or the left side of the field to remain readable, while the rest of the value is masked.

To apply partial character masking:

1. In the **Partial Character Mask** field, select `Left Direction` (to leave the leftmost characters readable) or `Right Direction` (to leave the rightmost characters readable).
2. In the **Partial Unmasked Length** field, specify the number of characters that you want to remain readable. The remaining characters will be masked.

For example, to show only the final three digits of a Social Security Number field, select `Right Direction`, and enter 3 for the **Partial Unmasked Length**. An example of the resulting masked field might look like this: `*****789`.



**Note:** This partial character masking solution, introduced in Release 8.0, replaced the limited character masking solution provided with Release 7.0. The older solution required custom programming and was not usable in some forms and fields. With the newer partial character masking method, it can be set up as easily as any other masking option.

## SSN Field on the Extended ID/Name Search Window

Visibility of the Social Security Number (SSN) field on the Extended ID/Name Search Window is not controlled by field-level rules for masking, hiding, or concealing. Instead, this field’s visibility is controlled through Banner object security. The SSN field in this window will display for a user only if the user has been granted access to the `SSN_SEARCH` security object.

Users who do not have security access to the `SSN_SEARCH` security object will not see the SSN field on the Extended ID/Name Search Window and will not be able to do an extended search by SSN.

See the Banner Security Administration Handbook for more information about `SSN_SEARCH` and other special security objects.

## Masking “International” Person Information Fields

Four person information fields were added to the database tables in Banner General 8.0. With Banner General 8.2, in support of international formats for names, telephones, and addresses those fields can now be displayed on the Banner INB pages and Self Service pages if the Institution desires to do so.

- **House Number** (`_HOUSE_NUMBER`)
- **Street Line 4** (`_STREET_LINE_4`)
- **Telephone Country Code** (`_CTRY_CODE_PHONE`)
- **Last Name Prefix** (`_SURNAME_PREFIX`)

Because most Banner customers were expected to choose not to use these four fields, masking rules were delivered to hide these fields by default. You can show or hide these fields by doing either of the following:

- Running the `gssmask.sql` script. The script does a mass update of masking rules for these four fields, so that you can change the rules all at once for the fields wherever they appear in Banner and Banner Self-Service.
- Changing the masking rules for these fields in the Data Display Mask Rules form (`GORDMSK`). There is a separate rule for each field on each form, adding up to several hundred rules in total, so making manual changes on `GORDMSK` is much slower than making mass changes through the `gssmask.sql` script. You should use the `gssmask.sql` script if possible and resort to manual changes only if the script does not provide fine enough control for your needs.

## Using `gssmask.sql`

The `gssmask.sql` script, delivered with Banner General 8.2, can do a mass update of masking rules for any or all of the four new person information fields. When you update a field with `gssmask.sql`, the changes apply to all forms where the field appears, and also all Self-Service pages where the field appears.

In addition to the four fields added in Release 8.2, the `gssmask.sql` script can also be used for a mass update of the **International Access Telephone** field. This field was used for freeform telephone numbers during the time when Banner required that telephone numbers conform to U.S./Canada standards. Banner's telephone fields now support all international telephone number formats, and with the introduction of the **Telephone Country Code** field, the **International Access** field is no longer needed. If your institution chooses to hide the **International Access** field on all Banner forms, you can use the `gssmask.sql` script to do so.



**Warning!** You should not hide the **International Access Telephone** field until after you have taken steps to review its existing data and migrate the data to the other telephone fields.

When you run `gssmask.sql`, the script will prompt you for the following options:

- The field(s) for which you want to change masking rules. Specify one or more of the five fields.
- Whether you want to show (make visible) or hide the specified fields.

If you want to *selectively* show one of the new fields, for example, to show **Street Line 4** on some forms but hide it on other forms, you will have to manually change the field's masking rules on the GORDMSK form.



**Note:** You can minimize your manual effort by using `gssmask.sql` to set a field's visibility as it applies on most forms, and then using GORDMSK to change the rules for the exceptional cases.

## Using GORDMSK to Hide the “International” Fields

The Data Display Mask Rules form (GORDMSK) maintains rules for masking, hiding, or concealing fields on Banner forms. New GORDMSK rules are delivered with Release 8.2 for each of the new person information fields on each of the forms where they appear. By default, the new fields are set to hidden; that is, **Visible** is unchecked on GORDMSK for the new fields.

You can use `gssmask.sql` to make a mass update of the **Visible** indicator for each of the new fields across all Banner forms where they appear. You can also manually change the rules one at a time on GORDMSK.

## Hiding the “International” Fields in Self-Service

The GORDMSK form also allows you to manage showing and hiding the same four person information fields in Banner Self-Service. To modify a masking rule for one of the new fields in Self-Service:

1. On GORDMSK, select the object `**SSB_MASKING`. This is a special object added for the purpose of applying masking rules to fields in Self-Service.
2. Select the rule for the field that you want to modify.

For `**SSB_MASKING` rules, the **Block** field shows the Self-Service function or procedure that displays the data.



**Note:** You can use wildcards in the **Block** field to select multiple Self-Service functions or procedures.

3. Change the **Visible** check box. It should be checked to display the field, or unchecked to hide the field.

With `**SSB_MASKING` rules, you cannot use **Display Mask**, **Conceal**, or the partial character masking feature.

4. Save your changes.



**Note:** This Self-Service masking functionality has been put in place only for these four person information fields and the **International Access Phone** field, and only for packages specifically enabled for this enhancement. To use this feature for other Self-Service fields or in other packages, you would have to make custom modifications to your Self-Service packages.

## Including Last Name Prefix in Full Name

The **Last Name Prefix** field can be included in concatenated full names. A special masking rule was delivered with Release 8.2 that hides the last name prefix from the function that creates the full name.

To include the last name prefix in the concatenated full name, use the following steps to edit the special masking rule.

1. On GORDMSK, select the object `**SSB_MASKING`.
2. Select the rule for the **Block** `F_FORMAT_NAME` and the **Item** `SPRIDEN_SURNAME_PREFIX`.
3. Check the **Visible** check box.
4. Save your changes.



**Note:** In this masking rule, **Visible** is the only masking attribute applied. You cannot use the **Conceal** attribute, apply a format mask, or use partial character masking.



**Note:** After changing this rule, you will have to log out of Banner and log in again in order to see the change take effect.

After this masking rule has been changed, the last name prefix will be included in the concatenated full name. If a last name prefix exists for a given record, then the last name prefix will appear, followed by a separator space and the last name.

## Priority of Rules

A GOQRPLS procedure, `G$_MASKS.G$_STARTUP_MASKS`, executes from the `G$_FORM_STARTUP` to set item properties according to the basic options.

Each GORDMSK formatting rule will be assigned either to ALL USERS, a business profile, or an individual user. You can build multiple rules per item.

The following may be a typical implementation:

Form SOAIDEN, Block SPRIDEN, Item BIRTHDATE

- ALL USERS set visibility to `FALSE`
- Business Profile HRUSERS set visibility to `TRUE` and apply format mask DD-MON
- User HREXECVP set visibility to `TRUE`

Formatting rules for ALL USERS are implemented first, followed by the business profiles to which the user belongs, and finally those set for the individual users. Thus, the individual user's rules supersede the rules for a business profile and for ALL USERS.

## Implementation Issues with Masking, Hiding, and Concealing

GORDMSK rules setup should be planned carefully to avoid creating problems for the end user. You must test your masking, hiding, and concealing rules by logging in as an end user to verify that the settings do not cause issues. Take note of the specific issues listed below when planning your rules.

### The First Field in a Block

An Oracle error may occur when Banner attempts to navigate to a field that is not navigable. The error most often occurs when a `go_item` command points to the first field in a block and the field is either hidden (**Visible** is unchecked) or concealed (**Concealed** is checked). Both hiding and concealing prevent a field from being navigable. An example of the Oracle error you might see as a result is *FRM-40112: Attempted go\_item to non enabled item DISPLAY\_OPTIONS:FORMNAME\_CB*

The problem will not occur on all forms. Many forms use a `go_block` command instead of `go_item`. On those forms, this issue will not arise. But there are forms that must use `go_item` because of form processing dependencies, and in those cases the Oracle error may occur when you try to protect the data in the first field in a block.

This same error can occur on any field, if a `go_item` command is used to navigate to the field and the field is hidden or concealed. It affects the first field in a block most often simply because the first field tends to be the target of `go_item` commands in Banner.

### Hiding All Fields on a Block or Form

Rules should be set up in a way that ensures that at least one field will be navigable for any user on a block or form. Hiding or concealing all fields for any given user will result in an Oracle navigation error when that user access the block or form. For example, the user may see the error *FRM-40106: No navigable items in destination block*.

Protecting all fields on a form is not recommended or supported. Hiding, concealing, and masking are not intended to hide the entire contents of a form from a user. A better way to

achieve that goal is to use Banner's user/object security (on GSASECR) to prevent access to the form.

When building rules, we recommend hiding fields selectively so that in all cases there is at least one navigable item.

## Incorrect or Mistyped Item Names

If you enter an item on GORDMCL that is incorrect and then assign masking to the item on GORDMSK, an error will occur when an affected user opens the form.

For example, on the SOAIDEN form, the block is SPRIDEN. If the SPBPERS block is typed instead of SPRIDEN, the masking procedure returns an error because the SPBPERS block cannot be found.

The error *FRM-40105 Unable to resolve reference to item* appears when the form is opened, and may in some cases prevent the form from opening at all.

GORDMCL cannot validate the existence of an item on a form. When setting up rules on GORDMCL, you should manually double-check the object, block, and item name.

## Masking a Required Data Entry Field

Assigning a mask to a required item on a data entry form will make it impossible for users to enter new records on the form. The masking process makes the item non-updateable so the user cannot enter a value. The user will not be able to successfully insert records because the item is protected against update.

## Masking All Items on a Search Form

Assigning masks to all items on a search form makes it unusable. The masking turns "enter query" off for items. The end user will be unable to enter a query if format masks are implemented on all queryable items on a search form.

## Masking or Concealing Banner ID

Protecting the Banner ID field on search forms, either by assigning a character mask to the ID or concealing the field, is not recommended.

If the Banner ID is concealed on the form, the end user eventually sees the ID anyway by selecting it and navigating to a form with the ID in the key block. The ID carries over to the newly opened form even though it was concealed on the search form.

If a character string format mask is applied to the ID, the mask-modified version of the ID will carry over to the subsequent form. For example, suppose the ID has been formatted to `XXX*****`. The ID that carries over to the new form will be in that same format.

## Dropdown Lists

If you need to prevent users from seeing the values on a dropdown list, the only option available is to hide the list (in other words, to set the item's visibility to `False`).

The current implementation does not support concealing or masking the values on a dropdown list.

# Controlling and Troubleshooting FGAC

---

Oracle FGAC is active across the enterprise database. Once a policy on a table is in place, the policy occurs regardless of the method of accessing the database.

Here are some simple guidelines for controlling FGAC.

1. Add a Banner® object to GORFEOB, and all GOKFGAC VBS and PII rules are effectively turned OFF.

GSPSECR security checks for the existence of the object in GORFEOB and sets a flag that makes GOKFGAC ignore all VBS (and PII) rules.

2. DBProcs are not included in GORFEOB, but you can easily control GOKFGAC VBS and PII by adding the following lines:

At the start of your dbproc:  
`gokfgac.p_turn_fgac_off;`

At the end of the your dbproc:  
`gokfgac.p_turn_fgac_on;`

This may be necessary for calculations that required data outside of the user's normal business processing area.

3. Self-Service

FGAC works for Banner Self-Service. To create the cross-reference between a Self-Service login ID and a Banner ID, you must code the person in GOAEACC.

## Debugging and Oracle Trace Output

---

The parsed predicate does not display if you query a table in SQLPlus with `Set Autotrace on Explain`.

Example:

```
Set autotrace on explain
Select * from spraddr where spraddr_pidm = 12345;
```

You can view details from the GOKFGAC execution that parses your predicate, if you first issue the command `ALTER SESSION SET SQL_TRACE TRUE` and run `tkprof` on the trace file. Note that the trace file does not show the predicate, it will only help with the GOKFGAC parsing process.

The best (and only) way to debug is to use `GOIFGAC` and query by User ID and table.



**Note:** In order to see the VBS predicates during SQL tracing you can enter the command:

```
ALTER SESSION SET EVENTS '10730 trace name
context forever, level 12';
```

This will show the predicate being used during the select process.

## Policy Errors

---

### SELECT

If a VBS rule is for SELECT, and you query outside the values defined in the rule, a no rows found message is returned.

Example VBS Rule on SPRADDR: SPRADDR\_STAT\_CODE = 'PA'

Example Query 1: SELECT SPRADDR\_PIDM FROM SPRADDR WHERE SPRADDR\_STAT\_CODE = 'MD'

The above query returns a no rows found.

Example Query 2: SELECT SPRADDR\_PIDM FROM SPRADDR WHERE SPRADDR\_STATUS\_IND = 'I';

The above query returns all rows with a state code of PA that have a status indicator of I.

### INSERT

If a VBS rule is for INSERT and you insert outside the values defined in the rule, a Check Policy Oracle error message is returned.

Example VBS Rule on table STVSTAT: STVSTAT\_CODE = 'PA'

Example Insert 1: INSERT INTO STVSTAT (STVSTAT\_CODE, STVSTAT\_DESC, STVSTAT\_ACTIVITY\_DATE) VALUES ('MD', 'Maryland', SYSDATE);

The above returns the Oracle error 28115, policy error - ORA-28115: policy with check option violation.

Banner forms return the error message *Security violation, transaction not complete*. You can view the Oracle error by selecting the Display Error function from the help toolbar.

Cause: The policy predicate was evaluated to FALSE with the updated values.

## UPDATE

If a VBS rule is for UPDATE, and you change a value that is defined on the rule to one that is outside the rule you will get the Check Policy Oracle error message. If you change a value on a column, and the column is not part of the rule but one of the columns is defined in the rule and its value is outside the rule, you will get a check policy error.

Example VBS Rule for SPRADDR: `SPRADDR_STAT_CODE = 'PA'`

Example Update 1:

```
UPDATE SPRADDR
SET SPRADDR_STAT_CODE = 'MD'
WHERE SPRADDR_PIDM = 1234 AND SPRADDR_STAT_CODE = 'PA';
```

The above returns the Oracle error 28115, policy error. - ORA-28115: policy with check option violation.

Banner forms return the error message *Security violation, transaction not complete*. You can view the Oracle error by selecting the Display Error function from the help toolbar.

Cause: Policy predicate was evaluated to FALSE with the updated values.

Example Update 2:

```
UPDATE SPRADDR
SET SPRADDR_ACTIVE_IND = 'I'
WHERE SPRADDR_PIDM = 1234 AND
SPRADDR_STAT_CODE = 'MD';
```

The above update does not return an error, although if you re-query that row you will see that the update did not take place. The where-clause of an update statement is controlled by the Oracle SELECT policy. You can select all rows from SPRADDR, but you can only update those with a state code of PA.

The API DMLs were modified to interrogate the `sql%row_count` directly after the update. If the row count is 0, an error is raised. The message displayed is *Update failed. Exactly one row must be updated*.

## DELETE

If a VBS rule is for DELETE, you never get a check policy error if you delete outside of the VBS rule.

The API DMLs were modified to interrogate the `sql%row_count` directly after the delete. If the row count is 0, an error is raised. The error displayed is *Delete Failed. Exactly one row must be deleted*.

# Best Practices and Guidelines for FGAC Implementation

---



**Note:** Note: SunGard® recommends hiring technical professional services for implementation of FGAC using VBS and PII, especially when setting up complex rules.

## Oracle Parameter `_DYNAMIC_RLS_POLICIES`

Since FGAC can be turned off for specified objects, the parameter `_DYNAMIC_RLS_POLICIES` must not be set as `FALSE` if using FGAC-based VPD or PII. The default value for this hidden parameter is the desired value of `TRUE`, so you only need to change your initialization parameter file if you have previously set it to `FALSE`. The way to set this parameter depends on whether you are using an `INIT.ORA` or a `SPFILE`.

- If you are using an `INIT.ORA`, edit it and remove or comment out the following line, if present:

```
_DYNAMIC_RLS_POLICIES=FALSE
```

- If you are using an `SPFILE`, then enter the following SQL while logged on with DBA privilege:

```
ALTER SYSTEM SET "_DYNAMIC_RLS_POLICIES"=TRUE  
SCOPE=SPFILE ;
```

In either case, the database must be bounced in order for the change to take effect.

The VBS predicate functions called from GOKFGAC store the final result as a context variable. The next time the table is accessed for the database activity (Select, Insert, Update, Delete) the predicate is retrieved from the context variable. Clients using Banner FGAC can turn FGAC off for specified objects.

Example: SOAIDEN, FOIIDEN, GUIALTI, GOAMTCH and GUISYST are delivered as exempt from FGAC. The Banner security procedure GSPSECR will set a flag to turn off FGAC for these objects and any other exempt objects. When GOKFGAC predicate functions execute and see this flag turned off it will return a null predicate. The user may execute a different Banner object which does not have FGAC turned off, and the GOKFGAC predicate function will return the full predicate.

## Business Profiles

Create business profiles to reduce data entry in assigning users to VBS rules. Remove users from business profiles when they leave their job position. Include new procedure in security practices to remove them from business profiles when user leaves position.

To view business profiles in which the user is a member, access the FGAC Person User Defaults Form (GOAFPUD) and open the Business Profile Assignments tab.

## Creating Rules

Be aware of all the table dependencies for the table for which you are creating the rule. Review ERD diagrams, or use Banner Enlighten for this purpose. Search for all programs that access the table for which you are creating restrictions. You should test VBS rules on all places the table is accessed. Banner Enlighten can be used for this purpose.

## Building Restrictions

Do not build restrictions that allow `update` or `insert` but not `select`. The rule for `select` should not be less than DML privileges.

Example: VBS Rule for SPRADDR: `SPRADDR_STAT_CODE = 'PA'` and the `Select` is checked on GOAFGAC. `Update`, `Delete` and `Insert` are unchecked. User can select any rows from SPRADDR with a state code PA. The user can execute the following in SQLPlus:

```
Update spraddr  
Set spraddr_activity_date = sysdate;
```

Result: The above SQL will update all rows in SPRADDR. Had the VBS rule included `Update`, `Delete` and `Insert` checked, the above SQL would update just the rows with a state code of PA.

## Checking Access

Check individual user access by table on GOIFGAC after setting up rules on GOAFGAC.

The following setup situation may occur: A user exists in a business profile which has `select`, `insert`, `update` and `delete` restrictions on a particular VBS rule. The user is added to the User Access section of the same rule with `insert`, `update` and `delete` restrictions.

There are two different `select` rules entered for the user. The GOKFGAC predicate process first pulls the rule for a user where the `select` is checked. In the above case, the business profile `select` restriction will be used.

## Verifying Rules

Run an explain plan on the predicate to verify the rule does not introduce performance issues. Use the steps that follow to run an explain plan.

Run trace on forms where predicates on tables are used to verify index usage is not changed. This is a continuation of the above rule. Verify that VBS rules, when used in

Banner objects, do not change the indexes used to access tables which may result in full table scans or create other performance issues.

1. Open GOIFGAC and query the table. Use the editor button to get a full screen of the predicate. Select the SQL and copy to clipboard.

2. Create a SQL file (testvbs.sql)

```
Select 'x' from <fgac table>  
Where <copy from clipboard>;
```

3. From SQLPlus:

```
Set autotrace on explain  
@testvbs
```

The ID that executes the above needs appropriate privileges to create explain plans in SQLPlus.

## Reviewing GOIFGAC

Review GOIFGAC for each user/table after rules are constructed. There may be other rules on the same table that may impact the user and how they access the table.

## Where Not to Use VBS

- Avoid VBS rules on common tables, such as SPRIDEN and SPBPERS.
- Avoid creating VBS restrictions on parent tables that are based on children record results.

Example: Using domain GB\_SPRADDR\_VBS where the driver table is SPRIDEN, set up rule on GOAFGAC with the following:

```
Select, Insert, Update and Delete  
EXISTS ( SELECT 'X' FROM SPRADDR WHERE  
SPRADDR_PIDM = SPRIDEN_PIDM AND  
SPRADDR_STAT_CODE IN ( 'MA', 'RI' ) )
```

This will restrict the user from creating a SPRIDEN row. The SPRADDR row cannot exist until the SPRIDEN row has been created.

- Do not create VBS rules with select restrictions on Banner Finance accounting distribution and/or job labor distributions tables where multiple rows need to add up to 100%.
- Do not create VBS rules with select restrictions on Banner Student Catalog and Schedule. The impact is far reaching on display of student data. This data is selected on the registration page, transcripts, student course display, and academic history displays.
- Stay away from using functions in your predicate since this may adversely impact performance. This impacts only tables where large numbers of rows are selected and returned, as in a name search.

## Recursive FGAC Processing

Note that when the same policy table is used in a predicate, it is not subject to FGAC. This will result in recursive FGAC processing.

Example VBS Rule for SPRADDR:

```
SPRADDR_STAT_CODE = 'PA' AND  
EXISTS ( SELECT 'X' FROM SPRADDR SUB  
WHERE SUB.SPRADDR_PIDM =  
SPRADDR.SPRADDR_PIDM AND  
SUB.SPRADDR_ATYP_CODE = 'MA' )
```

Result: FGAC will not execute for the select in the sub-query SPRADDR SUB.

## Working with Different Domains

Rules on the same table in different domains are combined with a logical OR and may not give expected results. You should review predicates on GOIFGAC and test logic. The following is the conditional logic for how predicates from different domains and rules are combined:

```
<executing SQL statement>  
AND ( /*Domain 1 */  
( /*Group 1*/ (Predicate 1 AND Predicate 2 AND Predicate  
3))  
OR ( /*Group 2*/ (Predicate 4 AND Predicate  
OR ( /*Domain 2*/ ( /*Group 3*/...)))
```

## Predicate Errors

Oracle error 28113 means your predicate has a syntax error.

Example: Using domain GB\_SPRADDR\_VBS set up the rule on GOAFGAC with the following:

```
SPRADDR_ATYP_CODES IN (MA,PA)
```

The above is not a valid SQL syntax and will result in the oracle error 28113 when users access SPRADDR.

After typing in a predicate, click the **Validate SQL** button on GOAFGAC to verify no syntax issues are being introduced.

## Syntax for the Join Condition on GORFDPL

Be aware that when adding a table to a domain on GORFDPL that requires a join clause that the end parenthesis of the sub-query should not be added.

Example: Policy table is SARCHKL for domain SB\_ADMISSIONS\_VBS and the domain driver is SARADAP. The VBS rule is:

```
SARADAP_RES_D_CODE = 'M'
```

A normal query to join SARCHKL and SARADAP is the following:

```
SELECT 'X' FROM SARCHKL
WHERE EXISTS
( SELECT 'X' FROM SARADAP
WHERE SARADAP_PIDM = SARCHKL_PIDM
AND SARADAP_TERM_CODE_ENTRY =
SARCHKL_TERM_CODE_ENTRY
AND SARADAP_APPL_NO = SARCHKL_APPL_NO)
```

The partial SQL we want for our driver SQL on GORFDPL for the SARCHKL table is:

```
EXISTS
( SELECT 'X' FROM SARADAP
WHERE SARADAP_PIDM = SARCHKL_PIDM
AND SARADAP_TERM_CODE_ENTRY =
SARCHKL_TERM_CODE_ENTRY
AND SARADAP_APPL_NO = SARCHKL_APPL_NO
```

When the predicate eventually is built, the above partial SQL is appended to the executing SQL statement along with the VBS rule SARADAP\_RES\_D\_CODE = 'M'.

SQL Predicate	Source
Select sarchkl_pidm, sarchkl_admr_code From sarchkl Where sarchkl_pidm = :pidm And sarchkl_term_code_entry = :term_in	User executes this SQL query.
AND	Oracle automatically connects the SQL query to the FGAC predicate with an AND.
Exists ( select 'x' from saradap Where saradap_pidm = sarchkl_pidm And saradap_term_code_entry = sarchkl_term_code_entry and saradap_appl_no = sarchkl_appl_no	The join statement you defined for the domain table is appended.
And saradap_resd_code = 'M'	The VBS rule coded on GOAFGAC along with the AND connector.
)	GOKFGAC concludes the predicate with ) if necessary.

## VBS and Sequence Number Fields in Tables

Beware if the domain driver table has a sequence number as part of the primary key and is incremented using addition based on maximum number of existing rows. This may present key constraint issues.

Example: The domain driver is SARADAP, and the primary key includes SARADAP\_APPL\_NO which is incremented by one based on the number of existing applications for the person.

Our user can only select, insert, update and delete applications with college code 'AS'. The applicant already has an application number 1 for the college 'ED'. The user tries to insert a new application for college AS but the process will not find existing applications and think the application number is 1. This will cause a foreign constraint because the application number 1 already exists. In Banner 7.0 the SARADAP processing has been modified so the application increment process turns FGAC off; calculates the new application number and then turns FGAC back on.

## Using Select Restrictions

Use caution when giving select restrictions to users who perform a lot of data entry. The reason for this is that in Banner, in some areas when data is entered, a substantial amount of internal processing happens behind the scenes. If there are select restrictions, this may impede the additional processing that needs to happen to complete the transaction.

## Restrictions and Validation Table Codes

If select restrictions are built on validation table codes, it is best to build similar restrictions on tables that have that code value. The impact of not doing this is missing descriptions and form error messages during the query process on pages that display the code and description.

Example: Domain GB\_SPRADDR\_VBS with driver SPRADDR and create new domain GB\_STVATYP\_VBS with driver STVATYP.

Create rule with select, insert, update and select restrictions on GB\_STVATYP\_VBS with predicate `STVATYP_CODE IN ( 'MA' , 'PR' )`.

If a user queries Banner ID with addresses MA and BI on SPAIDEN address tab, form errors will appear that the address is not valid. The user may be prevented from viewing the BI address due to validation errors caused by BI not being found in STVATYP.

# FGAC Reference

---

This chapter provides technical details of the Banner® objects that support FGAC processing.

## Tables

---

### FGAC Domain Table (GTVFDMN)

This table stores the valid domains used in FGAC security.

GTVFDMN_CODE	NOT NULL	VARCHAR2 ( 30 )
GTVFDMN_DESC	NOT NULL	VARCHAR2 ( 60 )
GTVFDMN_ACTIVITY_DATE	NOT NULL	DATE
GTVFDMN_USER_ID	NOT NULL	VARCHAR2 ( 30 )

### FGAC Business Profile Validation Table (GTVFBPR)

This table stores the business profile codes to which a user can be assigned for FGAC.

GTVFBPR_CODE	NOT NULL	VARCHAR2 ( 30 )
GTVFBPR_DESC	NOT NULL	VARCHAR2 ( 60 )
GTVFBPR_ACTIVITY_DATE	NOT NULL	DATE
GTVFBPR_USER_ID	NOT NULL	VARCHAR2 ( 30 )

### FGAC Group Code Validation Table (GTVFGAC)

This table stores the groups for which FGAC predicates are defined.

GTVFGAC_CODE	NOT NULL	VARCHAR2(30)
GTVFGAC_DESC	NOT NULL	VARCHAR2(60)
GTVFGAC_ACTIVITY_DATE	NOT NULL	DATE
GTVFGAC_USER_ID	NOT NULL	VARCHAR2(30)

## FGAC Domain Type Code Validation Table (GTVFDTP)

This table stores the types of FGAC predicates defined for domains. The two values delivered with this project are VBS and PII.

GTVFDTP_CODE	NOT NULL	VARCHAR2(5)
GTVFDTP_DESC	NOT NULL	VARCHAR2(60)
GTVFDTP_ACTIVITY_DATE	NOT NULL	DATE
GTVFDTP_USER_ID	NOT NULL	VARCHAR2(30)

## FGAC Domain Driver Rules Table (GOBFDMN)

This table defines the driver table for a FGAC domain.

GOBFDMN_FDMN_CODE	NOT NULL	VARCHAR2(30)
GOBFDMN_TABLE_NAME	NOT NULL	VARCHAR2(30)
GOBFDMN_FDTP_CODE	NOT NULL	VARCHAR(5)
GOBFDMN_PII_ENABLE_IND	NOT NULL	VARCHAR2(1)
GOBFDMN_ACTIVITY_DATE	NOT NULL	DATE
GOBFDMN_USER_ID	NOT NULL	VARCHAR2(30)
GOBFDMN_SYS_REQ_IND	NOT NULL	VARCHAR2(1)
GOBFDMN_PII_COLUMN_NAME		VARCHAR2(30)

## FGAC Domain Type Driver Rule Table (GOBFDTP)

This table defines if a domain type requires a predicate.

GOBFDTP_FDTP_CODE	NOT NULL	VARCHAR2 ( 5 )
GOBFDTP_PREDICATE_IND	NOT NULL	VARCHAR ( 1 )
GOBFDTP_ACTIVITY_DATE	NOT NULL	DATE
GOBFDTP_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GOBFDTP_SYS_REQ_IND	NOT NULL	VARCHAR2 ( 1 )

## FGAC User Business Profile Assignments Table (GORFBPR)

This table stores the FGAC business profile assignments for a user.

GORFBPR_FGAC_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GORFBPR_FBPR_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFBPR_ACTIVITY_DATE	NOT NULL	DATE
GORFBPR_USER_ID	NOT NULL	VARCHAR2 ( 30 )

## FGAC PII Business Profile Assignments (GORFBPI)

This table stores the PII domains for a business profile.

GORFBPI_FBPR_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFBPI_FDMN_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFBPI_ACTIVITY_DATE	NOT NULL	DATE
GORFBPI_USER_ID	NOT NULL	VARCHAR2 ( 30 )

## FGAC User Profile Table (GOBFPUD)

This table defines the home PII domain for a user. This is also where the user can be defined as exempt from all PII processing.

GOBFPUD_FGAC_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GOBFPUD_EXEMPT_IND	NOT NULL	VARCHAR2 ( 1 )
GOBFPUD_FDMN_CODE_HOME		VARCHAR2 ( 30 )
GOBFPUD_CROSS_DOMAIN_IND	NOT NULL	VARCHAR2 ( 1 )
GOBFPUD_ACTIVITY_DATE	NOT NULL	DATE
GOBFPUD_USER_ID	NOT NULL	VARCHAR2 ( 30 )

## PII Domain Maintenance Table (GORFPPII)

This table defines the PII domains for a user.

GORFPPII_FGAC_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GORFPPII_FDMN_CODE_PII	NOT NULL	VARCHAR2 ( 30 )
GORFPPII_ACTIVITY_DATE	NOT NULL	DATE
GORFPPII_USER_ID	NOT NULL	VARCHAR2 ( 30 )

## FGAC Group Definition Table (GOBFGAC)

This table defines the FGAC rule.

GOBFGAC_FGAC_CODE	NOT NULL	VARCHAR2 ( 30 )
GOBFGAC_ACTIVE_IND	NOT NULL	VARCHAR2 ( 1 )
GOBFGAC_EFF_DATE	NOT NULL	DATE
GOBFGAC_ACTIVITY_DATE	NOT NULL	DATE
GOBFGAC_USER_ID	NOT NULL	VARCHAR2 ( 30 )

## User Access for Domain and Group Table (GORFGUS)

This table is where the user's individual access is defined for a specific FGAC group and domain.

GORFGUS_FGAC_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFGUS_FDMN_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFGUS_FPRD_SEQNO	NOT NULL	NUMBER ( 4 )
GORFGUS_FGAC_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GORFGUS_SELECT_IND	NOT NULL	VARCHAR2 ( 1 )
GORFGUS_INSERT_IND	NOT NULL	VARCHAR2 ( 1 )
GORFGUS_UPDATE_IND	NOT NULL	VARCHAR2 ( 1 )
GORFGUS_DELETE_IND	NOT NULL	VARCHAR2 ( 1 )
GORFGUS_ACTIVITY_DATE	NOT NULL	DATE
GORFGUS_USER_ID	NOT NULL	VARCHAR2 ( 30 )

## Business Profile Access to Domain and Group Table (GORFGPR)

This table is where the business profile access is defined for each FGAC Group and Domain predicate.

GORFGPR_FGAC_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFGPR_FDMN_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFGPR_FPRD_SEQNO	NOT NULL	NUMBER ( 4 )
GORFGPR_FBPR_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFGPR_ACTIVITY_DATE	NOT NULL	DATE
GORFGPR_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GORFGPR_SELECT_IND	NOT NULL	VARCHAR2 ( 1 )
GORFGPR_INSERT_IND	NOT NULL	VARCHAR2 ( 1 )
GORFGPR_UPDATE_IND	NOT NULL	VARCHAR2 ( 1 )
GORFGPR_DELETE_IND	NOT NULL	VARCHAR2 ( 1 )

## FGAC Predicates Table (GORFPRD)

This table is where the predicate is defined for the FGAC code and the domain.

GORFPRD_FGAC_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFPRD_FDMN_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFPRD_SEQNO	NOT NULL	NUMBER ( 4 )
GORFPRD_ACTIVITY_DATE	NOT NULL	DATE
GORFPRD_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GORFPRD_PREDICATE		VARCHAR2 ( 4000 )

## FGAC VBS Table Rules Table (GORFDPL)

This table provides list of all VBS tables that are part of the VBS domains. This table acts a launching pad for creating the Oracle policies. The PLUS script GFVBSADDPOL . SQL reads this table to create table policies. The active indicator allows inactivation of VBS rules for a domain and table without dropping the policy.

GORFDPL_FDMN_CODE	NOT NULL	VARCHAR2 ( 30 )
GORFDPL_TABLE_NAME	NOT NULL	VARCHAR2 ( 30 )
GORFDPL_SYS_REQ_IND	NOT NULL	VARCHAR2 ( 1 )
GORFDPL_ACTIVE_IND	NOT NULL	VARCHAR2 ( 1 )
GORFDPL_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GORFDPL_ACTIVITY_DATE	NOT NULL	DATE
GORFDPL_DRIVER_SQL		VARCHAR2 ( 2000 )

## FGAC PII Table Rules Table (GORFDPI)

This table provides list of tables that have PII policies. This table also acts as a launching pad for creating the Oracle policies for PII tables. The PLUS script GFPIIADDPOL . SQL reads this table to create the table policies. The active indicator allows the inactivation of the PII predicates while keeping the policy enabled.

GORFDPI_TABLE_NAME	NOT NULL	VARCHAR2 ( 30 )
GORFDPI_ACTIVE_IND	NOT NULL	VARCHAR2 ( 1 )

GORFDPI_SYS_REQ_IND	NOT NULL	VARCHAR2(1)
GORFDPI_USER_ID	NOT NULL	VARCHAR2(30)
GORFDPI_ACTIVITY_DATE	NOT NULL	DATE
GORFDPI_DRIVER_SQL		VARCHAR2(2000)

## FGAC Excluded Objects Table (GOBFEOB)

This table lists all objects (Jobs and Forms) that are excluded from FGAC processing. When the object is opened and Banner security set up, this table is queried. If the object exists in this table, the context variable to exclude FGAC is set to Y. The GOKFGAC predicate processing returns a null predicate if the context variable is Y. If the object does not exist in this table, all FGAC processing will be executed.

GOBFEOB_OBJECT	NOT NULL	VARCHAR2(30)
GOBFEOB_ACTIVITY_DATE	NOT NULL	DATE
GOBFEOB_USER_ID	NOT NULL	VARCHAR2(30)

## Data Display Mask Columns Table (GORDMCL)

This table stores the columns that can be used in the display mask form. The data dictionary behind this table are the blocks and items from the form. The data in this table will be delivered as required seed data.

GORDMCL_OBJS_CODE	NOT NULL	VARCHAR2(30)
GORDMCL_BLOCK_NAME	NOT NULL	VARCHAR2(30)
GORDMCL_COLUMN_NAME	NOT NULL	VARCHAR2(30)
GORDMCL_SYS_REQ_IND	NOT NULL	VARCHAR2(1)
GORDMCL_DATA_TYPE_CDE	NOT NULL	VARCHAR2(1)
GORDMCL_DATA_LENGTH	NOT NULL	NUMBER(3)
GORDMCL_ACTIVITY_DATE	NOT NULL	DATE
GORDMCL_USER_ID	NOT NULL	VARCHAR2(30)
GORDMCL_QUERY_COLUMN		VARCHAR2(30)
GORDMCL_NUMERIC_PRECISION		NUMBER(3)

## Data Display Mask Rules Table (GORDMSK)

This table stores the display masks.

GORDMSK_OBJS_CODE	NOT NULL	VARCHAR2 ( 30 )
GORDMSK_BLOCK_NAME	NOT NULL	VARCHAR2 ( 30 )
GORDMSK_COLUMN_NAME	NOT NULL	VARCHAR2 ( 30 )
GORDMSK_SEQNO	NOT NULL	NUMBER ( 8 )
GORDMSK_DISPLAY_IND	NOT NULL	VARCHAR2 ( 1 )
GORDMSK_CONCEAL_IND	NOT NULL	VARCHAR2 ( 1 )
GORDMSK_MASK_DIRECTION		VARCHAR2 ( 1 )
GORDMSK_MASK_LENGTH		NUMBER ( 2 )
GORDMSK_ALL_USER_IND		VARCHAR2 ( 1 )
GORDMSK_ACTIVITY_DATE	NOT NULL	DATE
GORDMSK_USER_ID	NOT NULL	VARCHAR2 ( 30 )
GORDMSK_DATA_MASK		VARCHAR2 ( 60 )
GORDMSK_FGAC_USER_ID		VARCHAR2 ( 30 )
GORDMSK_FBPR_CODE		VARCHAR2 ( 30 )

## Person Identification/Name Repeating Table (SPRIDEN)

The following column was added to this table to support PII:

SPRIDEN\_CREATE\_FDMN\_CODE NOT NULL VARCHAR2 ( 30 )

The SPRIDEN API populates this column with the CTX\_FG\_FDMN\_CODE\_HOME value. This is a context variable that houses the home PII domain from GOAFPUD.

## Views

---

### FGAC Restricted Users View (GOVFGAC)

This view lists users who have PII or VBS restrictions.

GOVFGAC_FGAC_USER_ID		VARCHAR2(4000)
GOVFGAC_TABLE_NAME	NOT NULL	VARCHAR2(30)
GOVFGAC_INSERT_POLICY_STATUS		VARCHAR2(4000)
GOVFGAC_INSERT_PREDICATE		VARCHAR2(4000)
GOVFGAC_SELECT_POLICY_STATUS		VARCHAR2(4000)
GOVFGAC_SELECT_PREDICATE		VARCHAR2(4000)
GOVFGAC_UPDATE_POLICY_STATUS		VARCHAR2(4000)
GOVFGAC_UPDATE_PREDICATE		VARCHAR2(4000)
GOVFGAC_DELETE_POLICY_STATUS		VARCHAR2(4000)
GOVFGAC_DELETE_PREDICATE		VARCHAR2(4000)

## FGAC Restricted Users View (GOVFUSR)

This view lists users who have VBS restrictions.

GOVFUSR_FGAC_USER_ID	VARCHAR2(30)
GOVFUSR_FDMN_CODE	VARCHAR2(30)

## Procedures

---

### GOKFGAC Database Package Procedures to Support VBS and PII Processing

#### P\_ADD\_POLICY

This procedure adds a policy on a table and is executed from GVBSADDPOL.sql and GFPIIADDPOL.SQL.

## P\_DROP\_POLICY

This procedure drops a policy on a table and is executed from GFGACDROPPOL . SQL.

## P\_REQUIRE\_PII\_PROMPT

This procedure calls the GTVSDAX function to get the external code from the PIIPROMPT GTVSDAX entry. If Y, the user must identify their active PII domain.

## P\_ENABLE\_FDMN

This procedure enables or disables a domain.

## P\_DOMAIN\_ENABLED

This procedure fills in a context variable to indicate if a domain is enabled or disabled.

## P\_GET\_USER\_INFO

This procedure gets preliminary information about the user from the FGAC Person User Defaults Table (GOBFPUD).

## P\_FIND\_USER\_PII\_DOMAINS

This procedure stores all of user's PII domains for future processing.

## P\_SET\_HOME\_DOMAIN

This procedure sets the home PII domain context variable.

## P\_DISABLE\_FGAC

This procedure sets the exclude object context variable.

## P\_UPDATE\_TEMP\_PIDM

This procedure updates the temporary PIDM context.

## P\_OBJECT\_EXCLUDED

This procedure queries GOBFEOB to see if an object is excluded from FGAC.

## P\_TURN\_FGAC\_OFF

This procedure turns FGAC off and saves the current FGAC status for the period in which it is turned back on.

## P\_TURN\_FGAC\_ON

This procedure turns FGAC back on if FGAC was used at an earlier time.

## P\_SEARCH\_FORM\_UPDATE\_TEMP\_PIDM

This procedure saves the PIDM in a global context area if the user has cross domain search privileges. This is called from select Banner forms.

# Functions

---

## GOKFGAC Database Package Functions to Support VBS and PII Processing

### f\_get\_pidm

This function returns the PIDM stored in the temporary context variable.

### f\_get\_temp\_pidm

This function returns the PIDM to which the user was given temporary access.

### f\_get\_home\_domain

This function returns the home domain.

### f\_is\_user\_exempt

This function returns the user exempt indicator found on the GOAFPUD form.

### f\_find\_pii\_domain

This function returns the predicate for the PII domain.

## f\_spriden\_pii\_active

This function returns a *Y* or *N* if the SPRIDEN GORFDPI entry is active.

## f\_predicate\_header

This standard function is called by `f_common_predicate` for standard preliminary processing.

## f\_find\_predicate

This standard function is called by `f_common_predicate` to select PII access and VBS rules defined for the user and construct the predicate.

## f\_common\_predicate

This function is the mainline for all predicate functions.

## f\_query\_policy

This function is called from the GOVFGAC view and returns `ACTIVE`, `INACTIVE`, or `NOTEXIST` for the status of the policy.

## f\_query\_predicate

This function is called from the GOVFGAC view and returns the `f_common_predicate`.

## f\_select\_fnc

This function returns the predicate for `SELECT` database activity.

## f\_insert\_fnc

This function returns the predicate for `INSERT` database activity.

## f\_update\_fnc

This function returns the predicate for `UPDATE` database activity.

## f\_delete\_fnc

This functions returns the predicate for DELETE database activity.

## f\_validate\_predicate

This function is called from the GOAFGAC form. It checks the VBS predicate for valid SQL code.

## f\_validate\_vbs\_driver

This function checks the VBS driver SQL for valid SQL code. This function is called from the GORFDPL form.

## f\_format\_data

This function is called from the GOQRPLS . G\$\_MASKS . G\$\_APPLY\_STRING\_MASKS procedure. It uses the GORDMSK mask to conceal portions of text data.

# Scripts

---

## GENROLE.SQL

The GENROLE.SQL script creates a new role, BAN\_DEFAULT\_WEBPRIVS, and issues appropriate grants to it. This new role allows for the use of VBS in Self-Service. Running this script is recommended even if you do not have Self-Service, so that minimally the BAN\_DEFAULT\_WEBPRIVS role is created.

# APIs

---

Database API procedures exist for all new database tables. The API contains the procedure to create, update, and delete the database table rows, The APIs include a procedure to execute the validation business logic.

## GORFDPL API

Product	Entity	Filename	Form	Table
GENERAL	VBS_TABLES	gokb_vbs_tables0.sql gokb_vbs_tables1.sql gokb_vbs_tables_r0.sql gokb_vbs_tables_r1.sql gokb_vbs_tables_s0.sql gokb_vbs_tables_s1.sql	GORFDPL.fmb	GORFDPL

FGAC VBS Table Rules define the driver (master) table and additional associated tables for each domain (business function area) defined in the FGAC Domain Validation table (GTVFDMN)

The associated tables are linked to the main driver table by specifying an appropriate SQL statement. VBS table Rules may be set up in advance and enabled with the Active Indicator. When a Table Rule is toggled between active ('Y') and inactive ('N'), the DBMS\_RLS set context function will enable or disable the policies in that domain. After VBS Table Rules have been created, the policies must be added to the database with gorfdpl.sql. This script creates the respective \_SEL, \_INS, \_UPD and \_DEL records in dba\_policies. While only one policy may exist per table, each table may belong to multiple VBS domains.

FGAC VBS processing is centralized in the GOKFGAC package by appending the FGAC Group Rule predicate(s) for a domain (GORFPRD) to the driver SQL (if specified) for each VBS Table Rule (GORFDPL). The benefit of this approach to securing data access and manipulation to only authorized users is that it is completely table-driven.

## GOBFDMN API

Product	Entity	Filename	Form	Table
GENERAL	DOMAIN	gokb_fg_domains0.sql gokb_fg_domains1.sql gokb_fg_domains_r0.sql gokb_fg_domains_r1.sql gokb_fg_domains_s0.sql gokb_fg_domains_s1.sql	GORFDMN.fmb	GOBFDMN

FGAC Domain Definition. The domain is the central entity in the Banner FGAC design. The domain controls the purpose of the restriction. The domain definition table includes all domain types and has a domain type code value which defines if it is *PII* or *VBS*. The domain table also has the driver table associated with the domain. The driver table for VBS defines the purpose of the predicate. The PII driver table determines where the PIDM must exist in order for the user to have access.

## GOBFDTP API

Product	Entity	Filename	Form	Table
GENERAL	DOMAINTYPE	gokb_fg_domaintype0.sql gokb_fg_domaintype1.sql gokb_fg_domaintype_r0.sql gokb_fg_domaintype_s0.sql gokb_fg_domaintype_s1.sql	GORFDTP.fmb	GOBFDTP

FGAC domain type rule for defining predicate requirements. Types are *PII* and *VBS*. The type defines if a predicate on the GOAFGAC Form is allowed. VBS type domains allow entry of a user-defined SQL predicate. PII type domains execute without additional user-defined predicates. You can only enter VBS types on the GOAFGAC Form.

## GOBFGAC API

Product	Entity	Filename	Form	Table
GENERAL	GROUP_RULES	gokb_fg_grprule0.sql gokb_fg_grprule1.sql gokb_fg_grprule_r0.sql gokb_fg_grprule_r1.sql gokb_fg_grprule_s0.sql gokb_fg_grprule_s1.sql	GOAFGAC.fmb	GOBFGAC

This package provides the Common Business interface for the GROUP\_RULES API and supports the FGAC VBS Group Rules Attributes Table (GOBFGAC).

FGAC VBS Group Rules Attributes define the *Active* and *Effective Date* characteristics for FGAC VBS group rules. Inactive rules ('N') are not included in FGAC processing. Active rules ('Y') with a date greater than or equal to the *Effective Date* are included in FGAC processing. To deactivate a rule, set the *Active* indicator to 'N'. The Group Rules attributes are applied to one or more domain/predicate combinations (stored in the GORFPRD table) associated with the Group Rule.

## GORFPRD API

Product	Entity	Filename	Form	Table
GENERAL	VBS_PREDICATE	gokb_fg_predrule0.sql	GOAFGAC.fmb	GORFPRD
		gokb_fg_predrule1.sql		
		gokb_fg_predrule_r0.sql		
		gokb_fg_predrule_r1.sql		
		gokb_fg_predrule_s0.sql		
		gokb_fg_predrule_s1.sql		

This package provides the Common Business interface for the *vbs\_predicate* and supports the FGAC VBS Predicate Rules Table (GORFPRD).

The FGAC VBS Predicate Rules define the `WHERE` clause that will be executed for the associated domain (business function area) and Group Rule as defined on the GOBFGAC Table. Each domain is associated with a valid SQL statement. Multiple domains per group are permitted, as well as multiple predicates per domain. Multiple predicates are each assigned a unique sequence number. If multiple predicate rules are specified for the same FGAC Rule Group, the predicates \* will be treated as `AND` conditions by FGAC processing. Predicate rules that apply to the same business profile/users for different Rule Groups are treated as `OR` conditions by FGAC processing. Only VBS domains may have VBS predicate rules.

## GOBFEOB API

Product	Entity	Filename	Form	Table
GENERAL	FGACEXCLUDEDOBJECTS	gokb_fg_exclobj0.sql	GORFEOB.fmb	GOBFEOB
		gokb_fg_exclobj1.sql		
		gokb_fg_exclobj_r0.sql		

Product	Entity	Filename	Form	Table
		gokb_fg_exclobj_r1.sql		
		gokb_fg_exclobj_s0.sql		
		gokb_fg_exclobj_s1.sql		

This package provides the Common Business interface for `gb_fgacexcluded` objects and supports the GOBFEOB Table.

Excluded objects provide a way to exempt a specific user from all FGAC processing rules (GOKFGAC package) that would otherwise be applied to VBS and PII restrictions assigned to a user ID, or to a business profile assigned to a user ID.

An excluded object will override all FGAC rules (VBS and PII) and allow the user full and unrestricted access to the data associated with the object.

When a user accesses a Banner object (form, batch process, etc.) the excluded object rules are queried to determine if the object exists for the user. If the object exists, the GOKFGAC procedures that configure VBS and PII predicates send back a “true” statement instead of the SQL statements coded in the FGAC VBS Group Rules and the PII table rules.

## GORFBPR API

Product	Entity	Filename	Form	Table
GENERAL	BUSINESSPROFILE	gokb_busprofile0.sql	GOAFBPR.fmb	GORFBPR
		gokb_busprofile1.sql		
		gokb_busprofile_r0.sql		
		gokb_busprofile_r1.sql		
		gokb_busprofile_s0.sql		
		gokb_busprofile_s1.sql		

This package provides an FGAC business profile for grouping users with like responsibilities. This table lists the users associated with each business profile. Business profiles can be assigned to VBS rules. The business profiles are also used to simplify PII domain assignments.

## GORFGPR API

Product	Entity	Filename	Form	Table
GENERAL	BUS_PROF_RULE	gokb_fg_bprofrule0.sql	GOAFGAC.fmb	GORFGPR
		gokb_fg_bprofrule1.sql		
		gokb_fg_bprofrule_r0.sql		
		gokb_fg_bprofrule_r1.sql		
		gokb_fg_bprofrule_s0.sql		
		gokb_fg_bprofrule_s1.sql		

The FGAC Business Profile Rules assign the categories of users and their privileges (Select, Insert, Update, Delete) to each domain and predicate for a FGAC VBS Group Rule. Business Profiles provide a convenient way to categorize multiple users who share the same security access privileges and restrictions that should be enforced with FGAC processing. Users assigned to a business profile automatically inherit the characteristics and predicates of the profile. The same business profile may be assigned to multiple predicate rules within the same Group Rule. If a user is assigned to a business profile with access restrictions it is not necessary to have user specific access restrictions for the same predicate rule for the FGAC Rule Group.

If a user is associated with multiple business profiles that have different access privileges for the same domain and predicate rule within an FGAC Rule Group, the restrictions associated with any one of the business profiles will always be applied in FGAC processing. For example, a user's insert privilege restricted by FGAC processing if they are associated with one business profile that has insert restrictions and also associated with a second business profile that does not have insert restrictions.

## GORFGUS API

Product	Entity	Filename	Form	Table
GENERAL	FGAC_USER_RULE	gokb_fg_userrule0.sql	GOAFGAC.fmb	GORFGUS
		gokb_fg_userrule1.sql		
		gokb_fg_userrule_r0.sql		
		gokb_fg_userrule_r1.sql		
		gokb_fg_userrule_s0.sql		

Product	Entity	Filename	Form	Table
		gokb_fg_userrule_s1.sql		

This package provides the Common Business interface for FGAC\_USER\_RULE and supports the FGAC User Access to Predicate Rules Table (GORFGUS).

The FGAC User Profile Access Rules assign individual users and their privileges (Select, Insert, Update, Delete) to each domain or predicate for a FGAC VBS Group Rule. Users assigned to a rule automatically inherit the characteristics of the predicates of the profile. If a specific user has access restrictions, it is not necessary to include the user in a business profile with access restrictions for the same predicate rule for the FGAC Rule Group.

If a user is also associated with one or more business profiles that have different access privileges for the same domain or predicate rule within a Group Rule, the restrictions associated with any one of the business profiles in addition to their user-specific restrictions will always be applied in FGAC processing. For example, a user will have their insert privilege restricted if they belong to a business profile that has insert restrictions, but their user access does not have insert restrictions.

## GORFPPI API

Product	Entity	Filename	Form	Table
GENERAL	USERPIIDOMAINS	gokb_pii_usrdomain0.sql	GOAFPUD.fmb	GORFPPI
		gokb_pii_usrdomain_r1.sql		
		gokb_pii_usrdomain_r0.sql		
		gokb_pii_usrdomain1.sql		
		gokb_pii_usrdomain_s0.sql		
		gokb_pii_usrdomain_s1.sql		

FGAC PII domains assigned to user ID. This table includes all the PII domain type domains to which an individual user has access.

## GOBFPUD API

Product	Entity	Filename	Form	Table
GENERAL	USERDEFAULT	gokb_pii_usdefault0.sql	GOAFPUD.fmb	GOBFPUD

Product	Entity	Filename	Form	Table
		gokb_pii_usdefault1.sql		
		gokb_pii_usdefault_r0.sql		
		gokb_pii_usdefault_r1.sql		
		gokb_pii_usdefault_s0.sql		
		gokb_pii_usdefault_s1.sql		

This package provides the FGAC user definition for those cases when a user is exempt from all PII and all user VBS rules. The package allows cross-PII domain searching and establishes the PII home domain.

## GORDMCL API

Product	Entity	Filename	Form	Table
GENERAL	DISPLAYCOLUMNS	gokb_display_cols0.sql	GORDMCL.fmb	GORDMCL
		gokb_display_cols1.sql		
		gokb_display_cols_r0.sql		
		gokb_display_cols_r1.sql		
		gokb_display_cols_s0.sql		
		gokb_display_cols_s1.sql		

This package defines forms, blocks, and columns that you can format with a display mask. This table includes all form items for which you can define display masks.

## GORDMSK API

Product	Entity	Filename	Form	Table
GENERAL	DISPLAYMASKS	gokb_display_mask0.sql	GORDMSK.fmb	GORDMSK
		gokb_display_mask1.sql		
		gokb_display_mask_r0.sql		
		gokb_display_mask_r1.sql		

Product	Entity	Filename	Form	Table
		gokb_display_mask_s0.sql		
		gokb_display_mask_s1.sql		

This package provides you with form display masks with limited options to turn visibility off, conceal data, and utilize date and numeric formatting masks. You must define items on the display columns table (GORDMCL) before you can define masking actions.

## GORFDPI API

Product	Entity	Filename	Form	Table
GENERAL	PII_TABLES	gokb_pii_tables0.sql	GORFDPI.fmb	GORFDPI
		gokb_pii_tables1.sql		
		gokb_pii_tables_r0.sql		
		gokb_pii_tables_r1.sql		
		gokb_pii_tables_s0.sql		
		gokb_pii_tables_s1.sql		

PII Table Rules define the policies for each PII domain. PII Table Rules may be set up in advance and enabled with the *Active* Indicator. The driver table should have a SQL statement specified. When a Table Rule is toggled between active ('Y') and inactive ('N'), the `DBMS_RLS` set context function will enable or disable the policies in that domain. After PII Table Rules have been created, the policies must be added to the database with `gorfdpl.sql`. This script creates the respective “\_SEL”, “\_INS”, “\_UPD” and “\_DEL” records in `dba_policies`. While only one policy may exist per table.

FGAC PII processing is centralized in the GOKFGAC package. The benefit of this approach to secure data access and manipulation only to authorized users is that it is completely table-driven.

## GORFBPI API

Product	Entity	Filename	Form	Table
GENERAL	BUSPROFPPII	gokb_busprofpii0.sql	GOAFBPI.fmb	GORFBPI
		gokb_busprofpii1.sql		

Product	Entity	Filename	Form	Table
		gokb_busprofpii_r0.sql		
		gokb_busprofpii_r1.sql		
		gokb_busprofpii_s0.sql		
		gokb_busprofpii_s1.sql		

This package includes all PII domain type domains to which the users defined for a business profile have access.

## Login Script GOTVBSI0

The login database trigger `gt_login_set_vbsi_context` executes three procedures from GOKFGAC.

This trigger is required for PII and must be executed using the BANINST1 ID.

1. `gokfgac.p_get_user_info`  
This procedure gets basic user information, such as if the user is exempt from all PII processing and the PII Home domains.
2. `gokfgac.p_find_user_pii_domains`  
This procedure retrieves the Users PII domains and stores them in a context variable. This will save on database activity when PII access is determined.
3. `gokfgac.p_require_pii_prompt`  
Determine if the site requires the User update the PII domain they wish to access. This is stored on the new GTVSDAX variable PIIPROMPT.

## Context Variables

The following context variables are temporary variables used by the GOKFGAC package. These global temporary variables can be updated only from within the GOKFGAC package functions and procedures. The context area is defined as `g$_vbsi_context`, and all references to these context variables are prefaced with the context area, for example: `g$_vbsi_context.ctx_fg_fdmn_code_home`.

### CTX\_FDMN\_CODE\_HOME

This context variable stores the home domain. The `CTX_FG_FDMN_CODE_HOME` field is populated during the login process, but can be changed on the GOAFPPII form.

## CTX\_FG\_PIDM

This context variable stores the PIDM to which the user is granted temporary access. This field is populated on select Banner search forms (SOAIDEN, GUIALTI, FOIADEN, and GOAMTCH). The user is granted temporary access to the PIDM if they are allowed cross domain searching.

## CTX\_FG\_TABLENAME\_ACTIVITY

Context variable for each table and activity to store the FGAC predicate WHERE clause. This variable is used to store a parsed predicate statement for a table and database activity (select, insert, update, or delete). The first time a user accesses a table with an Oracle policy, the appropriate GOKFGAC function is executed to parse the predicate. The predicate is copied into this context variable. The predicate is derived from the context variable for all subsequent times during the same login session where the same table and activity occurs.

## CTX\_FG\_USER\_EXEMPT\_IND

This context variable indicates if a user is exempt from PII predicates.

## CTX\_FG\_PII\_PROMPT

This context variable indicates if PII prompting is required. This is set during the login trigger. The GTVSDAX PIIPROMPT function is the source of this variable. If that variable is set to Y, the user is required to identify their PII domain on the GOAFPII page.

## CTX\_FG\_EXCLUDE\_OBJECT

This context variable stores whether or not an object is excluded from FGAC security restrictions. This variable will contain a Y/N value gathered during the login procedure. If this variable is Y, FGAC VBS and PII processing is suspended.

The variable `CTX_FG_SAVE_SAVE_STATUS` is used in conjunction with this variable. Some processes require temporary suspension of FGAC VBS and PII and set the exclude object variable to Y, but need to turn it back to the original status after the processing. The original status is stored in this variable.

The procedures `GOKFGAC.P_TURN_FGAC_ON` and `GOKFGAC.P_TURN_FGAC_OFF` are used to toggle the excluded flag. These procedures will not work correctly if called multiple times in nested sequence. For example: Procedure X calls `P_TURN_FGAC_OFF`, and then Procedure X calls Procedure Y. Procedure Y also calls `P_TURN_FGAC_OFF`. The correct processing is for Procedure Y to check if the excluded variable is already set to Y, and not call `P_TURN_FGAC_OFF` if that is true.

# Database Objects

---

## gspsecr.g\$\_verify\_password2\_prd

This object executes `gokfgac.p_object_excluded` and passes the object name. This procedure selects from GOBFEOB for the object passed in the input parameters. The context variable `CTX_FG_EXCLUDE_OBJECT` is set to `Y` if the row exists. Otherwise, this context variable is set to `N`.

## GOQRPLS.PLL

The GOQRPLS procedure is executed from the load form header trigger. This procedure will read the GORDMSK rows and change the form item properties.

The item properties will be as follows:

1. Look at the 'apply to all users' and apply these item properties.
2. Select the rows with a business profile, and determine if the user is in that list. Apply these item properties.
3. Select rows with the User coded and apply these item properties.

## How Form Item Properties Are Applied

Two procedures, each a member of GOQRPLS, are used to apply the masking changes. These are part of the GOQRPLS package `G$_MASKS`.

## G\$\_STARTUP\_MASKS

The `G$_STARTUP_MASKS` procedure will apply the conceal data, visibility off and the date or number masks. This is called in the `g$_form_startup` procedure which is part of all three form classes.

Example of turning the visibility off:

```
If get_item_property
gordmsk_row.gordmsk_block_name.gordmsk_row.gordmsk_column_name, VISIBLE) is TRUE then
    Set_item_property( gordmsk_row.gordmsk_block_name.gordmsk_row.gordmsk_column_name,
VISIBLE, PROPERTY_FALSE) ;
End if;
```

Example of changing the format mask on dates and numbers:

```
If gordmsk_row.gordmsk_data_mask is not null then
If get_item_property( gordmsk_row.gordmsk_block_name.gordmsk_row.gordmsk_column_name,
DATATYPE) not in ('CHAR', 'LONG') then
```

```
then Set_item_property(gordmsk_row.gordmsk_block_name.gordmsk_row.gordmsk_column_name,  
FORMAT_MASK, gordmsk_row.gordmsk_data_mask) ;  
End if;  
End if;
```

## G\$\_QUERY\_STRING\_MASKS

The G\$\_QUERY\_STRING\_MASKS procedure is called from a block's post-query triggers. The GOAMTCH form was modified to call this procedure in the post query. This procedure implements a partial string masking, replacing part of a string value with the conceal (\*) character.

## GOAMTCH Form

This form stores the selected PIDM in the temporary access context variable—if the PII Cross Domain Search checkbox is selected for the user. (See [“Cross Domain Search Forms” on page 16.](#))

## SOAIDEN and GUIALTI Forms

These forms store the selected PIDM in the temporary access context variable, if the user has cross domain search capabilities. (See [“Cross Domain Search Forms” on page 16.](#))

## Seed Data

---

The following seed data supports FGAC. Additional seed data is listed in [“Value-Based Security” on page 23](#) and [“PII Security” on page 89.](#)

## FGAC Domain Type Validation Table (GTVFTYP)

The following domain types are included as seed data.

- VBS
- PII

## FGAC Domain Type Predicate Rule Table (GOBFDTP)

The following is delivered as seed data for this table:

Code	Description	Predicate Required	System Required
PII	Personal Identification Information	N	Y
VBS	Value-Based Security	Y	Y

## FGAC PII Table Rules Table (GORFDPI)

Table Name	Function that assembles the PII Predicate from active PII Domains (GOBFDMN)
SPRIDEN	gokfgac.f_find_pii_domain

## Crosswalk Validation Table (GTVSDAX)

The PIIPROMPT internal code was delivered for this table's internal group code VBS. This code allows an institution to specify if a user must identify their active PII domain. In the **External Code** field, possible values are Y or N.

- Y Yes, the user must identify their active domain on the PII Domain Maintenance Form (GOAFPII).
- N No, the user does not have to identify their active PII domain. This is the default value.



**Note:** When the External Code value is changed, you must exit Banner and then log back on for the change to take effect.

Internal Code Group	Internal Code	External Code
VBS	PIIPROMPT	N

## Data Display Mask Columns Table (GORDMCL)

The following is seed data delivered for the columns and tables that can be used as format masks when applied to the GOAMTCH form.

Form	Block	Column Name	Type	Length
GOAMTCH	GOTCMRT_MATCH	MATCH_BIRTH_DATE	Date	12
GOAMTCH	GOTCMRT_MATCH	MATCH_CITY_STATE_ZIP	Char	30
GOAMTCH	GOTCMRT_MATCH	MATCH_COUNTY_COUNTRY	Char	30
GOAMTCH	GOTCMRT_MATCH	MATCH_EMAIL	Char	30
GOAMTCH	GOTCMRT_MATCH	MATCH_ID	Char	9
GOAMTCH	GOTCMRT_MATCH	MATCH_NAME	Char	99
GOAMTCH	GOTCMRT_MATCH	MATCH_PHONE	Char	30
GOAMTCH	GOTCMRT_MATCH	MATCH_SEX	Char	30
GOAMTCH	GOTCMRT_MATCH	MATCH_SSN	Char	9
GOAMTCH	GOTCMRT_MATCH	MATCH_STREET_LINE1	Char	30
GOAMTCH	GOTCMRT_MATCH	MATCH_STREET_LINE2	Char	30
GOAMTCH	GOTCMRT_MATCH	MATCH_STREET_LINE3	Char	30
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_BIRTH_DATE	Date	12
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_CITY	Char	30
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_EMAIL	Char	30
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_ID	Char	9
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_NAME	Char	99
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_NATN_CODE	Char	30
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_PHONE	Char	30
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_SEX	Char	30
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_SSN	Char	9
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_STATE	Char	3
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_STREET_LINE1	Char	30
GOAMTCH	GOTCMRT_SUSPENSE	MATCH_ZIP	Char	9

## FGAC Excluded Objects Table (GOBFEOB)

All of the Banner objects with a JOBS type from the GUBOBS table are inserted into the GOBFEOB list of objects excluded from FGAC processing.

In addition, GOAFPII, GOAEACC, GUASYST, SOAIDEN, GOAMTCH, and GUIALTI are included in the list of excluded objects as seed data. If the Finance product is installed, the FOIIDEN form is also included as seed data.