

**Banner Technical  
Student Technical Training**

---

---

---

---

---

---

---

---

**Who is in Attendance?**

- **Introductions:**
  - Participants
    - **Banner Experience**
    - **Relational Database Experience**
    - **Other Relevant Experience**
    - **Responsibilities**
  - Instructor
  - SunGard Higher Education

2

---

---

---

---

---

---

---

---



**Banner Technical Training  
Let's Get Started!**

---

---

---

---

---

---

---

---

**Agenda:**  
**Student Technical Training**

- Student System Overview
- Student System Details
  - Major Forms and Related Tables
  - Data Standards
  - Schematics and Process Flows
  - Database Principles and Structures
  - Reports, Processes, Procedures
  - Conversion Issues
  - Hands-on Exercises

4

---

---

---

---

---

---

---

---

**Your Questions?**

- Do you have other issues or topics that you would like to discuss?

5

---

---

---

---

---

---

---

---

**Banner System Overview**

- Where does Banner STUDENT fit into the entire Banner system?
- What are the major components of Banner STUDENT?

6

---

---

---

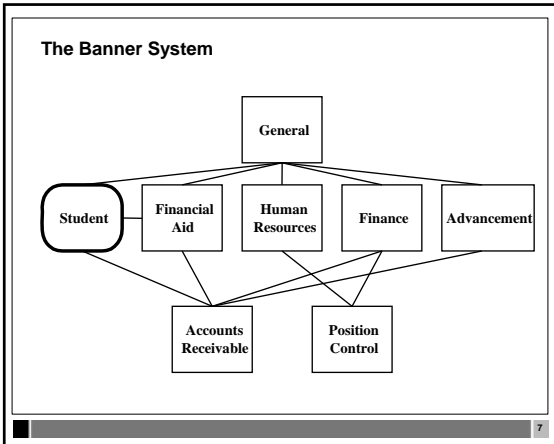
---

---

---

---

---




---

---

---

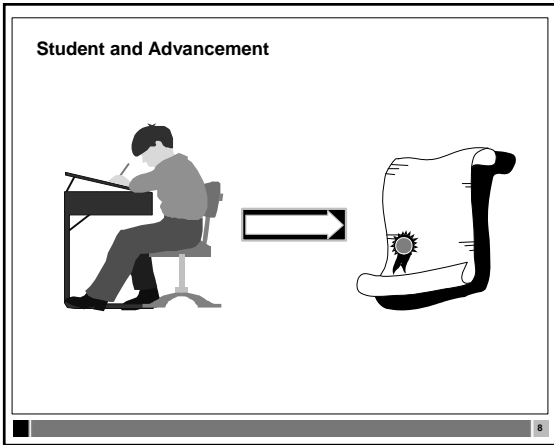
---

---

---

---

---




---

---

---

---

---

---

---

---

- Student and Advancement** (cont.)
- **Student to Advancement Interface (APPSTDI)**
    - adds records that define individuals as constituents
    - updates information on existing constituents

---

---

---

---

---

---

---

---

**Student and Advancement** (cont.)

- **Shared information across Banner**
  - Identification, Person and Address Information
- **Information pulled from Student into Advancement**
  - Academic information is pulled from Admissions, Academic History and Registration
  - Student Cooperative information may also be retrieved for employment history
  - Student activities will also be retrieved

---

---

---

---

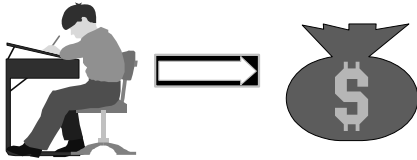
---

---

---

---

**Student and Finance**



---

---

---

---

---

---

---

---

**Student and Finance** (cont.)

- **Accounts Receivable**
  - Charges can be posted to an account through the following Student modules:
    - Admissions
    - Registration
    - Location Management
    - Academic History
    - CAPP
  - Cashiering sessions would be created for the above transactions

---

---

---

---

---

---

---

---

**Student and Finance** (cont.)

• **Accounts Receivable**

- TGRFEED/FURFEED - processes to move the AR transactions from AR to Finance
  - TGRFEED inserts rows into the GURFEED table
  - FURFEED reads each row and loads the data into the Finance system
- TSRRFND/FURAPAY - Processes to move AP transactions from AR to Finance
  - TSRRFND inserts rows into the GURAPAY table
  - FURAPAY reads each row and loads data into the Finance system

---

---

---

---

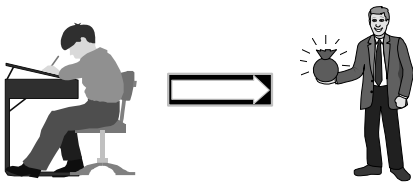
---

---

---

---

**Student and Financial Aid**



---

---

---

---

---

---

---

---

**Student and Financial Aid** (cont.)

• **Disbursements**

- TSASPAY - Student Payment form
  - Users can disburse Financial Aid from this form
    - If the automatic disbursement flag on TSACTRL is checked, then disbursement is done automatically
    - If this flag is unchecked, the user can perform disbursement manually
    - Manual disbursement is performed by entering a 'Y' in the 'Recalculate Financial Aid?' field on the Financial Aid Recalculation window
  - An AR transaction will be created if disbursement occurred

---

---

---

---

---

---

---

---

**Student and Financial Aid** (cont.)

• **Disbursements**

- TSASPAY - Student Payment form
  - Authorized and memoed Financial Aid will display on this form
  - Authorized Financial Aid can reduce the amount due on this form if the Committed/Authorized FA Reduces Amount Due Indicator on the TSACTRL form is checked
  - Memos never reduce amount due
- TSRCBIL - Student Billing Process
  - Can have authorized FA reduce amount due if flag is set on TSACTRL
  - Memos can only be printed

---

---

---

---

---

---

---

---

**Student and Financial Aid** (cont.)

• **Disbursements**

- RPEDISB - allows disburseable aid for a specified term to be credited to a student's account and/or bill in three ways:
  - payments
  - authorizations and/or
  - memos
- Students must pass all user-defined edits and any applicable federal requirements
- Any adjustments made by the Financial Aid office to student awards or due to funds failing disbursement edits may be posted to a student's account and/or bill

---

---

---

---

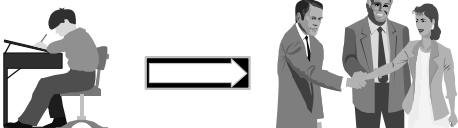
---

---

---

---

**Student and Human Resources**



---

---

---

---

---

---

---

---

**Student and Human Resources (cont.)**

- Data entered through either General Person module is shared
- Reports pull information from both the Faculty Load module and the HR system for reports
  - SIRTAL (Faculty Load Contract Analysis)
    - Salary information can be added with a parameter entry
  - PERFACL (Faculty Load Comparison)
    - Process identifies where data does not match
    - If data matches, total and recording of Total Contact Hours and FTE is updated in the HR system

---

---

---

---

---

---

---

---

---

---

**Shared Student Validation Forms**

Table	Table Description	Alumni	Finance	FinAid	General	HR
STVACCG	Activity Category	A				
STVACTC	Activity	A				
STVACTP	Activity Type	A				
STVACYR	Academic Year				R	
STVADMT	Admission Type				R	
STVAPDC	Admission Application Decision				R	
STVAPST	Admission Application Status				R	
STVASCD	Room Assignment Status				R	
STVASRC	Address Source	A	F	R		H
STVASTD	Academic Standing				R	
STVATYP	Address Type	A	F	R		H
STVBLDG	Building				R	G
STVCAMP	Campus				R	
STVCIPC	CIP Code					H

---

---

---

---

---

---

---

---

---

---

**Shared Student Validation Forms (cont.)**

Table	Table Description	Alumni	Finance	FinAid	General	HR
STVCLAS	Class				R	
STVCNTY	County	A	F	R		H
STVCOLL	College	A		R	G	
STVDEGC	Degree	A		R		H
STVDEGS	Degree Status			R		
STVDEPT	Department			R	G	
STVDIVS	Division			R		
STVDLEV	Degree Level			R		H
STVDPLM	Diploma Type			R		
STVESTS	Enrollment Status			R		
STVETHN	Ethnic	A	F	R		H
STVETYP	Event Type				G	
STVGEOD	Geographic Region Division	A				
STVGEOR	Geographic Region	A				

---

---

---

---

---

---

---

---

---

---







**Banner Student System (cont.)**

- Faculty Load
- General Student
- Accounts Receivable
- Registration
- Academic History
- Curriculum, Advising & Program Planning (CAPP)

28

---

---

---

---

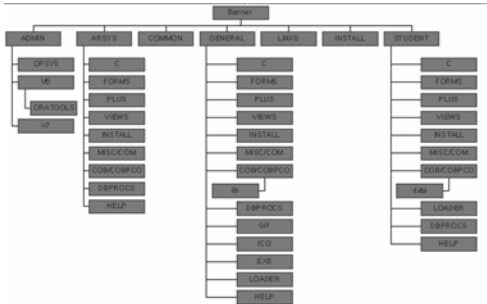
---

---

---

---

**Banner Student Directories**



29

---

---

---

---

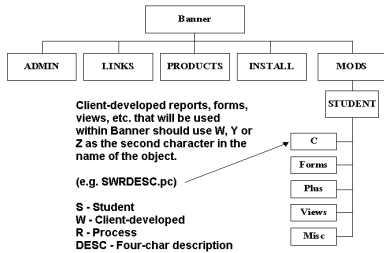
---

---

---

---

**Directory Structure for Client-developed Items**



30

---

---

---

---

---

---

---

---



## Banner Student Technical Review of Database Tools

---

---

---

---

---

---

---

---

### Review

- The Data Dictionary
- GURPED process from Job Submission
- Technical Addendum

---

---

---

---

---

---

---

---

### Database Information

- How do you get more information about the structure and content of tables?
- How do you find out about users, indexes, primary keys, foreign keys?
- How do you find out about table relationships?

---

---

---

---

---

---

---

---

### The Data Dictionary

- A set of tables and views that are used as a read-only reference about the database.
- Stores information about both the logical and physical structure of the database.\*

\* Oracle Server Concepts

34

---

---

---

---

---

---

---

---

### Types of Data Dictionary Views

- USER\_XXXX -- shows objects and events owned by user
- ALL\_XXXX -- shows all objects and events to which user has access
- DBA\_XXXX -- restricted; assigned only to those with DBA role

35

---

---

---

---

---

---

---

---

### Data Dictionary Views

- ALL\_TABLES - descriptions of tables
- ALL\_COL\_COMMENTS - comments on columns of accessible tables
- ALL\_TAB\_COLUMNS - lists of columns of all tables
- ALL\_TAB\_COMMENTS - comments on tables
- ALL\_USERS - information on all users in database
- ALL\_VIEWS - lists text of views accessible to user
- ALL\_INDEXES - descriptions of indexes
- ALL\_IND\_COLUMNS - lists columns of the indexes

36

---

---

---

---

---

---

---

---

**GURPDED Procedure**

- Run from Process Submission Controls Form
  - GJAPCTL
  - Enter Parameters:
    - Table Name
    - Table Owner
  - Output = Technical Addendum
    - To DATABASE
    - View or Print from GJIREVO

---

---

---

---

---

---

---

---

---

---

**Data Dictionary Exercise**

- Find out what indexes exist for the course catalog table (SCBCRSE)
- List the columns in the SCBCRSE indexes that you discovered

---

---

---

---

---

---

---

---

---

---



**Banner Student Technical  
Banner 8 Common Enhancements**

---

---

---

---

---

---

---

---

---

---

### Internationalization Enhancements

- Unicode support
- Additional IDs
  - Unlimited IDs for each person in Banner
  - New table on SPAIDEN
- Expanded fields
  - Name, address, telephone, email, ID, currency amount and rate
- Please refer to the *Banner General 8.0 Release Guide* for more details

---

---

---

---

---

---

---

---

### Partial Data Masking

- The ability to partially mask a field, which was introduced initially in Banner 7.0, has been extended in Banner 8.0 to character fields
- You can allow a specified number of characters at one side of a field to remain readable while masking the remainder of a value
- Two new fields (Partial Character Mask and Partial Unmasked Length) have been added to the Data Display Mask Rules Form (GOTDMASK)
- Please refer to the workbook and to the *Banner General 8.0 Release Guide* for more details

---

---

---

---

---

---

---

---

### PIN Maintenance Enhancements

- PIN Security enhancements
  - Stored in encrypted form
  - Institutions can set strong password standards
  - Better PIN reset mechanism
- Please refer to the *Banner General 8.0 Release Guide* for more details

---

---

---

---

---

---

---

---

### Supplemental Data Engine

- Allows storage of additional data that are not part of the existing Banner data model
- Tied to forms only indirectly
- No form or table customization required
  - Generally not impacted by Banner upgrades
- Some limitations
  - Not all tables, blocks and forms work with SDE
- Please refer to the *Banner General 8.0 Release Guide* for more details

43

---

---

---

---

---

---

---

---

### Common Security Enhancements

- Greater efficiency in security maintenance
- Enhanced Banner user account rules
- Additional user login controls
- Improved security auditing capabilities
- Secured delegation of responsibilities for distributed users

44

---

---

---

---

---

---

---

---

### Common Security Enhancements (cont.)

- Business profiles can be assigned to users on GSASECR
- Security Groups can have security classes, objects, users and roles assigned to them
- New user accounts can be assigned to existing IDs
- Logon calendars

45

---

---

---

---

---

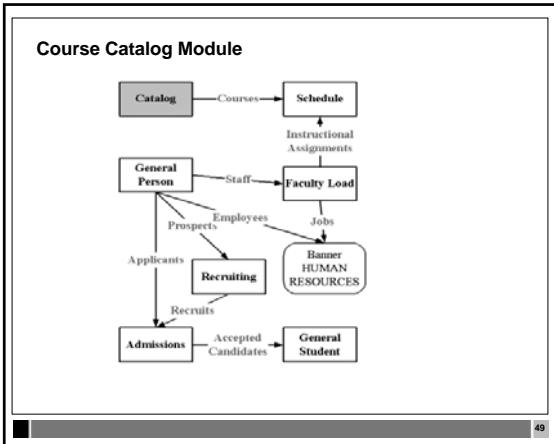
---

---

---








---

---

---

---

---

---

---

---

- ### Course Catalog Module: Objectives
- **Examine/Review**
    - Major and Required Forms and Tables
    - Reports, Processes and Procedures
    - Review Referential Integrity
    - Conversion of Data
- 50

---

---

---

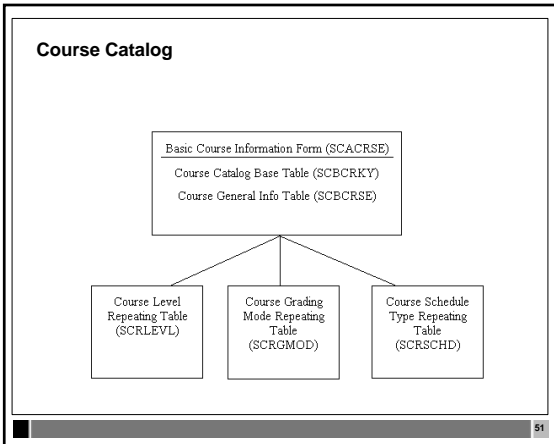
---

---

---

---

---




---

---

---

---

---

---

---

---

**Course Catalog** (cont.)

- **Major Forms:**
  - SCABASE
  - SCACRSE
- **Major Tables:**
  - SCBCRSE
  - SCBCRKY
  - SCRLEVL
  - SCRGMOD
  - SCRSCHD

52

---

---

---

---

---

---

---

---

**Course Catalog** (cont.)

- **Major Validation Tables/Forms:**
  - STVSUBJ
  - STVTERM
  - STVCOLL
  - STVCSTA
  - STVLEVL
  - STVGMOD
  - STVSCHD

53

---

---

---

---

---

---

---

---

**SCACRSE**

- Key Block
- From and To Terms
- Fields Related to AR
- LOV Fields
- Level, Grade Mode, Schedule Type

54

---

---

---

---

---

---

---

---

**Course Catalog: SQL\*Plus**

- How are the SCBCRSE and the SCBCRKY tables related?
- What data elements are required in SCBCRSE, SCBCRKY and SCRLEVL ?
- How are level, grading mode, schedule types connected to a course?

55

---

---

---

---

---

---

---

---

**SCBCRSE and SCBCRKY**

SCBCRSE Name	Null?	Type
-----	-----	-----
SCBCRSE_SUBJ_CODE	NOT NULL	VARCHAR2(4)
SCBCRSE_CRSE_NUMB	NOT NULL	VARCHAR2(5)
SCBCRSE_EFF_TERM		VARCHAR2(6)
..		
SCBCRKY_SUBJ_CODE	NOT NULL	VARCHAR2(4)
SCBCRKY_CRSE_NUMB	NOT NULL	VARCHAR2(5)
SCBCRKY_TERM_CODE_START	NOT NULL	VARCHAR2(6)
SCBCRKY_TERM_CODE_END	NOT NULL	VARCHAR2(6)

56

---

---

---

---

---

---

---

---

**Course Catalog:  
Conversion Issues**

- Will Course Catalog data be converted or entered manually by the users?
- What course catalog data do you have in your legacy system?
- How do you determine where to put it in Banner?

57

---

---

---

---

---

---

---

---

**Course Catalog: Report**

- Reports/Processes
  - SCRBLT -- Bulletin Report
    - Prints catalog of courses
    - Parameter: Academic Year (see STVACYR)
    - C program
    - Run via Job Submission

---

---

---

---

---

---

---

---

**Course Catalog: Summary**

- Major Form: SCACRSE
- Major Tables
  - SCBCRSE
  - SCBCRKY
  - SCRLEVL
  - SCRGMOD
  - SCRSCHD
  - VALIDATION TABLES

---

---

---

---

---

---

---

---

**Course Catalog: Exercise 1**

- Get the following information about any two of the Course Catalog module tables:
  - Table Owner
  - Table Name
  - Column Name
  - Data Type
  - Null/Not Null Column

---

---

---

---

---

---

---

---

**Course Catalog: Exercise 2**

- Get the following information from the course catalog module about all courses with a subject code of 'ENGL':
  - subject code,
  - course number,
  - course title,
  - effective term,
  - start term,
  - end term

---

---

---

---

---

---

---

---

---

---

**Course Catalog: Exercise 3**

- Write a select statement that would produce a catalog report which includes the following (no formatting necessary):
  - subject code,
  - course number,
  - course title,
  - effective term,
  - start term,
  - course level,
  - grade mode

---

---

---

---

---

---

---

---

---

---



**Banner Student Technical  
Review of Referential Integrity**

---

---

---

---

---

---

---

---

---

---

### Data Integrity

- Types of Data Integrity\*
  - Nulls
  - Unique Column Values
  - Primary Key Values
  - Referential Integrity

\* Source: [Oracle Server Concepts](#)

84

---

---

---

---

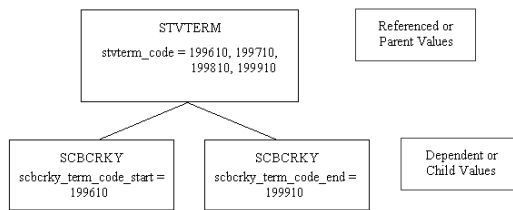
---

---

---

---

### Referential Integrity Illustrated



85

---

---

---

---

---

---

---

---

### Referential Integrity

- A rule defined on a column (or set of columns) in one table that allows the insert or update of a row only if the value for the column or set of columns (the dependent or child value) matches a value in a column of a related table (the referenced or parent value).\*

\* [Oracle Server Concepts](#)

86

---

---

---

---

---

---

---

---

### Referential Integrity (cont.)

- Relies on two types of keys
  - Primary Keys
  - Foreign Keys
- Keys implemented as *constraints*

67

---

---

---

---

---

---

---

---

### Primary Key Constraints

- Enforce *Unique, Non-null* keys
- Banner Naming Convention:  
PK\_ + primary key table name
- PK\_STVTERM is defined by:

```
Alter table STVTERM  
add constraint PK_STVTERM  
Primary key (stvterm_code)
```

68

---

---

---

---

---

---

---

---

### Primary Key Constraints (cont.)

- To see "alter" statements that create primary key constraints for a table (ex: SCBCRKY):  
In SQL\*PLUS, run GURRDDL script for SCBCRKY

```
Alter table SCBCRKY  
add constraint PK_SCBCRKY  
Primary key (scbcrky_subj_code, scbcrky_crse_num)
```

69

---

---

---

---

---

---

---

---

## Foreign Key Constraints

- **Foreign Key:** The column or set of columns included in the definition of the referential integrity constraint that reference a *referenced key*\*.
- **Referenced Key:** The unique or primary key of the same or different table that is referenced by a foreign key\*.

\* Oracle Server Concepts

70

---

---

---

---

---

---

---

---

## Foreign Key Constraints (cont.)

### • Naming Convention:

- FKn\_ + foreign table\_ + INV\_ + Primary table\_ + CODE
  - WHERE
    - "n" is a one-up number
    - foreign table is the table that contains the constraint
    - primary table is the table which contains the primary or referenced key
- Example: FK1\_SCBCKRY\_INV\_STVTERM\_CODE
- Note: The underscore character ( \_ ) separates each element of the name.

71

---

---

---

---

---

---

---

---

## Creating Foreign Key Constraints

### • Alter Statement

```
FKn_ + foreign table_ + INV_ + Primary table_ + CODE
FK1_SCBCKRY_INV_STVTERM_CODE
Alter table SCBCKRY
add constraint FK1_SCBCKRY_INV_STVTERM_CODE
foreign key (SCBCKRY_TERM_CODE_START)
references SATURN.STVTERM (STVTERM_CODE);
```

- **Before you can enter a term code in SCBCKRY it must exist in STVTERM**
- **STVTERM codes cannot be deleted if they exist in other tables**

72

---

---

---

---

---

---

---

---



### Foreign Key Constraints (cont.)

```
SQL> SELECT constraint_name,  
           constraint_type,  
           status  
       FROM all_constraints  
       WHERE table_name = 'SCBCRSE';
```

73

---

---

---

---

---

---

---

---

### Validation Tables/Codes

POSITIONS            9th            14th  
S C B C R S E \_ S U B J \_ C O D E

TABLE NAME    Validation  
              Description  
              |  
              V  
              STV SUBJ

74

---

---

---

---

---

---

---

---

### Referential Integrity: Summary

- Enforces unique, not-null columns
- Establishes relationship between parent and child tables
- Parent table row has the Primary Key constraint
- Child table row has the Foreign Key constraint
- Parent row can not be deleted when a child row exists
- Deletion Sequence: Child, Parent

75

---

---

---

---

---

---

---

---



## Banner Student Technical General Person

---

---

---

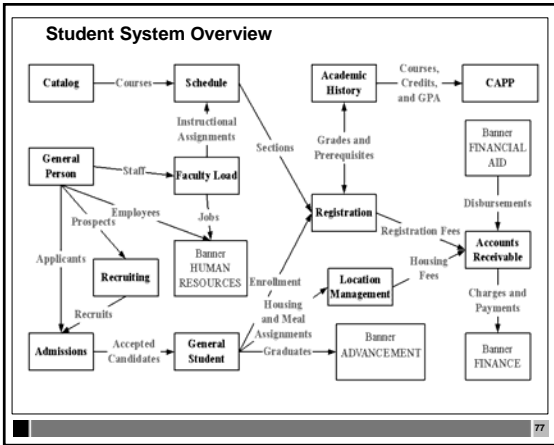
---

---

---

---

---




---

---

---

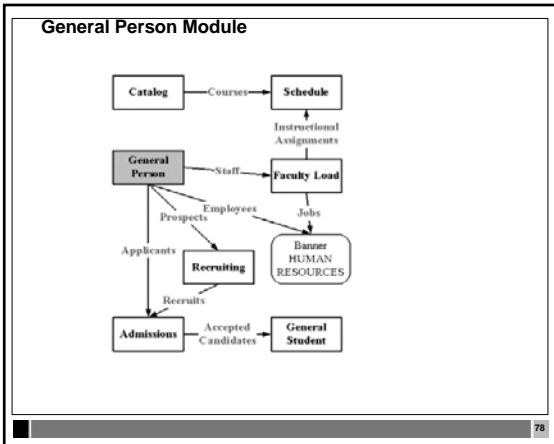
---

---

---

---

---




---

---

---

---

---

---

---

---

**General Person Module:  
Objectives**

- **Examine**
  - Major and Required Forms and Tables
  - SOBSEQN, Oracle Sequences and PIDM
  - Data standards
  - SPRIDEN, SPBPERS indicators
  - SPRDIR process
  - Conversion of data

---

---

---

---

---

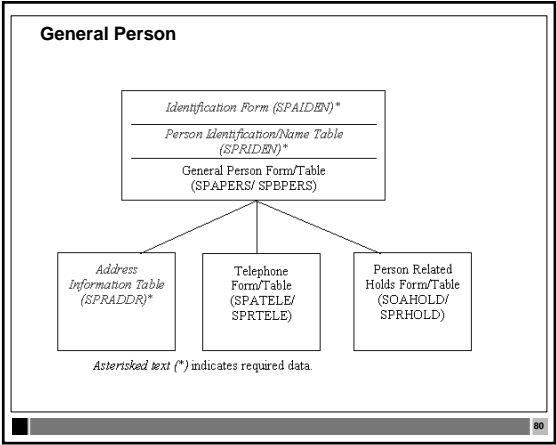
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

**General Person (cont.)**

- **Major Forms:**
  - SPAIDEN
  - SPAPERS
  - SPATELE
  - SOAHOLD

---

---

---

---

---

---

---

---

---

---

**General Person** (cont.)

- **Major Tables:**
  - SPRIDEN
  - SPBPERS
  - SPRADDR
  - SPRTELE
  - SPRHOLD

82

---

---

---

---

---

---

---

---

**General Person** (cont.)

- **Major Validation Tables/Forms:**
  - STVATYP
  - STVSTAT
  - STVNATN
  - GTVZIPC
  - STVTELE
  - STVHLDD

83

---

---

---

---

---

---

---

---

**PIDM**

- **What is PIDM?**
  - Person Identification Master
  - Internal key field used to identify and store records
  - Data type: number
- An Oracle Index is used to generate one-up numbers for PIDM creation.
- gp\_common API
- pidm\_sequence
- f\_generate\_pidm

84

---

---

---

---

---

---

---

---

### Generated ID numbers

#### IDs – Manual, Generated, Previous

- Manual IDs entered by Users
- Generated IDs use Oracle sequence
  - id\_sequence
  - f\_generate\_id
  - Prefix for Generated stored in SOBSEQN
- Previous IDs migrated or entered by Users
  - Spriden\_change\_ind = 'N' or 'I'

85

---

---

---

---

---

---

---

---

### Generated IDs

- id\_sequence
- f\_generate\_id
- Select gb\_common.f\_generate\_id() from dual;
- Select id\_sequence.nextval from dual;
- Prefix pulled from SOBSEQN
- A00000001 = Generated ID

86

---

---

---

---

---

---

---

---

### Contents of SOBSEQN

```
select * from sobseqn;
```

SOBSEQN_FUNCTION	S	SOBSEQN_MAXSEQNO	SOBSEQN_A
RECEIPT		210	15-JUN-98
ID	A	47	24-JUN-98
PIDM		559	26-JUN-98
ALUMNIGIFT		43	16-JUN-98
ALUMNIPLEGE		23	07-JUN-98
EDIREQUESTID		1	25-APR-95
EDI_DCHT_SEQNO		1	08-DEC-95
ALUMNIDUES		3	06-MAY-97
ALUMNIRECEIPT		1	31-JAN-96
EVENT	A	4	18-JUN-98
HRREQ	R	0	31-JAN-96

87

---

---

---

---

---

---

---

---

### ID and SOBSEQN

- **sobseqn\_seqno\_prefix**
  - The column `sobseqn_seqno_prefix` allows the client to determine the character which will precede a generated ID
  - For example, a `sobseqn_seqno_prefix` set to "A" precedes the generated ID: A0000001

88

---

---

---

---

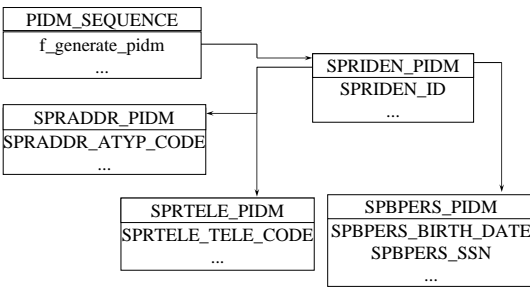
---

---

---

---

### PIDM and ID Relationships



89

---

---

---

---

---

---

---

---

### Data Standards

- Names, addresses, and special characters
- GUAINST
- Date Format
- Century Pivot

90

---

---

---

---

---

---

---

---

**Names**

- Omit spaces within prefixed last names:  

MacArthur	O'Connor	VanHusen
St.John	deBolt	DuShen
- Omit spaces within hyphenated last names:  

Smith-Jones	Cochram-Ashley
-------------	----------------
- Use conventional mixed-case format
- Use periods after prefixes and suffixes where applicable:  

Miss	Mrs.	Mr.	Jr.	III
------	------	-----	-----	-----

EXAMPLE:

Prefix	Firstname	Hyphenated Last name
Mrs.	Joann	Robinson-O'Connor

91

---

---

---

---

---

---

---

---

---

---

**Special Characters**

- Avoid use of the pound sign (#)  
 — Banner Letter Generation identifies a pound sign as a formatting command
- Avoid the use of the following special characters (see Student Technical Reference Manual)  
 - / \* + # & @ \$
- This will increase the efficiency of Banner and any other database accessing tool and minimize confusion for users

92

---

---

---

---

---

---

---

---

---

---

**Dates**

- GUAINST uses radio buttons
- Date Formats (MDY, DMY, YMD)
  - MDY January 5, 1995 is entered as 01/05/95
  - DMY January 5, 1995 is entered as 05/01/95
  - YMD January 5, 1995 is entered as 95/01/05
- Notes
  - If you enter only part of the date, the rest of the current date defaults
  - If you are including a date in query criteria, always include the century
  - You can enter a dash (-) instead of a slash (/)
- Job Submission uses DD-MON-YYYY format in GJAPCTL
- Accounts Receivable uses DD-MON-YY format

93

---

---

---

---

---

---

---

---

---

---

### Century

- The setting of the Century Pivot field in GUAINST indicates the years before which the century defaults to 20 as opposed to 19. For example, if Century Pivot is set to 50 and the Date Format record group is set to MDYY, then dates convert in this manner:
  - 1-5-19 Converts to 05-JAN-2019
  - 1-5-20 Converts to 05-JAN-2020
  - 1-5-27 Converts to 05-JAN-2027
  - 1-5-28 Converts to 05-JAN-2028
  - 1-5-78 Converts to 05-JAN-1978
  - 1-5-92 Converts to 05-JAN-1992
- If you are querying information and part of the query is a year, you need to enter the century and the year to insure accuracy in your selections

94

---

---

---

---

---

---

---

---

### General Person: Step 1

- **SPAIDEN**
  - Add a person to form
  - Generate ID
  - Enter Name Information (including suffix or prefix)
  - Add Address Information
- **Save this information**
- **Rollback to the key block**

95

---

---

---

---

---

---

---

---

### General Person: Step 2

- **Enter SPAIDEN again (Next Block)**
  - Change the ID
    - Save
  - Change the middle name
    - Save
  - Add another address (different type)
    - Save
- **Rollback to the key block**

96

---

---

---

---

---

---

---

---



**General Person: Step 3**

- Use the LOV field to access SOAIDEN
  - Do a query to find the person you just entered
  - Notice the change indicators (I,N)
- Exit SPAIDEN

97

---

---

---

---

---

---

---

---

**General Person: Step 4**

- SPAPERS
  - Enter SSN (SIN in Canada)
  - Enter Birth Date
  - Enter Confidentiality Indicator
- Save
- Exit SPAPERS

98

---

---

---

---

---

---

---

---

**General Person: Step 5**

- SOAHOLD
- Use LOV field to see list of holds
- Place two different types of holds on your record
- Save

99

---

---

---

---

---

---

---

---

**General Person: Step 6**

- Describe SPRIDEN
- Write a query to retrieve the data that was entered in SPRIDEN today  
— where spriden\_activity\_date like sysdate

---

---

---

---

---

---

---

---

**General Person: Step 7**

- Notice the data in:  
— spriden\_change\_ind  
— spriden\_search\_last\_name  
— spriden\_soundex\_last\_name  
— spriden\_entity\_ind  
— spriden\_pidm

---

---

---

---

---

---

---

---

**General Person: Step 8**

- Describe SPBPERS
- Write a query to retrieve the data that you entered in SPBPERS  
— where spbpers\_activity\_date like sysdate

---

---

---

---

---

---

---

---

**General Person: Step 9**

- Notice the data in:
  - spbpers\_prefix
  - spbpers\_suffix
  - spbpers\_ssn
  - spbpers\_confidential\_ind
  - spbpers\_activity\_date

---

---

---

---

---

---

---

---

**General Person: Step 10**

- Describe SPRADDR
- Write a query to retrieve the data that you entered in SPRADDR
- Notice:
  - spraddr\_atyp\_code
  - spraddr\_seqno
  - spraddr\_from\_date
  - spraddr\_to\_date

---

---

---

---

---

---

---

---

**General Person: Step 11**

- Describe SPRHOLD
- Write a query to retrieve the data that you entered in SPRHOLD
- Notice:
  - sprhold\_hidd\_code
  - sprhold\_user
  - sprhold\_from\_date
  - sprhold\_to\_date

---

---

---

---

---

---

---

---

**General Person: SPRPDIR**

- **SPRPDIR -- Person Directory**
  - Produces a list of persons, addresses, and primary phone numbers
  - By type of person:
    - Recruit (R)
    - Applicant (A)
    - Student (S)
    - Faculty (F)

---

---

---

---

---

---

---

---

**General Person: SPRPDIR (cont.)**

- **Tables used in SPRPDIR.pc:**
  - SPBPERS - General Person Info Table
  - SRBRECR - Recruit Information Table
  - SARADAP - Applicant Information Table
  - SGBSTDN - Student Information Table
  - SIBINST - Faculty Information Table
  - SPRCOLR - Address Collector File
  - SPRTELE - Telephone Number Table

---

---

---

---

---

---

---

---

**General Person: SPRPDIR (cont.)**

- **View used in SPRPDIR.pc:**
  - SPVADDs - Address View
- **Parameters:**
  - Term, Type, Confidentiality Indicator
  - Address Type, Print ID, Faculty type (A,I,B)
  - Population Selection Can Be Used
- **C program**
- **Run via Job Submission**

---

---

---

---

---

---

---

---

**General Person: Conversion Issues**

- **What additional general person data do you have in your legacy system?**
- **How do you determine where to put it in Banner?**
  - **Become familiar with all General Person forms and tables**
    - `select table_name, comments`  
  `from all_tab_comments`  
  `where table_name like 'SP%';`
- **Consult users about where to put data**

109

---

---

---

---

---

---

---

---

**General Person Module: Other Scripts**

- **\$BANNER\_HOME/general/views**
  - **views (gpv\*)**
  - **ag\_entity\_data: Object:Access view which presents general person data (gpvent0.sql)**
  - **Object:Access views used in conjunction with Object:Access method of retrieving data from database**
    - uses concept of "layered" views
    - must have GTVSDAX form/table populated with crosswalk values

110

---

---

---

---

---

---

---

---

**General Person Summary**

- **Major Forms:**
  - SPAIDEN
  - SPAPERS
  - SOAHOLD
  - SOAIDEN
- **Major Tables**
  - SPRIDEN
  - SPBPERS

111

---

---

---

---

---

---

---

---

**General Person Summary** (cont.)

- Major Tables
  - SPRADDR
  - SPRTELE
  - SPRHOLD
- Report: SPRPDIR
- PIDM, SOBSEQN
- Data Standards

112

---

---

---

---

---

---

---

---

**General Person: Exercise 1**

- Write a query to return the pidm, id, first name, middle name, last name, and change indicator for persons who have had changes made to their SPRIDEN records today.

113

---

---

---

---

---

---

---

---

**General Person: Exercise 2**

- Write a query to return the id, first name, last name, and change indicator for the record that you entered about yourself in the database today.
  - There should be an ID change indicator ('I'), a name change indicator ('N') and a record in which the change indicator IS NULL.
  - Have the query prompt you for the pidm.

114

---

---

---

---

---

---

---

---

**General Person: Exercise 3**

- Write a query to select the pidm, id, first name, last name, change indicator, social security number (from SPBPERS) where changes were made to the ID records in SPRIDEN.

---

---

---

---

---

---

---

---

**General Person: Exercise 4**

- Write a query to extract information that you would use on a mailing label.
  - For this query, select the address type that appears the maximum number of times in the SPRADDR table.
  - You should extract the most current record from the SPRIDEN table.
  - For purposes of simplicity, assume that all SPRADDR records for this address type are current.

---

---

---

---

---

---

---

---



**Banner Student Technical Curriculum/Program Rules**

---

---

---

---

---

---

---

---

**Curriculum/Program Rules: Objectives**

- **Examine**
  - Forms used to build rules
  - Table relationships

---

---

---

---

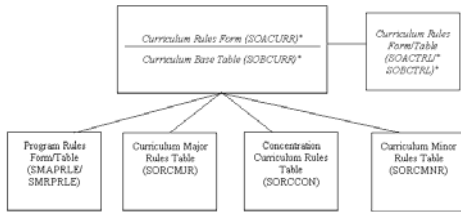
---

---

---

---

**Curriculum/Program Rules**



---

---

---

---

---

---

---

---

**Curriculum/Program Rules (cont.)**

- **Major Forms:**
  - SMARPLE
  - SOACURR
  - SOACTRL

---

---

---

---

---

---

---

---



**Curriculum/Program Rules** (cont.)

- **Major Tables:**
  - SMRPRLE
  - SOBCURR
  - SOBCTRL
  - SORCMJR
  - SORCMNR
  - SORCCON

121

---

---

---

---

---

---

---

---

**Curriculum/Program Rules** (cont.)

- **Major Validation Tables:**
  - STVTERM
  - STVLEVL
  - STVCOLL
  - STVDEGC
  - STVCAMP
  - STVDEPT
  - STVMAJR

122

---

---

---

---

---

---

---

---

**Program Definition Rules**

- **SMAPRLE**
  - Program Definition Rules Form
  - SMRPRLE is underlying table
  - Program is required only if CAPP's Program Planning indicator is set to 'Yes' in SOACTRL
    - sobctrl\_program\_ind = 'Y'

123

---

---

---

---

---

---

---

---

**Program Definition Rules** (cont.)

- **SMAPRLE/SMRPRLE**
  - When program is used on a curriculum rule, the following must match on SOACURR what is defined in SMAPRLE:
    - Level
    - Campus
    - College
    - Degree

124

---

---

---

---

---

---

---

---

**Curriculum Rules**

- **SOACURR**
  - Curriculum Rules Form
  - SOBCURR is underlying table
  - Used to view or create curriculum rules
  - Rules are based on Program Definitions if you are using program rules; otherwise, program is not a required field

125

---

---

---

---

---

---

---

---

**Curriculum Rules Control**

- **SOACTRL**
  - Curriculum Rules Control Form
  - SOBCTRL is underlying table
  - Indicators determine if/how various areas related to curriculum are used
    - Can set Use CAPP's Program Planning to 'Y' or 'N'
  - Indicators set severity level of error checking by module if curriculum rules are used

126

---

---

---

---

---

---

---

---

**Major, Minor, Concentration**

- SORCMJR - Curriculum Major Rules
- SORCMNR - Curriculum Minor Rules
- SORCCON - Concentration Curr. Rules
- Each table contains on/off indicators for each module using curriculum rules
  - e.g. Admissions: sorcmjr\_adm\_ind = 'Y'
- The data from these tables shows up through SOACURR.

---

---

---

---

---

---

---

---

**Curriculum/Program Rules:  
Summary**

- Build rules in SOACURR
  - All curriculum rules must be built before setting indicators in SOACTRL
- Build Program Rules on SMAPRLE (if you plan to use CAPP's Program Planning)
- Build control rules in SOACTRL
  - if sobctrl\_curr\_rule\_ind = 'Y', then sobctrl\_program\_ind must = 'Y'
    - Note: This means that you are using CAPP's Program Planning

---

---

---

---

---

---

---

---

**Curriculum/Program Rules:  
Conversion Issues**

- Will your users build curriculum rules?
- If so, then can you use the rules to your advantage when converting student data?
  - Can you use the student's major (on legacy side) to get the valid department and program codes from SOBCURR and SORCMJR?

---

---

---

---

---

---

---

---

**Curriculum/Program Rules: Exercise**

- Write a query to retrieve curriculum rules that apply to STUDENT, listing:
  - major and program
  - department code
  - level code
  - college code
  - campus code
  - degree code

---

---

---

---

---

---

---

---



**Banner Student Technical  
Concurrent Curricula**

**Concurrent Curricula: Objectives**

- Examine
  - Define Concurrent Curricula
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**Concurrent Curricula**

- **What is Concurrent Curricula?**
  - Allows an institution to record and use multiple curricula for a person as they move through the student cycle
- **Key terms**
  - Recruit
  - Admission
  - Learner/General Student
  - Outcome/Academic History

---

---

---

---

---

---

---

---

**Concurrent Curricula (cont.)**

- **Major Form: SOILCUR**
- **Major Tables:**
  - SORLCUR
  - SORLFOS
  - SOBCACT
  - SOBLMOD
  - Tables in Recruiting, Admissions, General Student (Learner) and Academic History (Outcome)

---

---

---

---

---

---

---

---

**Concurrent Curricula (cont.)**

- **Major Validation Tables/Forms:**
  - GTVLFST
  - STVCACT
  - STVCSTS
  - STVLMOD

---

---

---

---

---

---

---

---

**Concurrent Curricula** (cont.)

- **Processes:**
  - SOPLCV – for those schools upgrading to Banner 7

---

---

---

---

---

---

---

---

**Concurrent Curricula: SQL\*Plus**

- **What data elements are required?**
  - DESC SORLCUR
  - DESC SORLFOS
  - DESC SOBCACT
  - DESC SOBLCUR

---

---

---

---

---

---

---

---

**Concurrent Curricula: Other Scripts**

- The package SOKLCUR is used to query curriculum data
- P\_backload\_curr updates the primary and secondary curricula on any student records
  - This is necessary until final implementation of the enhancement to allow current reports, process and procedures continue to work as designed

---

---

---

---

---

---

---

---

**Concurrent Curricula: Conversion Issues**

- Will data be loaded to both concurrent curricula tables and existing table structures as they were in Banner 6.x?
- Or will data be loaded only to those existing tables and use the SOPLCUR process to create SORLCUR and SORLFOS table data?
- What is backfilling and why do you need to know about it?

139

---

---

---

---

---

---

---

---

**Concurrent Curricula: Exercise 1**

- Write a query that will return one student's current curriculum

140

---

---

---

---

---

---

---

---



BANNER HIGHER EDUCATION

**Banner Student Technical  
Recruiting**

---

---

---

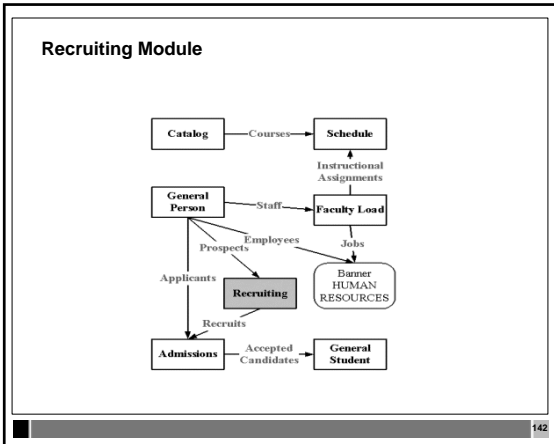
---

---

---

---

---




---

---

---

---

---

---

---

---

- ### Recruiting Module: Objectives
- Examine
    - Major and Required Forms and Tables
    - Reports, Processes and Procedures
    - Conversion of Data
- 143

---

---

---

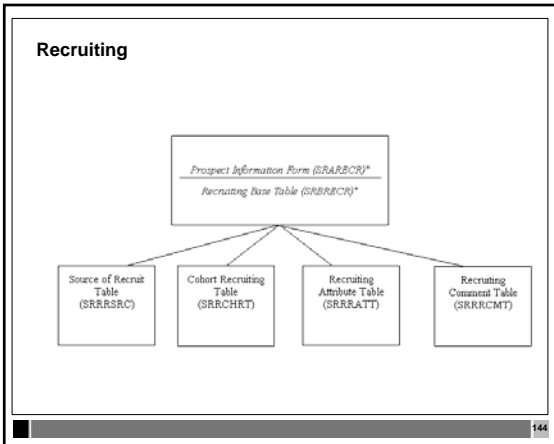
---

---

---

---

---




---

---

---

---

---

---

---

---



**Recruiting** (cont.)

- Major Form: SRARECR
- Major Tables:
  - SRBRECR
  - SRRRSRC
  - SRRCHRT
  - SRRRATT
  - SRRRCMT
  - Curriculum Rules Tables

145

---

---

---

---

---

---

---

---

**Recruiting** (cont.)

- Major Validation Tables/Forms:
  - STVTERM
  - STVLEVL
  - STVCOLL
  - STVDEGC
  - STVMAJR
  - STVSTYP
  - STVRESO

146

---

---

---

---

---

---

---

---

**SRARECR**

- Prospect Information Form
  - Provides information necessary for all recruitment related activities
  - The basis for all related recruiting forms
  - Can go to SPAIDEN form to create a person record from this form
  - Notice connections to Curriculum

147

---

---

---

---

---

---

---

---

### SRAQUIK

- Quick Recruit Form
  - Allows entry of new prospective students
  - General Person information is created via this form (populating tables: SPRIDEN, SPBPER, SPRADDR, etc.)
  - Other information is created via this form (populating tables: SORHSCH, SORPCOL, SORINTS, SRRRSRC, SORCONT, etc.)

148

---

---

---

---

---

---

---

---

### Recruiting: SQL\*Plus

- What tables are part of Recruiting Module?

```
-- select table_name  
from all_tables  
where table_name like 'SR%'
```

- What data elements are required?

```
-- desc srbrecr  
-- notice the "NOT NULL" columns
```

- What are the key fields in srbrecr?

```
SQL> select column_name from all_cons_columns  
2 where table_name = 'SRBRECR'  
3 and constraint_name = 'FK_SRBRECR';
```

149

---

---

---

---

---

---

---

---

### Recruiting Module: Reports

- SRRENRL - Recruiting Enrollment Analysis
- SRRINQR - Recruits Never Applied to Institution Report

150

---

---

---

---

---

---

---

---

**Recruiting Module: Other Scripts**

- **\$BANNER\_HOME/student/dbprocs**
  - functions (srf\*)
- **\$BANNER\_HOME/student/views**
  - views (srv\*): srvrecr0.sql creates view called as\_recruiting\_data
  - some views used in conjunction with Object:Access method of retrieving data from database; uses concept of "layered" views; must have GTVSDAX form/table populated with crosswalk values

---

---

---

---

---

---

---

---

**Recruiting:  
Conversion Issues**

- Will Recruiting data be converted or entered manually by the users?
- What Recruiting data do you have in your legacy system?
- How do you determine where to put it in Banner?
- Will you use curriculum rules?

---

---

---

---

---

---

---

---

**Recruiting Module: Summary**

- Major Forms
  - SRARECR
  - SRAQUIK
- Major Tables
  - SRBRECR
  - Validation Tables

---

---

---

---

---

---

---

---



# Banner Student Technical Admissions

---

---

---

---

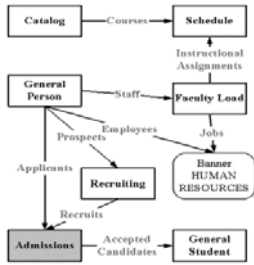
---

---

---

---

## Admissions Module



---

---

---

---

---

---

---

---

## Admissions: Objectives

- **Examine/Review**
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

---

---

---

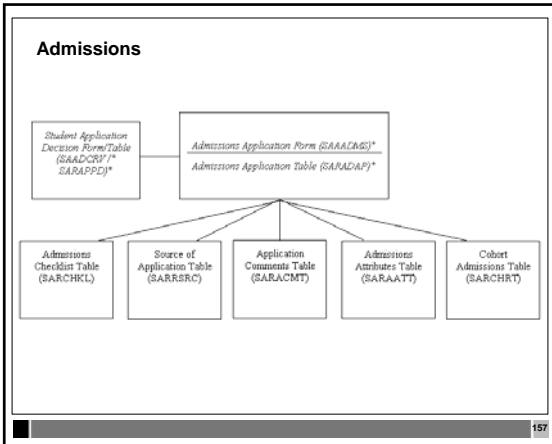
---

---

---

---

---




---



---



---



---



---



---



---



---

- Admissions** (cont.)
- **Major Validation Tables/Forms:**
    - STVTERM
    - STVLEVL
    - STVCOLL
    - STVDEGC
    - STVMAJR
    - STVSTYP
    - STVRESO
- 158

---



---



---



---



---



---



---



---

- Admissions** (cont.)
- **Validation Tables/Forms (cont'd):**
    - STVAPST
    - STVAPDC
    - STVTESE
    - STVDLEV
    - STVACAT
    - STVSTAT
    - GTVLETR
    - GTVPARA
- 159

---



---



---



---



---



---



---



---

**SAAADMS**

- **Admissions Application Form**
  - Used for maintaining applications submitted to the institution
  - Can maintain an unlimited number of applications for any given term (saradap\_term\_code\_entry, saradap\_appl\_no)

---

---

---

---

---

---

---

---

**SAAQUIK:  
Quick Admit Form**

- Allows entry and registration of new students with minimal effort
- General Person information is created via this form (populating tables: SPRIDEN, SPBPERS, SPRADDR, SPRTELE, etc.)
- Admissions and/or Recruitment records may be created through this form
- Other information can be accessed via this form (tables: SORHSCH, SORPCOL, SPRHOLD, SORTEST, SPRINTL, etc.)

---

---

---

---

---

---

---

---

**SAADCRV/SARAPPD**

- SAADCRV – Admissions Decision Form
- Underlying table is SARAPPD
- Once an applicant is accepted through SAADCRV, a student record is created (SGASTDN form/SGBSTDN table)

---

---

---

---

---

---

---

---

**Admissions: SQL\*Plus**

- **What tables are part of the Admissions Module?**

```
-- select table_name  
from all_tables  
where table_name like 'SA%'
```

- **What data elements are required?**

```
-- desc saradap  
-- notice the "NOT NULL" columns
```

- **What are the key fields in saradap?**

```
select column_name  
from all_cons_columns  
where table_name = 'SARADAP'  
and constraint_name = 'FK_SARADAP';
```

---

---

---

---

---

---

---

---

**Admissions Module: Reports**

- **SARACTM - Admissions Count by College/Major**
- **SARADMS - Admissions Application**
- **SARDCSN - Admission Decision Criteria**
- **Other reports and purge processes are also available for admissions module**

---

---

---

---

---

---

---

---

**SARACTM**

- **Prints admission application count by college/major**
- **C program run from job submission**

---

---

---

---

---

---

---

---

**Admissions Module:  
Other Scripts**

- **\$BANNER\_HOME/student/dbprocs**
  - functions (saf\*)
- **\$BANNER\_HOME/student/views**
  - views (sav\*): savadm0.sql creates as\_admissions\_applicant
  - some views used in conjunction with Object:Access method of retrieving data from database; uses concept of "layered" views; must have GTVSDAX form/table populated with crosswalk values

---

---

---

---

---

---

---

---

**Admissions: Conversion Issues**

- Will Admissions data be converted or entered manually by the users?
- What Admissions data do you have in your legacy system?

---

---

---

---

---

---

---

---

**Admissions Module: Summary**

- Major Forms
  - SAAADMS
  - SAAQUIK
  - SAADCRV
- Major Tables
  - SARADAP
  - SARAPPD
  - VALIDATION TABLES

---

---

---

---

---

---

---

---



**Admissions: Exercise 1**

- Write a query to get the id, last name, term of entry and student type for applicants for a specific future term (prompt user for term code).

The records returned should be for the most current application for that term and the decision should be the most recent decision made that matches that application.

---

---

---

---

---

---

---

---

---

---



**Banner Student Technical  
Overall Forms and Tables**

**Overall Forms and Tables: Objectives**

- Examine/Review
  - Major and Required Forms and Tables
  - Conversion of Data
  - Reports and Processes

---

---

---

---

---

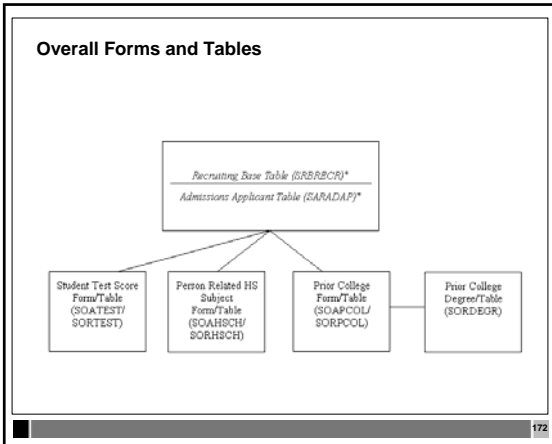
---

---

---

---

---




---

---

---

---

---

---

---

---

- Overall Forms and Tables (cont.)**
- **Major Forms:**
    - SOATEST
    - SOAHSCH
    - SOAPCOL
  - **Major Tables:**
    - SORTEST
    - SORHSCH
    - SORPCOL
    - SORDEGR
- 173

---

---

---

---

---

---

---

---

- Overall Forms and Tables (cont.)**
- **Major Validation Tables/Forms:**
    - STVTEC
    - STVSBGI
    - STVDEGC
- 174

---

---

---

---

---

---

---

---

**Overall: Major Forms**

- **SOATEST**  
—Used to maintain test score information
- **SOAHSCH**  
—Used with Recruiting and Admissions for high school information
- **SOAPCOL**  
—Used with Recruiting, Admissions, and Faculty Load for prior college information

175

---

---

---

---

---

---

---

---

**Overall: SQL\*Plus**

- **What are the tables are that are used for multiple modules?**  

```
select table_name, table_type, comments
  from all_tab_comments
 where table_name like 'SO%'
    and table_type = 'TABLE';
```
- **Are any data elements required in SORTEST, SORHSCH, SORPCOL?**

176

---

---

---

---

---

---

---

---

**Overall: Conversion Issues**

- **Will Overall data be converted or entered manually by the users?**
- **What Overall data do you have in your legacy system?**
- **How do you determine where to put it in Banner?**

177

---

---

---

---

---

---

---

---

**Overall Reports**

- Reports/Processes
  - SRTLOAD - Interface Tape Load Process
    - SOAINFR - Tape Interface Rules Form
    - SOTCNVT - Tape Code Conversion Form
    - SRIPREL - View Suspended Records

178

---

---

---

---

---

---

---

---

**Overall: Summary**

- Overall Forms and Tables are used for multiple modules
- Major Forms/Tables:
  - SOATEST/SORTEST
  - SOAHSCH/SORHSCH
  - SOAPCOL/SORPCOL
- SORTAPE process

179

---

---

---

---

---

---

---

---

**Overall: Exercise**

- Get the following information about all applicants for a term (prompt for term):
  - Full Name, Entry Term, Test Code, Test Score, and High School GPA for Students who took either the ACT English or the SAT Verbal tests
- Your query should return only records with values in all the above areas.

180

---

---

---

---

---

---

---

---



## Banner Student Technical Faculty Load

---

---

---

---

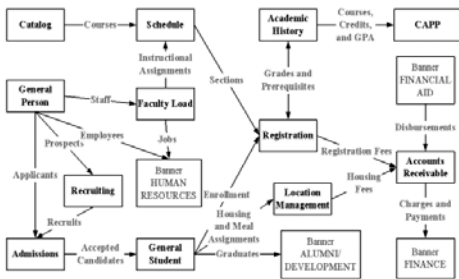
---

---

---

---

### Student System Overview




---

---

---

---

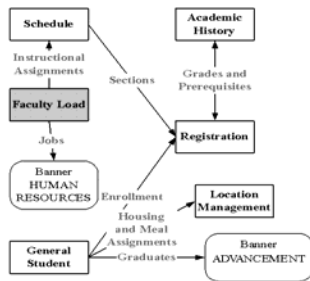
---

---

---

---

### Faculty Load Module




---

---

---

---

---

---

---

---

**Faculty Load Module: Objectives**

- **Examine/Review**
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

---

---

---

---

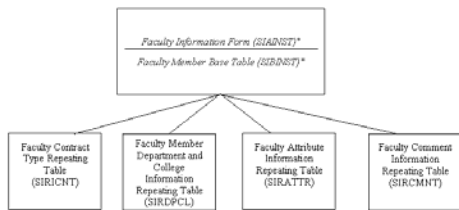
---

---

---

---

**Faculty Load**



---

---

---

---

---

---

---

---

**Faculty Load (cont.)**

- **Major Validation Tables/Forms :**
  - STVFCST\*
  - STVFCFG
  - STVFSTP
  - STVFCNT
  - STVWKLD
  - STVCNTR

\*Required

---

---

---

---

---

---

---

---

**Faculty Load** (cont.)

- Major Form/Table:
  - SIAINST/SIBINST
    - Used to maintain Faculty Information
  - Codes and Indicators for:
    - Active/Inactive (sibinst\_fcst\_code)
    - Instructor (sibinst\_schd\_ind)
    - Advisor (sibinst\_advr\_ind)

187

---

---

---

---

---

---

---

---

**Faculty Load:  
Additional Information**

- SIAINST form also utilizes these tables (but data is not required in SIBINST):
  - SIRICNT - Contract Info
  - SIRDPCL - Department/College Info
  - SIRATTR - Attributes
  - SIRCMTT - Comments
- SORDEGR table is used in SIAFDEG form to maintain faculty degree information

188

---

---

---

---

---

---

---

---

**Faculty Load** (cont.)

- SIAASGN/SIRASGN
  - This form/table contains faculty teaching assignments for a particular term
- Populated automatically when a faculty member is entered on the SSASECT form in the schedule module IF records exist in SIBINST (faculty status)

189

---

---

---

---

---

---

---

---

**Faculty Load: SQL\*Plus**

- **What tables are part of the Faculty Load Module?**

```
— select table_name  
  from all_tables  
 where table_name like 'S%'
```

- **What data elements are required?**

```
— desc sibinst  
— notice the "NOT NULL" columns
```

- **What are the key fields in sibinst?**

```
select column_name  
  from all_cons_columns  
 where table_name = 'SIBINST'  
       and constraint_name = 'PK_SIBINST';
```

---

---

---

---

---

---

---

---

**Faculty Load:  
Reports and Processes**

- SIPASGN -- Faculty Load Purge
- SIRASGQ -- Faculty Schedule Report
- SIRCTAL -- Faculty Load Contract Analysis Report
- SIRTRAL -- Faculty Load Term Analysis Report

---

---

---

---

---

---

---

---

**Faculty Load Module: Other Scripts**

- \$BANNER\_HOME/student/dbprocs  
 — functions (sif\*)
- \$BANNER\_HOME/student/views  
 — views (siv\*)  
 — some views used in conjunction with Object:Access  
 method of retrieving data from database
  - uses concept of "layered" views
  - must have GTVSDAX form/table populated with crosswalk  
 values

---

---

---

---

---

---

---

---



**Faculty Load: Conversion Issues**

- Will Faculty Load data be converted or entered manually by the users?
- What Faculty Load data do you have in your legacy system?
- How do you determine where to put it in Banner?

---

---

---

---

---

---

---

---

**Faculty Load Module: Summary**

- Major Form/Table:
  - SIAINST/SIBINST
- Other Tables:
  - SIRICNT
  - SIRDPCL
  - SIRATTR
  - SIRCMNT
  - VALIDATION TABLES

---

---

---

---

---

---

---

---

**Faculty Load: Exercise**

- Write a query which would return the full name, id, faculty status and effective term for that status for an instructor.

---

---

---

---

---

---

---

---



# Banner Student Technical Location Management

---

---

---

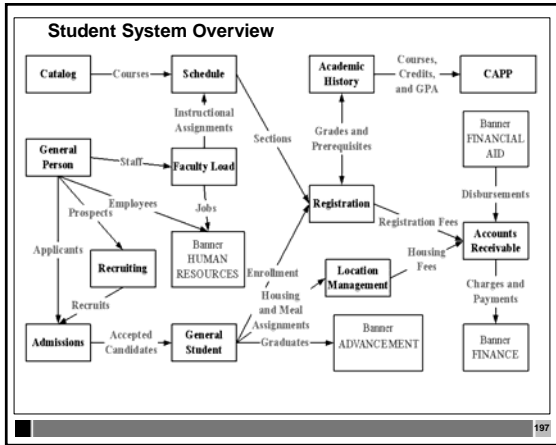
---

---

---

---

---



---

---

---

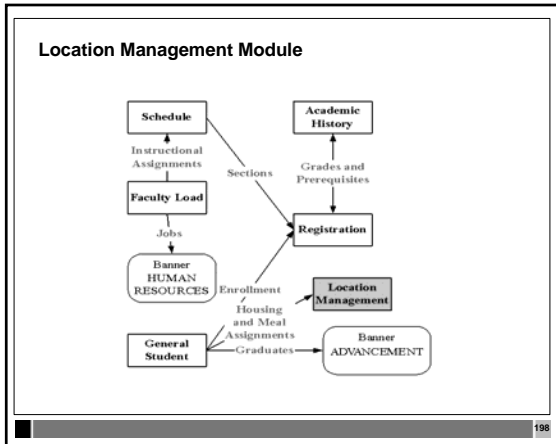
---

---

---

---

---



---

---

---

---

---

---

---

---

### Location Management Module: Objectives

- **Examine/Review**
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

---

---

---

---

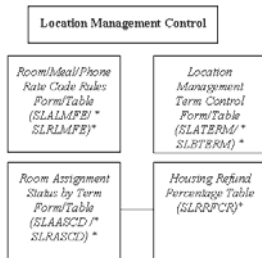
---

---

---

---

### Location Management



---

---

---

---

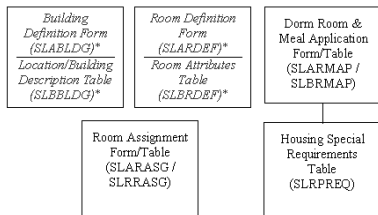
---

---

---

---

### Location Management (cont.)



---

---

---

---

---

---

---

---

### Location Management (cont.)

- **Major Validation Tables/Forms:**
  - STVBLDG STVCAMP
  - STVRRCD STVHAPS
  - STVARTP STVTERM
  - STVASCD TTVDCAT
  - TTVTAXT (for Canadian use only)
- **TBBDETC must also be populated for those making residence hall assignments**

202

---

---

---

---

---

---

---

---

### Major Forms and Tables

- **SLABLDG: Building Definition Form**
  - Table = SLBBLDG
- **SLARDEF: Room Definition Form**
  - Table = SLBRDEF
- **If your institution is residential:**
  - SLARMAP/SLBRMAP:
    - Room Application Form/Table
  - SLARASG/SLRRASG:
    - Room Assignment Form/Table

203

---

---

---

---

---

---

---

---

### Location Management: SQL\*Plus

- **What tables are part of the Location Management Module?**
  - ```
select table_name
  from all_tables
 where table_name like 'SL%'
```
- **What data elements are required?**
  - desc slbrdef
  - notice the "NOT NULL" columns
- **What are the key fields in slbrdef?**
  - ```
select column_name
  from all_cons_columns
 where table_name = 'SLBRDEF'
    and constraint_name = 'PK_SLBRDEF';
```

204

---

---

---

---

---

---

---

---

**Location Management Module:  
Reports and Processes**

- **SLRFASM -- Batch Room, Meal and Phone Assessment**
  - selects records based on slrrasg\_assess\_needed
  - updates slrrasg\_ar\_ind in records that were assessed
- **SLRHLST -- Active Housing Assignments Report**
- **SLRROLL -- Room Assignment Roll Process -- Roll like terms -- fall to fall, etc.**

205

---

---

---

---

---

---

---

---

**Location Management Module:  
Other Scripts**

- **\$BANNER\_HOME/student/dbprocs**
  - functions (sif\*)
- **\$BANNER\_HOME/student/views**
  - views (slv\*): slvres0.sql creates view as\_residential\_life
  - some views used in conjunction with Object:Access method of retrieving data from database; uses concept of "layered" views; must have GTVSDAX form/table populated with crosswalk values

206

---

---

---

---

---

---

---

---

**Location Management:  
Conversion Issues**

- **Will your institution convert or manually enter Location Management information?**
- **What Location Management data do you have in your legacy system?**
- **How do you determine where to put it in Banner?**

207

---

---

---

---

---

---

---

---

**Location Management Module:  
Summary**

- **Major Forms/Tables:**
  - SLABLDG/SLBBLDG, SLARDEF/SLBRDEF
- **For Residential Institutions**
  - SLARMAP/SLBRMAP, SLARASG/SLRRASG
- **Processes/Reports**
  - SLRROLL, SLRFASM, SLRHLST

208

---

---

---

---

---

---

---

---

**Location Management: Exercise**

- **Write a simple report that will show the residence hall assignments for a term (prompt the user for the term). On the report, show last name, id, term, building description and room.**

209

---

---

---


---

---

---

---

---



**Banner Student Technical  
Class Schedule**

---

---

---

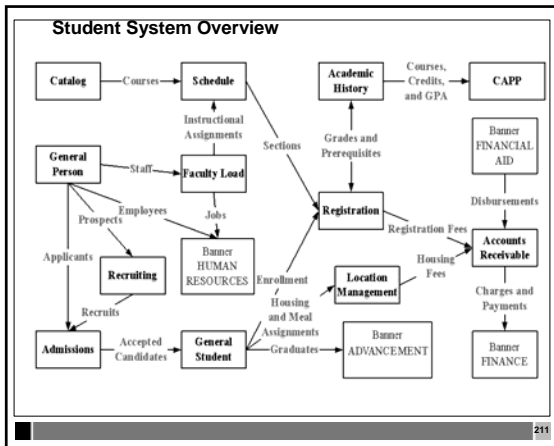
---

---

---

---

---




---

---

---

---

---

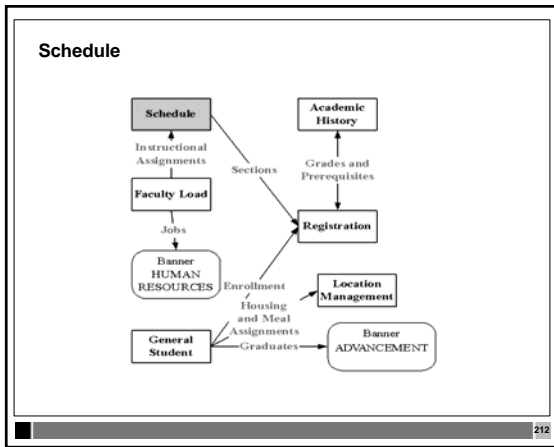
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

- Schedule: Objectives**
- **Examine/Review**
    - Major and Required Forms and Tables
    - Reports, Processes and Procedures
    - Conversion of Data
- 213

---

---

---

---

---

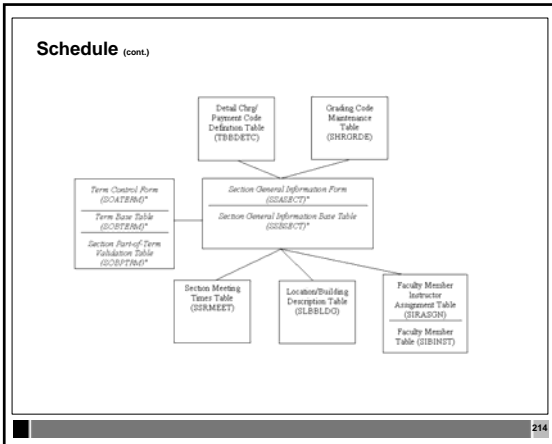
---

---

---

---

---




---

---

---

---

---

---

---

---

**Schedule: SSASECT**

- **SSASECT/SSBSECT**
  - Used to build and maintain schedule of classes
  - Much of the data defaults from Course Catalog (SCBCRSE, etc)
  - Connections with AR module through billing hours, tuition waivers

215

---

---

---

---

---

---

---

---

**Schedule: SOATERM**

- **Major Form/Table:**
  - SOATERM Form
  - SOATERM and SOBPTRM
    - Used to set up controls for each term's schedule, registration, and fee assessment
  - Before building the Schedule for a term, a beginning CRN must be set (CRN Oneup)

216

---

---

---

---

---

---

---

---



**Schedule** (cont.)

- Major Validation Tables/Forms:
  - STVTERM STVLEVL
  - STVPTRM STVCAMP
  - STVSSTS STVSCHD
  - STVSUBJ STVGMOD
  - STVDAYS TBBDETC

217

---

---

---

---

---

---

---

---

**Schedule** (cont.)

- Other Forms/Tables:
  - SSRMEET
    - Section Meeting Times Table
  - SLBBLDG
    - Location/Building Description Table
  - SIAINST/SIBINST
    - Faculty Information Form/Table
  - SIAASGN/SIRASGN
    - Faculty Assignment Form/Table

218

---

---

---

---

---

---

---

---

**SLQMEET and SSAMATX**

- SLQMEET -- Available Classroom Query Form
  - Only accessible through SSASECT
- SSAMATX -- Building/Room Schedule Form
  - Accessible through menu, direct access, other form (SSASECT)

219

---

---

---

---

---

---

---

---

**Schedule: SQL\*Plus**

- What tables are part of the Schedule Module?

```
-- select table_name  
from all_tables  
where table_name like 'SS%'
```

- What data elements are required?

```
-- desc ssbsect  
-- notice the "NOT NULL" columns
```

- What are the key fields in ssbsect?

```
-- select column_name  
from all_cons_columns  
where table_name = 'SSBSECT'  
and constraint_name = 'FK_SSBSECT';
```

---

---

---

---

---

---

---

---

**Schedule: Reports and Processes**

- SSPSCHD -- Schedule Purge Process
- SSRROLL -- Term Roll Process -- Roll like terms -- Fall to Fall
- SSRSECT -- Class Schedule Report
- SSRTALY -- Scheduled Section Tally

---

---

---

---

---

---

---

---

**Schedule: Other Scripts**

- \$BANNER\_HOME/student/dbprocs
  - functions (ssf)
- \$BANNER\_HOME/student/views
  - views (ssv): ssvsec0.sql creates view as\_catalog\_schedule
  - some views used in conjunction with Object:Access method of retrieving data from database
    - uses concept of "layered" views
    - must have GTVSDAX form/table populated with crosswalk values

---

---

---

---

---

---

---

---

**Schedule: Conversion Issues**

- Will Schedule data be converted or entered manually by the users?
  - What Schedule data do you have in your legacy system?
  - How do you determine where to put it in Banner?
- \*It is generally not recommended to convert schedule but to enter it manually.

223

---

---

---

---

---

---

---

---

**Schedule: Summary**

- Major Form: SSASECT
- Major Tables
  - SSBSECT SSRMEET
  - SLBBLDG SIRASGN
  - SIBINST SOBTERM
  - SOBPTRM SHRGRDE
  - SLBRDEF TBBDETC
  - VALIDATION TABLES

224

---

---

---

---

---

---

---

---

**Schedule: Exercise**

- Write a query that returns full name, id, crn, subject code, course number, section number, course title and term code for all faculty members teaching any English course. Prompt the user for the term.

225

---

---

---

---

---

---

---

---



# Banner Student Technical General Student

---

---

---

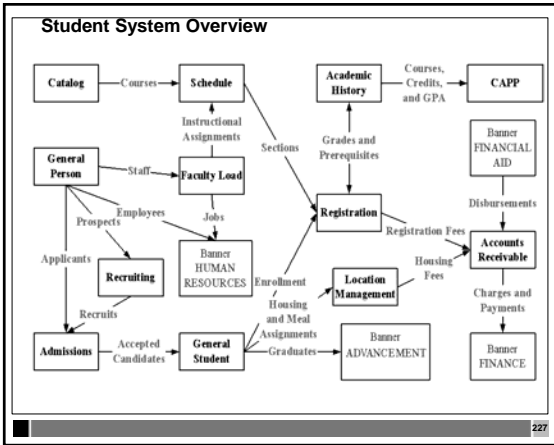
---

---

---

---

---



---

---

---

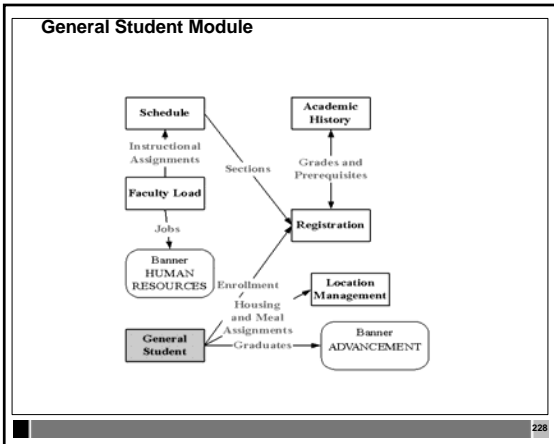
---

---

---

---

---



---

---

---

---

---

---

---

---

**General Student Module: Objectives**

- **Examine/Review**
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

---

---

---

---

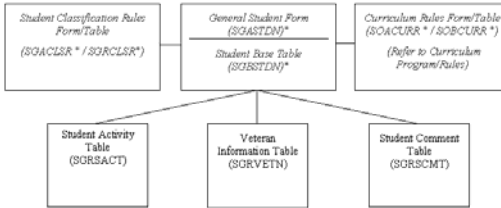
---

---

---

---

**General Student**



---

---

---

---

---

---

---

---

**General Student (cont.)**

- **Major Form/Table:**
  - SGASTDN/SGBSTDN
    - Used to maintain current and historical information about a student
  - SGASTDN form also utilizes the following tables:
    - SGRSACT
    - SGRVETN
    - SGRSCMT

---

---

---

---

---

---

---

---

**General Student** (cont.)

- **Rule Forms/Tables:**
  - SGACLSR/SGRCLSR
    - Used to establish classification rules based on range of credit hours entered and student attributes
  - SOACURR/SOBCURR
    - Refer to section on Curriculum/Program Rules
    - If rules are to be used, indicator will be 'ON' for General Student

232

---

---

---

---

---

---

---

---

**General Student** (cont.)

- **Major Validation Tables/Forms:**

STVTERM	STVRES
STVLEVL	STVSTST
STVCAMP	STVCLAS
STVCOLL	STVDLEV
STVDEGC	STVACAT
STVMAJR	STVSTYP

233

---

---

---

---

---

---

---

---

**General Student** (cont.)

- **Additional Information:**
  - SGRADV - Multiple advisors
  - SGRSPRT - Sports
  - SGRCHRT - Cohorts
  - SGRSATT - Attributes
  - SGRDISA - Disability Services
  - And other tables

234

---

---

---

---

---

---

---

---

**General Student:  
SQL\*Plus**

- What tables are part of the General Student Module?

```
— select table_name  
  from all_tables  
  where table_name like 'SG%'
```

- What data elements are required?

```
— desc sgbstdn  
— notice the “NOT NULL” columns
```

- What are the key fields in sgbstdn?

```
select column_name  
  from all_cons_columns  
  where table_name = 'SGBSTDN'  
        and constraint_name = 'FK_SGBSTDN';
```

---

---

---

---

---

---

---

---

**General Student:  
Reports and Processes**

- SGP HOLD -- Hold Purge
- SGPSTDN -- General Student Purge
- SGRSTDN -- Student Report

---

---

---

---

---

---

---

---

**General Student: Other Scripts**

- \$BANNER\_HOME/student/dbprocs  
 — functions (sgf')
- \$BANNER\_HOME/student/views  
 — views (sgv'): sgvstd0.sql creates view as\_student\_data  
 — some views used in conjunction with Object:Access method of  
 retrieving data from database
  - uses concept of “layered” views
  - must have GTVSDAX form/table populated with crosswalk  
 values

---

---

---

---

---

---

---

---

**General Student:  
Conversion Issues**

- What General Student data do you have in your legacy system?
- How far back do you wish to go with your data conversion?
- How do you determine where to put legacy data in Banner?
- Must have a student record with sgbstdn\_term\_code\_eff = first term of history

238

---

---

---

---

---

---

---

---

**General Student:  
Summary**

- Major Forms:
  - SGASTDN
  - SGACLSR
  - SOACURR
- Major Tables
  - SGBSTDN
  - VALIDATION TABLES

239

---

---

---

---

---

---

---

---

**General Student: Exercise**

- Write a query that returns the student's full name, id, advisor's name, major code, and residency code from the current student record.

240

---

---

---

---

---

---

---

---





# Banner Student Technical Accounts Receivable

---

---

---

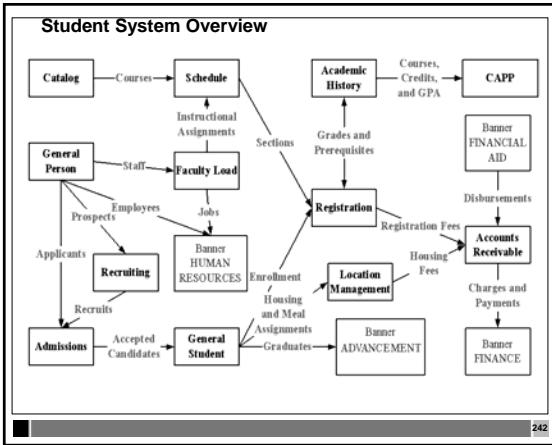
---

---

---

---

---



---

---

---

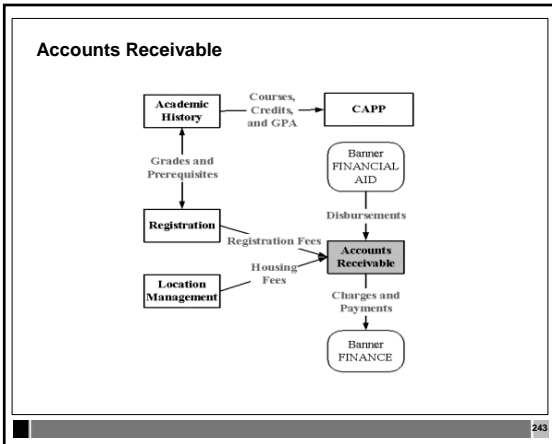
---

---

---

---

---



---

---

---

---

---

---

---

---

### Accounts Receivable: Objectives

- **Examine/Review**
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

244

---

---

---

---

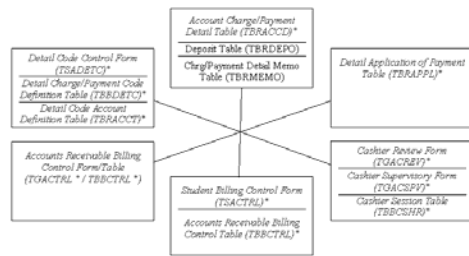
---

---

---

---

### Accounts Receivable



245

---

---

---

---

---

---

---

---

### Accounts Receivable (cont.)

- **Major Validation Tables/Forms:**
  - TTVBILL
  - TTVDCAT
  - TTVDELI
  - TTVDTYP
  - TTVPAYT
  - TTVSRCE
  - TTVTBDS
  - TTVTAXT (for Canadian Inst. only)

246

---

---

---

---

---

---

---

---

**TGACTRL/TSACTRL**

- TGACTRL -- Accounts Receivable Billing Control Form: TBBCTRL table
- TSACTRL -- Student Billing Control Form: TBBCTRL table
- Required for Conversion

---

---

---

---

---

---

---

---

**TSADETC**

- Enter detail code information
- Establish payment priorities used in TGRAPPL (application of payments process)
- TBBDETC, TBRACCT tables
- Establishes Interface with Finance Package
  - Banner Finance or Third Party Package
- Set up Fund Codes, Account Numbers
- Required for conversion

---

---

---

---

---

---

---

---

**AR Rules Forms**

- TSASBRL -- Schedule/Bill Rules Form:
  - TBBSBRL table
  - Sets up parameters used in TSRCBIL
- TSATBDS -- Term-based Designator Rules Form:
  - TBBTBDS table
  - Allows users to establish relationships between term codes and term-based AR designators

---

---

---

---

---

---

---

---

**TGACREV/TGACSPV**

- **TGACREV -- Cashier Session Review Form:**
  - TBBCSHR table
  - Used to review all charge or payment activity for a specific session
- **TGACSPV -- Cashier Supervisory Form:**
  - TBBCSHR table
  - Used to display all active and inactive cashiering sessions on the system

---

---

---

---

---

---

---

---

**TSADETL**

- **Student Account Detail Form**
  - Holds Account Detail by Detail Code
    - Major Table = TBRACCD
  - Also Shows Deposits, Memos and Comments
    - Table = TBRDEPO
    - Table = TBRMEMO
    - Table = TBRCMNT

---

---

---

---

---

---

---

---

**TSAAREV**

- **Student Account Detail Review Form**
  - Used to review and enter information about an account
  - Presents an online view of each transaction by term
  - Table: TBRACCD
- **Also accessed from SFAREGS**

---

---

---

---

---

---

---

---

**TSASPAY**

- **Student Payment Form: TBRACCD table**
  - Used to determine status of student's account for a term
  - Can be used to accept charges, disburse financial aid
  - Affected by changes in TSADETL, SFAREGS, SLAMASG and others

---

---

---

---

---

---

---

---

**Accounts Receivable: SQL\*Plus**

- **What tables are part of the Accounts Receivable Module?**

```
-- select table_name  
from all_tables  
where table_name like 'T'
```

- **What data elements are required?**

```
-- desc tbraccd  
-- notice the "NOT NULL" columns
```

- **What are the key fields in tbraccd?**

```
select column_name  
from all_cons_columns  
where table_name = 'TBRACCD'  
and constraint_name = 'FK_TBRACCD';
```

---

---

---

---

---

---

---

---

**Accounts Receivable:  
Reports and Processes**

- TGRAPPL -- Application of Payment
- TGRFEED -- Accounting Feed Process
- TSRCBIL -- Invoice/Billing Statements

---

---

---

---

---

---

---

---

**TGRAPPL**

- Application of Payments Process
  - Applies payments to charges for accounts based on priority (tbbdetc\_priority)
  - Creates correct accounting entries to be fed by TGRFEED process
  - “C” program run from Job Submission
  - Gets other rules from TBBCTRL table

---

---

---

---

---

---

---

---

---

---

**TGRAPPL** (cont.)

- Populates TBRAPPL table
- Results visible in TSIAPPL form
- tbrappl\_acct\_feed\_ind
  - Indicates status of the application of payment in accounting feed process.  
Valid values:
    - Y - Transaction ready to feed,
    - F - Transaction fed

---

---

---

---

---

---

---

---

---

---

**TGRAPPL** (cont.)

TBRACCD TABLE	TGRAPPL APPLIES:	BALANCES	TBRAPPL TABLE
\$35 ACTF [tran num 1]	\$7000 AMEX pmt [tbraccd tran num 3]	chg = \$0 pmt = \$6965	tbrappl_chg_tran_number 1 tbrappl_pay_tran_number 3 tbrappl_amount = \$35
\$7500 T101 [tran num 2]	\$6965 pmt bal [bal. of AMEX pmt tbraccd tran num 3]	chg = \$535 pmt = \$0	tbrappl_chg_tran_number 2 tbrappl_pay_tran_number 3 tbrappl_amount = \$6965
\$7000 AMEX [tran num 3]	applied to charges in trans 1 & 2		
\$535 CHEK [tran num 4]	\$535 CHEK pmt	chg. = \$0 pmt. = \$0	tbrappl_chg_tran_number 2 tbrappl_pay_tran_number 4 tbrappl_amount = \$535

---

---

---

---

---

---

---

---

---

---

**TGRFEED**

- **Accounting Feed Process**
  - C Program run from Job Submission
  - Uses data from TBRAPPL, TBRDEPO, TBRMISD, TBRACCD
  - Refers to TBRACCT, TBBDETC, TBBCTRL tables for distribution, detail information

---

---

---

---

---

---

---

---

**TGRFEED** (cont.)

- Creates GURFEED entries -- accounting detail records that interface Accounts Receivable module with the institution's financial accounting system
- Creates GURAPAY entries -- refund and check information
- Updates source tables to show that those records have been fed into the General Ledger
- Produces report that details debit and credit entries by account number

---

---

---

---

---

---

---

---

**TSRCBIL**

- **Student Invoice/Billing Statement**
  - C program run from Job Submission
  - Can be run in sleep/wake mode
  - Rule parameters set on TSASBRL Bill Selection Parameters Window (TBBSBRL table)
  - Updates AR indicators in SLRMASG, SLRPASG, SLRRASG, SFBETRM

---

---

---

---

---

---

---

---

**Accounts Receivable: Other Scripts**

- **\$BANNER\_HOME/student/arsys**
  - functions (t\*f) ex: tofbala.sql
- **\$BANNER\_HOME/student/views**
  - views (t\*v): tofbal0.sql creates view at\_ar\_history\_by\_balance
  - some views used in conjunction with Object:Access method of retrieving data from database
    - uses concept of “layered” views
    - must have GTVSDAX form/table populated with crosswalk values

---

---

---

---

---

---

---

---

**Accounts Receivable:  
Conversion Issues**

- **Balance Forward**
  - Set up controls on TSACTRL
  - Populate TBBDETC -- Detail Charge/Payment Code Definition Table
  - Populate TBRACCT -- Detail Code Account Definition Table (Fund and Account Codes)

---

---

---

---

---

---

---

---

**Accounts Receivable Module:  
Summary**

- **Major Forms**
  - TGACTRL, TSACTRL, TSADETC, TSADETL, TSAAREV, TGACREV, TGACSPV
- **Major Tables**
  - TBBCTRL, TBRACCT, TBBDETC, TBRACCD, TBRDEPO, TBRAPPL

---

---

---

---

---

---

---

---



**Accounts Receivable Module:**  
**Summary** (cont.)

- **Processes**
  - TGRAPPL -- Application of Payments
  - TGRFEED -- Accounting Feed Process
  - TSRCBIL -- Student Invoice/Billing Statement; can be run in sleep/wake mode

■ 265

---

---

---

---

---

---

---

---

**Accounts Receivable:**  
**Exercise 1**

- Find all columns in the Accounts Receivable module that have to do with detail codes.

■ 266

---

---

---

---

---

---

---

---

**Accounts Receivable:**  
**Exercise 2**

- Write a simple report that will show full name, id, term and balance from the student account detail table for a given term for those students with a balance > 0. Prompt user for term.

■ 267

---

---

---

---

---

---

---

---



BANNER HIGHER EDUCATION

# Banner Student Technical Registration

---

---

---

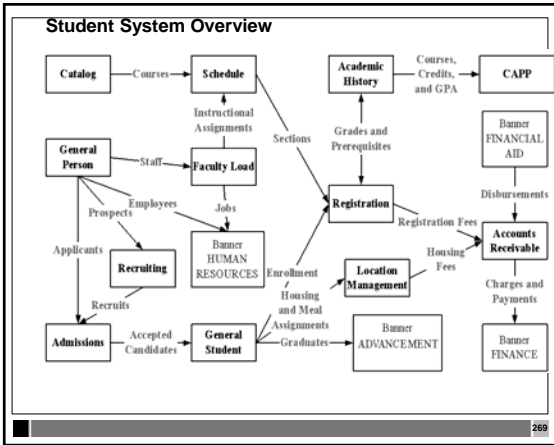
---

---

---

---

---



---

---

---

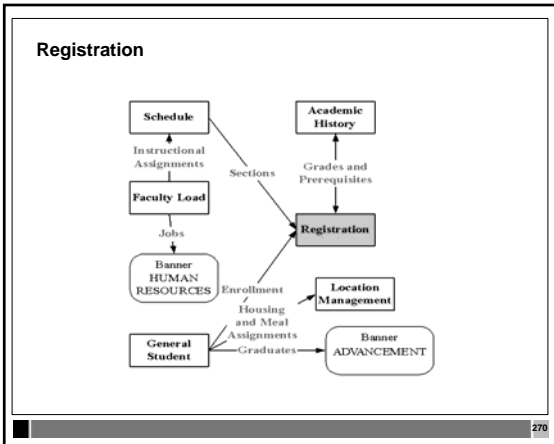
---

---

---

---

---



---

---

---

---

---

---

---

---

### Registration: Objectives

- **Examine/Review**
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

271

---

---

---

---

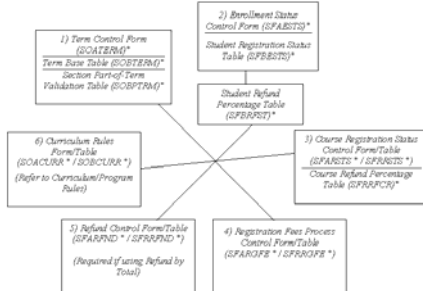
---

---

---

---

### Registration



272

---

---

---

---

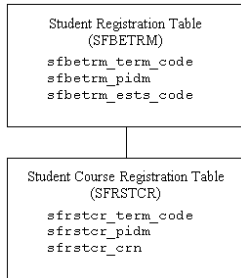
---

---

---

---

### Registration (cont.)



273

---

---

---

---

---

---

---

---

**Registration: Rules**

- **SOACURR Form**
  - SOBCURR Table
  - Curriculum Rules
- **SOATERM Form**
  - SOBTERM & SOBPTRM Tables
  - Term Controls
  - Set On-line Fee Assessment
  - Error Checking & Severity Level, etc.

274

---

---

---

---

---

---

---

---

**Registration: Rules (cont.)**

- **SFAESTS Form**
  - SFBESTS & SFBRFST Tables
  - Enrollment Status Control
  - Student Refund Percentage Table
  - STVESTS: Status Codes
- **SFARSTS Form**
  - SFRRSTS & SFRRFCR Tables
  - Course Registration Status
  - STVRSTS: Status Codes

275

---

---

---

---

---

---

---

---

**Registration: Rules (cont.)**

- **SFARGFE Form**
  - SFRRGFE Table
  - Registration Fees Process Control
- **SFARFND Form**
  - SFRRFND Table
  - Refund Control

276

---

---

---

---

---

---

---

---

**Registration** (cont.)

- **Major Validation Tables/Forms:**
  - All that were mentioned in the previous modules
  - For complete list, see Student User Manual, Registration Chapter

---

---

---

---

---

---

---

---

**Registration** (cont.)

- **Major Form/Tables:**
  - SFAREGS Form
    - Form that is the mechanism for registering students
  - SFBETRM - Table containing Registration Status
  - SFRSTCR - Table containing Course Registrations

---

---

---

---

---

---

---

---

**Registration: Fee Assessment**

- Done online or in batch process (SFRFASM)
- Uses rules built in Catalog, Schedule and Registration Modules
- Always writes a record to SFRCOLR (Registration Fee Assessment View Collector Table)
- SFRCOLR should be cleaned out periodically

---

---

---

---

---

---

---

---

**Registration: SQL\*Plus**

- **What tables are part of the Registration Module?**

```
-- select table_name  
from all_tables  
where table_name like 'SF%'
```

- **What data elements are required?**

```
-- desc sfrstcr  
-- notice the "NOT NULL" columns
```

- **What are the key fields in sfrstcr?**

```
select column_name  
from all_cons_columns  
where table_name = 'SFRSTCR'  
and constraint_name = 'FK_SFRSTCR';
```

---

---

---

---

---

---

---

---

**Registration:  
Reports and Processes**

- **SFRSCHD -- Student Schedule**  
— Can be run in sleep/wake mode
- **SFRSLST -- Class Roster**
- **SFRFASM -- Batch Fee Assessment**
- **SFRRNOP -- Registered, Not Paid Process**
- **SFPREGS -- Registration Purge**

---

---

---

---

---

---

---

---

**SFRSCHD**

- **C program run from Job Submission**
- **Can be run in sleep/wake mode**
- **Prints student schedule for term**

---

---

---

---

---

---

---

---

**SFRFASM**

- **Batch Fee Assessment Process**
  - C program run from Job Submission
  - Done if institution decides not to do online fee assessment
  - Registration charges posted to the student's account in Accounts Receivable module

283

---

---

---

---

---

---

---

---

**SFRRNOP**

- **Registered, Not Paid Process**
- C program run from Job Submission
- Can be run in query or update mode
- Prints/purges all students registered but not paid for a term

284

---

---

---

---

---

---

---

---

**Sleep/Wake Process**

- Method of running jobs in cyclical or “sleep/wake-up” manner
- Two possible methods
  1. Submit from Operating System and terminate manually
    - scripts are in \$BANNER\_HOME/general/misc and \$BANNER\_HOME/general/plus
  2. Submit through Banner Job Submission (GJAPCTL form)

285

---

---

---

---

---

---

---

---

**Sleep/Wake Process** (cont.)

- **Method 2: Sleep/Wake**
  - Define Printer and print command on GTVPRNT
  - On the SOADEST or TOADEST form, enter the correct printer code from GTVPRNT
  - On GJAPCTL, for the valid sleep/wake jobs, enter the parameters that specify sleep/wake processing
  - Stop sleep/wake process on GJASWPT form

286

---

---

---

---

---

---

---

---

**Sleep/Wake Process** (cont.)

- **Jobs that can be run in sleep/wake mode:**
  - SFRSCHD -- Student Schedule
  - SHRTRTC -- Academic Transcripts
  - TGRRRCPT -- Account Receipts
  - TSRCBIL -- Student Billing Statement (Invoices)
  - TGRMISC -- Miscellaneous Receipt

287

---

---

---

---

---

---

---

---

**Registration Module: Other Scripts**

- **\$BANNER\_HOME/student/dbprocs**
  - functions (sff) ex: sffrgfe1.sql
- **\$BANNER\_HOME/student/views**
  - views (sfv): sfvstc0.sql creates view as\_student\_registration\_detail
  - some views used in conjunction with Object:Access method of retrieving data from database
    - uses concept of "layered" views
    - must have GTVSDAX form/table populated with crosswalk values

288

---

---

---

---

---

---

---

---



**Registration:  
Conversion Issues**

- Not recommended
- Possibly could run parallel
  - Legacy and Banner

289

---

---

---

---

---

---

---

---

**Registration Module: Summary**

- Major Form: SFAREGS
- Major Tables
  - SFBETRM
  - SFRSTCR
  - Validation Tables
- Sleep/Wake
- Connection to AR: Fee Assessment

290

---

---

---

---

---

---

---

---

**Registration: Exercise**

- Write a query that returns student's full name and a list of courses for which he is registered for a given term, including: subject and course number, crn, and credit hours. Prompt user for term.

291

---

---

---

---

---

---

---

---



# Banner Student Technical Academic History

---

---

---

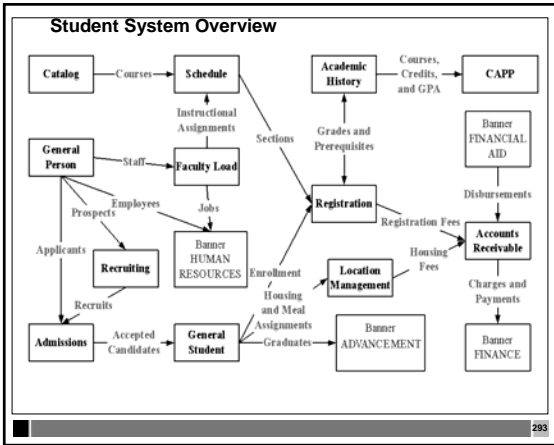
---

---

---

---

---



---

---

---

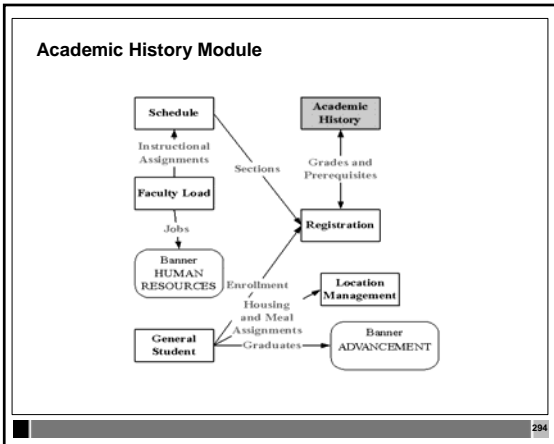
---

---

---

---

---



---

---

---

---

---

---

---

---

**Academic History Module: Objectives**

- **Examine/Review**
  - Major and Required Forms and Tables
  - Reports, Processes and Procedures
  - Conversion of Data

---

---

---

---

---

---

---

---

---

---

**Academic History: Rules**

Grade Code Maintenance Form (SHAGRDE)*	Academic Standing Rules Form (SHAACST)*	Repeat/Equivalent Course Rules Form (SHARPTF)*
Grade Code Maintenance Table (SHGRDE)*	Academic Status Rules Table (SHRASTR)*	Repeat/Equivalent Course Rules Table (SHRPPTF)*
Valid Grading Modes Maintenance Table (SHRCMDO)*	Dean's List Calculation Rules Table (SHRASDL)*	Continuant Term Rules Form (SOACTRM)*
Transcript Type Rules Form (SHATPRT)*	Dean's List Grade Code Excluded Table (SHRASGE)*	Continuant Terms Table (SORCTRM)*
Transcript Rule Request Type Table (SHRTPRT)*		

---

---

---

---

---

---

---

---

---

---

**Academic History: Rule Forms/Tables**

- **SHAGRDE/SHGRDE**
  - Grade Code Maintenance
- **SHATPRT/SHRTPRT**
  - Transcript Type Rules
- **SHAACST**
  - Academic Standing Rules
    - SHRASTR - Academic Status
    - SHRASDL - Deans List Calculation
    - SHRASGE - Deans List Grade Code Excluded

---

---

---

---

---

---

---

---

---

---

**Academic History:  
Rule Forms/Tables**

- SHARPTR/SHBRPTR  
— Repeat Equivalent Course Rules
- SOACTRM/SORCTRM  
— Continuant Term Rules

---

---

---

---

---

---

---

---

**Academic History (cont.)**

- Major Validation Tables/Forms:
  - STVGCHG
  - STVGMOD
  - STVASTD
- Also many of the validation tables that have been referenced in previous modules

---

---

---

---

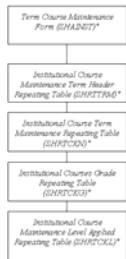
---

---

---

---

**Academic History:  
Institutional Courses**



---

---

---

---

---

---

---

---

**Academic History:  
Institutional Courses - SHAINST**

- SHRTTRM - Term Header
  - Academic Status
  - Dean's List
- For each institutional course taken:
- SHRTCKN - subjects, course numbers, titles, etc.
- SHRTCKG - credit hours, final grade, etc.
- SHRTCKL - course level applied
- Records are associated by term and SHRTCKN sequence number

301

---

---

---

---

---

---

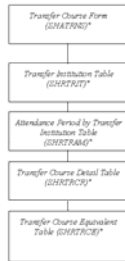
---

---

---

---

**Academic History: Transfer Courses**



302

---

---

---

---

---

---

---

---

---

---

**Academic History:  
Transfer Courses - SHATRNS**

- Required only if detail of transfer courses is to be converted
- For each course transferred:
- SHRTRIT - transfer institution
- SHRTRAM - attendance period by institution
- SHRTRCR - transfer course detail
- SHRTRCE - equivalent subject, course number, grade, etc.
- Records are associated by term, and SHRTRAM, SHRTRIT, SHRTRCR sequence numbers (review carefully)

303

---

---

---

---

---

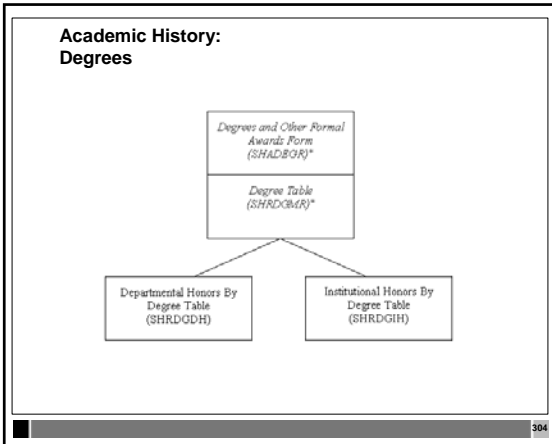
---

---

---

---

---




---

---

---

---

---

---

---

---

**Academic History:  
Degree Information - SHADEGR**

- **SHRDGMR - Degree Repeating Table**
  - Required even if student does not have degree
  - SHRDGMR\_DEGC\_CODE = 'SO' for "seeking"
  - SHRDGMR\_DEGC\_CODE = 'DA' or 'AW' for "degree awarded" if student has degree
  - Contains major and term awarded
- **SHRDGIH - Used if student had institutional honors associated with the degree**
- **SHRDGDH - Used if student had departmental honors associated with the degree**
  - Records in SHRDGIH and SHRDGDH are associated by SHRDGMR sequence number

305

---

---

---

---

---

---

---

---

**Academic History: GPA**

- **SHRLGPA - Level GPA Table**
  - Cumulative institutional courses (I)
  - Cumulative transfer courses (T)
  - Overall GPA (O)
    - Includes both institutional and transfer courses
- **SHRTGPA - Term GPA Table**
  - Term statistics for institutional courses (I)
  - Term statistics for transfer courses (T)

306

---

---

---

---

---

---

---

---

**Academic History:  
Pre-Banner Summary**

*Pre-Banner Summary  
Hours and GPA Form  
(SHAPCMP)\**

*Term GPA Table  
(SHRTGPA)\**

*Term Sequence Course  
History Form  
(SHATERM)\**

*Level GPA Table  
(SHRLGPA)\**

307

---

---

---

---

---

---

---

---

**Academic History:  
Summary Conversion**

- **SHRTGPA**
  - Results displayed in SHAPCMP
  - At least one record per student in SHRTGPA
    - shrtgpa\_type\_ind = 'I'
    - would reflect total cumulative statistics
  - Use '000000' as the term code
  - Two possible records in SHRTGPA ('I' and 'T' GPA types)

308

---

---

---

---

---

---

---

---

**Academic History:  
Summary Conversion** (cont.)

- **SHRLGPA**
  - SHRLGPA could contain up to three records per student per level, but at least two records
  - (I)nstitutional GPA and (O)verall GPA -- (T)ransfer GPA is the other possibility
  - (O)verall will reflect Institutional and Transfer GPAs combined

309

---

---

---

---

---

---

---

---

**Academic History: SQL\*Plus**

- **What tables are part of the Academic History Module?**

```
— select table_name  
  from all_tables  
  where table_name like 'SH%'
```

- **What data elements are required?**

```
— desc shrdgmr  
— notice the "NOT NULL" columns
```

- **What are the key fields in shrdgmr?**

```
select column_name  
  from all_cons_columns  
  where table_name = 'SHRDGMR'  
  and constraint_name = 'FK_SHRDGMR';
```

---

---

---

---

---

---

---

---

**Academic History:  
Other Scripts**

- **\$BANNER\_HOME/student/dbprocs**

— Functions (shf\*) ex: shfttm.sql

- **\$BANNER\_HOME/student/views**

— Views (shv\*): shvsum0.sql creates view as\_academic\_history\_summary

— Some views used in conjunction with Object:Access method of retrieving data from database

- Uses concept of "layered" views
- Must have GTVSDAX form/table populated with crosswalk values

---

---

---

---

---

---

---

---

**Academic History:  
Conversion Issues**

- Will detailed academic history data be converted?
- Will you convert ALL academic history records or only a certain number of years?
- What academic history data do you have in your legacy system?
- How do you determine where to put it in Banner?

---

---

---

---

---

---

---

---



**Academic History:  
Reports/Processes - End of Term**

- SHRROLL -- Grade Roll Process
- SHRRPTS -- Repeat Course Check
- SHRCGPA -- Calculate GPA
- SHRASTD -- Calculate Academic Standing
- SHRGRDE -- Grade Mailer
- SHRTYPE -- Student Type Update

313

---

---

---

---

---

---

---

---

**Academic History:  
Reports and Processes**

- SHRTRTC -- Academic Transcript
  - Can be run in sleep/wake mode
- SHRDEGS -- Degree Status Update

314

---

---

---

---

---

---

---

---

**Academic History: Summary**

- Major Forms: Major Tables
  - Institutional Courses
  - Transfer Courses
  - Degree Information
  - GPAs
  - Pre-Banner Summary
  - Validation Tables
- Reports and Processes

315

---

---

---

---

---

---

---

---

**Academic History: Exercise 1**

- Write a query which returns full name, id, level (the level code associated with gpa hours and calculations), and term gpa (for institutional work) for a given term. Prompt user for term.

---

---

---

---

---

---

---

---

**Academic History: Exercise 2**

- Write a query which returns full name, id, course level, crn, subject code, course number, and grades for a given term. Prompt user for term.

---

---

---

---

---

---

---

---

**Academic History: Exercise 3**

- Write a query which returns full name, id, level, all transfer courses, and grades for all students who have transfer work. Order by student last name.

---

---

---

---

---

---

---

---



## Banner Student Technical Conversion

---

---

---

---

---

---

---

---

### Conversion

- Conversion Considerations
- Conversion Steps
- Conversion Strategies
- Seed Data
- Conversion Example

---

---

---

---

---

---

---

---

### Conversion Considerations

- Keeping track of PIDM on legacy system
- Generated ID or SSN?
- Name/Address formatting
  - Avoid “#” if using letter generation
  - Additional data standards if using BannerQuest
- Address types
- Do you have Multiple ID's on legacy system?

---

---

---

---

---

---

---

---

**Conversion Steps**

1. Document steps as you proceed
2. Review current data
3. Determine scope:
  - What will you convert?
4. Map legacy data to Banner tables
5. Write a detail plan of:
  - Data to be converted
  - Banner tables to be populated
  - Deadlines/timelines

322

---

---

---

---

---

---

---

---

**Conversion Steps (cont.)**

6. Review plan and get approval from users
7. Develop procedures and programs
8. Test conversion in TEST or PPRD database
9. Users verify data
10. TEST again and make corrections to procedures and programs
11. Do conversion in production
12. Users verify data

323

---

---

---

---

---

---

---

---

**Conversion Strategies**

- Create data standards, especially for names and addresses
  - All offices need to agree to and document data standards
- Determine whether you will enter the data electronically or manually
  - E.g., Some validation tables/forms can be entered manually in both the preproduction and production databases *if* the number of records is small (unlike STVSBG!!!)

324

---

---

---

---

---

---

---

---

**Conversion Strategies** (cont.)

- **Determine which tables you will be using**
  - May be helpful to look at the forms with the users, then you will be able to determine the tables used
- **Mapping legacy data to Banner**
  - Review the legacy to Banner mapping with the users
- **Create a document programmers can use that tells how to convert the data**

---

---

---

---

---

---

---

---

**Conversion Strategies** (cont.)

- **Create a Conversion Plan document**
  - Review the steps that are involved to get to your "go live" dates
  - Create a time line
  - Determine the processes that need to be written
    - Will data need to be translated?
    - Will data need to be cleaned up on legacy side?

---

---

---

---

---

---

---

---

**Seed Data**

- The Student Technical Reference Manual lists all validation table values that must be kept in production
  - System required indicator = 'Y'
- This does not mean one must use this information
- Most of this information is needed by external reports to third-parties
  - E.g., IPEDS

---

---

---

---

---

---

---

---

**Conversion Example**

**Shows How To:**

- Convert data to three Banner tables
- Create, drop, and alter temporary tables
- Assign a pidm
- Use SQL\*LOADER to load temporary tables
- Use Update statement and Decode function to do cross-walk (translation)

---

---

---

---

---

---

---

---

**Conversion Example**

**Shows How To:** (cont.)

- Use Insert statement
- Use a shell script or command procedure
- Check the data when complete
- Clean up data if it is incorrect

---

---

---

---

---

---

---

---

**Conversion Example**

- Uses a flat file containing
  - Person's (student's) SSN
  - Last name
  - First name
  - Street
  - City
  - State
  - Zip
  - Sex
  - Birth date

---

---

---

---

---

---

---

---

### Conversion Example (cont.)

- Example of a Flat File Layout

```
210009506Abbe Anthony PO Box 21049 Malvern PA19355226-MAR-77
610009711Abbot James PO Box 27 Malvern PA19355217-NOV-79
210009101Adams Andrew 883 King Street Malvern PA19355210-DEC-72
610009101Adams Anthony 20789 Lancaster Ln Clarksville PA15122210-DEC-74
710000011Adams Eugene 3400 Wendrow Way University ParkPA16802201-JAN-01
210009619Barker Clementine 83 Park Avenue New York NY10013128-APR-72
210009613Barker James 854 Charlestown Pk King of PrussiaPA19401201-DEC-77
```

331

---

---

---

---

---

---

---

---

---

---

### Conversion Example: Create Statement

```
Create temporary tables (create_temp.sql):
spool create_tables
drop table sytiden;
drop table sytaddr;
drop table sytpers;
create table sytiden as select * from spriden where 1 = 2;
create table sytaddr as select * from spraddr where 1 = 2;
create table sytpers as select * from spbpers where 1 = 2;
spool off
```

332

---

---

---

---

---

---

---

---

---

---

### Conversion Example: Alter Statement

```
Alter temporary tables (alter_temp.sql):
spool alter_tables
alter table sytiden modify spriden_pidm null;
alter table sytaddr modify spraddr_pidm null;
alter table sytpers modify spbpers_pidm null;
spool off
```

333

---

---

---

---

---

---

---

---

---

---

### Conversion Example: SQL\*LOADER

```
SQL*LOADER (load.ct1):

load data
infile 'data.file.dat'
badfile 'bad.data.txt'
discardfile 'discard_file.txt'
append
into table sytiden (
  spriden_pidm          sequence(77777777,1),
  spriden_id            position(1:9),
  spriden_last_name     position(10:23),
  spriden_first_name    position(24:39),
  -- spriden_change_ind null,
  spriden_entity_ind    constant 'P',
  spriden_activity_date constant '25-DEC-98',
  spriden_user          constant 'CONVERSION',
  spriden_origin        constant 'CONVERSION')

```

334

---

---

---

---

---

---

---

---

---

---

### Conversion Example: SQL\*LOADER (cont.)

```
into table sytaddr (
  spraddr_pidm          sequence(77777777,1),
  spraddr_atyp_code     constant 'MA',
  spraddr_segno        constant '1',
  spraddr_street_line1 position(40:58),
  spraddr_city          position(59:73),
  spraddr_stat_code     position(74:75),
  spraddr_zip           position(76:80),
  spraddr_activity_date constant '25-DEC-98',
  spraddr_user          constant 'CONVERSION')
into table sytpers (
  spbpers_pidm          sequence(77777777,1),
  spbpers_ssn           position(1:9),
  spbpers_sex           position(81:81),
  spbpers_birth_date    position(82:90),
  spbpers_activity_date constant '25-DEC-98')

```

335

---

---

---

---

---

---

---

---

---

---

### Conversion Example: Decode Statement

```
Decode SPBPERS_SEX (decode_sex.sql):

spool decode
update sytpers
set spbpers_sex = decode (spbpers_sex,'1','F','2','M','N');
spool off

```

336

---

---

---

---

---

---

---

---

---

---



**Conversion Example:  
Check Data in the Temp Tables**

```
select      spriden_id, substr(spriden_last_name,1,15) ||
           ',' || spriden_first_name,
           spriden_change_ind IND,
           spriden_entity_ind ENT,
           spriden_activity_date, spriden_pidm,
           spraddr_pidm, spbpers_pidm,
           spraddr_street_line1, spraddr_city,
           spraddr_stat_code, spraddr_zip, spbpers_sex,
           spbpers_birth_date
           from      sytiden, sytaddr, sytpers
           where     spriden_pidm = spraddr_pidm
                   and spriden_pidm = spbpers_pidm
           order by  spriden_pidm;
```

337

---

---

---

---

---

---

---

---

---

---

**Conversion Example:  
Insert Statement**

```
Insert into SATURN tables (insert_real.sql):

spool insert_real
insert into spriden select * from sytiden;
insert into spraddr select * from sytaddr;
insert into spbpers select * from sytpers;
spool off
```

338

---

---

---

---

---

---

---

---

---

---

**Conversion Example:  
Check the Data in Banner**

```
select      spriden_pidm, substr(spriden_last_name ||
           ',' || spriden_first_name,1,25),
           spriden_entity_ind, spraddr_atyp_code,
           spraddr_seqno, spraddr_street_line1,
           spraddr_city, spraddr_stat_code,
           spraddr_zip, spbpers_sex,
           spbpers_birth_date
           from      spraddr, spbpers, spriden
           where     spriden_pidm > 7777776
                   and spriden_pidm = spraddr_pidm
                   and spriden_pidm = spbpers_pidm
           order by  spriden_pidm;
```

339

---

---

---

---

---

---

---

---

---

---

**Conversion Example:  
Update SOBSEQN**

```
update sobseqn
set sobseqn_maxseqno = 77777783,
    sobseqn_activity_date = sysdate
where sobseqn_function = 'PIDM';
```

340

---

---

---

---

---

---

---

---

**Conversion Example:  
Clean the Data in Banner**

Clean SATURN tables (clean\_tables.sql)

```
spool clean_tables
delete from spriden where spriden_pidm > 77777776;
delete from spraddr where spraddr_pidm > 77777776;
delete from spbpers where spbpers_pidm > 77777776;
spool off
```

341

---

---

---

---

---

---

---

---

**Conversion Example:  
Shell script**

```
Shell_Script (convert.shl):
export ORAENV_ASK=NO
export ORACLE_SID=YOURSID
. oraenv

sqlplus saturn/u_pick_it @create_temp
sqlplus saturn/u_pick_it @alter_temp
sqlldr saturn/u_pick_it control=load.ctl
sqlplus saturn/u_pick_it @decode_sex
sqlplus saturn/u_pick_it @insert_real
```

342

---

---

---

---

---

---

---

---

**Conversion Example:  
Summary**

- Determine flat file layout
- Create temporary tables
- Alter temporary tables, if necessary
- Create loader control file
- Load flat file data into temporary tables

343

---

---

---

---

---

---

---

---

**Conversion Example:  
Summary (cont.)**

- Translate codes using Decode, if necessary
- Check temporary table data
- Insert temporary table data into Banner tables
- Check data in Banner (in forms and tables)
- Update sobseqn
- Use shell script

344

---

---

---

---

---

---

---

---



**Banner Student Technical  
APIs**

---

---

---

---

---

---

---

---

### APIs

- Application Programming Interfaces (APIs) are used to facilitate the integration of Banner with other applications on campus and simplify code by encapsulating business logic in database packages
- An API is a central program that creates, updates, and deletes data
- APIs also execute and validate business rules before inserting or updating information

346

---

---

---

---

---

---

---

---

### APIs Disclaimer

- **Warning:** Please be advised that several APIs are currently intended to only support internal operations
- To ensure data integrity, these APIs are not supported when called by external applications or interfaces to manipulate data
- The recommendation for external applications is to use message level integration to integrate with these entities in Banner
- The following APIs come under this disclaimer:
  - gb\_stvterm
  - sb\_course\_registration
  - sb\_enrollment
  - sp\_grading

347

---

---

---

---

---

---

---

---

### APIs used in Banner Student

- Refer to the workbook for tables of APIs used in Student
- All APIs and packages used in Banner Student will:
  - begin with an 's,'
  - followed by a 'b'
  - and an underscore '\_'
  - followed by a verb/noun describing what the API does
  - If the 2nd letter is a 'p' instead of a 'b,' then that database package is a Business Process API defining a 'unit of work'
- Some Student API packages that are shared across other Banner Modules may begin with a 'g' instead of an 's'

348

---

---

---

---

---

---

---

---

### Curricula Checking and APIs

- The curriculum and field of study APIs are used to validate the curriculum according to the rules on SOACURR
- An in and out parameter exists on the curriculum and field of study p\_create APIs to override any raised failure and to return the severity level so the user interface can manage the message if necessary

349

---

---

---

---

---

---

---

---

### Curricula Checking and APIs (cont.)

- If the override is blank or *F*, the API will abort if there is a fatal curriculum error, and the API error message handler will display the error
- If the override value has a value of *N* (No Curriculum Checking), the API will not fail, and it will insert the data
- In all cases, the real severity level and error number are returned so the user interface can display any messages
- Refer to a workbook for a table of relevant APIs

350

---

---

---

---

---

---

---

---

### Curriculum Conversion using Functions and APIs

- Four of the functions in package SOKLCUR (f\_convert\_recruit, f\_convert\_applicant, f\_convert\_learner, and f\_convert\_outcome) are used to call APIs to perform the insert activity
- The functions can be launched from a batch process or when the a form with curriculum data is initially opened at your institution
  - This includes: SRARECR, SRAQUIK, SAAADMS, SAAQUIK, SGASTDN, SFAREGS, and SHADEGR
  - These forms will check to see if the curriculum data has been converted, and if it has not, the function will be executed for the PIDM
- Refer to the workbook for a table explaining how the functions convert the curriculum data using APIs

351

---

---

---

---

---

---

---

---



## Banner Student Technical Course Summary

---

---

---

---

---

---

---

---

- Student Technical Training: Summary**
- How to Contact SunGard Higher Education
  - Banner Navigation
  - Banner naming conventions
  - Directory structure and content
  - Data Dictionary
  - Referential Integrity
  - Joining tables
  - Job Submission, Sleep Wake, Reports and Processes
  - Conversion

---

---

---

---

---

---

---

---

- Student Technical Training: Summary (cont.)**
- **Student System**
    - Catalog
    - General Person
    - Curriculum/Program and Overall Tables
    - Recruiting and Admissions
    - Faculty Load, Location Management
    - Schedule, General Student
    - Accounts Receivable, Registration
    - Academic History
  - **Value-Based Security**

---

---

---

---

---

---

---

---

**Closing**  
Course Evaluations

955

---

---

---


---

---

---

---

---



**Questions and Answers**

---

---

---

---

---

---

---

---



**Thank you for your participation**

---

---

---

---

---

---

---

---