

**SCT Banner  
Technical Training**

**Financial Aid Technical**

---

---

---

---

---

---

---

---

**Prerequisites**

- ◆ **SCT Banner Navigation training**
- ◆ **Introduction to Oracle**
- ◆ **If the participant does not possess Navigation and Oracle experience, he or she may find it difficult to complete the course objectives**

---

---

---

---

---

---

---

---

2

**Prerequisites (Cont.)**

- ◆ **Database preparation includes:**
  - ◆ **Installation of the SCT Banner Financial Aid System**
  - ◆ **Training accounts that are accessible so that every class participant can connect to the Unix/VAX/NT machine**
  - ◆ **Ability to connect to SQL\*Plus and select from any SCT Banner table**

---

---

---

---

---

---

---

---

3

## Target Audience

- ◆ The target audience for this training course includes:
  - Technical Staff
  - "Power Users"
  - Financial Aid Managers who may use or teach others about SCT Banner Financial Aid tables and processes

---

---

---

---

---

---

---

---

4

## Introductions

- ◆ Instructor
- ◆ Participants
  - Organization
  - SCT Banner experience
  - Relational database experience
  - Other relevant experience
  - Responsibilities

---

---

---

---

---

---

---

---

5

## Performance objective

- ◆ Review SCT training resources
- ◆ Review exercises
- ◆ Review conversion steps and requirements
- ◆ Learn to identify the SCT Banner Financial Aid naming conventions, form types, directories and their contents, database structure, processes and procedures, and the basics of how Job Submission, Population Selection, and Letter Generation work

---

---

---

---

---

---

---

---

6

## Task Objectives

- ◆ Identify SCT Banner Financial Aid Forms and tables
- ◆ Query the SCT Banner Financial Aid Technical tables
- ◆ Follow Key Financial Aid Technical Processes
- ◆ Identify and read reports, processes, procedures and scripts in SCT Banner

---

---

---

---

---

---

---

---

7

## Overview

- ◆ SCT services
- ◆ SCT Banner naming conventions
- ◆ Data directories and their contents
- ◆ Database structure
- ◆ Job Submission
- ◆ Population Selection
- ◆ Letter Generation
- ◆ Resources for data and help

---

---

---

---

---

---

---

---

8

## SCT Education Centers

- ◆ Provide centralized location for training in SCT Banner and Oracle
- ◆ Provide an environment where SCT clients can focus on learning
- ◆ Provide an opportunity to interact with colleagues from other institutions

---

---

---

---

---

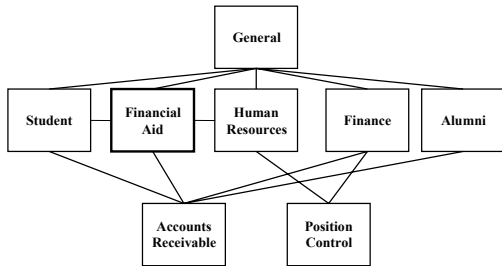
---

---

---

9

## SCT Banner products




---

---

---

---

---

---

---

---

10

## Financial Aid directories

SCT BANNER							
ADMIN	ARSYS	COMMON	GENERAL	LINKS	INSTALL	FINANCIAL AID	UTILITY
V7	C		C			C	
QRATDOLS	FORMS		FORMS			FORMS	
QPSYS	PLUS		PLUS			PLUS	
	VIEWS		VIEWS			VIEWS	
	INSTALL		INSTALL			INSTALL	
	MISC:COM		MISC:COM			MISC:COM	
	COB:COBPCO		COB:COBPCO			COB:COBPCO	
	HELP		LOADER			LOADER	
	BSPROCS		HELP			HELP	
			BSPROCS			BSPROCS	
			SY				
			ICD				
			EXE				

---

---

---

---

---

---

---

---

11

## Naming conventions

Example: RBRBCMP (Budget Component Rule Form)



- Position 1: Identifies primary SCT Banner product owning the form, report, process, or table
- Position 2: Identifies module that owns the form, report, process, or table
- Position 3: Identifies type of form, report, process, or table
- Position 4-7: A unique four-character acronym identifying the form, report, process, or table

---

---

---

---

---

---

---

---

12

### Position 1

- A Alumni/Development
- F Finance
- G General
- N Position Control
- P Human Resources/Payroll/Personnel
- R Financial Aid
- S Student
- T Accounts Receivable

13

---

---

---

---

---

---

---

---

### Position 2 When Position 1 = R

- |                                  |                              |
|----------------------------------|------------------------------|
| B Budgeting                      | N Need Analysis              |
| C Record Creation                | O Common Functions           |
| E Electronic Data Exchange (EDE) | P Packaging and Disbursement |
| F Funds Management               | R Requirements Tracking      |
| H History and Transcripts        | S Student System Shared Data |
| J Student Employment             | T Validation Form/Table      |
| L Logging                        |                              |

14

---

---

---

---

---

---

---

---

### Position 3

- A Application
- B Base Table
- I Inquiry Form
- P Process
- Q Query Form
- R Rule Form/Repeating/Report
- T Temporary Table
- V Validation Table/View

15

---

---

---

---

---

---

---

---

## Positions 4 - 7

- ◆ Unique four-character acronym identifying the form, report, process, or table

---

---

---

---

---

---

---

---

16

## Client-developed items

- ◆ Y or Z can be used as the first character for client-developed (non-SCT Banner) applications built to coexist with SCT Banner applications
- ◆ For client-developed reports or modules used within a SCT Banner application, the SCT System identifier should be used as the first character (i.e., R = Financial Aid) and W, Y, or Z should be used as the second character

---

---

---

---

---

---

---

---

17

## Form types

- ◆ **Menu**
  - Lists all related forms
  - Outlines the System
- ◆ **Application/Functional**
  - Allows one to enter, update, and query
- ◆ **Validation**
  - Lists all possible values for a given field
  - Information may be entered
- ◆ **Rule/Control**
  - Defines how variables, objects, and applications may be used
- ◆ **Query**
  - Allows information to be queried and returned to another form
- ◆ **Inquiry**
  - Used only to look up information

---

---

---

---

---

---

---

---

18

## Validation forms

- ◆ Always have TV as 2<sup>nd</sup> and 3<sup>rd</sup> characters
- ◆ Must have:
  - A code
  - A description
  - An activity date
- ◆ Validation tables
  - Table and form names are the same
- ◆ Examples :
  - Budget Type Validation table (RTVBTYPE)
  - Budget Group Validation table (RTVBGRP)
  - Holds Validation table (RTVHOLD)

19

---

---

---

---

---

---

---

---

---

---

## Query forms

- ◆ The 3<sup>rd</sup> character is a Q
- ◆ Can only be reached when called by another form
- ◆ Cannot be used to change information (used strictly for lookups)
- ◆ Examples:
  - Fund Determination Form (RFQDETC)
  - Fund Query Form (RFQFUND)
  - Disbursement Validation Query Form (ROQPVAL)
  - Award Disbursement Query Form (RPQAWRD)

20

---

---

---

---

---

---

---

---

---

---

## Inquiry forms

- ◆ The 3<sup>rd</sup> character is an I
- ◆ Cannot be used to change information
- ◆ Used strictly for lookups
- ◆ May be accessed from any menu
- ◆ Examples:
  - Applicant Need Analysis Application Inquiry Form (RNIAPPL)
  - Applicant Data Log Inquiry Form (ROIALOG)
  - Aid Year Inquiry Form (ROIADY)
  - Budget Inquiry Form (RBIBUDG)

21

---

---

---

---

---

---

---

---

---

---

## Database Dictionary

- ◆ Set of tables and views that are used as a read-only reference about the database
- ◆ One of the most important parts of an Oracle database, because it contains:
  - Names of schema objects (tables, views...)
  - Names of Oracle users
  - Privileges and roles granted to each user
- ◆ Maintained internally
  - System table space
  - Select only

22

---

---

---

---

---

---

---

---

---

---

## Database Dictionary (Cont.)

- ◆ In SQL\*Plus, describe dictionary (or *desc dict*)
- ◆ SYS owns all base dictionary tables
  - Users have access to views of the data dictionary
  - No user should alter any object contained in the SYS schema
- ◆ During normal operation, Oracle verifies the existence of objects and checks whether users have proper access to them

23

---

---

---

---

---

---

---

---

---

---

## Database Dictionary (Cont.)

- ◆ catalog.sql
  - Creates the Data Dictionary and public synonyms
  - Grants public access to the synonyms

24

---

---

---

---

---

---

---

---

---

---



## Database Dictionary (Cont.)

- ◆ The Data Dictionary is divided by users. To retrieve basic information, use:
  - ALL: Lists all objects that a user has been granted access to
  - USER: Lists all objects that a user owns
  - DBA: Lists all objects in the database

Product	Table Owner in Oracle	Module Owner in Banner
Accounts Receivable	FAISMOR	FAISUSR
Alumni	ALUMNI	ADISUSR
General	GENERAL	GENERAL
Finance	FISMGR	FISMUSR
Payroll	PAYROLL	HRISUSR
Position Control	COBNCTL	HRISUSR
Financial Aid (Resources)	FAISMOR	FAISUSR
Student	SATURN	SASUSR

25

---

---

---

---

---

---

---

---

---

---

## Database Dictionary (Cont.)

- ◆ ALL\_TABLES: Descriptions of tables
- ◆ ALL\_COL\_COMMENTS: Comments on columns of accessible tables
- ◆ ALL\_TAB\_COLUMNS: Lists columns contained in all tables
- ◆ ALL\_TAB\_COMMENTS: Comments on tables
- ◆ ALL\_USERS: Information on all users in database
- ◆ ALL\_VIEWS: Lists views accessible to user

Note: For complete reference, see Oracle Server Reference or select table\_name from dict in SQL\*Plus

26

---

---

---

---

---

---

---

---

---

---

## Data Dictionary Exercise 1.2.1

- ◆ Write a simple script to use as a tool to list information about SCT Banner Financial Aid tables, to include table\_name, table\_comment. Script should include ability to ask user for name of table and should accept partial input
- ◆ Save your script to a temporary directory on your computer for future use

27

---

---

---

---

---

---

---

---

---

---

## Referential integrity

- ◆ **Referential integrity:** A condition by which a dependent table's Foreign key must have either a null entry or a matching entry in the related table
- ◆ **Primary key:** A Candidate key selected as a unique entity identifier
- ◆ **Foreign key:** An attribute in one table whose values must match the Primary key in another table or whose values must be null

Note: Keys are implemented as constraints

28

---

---

---

---

---

---

---

---

---

---

## Referential integrity (Cont.)

- ◆ **The relationship between Parent and Child rows**
- ◆ **Child row has the Foreign key constraint and the Parent row has the Primary key constraint**
- ◆ **The Parent row cannot be deleted when a Child row exists**

29

---

---

---

---

---

---

---

---

---

---

## Primary key

- ◆ **Enforces unique, non-null keys**
- ◆ **PK\_ + Primary key table\_name**
  - **Defining PK\_STVTERM**
    - ◆ Alter Term Code Validation table (STVTERM)
    - ◆ Add constraint PK\_STVTERM
    - ◆ Primary key (column\_name)

30

---

---

---

---

---

---

---

---

---

---

## Foreign key

- ◆ A table can have any number of Foreign keys
- ◆ Usage
  - Referencing validation tables
  - Application hierarchy

---

---

---

---

---

---

---

---

31

## Foreign key: validation

### ◆ Validation tables

- "FK"+ lup# + "\_" + foreign\_table + "\_INV\_" + Primary\_table\_"CODE"
- FK3\_RPRATRM\_INV\_STVTERM\_CODE
  - ◆ Alter Applicant Award by Term table (RPRATRM)  
add constraint FK3\_RPRATRM\_INV\_STVTERM\_CODE  
Foreign key (sfrstcr\_code, )  
references STVTERM (STVTERM\_CODE, )

---

---

---

---

---

---

---

---

32

## Foreign key - hierarchy

### ◆ Application hierarchy

- "FK" + lup# + "\_" + foreign\_table + "\_INV\_" + primary\_table\_"KEY"
- FK1\_ROBUSDF\_INV\_RORSTAT\_KEY
  - ◆ Before one can create a record in ROBUSDF, it must exist in RORSTAT

---

---

---

---

---

---

---

---

33

### Referential Integrity Exercise 1.2.2 and 1.2.3

- ◆ Write a simple script to use as a tool to examine constraints placed on tables by SCT Banner designers.
- ◆ Save your work.
- ◆ Challenge script: Write a script that will show: constraint name, column name, position, type of constraint, constraint status and table name

---

---

---

---

---

---

---

---

---

34

### Data standardization

- ◆ IDs
- ◆ Names
- ◆ Addresses
- ◆ Dates

---

---

---

---

---

---

---

---

---

35

### Identification numbers

- ◆ Persons
  - Enter an identification number of your choice or use the identification number that the System generates on any identification form (SPAIDEN)
- ◆ Non-persons
  - Use the System-generated identification number

---

---

---

---

---

---

---

---

---

36

## Names

Prefix    First Name                      Last Name                      Suffix

**Mr. William                      Smith-Jones                      III**

- =====
- ◆ Upper/Lower Case                      Use conventional upper- and lower-case letters.
  - ◆ Last Names                      Do not enter spaces within prefixed last names:  
MacArthur                      O'Connor                      VanHusen  
St.John                      deBolt                      DuShen
  - ◆ First initials/middle name:                      Do not enter spaces within hyphenated last names:  
Smith-Jones                      Cochram-Ashley
  - ◆ Prefixes/Suffixes                      For names using a first initial and middle name,  
enter a period and one space between the initial  
and the name.  
Ex: R. Maureen Smith
  - ◆ Prefixes/Suffixes                      Use periods after prefixes and suffixes where  
applicable:  
Miss                      Mrs.                      Mr.                      Jr.

37

---

---

---

---

---

---

---

---

---

---

## Addresses

- ◆ Do not use a pound sign (#) when completing an Address field if your organization uses SCT Banner Letter Generation. SCT Banner Letter Generation recognizes a pound sign as a formatting command
- ◆ County codes are defined on the County Code Validation Form (STVCNTY)
- ◆ State and Province codes are defined on the State/Province Code Validation Form (STVSTAT). (State and Province names are entered into the State/Prov fields)

38

---

---

---

---

---

---

---

---

---

---

## Dates

- ◆ Installation Control Form (GUAINST) radio button
- ◆ Date formats (MDY, DMY, YMD)
  - MDY    November 6, 2005    is entered as 11/06/05
  - DMY    November 6, 2005    is entered as 06/11/05
  - YMD    November 6, 2005    is entered as 05/11/06
- ◆ 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 (/)
- ◆ For Job Submission dates, use DD-MON-YYYY format on the Process Submission Control Form (GJAPCTL)

39

---

---

---

---

---

---

---

---

---

---

## What is a PIDM?

- ◆ A "Primary Identification Master" uniquely identifies the records of one person or non-person entity in SCT Banner
- ◆ An internal key field used to identify and store records
- ◆ Assigned automatically when new records are created
- ◆ Generated from an Oracle sequence – PIDM\_SEQUENCE through function F\_GEN\_PIDM

40

---

---

---

---

---

---

---

---

## SOBSEQN

- ◆ A common overall base table used to store one-up numbers for various entities
- ◆ Should have maintenance access set at highest security level

41

---

---

---

---

---

---

---

---

## Contents of SOBSEQN

```
select * from sobseqn;
```

SOBSEQN_FUNCTION	S	SOBSEQN_MAXSEQNO	SOBSEQN_A
RECEIPT		203	22-FEB-05
ID	@	18	19-SEP-96
PIDM		571	07-JAN-05
ALUMNIGIFT		108	21-FEB-05
ALUMNIFLEDGE		44	15-FEB-05
EDIREQUESTID		1	25-APR-95
EDI_DCMT_SEQNO		1	08-DEC-95
ALUMNIDUES		3	15-FEB-05
ALUMNIRECEIPT		1	31-JAN-96
EVENT	A	4	15-FEB-05
HRREQ	R	1	23-NOV-04
PROSPECT_ID	P	0	04-JUN-03
HRENC096		10004	18-NOV-04
HRENC005		10003	17-FEB-05

42

---

---

---

---

---

---

---

---

## PIDMs and SOBSEQN

### ◆ How to get the next available PIDM:

```
SELECT pidm_sequence.currval
FROM dual;
SELECT pidm_sequence.nextval
FROM dual;
```

43

---

---

---

---

---

---

---

---

## General Person Exercises

---

---

---

---

---

---

---

---

## General Person Exercise 1.2.4

- ◆ Write a simple query to get ID, first name, middle name, last name, and the Change indicator for persons who have had changes made to their ID on the Identification Form (SPAIDEN) records.

45

---

---

---

---

---

---

---

---

### General Person Exercise 1 Answer

```
SELECT spriden_id,  
       spriden_first_name,  
       spriden_mi,  
       spriden_last_name,  
       spriden_change_ind  
FROM spriden  
WHERE spriden_change_ind = 'I'  
      AND spriden_entity_ind = 'P'
```

46

---

---

---

---

---

---

---

---

### General Person Exercise 1.2.5

♦ Write a simple query to get PIDM, ID, first name, middle initial, last name, and Social Security number for persons who have had changes made to their name records.

47

---

---

---

---

---

---

---

---

**Good Job!!!**

48

---

---

---

---

---

---

---

---



## Database Tool Set for SQL

Building scripts to look at database structure using  
SQL\*Plus

---

---

---

---

---

---

---

---

### Check\_col\_comments.sql

```
column comments format a30 word_wrap
select column_name,comments
       from all_col_comments
       where table_name like
             upper('&Table_Name%')
       order by column_name
/
```

---

---

---

---

---

---

---

---

50

### Check\_tab\_comments.sql

```
column comments format a30 word_wrap
select table_name, comments
       from all_tab_comments
       where table_name like upper('&Table_Name%')
             and table_name not like '%%%'
       order by table_name
/
```

---

---

---

---

---

---

---

---

51

## Check\_cons.sql

```
column constraint_name format a30 word_wrap
column column_name format a25
column ctype format a22 head 'Constraint Type'
  set linesize 130
select constraint_name,
       column_name,
       position,
       table_name
  from all_cons_columns
 where column_name like upper('&TableName%')
 order by table_name, position
/
```

52

---

---

---

---

---

---

---

---

---

---

## Check\_cons\_all.sql

```
column constraint_name format a30 word_wrap
column column_name format a25
column ctype format a22 head 'Constraint Type'
  set linesize 130
select all_cons_columns.constraint_name, all_cons_columns.column_name,
       all_cons_columns.position,
       decode(all_constraints.constraint_type,'R','Referential Integrity',
             'C','Not Null/Check','P','Primary Key','U','Unique',
             'V','With Check Option','O','With Read Only') ctype,
       all_constraints.status, all_cons_columns.table_name
  from all_cons_columns, all_constraints
 where all_cons_columns.column_name like upper('&TableName%')
       and all_cons_columns.constraint_name = all_constraints.constraint_name
       and all_cons_columns.table_name = all_constraints.table_name
 order by table_name, all_constraints.constraint_name, position
/
```

53

---

---

---

---

---

---

---

---

---

---

## Self-generating.sql

```
set head off
set echo off
/*
  At prompt enter name or partial name of a table.
  then enter part of a column name ie. code.
*/
spool on
select 'select distinct '||column_name||' from
       '||table_name||';'
  from dba_tab_columns
 where table_name like upper('&tables%')
       and column_name like upper('&column%')
/
spool off
pause Press any key to continue
edit on.lst
pause Don't forget to set head on and set echo on before running any of
these scripts. Press any key to continue.
```

54

---

---

---

---

---

---

---

---

---

---

## Review of Financial Aid Process

---

---

---

---

---

---

---

---

## Student Applies for Aid

- ◆ Free Application for Federal Student Aid (FAFSA)
  - Student competes on an ANNUAL Basis
  - Output from FAFSA is the Student Aid Report (SAR)
  - Student receives SAR in the mail
  - School receives SAR electronically (ISAR)
  - Expected Family Contribution (EFC)
- ◆ Institutional application
- ◆ Application for admission to the school

---

---

---

---

---

---

---

---

56

## Financial Aid Reviews Paperwork

Begin File processing:

- ◆ Verification
- ◆ Request additional documentation
- ◆ Tracking Letter
- ◆ Documents required for packaging or disbursement
- ◆ Paperwork completed – 'File Complete'
- ◆ SAP review

---

---

---

---

---

---

---

---

57

### **Packaging – Awarding of Aid**

- ◆ Federal Pell Grant determined by EFC and budget
- ◆ Campus-based aid (SEOG, Perkins Loan, and Federal Work Study) based on institutional packaging guidelines
- ◆ Scholarships – merit and need
- ◆ Loans – subsidized, unsubsidized and Plus

58

---

---

---

---

---

---

---

---

### **Types of Financial Aid**

- ◆ Scholarships
- ◆ Work-Study
- ◆ Loans (Subsidized, Unsubsidized, Parent, Perkins)
- ◆ Grants
  - Federal Pell Grant
  - Federal Supplemental Opportunity Grant (SEOG)
  - State Grants
  - Institutional Grants

59

---

---

---

---

---

---

---

---

### **Award Letter**

- ◆ Award letter is mailed to student
- ◆ Student accepts or declines aid
- ◆ Aid is accepted in the system
- ◆ Aid is then ready for memo, authorization, or payment

60

---

---

---

---

---

---

---

---

## Fee Assessment and Billing

- ◆ Students register for courses
- ◆ Students are billed for courses
- ◆ Aid is applied to student accounts

---

---

---

---

---

---

---

---

61

## Reporting

- ◆ Federal Pell Grant Report
- ◆ FISAP (Fiscal Operations Report)
- ◆ State Grant Reporting

---

---

---

---

---

---

---

---

62

## SCT Banner Financial Aid Database Structure

- ◆ We are going to review each area in SCT Banner Financial Aid and look at:
  - Major Forms
  - Major Tables
    - Description
    - Column Comments
    - Constraints/Relationships
  - Validation Forms and Tables
  - Process Flow
  - Processes and Procedures

---

---

---

---

---

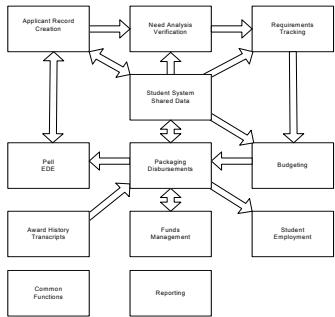
---

---

---

63

## FA System Process Flow



64

---

---

---

---

---

---

---

---

---

---

## Applicant Record Creation

- ◆ Created from several processes
  - Tape loads
    - ◆ MDE
    - ◆ Electronic Data Exchange
  - Manual – Major Forms
    - ◆ RRAAREQ
    - ◆ RNANAx

65

---

---

---

---

---

---

---

---

---

---

## Applicant Record Creation Forms

- Applicant Status Form
- Financial Aid Record Maintenance
- Need Analysis Form
- Need Analysis Processing Form
- Need Analysis Document Verification Form
- Need Analysis Results Form
- Applicant Requirements Form
- Applicant Budget Form
- Award Form
- Summary Academic History Inquiry Form
- Financial Aid Enrollment Form
- Basic Student Information Inquiry Form
- Admissions Application Inquiry Form
- Account Detail Review Inquiry Form
- Degrees and Other Formal Awards Inquiry Form
- Housing Information Inquiry Form
- Prior College Inquiry Form
- High School Inquiry Form
- Test Score Inquiry Form
- Veteran Certification Inquiry Form
- Third Party Contract Inquiry Form
- Billing Exemption Inquiry Form

- ROASTAT
- ROARMAN
- RNANAx
- RNAPRxx
- RNAVrx
- RNARSLT
- RRAAREQ
- RSIAAUD
- RPAAWRD
- RSIHST
- ROAENRL
- RSISTDN
- RSIAPPL
- RSIAREV
- RSIDIGR
- RSIHOS
- RSIPCOL
- RSIHSC
- RSITEST
- RSIVETN
- RSICONT
- RSIEXP

66

---

---

---

---

---

---

---

---

---

---

## Applicant Record Creation Data Load Procedures

- ◆ Loads electronic media from external sources
- ◆ Stored in temporary ORACLE tables
  - By processes RCBCTxx (CSS) or RCBTPxx (EDE)
  - Used by the RCPMTCH process
- ◆ RCRTPx process completes load

67

---

---

---

---

---

---

---

---

## Applicant Record Creation Data Load Procedures (Cont.)

- ◆ SCT Banner Financial Aid User Guide/Processing/Chapter 2 contains procedures. Let's review them
- ◆ Check the latest release guides for the most up-to-date information

68

---

---

---

---

---

---

---

---

## Data Management

- ▣ **Data Management**
- ▣ Financial Aid Suspended Records Maintenance
- ▣ Interface Data Translation Rules
- ▣ Name Translation Rules
- ▣ Common Matching Rules
- ▣ Common Matching Source Rules
- ▣ Data Source Code Validation
- ▣ Common Matching Source Code Validation

69

---

---

---

---

---

---

---

---

### Preparing for the Data Load Process

- ◆ Set up the Data Source Code Validation Form (RTVINFC)
- ◆ Build the Interface Data Translation Rules Form (RCRTPTR)
- ◆ Enter values and rules on Common Matching Forms
- ◆ Build the Institutional Financial Aid Options Form (ROAINST)
- ◆ Build the Data Source Rules Form (RCRDTSR)

---

---

---

---

---

---

---

---

70

### Preparing for the Data Load Process

(cont.)

- ◆ Move data to the \$DATA\_HOME/finaid directory for UNIX or DATA\$HOME for VMS
  - Must have write privileges in directory
  - Rename file to:
    - ◆ xxyyesar.tap (EDE ISIR diskette or download),
    - ◆ xxyyesat.tap (EDE ISIR tape),
    - ◆ xxyycssd.tap (CSS diskette or download), or
    - ◆ xxyycsst.tap (CSS tape)

---

---

---

---

---

---

---

---

71

### Running the Data Load Process

- ◆ Run RCBTPxx (Part 1 of Data Load for federal data) or
- ◆ Run RCBCTxx (Part 1 of Data Load for CSS PROFILE data only)
- ◆ Run RCPMTCH (Part 2 of Data Load)
- ◆ Run RCRTPxx (Part 3 of Data Load)
  - It is important to run the RCPMTCH and RCRTPxx steps as close together as possible. This decreases the chances of someone adding new students online, which would make the New status (as determined by RCPMTCH) incorrect.
- ◆ Review suspended records on RCRSUSP
- ◆ Run the RCPDTMP Process

---

---

---

---

---

---

---

---

72



### Applicant Record Creation Exercises 2.1.1 and 2.1.2

- ◆ Other than data load, how can new records be created?
- ◆ Explain what the process RCPDTMP does.

---

---

---

---

---

---

---

---

---

---

73

### Need Analysis/Verification Forms

- [-] Need Analysis [FRESNEED]
  - [-] 2004-2005 Need Analysis [FRESNEED05]
    - [+] 2004-2005 Need Analysis [RNANAD05]
    - [+] 2004-2005 Need Analysis Processing [RNAPR05]
    - [+] 2004-2005 Supplemental Need Analysis [RNASUR05]
    - [+] Need Analysis Results [RNARSLT]
    - [+] 2004-2005 Applicant Override [RNAOV05]
    - [+] Applicant Need Analysis Application Inquiry [RNAPP1]
    - [+] Applicant Student Loan Data Inquiry [RNINSLD]
    - [+] Calculated Need Analysis Detail Inquiry [RNINAIO]
    - [+] 2004-2005 Miscellaneous Results Inquiry [RNMSRS]
    - [+] 2004-2005 INAS Global Policy Options Rules [RNIRGL05]
    - [+] Applicant Status [RDASTAT]
    - [+] Applicant Full Grant [RDAPFL1]
    - [+] 2004-2005 Need Analysis Document Verification [RNADV05]
    - [+] Application Purge [RNAPUR05]
    - [+] 2004-2005 Student Loan Data [RNASLD05]
    - [+] NSLDS Transfer Monitoring Application [RNATMNT]

---

---

---

---

---

---

---

---

---

---

74

### Need Analysis/Verification Tables

INAS Global Policy Options Table	RNRGLBL
NSLDS Transfer Monitoring Process Table	RNRTMNT
Applicant Need Analysis Override Table	RNROVRD
Needs Analysis Verification Table	RNRVRFY

---

---

---

---

---

---

---

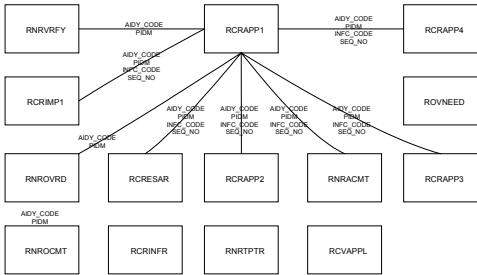
---

---

---

75

### Need Analysis/Verification Table Relationships



Reference: SCT Banner Financial Aid Supplement, page 2-6 (60 of 156) September 1999.

76

---

---

---

---

---

---

---

---

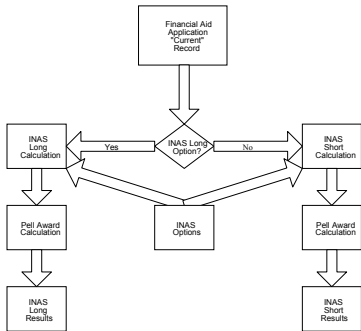
---

---

---

---

### Need Analysis Online Recalculation Process Flow



77

---

---

---

---

---

---

---

---

---

---

---

---

### Need Analysis/Verification Processes and Procedures 2



78

---

---

---

---

---

---

---

---

---

---

---

---

## Need Analysis/Verification Exercise

- ◆ Write a script that selects Dependent students selected for Verification. Include student ID, aid year, verification and dependency. Prompt for Aid Year.
- ◆ Hints
  - Dependency = RCRAPP2\_MODEL\_CDE
  - Selected for Verification = RCRAPP1\_VERIFICATION\_MSG
  - Be sure to check for Current record
  - Be sure to compare SEQ\_NOs and INFC\_CODEs

---

---

---

---

---

---

---

---

79

## Requirements Tracking Forms

- Requirements Tracking [RESTRACK]
- Applicant Requirements [RRAREQ]
- Financial Aid Record Maintenance [RQARIMAN]
- Applicant Requirements Mass Entry [RRAMASS]
- Applicant Comments [RHACOMM]
- Applicant Immediate Process [RQAIMMP]
- Requirements Tracking Control [RESTRACKCM]

---

---

---

---

---

---

---

---

80

## Requirements Tracking Tables

Tracking Group/Requirements Rules	RRRGREQ
Applicant Requirements	RRRAREQ
Tracking Requirement Message Rules	RRRTMSG
Requirements View	RRVAREQ
Object Access F/A Requirements View	RRVREQ1

---

---

---

---

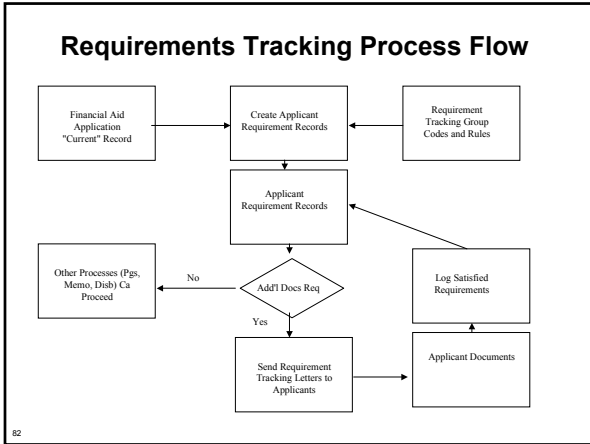
---

---

---

---

81




---

---

---

---

---

---

---

---

- ### Requirements Tracking Processes and Procedures
- RTVTREQ – Define tracking requirement codes
  - RTVTRST – Define tracking requirement status codes
  - RTVTGRP – Define tracking groups
  - RRRGREQ – Develop tracking group requirements for each group. Indicate which documents are required for each specific tracking group
  - RORRULE – Develop tracking group rules associated with each group
  - RRRMSG – Develop specific messages for each tracking code, if desired
- 83

---

---

---

---

---

---

---

---

- ### Requirements Tracking Processes and Procedures (Cont.)
- ◆ ROBBGRP (GJAPCTL form)
    - Performs Budget, Tracking and/or Packaging group assignment in batch mode for all applicants or a selected population of applicants
  - ◆ ROAUTO (ROAIMMP form)
    - Performs Online Budget, Tracking and Packaging group assignment for a specified applicant
- 84

---

---

---

---

---

---

---

---

## **ROBBGRP**

- ◆ The Batch Group Assignment process assigns applicants to Budget, Packaging and/or Requirements Tracking groups by using the appropriate group selection criteria

85

---

---

---

---

---

---

---

---

## **ROAIMMP and ROOAUTO**

- ◆ The Online Group Assignment process attempts to assign an individual applicant to a Budgeting, Tracking, or Packaging group

86

---

---

---

---

---

---

---

---

## **RORRULEs**

- ◆ Major Form RORRULE
- ◆ Compiled by ROOQSOL
  - The Online Group Assignment Selection Criteria Generation process creates a SQL statement from the Budgeting, Tracking and Packaging Group Selection rules created by the Financial Aid Selection Rules Form (RORRULE)

87

---

---

---

---

---

---

---

---

### RORRULE Exercise 3.2.2

- ◆ Write a rule to capture in a tracking group those individuals that have a 'C' code on their current record.
  - You will need to create a tracking group and tracking code
  - To complete this assignment, you should be able to run ROBBGRP and place a requirement on student records that fit the above category

---

---

---

---

---

---

---

---

88

### Requirements Tracking Exercise 1

- ◆ Write a script that selects students whose tracking requirements are not satisfied. Include Student ID, Aid year, Requirement Code, Status and Satisfied Indicator. Prompt for Aid year.
- ◆ Modify your script to look for one particular requirement.

---

---

---

---

---

---

---

---

89

### Requirements Tracking Exercise 2

- ◆ Write a script that will find all students who have a one time tracking document. Also, include a description for the tracking document code.

---

---

---

---

---

---

---

---

90

## Student System Shared Data

- ▣ Student System Shared Data [RSESSDAD]
- ▣ Summary Academic History Inquiry [RSIHST]
- ▣ Basic Student Information Inquiry [RSISTDN]
- ▣ Admissions Application Inquiry [RSIAPPL]
- ▣ Account Detail Review Inquiry [RSIAREV]
- ▣ Degrees and Other Formal Awards Inquiry [RSIDEGR]
- ▣ Housing Information Inquiry [RSIHOU5]
- ▣ Prior College Inquiry [RSIPCOL]
- ▣ High School Inquiry [RSIHSCH]
- ▣ Test Score Inquiry [RSITEST]
- ▣ Veteran Certification Inquiry [RSVETCN]
- ▣ Third Party Contract Inquiry [RSSCONT]
- ▣ Billing Exemption Inquiry [RSIEXP1]

91

---

---

---

---

---

---

---

---

---

---

## Student System Shared Data Tables and Views

- ◆ RORENRL - Enrollment Status  
Financial Aid Captured data
- ◆ ROVSTDN - Basic Student Information
- ◆ ROVADAP - Admissions selection
- ◆ ROVADxx - Admission information
- ◆ ROVDGMR - Degree information

92

---

---

---

---

---

---

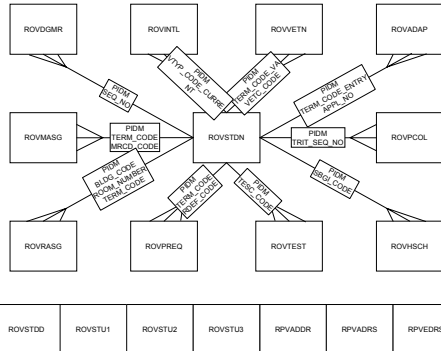
---

---

---

---

## Student System Shared Data



93

---

---

---

---

---

---

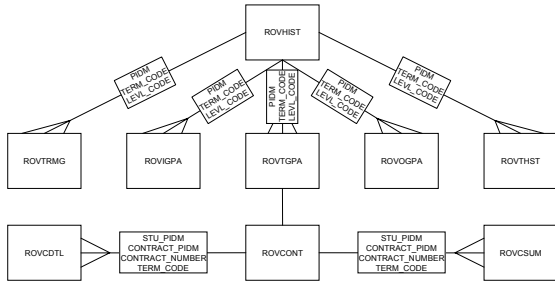
---

---

---

---

### Student System Shared Data 2



94

---

---

---

---

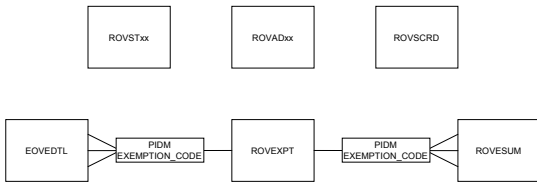
---

---

---

---

### Student System Shared Data 3



95

---

---

---

---

---

---

---

---

### Student System Shared Data Processes and Procedures

- ◆ Financial Aid Enrollment Hours Process (RSRENRL)
- ◆ Add credit and billing hours from student enrollment to Financial Aid
- ◆ C process, used to 'capture' or 'freeze' hours

96

---

---

---

---

---

---

---

---



## Pell Electronic Data Exchange

- Electronic Data Exchange [RESEDE]
  - 2004-2005 Electronic Data Exchange [RESEDE05]
    - ISIR Corrections Request [REACORR]
    - Batch Control [REBATCH]
    - COD Document Control [RECODD]
    - EDE Directory [READRCT]
    - 2004-2005 Miscellaneous Results Inquiry [RHMISRS]
    - 2004-2005 Pell Origination/Acknowledgement [REACORS]
    - 2004-2005 Pell Disbursement/Acknowledgement [READRS]
    - Rejection Code Validation [RTRVRC]
    - Summary Statement of Account Inquiry [REISSOA]
    - Detail Statement of Account Inquiry [REIDSOA]
    - Data Request Record [REBRDR]
    - Multiple Reporting Record Inquiry [REMRIR]
    - Year to Date Pell Summary Inquiry [REYDTS]
    - NSLDS Transfer Monitoring Application [RMA1MNT]
    - COD Student Identifier [REASTID]

87

---

---

---

---

---

---

---

---

---

---

## Pell Electronic Data Exchange

COD Document Control Table	RERCODD
COD Identifier Table	RERSTID
EDE Directory	REBDRCT
Applicant Payment Voucher	REBPAYV

88

---

---

---

---

---

---

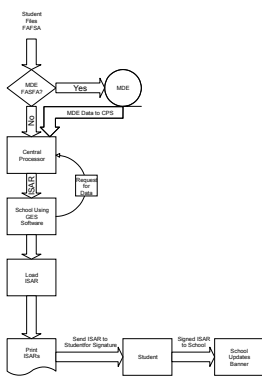
---

---

---

---

## ISIR Data Receipt Process Flow



89

---

---

---

---

---

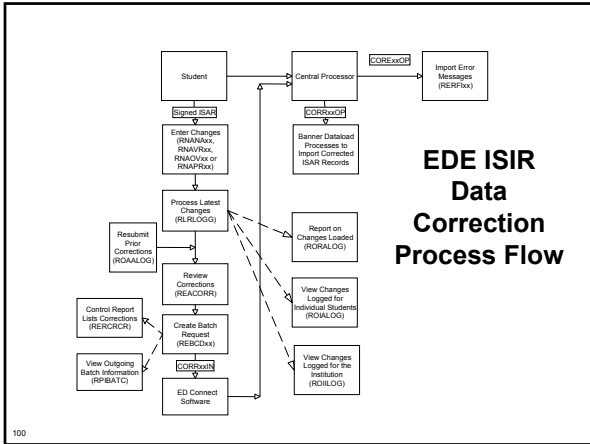
---

---

---

---

---




---

---

---

---

---

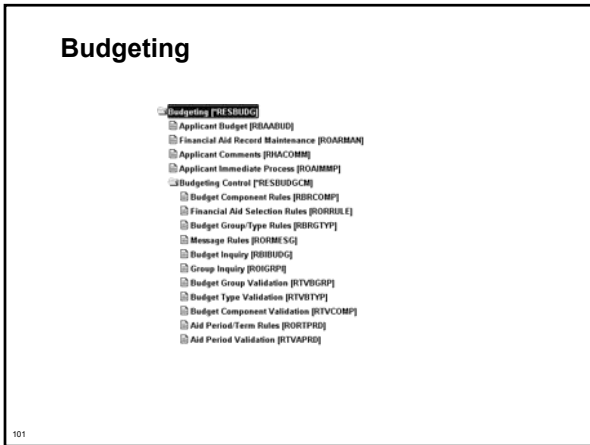
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

### Budgeting Tables

Applicant Budget Table	RBBABUD
Applicant Budget Component Table	RBRACMP

102

---

---

---

---

---

---

---

---

---

---

## Budgeting Processes & Procedures

- ◆ There are none.
- ◆ Two reports:
  - Applicant Budget Report (RBAABUD)
  - Budget Component Report (RBRBCMP)

---

---

---

---

---

---

---

---

103

## Budgeting Exercises

- ◆ Write a script that will list student ID, aid year, budget type code for aid year 9596.
- ◆ Modify your script to include the budget component code and amount.

---

---

---

---

---

---

---

---

104

## Funds Management

- ◆ Funds Management [RFSFUND]
- ▣ Fund Management [RFSMGMT]
- ▣ Funds Management [RQAMGMT]
- ▣ Fund Budget Inquiry [RFBUDG]
- ▣ Federal Fund ID Inquiry [RFFID]
- ▣ Federal Rules Inquiry [RPFEDS]
- ▣ Default Award and Disbursement Schedule Rules [RDEFSA]
- ▣ Fund Award and Disbursement Schedule Rules [RFRASCH]
- ▣ Financial Aid Selection Rules [RORRULE]
- ▣ Fund Base Data [RFRBASE]
- ▣ Fund Source Code Validation [RIVFSRC]
- ▣ Financial Aid Fund Type Validation [RIVFTYP]

---

---

---

---

---

---

---

---

105

## Funds Management Major Tables

Fund Base Data Table	RFRBASE
Fund/Budget Component Table	RFRBCMP
Federal Fund ID Table	RFRFFID

106

---

---

---

---

---

---

---

---

## Funds Management Processes & Procedures

- ◆ There are no processes in Funds Management other than procedures for the functional staff to follow. They may need assistance with RORRULES.

107

---

---

---

---

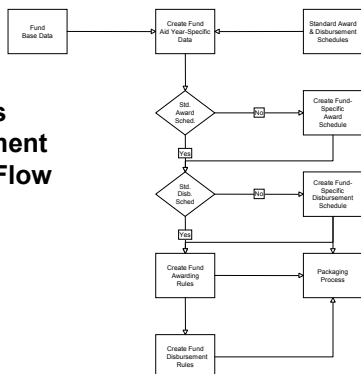
---

---

---

---

## Funds Management Process Flow



108

---

---

---

---

---

---

---

---

## Funds Management Exercises

- ◆ Write SQL that will list fund code, fund title and detail code of all existing funds.
- ◆ Write SQL that will list all Title IV funds in the Fund Base Data Table

---

---

---

---

---

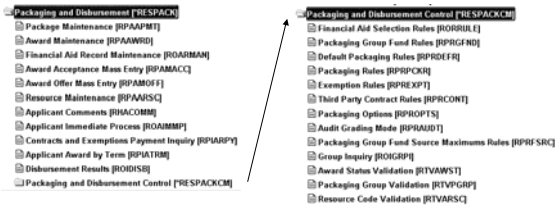
---

---

---

109

## Packaging and Disbursement



---

---

---

---

---

---

---

---

110

## Packaging and Disbursement Major Tables

- ◆ RPROWRD - Applicant Award Table
- ◆ RPRATRM - Applicant Award by Term Table
- ◆ RPRADSB - Applicant Disbursement Table
- ◆ RPRLAPP - Loan Application Table

---

---

---

---

---

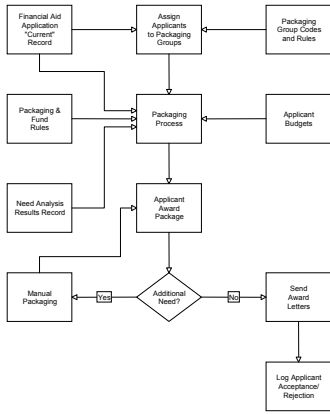
---

---

---

111

## Packaging Process Flow



112

---

---

---

---

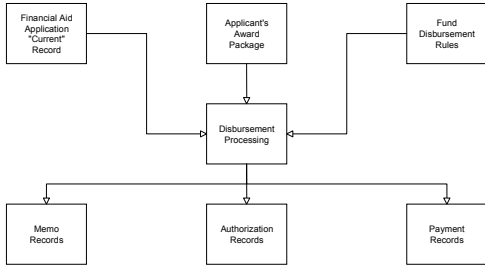
---

---

---

---

## Disbursement Process Flow



113

---

---

---

---

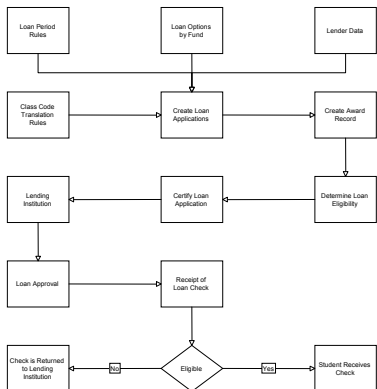
---

---

---

---

## Loan Process Flow



114

---

---

---

---

---

---

---

---

**Packaging and Disbursement  
Pell Grant Process**

- ◆ Pell Grant Calculation Process (RPEPELL)
  - Online, batch or from database trigger
  - Triggers when certain data changes on forms:
    - RNARSLT, RBAABUD, RNASUxx, RNANAx and ROAPELL

---

---

---

---

---

---

---

---

115

**Packaging and Disbursement  
Packaging Process**

- ◆ Packaging Process (RPEPCKG)
  - Actual or Simulation
  - RPBPDV - Creates reports
    - ◆ RPBWARD.out – The Award Report that results from the packaging run
    - ◆ RPBUND.out – The Fund Report that results from the packaging run

---

---

---

---

---

---

---

---

116

**Packaging and Disbursement Online Award  
Validation Process**

- ◆ Online Award Validation Process (RPOVLDT )
  - Validates all awards within an applicant's package to ensure that no changes made to the applicant or the fund(s) invalidate the award(s)
  - Ascertains whether the applicant is eligible for an individual fund being added to the applicant's package either on-line or during the Packaging process
  - Validates that the applicant is eligible for system or manual disbursements from the awards within the applicant's package

---

---

---

---

---

---

---

---

117

### Packaging and Disbursement Batch Validation Process

- ◆ **Batch Validation Process (RPBVLDT)**
  - Validates all awards within an applicant's package to ensure that no changes made to the applicant or the fund(s) invalidate the award(s)
  - Ascertains whether the applicant is eligible for an individual fund being added to the applicant's package either on-line or during the Packaging process
  - Validates that the applicant is eligible for system or manual disbursements from the awards within the applicant's package

118

---

---

---

---

---

---

---

---

### Packaging and Disbursement Disbursement Process

- ◆ **Disbursement Process (RPEDISB) online and batch**
  - Passes the amount of deferred Financial Aid available to an applicant within a specific term to the BANNER Student system. This is displayed as a memo transaction in the Student Billing module.
  - Passes the amount of Financial Aid available to be disbursed to an applicant within a specific term to the BANNER Student system. This is displayed as an authorization in the Student Billing module.
  - Passes the amount of Financial Aid scheduled to be disbursed to an applicant within a specific term to the BANNER Student system. All scheduled disbursements with a date less than or equal to the processing date that have not been disbursed will be processed. These are displayed as payment transactions in the Student Billing module.

119

---

---

---

---

---

---

---

---

### Packaging and Disbursement Exercise 3.2.3

- ◆ **Write a script that will list the student IDs, aid year, fund code, and Paid Award Amount for students where aid year is entered as a parameter. We want students with paid amounts.**

120

---

---

---

---

---

---

---

---



### Packaging and Disbursement Exercise 3.2.4

- ◆ Write a script to list the student IDs, aid year, term code, fund description, and offered award amount with term code are entered as a parameter.
  - Allow for the use of wild card entry for term code

---

---

---

---

---

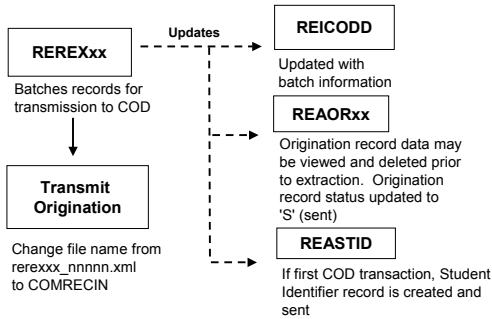
---

---

---

121

### COD Extract – Pell and DL Award and Disbursement Originations



---

---

---

---

---

---

---

---

122

### COD Extract - Originations

#### COD Extract Process (REREXxx)

- ◆ Process extracts
  - Origination Data
  - Student Identifier Record
- ◆ Process updates
  - Origination Record (REAORxx)
  - Student Identifier Record (REASTID)
  - COD Document Control (REICODD)

---

---

---

---

---

---

---

---

123

## COD Extract - Originations

### Send Origination File to COD

- ◆ Access FTP software
- ◆ Move rere04\_nnnnn.xml file from jobsub to a local PC
- ◆ Rename rere04\_nnnnn.xml file to COMRECIN
- ◆ Upload COMRECIN file to COD website

124

---

---

---

---

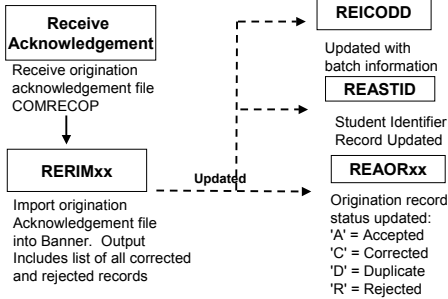
---

---

---

---

## COD Import – Pell and DL Responses



125

---

---

---

---

---

---

---

---

## COD Import - Responses

### COD File Import Process (RERIMxx)

- ◆ Loads the acknowledgement files received from COD into SCT Banner
- ◆ FTP the COMRECOP file into the dataload directory
- ◆ COD File Import Process updates
  - Origination Record (REAORxx)
  - Student Identifier Record (REASTID)
  - COD Document Control (REICODD)

126

---

---

---

---

---

---

---

---

## Data Request - Extract

- ◆ It is better to only load the MRR records the COD sends to you and not request MRR records
  - It is highly recommended that discretion be used when requesting MRR records
    - ◆ For example, never request all Origination Records for all your students or all institutions
    - ◆ There are many schools that originate for every ISIR that they receive
    - ◆ If you request records in this way, you may receive thousands of records

127

---

---

---

---

---

---

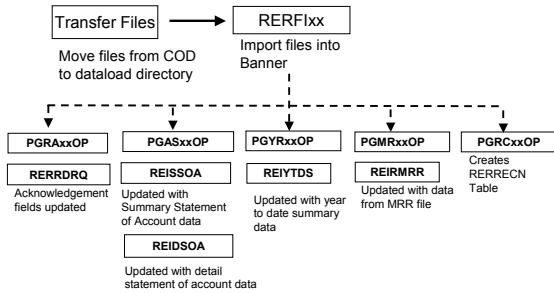
---

---

---

---

## Data Request - Import



128

---

---

---

---

---

---

---

---

---

---

## Data Request - Import

- ◆ RFMS Import Process (RERFIxx) loads the following records into Banner:
  - Data Request Acknowledgement (PGRxOP)
  - Year-to-Date Records (PGYRxxOP)
  - Statements of Account (PGASxOP)
  - Multiple Reporting Records (PGMRxOP)
  - Reconciliation Files (PGRCxxOP)

129

---

---

---

---

---

---

---

---

---

---

## Reports and Inquiries

### Pell RFMS Miscellaneous Report (RERPELL)

- ◆ Used to help identify exception-processing requirements
- ◆ Available data include:
  - All MRR records
  - Year-to-Date reporting
  - Just in Time, no disbursement
  - Just in Time, no charge
  - More....

130

---

---

---

---

---

---

---

---

## Reports and Inquiries

- ◆ Review multiple reporting record data (REIRMRR)
  - Used to view a student's multiple reporting record status and to identify any open issues for the student

131

---

---

---

---

---

---

---

---

## Reports and Inquiries

- ◆ Summary Statement of Account Inquiry Form (REISSOA)
  - Accesses the summary statement of the Electronic Statement of Account (ESOA) data online

132

---

---

---

---

---

---

---

---

## Reports and Inquiries

- ◆ Detail Statement of Account Inquiry Form (REIDSOA)
  - Accesses the detail statement of the Electronic Statement of Account (ESOA) data online

---

---

---

---

---

---

---

---

---

---

133

## Common Functions

- ▢ Financial Aid Common Functions [FRESCOMM]
  - ▢ General Person Identification [SPAIDEM]
  - ▢ Person Name/ID Search [ROAIDEN]
  - ▢ Aid Year Inquiry [ROAIDY]
  - ▢ Logging Activity Inquiry [ROLOGA]
  - ▢ Data Log Inquiry [ROHLOG]
  - ▢ SAR ID Inquiry [ROISARQ]
  - ▢ FISAP Person Maintenance [ROFSAPI]
  - ▢ Data Management [RESIDATA]
  - ▢ Financial Aid Common Functions Control [FRESCOMMCM]
  - ▢ QuickFlow Sequence [FRESCOMMCA]
  - ▢ Applicant Common Functions [RESAPPLCT]
  - ▢ Applicant Data Log Inquiry [ROIALOG]
  - ▢ Applicant Data Log Application [ROAALOG]
  - ▢ Saved Output Review [RUBREVQ]
  - ▢ Group Inquiry [ROGRPI]
  - ▢ Message Rules [ROHMSG]

---

---

---

---

---

---

---

---

---

---

134

## Common Functions Major Tables 1

Applicant Status Table	RORSTAT	Satisfactory Academic Progress Rules Table	RORSAPR
PELL Grant Payment Schedule Table	RORPELL	Financial Aid Messages	RORMESG
Applicant Holds Table	RORHOLD	Financial Aid Enrollment Table	ROENRL

---

---

---

---

---

---

---

---

---

---

135

## Common Functions Major Tables 2

Audit Log Form/Data Code Table	RORDVAL	Credit Hours Table	RORCRHR
Audit Log Table	ROBALOG	Financial Aid Institution Table	ROBINST
Applicant User-Defined Data Table	ROBUSDF	Generated SQL Selection Identifier Table	RORGSQL
Selection Identifier Compiled Statements	RORCMPL		

136

---

---

---

---

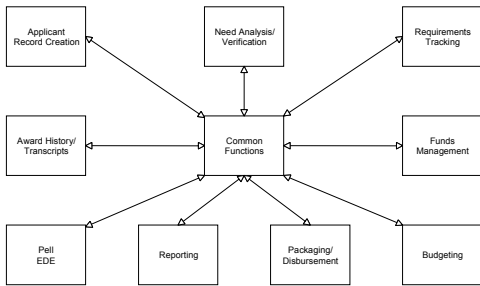
---

---

---

---

## Common Functions Process Flow



137

---

---

---

---

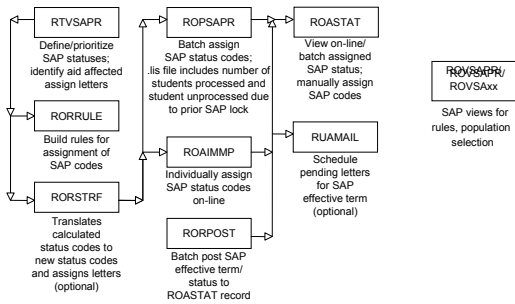
---

---

---

---

## Satisfactory Academic Progress Process



138

---

---

---

---

---

---

---

---

## SAP History Exercise 3.2.5

- ◆ Write a script to select ID, name, progress code, and term for each student coded in Banner for Satisfactory Academic Progress.
  - ◆ Some students are coded for more than one term
  - ◆ If students have more than one term, select the most current term and select that term's progress code
  - ◆ Hint: The academic progress table is RORSAPR

---

---

---

---

---

---

---

---

---

---

139

## Reporting

### Chapter 3 Reports and Processes

#### Financial Aid Report/Process Descriptions

Report	Name	Description
RBRADUD	Applicant Budget Report	Print applicants' budgets by budget type and budget assignment.
RBRCHMP	Budget Component Report	List budget component amounts by budget group, budget type and aid period.
RCRCDSX	Financial Aid CBS Data Load, Part 1	This data load process loads financial aid CBS PROFILE applicant records into the BANNER Financial Aid System. Only those institutions that receive the CBS PROFILE Data Load Process from SCT will receive the RCRCDSX program.
RCRTPes	Financial Aid Data Load, Part 1	This data load process loads financial aid applicant records into the BANNER Financial Aid System.
RRCUMPF	Data Load Comparison Process	Compares incoming applicant data from tape to existing applicant data on database.
RCRTPes	Financial Aid Data Load, Part 3	Loads new and matching applicant records to the permanent tables.
RFBF01W	FISAP Conversion (FISAP Worksheet)	Uses FTFP conversion and FISAP conversion tables.

---

---

---

---

---

---

---

---

---

---

140

## Commonly Used Reports

- ◆ Financial Aid Roll Rule Report (ROPROLL)
  - Create the scripts that rolls year rules
  - by aid year (Group, Fund, Budgeting, Tracking, Packaging)
- ◆ FISAP Report (RORFSxx)
  - Prints FISAP report
- ◆ Others you need to know!
  - Disbursement Process (RPEDISB)
  - Pell Awarding Process (RPEPELL)
  - Packaging Process (RPEPCKG)
- ◆ RNEINxx
  - INAS

---

---

---

---

---

---

---

---

---

---

141

## APIs

### ◆ Overview

- Application Programming Interfaces (API's) enhance processing and simplify code
- API definition - a central program that inserts, updates, and deletes data
- APIs execute and validate business rules before inserts, updates, or deletes

---

---

---

---

---

---

---

---

142

## APIs Used in Banner Financial Aid

### ◆ PELL ORIGATION

- Used during REAOR05 process to insert, update, delete information into REBPAYV table

### ◆ AWARD DISBURSEMENT

- Used from RPAAWRD form to insert, update, delete information into RPRADSB table

### ◆ DIRECT LOAN SERVICER-REFUND

- Used from RPRDU04 form to insert Direct Loan Servicer Refund record into RFRRFND table

---

---

---

---

---

---

---

---

143

## APIs Used in Banner Financial Aid (cont.)

### ◆ COD DOCUMENT

- Used by RREX05/RERIM05 processes to insert COD document record into RERCODD table

### ◆ COD IDENTIFIER

- Used by RREX05 process to create and update COD student and borrower identifier records

---

---

---

---

---

---

---

---

144



## APIs Used in Financial Aid

Table	Form/Process	API Object Name	API Entry Name	Task Performed
REPAYV	REACRDS	rb_pell_origination	PELL ORIGINATOR	Inserts, updates, and deletes information into the REPAYV table
RFRADS	RPAWED	rb_award_disbursement	AWARD DISBURSEMENT	Inserts, updates, and deletes information into the RFRADS table
RFRFND	RFRDU04	rb_dl_refund	DIRECT LOAN SERVICE-REFUND	Inserts a Direct Loan Service Refund record into the RFRFND table
RERCOOD	REREX05/ RERIM05	rb_cod_document	COO DOCUMENT	Inserts a COO document record into the RERCOOD table
RERSTD	REREX05	rb_cod_identifier	COO IDENTIFIER	Creates and updates COO student and borrower identifier records

145

---

---

---

---

---

---

---

---

---

---

## Award History/Transcripts

- [-] **History and Transcripts [HESHIS]**
  - [+] NSLOS Transfer Monitoring Application [RNTMNT]
  - [+] Transcript Rules [RHTRNS]
  - [+] Pre Banner Award Summary [RHPBAW]
  - [+] Prior Institution Transcript [RHPITRN]
  - [+] Fund Sequence History Inquiry [RHSFSH]
  - [+] Applicant Comments [RHACOMM]
  - [+] Transcript Information [RHATINF]
  - [+] Financial Aid Transcript Request [RHAROST]

146

---

---

---

---

---

---

---

---

---

---

## Student Employment

- [-] **Student Employment [RESEMP]**
  - [+] Student Employment Authorization [RJAEAR]
  - [+] Student Employment Referral [RJAERF]
  - [+] Student Employment Mass Entry [RJAEME]
  - [+] Student Employment Work History [RJAEMH]
  - [-] **Student Employment Control [RESEMP/CM]**
    - [+] Student Employment Default Rules [RJAESDR]
    - [+] Payroll Load Control [RJAEPAL]
    - [+] Placement Rules [RJAPLRL]
    - [+] Placement Base Data [RJAPLBD]
    - [+] Job Title Base Data [RJAJOBTD]
    - [+] Job Title Requirements [RJAJREQ]
    - [+] Referral Status Validation [RJAJSST]
    - [+] Employment Authorization Status Validation [RJAJSST]

147

---

---

---

---

---

---

---

---

---

---

## Student Employment Major Tables

Placement Base Data Table	RJBPLBD
Job Title Table	RJBJOBT
Student Employment Authorization Table	RJRSEAR
Student Employment Referral Table	RJRSERF
Student Employment Work History Table	RJRSEWH

148

---

---

---

---

---

---

---

---

## Student Employment Process & Procedures

- ◆ Payroll Load Process (RJRLOAD)
  - Interface payroll data from SCT Banner Human Resources to SCT Banner Financial Aid

149

---

---

---

---

---

---

---

---

## Student Employment Exercise

- ◆ Write a script to list student ID, aid year and job code.
- ◆ Modify the script to include the job description

150

---

---

---

---

---

---

---

---

### Database Structure Exercise 3.2.6

- ◆ Write a script to return those students that have different EFCs.
  - The script should return student ID, formatted name, EFCs and indicate if record has a 'lock'
  - Prompt the user for aid year

---

---

---

---

---

---

---

---

---

---

151

### Database Structure Advanced Exercise 3.2.7

- ◆ Write a script to select ID, name, unmet need, EFC, FWS work preference, and total paid awards for each student. Use wild card selection for aid year
- ◆ Modify your script to select only students that indicated they wanted FWS for 1998-99
- ◆ Modify your script to select students where they have any awards paid for aid year

---

---

---

---

---

---

---

---

---

---

152

### Logging

- ◆ Institution Financial Aid Options Form (ROAINST) (audit logging)
  - Budget logging
  - Needs Analysis logging
  - Packaging logging
  - Pell Correction logging
- ◆ Audit Log table (ROBALOG)

---

---

---

---

---

---

---

---

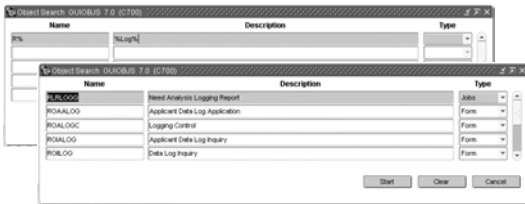
---

---

153

## Financial Aid Logging (Cont.)

◆ Let's review logging in the Banner Application.



---

---

---

---

---

---

---

---

154

## New Year Start

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

### Objectives

- ◆ To review all of the steps required to roll selected aid year specific information from one aid year to the new aid year, including:
  - Creation of an 0506 ROAINST instance
  - Running the ROPROLL process from Job Submission
  - Manually reviewing and editing rules and validation tables FOLLOWING the roll

---

---

---

---

---

---

---

---

156

## New Year Roll – 2005-2006

### Important Information

- ◆ BEFORE your IT Department can possibly install the appropriate Banner New Aid Year release, you MUST first create an 0506 instance of ROAINST

---

---

---

---

---

---

---

---

157

## New Year Roll – 2005-2006

### Three Very Important Considerations:

- ◆ 1) Do NOT copy or manually enter rules for a module on the RORRULE form for 0506 before running the ROPROLL process
- ◆ 2) Do NOT copy or manually enter 0506 data for a module that you wish to automatically roll to the 0506 aid year prior to running ROPROLL
- ◆ 3) Wait to edit and/or manually update the 0506 year-specific data you want to automatically roll until AFTER the ROPROLL process has been performed

---

---

---

---

---

---

---

---

158

## New Year Roll – 2005-2006

### RORPOLL Decisions – What data do you want to roll automatically?

	<u>Parameter</u>	<u>Value</u>
01	From Aid Year Code	ex. 0405
02	To Aid Year Code	ex. 0506
03	Perform roll of Group Rules	Y or N
04	Perform roll of FUND data	Y or N
05	Perform roll of BUDGET data	Y or N
06	Perform roll of TRACKING data	Y or N
07	Perform roll of PACKAGING data	Y or N
08	Perform roll of COMMON data	Y or N
09	Perform roll of EMPLOYMENT data	Y or N

---

---

---

---

---

---

---

---

159

## New Year Roll – 2005-2006

---

---

---

---

---

---

---

---

---

---

160

## New Year Roll – 2005-2006

### ◆ Common functions forms and data

#### - ROAINST

- RTVTREQ                      - RTVPGRP
- RTVTRST                     - STVLEVEL
- STVTERM                     - RORDATA
- RTVAPRD                    - RORTPRD
- RTVSAPR                    - RTVUSER
- RTVTGRP                    - RORPOST
- RTVBGRP                    - RCRINFR

---

---

---

---

---

---

---

---

---

---

161

## New Year Roll – 2005-2006

### ◆ Record creation module

- ROAINST
- RCRINFR - Create Data Source Matching Rules
- RTVINFC - Update Data Source Code
- RCRTPTR – Review for accuracy
- RPROPTS – Review Packaging and Disbursement options

---

---

---

---

---

---

---

---

---

---

162

## New Year Roll – 2005-2006

- ◆ Need Analysis module
  - RPROPTS – Check Packaging Fields for New Aid Year
  - ROAINST – Check for completion of 0506 Institutional Parameters
  - RORTPRD – Update with 0506 Aid Periods and Terms
  - RFRBASE – Review and Update Funds for 0506
  - RFRMGMT – Review Funds Management Forms for 0506
  - RPRCLSS – Review STVLEVL, STVCLAS and Banner Financial Aid Codes

163

---

---

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

- ◆ Tracking validation tables and rules
  - ROAINST – Review 0506 Tracking Parameters
  - RTVTGRP – Review Tracking Group Validation Form
  - RTVTREQ – Review and Edit Requirements Tracking Validation Form for 0506
  - RRRGREQ – Update 0506 Tracking Group/Requirements and Rules
  - RORRULE – Create Group Assignment Rules for 0506 Using the Copy Function
  - RRRMSG – Associate any new messages for 0506 to Tracking Requirement Codes

164

---

---

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

- ◆ Budgeting validation tables and rules
  - ROAINST – Create 0506 Budget Defaults
  - RTVAPRD – Confirm Aid Periods for 0506
  - RTVBTP – Confirm 0506 Budget Types
  - RBRCOMP – Update Budgets for 0506
  - RTVBGRP – Confirm 0506 Budget Groups
  - RBRGTYP – Confirm 0506 Budget Types
  - RORMESG – Associate Budget Messages created on RTVMESG with Budget Groups
  - RORRULE – Create 0506 Budget Group Assignment Rules

165

---

---

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

- ◆ Funds Management validation tables and rules
  - ROAINST
  - RFRBASE – Update FUNDS for the 0506 Aid Year
  - RFRMGMT – Build 0506 Aid-year Specific Data and Rules for Each Fund You Intend to Use (Initial Page of RFRMGMT Form does NOT roll)
  - RFRDEFA – Create Term Award and Disbursement Schedules for 0506 by Aid Period
  - RFRASCH – Create OPTIONAL Term Award and Disbursement Schedules for the 0506 Aid Year
  - RORRULE – Create OPTIONAL Fund Awarding Rules for 0506 by Fund Code

166

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

- ◆ Packaging and disbursement
  - ROAINST – Confirm that Full-Time, Half-Time, ¾-Time Values are associated with EACH STVTERM Code and EACH STVLEVL code for 0506 (pg. 4)
  - RPROPTS – Create 0506 Packaging and Disbursement Options
  - RTVPGRP – Confirm 0506 Packaging Groups
  - RPRGFND – Associate 0506 Packaging Groups with Funds, indicating the Priority Order and Respective Minimum and Maximum Awards
  - RPRCLSS – Create Class Code Translations for Disbursements
  - RORRULE – Create Packaging Group Assignment Rules for 0506 if using Automated Packaging

167

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

- ◆ Packaging and Disbursement (cont.)
  - RPRDEFR – OPTIONAL Set-up for Institutions Using Self-Help, Gap or Equity Packaging
  - RPRPCKR – Create Self-Help, Gap or Equity Packaging Rules if pertinent to Packaging Philosophy
  - RPRAUDT – Create OPTIONAL rules for Audit Grading Mode if using Adjusted Enrollment Hrs.
  - RORRULE – Create 0506 OPTIONAL Group/Fund Awarding Rules
  - RPREXPT / RPRCONT – Create OPTIONAL Exemption Interface Rules and/or Contract Interface Rules
  - RPRFSRC – Create OPTIONAL Fund Source Maximum Rules for 0506

168

---

---

---

---

---

---

---

---



## New Year Roll – 2005-2006

- ◆ Loan module
  - ROAINST – Update 0506 Institutional Loan Parameters
  - RPRLOPT – Create 0506 Packaging and Disbursement Options for Each Loan Fund
  - RPRLPRD – Create 0506 Loan Periods
  - RFRMGMT – Verify that the Loan Process Indicator is Checked for Each Loan Fund to be Awarded in 0506
  - RPRLNDR – Review Lender Information and Update for 0506

169

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

- ◆ EDE module
  - ROAINST – Update EDE Corrections Indicator form, and Complete Pell, COD and Commonline Windows for 0506
  - RORDATA – Verify Presence this SCT-Delivered Table. Modifications to RORDATA are NOT recommended

170

---

---

---

---

---

---

---

---

## New Year Roll – 2005-2006

### SUMMARY – Three Steps to Roll Aid Year Data

- ◆ Create an '0506' instance of ROAINST prior to installation of new year release
- ◆ Run the ROPROLL process from Job Submission
- ◆ Review/Edit each of the Financial Aid modules listed in the New Year Start-up Checklist

171

---

---

---

---

---

---

---

---

## Security in Financial Aid

- ◆ In general, no forms should be required that are not already included in Financial Aid security, such as student accounts, admissions status, or others

172

---

---

---

---

---

---

---

---

## Job Submission

---

---

---

---

---

---

---

---

## Job Submission

- ◆ Defines the parameters in which any given process is to execute
- ◆ Communicates with the database server environment to schedule the process
- ◆ Communicates with the database server environment to print the output of the process

174

---

---

---

---

---

---

---

---

## Using Job Submission

- ◆ Start Job Submission
- ◆ Establish printers
- ◆ Define processes
- ◆ Define process parameters
- ◆ Submit job
- ◆ Review results

175

---

---

---

---

---

---

---

---

## Start Job Submission--Sleep/Wake

- ◆ Job Submission table (GURJOBS) must be running in Sleep/Wake mode prior to Job Submission
  - User ID *JOBSUB* should start this process
  - *JOBSUB* owns all output from Job Submission

176

---

---

---

---

---

---

---

---

## Establish Printers

- ◆ Printer Validation table (GTVPRNT)
  - Defines the printers to which users can print
  - Printers must be accessible from the network
  - The database server must know about the printer and the print queue

177

---

---

---

---

---

---

---

---

## Define Processes

- ◆ **Process Maintenance Form (GJAJOBS)**
  - Allows for maintenance of overall process attributes
    - ◆ Process name
    - ◆ Process description
    - ◆ Process type (Pro\*C, Pro\*COBOL, etc.)

178

---

---

---

---

---

---

---

---

## Define Process Parameters

- ◆ Parameters for any given process can be maintained dynamically through SCT Banner forms
  - GJAPDEF - Parameter Definitions Form
  - GJAPVAL - Parameter Value Validations Form
  - GJAPDFT - Default Parameter Value Validations Form

179

---

---

---

---

---

---

---

---

## Submit Job

- ◆ **Process Submission Control Form (GJAPCTL)**
  - Creates a one-up number to uniquely identify the job
  - Passes the user-entered parameters and the one-up number to the GJBPRUN table
  - At run time, all processes access GJBPRUN for their parameters

180

---

---

---

---

---

---

---

---

## Review Results

- ◆ **Process Results Form (GJARSLT)**
  - Allows for the review of the process log
- ◆ **Output files**
  - All process output (.lis files) reside in the jobsub home directory
  - Log files (.log) also reside in the jobsub home directory

181

---

---

---

---

---

---

---

---

## Job Submission

### Before

- Process Maintenance Form (GJAJOBS)
  - ◆ Define and maintains jobs
- Parameter Definitions Form (GJAPDEF)
  - ◆ Names and defines parameters
- Default Parameter Value Validations Form (GJAPDFT)
  - ◆ Define default parameters
- Parameter Value Validations Form (GJAPVAL)

### After

- Process Submission Controls Form (GJAPCTL)
  - ◆ Enter parameters and run job
- Process Results Form (GJARSLT)

182

---

---

---

---

---

---

---

---

## JOBSUB

- ◆ **JOBSUB** is a System user ID, not an Oracle user ID
- ◆ **JOBSUB** starts Job Submission table (GURJOBS) for each database
- ◆ All reports (.lis files) reside in **JOBSUB**'s directory

183

---

---

---

---

---

---

---

---

## Job Submission Exercise

- Run the Budget Component Report
- Type the job name into the Go To window
- Printer: DATABASE
- Parameter 01: 0304
- From Submission block, Save to run
- Options – Review Output
- Double-click Filename - \*.Jis – OK
- Options – Show Document
- GJARSLT

184

---

---

---

---

---

---

---

---

## Population Selection

---

---

---

---

---

---

---

---

## Population Selection

- ◆ Extracts a subset of PIDMs for use in SCT Banner reports and letters
- ◆ Allows processes to be run using that subset of PIDMs, such as people, vendors, or organizations
- ◆ Used for reporting purposes or for facilitating Letter Generation

186

---

---

---

---

---

---

---

---

### How does Population Selection Work?

- ◆ Selects the PIDM\* for individuals who meet the selection criteria

```
SELECT  Pidm
FROM    Table
WHERE   Criteria
```

\*Primary Identification Master

---

---

---

---

---

---

---

---

187

### Four Basic Steps

- ◆ Identify an application
- ◆ Define the Population Selection criteria
- ◆ Extract the IDs that match the criteria
- ◆ Review the Population Selection

---

---

---

---

---

---

---

---

188

### Step 1: Identify An Application

- ◆ An application owns a grouping of Population Selections
  - The rules established by the parent application will be inherited by the child objects
  - Examples: General, Recruiting, Alumni
- ◆ Although the application must be defined in the Application Rules table (GLRAPPL), the rules (criteria) at this level are optional

---

---

---

---

---

---

---

---

189

## Step 2: Define Population Selection Criteria

### ◆ Example: test\_female

```
SELECT spbpers_pidm
FROM spbpers, spriden
WHERE spriden_pidm = spbpers_pidm
***AND spriden_change_ind IS NULL
AND spraddr_stat_code = 'PA'
AND spbpers_sex = 'F'
AND spriden_entity_ind = 'P';
```

190

---

---

---

---

---

---

---

---

---

---

## Define Population Selection Criteria (Cont.)

### ◆ Population Selection Definition Rules table (GLRSLCT)

- Allows for entry of the specific criteria which distinguishes one population from another
- If the tables are joined by PIDM, the PIDM join is automatic
- If the tables are joined by additional columns, the conditions must be explicitly defined

### ◆ As the form is exited, the process Parameter Selection Process (GLBPARM) compiles the select statement

191

---

---

---

---

---

---

---

---

---

---

## Step 3: Extract the IDs

### ◆ Run the Population Selection Extract Process (GLBDATA) from Job Submission

- Execute GLBDATA from the Job Submission Form (GJAPCTL)
- Or enter GLBDATA in Direct Access

192

---

---

---

---

---

---

---

---

---

---



## Extract the IDs (Cont.)

### ♦ Population Selection Extract Process (GLBDATA) parameters

- Selection Identifier 1: Required/Capital Letters
- Selection Identifier 2:\*
- New Selection Identifier\*
- Enter description of Selection\*
- Enter (U)nion, (I)ntersect, (M)inus\*
- Application : Required/Capital Letters
- Creator ID : Required/Capital Letters
- Detailed Execution Report

\* Used only for Union, Intersect, or Minus functionality

193

---

---

---

---

---

---

---

---

---

---

## Step 4: Review Population Selection

### ♦ Population Selection Extract Data Form (GLAEXTR)

- Review the results of running Population Selection Extract Process (GLBDATA)
- Add other IDs not processed by GLBDATA
- Remove IDs processed by GLBDATA

194

---

---

---

---

---

---

---

---

---

---

## Population Selection

- ♦ Application definition
  - GLIAPPL
  - GLRAPPL
- ♦ Objects
  - GLIOBJC
  - GLIOBJT
- ♦ Variable usage
- ♦ Variable definitions
  - GLRVRBL -
- ♦ Population Selection rules
  - GLRSLCT
- ♦ Manual population creation
  - GLAEXTR
- ♦ View population
  - GLIEXTR

195

---

---

---

---

---

---

---

---

---

---

### Population Selection Exercise 5.2.1

- ◆ Write a Population Selection to extract students who have an EFC between 100 and 200. Extract the students and then run a report (one that's in baseline Banner) to print to the database the extracted student's full name and ID.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

### Letter Generation

### What Is Letter Generation?

- ◆ A way to extract data from the SCT Banner database and merge that data with text, then print the results and log what letter was created and for whom the letter was created

---

---

---

---

---

---

---

---

## Letter Generation

- ◆ Extracts data from SCT Banner based on a given Population Selection
- ◆ Merges extracted data with text
- ◆ Prints the results
- ◆ Maintains a log of printed letters

---

---

---

---

---

---

---

---

199

## How does Letter Generation work?

- ◆ It extracts specific data

```
SELECT spriden_first_name,  
       spriden_last_name  
FROM   spriden
```

- ◆ Only from those PIDMs extracted during a Population Selection

```
WHERE pop_sel criteria
```

---

---

---

---

---

---

---

---

200

## How does Letter Generation work? (Cont.)

- ◆ It then merges the extracted data with paragraphs customized for your implementation

---

---

---

---

---

---

---

---

201

## Example letter

April 20, 2000

John Doe  
123 Street  
Malvern, PA 19355

Dear John,

The University is in the process of verifying student information in order to prepare for registration using SCT Banner WEB. Please verify that the following information is current and correct:

ID: 123456789  
Birth date: 01/01/80

If the information is incorrect, notify our office immediately. Thank you for your assistance.

Sincerely,

202

---

---

---

---

---

---

---

---

## Example letter (Cont.)

April 20, 2000

John Doe  
123 Street  
Malvern, PA 19355

Dear John,

The University is in the process of verifying student information in order to prepare for registration using SCT Banner WEB. Please verify that the following information is current and correct:

ID: 123456789  
Birth date: 01/01/80

If the information is incorrect, notify our office immediately. Thank you for your assistance.

Sincerely,

Heading

New Paragraph

203

---

---

---

---

---

---

---

---

## Steps to create a letter

- ◆ Define variables to be used
- ◆ Construct paragraphs
- ◆ Combine paragraphs into letters
- ◆ Extract population
- ◆ Extract letter variables
- ◆ Print letter or create merge file

204

---

---

---

---

---

---

---

---

## Step 1: Define variables

1st Paragraph

April 20, 2000	*DATE1
John Doe	*FUNC_PERSON_NAME
123 Street	*FUNC_PERSON_ADDRESS_LINE1 (LINE2, LINE3)
Malvern, PA 19355	*FUNC_PERSON_CITY_STATE_ZIP
Dear John,	*FUNC_PERSON_SALUTATION

### ◆ Variable Rules Definition table (GLRVRBL)

- Builds selection criteria for data element
- Owned by an application
- Must have data associated with an ID

205

---

---

---

---

---

---

---

---

---

---

## Step 1: Define variables

2nd Paragraph

The University is in the process of verifying student information in order to prepare for registration using SCT Banner WEB. Please verify that the following information is current and correct:

ID: 123456789                    \*ID  
Birth date: 01/01/80           \*BIRTH\_DATE

If the information is incorrect, notify our office immediately. Thank you for your assistance.

Sincerely,

- ◆ As the form is exited, the Parameter Selection Process (GLBPARM) compiles the select statement

206

---

---

---

---

---

---

---

---

---

---

## Step 2: Construct paragraphs

- ◆ Paragraph Code Validation Form (GTVPARA)
  - Define codes and descriptions for paragraphs
- ◆ Letter Generation Paragraph Form (GUAPARA)
  - Define the contents of each paragraph
  - Paragraphs contain variables, formatting commands, and text
  - Paragraphs for merge files contain only variables

207

---

---

---

---

---

---

---

---

---

---

### Step 3: Combine paragraphs

- ◆ Letter Code Validation Form (GTVLETR)
  - Define codes and descriptions for letters
- ◆ Letter Process Form (GUALETR)
  - Combine paragraphs into a letter

---

---

---

---

---

---

---

---

208

### Example Letter - 1st Paragraph

```
*DATE1
#nl
*FUNC_PERSON_NAME
#nl
*FUNC_PERSON_ADDRESS_LINE1 (LINE2, LINE3)
#nl
*FUNC_PERSON_CITY_STATE_ZIP
#nl
#nl
Dear
*FUNC_PERSON_SALUTATION
```

---

---

---

---

---

---

---

---

209

### Example letter - 2nd Paragraph

```
#nl
The University is in the process of verifying student information in
#nl
order to prepare for registration using SCT Banner WEB. Please verify
#nl
that the following information is current and correct:
#nl
#nl
ID:
#nl
*ID
#nl
Birth date:
#nl
*BIRTH_DATE
#nl
If the information is incorrect, notify our office immediately. Thank
#nl
you for your assistance.
#nl
#nl
Sincerely,
```

---

---

---

---

---

---

---

---

210

## Example letter - Merge file

```
*DATE1
*FUNC_PERSON_NAME
*FUNC_PERSON_ADDRESS_LINE1 (LINE2, LINE3)
*FUNC_PERSON_CITY_STATE_ZIP
*FUNC_PERSON_SALUTATION
*ID
*BIRTH_DATE
```

211

---

---

---

---

---

---

---

---

---

---

## Step 4: Extract population

- ◆ A population must have been extracted prior to the next few steps in Letter Generation

212

---

---

---

---

---

---

---

---

---

---

## Step 5: Extract letter variables

- ◆ Run the Letter Selection Process (GLBLSEL) from Job Submission
  - Execute GLBLSEL from the Process Submission Control Form (GJAPCTL)
  - Or enter GLBLSEL in Direct Access
- ◆ Extracts data from the database for selected PIDMs and stores it in a collector table

213

---

---

---

---

---

---

---

---

---

---

## Step 6: Print the letter

- ◆ **Submit the Letter Generation Print Report (GLRLETR) through the Process Submission Control Form (GJAPCTL)**
  - Removes the selected variables from the Collector table
  - Merges that data with text specified in a letter
  - Prints the letter or creates the appropriate word processing file

214

---

---

---

---

---

---

---

---

## Creating a letter

- ◆ **Application Definition Rules Form (GLRAPPL)**  
(Create an application. Only once.)
- ◆ **Object Definition Rules Form (GLROBJT)**  
(Created once per object.)
- ◆ **Letter Code Validation Form (GTVLETR)**  
(Needed for every letter.)
- ◆ **Variable Definition Rules Form (GLRVRBL)**  
(Every variable has to be created only once.)\*
- ◆ **Paragraph Code Validation Form (GTVPARA)**  
(Needed for every paragraph.)\*
- ◆ **Letter Generation Paragraph Form (GUAPARA)**  
(Details needed for each paragraph.)
  - \*May be used for multiple letters in the same module

215

---

---

---

---

---

---

---

---

## Conversions

---

---

---

---

---

---

---

---



## The Big W!

Who?  
What?  
When?  
Where?  
Why?

**W?**

217

---

---

---

---

---

---

---

---

## Which way do I go?

### Manual

- Slower
- Data entry errors
- Small volume only

### Electronic

- Large volumes
- Requires development and testing

218

---

---

---

---

---

---

---

---

## Conversion Considerations

- Keeping track of PIDM on Legacy System
- What type of data you are bringing across
- Random ID# or SSN?
- Name/Address formatting
  - No number signs: "#"
  - No periods: "."
- Address types
- Multiple IDs on Legacy System?

219

---

---

---

---

---

---

---

---

## Timing Considerations

### ◆ Timing

#### ■ New year conversion

- Recommended
- Starts January 1

#### ■ Mid-year conversion

- Not recommended
- Each table must be populated to suggest SCT Banner Financial Aid was running since the beginning of the aid year

220

---

---

---

---

---

---

---

---

---

---

## Conversion Steps

1. Document!!
  2. Review current data
  3. Determine Scope (What will you convert?)
  4. Map current data to SCT Banner
  5. Write a detail data plan
  6. Review plan and get approval
  7. Develop procedures and programs
  8. Test
  9. Create database
  10. Do conversion
  11. Verify and correct
  12. Test!
- \*You will need your end users!

221

---

---

---

---

---

---

---

---

---

---

## Financial Aid System Conversions

### ◆ Persons

### ◆ Students

### ◆ Award history

### ◆ Tracking requirements

Refer to Chapters 5 and 6 of the *General Technical Reference Manual*

222

---

---

---

---

---

---

---

---

---

---

## General Person Records

- ◆ Identification table (SPRIDEN) - Person ID (required)
- ◆ Address table (SPRADDR)
- ◆ Telephone Number table (SPRTELE)
- ◆ Person table (SPBPERS) - Basic person
- ◆ Applicant Holds table (SPRHOLD) - Holds

223

---

---

---

---

---

---

---

---

## General Student Records

- ◆ Student Base table (SGBSTDN) - (Required)
  - Residency
  - Academic status
  - Educational goal
- ◆ Other student data
  - Assistantship/Fellowship/Internship
  - Educational opportunity
  - Student disability
  - Veteran status
  - Sports information

224

---

---

---

---

---

---

---

---

## Tracking Requirements

- ◆ Financial Aid Institution table (ROBINST)
  - Create one record for each aid year that you load
  - Enter data manually through the Financial Aid Institution Options Form (ROAINST)
- ◆ Tracking Requirements Status Validation table (RTVTRST)
  - All tracking requirement status codes must exist before you can load historic records

225

---

---

---

---

---

---

---

---

## Tracking Requirements (Cont.)

- ◆ Source/Background Institution Code Validation Form (STVSBGI)
  - Source background institution codes must exist before you can load historic records with SBI codes
- ◆ Tracking Requirements Validation table (RTVTREQ)
  - All tracking requirements codes must exist before you can load historic records
- ◆ Tracking Group Validation table (RTVTGRP)
  - All tracking group codes must exist before you can load historic records

226

---

---

---

---

---

---

---

---

## Award History

- ◆ Functional Staff (tables)
  - Financial Aid Institution table (ROBINST)
  - Fund Base Data table (RFRBASE)
  - Satisfactory Academic Progress Validation table (RTVSAPR)
  - Award Status Validation table (RTVAWST) - Award Status codes must exist here
  - Student Term Validation table (STVTERM) - Term codes must exist here before you can load historic records
- ◆ Technical Staff (tables)
  - Fund Aid Year Specific Data table (RFRASPC) - Packaging and Disbursement uses aid year-specific fund codes. Only required for mid-year conversion
  - Satisfactory Academic Progress Rules table (RORSAPR)
  - Applicant Status Report (RORSTAT)
  - Applicant Award table (RPRAWRD) - Create award records, one per aid year, per PIDM, per fund code.

227

---

---

---

---

---

---

---

---

## Conversion

- ◆ Financial Aid Hold
  - Applicant Status table (RORSTAT)
  - Financial Aid Holds table (RORHOLD):  
Financial Aid holds are linked to the student by PIDM, by aid year and by hold code
  - Hold Type Validation Form (RTVHOLD):  
Financial Aid hold codes must exist here before you can load historic records
- ◆ Applicant Comments
  - Applicant Status table (RORSTAT)
  - Financial Aid Institution table (ROBINST): Financial Aid Institution Options Form (ROAINST)
  - Applicants Comments (RHRCOMM): Applicant comments are created by PIDM by user ID. Comments also require sequence number

228

---

---

---

---

---

---

---

---

## Testing

- ◆ Use SCT Banner forms to view inserted data
- ◆ Navigate through several forms
- ◆ User testing

229

---

---

---

---

---

---

---

---

## Database Requirements

- ◆ A clean instance must have at least:
  - RFRFFID - Federal Fund ID table
  - RORDATA - Data Dictionary table
  - RORDSUP - Supplemental Log Data table
  - RORDVAL - Audit Log Form Data Code table
  - RORPELL - PELL Grant Payment Schedule table
  - RPRFEDR - Federal Rules table

230

---

---

---

---

---

---

---

---

## Database Requirements (Cont.)

- ◆ RTVINFC - Application Tape ID Validation table
  - ◆ (Code, Desc and Activity Date, only)
  - ◆ All other columns should be null
- ◆ RTVRJCT - Reject Validation table
- ◆ RTVPTYP - Batch Posting Type Validation table
- ◆ RTVDLBT - Batch Type Validation table
- ◆ RTVPRCD - Program Code Validation table
- ◆ RTVYICD - Year in College Description table

231

---

---

---

---

---

---

---

---

## Cleaning Database

- ◆ Review `cln_finaid_7x.sql`

232

---

---

---

---

---

---

---

---

## Migration to Production

- ◆ Review spreadsheet of tables to export/import in the workbook
  - ◆ Non-supported data -> client needs to review their database and SCT Banner installation

233

---

---

---

---

---

---

---

---

## Course Summary

- ◆ You should know:
  - How to navigate in SCT Banner
  - SCT Banner naming conventions for tables, forms, and processes
  - The structure of directories and their contents
  - How Job Submission, Population Selection, and Letter Generation work
  - Where to get needed data. Check researching directories, documentation, or consultants

234

---

---

---

---

---

---

---

---

## Additional Resources

- ◆ Other Technical training sessions
- ◆ Functional training sessions
- ◆ Documentation
- ◆ Discussion lists
- ◆ ActionWeb
- ◆ ActionMail
- ◆ ActionLine
- ◆ Professional Services staff

235

---

---

---

---

---

---

---

---

## Questions?

- ◆ *Please complete your class evaluation form.*

236

---

---

---

---

---

---

---

---

## Exercise Solutions

---

---

---

---

---

---

---

---

## Data Dictionary Exercise Answer

- Write a simple script to use as a tool to list information about SCT Banner Financial Aid tables, to include table\_name, table\_comment. Script should include ability to ask user for name of table and should accept partial input.

```
column comments format a30 word wrap
select table_name, comments
  from all_tab_comments
 where table_name like upper('&TableName')
    and table_name not like '%%'
 order by table_name
```

238

---

---

---

---

---

---

---

---

---

---

## Referential Integrity Exercise Answer

- Write a simple script to use as a tool to examine constraints placed on tables by SCT Banner designers.

```
column constraint_name format a30 word_wrap
column column_name format a25
  set linesize 130
select constraint_name, column_name,
       position, table_name
  from all_cons_columns
 where column_name like upper('&TableName%')
 order by table_name, position
```

239

---

---

---

---

---

---

---

---

---

---

## Referential Integrity Exercise Challenge Answer

- Write a script that will show: constraint name, column name, position, type of constraint, constraint status and table name.

```
column constraint_name format a30 word_wrap
column column_name format a25
column ctype format a22 head 'Constraint Type'
  set linesize 130
select all_cons_columns.constraint_name,
       all_cons_columns.column_name,
       all_cons_columns.position,
       decode(all_constraints.constraint_type,
              'R','Referential Integrity',
              'C','Not Null/Check', 'P','Primary Key', 'U','Unique',
              'V','With Check Option', 'O','With Read Only') ctype,
       all_constraints.status, all_cons_columns.table_name
  from all_constraints, all_cons_columns
```

(continued on next slide)

240

---

---

---

---

---

---

---

---

---

---



## Referential Integrity Exercise Challenge Answer (Cont.)

```
from all_cons_columns, all_constraints
where all_cons_columns.column_name like upper('&TableName%')
and all_cons_columns.constraint_name =
    all_constraints.constraint_name
and all_cons_columns.table_name = all_constraints.table_name
order by table_name, all_constraints.constraint_name, position
```

241

---

---

---

---

---

---

---

---

## General Person Exercise 1 Answer

- Write a simple query to get ID, first name, middle name, last name, and the Change indicator for persons who have had changes made to their Identification Form (SPADEN) records.

```
select spriden_id,
       spriden_first_name,
       spriden_mi,
       spriden_last_name,
       spriden_change_ind
from spriden
where spriden_change_ind = 'I'
and spriden_entity_ind = 'P'
```

242

---

---

---

---

---

---

---

---

## General Person Exercise 2 Answer

- Write a simple query to get PIDM, ID, first name, middle initial, last name, and Social Security number for persons who have had changes made to their name records.

```
select spriden_pidm,
       spriden_id,
       spriden_first_name,
       spriden_mi,
       spriden_last_name,
       spbpers_ssn,
       spriden_change_ind
from spriden, spbpers
where spriden_change_ind = 'N'
and spriden_entity_ind = 'P'
and spriden_pidm = spbpers_pidm
```

243

---

---

---

---

---

---

---

---

## Applicant Record Creation Exercise Answer

### ◆ Other than data load, how can new records be created?

- New records may be created on RRAAREQ, RNANAx, or ROASTAT.

### ◆ Explain what the process RESDTMP does.

- It cleans out the temporary data load tables.

### ◆ Explain the RCPDTMP process.

- This process will allow individual selected records to be deleted if the delete flag is set on RCRSUSP.

---

---

---

---

---

---

---

---

244

## RORRULE Exercise Answer

- Write a rule to capture in a tracking group those individuals that have a 'C' code on their current records. You will need to create a tracking group and tracking code. To complete this assignment, you should be able to run ROBBGRP and place a requirement on student records that fit the above category.

### ◆ Answer:

- Create code on RTVTREQ
- Create Group on RTVTGRP
- Find field on RNARSLT: RCRAPP2\_ELIGIBILITY\_IND = '1'
- Write RORRULE:
  - RCRAPP2\_ELIGIBILITY\_IND = '1' and
  - RCRAPP1\_CURR\_REC\_IND = 'Y'
- Save, Compile, Execute Group to test.
- Run ROBBGRP.

---

---

---

---

---

---

---

---

245

## Packaging and Disbursement Exercise 1 Answer

- Write a script that will list the student ID, aid year, fund code, and Paid Award Amount for a student whose aid year are entered as parameters.

```
select spriden_id id,
       rprawd_aidy_code aidyr,
       rprawd_fund_code fund_code,
       '$'||rprawd_paid_amt pd_amount
from spriden, spbprs, rprawd
where spriden_pidm = rprawd_pidm
and spbprs_pidm = spriden_pidm
and spbprs_dead_ind is null
and spriden_change_ind is null
and spriden_entity_ind = 'P'
and rprawd_paid_amt is not null
and rprawd_aidy_code = 's'aidyr';
```

---

---

---

---

---

---

---

---

246

## Packaging and Disbursement Exercise 2 Answer

- Write a script to list the student ID, aid year, term code, fund description, and offered award amount for the term of a student whose term code is entered as parameters. Allow for the use of wild card entry for term code.

```
select spriden_id id,
       rpratr_aidy_code aidy,
       rpratr_term_code term_code,
       rfbase_fund_title fund_desc,
       '$'||rpratr_offer_amt offered_amt
from rfbase, spbpers, rpratr, spriden
where spriden_pidm = rpratr_pidm
and rpratr_pidm = spbpers_pidm
and rpratr_fund_code = rfbase_fund_code
and spbpers_dead_ind is null
and spriden_change_ind is null
                                     (continued on next slide)
```

247

---

---

---

---

---

---

---

---

## Packaging and Disbursement Exercise 2 Answer (Cont.)

```
and spriden_entity_ind = 'P'
and rpratr_term_code like '&term_code'
order by spriden_id, rpratr_aidy_code,
       rpratr_term_code, rpratr_fund_code
/
```

248

---

---

---

---

---

---

---

---

## SAP History Exercise Answer

- Write a script to select ID, name, progress code, and term for each student coded in Banner for Satisfactory Academic Progress.
  - Some students are coded for more than one term
  - If students have more than one term, select the most current term and select that term's progress code
  - Hint: The academic progress table is RORSAPR

```
set pagesize 60
set linesize 80
run
column ID format a9
column NAME format a30
column TERM format a8
column CODE format a8
run
spool PROGRESS.lis
run
```

(Continued on next slide)

249

---

---

---

---

---

---

---

---

## SAP History Exercise Answer (Cont.)

```
select spriden_id "ID",
       spriden_last_name|| ',' || spriden_first_name "NAME",
       x.rorsapr_term_code "TERM",
       x.rorsapr_sapr_code "CODE"
from spriden, spbpers, rorsapr x
where x.rorsapr_term_code =
      (select max(y.rorsapr_term_code)
       from rorsapr y
        where y.rorsapr_pidm = x.rorsapr_pidm)
and x.rorsapr_pidm = spriden_pidm
and x.rorsapr_pidm = spbpers_pidm(+)
and spbpers_dead_ind is null
and spriden_change_ind is null
and spriden_entity_ind = 'P'
order by 2
/
spool off
```

250

---

---

---

---

---

---

---

---

---

---

## Database Structure Exercise Answer

- Write a script to return those students that have different EFCs. Script should return student ID, formatted name, EFCs and indicate if record has a 'lock'. Prompt the user for aid year.

```
set linesize 100
column name format a30 head 'Student Name'
column locked format a9 head 'Locked?'
select spriden_id ID,
       f_format_name(spriden_pidm,'LF30') Name,
       rcvappl_pell_pgi SAR_EFC,
       rcvappl_pgi Pell_EFC,
       rcvappl_tfc SYS_EFC,
       rcvappl_used_trans_no Trans,
       rorstat_lock_ind Locked
from rcvappl, rorstat, spriden
```

(continued on next slide)

251

---

---

---

---

---

---

---

---

---

---

## Database Structure Exercise Answer (Cont.)

```
where rcvappl_pidm = rorstat_pidm
and rorstat_pidm = spriden_pidm
and spriden_change_ind is null
and spriden_entity_ind = 'P'
and rcvappl_aidy_code = rorstat_aidy_code
and rcvappl_infcode = 'EDE'
and rcvappl_aidy_code = '&Aid_Year'
and rcvappl_curr_rec_ind = 'Y'
and (rcvappl_tfc <> rcvappl_pgi or
     rcvappl_tfc <> rcvappl_pell_pgi or
     rcvappl_pgi <> rcvappl_pell_pgi)
order by name;
```

252

---

---

---

---

---

---

---

---

---

---

## Database Structure Advanced Exercise Answers

- Write a script to select ID, name, unmet need, EFC, FWS work preference, and total paid awards for each student. Use wild card selection for aid year

```
set pagesize 60
set linesize 80
rem
column ID format a9
column NAME format a30
column EFC format 99,990
column UNMET_NEED format 99,990
column WORK format a6
column PAID_AMT format 99,990
rem
spool NEED.lis
rem
select spriden_id "ID",
       spriden_last_name|| ',' || spriden_first_name "NAME",
       rorstat_pri_sar_pgi "EFC",
       rorstat_unmet_need "UNMET_NEED",
       rorapp4_work_study "WORK",
       sum(sprawrd_paid_amt) "PAID_AMT"
```

(Continued on next slide)

253

---

---

---

---

---

---

---

---

---

---

## Database Structure Advanced Exercise Answers (Cont.)

```
from spriden, spbpers, sprawrd, rorstat, rorapp4, rorapp1
where rorapp1_aidy_code = 'aidy_code'
and rorapp1_cur_seq_ind = 'Y'
and rorapp1_pidm = rorapp4_pidm
and rorapp1_infcode = rorapp4_infcode
and rorapp1_seq_no = rorapp4_seq_no
and rorapp1_aidy_code = rorapp4_aidy_code
and rorapp1_pidm = rorstat_pidm
and rorapp1_aidy_code = rorstat_aidy_code
and rorapp1_pidm = sprawrd_pidm
and rorapp1_pidm = spbpers_pidm(*)
and rorapp1_pidm = spriden_pidm
and spbpers_dead_ind is null
and spriden_change_ind is null
and spriden_enlity_ind = 'P'
group by spriden_id, spriden_last_name, spriden_first_name, rorstat_pri_sar_pgi,
       rorstat_unmet_need, rorapp4_work_study
order by 2
/
spool off
```

254

---

---

---

---

---

---

---

---

---

---

## Database Structure Advanced Exercise Answers (Cont.)

- Modify your script to select only students that indicated they wanted FWS for 1998-99

```
set pagesize 60
set linesize 80
rem
column ID format a9
column NAME format a30
column EFC format 99,990
column UNMET_NEED format 99,990
column WORK format a6
column PAID_AMT format 99,990
rem
spool FWS.lis
rem
select spriden_id "ID",
       spriden_last_name|| ',' || spriden_first_name "NAME",
       rorstat_pri_sar_pgi "EFC",
       rorstat_unmet_need "UNMET_NEED",
       rorapp4_work_study "WORK",
       sum(sprawrd_paid_amt) "PAID_AMT"
```

(Continued on next slide)

255

---

---

---

---

---

---

---

---

---

---

## Database Structure Advanced Exercise Answers (Cont.)

```
where rcrapp1_aidy_code = 'aidy_code'
and rcrapp1_curr_rec_ind = 'Y'
and rcrapp1_pidm = rcrapp4_pidm
and rcrapp1_infcode = rcrapp4_infcode
and rcrapp1_seq_no = rcrapp4_seq_no
and rcrapp1_aidy_code = rcrapp4_aidy_code
and rcrapp1_pidm = rcrstat_pidm
and rcrapp1_aidy_code = rcrstat_aidy_code
and rcrapp1_pidm = rprawrd_pidm
and rcrapp1_pidm = spbpwrs_pidm(+)
and rcrapp1_pidm = spriden_pidm
and spbpwrs_dead_ind is null
and spriden_change_ind is null
and spriden_entity_ind = 'P'
and rcrapp4_work_study = '1'
group by spriden_id, spriden_last_name, spriden_first_name, rcrstat_pri_sar_pg1,
        rcrstat_ummet_need, rcrapp4_work_study
order by 2
/
spool off
```

256

## Database Structure Advanced Exercise Answers (Cont.)

- Modify your script to select students where they have any awards paid for aid year.

```
set pagesize 60
set linesize 80
rem
column ID format a9
column NAME format a30
column EFC format 99,990
column UNMET_NEED format 99,990
column WORK format a6
column PAID_AMT format 99,990
rem
spool AWARDS.lis
rem
select spriden_id "ID",
        spriden_last_name|| ',' || spriden_first_name "NAME",
        rcrstat_pri_sar_pg1 "EFC",
        rcrstat_ummet_need "UNMET_NEED",
        rcrapp4_work_study "WORK",
        sum(rprawrd_paid_amt) "PAID_AMT"
(Continued on next slide)
```

257

## Database Structure Advanced Exercise Answers (Cont.)

```
from spriden, spbpwrs, rprawrd, rcrstat, rcrapp4, rcrapp1
where rcrapp1_aidy_code = 'aidy_code'
and rcrapp1_curr_rec_ind = 'Y'
and rcrapp1_pidm = rcrapp4_pidm
and rcrapp1_infcode = rcrapp4_infcode
and rcrapp1_seq_no = rcrapp4_seq_no
and rcrapp1_aidy_code = rcrapp4_aidy_code
and rcrapp1_pidm = rcrstat_pidm
and rcrapp1_aidy_code = rcrstat_aidy_code
and rcrapp1_pidm = rprawrd_pidm
and rcrapp1_pidm = spbpwrs_pidm(+)
and rcrapp1_pidm = spriden_pidm
and spbpwrs_dead_ind is null
and spriden_change_ind is null
and spriden_entity_ind = 'P'
and rprawrd_paid_amt <> 0
group by spriden_id, spriden_last_name, spriden_first_name, rcrstat_pri_sar_pg1,
        rcrstat_ummet_need, rcrapp4_work_study
order by 2
/
spool off
```

258

## Population Selection Exercise Answer

- Write a Population Selection to extract students who have an EFC between 100 and 200. Extract the students and then run a report (one that's in baseline Banner) to print to the database the extracted student's full name and ID.
- Report to run is the Basic Applicant Report (RORAPLT)

The screenshot shows the Banner SQL Population Selection window. The 'Selection ID' is 'RORAPLT'. The 'Definition Description' is 'Student Records in Core Banner'. The 'Definition' section shows the following SQL query:

```
SELECT S000011.FULL_NAME, S000011.ID FROM S000011 WHERE S000011.EFC >= 100 AND S000011.EFC <= 200
```

The 'Fields' section shows the following columns:

Field	Data Element	Operator	Value	Y	NO
S000011.FULL_NAME	S000011.FULL_NAME	>=	100		
S000011.ID	S000011.ID	<=	200		

---

---

---

---

---

---

---

---

---

---

259

Thank you for your participation

---

---

---

---

---

---

---

---

---

---