SCT Banner HR
Technical Training

Introductions

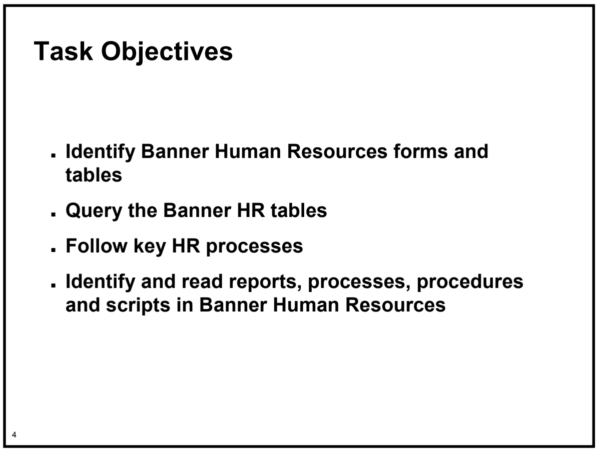
- Purpose
  - Instructor introduction
  - Attendee introductions
    - Name and Title
    - Banner Background
    - Oracle Background
    - Banner Responsibilities
    - Expectations for the course

2

Performance Objective

- To prepare the technical staff to support HR in the implementation and the operations of the Banner Human Resources product

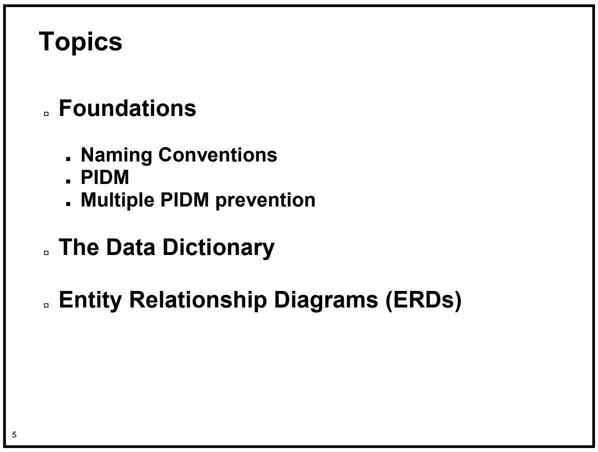
3



Task Objectives

- Identify Banner Human Resources forms and tables
- Query the Banner HR tables
- Follow key HR processes
- Identify and read reports, processes, procedures and scripts in Banner Human Resources

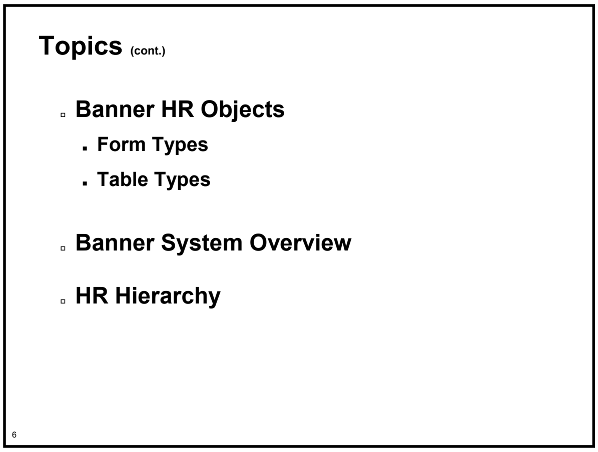
4



Topics

- Foundations
  - Naming Conventions
  - PIDM
  - Multiple PIDM prevention
- The Data Dictionary
- Entity Relationship Diagrams (ERDs)

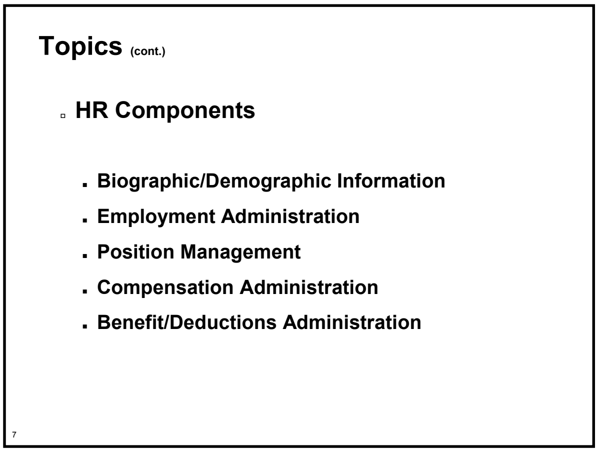
5



Topics (cont.)

- Banner HR Objects
  - Form Types
  - Table Types
- Banner System Overview
- HR Hierarchy

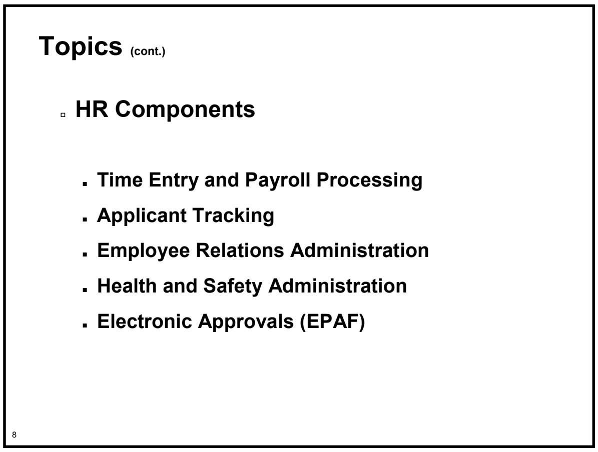
6



Topics (cont.)

- HR Components
  - Biographic/Demographic Information
  - Employment Administration
  - Position Management
  - Compensation Administration
  - Benefit/Deductions Administration

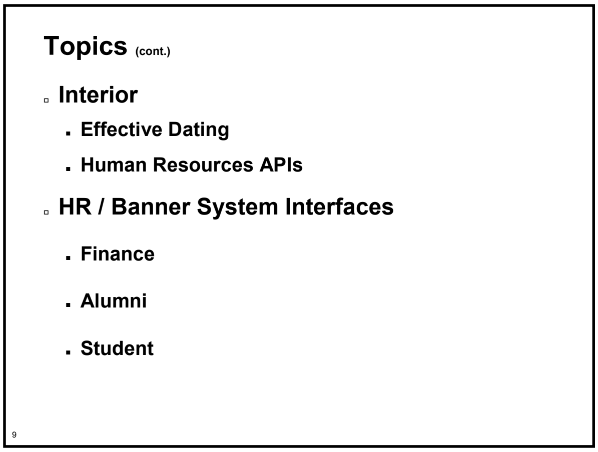
7



Topics (cont.)

- HR Components
  - Time Entry and Payroll Processing
  - Applicant Tracking
  - Employee Relations Administration
  - Health and Safety Administration
  - Electronic Approvals (EPAF)

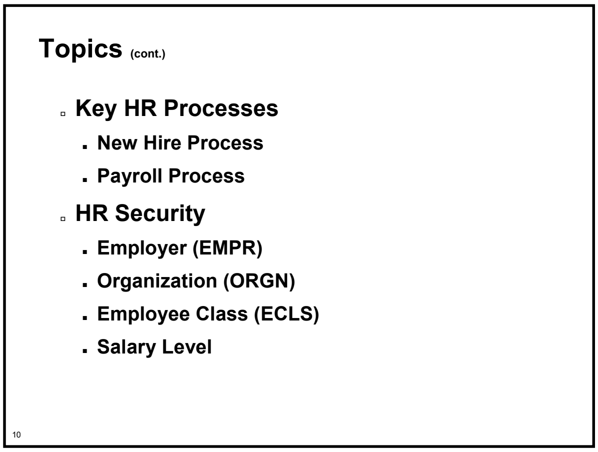
8



Topics (cont.)

- Interior
  - Effective Dating
  - Human Resources APIs
- HR / Banner System Interfaces
  - Finance
  - Alumni
  - Student

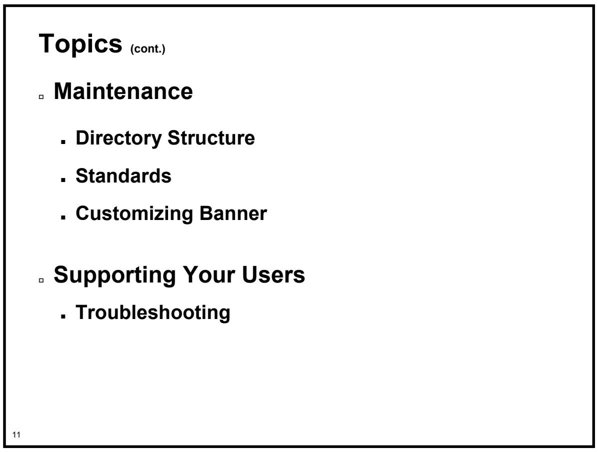
9



Topics (cont.)

- Key HR Processes
  - New Hire Process
  - Payroll Process
- HR Security
  - Employer (EMPR)
  - Organization (ORGN)
  - Employee Class (ECLS)
  - Salary Level

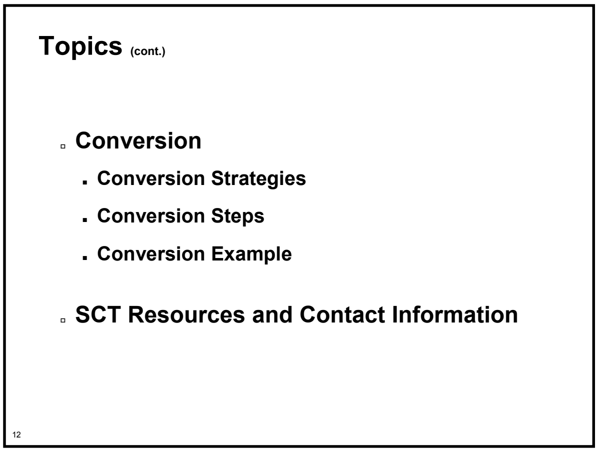
10



Topics (cont.)

- Maintenance
  - Directory Structure
  - Standards
  - Customizing Banner
- Supporting Your Users
  - Troubleshooting

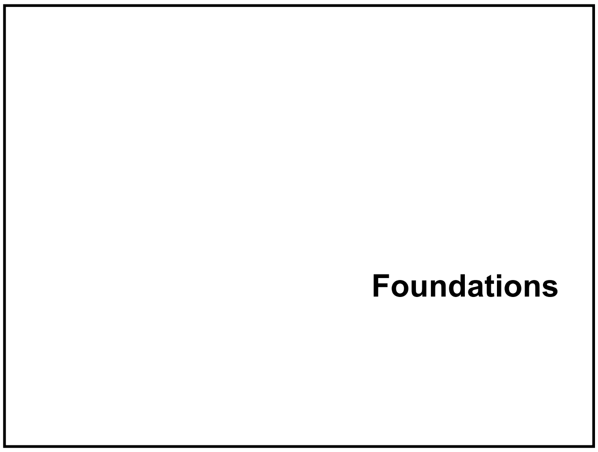
11



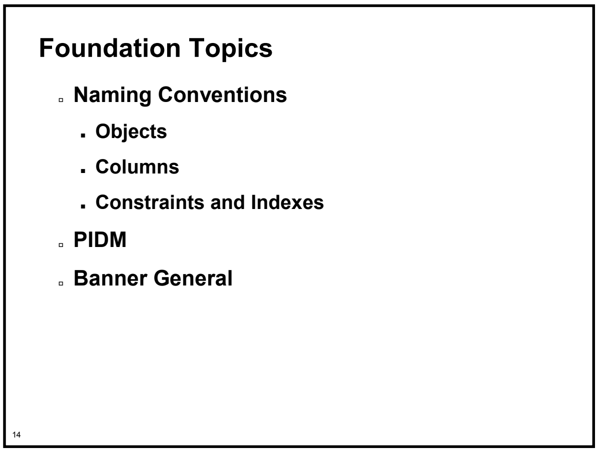
Topics (cont.)

- Conversion
  - Conversion Strategies
  - Conversion Steps
  - Conversion Example
- SCT Resources and Contact Information

12



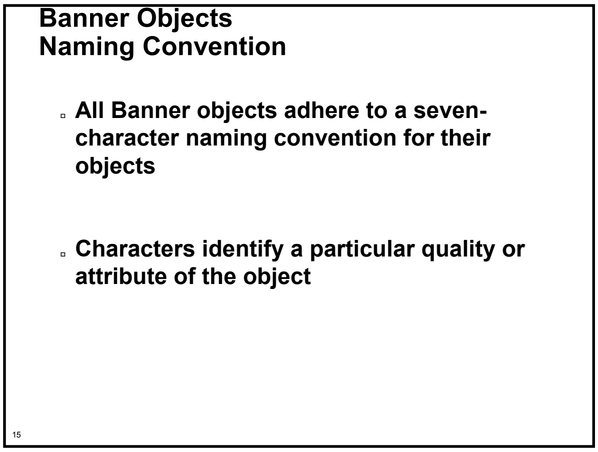
Foundations



Foundation Topics

- Naming Conventions
  - Objects
  - Columns
  - Constraints and Indexes
- PIDM
- Banner General

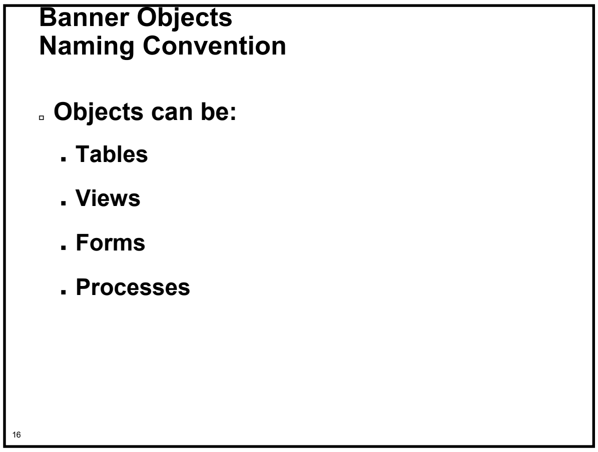
14



Banner Objects Naming Convention

- All Banner objects adhere to a seven-character naming convention for their objects
- Characters identify a particular quality or attribute of the object

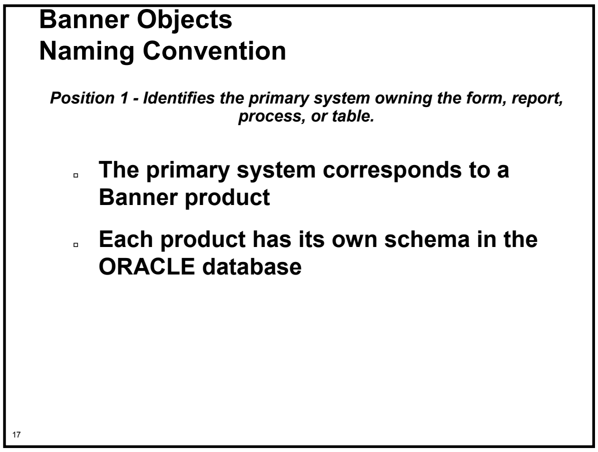
15



Banner Objects Naming Convention

- Objects can be:
  - Tables
  - Views
  - Forms
  - Processes

16

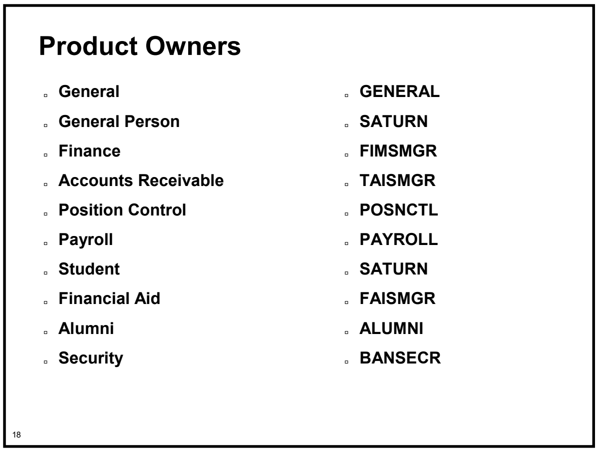


Banner Objects Naming Convention

Position 1 - Identifies the primary system owning the form, report, process, or table.

- The primary system corresponds to a Banner product
- Each product has its own schema in the ORACLE database

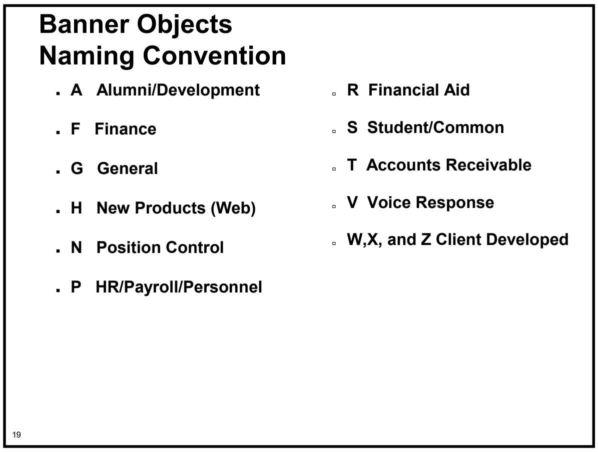
17



Product Owners

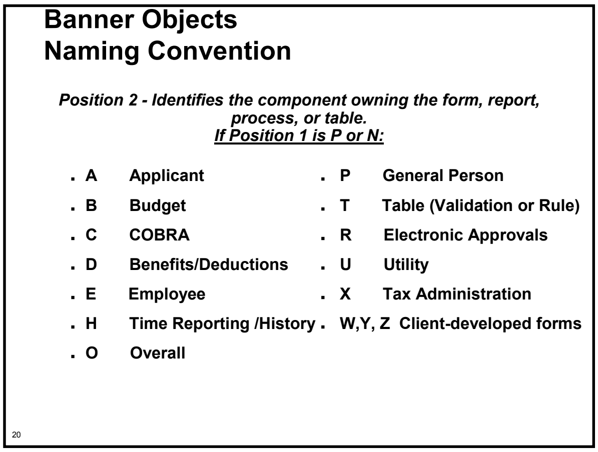
General	GENERAL
General Person	SATURN
Finance	FIMSMGR
Accounts Receivable	TAISMGR
Position Control	POSNCTL
Payroll	PAYROLL
Student	SATURN
Financial Aid	FAISMGR
Alumni	ALUMNI
Security	BANSECR

18



Banner Objects Naming Convention

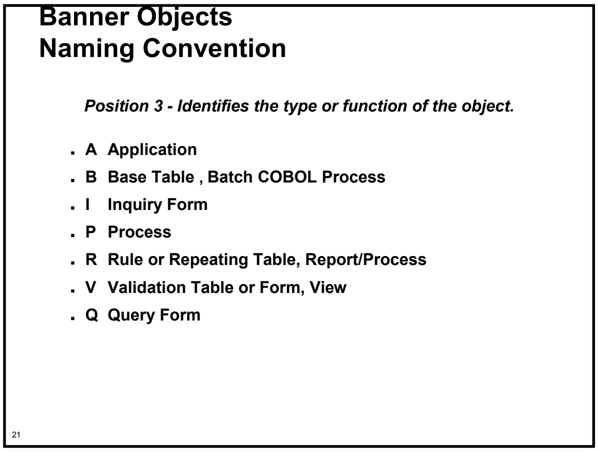
- A Alumni/Development
- F Finance
- G General
- H New Products (Web)
- N Position Control
- P HR/Payroll/Personnel
- R Financial Aid
- S Student/Common
- T Accounts Receivable
- V Voice Response
- W,X, and Z Client Developed



Banner Objects Naming Convention

Position 2 - Identifies the component owning the form, report, process, or table.
If Position 1 is P or N:

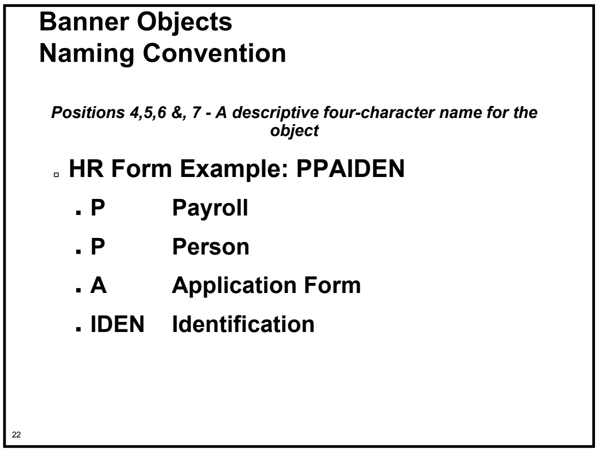
- A Applicant
- B Budget
- C COBRA
- D Benefits/Deductions
- E Employee
- H Time Reporting /History
- O Overall
- P General Person
- T Table (Validation or Rule)
- R Electronic Approvals
- U Utility
- X Tax Administration
- W,Y, Z Client-developed forms



Banner Objects Naming Convention

Position 3 - Identifies the type or function of the object.

- A Application
- B Base Table , Batch COBOL Process
- I Inquiry Form
- P Process
- R Rule or Repeating Table, Report/Process
- V Validation Table or Form, View
- Q Query Form



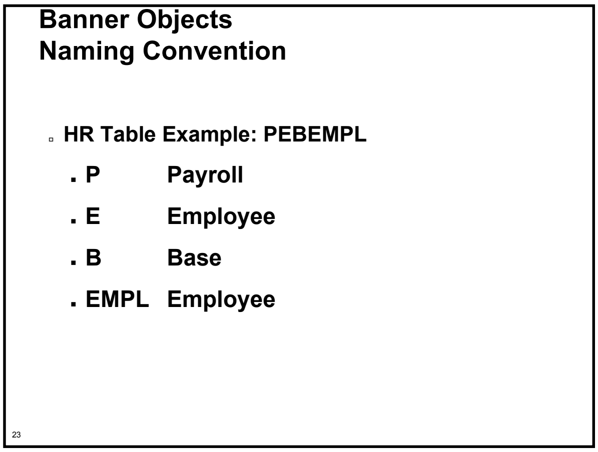
Banner Objects
Naming Convention

Positions 4,5,6 &, 7 - A descriptive four-character name for the object

- **HR Form Example: PPAIDEN**

P	**Payroll**
P	**Person**
A	**Application Form**
IDEN	**Identification**

22

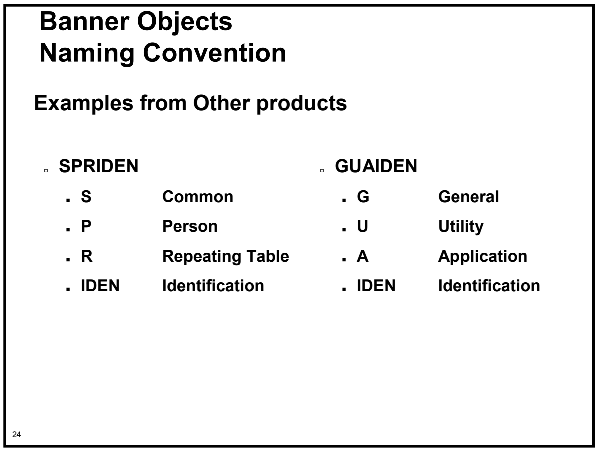


Banner Objects
Naming Convention

- **HR Table Example: PEBEMPL**

P	**Payroll**
E	**Employee**
B	**Base**
EMPL	**Employee**

23



Banner Objects
Naming Convention

Examples from Other products

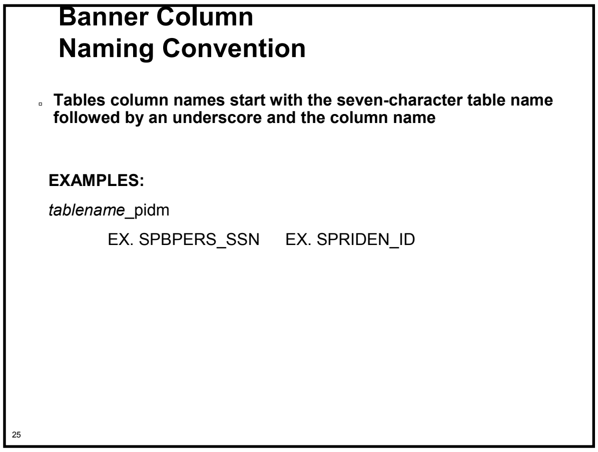
- **SPRIDEN**

S	**Common**
P	**Person**
R	**Repeating Table**
IDEN	**Identification**

- **GUAIDEN**

G	**General**
U	**Utility**
A	**Application**
IDEN	**Identification**

24



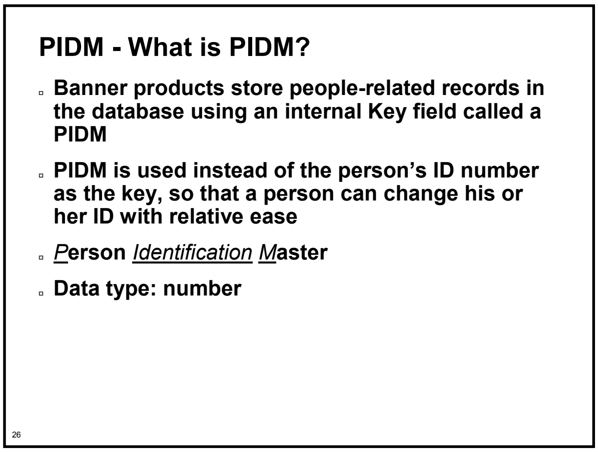
Banner Column Naming Convention

- **Tables column names start with the seven-character table name followed by an underscore and the column name**

EXAMPLES:

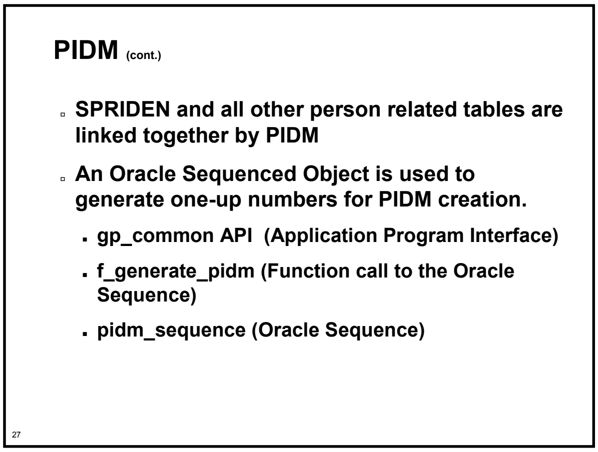
*tablename*_pidm

EX. SPBPERS_SSN EX. SPRIDEN_ID



PIDM - What is PIDM?

- **Banner products store people-related records in the database using an internal Key field called a PIDM**
- **PIDM is used instead of the person's ID number as the key, so that a person can change his or her ID with relative ease**
- ***P*****erson *Identification* *M*****aster**
- **Data type: number**



PIDM (cont.)

- **SPRIDEN and all other person related tables are linked together by PIDM**
- **An Oracle Sequenced Object is used to generate one-up numbers for PIDM creation.**
  - **gp_common API (Application Program Interface)**
  - **f_generate_pidm (Function call to the Oracle Sequence)**
  - **pidm_sequence (Oracle Sequence)**

Generated IDs

IDs – Manual, Generated, Previous

- Manual IDs entered by Users
- Generated IDs use Oracle sequence
  - f_generate_id
  - id_sequence
  - Prefix for Generated stored in SOBSEQN
- Previous IDs migrated or entered by Users
  - spriden_change_ind = 'N' or 'I'

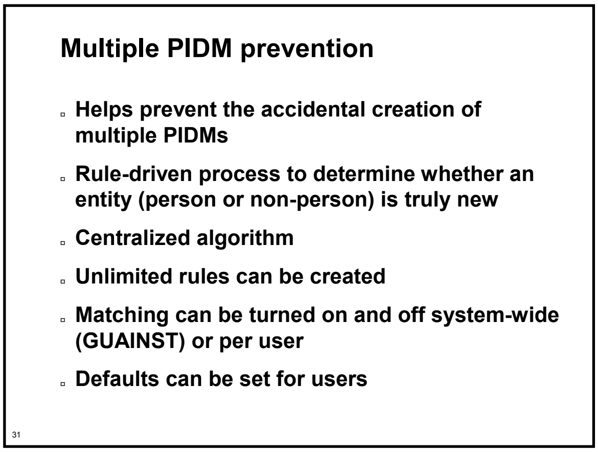
Generated IDs

- Example:
  - `select gb_common.f_generate_id() from dual;`
  - `select id_sequence.nextval from dual;`

  - A00000001 = Generated ID

Multiple PIDM prevention

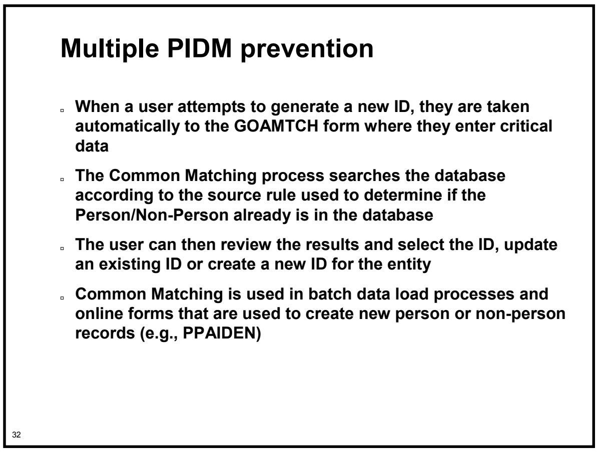
- Definition:
  - A single entity (person or non-person) is assigned two or more internal identification records in SCT Banner (PIDMs)
  - A single entity is now treated as multiple entities
- Since multiple names and IDs can be associated with a single PIDM in SCT Banner, each entity should have one and only one PIDM



Multiple PIDM prevention

- Helps prevent the accidental creation of multiple PIDMs
- Rule-driven process to determine whether an entity (person or non-person) is truly new
- Centralized algorithm
- Unlimited rules can be created
- Matching can be turned on and off system-wide (GUAINST) or per user
- Defaults can be set for users

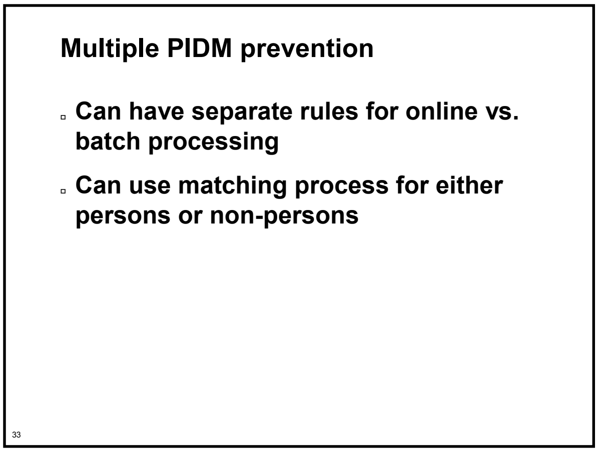
31



Multiple PIDM prevention

- When a user attempts to generate a new ID, they are taken automatically to the GOAMTCH form where they enter critical data
- The Common Matching process searches the database according to the source rule used to determine if the Person/Non-Person already is in the database
- The user can then review the results and select the ID, update an existing ID or create a new ID for the entity
- Common Matching is used in batch data load processes and online forms that are used to create new person or non-person records (e.g., PPAIDEN)

32



Multiple PIDM prevention

- Can have separate rules for online vs. batch processing
- Can use matching process for either persons or non-persons

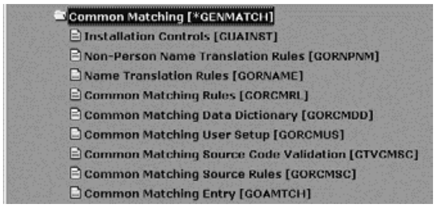
33

Multiple PIDM prevention

- HR Forms that Use Common Matching
  - PPAIDEN – Payroll Identification
  - PEAHIRE – Quick Hire
  - PEA1PAY – One Time Payments

Multiple PIDM prevention

- Common Matching Forms are located in the System Functions / Administration Menu under the General Menu


Common Matching [*GENMATCH]
Installation Controls [GUAINST]
Non-Person Name Translation Rules [GORNPNM]
Name Translation Rules [GORNAME]
Common Matching Rules [GORCMRL]
Common Matching Data Dictionary [GORCMDD]
Common Matching User Setup [GORCMUS]
Common Matching Source Code Validation [GTVCMSC]
Common Matching Source Rules [GORCMSC]
Common Matching Entry [GOAMTCH]

Multiple PIDM prevention

- Further information on Common Matching
  - Release Guides
  - User Manuals
  - Virtual Class
  - Common Matching Handbook

Exercise #1

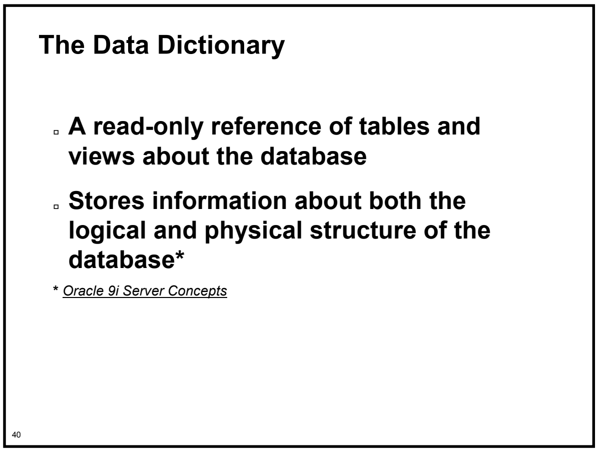
37

The Data Dictionary and Entity Relationship Diagrams

The Data Dictionary

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

39

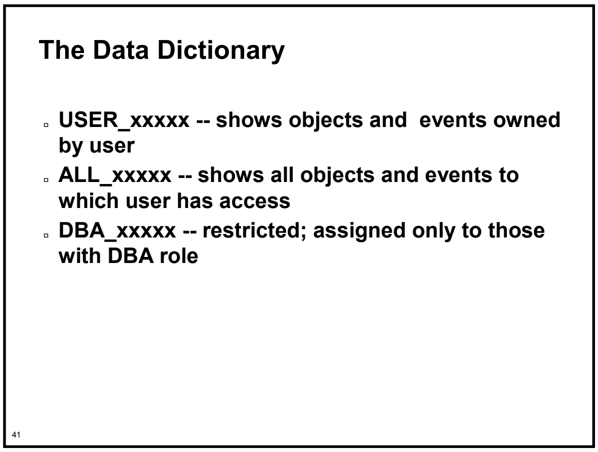


The Data Dictionary

- A read-only reference of tables and views about the database
- Stores information about both the logical and physical structure of the database*

* *Oracle 9i Server Concepts*

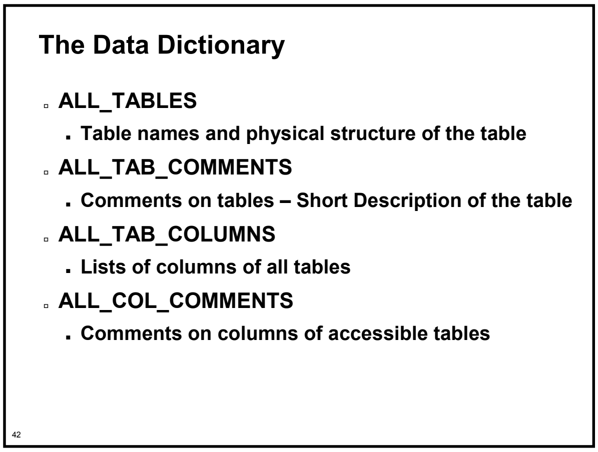
40



The Data Dictionary

- USER_xxxxx -- shows objects and events owned by user
- ALL_xxxxx -- shows all objects and events to which user has access
- DBA_xxxxx -- restricted; assigned only to those with DBA role

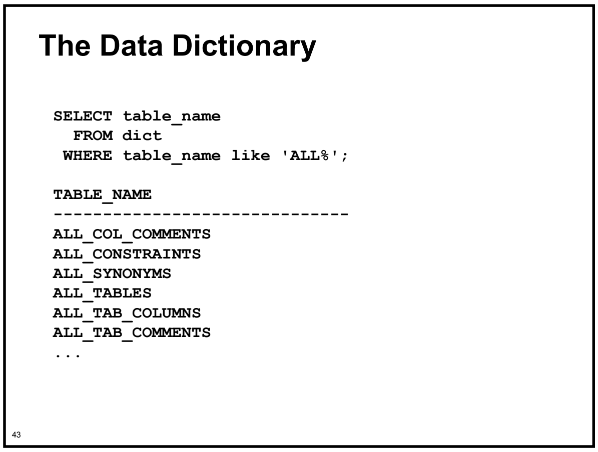
41



The Data Dictionary

- ALL_TABLES
  - Table names and physical structure of the table
- ALL_TAB_COMMENTS
  - Comments on tables – Short Description of the table
- ALL_TAB_COLUMNS
  - Lists of columns of all tables
- ALL_COL_COMMENTS
  - Comments on columns of accessible tables

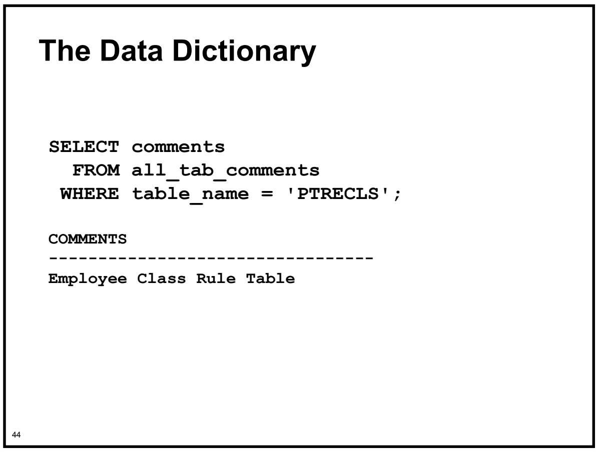
42



The Data Dictionary

```
SELECT table_name
  FROM dict
 WHERE table_name like 'ALL%';

TABLE_NAME
------------------------------
ALL_COL_COMMENTS
ALL_CONSTRAINTS
ALL_SYNONYMS
ALL_TABLES
ALL_TAB_COLUMNS
ALL_TAB_COMMENTS
...
```

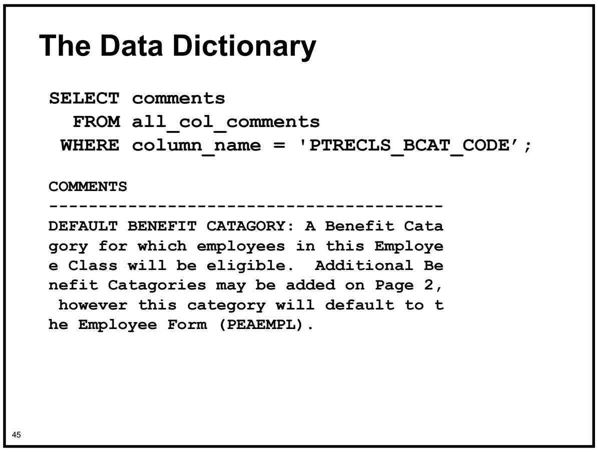
43



The Data Dictionary

```
SELECT comments
  FROM all_tab_comments
 WHERE table_name = 'PTRECLS';

COMMENTS
---------------------------------
Employee Class Rule Table
```

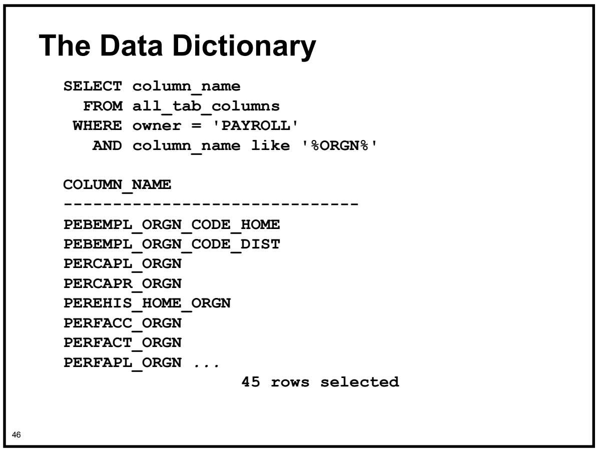
44



The Data Dictionary

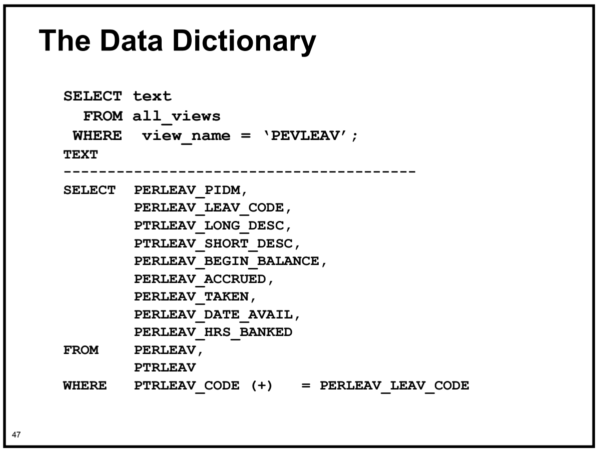
```
SELECT comments
  FROM all_col_comments
 WHERE column_name = 'PTRECLS_BCAT_CODE’;

COMMENTS
----------------------------------------
DEFAULT BENEFIT CATAGORY: A Benefit Cata
gory for which employees in this Employe
e Class will be eligible.  Additional Be
nefit Catagories may be added on Page 2,
 however this category will default to t
he Employee Form (PEAEMPL).
```

45



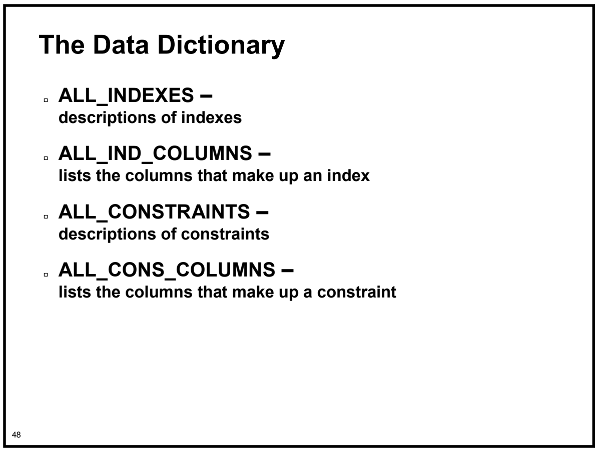
The Data Dictionary

```
SELECT column_name
  FROM all_tab_columns
 WHERE owner = 'PAYROLL'
   AND column_name like '%ORGN%'

COLUMN_NAME
------------------------------
PEBEMPL_ORGN_CODE_HOME
PEBEMPL_ORGN_CODE_DIST
PERCAPL_ORGN
PERCAPR_ORGN
PEREHIS_HOME_ORGN
PERFACC_ORGN
PERFACT_ORGN
PERFAPL_ORGN ...
                  45 rows selected
```



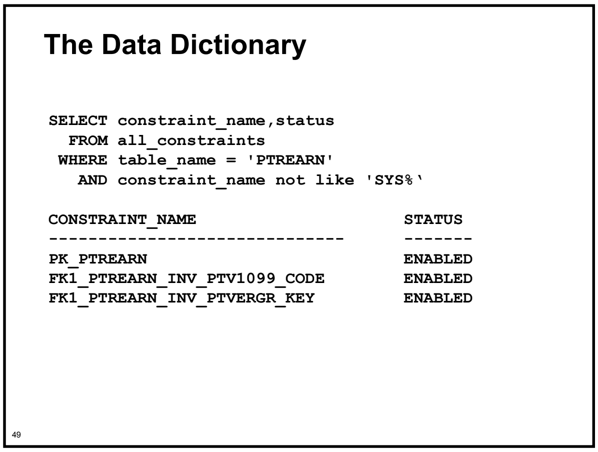
The Data Dictionary

```
SELECT text
  FROM all_views
 WHERE  view_name = ‘PEVLEAV’;
TEXT
------------------------------------
SELECT  PERLEAV_PIDM,
        PERLEAV_LEAV_CODE,
        PTRLEAV_LONG_DESC,
        PTRLEAV_SHORT_DESC,
        PERLEAV_BEGIN_BALANCE,
        PERLEAV_ACCRUED,
        PERLEAV_TAKEN,
        PERLEAV_DATE_AVAIL,
        PERLEAV_HRS_BANKED
FROM    PERLEAV,
        PTRLEAV
WHERE   PTRLEAV_CODE (+)   = PERLEAV_LEAV_CODE
```



The Data Dictionary

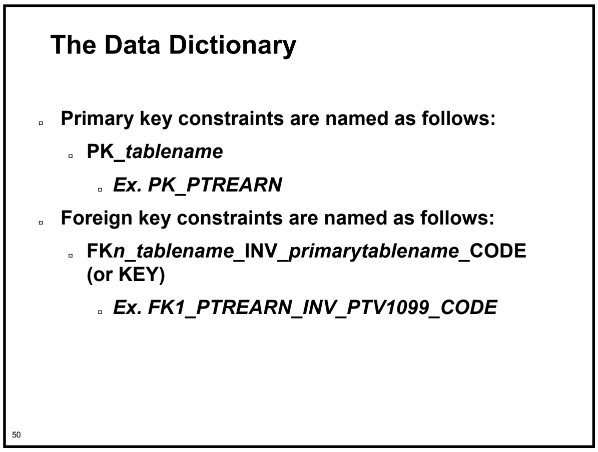
- **ALL_INDEXES –**
  descriptions of indexes
- **ALL_IND_COLUMNS –**
  lists the columns that make up an index
- **ALL_CONSTRAINTS –**
  descriptions of constraints
- **ALL_CONS_COLUMNS –**
  lists the columns that make up a constraint



The Data Dictionary

```
SELECT constraint_name,status
  FROM all_constraints
 WHERE table_name = 'PTREARN'
   AND constraint_name not like 'SYS%‘

CONSTRAINT_NAME                    STATUS
------------------------------     -------
PK_PTREARN                         ENABLED
FK1_PTREARN_INV_PTV1099_CODE       ENABLED
FK1_PTREARN_INV_PTVERGR_KEY        ENABLED
```

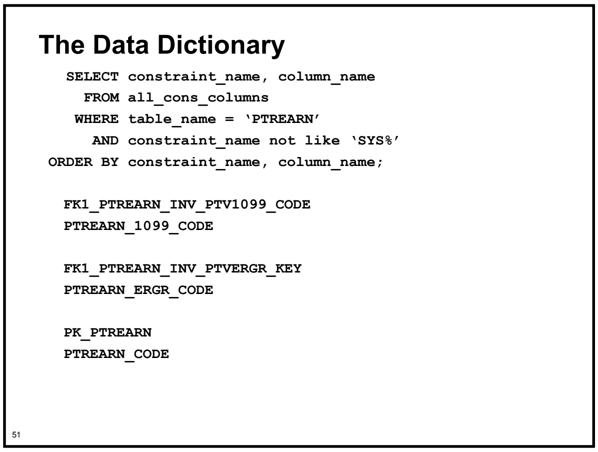
49



The Data Dictionary

- **Primary key constraints are named as follows:**
  - **PK_*tablename***
    - ***Ex. PK_PTREARN***
- **Foreign key constraints are named as follows:**
  - **FK*n_tablename*_INV_*primarytablename*_CODE (or KEY)**
    - ***Ex. FK1_PTREARN_INV_PTV1099_CODE***

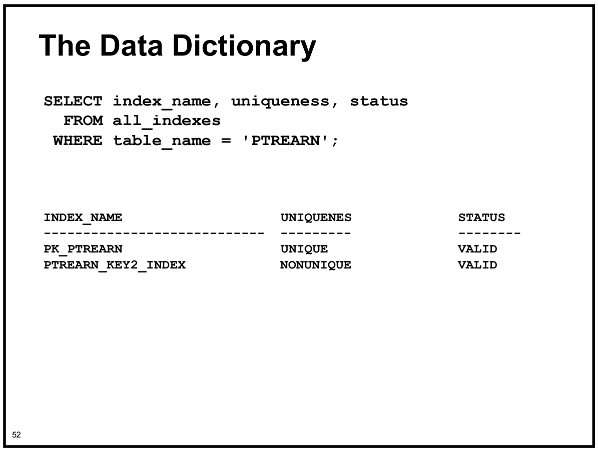
50



The Data Dictionary

```
   SELECT constraint_name, column_name
     FROM all_cons_columns
    WHERE table_name = ‘PTREARN’
      AND constraint_name not like ‘SYS%’
ORDER BY constraint_name, column_name;

  FK1_PTREARN_INV_PTV1099_CODE
  PTREARN_1099_CODE

  FK1_PTREARN_INV_PTVERGR_KEY
  PTREARN_ERGR_CODE

  PK_PTREARN
  PTREARN_CODE
```

51

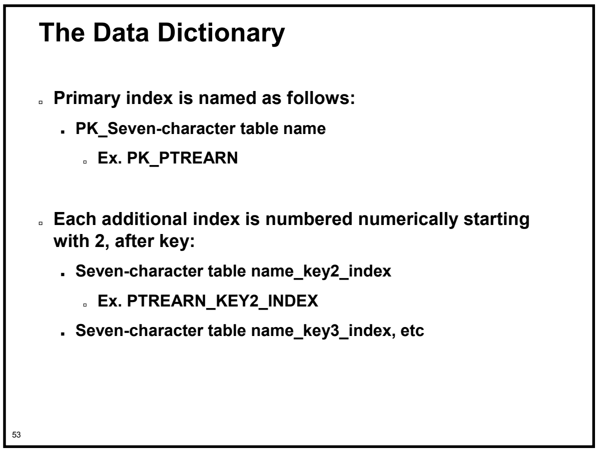


The Data Dictionary

```
SELECT index_name, uniqueness, status
  FROM all_indexes
 WHERE table_name = 'PTREARN';
```

```
INDEX_NAME                    UNIQUENES             STATUS
----------------------------  ---------             --------
PK_PTREARN                    UNIQUE                VALID
PTREARN_KEY2_INDEX            NONUNIQUE             VALID
```

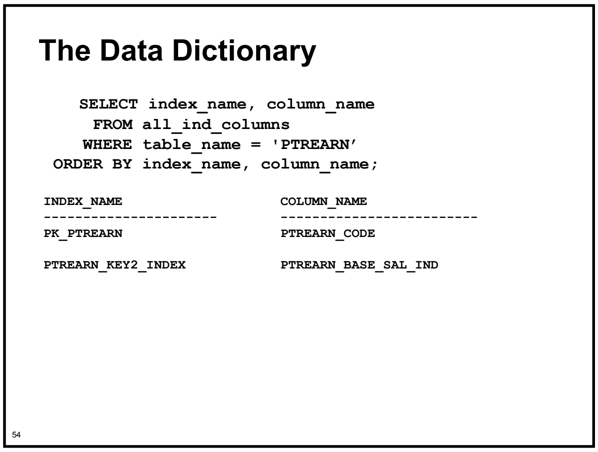
52



The Data Dictionary

- Primary index is named as follows:
  - PK_Seven-character table name
    - Ex. PK_PTREARN
- Each additional index is numbered numerically starting with 2, after key:
  - Seven-character table name_key2_index
    - Ex. PTREARN_KEY2_INDEX
  - Seven-character table name_key3_index, etc

53

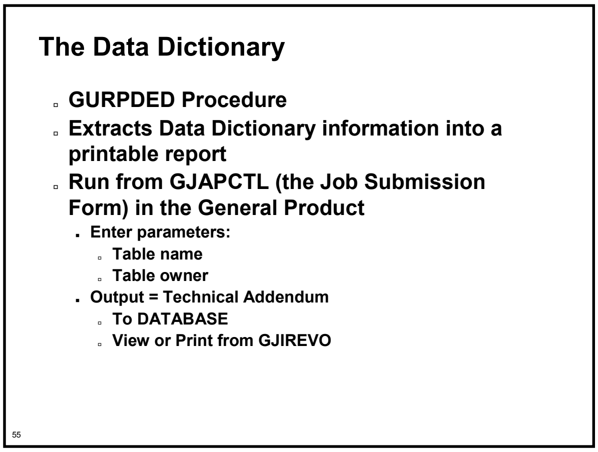


The Data Dictionary

```
    SELECT index_name, column_name
      FROM all_ind_columns
     WHERE table_name = 'PTREARN'
 ORDER BY index_name, column_name;
```

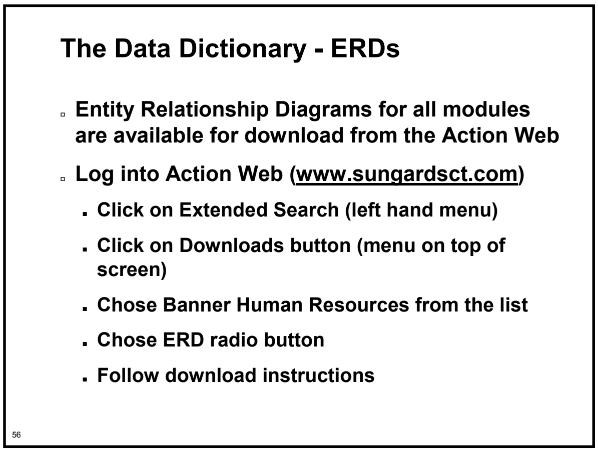
```
INDEX_NAME                    COLUMN_NAME
----------------------        -------------------------
PK_PTREARN                    PTREARN_CODE

PTREARN_KEY2_INDEX            PTREARN_BASE_SAL_IND
```

54



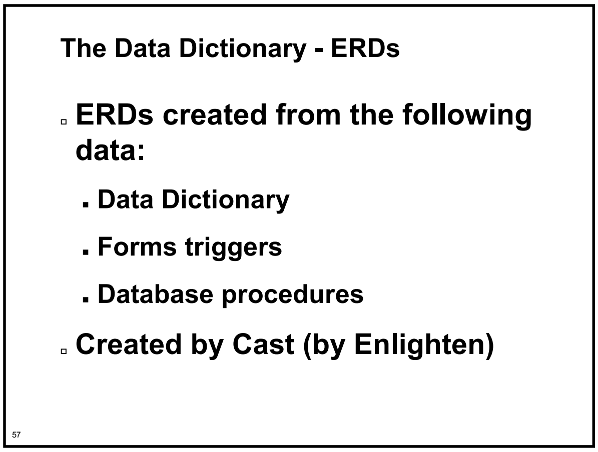
The Data Dictionary

- GURPDED Procedure
- Extracts Data Dictionary information into a printable report
- Run from GJAPCTL (the Job Submission Form) in the General Product
  - Enter parameters:
    - Table name
    - Table owner
  - Output = Technical Addendum
    - To DATABASE
    - View or Print from GJIREVO



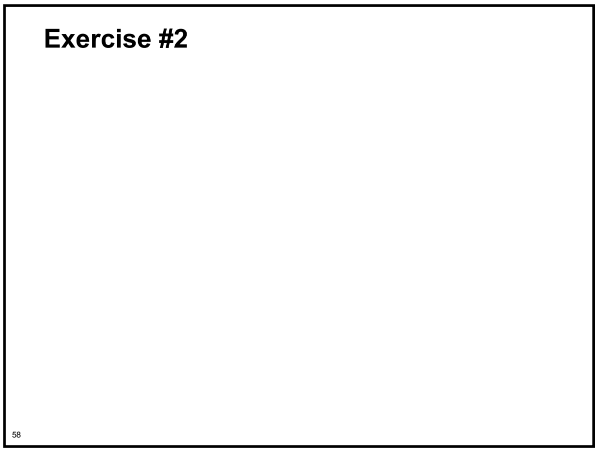
The Data Dictionary - ERDs

- Entity Relationship Diagrams for all modules are available for download from the Action Web
- Log into Action Web (www.sungardsct.com)
  - Click on Extended Search (left hand menu)
  - Click on Downloads button (menu on top of screen)
  - Chose Banner Human Resources from the list
  - Chose ERD radio button
  - Follow download instructions



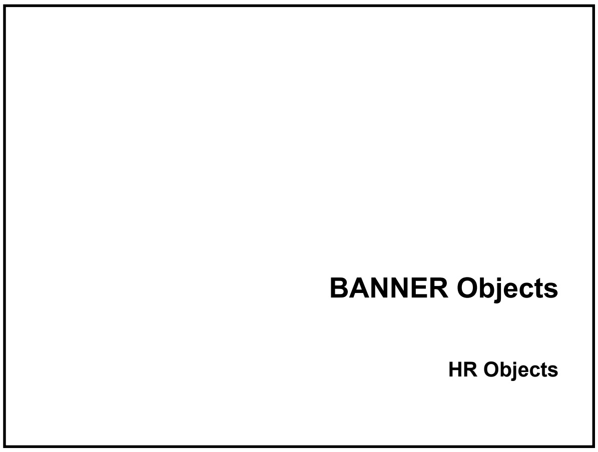
The Data Dictionary - ERDs

- ERDs created from the following data:
  - Data Dictionary
  - Forms triggers
  - Database procedures
- Created by Cast (by Enlighten)



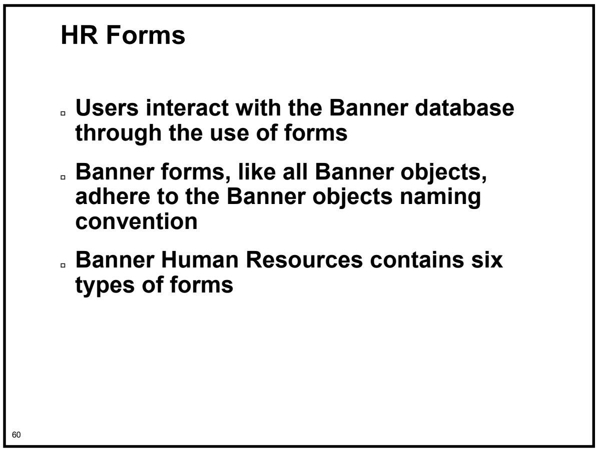
Exercise #2

58



BANNER Objects

HR Objects



HR Forms

- Users interact with the Banner database through the use of forms
- Banner forms, like all Banner objects, adhere to the Banner objects naming convention
- Banner Human Resources contains six types of forms

60

HR Forms

- Six Types of HR Forms
  - Menu
  - Application
  - Validation
  - Rules
  - Query
  - Inquiry

61

HR Forms

- **Menu Forms**

* Will not follow BANNER objects naming convention

- Lists all related forms
- Outlines the System
  - Ex. HRSEMPLOYEE

    (Employment Administration Menu)

62

HR Forms

Application Forms

- Enter Data
- Update Data
- Query the System
  - Ex. PEAEMPL (Employee Form)

63

HR Forms

Validation Forms

- Lists all possible values for a given field
- Data entry allowed
- Table and form names are the same
- Second and third characters are *TV*
- Has code, description, and activity date fields
  - Ex. PTVESKL (Employee Skills)

64

HR Forms

Rule Forms

- Define use of variables, objects, and application
  - Ex. PTRECLS (Employee Class)
  - Ex. PTRBDCA (Benefits and Deductions)
  - Ex. NTRPCLS (Position Class)

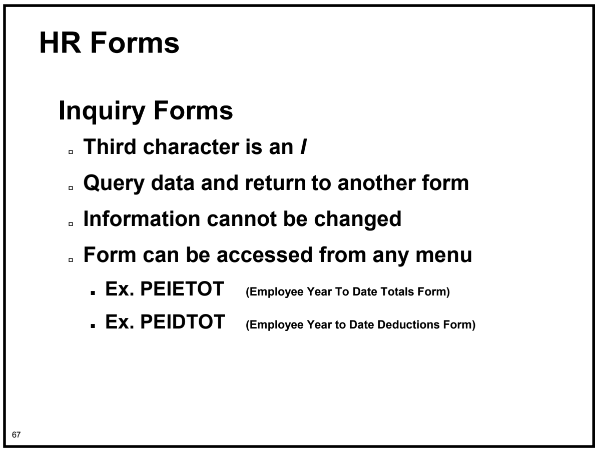
65

HR Forms

Query Forms

- Third character is a Q
- Must be called by another form
- Look-up information only
- Information cannot be changed
  - Ex. PTQECLS
    (Employee Class Query Form -
    called from Search icon in key block of PTRECLS)

66

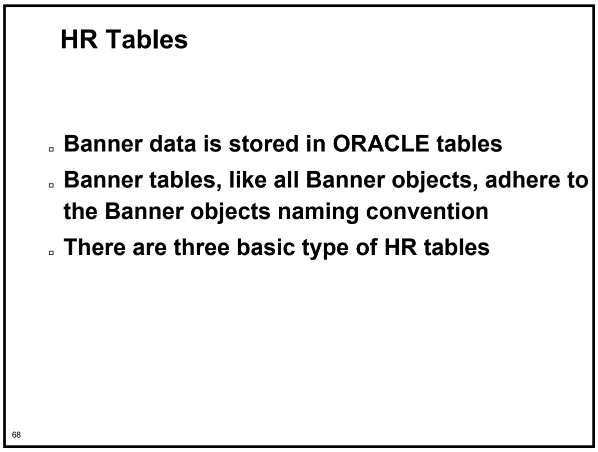


HR Forms

Inquiry Forms

- Third character is an *I*
- Query data and return to another form
- Information cannot be changed
- Form can be accessed from any menu
  - Ex. PEIETOT (Employee Year To Date Totals Form)
  - Ex. PEIDTOT (Employee Year to Date Deductions Form)

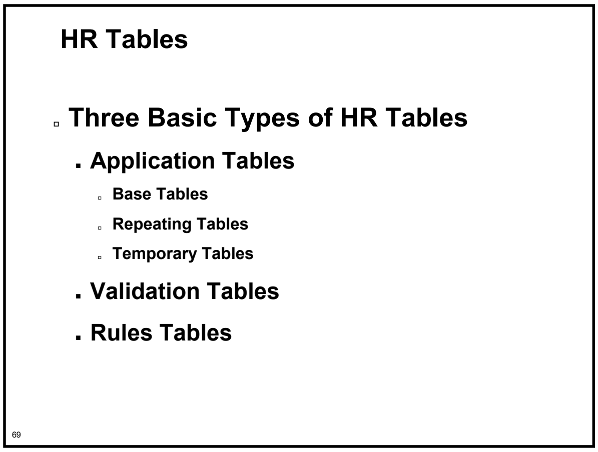
67



HR Tables

- Banner data is stored in ORACLE tables
- Banner tables, like all Banner objects, adhere to the Banner objects naming convention
- There are three basic type of HR tables

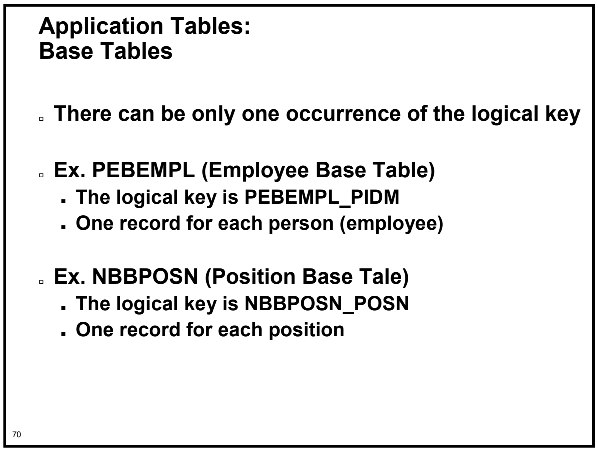
68



HR Tables

- Three Basic Types of HR Tables
  - Application Tables
    - Base Tables
    - Repeating Tables
    - Temporary Tables
  - Validation Tables
  - Rules Tables

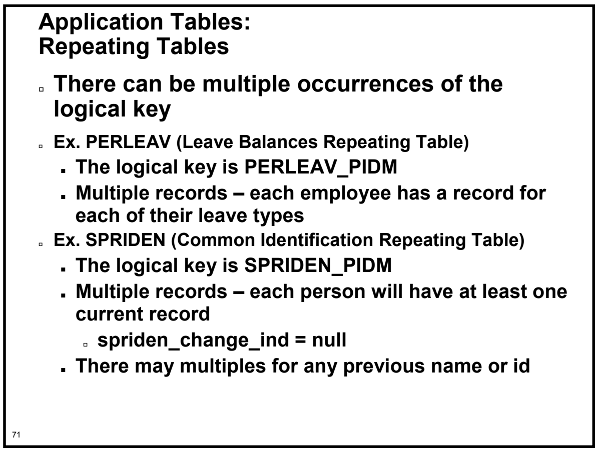
69



Application Tables: Base Tables

- There can be only one occurrence of the logical key
- Ex. PEBEMPL (Employee Base Table)
  - The logical key is PEBEMPL_PIDM
  - One record for each person (employee)
- Ex. NBBPOSN (Position Base Tale)
  - The logical key is NBBPOSN_POSN
  - One record for each position

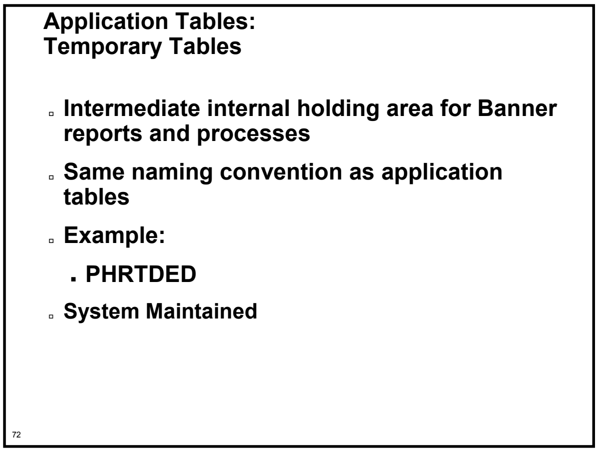
70



Application Tables: Repeating Tables

- There can be multiple occurrences of the logical key
- Ex. PERLEAV (Leave Balances Repeating Table)
  - The logical key is PERLEAV_PIDM
  - Multiple records – each employee has a record for each of their leave types
- Ex. SPRIDEN (Common Identification Repeating Table)
  - The logical key is SPRIDEN_PIDM
  - Multiple records – each person will have at least one current record
    - spriden_change_ind = null
  - There may multiples for any previous name or id

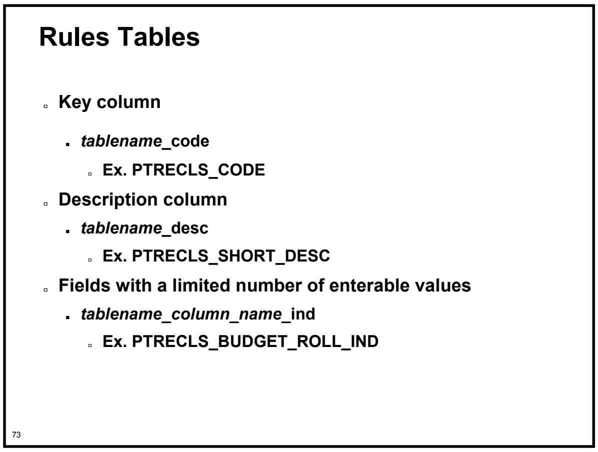
71



Application Tables: Temporary Tables

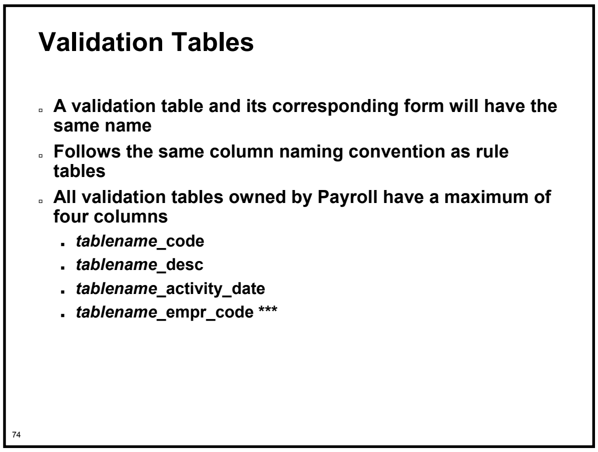
- Intermediate internal holding area for Banner reports and processes
- Same naming convention as application tables
- Example:
  - PHRTDED
- System Maintained

72



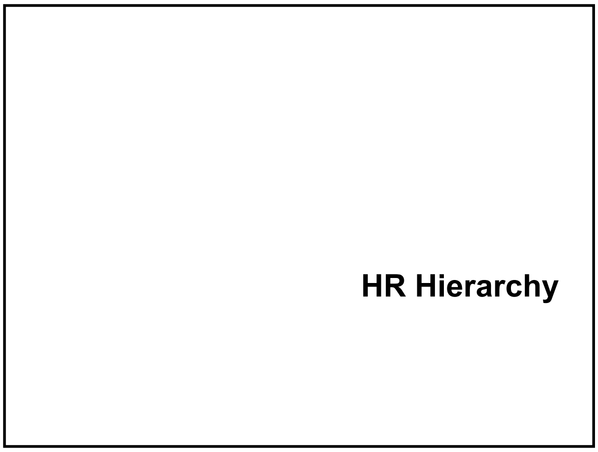
Rules Tables

- Key column
  - *tablename*_code
    - Ex. PTRECLS_CODE
- Description column
  - *tablename*_desc
    - Ex. PTRECLS_SHORT_DESC
- Fields with a limited number of enterable values
  - *tablename_column_name*_ind
    - Ex. PTRECLS_BUDGET_ROLL_IND



Validation Tables

- A validation table and its corresponding form will have the same name
- Follows the same column naming convention as rule tables
- All validation tables owned by Payroll have a maximum of four columns
  - *tablename*_code
  - *tablename*_desc
  - *tablename*_activity_date
  - *tablename*_empr_code ***



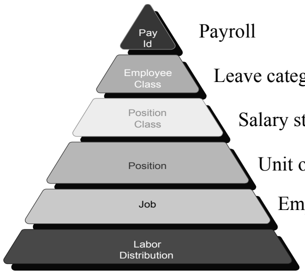
HR Hierarchy

HR Hierarchy

- **Banner HR uses a hierarchy of classes to ease data entry**
- **By associating an employee with a class or grouping, class information can be automatically entered by the System**
- **These classes and groupings are defined in the rules and validation tables**

76

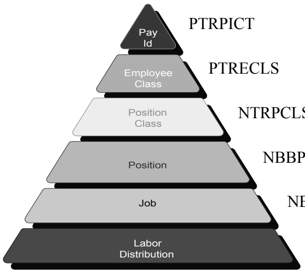
HR Hierarchy Pyramid (cont.)



Pay Id	Payroll
Employee Class	Leave category, hourly/salary
Position Class	Salary structures, titles
Position	Unit of work
Job	Employee in a unit of work
Labor Distribution	Spending source(s)

77

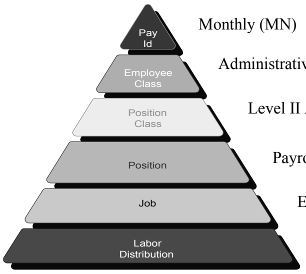
HR Hierarchy Pyramid (cont.)



Pay Id	PTRPICT
Employee Class	PTRECLS
Position Class	NTRPCLS
Position	NBBPOSN, NBRPTOT, NBRPLBD
Job	NBRBJOB, NBRJOBS...
Labor Distribution	NBRJLBD

78

HR Hierarchy Pyramid (cont.)



Monthly (MN)

Administrative (01)

Level II Accountant (P01293)

Payroll Accountant (A00012)

Employee (101) & Position (A00012)

FOAPAL

79

HR Hierarchy

- Data defaults down (NOT UP) the pyramid
  - Ex. A Labor Distribution Override at the Job level will not change the labor distribution on the position
- Data defaults once
  - Ex. A Labor Distribution Change on a position will NOT change the labor distribution of employees already assigned to that position.

80

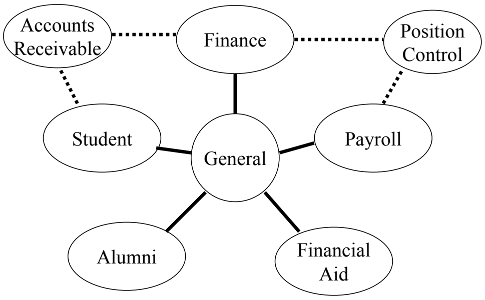
Banner Overview

Banner System Overview

- Where does Banner Human Resources fit into the entire Banner System?

82

The SCT Banner Systems



83

HR Components

Banner HR Components
Bio - Demographic

- Biographic/Demographic Information
  - Establish a unique identifier for each individual
  - Maintain:
    - Biographic information
    - Educational background
    - Professional qualifications
- Part of the General Module
- Chapter 9 of Human Resources Using Banner Guide

85

Banner HR Components
Bio - Demographic

- Core Application Forms
  - PPAIDEN
    - Establishes a unique identifier PIDM
    - Maintains biographic and demographic information
  - PPAGENL
    - Maintains professional qualifications
  - GXADIRD
    - Maintains direct deposit information
  - GOAINTL
    - Maintains international data

86

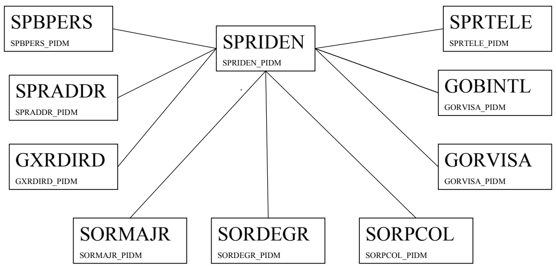
Banner HR Components
Bio - Demographic

- Rules Forms
  - PTRCERT
    - Certification Code
  - STVSBGI
    - College Codes

- Validation Forms
  - STVATYP
    - Address Type Code
  - STVCITZ
    - Citizen Type Code
  - STVETHN
    - Ethnic Code

87

Banner HR Components
Bio - Demographic



SPBPERS
SPBPERS_PIDM

SPRADDR
SPRADDR_PIDM

GXRDIRD
GXRDIRD_PIDM

SPRIDEN
SPRIDEN_PIDM

SPRTELE
SPRTELE_PIDM

GOBINTL
GORVISA_PIDM

GORVISA
GORVISA_PIDM

SORMAJR
SORMAJR_PIDM

SORDEGR
SORDEGR_PIDM

SORPCOL
SORPCOL_PIDM

88

Banner HR Components
Bio - Demographic

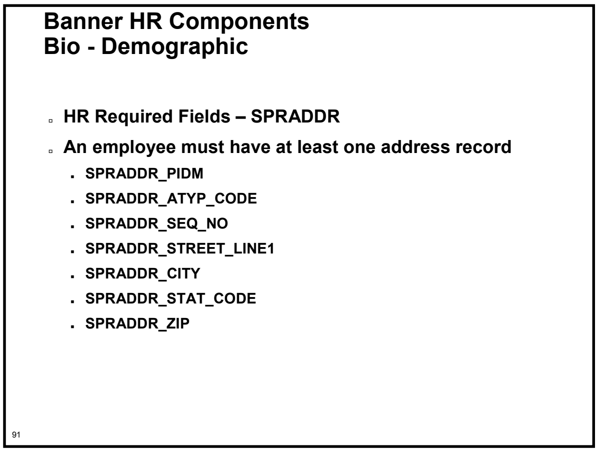
- HR Required Fields – SPRIDEN
  - SPRIDEN_PIDM
  - SPRIDEN_ID
  - SPRIDEN_FIRST_NAME
  - SPRIDEN_LAST_NAME
  - SPRIDEN_CHANGE_IND
    - At least one record with a null
    - PPAIDEN maintains
  - SPRIDEN_ENTITY_IND
    - = ‘P’ for person (‘C’ for company)
    - PPAIDEN maintains
  - SPRIDEN_ACTIVITY_DATE **

89

Banner HR Components
Bio - Demographic

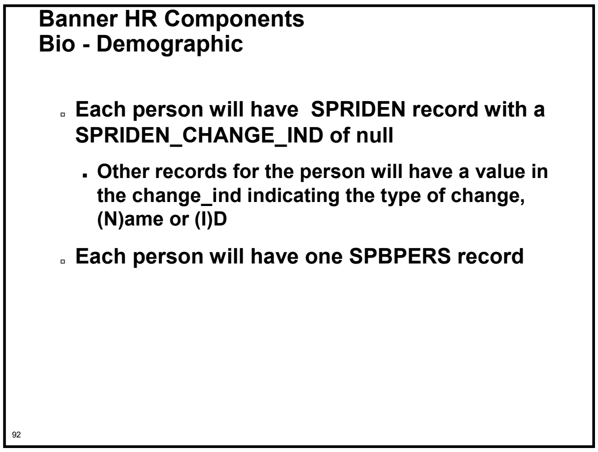
- HR Required Fields – SPBPERS
  - SPBPERS_PIDM
  - SPBPERS_SSN
  - SPBPERS_BIRTH_DATE
  - SPBPERS_ETHN_CODE
  - SPBPERS_SEX
  - SPBPERS_CITZ_CODE

90



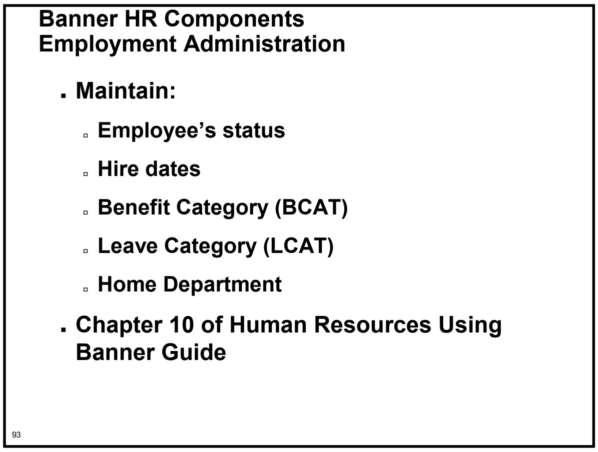
Banner HR Components
Bio - Demographic

- HR Required Fields – SPRADDR
- An employee must have at least one address record
  - SPRADDR_PIDM
  - SPRADDR_ATYP_CODE
  - SPRADDR_SEQ_NO
  - SPRADDR_STREET_LINE1
  - SPRADDR_CITY
  - SPRADDR_STAT_CODE
  - SPRADDR_ZIP



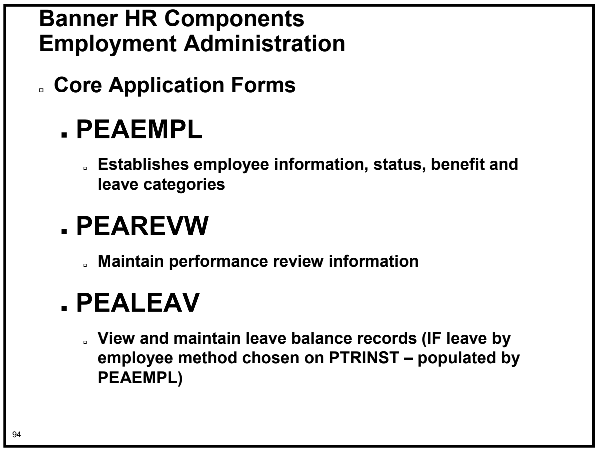
Banner HR Components
Bio - Demographic

- Each person will have SPRIDEN record with a SPRIDEN_CHANGE_IND of null
  - Other records for the person will have a value in the change_ind indicating the type of change, (N)ame or (I)D
- Each person will have one SPBPERS record



Banner HR Components
Employment Administration

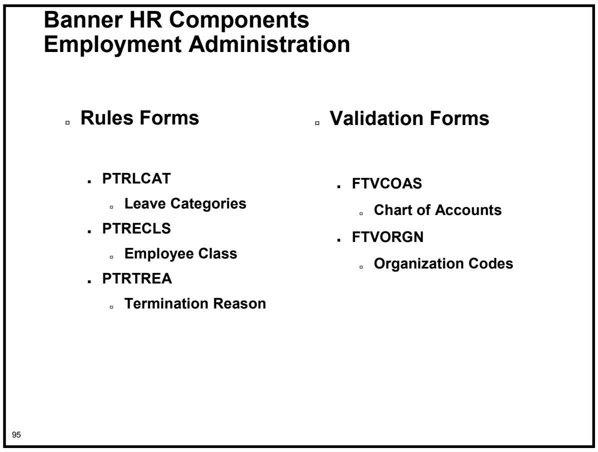
- Maintain:
  - Employee's status
  - Hire dates
  - Benefit Category (BCAT)
  - Leave Category (LCAT)
  - Home Department
- Chapter 10 of Human Resources Using Banner Guide



Banner HR Components
Employment Administration

- Core Application Forms
  - **PEAEMPL**
    - Establishes employee information, status, benefit and leave categories
  - **PEAREVW**
    - Maintain performance review information
  - **PEALEAV**
    - View and maintain leave balance records (IF leave by employee method chosen on PTRINST – populated by PEAEMPL)

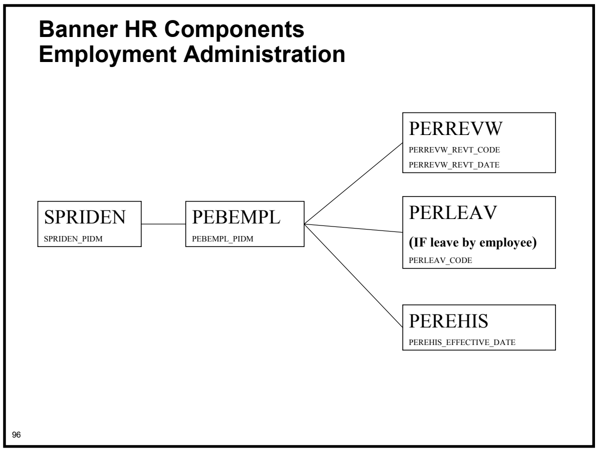
94



Banner HR Components
Employment Administration

- Rules Forms
  - PTRLCAT
    - Leave Categories
  - PTRECLS
    - Employee Class
  - PTRTREA
    - Termination Reason
- Validation Forms
  - FTVCOAS
    - Chart of Accounts
  - FTVORGN
    - Organization Codes

95



Banner HR Components
Employment Administration

96

Banner HR Components
Employment Administration

- A row in PEAEMPL defines a person as an employee
- Each employee will have one PEBEMPL record
- A history of changes made to PEBEMPL through PEAEMPL is stored in PEREHIS
- Must be have required Bio-Demo data first
- Much of the required data for the employee record defaults from the HR Hierarchy (Rules and Validation Tables)
- Other required fields are defaulted by the form PEAEMPL, but most can be overridden

97

Banner HR Components
Employment Administration

- HR Required Fields – PEBEMPL
  - PEBEMPL_PIDM
  - PEBEMPL_EMPL_STATUS
  - PEBEMPL_COAS_CODE_HOME
  - PEBEMPL_ORGN_CODE_HOME
  - PEBEMPL_COAS_CODE_DIST
  - PEBEMPL_ORGN_CODE_DIST
  - PEBEMPL_ECLS_CODE
  - PEBEMPL_LCAT_CODE

98

Banner HR Components
Employment Administration

- HR Required Fields – PEBEMPL (cont.)
  - PEBEMPL_BCAT_CODE
  - PEBEMPL_FIRST_HIRE_DATE
  - PEBEMPL_CURRENT_HIRE_DATE
  - PEBEMPL_ADJ_SERVICE_DATE
  - PEBEMPL_SENIORITY_DATE
  - PEBEMPL_FLSA_IND
  - PEBEMPL_INTERNAL_FT_PT_IND

99

Banner HR Components
Position Management

- Position Management
  - Define positions
  - Assign positions to budgets
  - Assign positions to labor distribution
  - Maintain position history
- Chapter 15 of Human Resources Using Banner Guide

100

Banner HR Components
Position Management

- Core Application Forms
  - NBAPOSN
    - Defines all positions within a position classification and fiscal year
  - NBAPBUD
    - Assign positions to budgets and FOAPAL
  - NBAFISC
    - Maintain current fiscal year

101

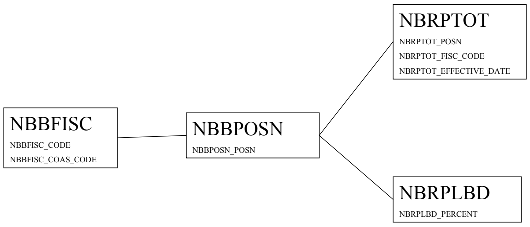
Banner HR Components
Position Management

- Rules Forms
  - NTRPCLS
    - Position Class
  - NTRSALA
    - Salary rate
  - NTRSGRP
    - Salary Group

- Validation Forms
  - FTVORGN
    - Organization
  - FTVCOAS
    - Chart of Accounts

102

Banner HR Components
Position Management



NBRPTOT
NBRPTOT_POSN
NBRPTOT_FISC_CODE
NBRPTOT_EFFECTIVE_DATE

NBBFISC
NBBFISC_CODE
NBBFISC_COAS_CODE

NBBPOSN
NBBPOSN_POSN

NBRPLBD
NBRPLBD_PERCENT

103

Banner HR Components
Position Management

- **One record in NBBPOSN for each position**
- **At least one NBRPTOT record for each position/fiscal year combination**
- **At least one NBRPLBD record for each position/fiscal year combination**

104

Banner HR Components
Compensation Administration

- **Compensation Administration**
  - **Defaults come from HR hierarchy**
  - **Maintain**
    - **Employee’s Title**
    - **Employee’s Salary**
    - **Compensation History**
- **Chapter 11 of Human Resources Using Banner Guide**

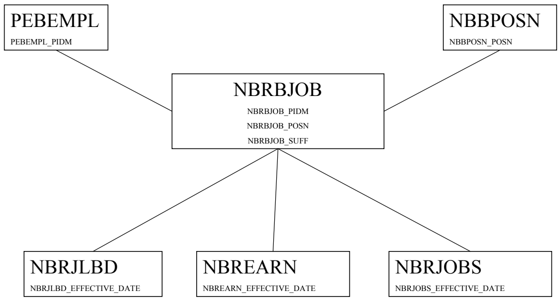
105

Banner HR Components
Compensation Administration

- **NBAJOBS**
  - **Defines the job for a particular employee with begin and end dates, title, status, labor distribution, and salary information**
- **PEIJHIS**
  - **Employee Job History Form**
- **PEALEAV**
  - **View and maintain leave balance records (IF leave by job method chosen on PTRINST – populated by NBAJOBS)**

106

Banner HR Components
Compensation Administration

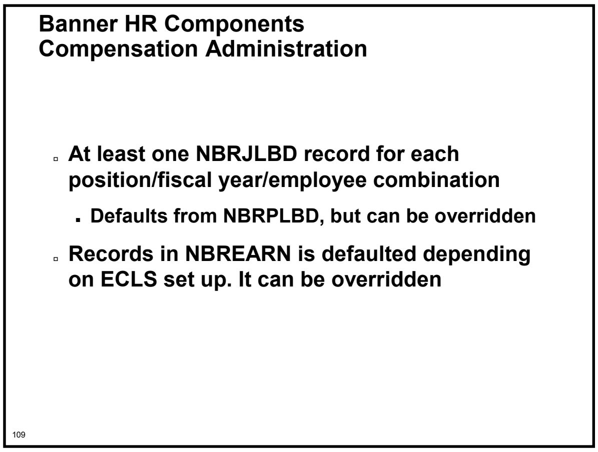


107

Banner HR Components
Compensation Administration

- **One record in NBRBJOB for each position/employee combination**
- **At least one record in NBRJOBS for each position/employee combination**
  - **Additional records for subsequent changes to job information**
    - **Example: Salary, Title**

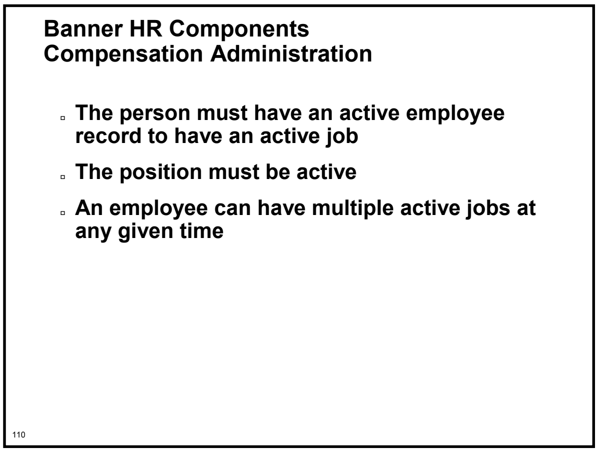
108



Banner HR Components
Compensation Administration

- At least one NBRJLBD record for each position/fiscal year/employee combination
  - Defaults from NBRPLBD, but can be overridden
- Records in NBREARN is defaulted depending on ECLS set up. It can be overridden

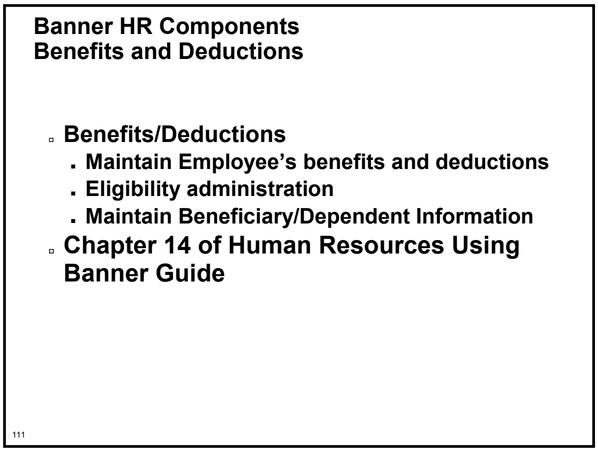
109



Banner HR Components
Compensation Administration

- The person must have an active employee record to have an active job
- The position must be active
- An employee can have multiple active jobs at any given time

110



Banner HR Components
Benefits and Deductions

- Benefits/Deductions
  - Maintain Employee’s benefits and deductions
  - Eligibility administration
  - Maintain Beneficiary/Dependent Information
- Chapter 14 of Human Resources Using Banner Guide

111

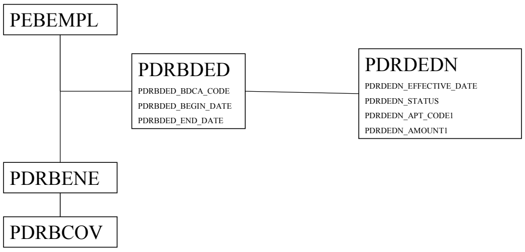
Banner HR Components
Benefits and Deductions

- Core Application Forms
  - PDADEDN
    - Establish/Maintain deductions for benefits, taxes and other withholdings
  - PDABENE
    - Beneficiary Form
  - PDABCOV
    - Beneficiary coverage Form

Banner HR Components
Benefits and Deductions

- Rules Forms
  - PTRBCAT
    - Benefit Category
  - PTRBDCA
    - Benefit/Deduction Code

Banner HR Components
Benefits and Deductions



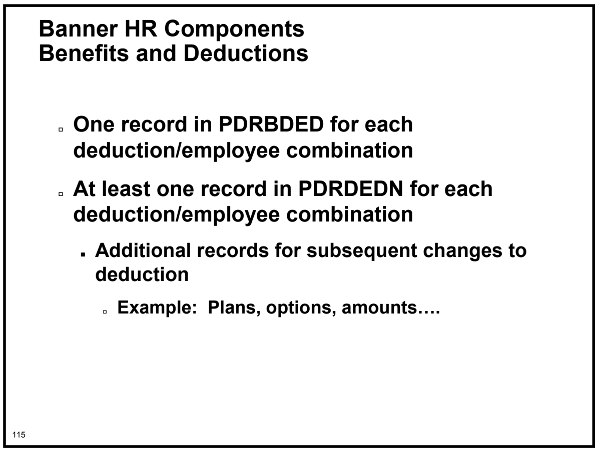
PEBEMPL

PDRBDED
PDRBDED_BDCA_CODE
PDRBDED_BEGIN_DATE
PDRBDED_END_DATE

PDRDEDN
PDRDEDN_EFFECTIVE_DATE
PDRDEDN_STATUS
PDRDEDN_APT_CODE1
PDRDEDN_AMOUNT1

PDRBENE

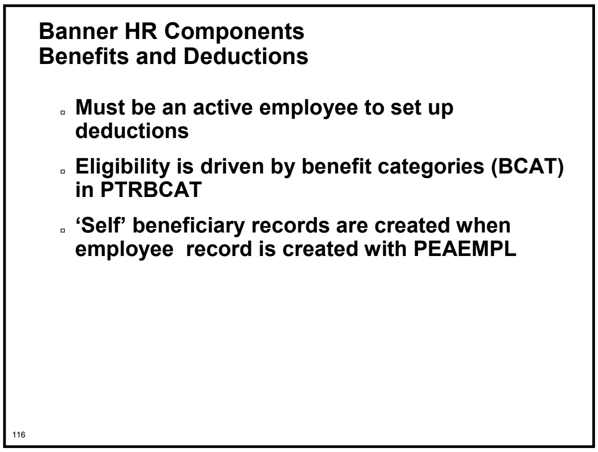
PDRBCOV



Banner HR Components
Benefits and Deductions

- One record in PDRBDED for each deduction/employee combination
- At least one record in PDRDEDN for each deduction/employee combination
  - Additional records for subsequent changes to deduction
    - Example: Plans, options, amounts….

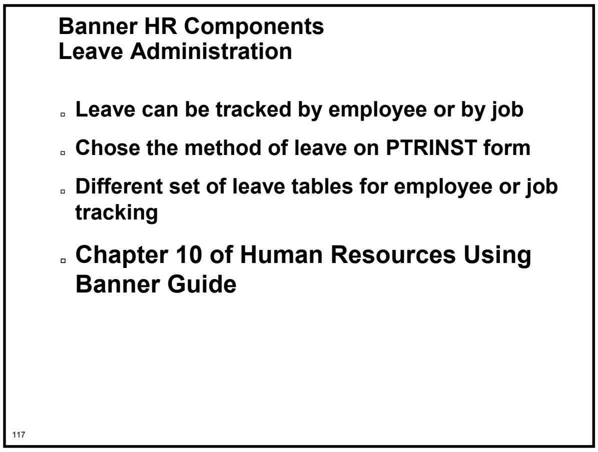
115



Banner HR Components
Benefits and Deductions

- Must be an active employee to set up deductions
- Eligibility is driven by benefit categories (BCAT) in PTRBCAT
- ‘Self’ beneficiary records are created when employee record is created with PEAEMPL

116



Banner HR Components
Leave Administration

- Leave can be tracked by employee or by job
- Chose the method of leave on PTRINST form
- Different set of leave tables for employee or job tracking
- Chapter 10 of Human Resources Using Banner Guide

117

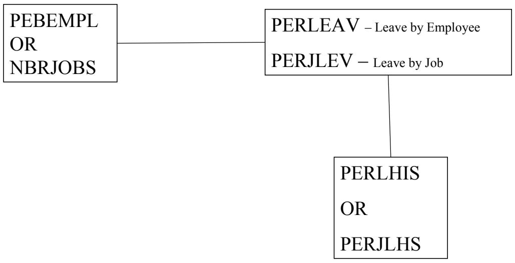
Banner HR Components
Leave Administration

- Forms
  - PEALEAV – View, track, and update balances for each leave code
  - PEILHIS – View leave balance history
  - PHIACCR – View leave accrual history

Banner HR Components
Leave Administration

- Leave by Employee
  - PEAEMPL – Leave balance records are created for those leave types for which their leave category (LCAT) on PEAEMPL dictates
- Leave by Job
  - NBAJOBS - Leave balance records are created for those leave types for which their leave category (LCAT) on NBAJOBS dictates

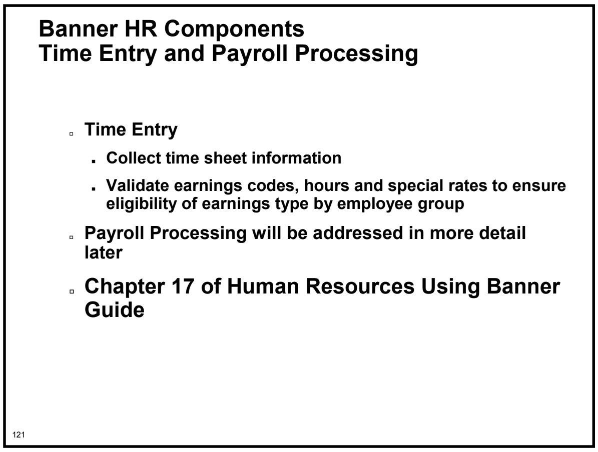
Banner HR Components
Leave Administration



PEBEMPL
OR
NBRJOBS

PERLEAV – Leave by Employee
PERJLEV – Leave by Job

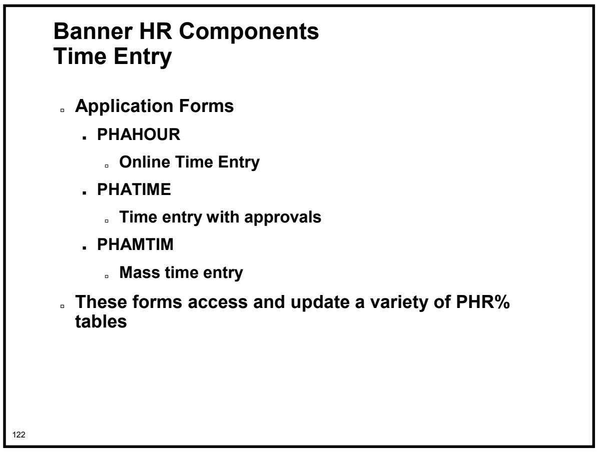
PERLHIS
OR
PERJLHS



Banner HR Components
Time Entry and Payroll Processing

- Time Entry
  - Collect time sheet information
  - Validate earnings codes, hours and special rates to ensure eligibility of earnings type by employee group
- Payroll Processing will be addressed in more detail later
- Chapter 17 of Human Resources Using Banner Guide

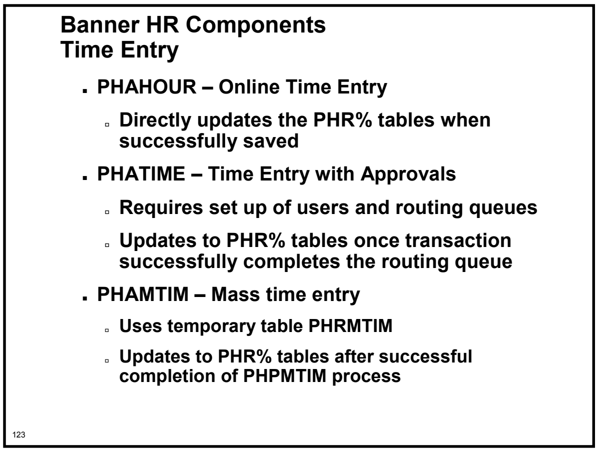
121



Banner HR Components
Time Entry

- Application Forms
  - PHAHOUR
    - Online Time Entry
  - PHATIME
    - Time entry with approvals
  - PHAMTIM
    - Mass time entry
- These forms access and update a variety of PHR% tables

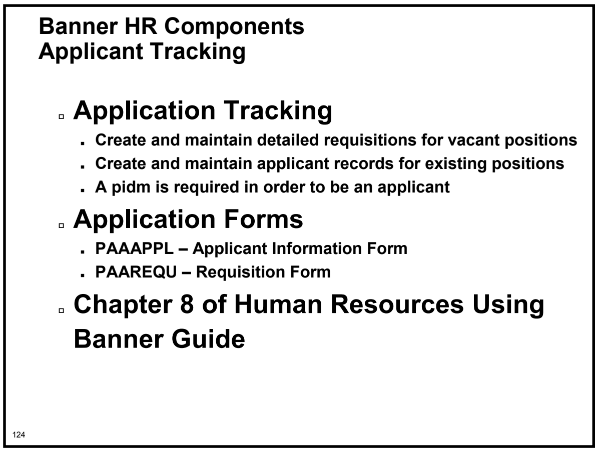
122



Banner HR Components
Time Entry

- PHAHOUR – Online Time Entry
  - Directly updates the PHR% tables when successfully saved
- PHATIME – Time Entry with Approvals
  - Requires set up of users and routing queues
  - Updates to PHR% tables once transaction successfully completes the routing queue
- PHAMTIM – Mass time entry
  - Uses temporary table PHRMTIM
  - Updates to PHR% tables after successful completion of PHPMTIM process

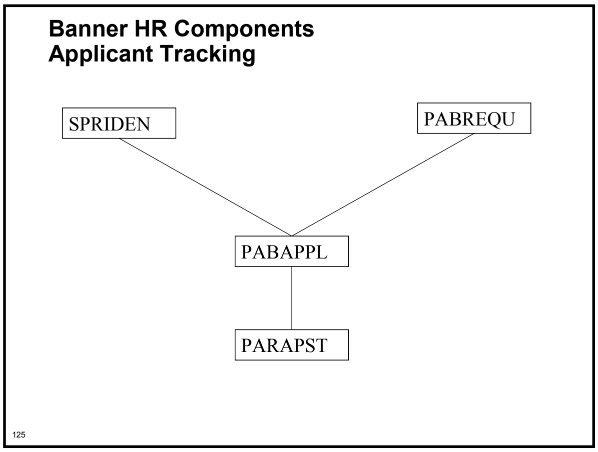
123



Banner HR Components
Applicant Tracking

- **Application Tracking**
  - Create and maintain detailed requisitions for vacant positions
  - Create and maintain applicant records for existing positions
  - A pidm is required in order to be an applicant
- **Application Forms**
  - PAAAPPL – Applicant Information Form
  - PAAREQU – Requisition Form
- **Chapter 8 of Human Resources Using Banner Guide**

124



Banner HR Components
Applicant Tracking

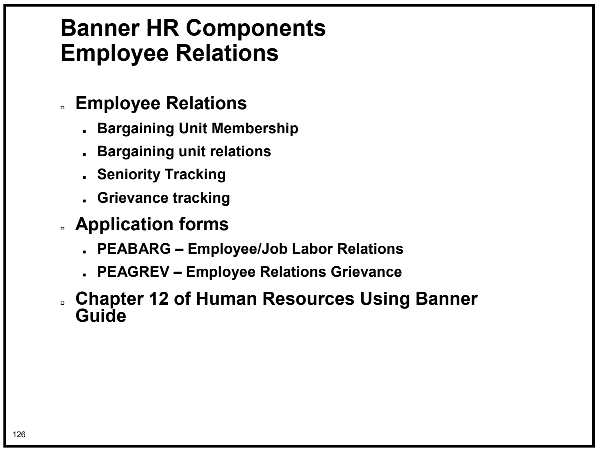
SPRIDEN

PABREQU

PABAPPL

PARAPST

125

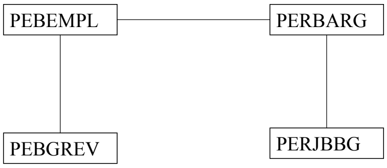


Banner HR Components
Employee Relations

- **Employee Relations**
  - Bargaining Unit Membership
  - Bargaining unit relations
  - Seniority Tracking
  - Grievance tracking
- **Application forms**
  - PEABARG – Employee/Job Labor Relations
  - PEAGREV – Employee Relations Grievance
- **Chapter 12 of Human Resources Using Banner Guide**

126

Banner HR Components
Employee Relations



127

Banner HR Components
Health and Safety

- Health and Safety
  - Record and report employee health information
  - Satisfy OSHA requirements
- Application Forms
  - PEAHSIN – Health and Safety Incident Form
- Driving Table – PEBHSIN
- Chapter 13 of Human Resources Using Banner Guide

128

Banner HR Components
Electronic Approvals

- Electronic Approvals
  - Efficient approval signature process
  - Support the movement towards a paperless office
- Chapter 19 of Human Resources Using Banner Guide

129

Banner HR Components
Electronic Approvals

- Application Forms
  - NOAEPAF – Personnel actions form
- Driving Table - NOBTRAN
- Process
  - NOPEAMA – Mass Apply Process

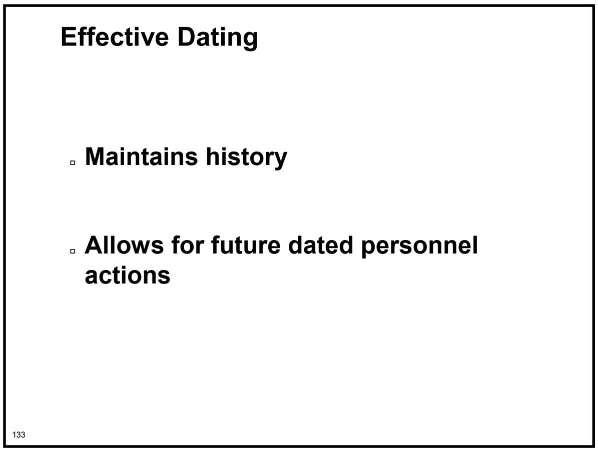
130

Interior

Interior

- Effective Dating
- HR APIs

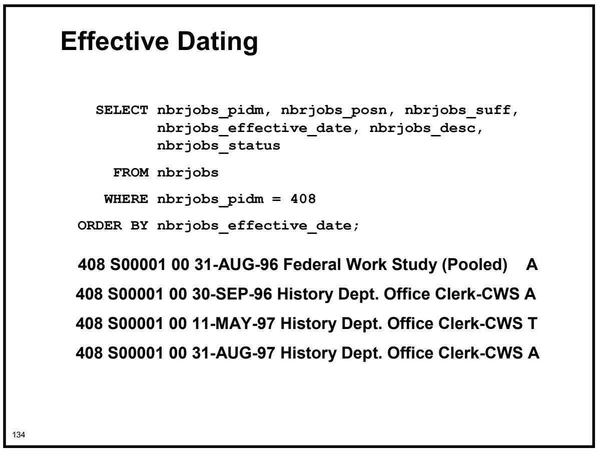
132



Effective Dating

- Maintains history
- Allows for future dated personnel actions

133

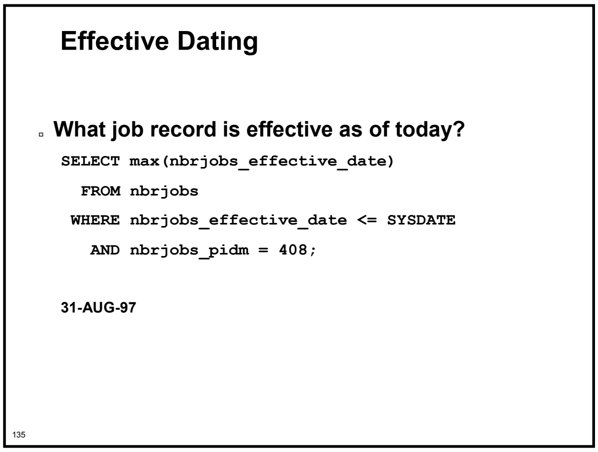


Effective Dating

```
  SELECT nbrjobs_pidm, nbrjobs_posn, nbrjobs_suff,
         nbrjobs_effective_date, nbrjobs_desc,
         nbrjobs_status
    FROM nbrjobs
   WHERE nbrjobs_pidm = 408
ORDER BY nbrjobs_effective_date;
```

```
408 S00001 00 31-AUG-96 Federal Work Study (Pooled)    A
408 S00001 00 30-SEP-96 History Dept. Office Clerk-CWS A
408 S00001 00 11-MAY-97 History Dept. Office Clerk-CWS T
408 S00001 00 31-AUG-97 History Dept. Office Clerk-CWS A
```

134



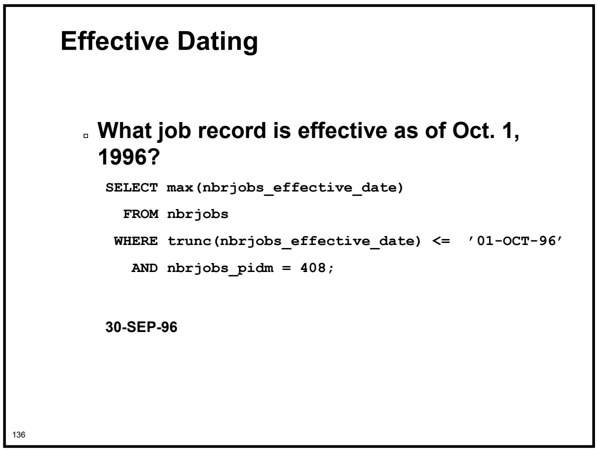
Effective Dating

- What job record is effective as of today?

```
SELECT max(nbrjobs_effective_date)
  FROM nbrjobs
 WHERE nbrjobs_effective_date <= SYSDATE
   AND nbrjobs_pidm = 408;
```

31-AUG-97

135



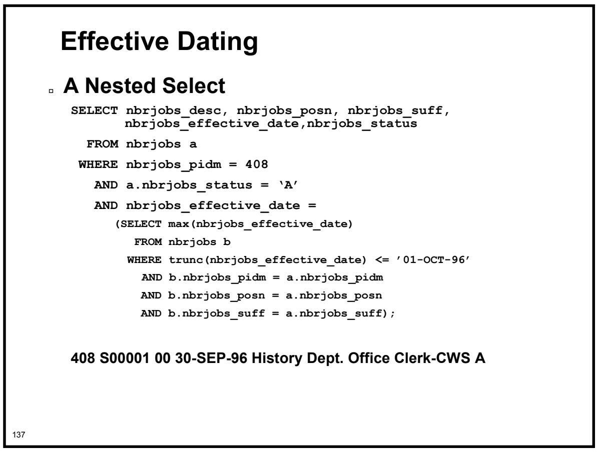
Effective Dating

- What job record is effective as of Oct. 1, 1996?

```
SELECT max(nbrjobs_effective_date)
  FROM nbrjobs
 WHERE trunc(nbrjobs_effective_date) <=  ’01-OCT-96’
   AND nbrjobs_pidm = 408;
```

30-SEP-96

136



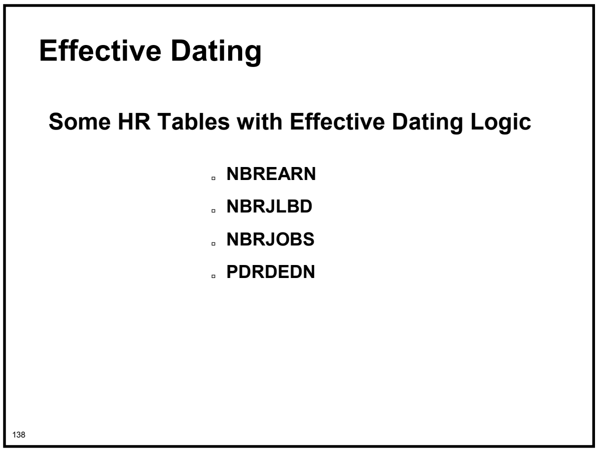
Effective Dating

- **A Nested Select**

```
SELECT nbrjobs_desc, nbrjobs_posn, nbrjobs_suff,
       nbrjobs_effective_date,nbrjobs_status
  FROM nbrjobs a
 WHERE nbrjobs_pidm = 408
   AND a.nbrjobs_status = ‘A’
   AND nbrjobs_effective_date =
      (SELECT max(nbrjobs_effective_date)
         FROM nbrjobs b
        WHERE trunc(nbrjobs_effective_date) <= ’01-OCT-96’
          AND b.nbrjobs_pidm = a.nbrjobs_pidm
          AND b.nbrjobs_posn = a.nbrjobs_posn
          AND b.nbrjobs_suff = a.nbrjobs_suff);
```

408 S00001 00 30-SEP-96 History Dept. Office Clerk-CWS A

137

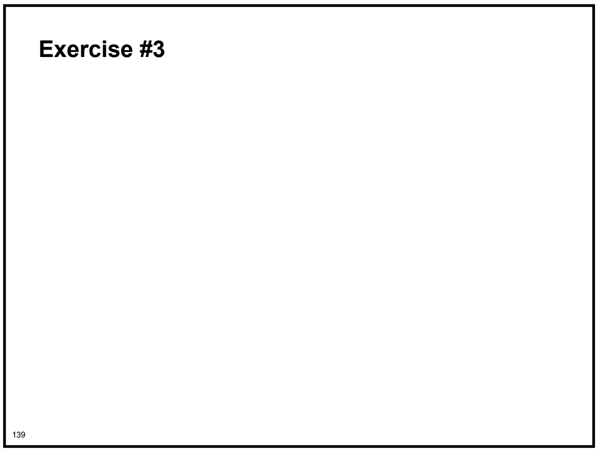


Effective Dating

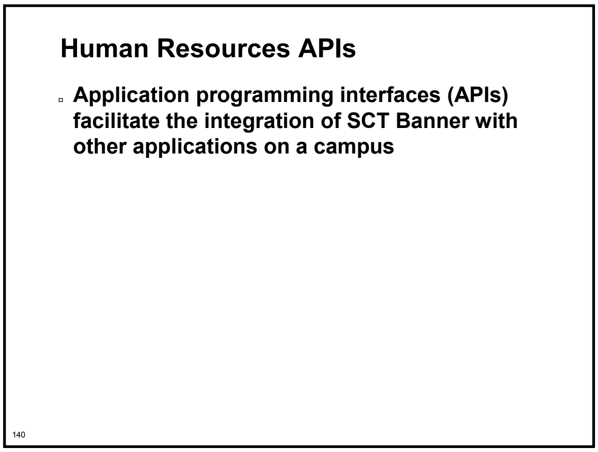
Some HR Tables with Effective Dating Logic

- **NBREARN**
- **NBRJLBD**
- **NBRJOBS**
- **PDRDEDN**

138

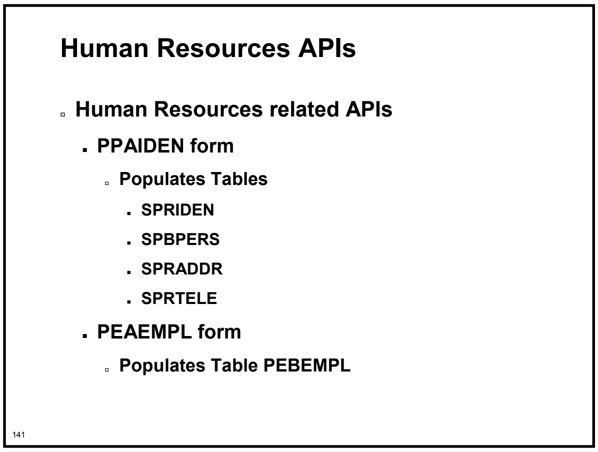


Exercise #3



Human Resources APIs

- Application programming interfaces (APIs) facilitate the integration of SCT Banner with other applications on a campus



Human Resources APIs

- Human Resources related APIs
  - PPAIDEN form
    - Populates Tables
      - SPRIDEN
      - SPBPERS
      - SPRADDR
      - SPRTELE
  - PEAEMPL form
    - Populates Table PEBEMPL

HR / Banner System Interfaces

Interfaces and Integration with other Banner Systems

- Chapter 7 of Human Resources Using Banner Guide
- Alumni Interface
- Finance Interface
- Student Integration

143

Alumni Pledge Payments

- Deductions can be gifts to the institution
  - Deduction code set up by HR (PTRBDCA)
- Alumni sets up pledge (AGAPLDG) and installments (AGAPINS)
  - Pledge information appears in AGCFDED and PEAFDED
- HR uses form PEAFDED to activate the deduction
  - Check the ‘Received Signature’ box
  - Assign appropriate BDCA code

144

Alumni Pledge Payments

- During payroll processing, PHPUPDT creates pledge payment records, (GURALMP), to be processed in Alumni
- Alumni runs Automatic Deduction Process (AGPALMP) to create pledge payments

145

Faculty Load Data

- The General information form, PPAGENL, shares faculty history data with SIAFPER, and faculty academic history data with SIAFDEG
- The Employee form, PEAEMPL, shares personnel data with SIAFPER
- Faculty Load Analysis Report (SIRCTAL)– displays salary info from HR
- The HR Faculty Load process, PEPFACL, updates HR with contact hours and FTE from Faculty Load

146

Finance Interface Setup

- NTRFINI - HR/Finance Setup
  - FTVCOAS - Chart of Accounts
  - FTVFUND - Fund
  - FTVORGN - Organization
  - FTVACCT - Account
  - FTVPROG - Program

147

Note: Finance Items To Check

- NTRFINI - Be sure that all Rule codes and Net Distribution FOAPAL elements are in place
- PTREARN - Be sure that all Earn codes have Labor Distribution Overrides (optional)
- PTRBDCA - Be sure that all Deduction codes have Labor Distribution Overrides (mandatory)
- FTMRUCL - Be sure that all ‘H***’ Rule codes with ‘G026’ Process codes have Payroll Clearing Account in Posting modifier

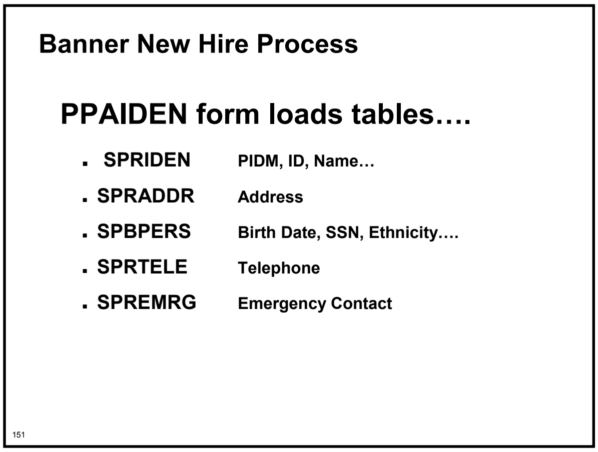
148

Finance Interface Processes

- NBAPOSN and NBAPBUD –
  build position budgets
- NBPBUDM and NHPFIN1 –
  run to build budget and encumbrance transactions
- PHPFEXP –
  run to build payroll expense transactions
- NHPFIN2 –
  insert transactions into GURFEED

149

Key HR Processes

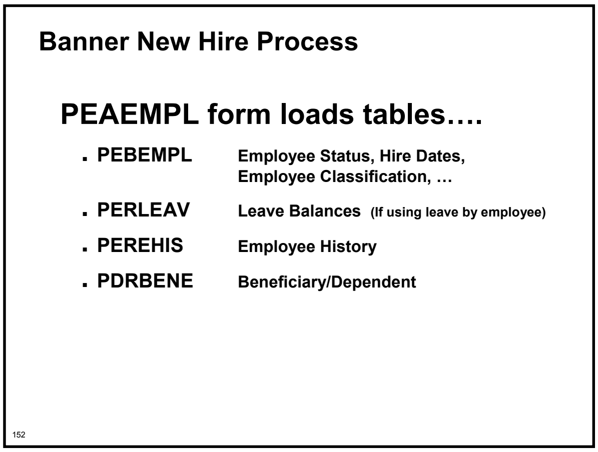


Banner New Hire Process

PPAIDEN form loads tables….

SPRIDEN	PIDM, ID, Name…
SPRADDR	Address
SPBPERS	Birth Date, SSN, Ethnicity….
SPRTELE	Telephone
SPREMRG	Emergency Contact

151

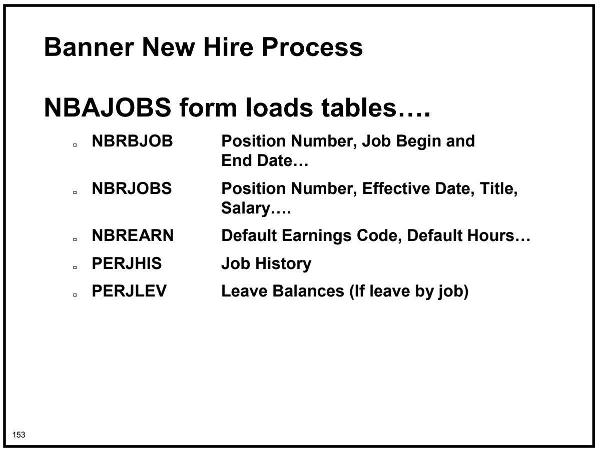


Banner New Hire Process

PEAEMPL form loads tables….

PEBEMPL	Employee Status, Hire Dates, Employee Classification, …
PERLEAV	Leave Balances (If using leave by employee)
PEREHIS	Employee History
PDRBENE	Beneficiary/Dependent

152



Banner New Hire Process

NBAJOBS form loads tables….

NBRBJOB	Position Number, Job Begin and End Date…
NBRJOBS	Position Number, Effective Date, Title, Salary….
NBREARN	Default Earnings Code, Default Hours…
PERJHIS	Job History
PERJLEV	Leave Balances (If leave by job)

153

Banner New Hire Process

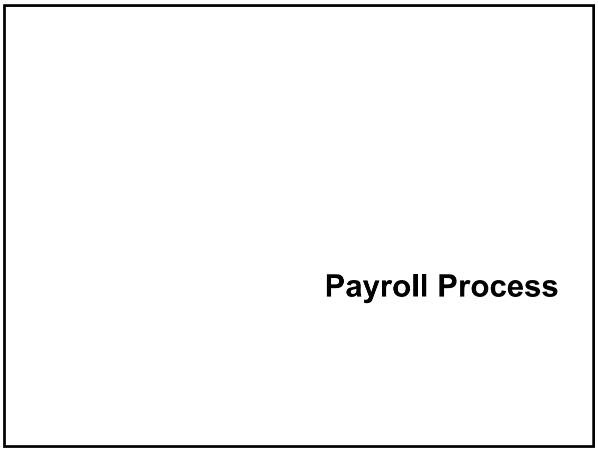
PDADEDN form loads tables…

PDRBDED	Deduction Code (BDCA), Deduction Begin Date and End Date….
PDRDEDN	Deduction Code (BDCA), Effective Date, Deduction Plan, Deduction Amounts…
PERDHIS	Deduction History

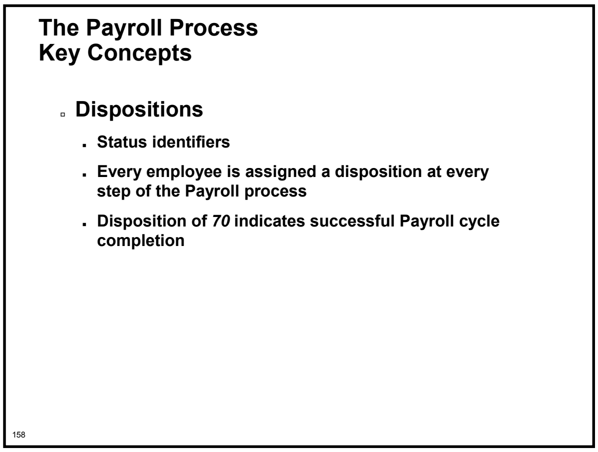
Banner New Hire Process

- NEW HIRE QUICK SET UP
  - PEAHIRE
    - Allows you to move through a series of steps that will load tables behind PPAIDEN, PEAEMPL, and NBAJOBS
    - Initial Set up only
  - PDABDSU
    - Loads the tables behind PDADEDN
    - Initial Set up only

Exercise #4



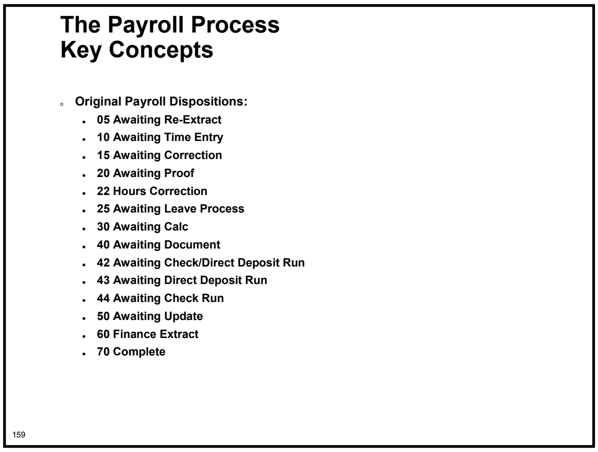
Payroll Process



The Payroll Process Key Concepts

- **Dispositions**
  - **Status identifiers**
  - **Every employee is assigned a disposition at every step of the Payroll process**
  - **Disposition of *70* indicates successful Payroll cycle completion**

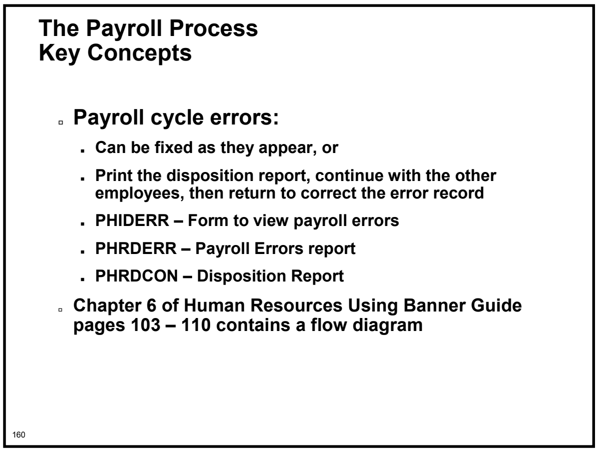
158



The Payroll Process Key Concepts

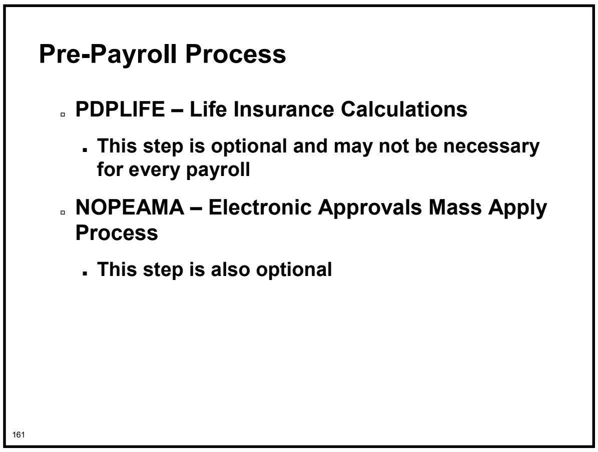
- **Original Payroll Dispositions:**
  - **05 Awaiting Re-Extract**
  - **10 Awaiting Time Entry**
  - **15 Awaiting Correction**
  - **20 Awaiting Proof**
  - **22 Hours Correction**
  - **25 Awaiting Leave Process**
  - **30 Awaiting Calc**
  - **40 Awaiting Document**
  - **42 Awaiting Check/Direct Deposit Run**
  - **43 Awaiting Direct Deposit Run**
  - **44 Awaiting Check Run**
  - **50 Awaiting Update**
  - **60 Finance Extract**
  - **70 Complete**

159



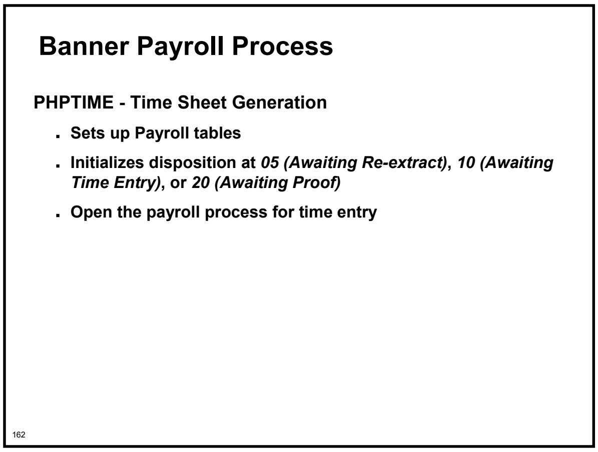
The Payroll Process
Key Concepts

- Payroll cycle errors:
  - Can be fixed as they appear, or
  - Print the disposition report, continue with the other employees, then return to correct the error record
  - PHIDERR – Form to view payroll errors
  - PHRDERR – Payroll Errors report
  - PHRDCON – Disposition Report
- Chapter 6 of Human Resources Using Banner Guide pages 103 – 110 contains a flow diagram



Pre-Payroll Process

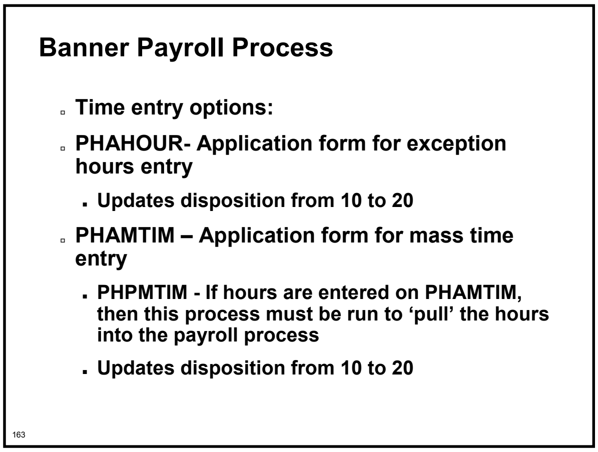
- PDPLIFE – Life Insurance Calculations
  - This step is optional and may not be necessary for every payroll
- NOPEAMA – Electronic Approvals Mass Apply Process
  - This step is also optional



Banner Payroll Process

PHPTIME - Time Sheet Generation

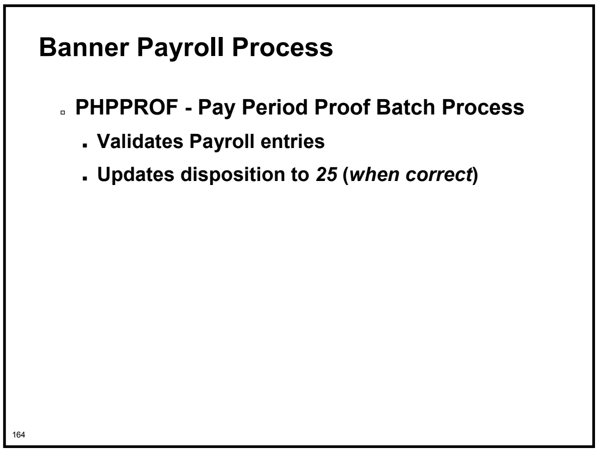
- Sets up Payroll tables
- Initializes disposition at *05 (Awaiting Re-extract)*, *10 (Awaiting Time Entry)*, or *20 (Awaiting Proof)*
- Open the payroll process for time entry



Banner Payroll Process

- Time entry options:
- PHAHOUR- Application form for exception hours entry
  - Updates disposition from 10 to 20
- PHAMTIM – Application form for mass time entry
  - PHPMTIM - If hours are entered on PHAMTIM, then this process must be run to ‘pull’ the hours into the payroll process
  - Updates disposition from 10 to 20

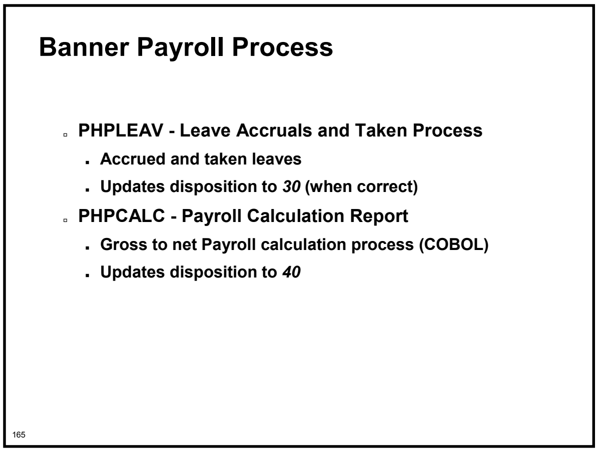
163



Banner Payroll Process

- PHPPROF - Pay Period Proof Batch Process
  - Validates Payroll entries
  - Updates disposition to *25* (*when correct*)

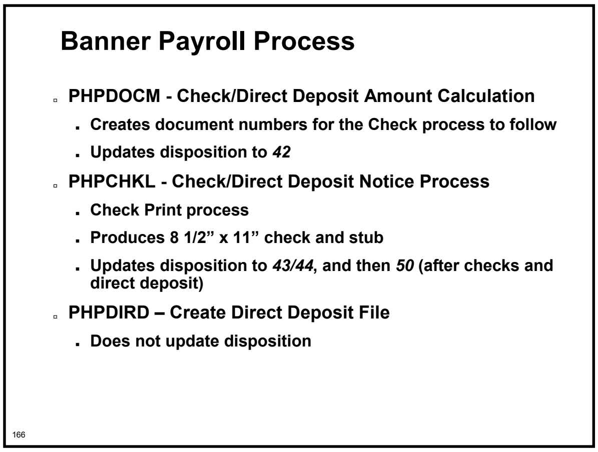
164



Banner Payroll Process

- PHPLEAV - Leave Accruals and Taken Process
  - Accrued and taken leaves
  - Updates disposition to *30* (when correct)
- PHPCALC - Payroll Calculation Report
  - Gross to net Payroll calculation process (COBOL)
  - Updates disposition to *40*

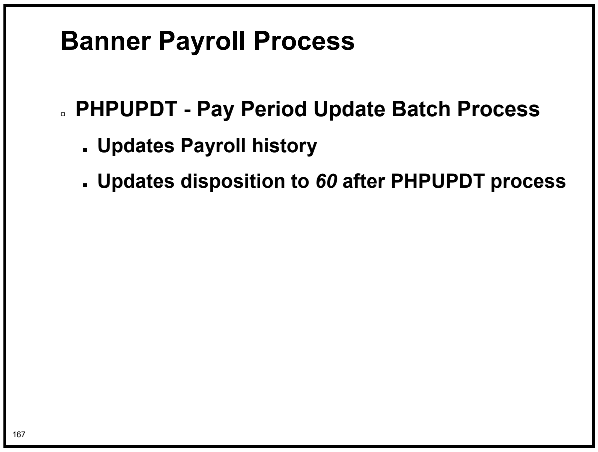
165



Banner Payroll Process

- PHPDOCM - Check/Direct Deposit Amount Calculation
  - Creates document numbers for the Check process to follow
  - Updates disposition to *42*
- PHPCHKL - Check/Direct Deposit Notice Process
  - Check Print process
  - Produces 8 1/2” x 11” check and stub
  - Updates disposition to *43/44*, and then *50* (after checks and direct deposit)
- PHPDIRD – Create Direct Deposit File
  - Does not update disposition

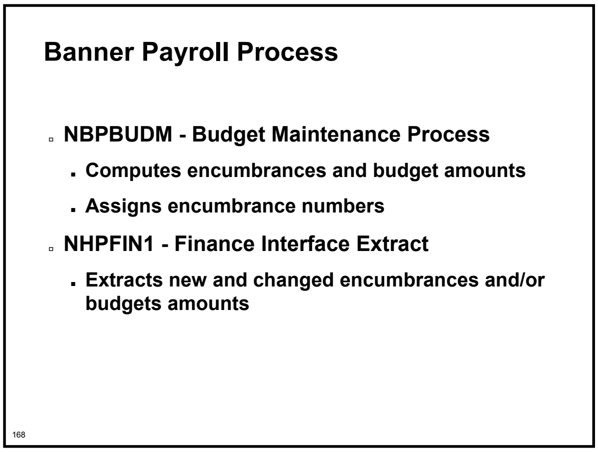
166



Banner Payroll Process

- PHPUPDT - Pay Period Update Batch Process
  - Updates Payroll history
  - Updates disposition to *60* after PHPUPDT process

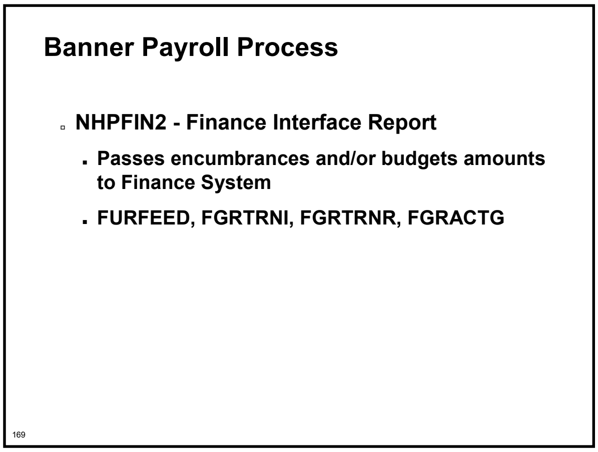
167



Banner Payroll Process

- NBPBUDM - Budget Maintenance Process
  - Computes encumbrances and budget amounts
  - Assigns encumbrance numbers
- NHPFIN1 - Finance Interface Extract
  - Extracts new and changed encumbrances and/or budgets amounts

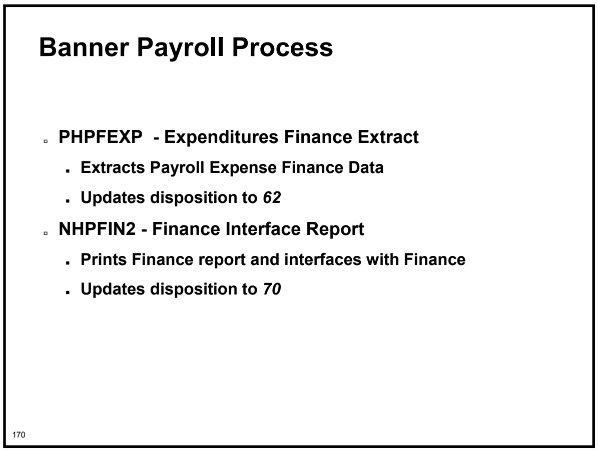
168



Banner Payroll Process

- NHPFIN2 - Finance Interface Report
  - Passes encumbrances and/or budgets amounts to Finance System
  - FURFEED, FGRTRNI, FGRTRNR, FGRACTG

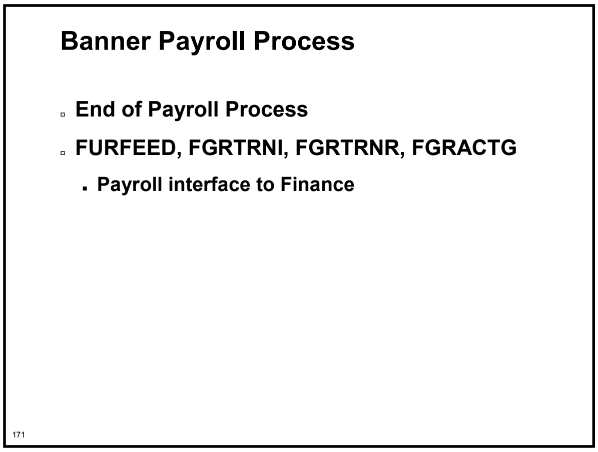
169



Banner Payroll Process

- PHPFEXP - Expenditures Finance Extract
  - Extracts Payroll Expense Finance Data
  - Updates disposition to *62*
- NHPFIN2 - Finance Interface Report
  - Prints Finance report and interfaces with Finance
  - Updates disposition to *70*

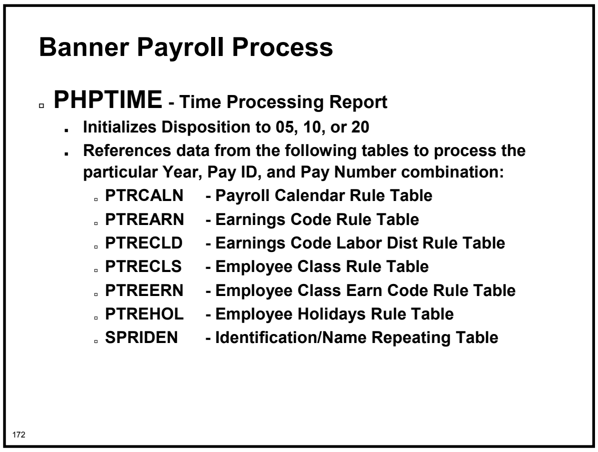
170



Banner Payroll Process

- End of Payroll Process
- FURFEED, FGRTRNI, FGRTRNR, FGRACTG
  - Payroll interface to Finance

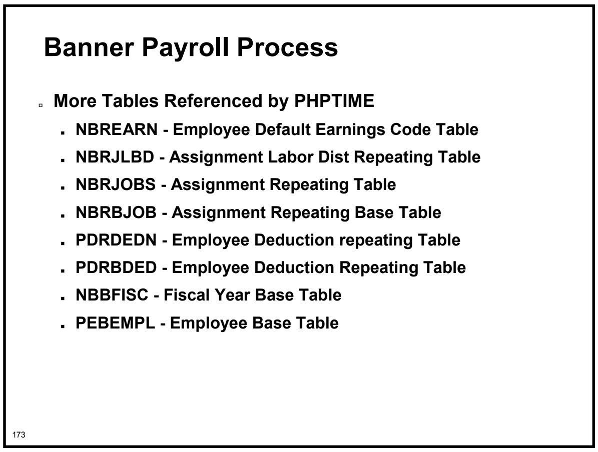
171



Banner Payroll Process

- **PHPTIME** - Time Processing Report
  - Initializes Disposition to 05, 10, or 20
  - References data from the following tables to process the particular Year, Pay ID, and Pay Number combination:
    - PTRCALN - Payroll Calendar Rule Table
    - PTREARN - Earnings Code Rule Table
    - PTRECLD - Earnings Code Labor Dist Rule Table
    - PTRECLS - Employee Class Rule Table
    - PTREERN - Employee Class Earn Code Rule Table
    - PTREHOL - Employee Holidays Rule Table
    - SPRIDEN - Identification/Name Repeating Table

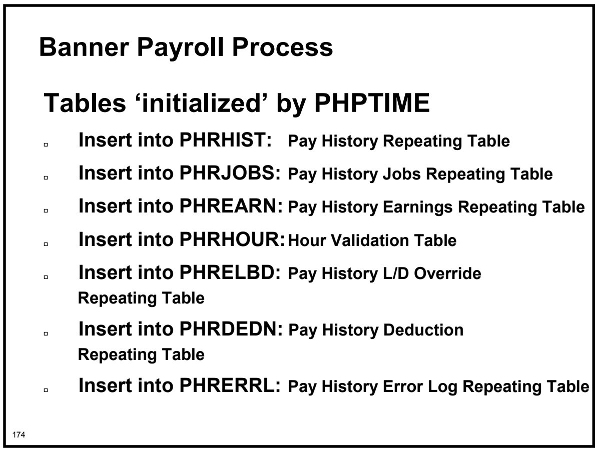
172



Banner Payroll Process

- More Tables Referenced by PHPTIME
  - NBREARN - Employee Default Earnings Code Table
  - NBRJLBD - Assignment Labor Dist Repeating Table
  - NBRJOBS - Assignment Repeating Table
  - NBRBJOB - Assignment Repeating Base Table
  - PDRDEDN - Employee Deduction repeating Table
  - PDRBDED - Employee Deduction Repeating Table
  - NBBFISC - Fiscal Year Base Table
  - PEBEMPL - Employee Base Table

173

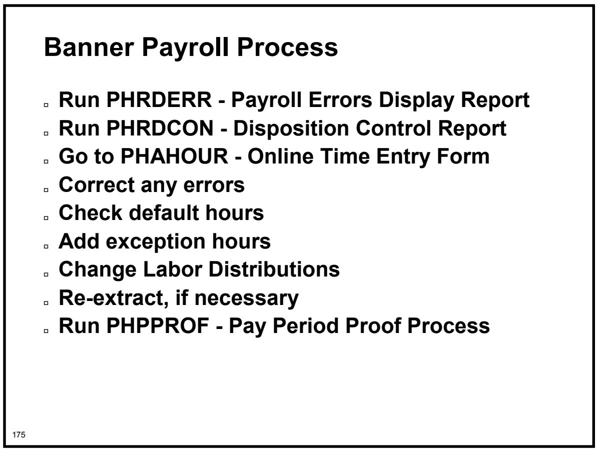


Banner Payroll Process

Tables ‘initialized’ by PHPTIME

- **Insert into PHRHIST:** Pay History Repeating Table
- **Insert into PHRJOBS:** Pay History Jobs Repeating Table
- **Insert into PHREARN:** Pay History Earnings Repeating Table
- **Insert into PHRHOUR:** Hour Validation Table
- **Insert into PHRELBD:** Pay History L/D Override Repeating Table
- **Insert into PHRDEDN:** Pay History Deduction Repeating Table
- **Insert into PHRERRL:** Pay History Error Log Repeating Table

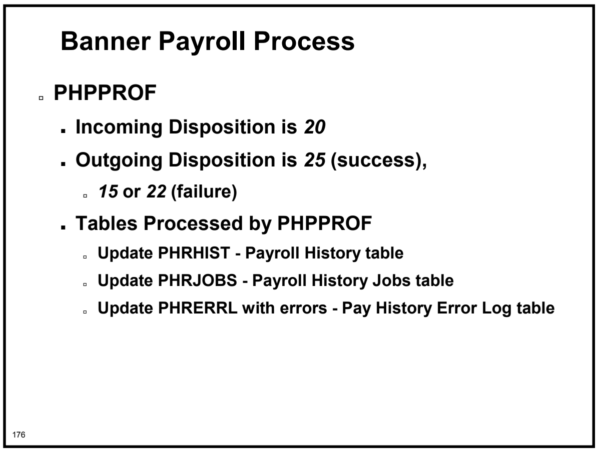
174



Banner Payroll Process

- Run PHRDERR - Payroll Errors Display Report
- Run PHRDCON - Disposition Control Report
- Go to PHAHOUR - Online Time Entry Form
- Correct any errors
- Check default hours
- Add exception hours
- Change Labor Distributions
- Re-extract, if necessary
- Run PHPPROF - Pay Period Proof Process

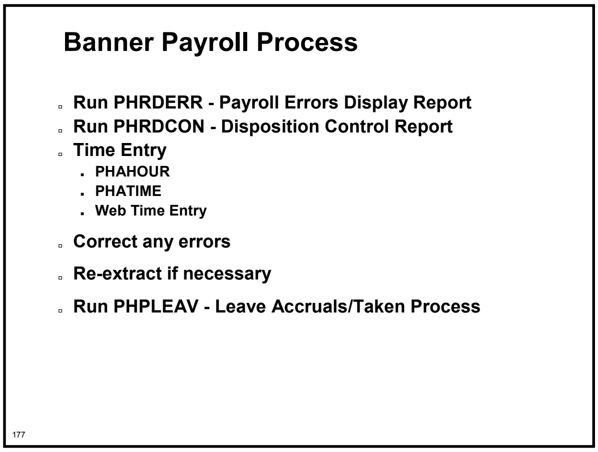
175



Banner Payroll Process

- PHPPROF
  - Incoming Disposition is *20*
  - Outgoing Disposition is *25* (success),
    - *15* or *22* (failure)
  - Tables Processed by PHPPROF
    - Update PHRHIST - Payroll History table
    - Update PHRJOBS - Payroll History Jobs table
    - Update PHRERRL with errors - Pay History Error Log table

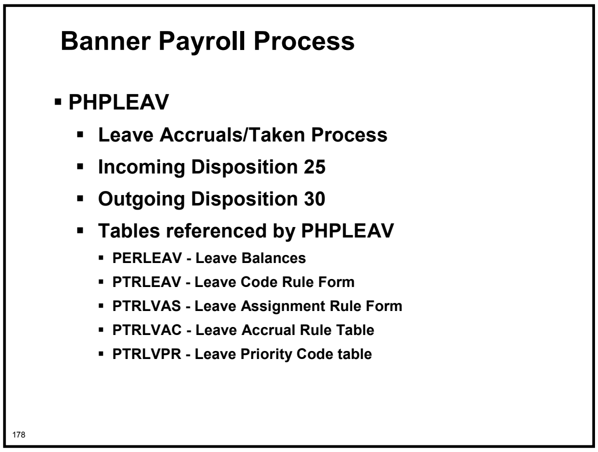
176



Banner Payroll Process

- Run PHRDERR - Payroll Errors Display Report
- Run PHRDCON - Disposition Control Report
- Time Entry
  - PHAHOUR
  - PHATIME
  - Web Time Entry
- Correct any errors
- Re-extract if necessary
- Run PHPLEAV - Leave Accruals/Taken Process

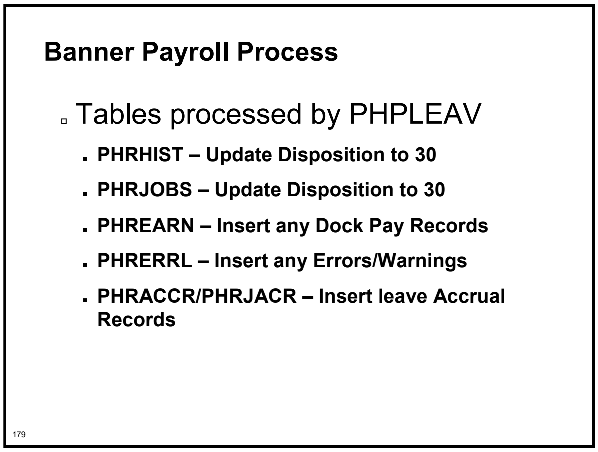
177



Banner Payroll Process

- PHPLEAV
  - Leave Accruals/Taken Process
  - Incoming Disposition 25
  - Outgoing Disposition 30
  - Tables referenced by PHPLEAV
    - PERLEAV - Leave Balances
    - PTRLEAV - Leave Code Rule Form
    - PTRLVAS - Leave Assignment Rule Form
    - PTRLVAC - Leave Accrual Rule Table
    - PTRLVPR - Leave Priority Code table

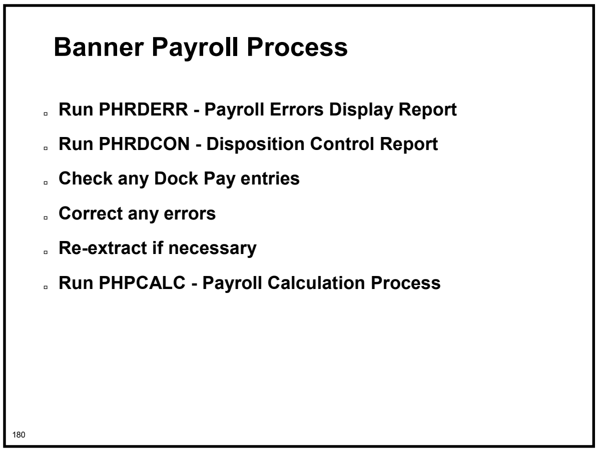
178



Banner Payroll Process

- Tables processed by PHPLEAV
  - PHRHIST – Update Disposition to 30
  - PHRJOBS – Update Disposition to 30
  - PHREARN – Insert any Dock Pay Records
  - PHRERRL – Insert any Errors/Warnings
  - PHRACCR/PHRJACR – Insert leave Accrual Records

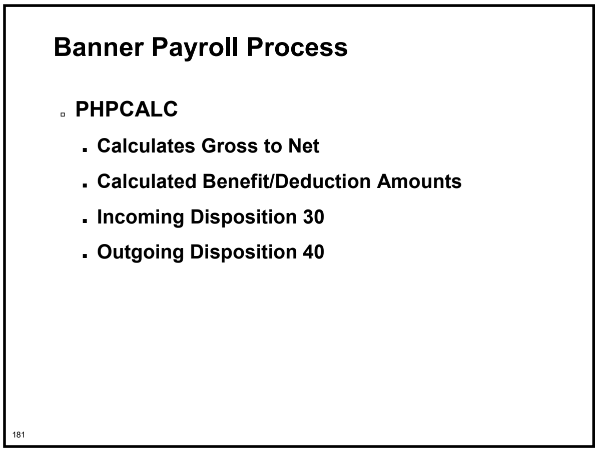
179



Banner Payroll Process

- Run PHRDERR - Payroll Errors Display Report
- Run PHRDCON - Disposition Control Report
- Check any Dock Pay entries
- Correct any errors
- Re-extract if necessary
- Run PHPCALC - Payroll Calculation Process

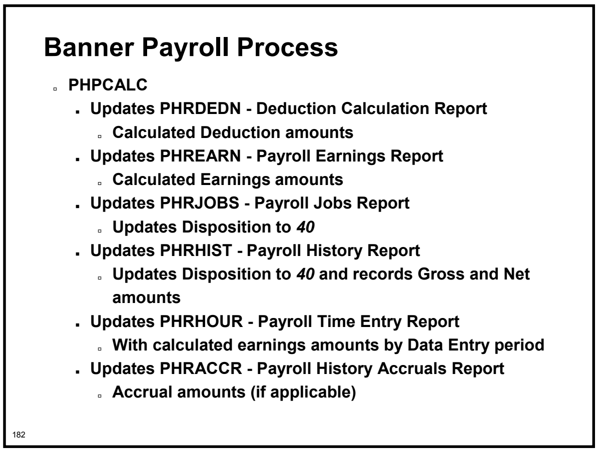
180



Banner Payroll Process

- PHPCALC
  - Calculates Gross to Net
  - Calculated Benefit/Deduction Amounts
  - Incoming Disposition 30
  - Outgoing Disposition 40

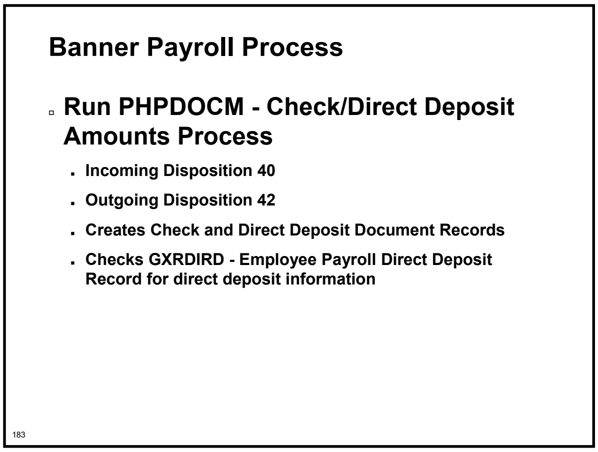
181



Banner Payroll Process

- PHPCALC
  - Updates PHRDEDN - Deduction Calculation Report
    - Calculated Deduction amounts
  - Updates PHREARN - Payroll Earnings Report
    - Calculated Earnings amounts
  - Updates PHRJOBS - Payroll Jobs Report
    - Updates Disposition to *40*
  - Updates PHRHIST - Payroll History Report
    - Updates Disposition to *40* and records Gross and Net amounts
  - Updates PHRHOUR - Payroll Time Entry Report
    - With calculated earnings amounts by Data Entry period
  - Updates PHRACCR - Payroll History Accruals Report
    - Accrual amounts (if applicable)

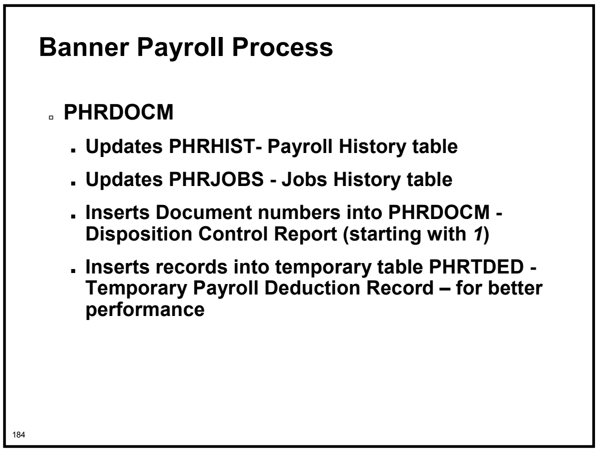
182



Banner Payroll Process

- Run PHPDOCM - Check/Direct Deposit Amounts Process
  - Incoming Disposition 40
  - Outgoing Disposition 42
  - Creates Check and Direct Deposit Document Records
  - Checks GXRDIRD - Employee Payroll Direct Deposit Record for direct deposit information

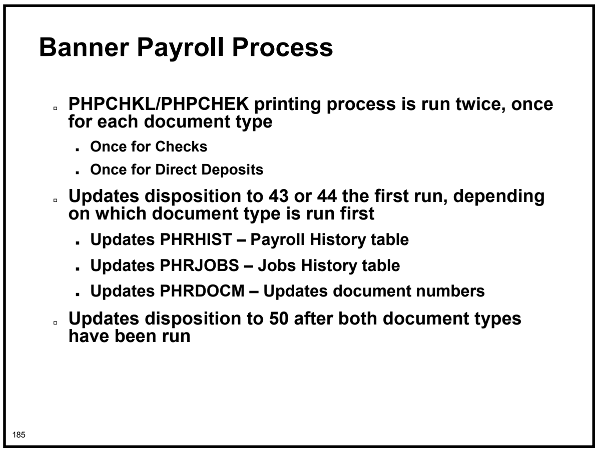
183



Banner Payroll Process

- PHRDOCM
  - Updates PHRHIST- Payroll History table
  - Updates PHRJOBS - Jobs History table
  - Inserts Document numbers into PHRDOCM - Disposition Control Report (starting with *1*)
  - Inserts records into temporary table PHRTDED - Temporary Payroll Deduction Record – for better performance

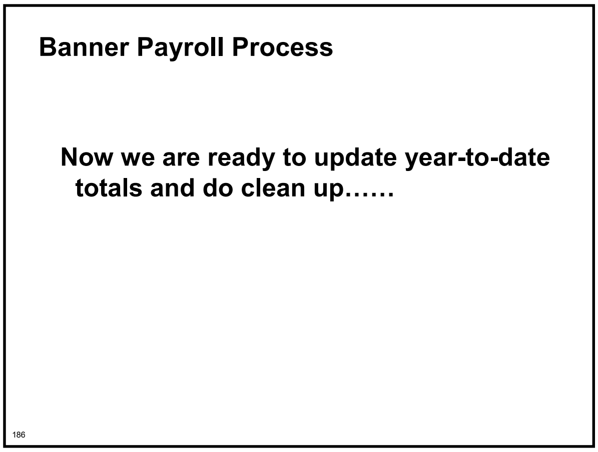
184



Banner Payroll Process

- PHPCHKL/PHPCHEK printing process is run twice, once for each document type
  - Once for Checks
  - Once for Direct Deposits
- Updates disposition to 43 or 44 the first run, depending on which document type is run first
  - Updates PHRHIST – Payroll History table
  - Updates PHRJOBS – Jobs History table
  - Updates PHRDOCM – Updates document numbers
- Updates disposition to 50 after both document types have been run

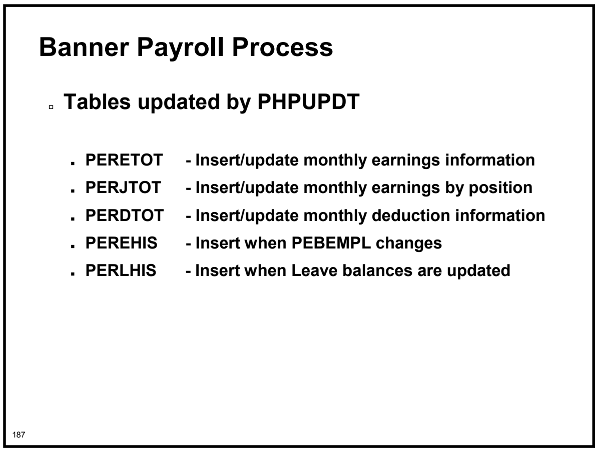
185



Banner Payroll Process

Now we are ready to update year-to-date totals and do clean up……

186

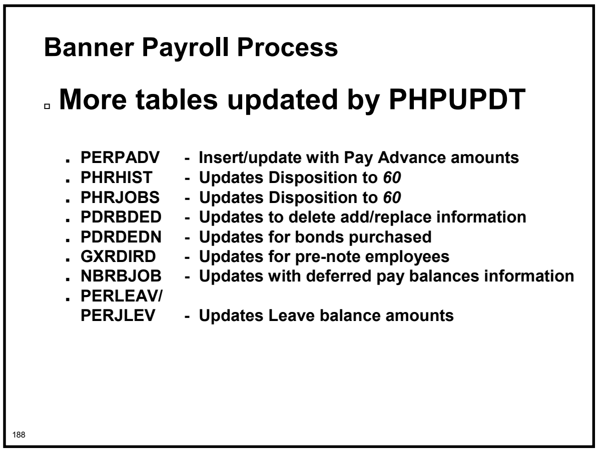


Banner Payroll Process

- Tables updated by PHPUPDT

PERETOT	- Insert/update monthly earnings information
PERJTOT	- Insert/update monthly earnings by position
PERDTOT	- Insert/update monthly deduction information
PEREHIS	- Insert when PEBEMPL changes
PERLHIS	- Insert when Leave balances are updated

187

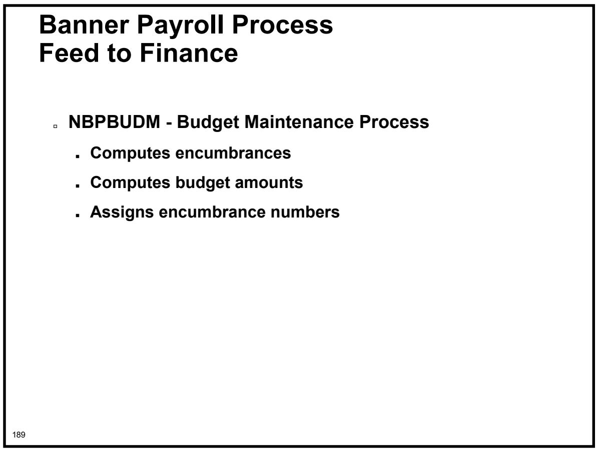


Banner Payroll Process

- More tables updated by PHPUPDT

PERPADV	- Insert/update with Pay Advance amounts
PHRHIST	- Updates Disposition to *60*
PHRJOBS	- Updates Disposition to *60*
PDRBDED	- Updates to delete add/replace information
PDRDEDN	- Updates for bonds purchased
GXRDIRD	- Updates for pre-note employees
NBRBJOB	- Updates with deferred pay balances information
PERLEAV/ PERJLEV	- Updates Leave balance amounts

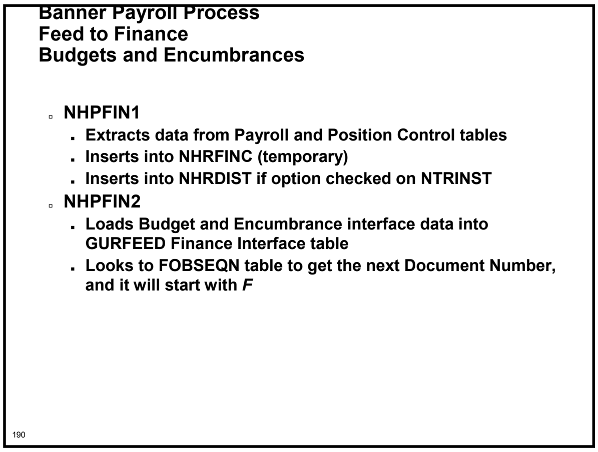
188



Banner Payroll Process
Feed to Finance

- NBPBUDM - Budget Maintenance Process
  - Computes encumbrances
  - Computes budget amounts
  - Assigns encumbrance numbers

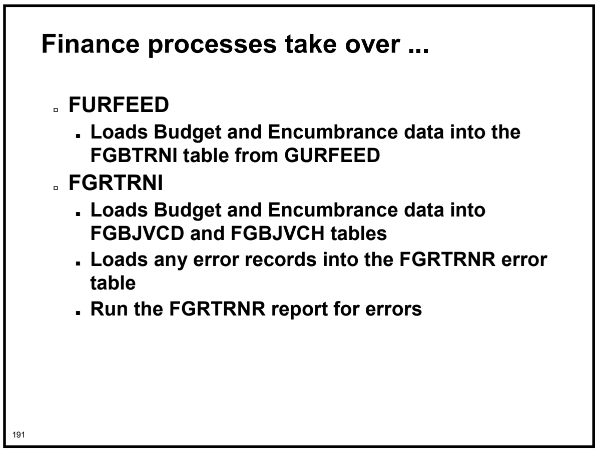
189



Banner Payroll Process
Feed to Finance
Budgets and Encumbrances

- NHPFIN1
  - Extracts data from Payroll and Position Control tables
  - Inserts into NHRFINC (temporary)
  - Inserts into NHRDIST if option checked on NTRINST
- NHPFIN2
  - Loads Budget and Encumbrance interface data into GURFEED Finance Interface table
  - Looks to FOBSEQN table to get the next Document Number, and it will start with *F*

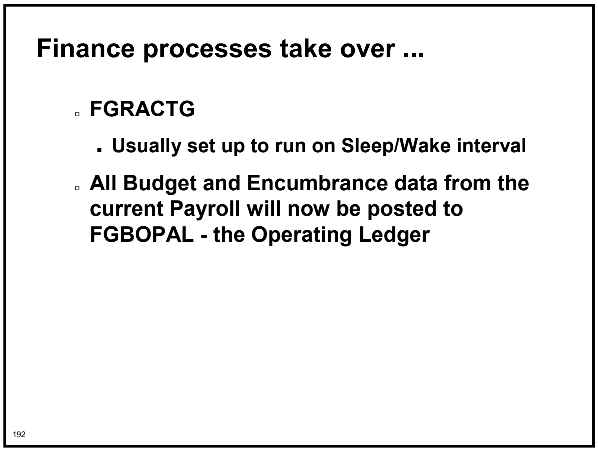
190



Finance processes take over ...

- FURFEED
  - Loads Budget and Encumbrance data into the FGBTRNI table from GURFEED
- FGRTRNI
  - Loads Budget and Encumbrance data into FGBJVCD and FGBJVCH tables
  - Loads any error records into the FGRTRNR error table
  - Run the FGRTRNR report for errors

191



Finance processes take over ...

- FGRACTG
  - Usually set up to run on Sleep/Wake interval
- All Budget and Encumbrance data from the current Payroll will now be posted to FGBOPAL - the Operating Ledger

192

Banner Payroll Process
Feed to Finance
Payroll Expenses

- PHPFEXP
  - Extracts data from Payroll and Position Control tables
  - Inserts into NHRFINC (temporary)
  - Inserts into NHRDIST if option checked on NTRINST
  - Updates Disposition to *62*
- NHPFIN2
  - Loads Actual interface data into GURFEED - the Finance Interface table
  - Looks to FOBSEQN table to get the next Document number, and it will start with *F*

193

Finance processes take over ...

- FURFEED
  - Loads Payroll Expense data into the FGBTRNI table from GURFEED
- FGRTRNI
  - Loads Payroll Expense data into the FGBJVCD and FGBJVCH tables
  - Loads any error records into the FGRTRNR error table
  - Run the FGRTRNR report for errors

194

Finance processes take over ...

- FGRACTG
  - Usually set up to run on Sleep/Wake interval
- All Actual Payroll data from the current Payroll will now be posted to FGBOPAL - the Operating Ledger

195

Exercise #5

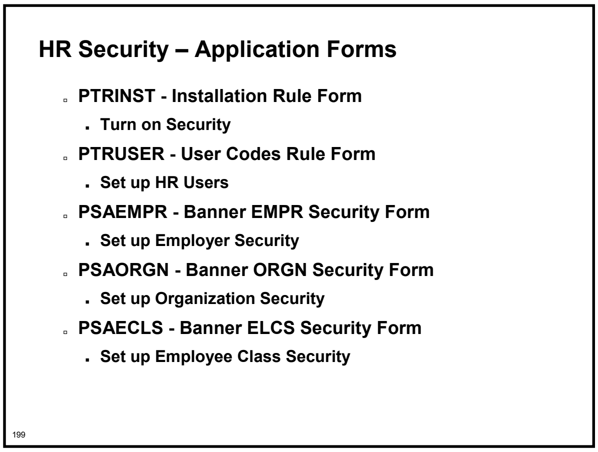
196

HR Security

HR Security

- In addition to Banner security
- Four types of security
  - Employer
  - Organization
  - Employee Class
  - Salary Level
- The types can be used in any combination
- Applies to forms only, NOT reports and processes

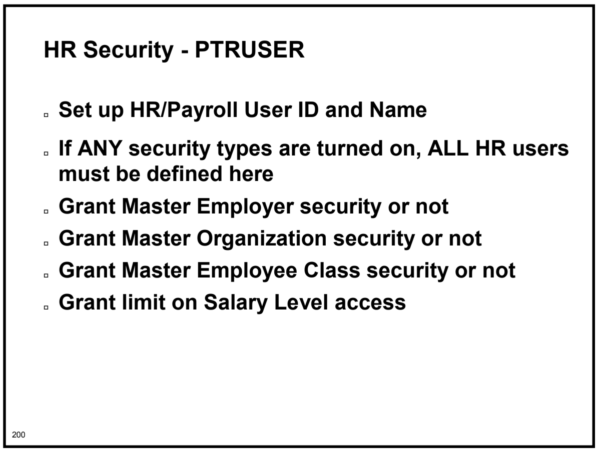
198



HR Security – Application Forms

- PTRINST - Installation Rule Form
  - Turn on Security
- PTRUSER - User Codes Rule Form
  - Set up HR Users
- PSAEMPR - Banner EMPR Security Form
  - Set up Employer Security
- PSAORGN - Banner ORGN Security Form
  - Set up Organization Security
- PSAECLS - Banner ELCS Security Form
  - Set up Employee Class Security

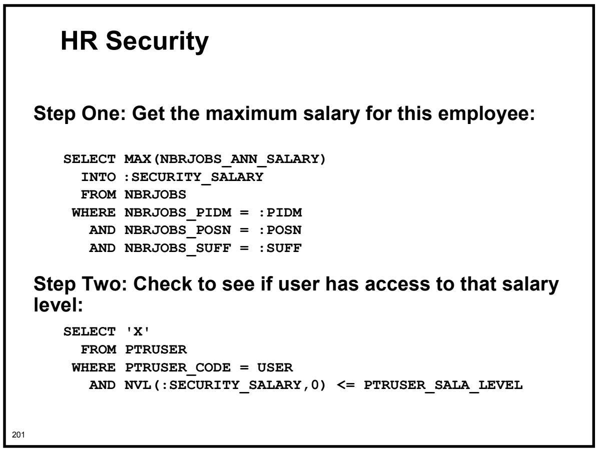
199



HR Security - PTRUSER

- Set up HR/Payroll User ID and Name
- If ANY security types are turned on, ALL HR users must be defined here
- Grant Master Employer security or not
- Grant Master Organization security or not
- Grant Master Employee Class security or not
- Grant limit on Salary Level access

200



HR Security

Step One: Get the maximum salary for this employee:

```
SELECT MAX(NBRJOBS_ANN_SALARY)
  INTO :SECURITY_SALARY
  FROM NBRJOBS
 WHERE NBRJOBS_PIDM = :PIDM
   AND NBRJOBS_POSN = :POSN
   AND NBRJOBS_SUFF = :SUFF
```

Step Two: Check to see if user has access to that salary level:

```
SELECT 'X'
  FROM PTRUSER
 WHERE PTRUSER_CODE = USER
   AND NVL(:SECURITY_SALARY,0) <= PTRUSER_SALA_LEVEL
```

201

HR Security - PSAEMPR

- Form inserts/updates/deletes rows in PSREMPR table
- For Each HR/Payroll user, enter the employer(s) that the user can view
- Used less than Organization and Employee Class security
- Works with same logic as ORGN and ECLS security

HR Security - PSAORGN

- Form inserts/updates/deletes rows in PSRORGN table
- Form allows copying of information from one user to another
- Form allows for granting access to a range of organizations - low to high organizations
- HR Org security does not use the chart hierarchy

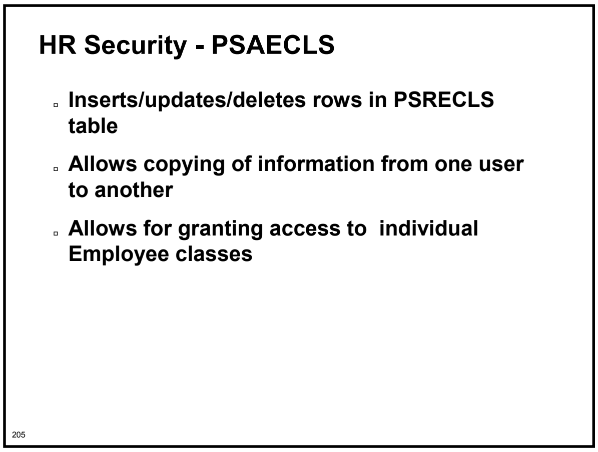
HR Security

Step One: Check to see if user has Master Authority:

```
SELECT 'X'
  FROM PTRUSER
 WHERE PTRUSER_MASTER_ORGN_IND = 'Y'
   AND PTRUSER_CODE            = USER
```

Step Two: If Step One fails, check to see if user has Specific Organization:

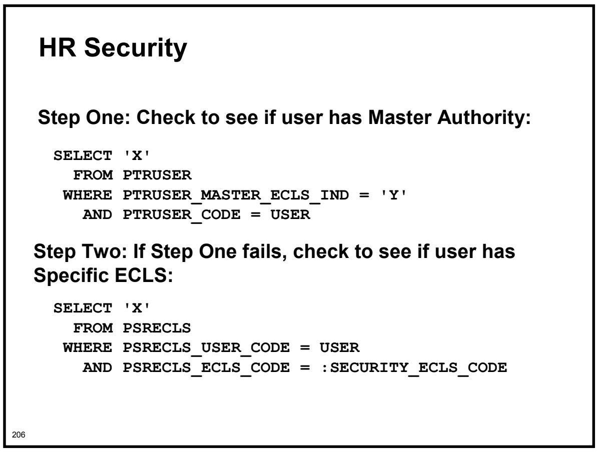
```
SELECT 'X'
  FROM PSRORGN, PTRINST
 WHERE PSRORGN_USER_CODE           = USER
   AND PTRINST_CODE                = 'PAYROLL'
   AND NVL(PTRINST_COAS_CODE,*') =
       NVL(PSRORGN_COAS_CODE,'*')
   AND :SECURITY_ORGN_CODE       >= PSRORGN_ORGN_LOW
   AND :SECURITY_ORGN_CODE       <= PSRORGN_ORGN_HIGH
```



HR Security - PSAECLS

- Inserts/updates/deletes rows in PSRECLS table
- Allows copying of information from one user to another
- Allows for granting access to individual Employee classes

205



HR Security

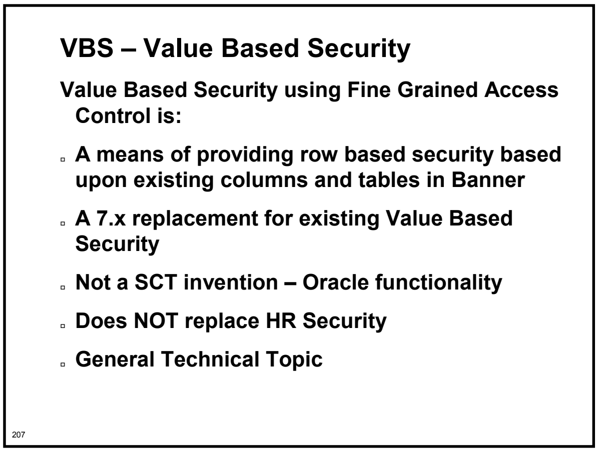
Step One: Check to see if user has Master Authority:

```
SELECT 'X'
  FROM PTRUSER
 WHERE PTRUSER_MASTER_ECLS_IND = 'Y'
   AND PTRUSER_CODE = USER
```

Step Two: If Step One fails, check to see if user has Specific ECLS:

```
SELECT 'X'
  FROM PSRECLS
 WHERE PSRECLS_USER_CODE = USER
   AND PSRECLS_ECLS_CODE = :SECURITY_ECLS_CODE
```

206



VBS – Value Based Security

Value Based Security using Fine Grained Access Control is:

- A means of providing row based security based upon existing columns and tables in Banner
- A 7.x replacement for existing Value Based Security
- Not a SCT invention – Oracle functionality
- Does NOT replace HR Security
- General Technical Topic

207

Maintenance

Maintenance

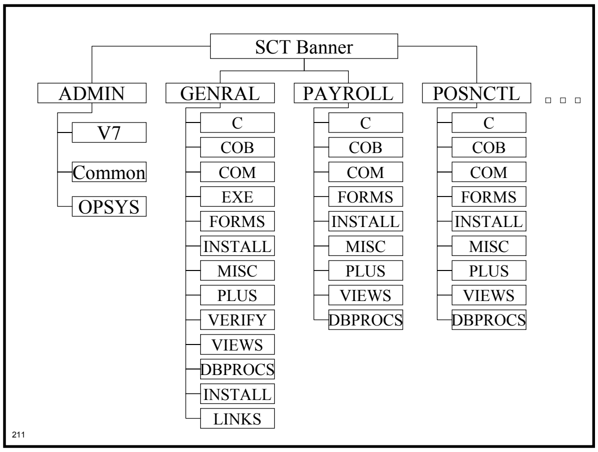
- Directory Structures
- Coding Standards
- Customizing Banner
- Reporting
- Migration
- Troubleshooting

209

Directory Structures

Where are all these forms and processes stored?

210



SCT Banner

ADMIN
- V7
- Common
- OPSYS

GENRAL
- C
- COB
- COM
- EXE
- FORMS
- INSTALL
- MISC
- PLUS
- VERIFY
- VIEWS
- DBPROCS
- INSTALL
- LINKS

PAYROLL
- C
- COB
- COM
- FORMS
- INSTALL
- MISC
- PLUS
- VIEWS
- DBPROCS

POSNCTL
- C
- COB
- COM
- FORMS
- INSTALL
- MISC
- PLUS
- VIEWS
- DBPROCS

▫ ▫ ▫

211

Directory Structures

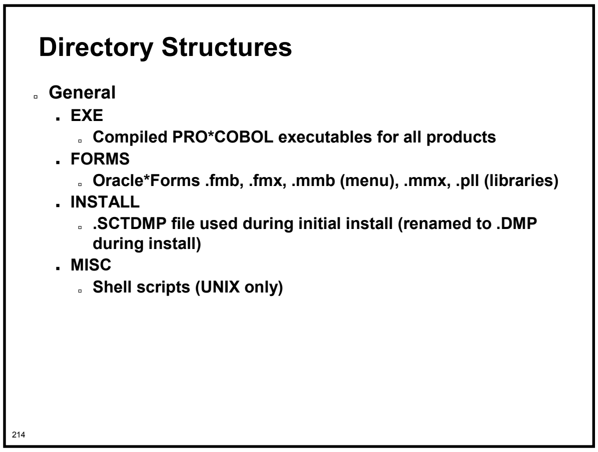
- ADMIN
  - V7
    - Scripts to create an Oracle Banner database
  - COMMON
    - Common objects shared by all products
  - OPSYS
    - Contains COBOL make files for platform (UNIX only)

212

Directory Structures

- General
  - C
    - Pro*C and C source files, C compile procedures, EXEC INCLUDE files
  - COB/COBPCO
    - Pro*COBOL files (VAX/VMS only)
    - COB/LIB
      - Links to copybooks with .cob extension and lowercase names (UNIX only)
  - COM
    - DCL command files (VAX/VMS only)

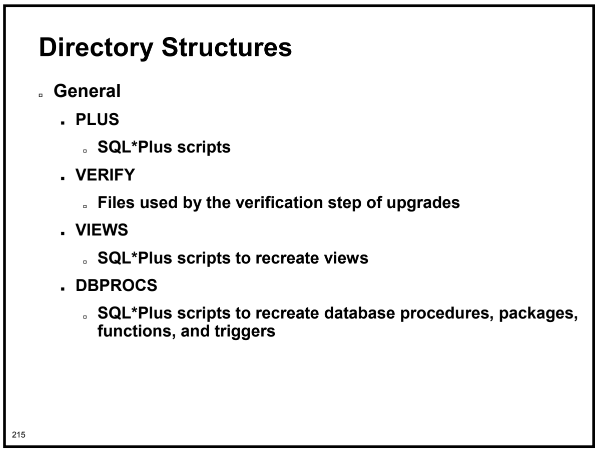
213



Directory Structures

- General
  - EXE
    - Compiled PRO*COBOL executables for all products
  - FORMS
    - Oracle*Forms .fmb, .fmx, .mmb (menu), .mmx, .pll (libraries)
  - INSTALL
    - .SCTDMP file used during initial install (renamed to .DMP during install)
  - MISC
    - Shell scripts (UNIX only)

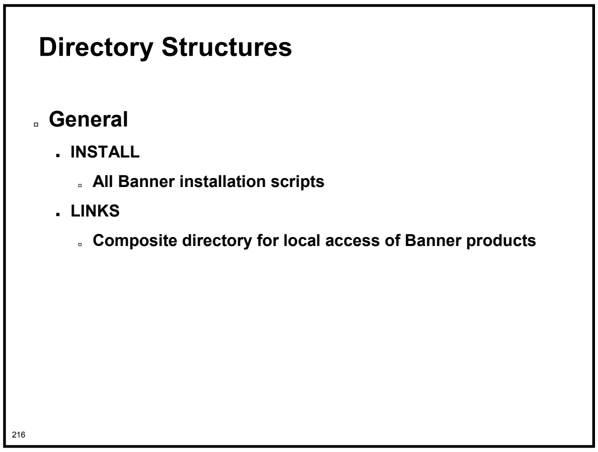
214



Directory Structures

- General
  - PLUS
    - SQL*Plus scripts
  - VERIFY
    - Files used by the verification step of upgrades
  - VIEWS
    - SQL*Plus scripts to recreate views
  - DBPROCS
    - SQL*Plus scripts to recreate database procedures, packages, functions, and triggers

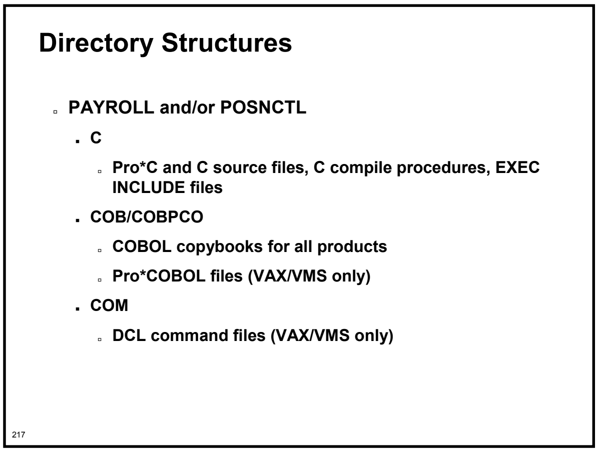
215



Directory Structures

- General
  - INSTALL
    - All Banner installation scripts
  - LINKS
    - Composite directory for local access of Banner products

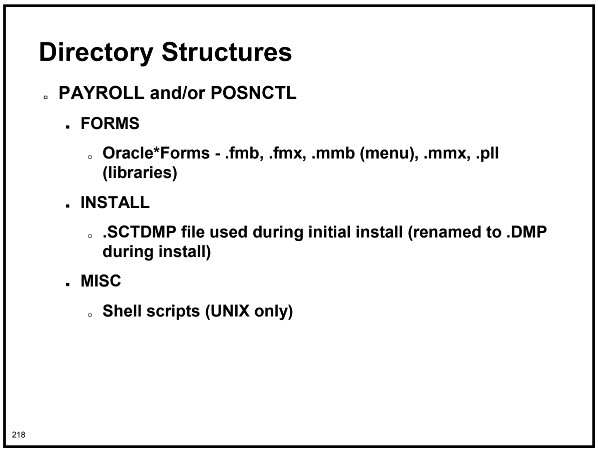
216



Directory Structures

- PAYROLL and/or POSNCTL
  - C
    - Pro*C and C source files, C compile procedures, EXEC INCLUDE files
  - COB/COBPCO
    - COBOL copybooks for all products
    - Pro*COBOL files (VAX/VMS only)
  - COM
    - DCL command files (VAX/VMS only)

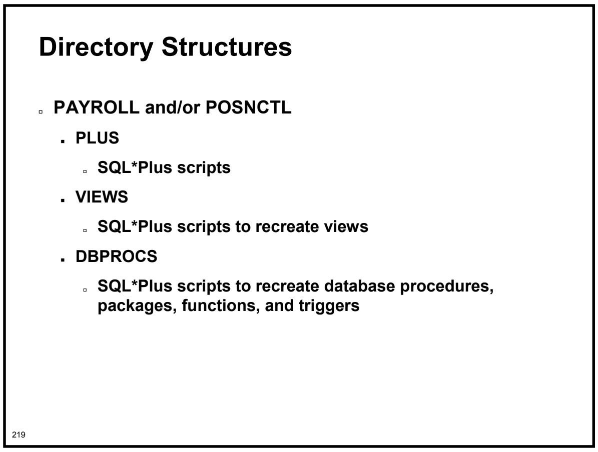
217



Directory Structures

- PAYROLL and/or POSNCTL
  - FORMS
    - Oracle*Forms - .fmb, .fmx, .mmb (menu), .mmx, .pll (libraries)
  - INSTALL
    - .SCTDMP file used during initial install (renamed to .DMP during install)
  - MISC
    - Shell scripts (UNIX only)

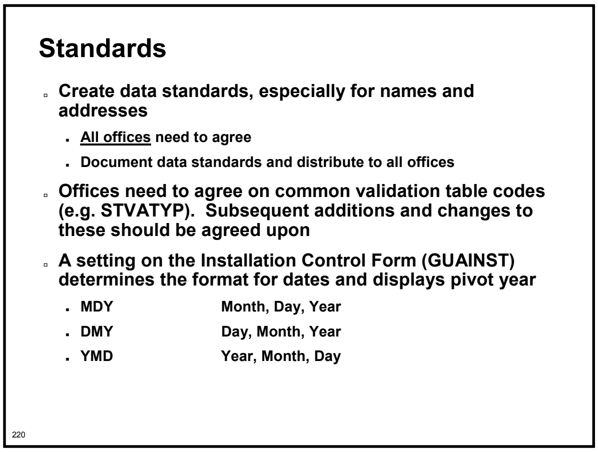
218



Directory Structures

- PAYROLL and/or POSNCTL
  - PLUS
    - SQL*Plus scripts
  - VIEWS
    - SQL*Plus scripts to recreate views
  - DBPROCS
    - SQL*Plus scripts to recreate database procedures, packages, functions, and triggers

219

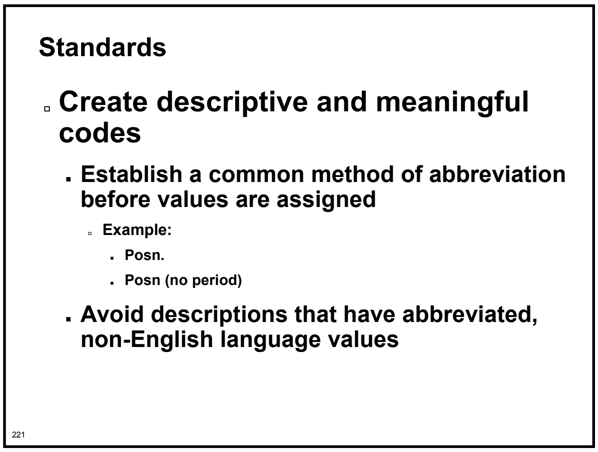


Standards

- Create data standards, especially for names and addresses
  - All offices need to agree
  - Document data standards and distribute to all offices
- Offices need to agree on common validation table codes (e.g. STVATYP). Subsequent additions and changes to these should be agreed upon
- A setting on the Installation Control Form (GUAINST) determines the format for dates and displays pivot year

MDY	Month, Day, Year
DMY	Day, Month, Year
YMD	Year, Month, Day

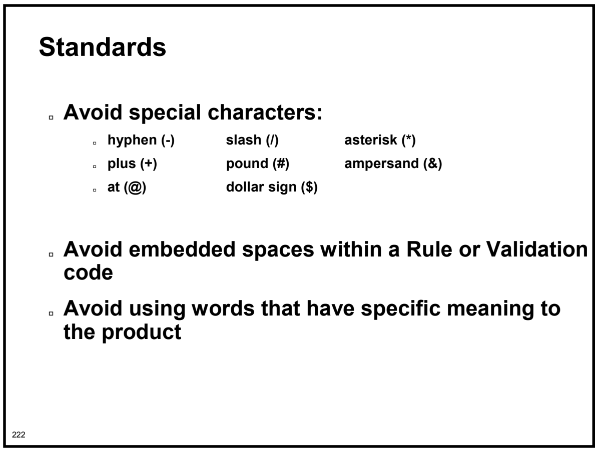
220



Standards

- Create descriptive and meaningful codes
  - Establish a common method of abbreviation before values are assigned
    - Example:
      - Posn.
      - Posn (no period)
  - Avoid descriptions that have abbreviated, non-English language values

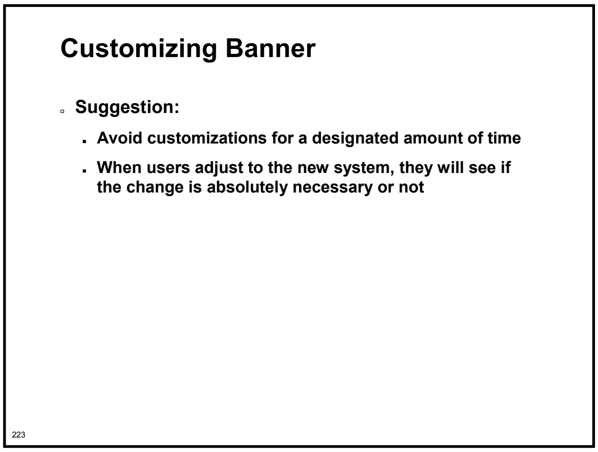
221



Standards

- Avoid special characters:
  - hyphen (-) slash (/) asterisk (*)
  - plus (+) pound (#) ampersand (&)
  - at (@) dollar sign ($)
- Avoid embedded spaces within a Rule or Validation code
- Avoid using words that have specific meaning to the product

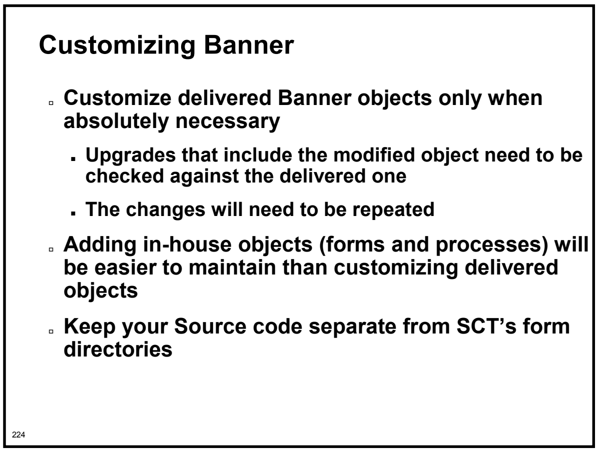
222



Customizing Banner

- Suggestion:
  - Avoid customizations for a designated amount of time
  - When users adjust to the new system, they will see if the change is absolutely necessary or not

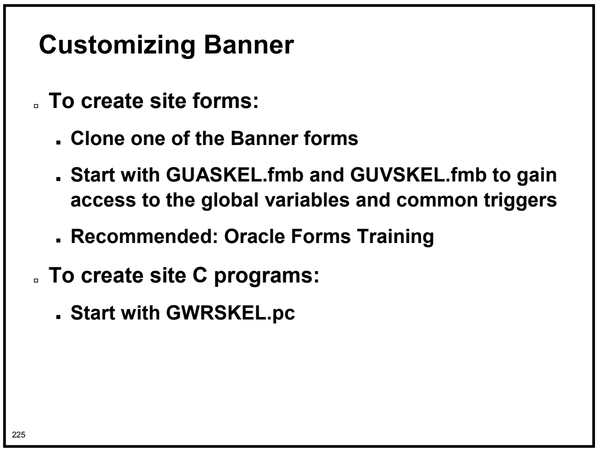
223



Customizing Banner

- Customize delivered Banner objects only when absolutely necessary
  - Upgrades that include the modified object need to be checked against the delivered one
  - The changes will need to be repeated
- Adding in-house objects (forms and processes) will be easier to maintain than customizing delivered objects
- Keep your Source code separate from SCT's form directories

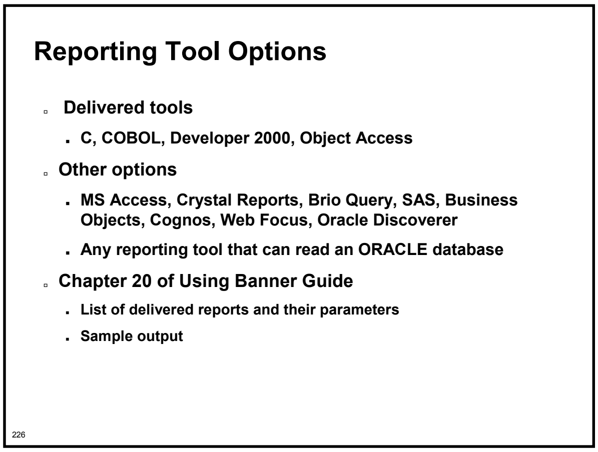
224



Customizing Banner

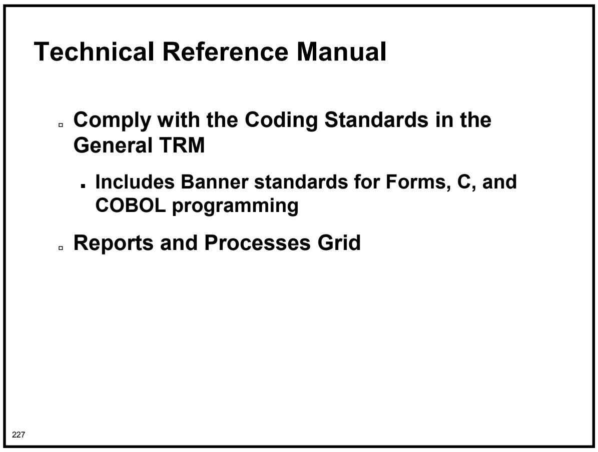
- To create site forms:
  - Clone one of the Banner forms
  - Start with GUASKEL.fmb and GUVSKEL.fmb to gain access to the global variables and common triggers
  - Recommended: Oracle Forms Training
- To create site C programs:
  - Start with GWRSKEL.pc

225



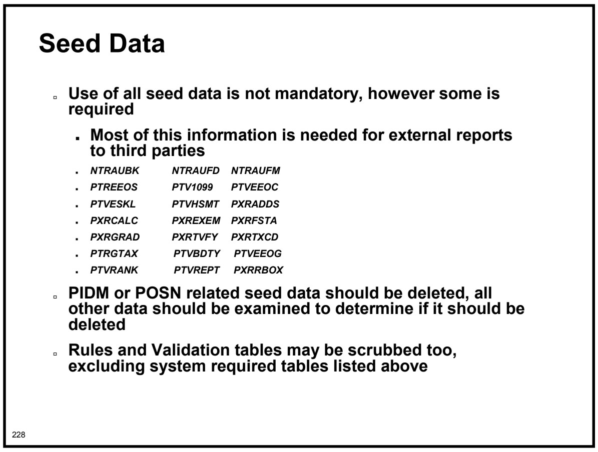
Reporting Tool Options

- Delivered tools
  - C, COBOL, Developer 2000, Object Access
- Other options
  - MS Access, Crystal Reports, Brio Query, SAS, Business Objects, Cognos, Web Focus, Oracle Discoverer
  - Any reporting tool that can read an ORACLE database
- Chapter 20 of Using Banner Guide
  - List of delivered reports and their parameters
  - Sample output



Technical Reference Manual

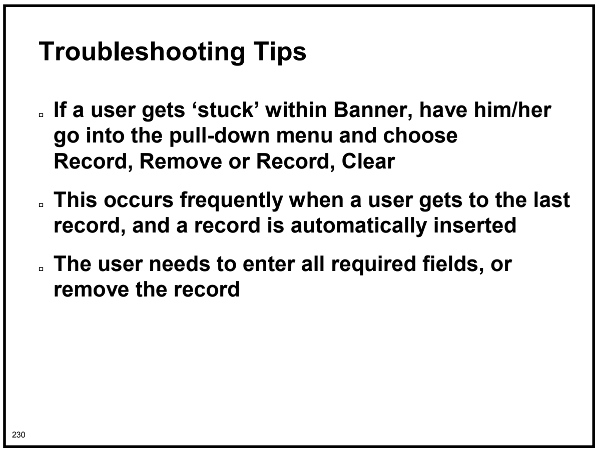
- Comply with the Coding Standards in the General TRM
  - Includes Banner standards for Forms, C, and COBOL programming
- Reports and Processes Grid



Seed Data

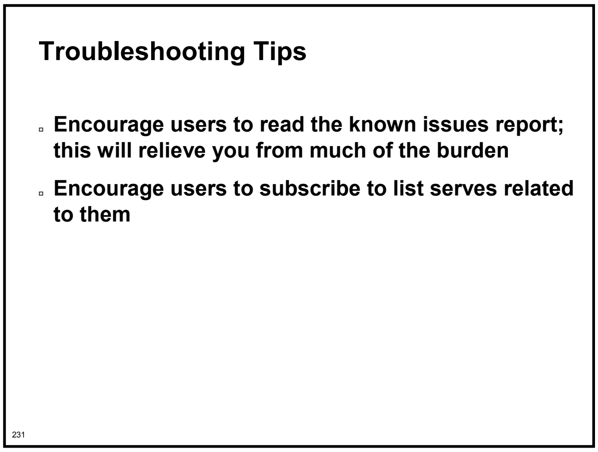
- Use of all seed data is not mandatory, however some is required
  - Most of this information is needed for external reports to third parties
  - *NTRAUBK NTRAUFD NTRAUFM*
  - *PTREEOS PTV1099 PTVEEOC*
  - *PTVESKL PTVHSMT PXRADDS*
  - *PXRCALC PXREXEM PXRFSTA*
  - *PXRGRAD PXRTVFY PXRTXCD*
  - *PTRGTAX PTVBDTY PTVEEOG*
  - *PTVRANK PTVREPT PXRRBOX*
- PIDM or POSN related seed data should be deleted, all other data should be examined to determine if it should be deleted
- Rules and Validation tables may be scrubbed too, excluding system required tables listed above

Supporting Your Users



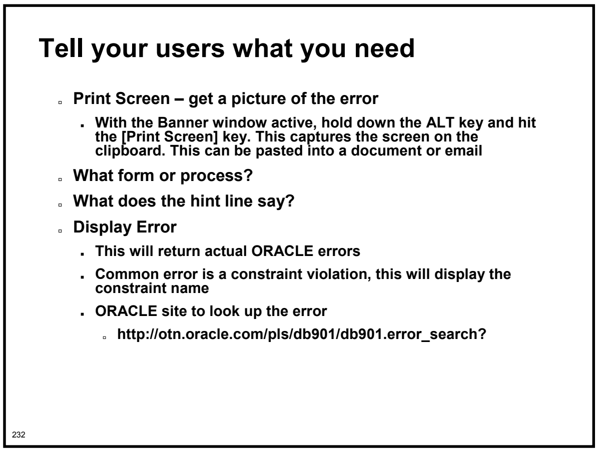
Troubleshooting Tips

- If a user gets 'stuck' within Banner, have him/her go into the pull-down menu and choose Record, Remove or Record, Clear
- This occurs frequently when a user gets to the last record, and a record is automatically inserted
- The user needs to enter all required fields, or remove the record



Troubleshooting Tips

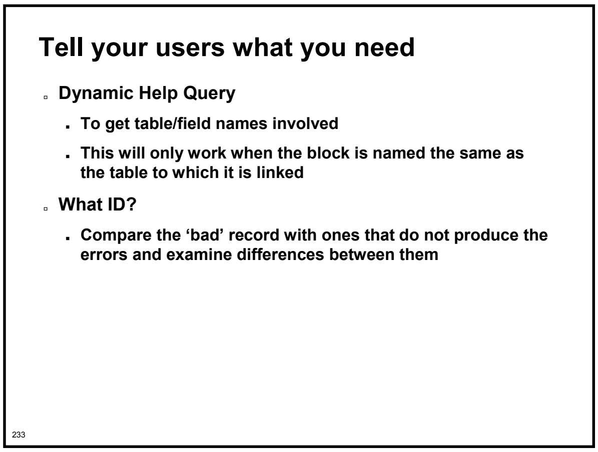
- Encourage users to read the known issues report; this will relieve you from much of the burden
- Encourage users to subscribe to list serves related to them



Tell your users what you need

- Print Screen – get a picture of the error
  - With the Banner window active, hold down the ALT key and hit the [Print Screen] key. This captures the screen on the clipboard. This can be pasted into a document or email
- What form or process?
- What does the hint line say?
- Display Error
  - This will return actual ORACLE errors
  - Common error is a constraint violation, this will display the constraint name
  - ORACLE site to look up the error
    - http://otn.oracle.com/pls/db901/db901.error_search?

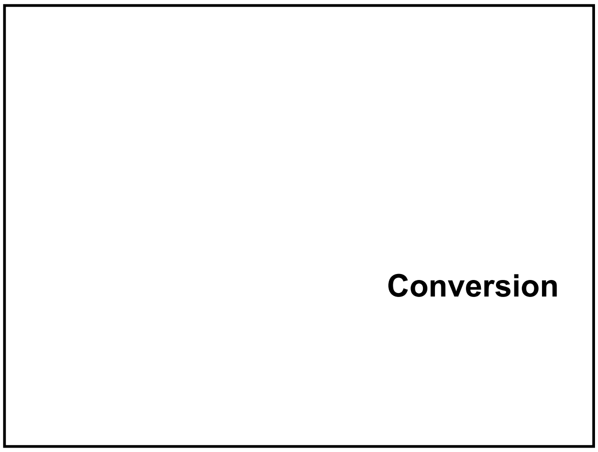
232



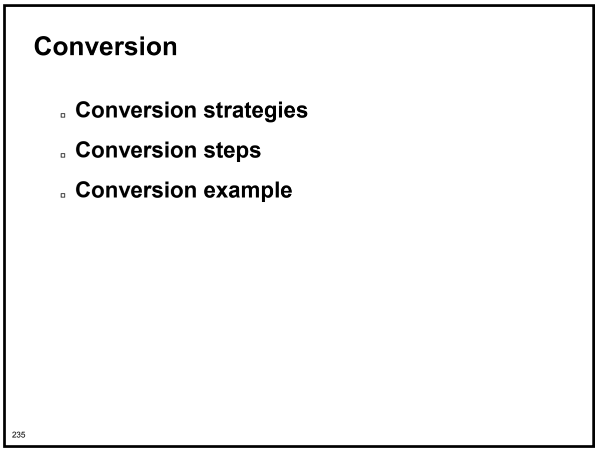
Tell your users what you need

- Dynamic Help Query
  - To get table/field names involved
  - This will only work when the block is named the same as the table to which it is linked
- What ID?
  - Compare the 'bad' record with ones that do not produce the errors and examine differences between them

233



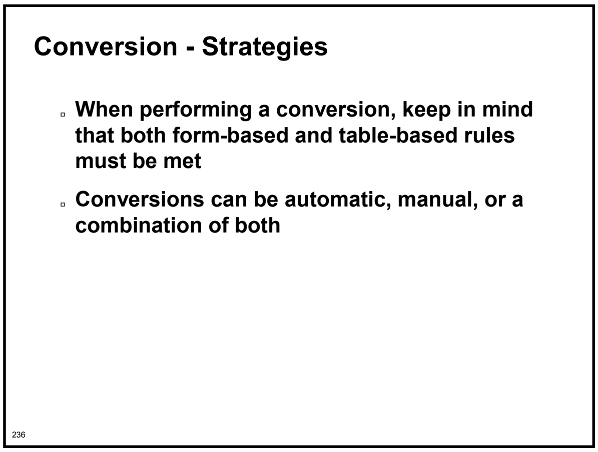
Conversion



Conversion

- Conversion strategies
- Conversion steps
- Conversion example

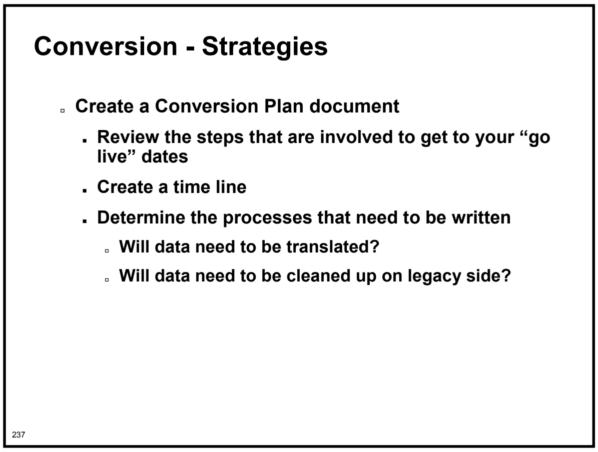
235



Conversion - Strategies

- When performing a conversion, keep in mind that both form-based and table-based rules must be met
- Conversions can be automatic, manual, or a combination of both

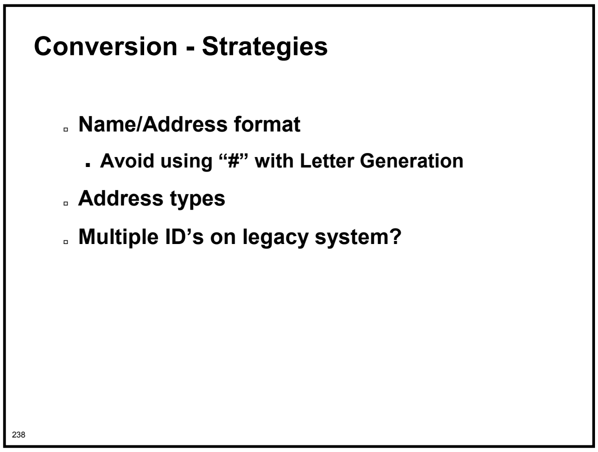
236



Conversion - Strategies

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

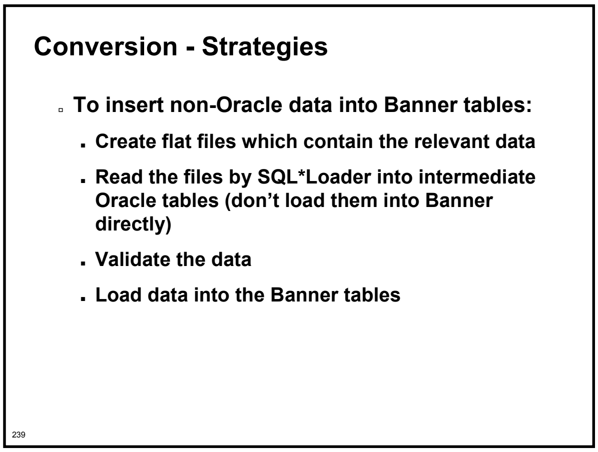
237



Conversion - Strategies

- Name/Address format
  - Avoid using “#” with Letter Generation
- Address types
- Multiple ID’s on legacy system?

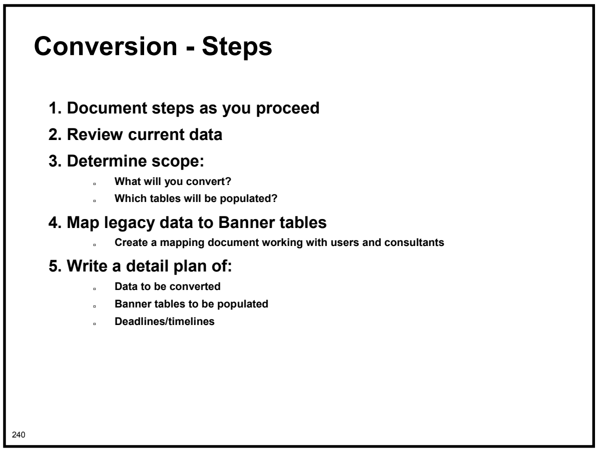
238



Conversion - Strategies

- To insert non-Oracle data into Banner tables:
  - Create flat files which contain the relevant data
  - Read the files by SQL*Loader into intermediate Oracle tables (don’t load them into Banner directly)
  - Validate the data
  - Load data into the Banner tables

239



Conversion - Steps

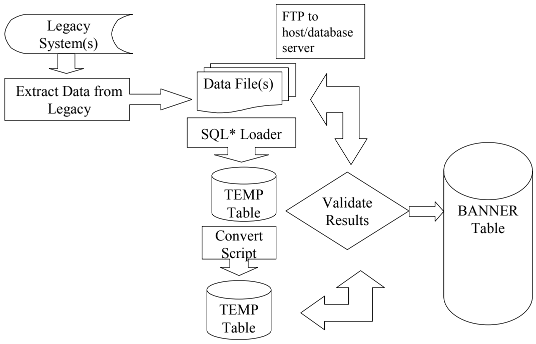
1. Document steps as you proceed
2. Review current data
3. Determine scope:
   - What will you convert?
   - Which tables will be populated?
4. Map legacy data to Banner tables
   - Create a mapping document working with users and consultants
5. Write a detail plan of:
   - Data to be converted
   -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

240

Conversion - Steps

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

Conversion - Steps



Legacy System(s)

Extract Data from Legacy

FTP to host/database server

Data File(s)

SQL* Loader

TEMP Table

Convert Script

TEMP Table

Validate Results

BANNER Table

Conversion - Example

- Shows how to:
  - Convert data to three Banner tables
  - Create, drop, and alter temporary tables
  - Assign a PIDM
  - Use SQL*LOADER to load temporary tables
  - Use UPDATE statement and DECODE function to do cross-walk (translation)

Conversion - Steps

- Shows how to (cont.) :
  - Use Insert statement
  - Use a Shell Script or Command procedure
  - Check the data when complete
  - Clean up data if it is incorrect

244

Conversion - Steps

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

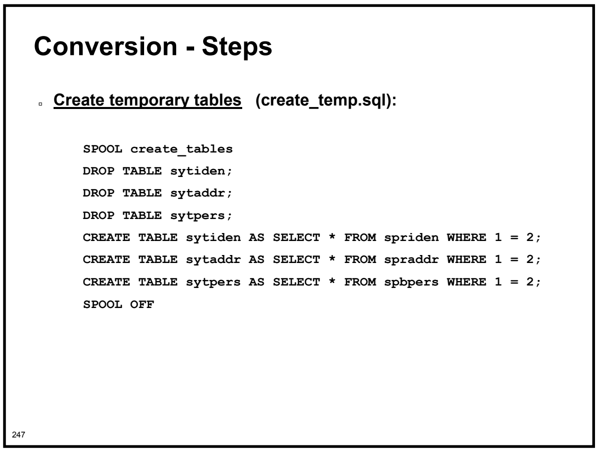
245

Conversion - Steps

- Example of a Flat File Layout

```
210009506Abbe     Anthony      PO Box 21049       Malvern        PA19355226-MAR-77
610009711Abbot    James        PO Box 27          Malvern        PA19355217-NOV-79
210009101Adams    Andrew       803 King Street    Malvern        PA19355210-DEC-72
610009101Adams    Anthony      20789 Lancaster Ln Clarksville    PA15122210-DEC-74
710000011Adams    Eugene       3400 Wendrow Way   University ParkPA16802201-JAN-01
210009619Barker   Clementine   83 Park Avenue     New York       NY10013128-APR-72
210009613Barker   James        854 Charlestown Pk King of PrussiaPA19401201-DEC-77
```

246

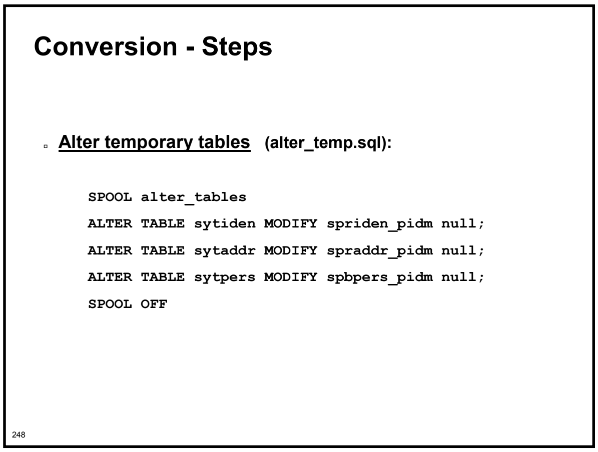


Conversion - Steps

- **Create temporary tables** **(create_temp.sql):**

```
SPOOL create_tables
DROP TABLE sytiden;
DROP TABLE sytaddr;
DROP TABLE sytpers;
CREATE TABLE sytiden AS SELECT * FROM spriden WHERE 1 = 2;
CREATE TABLE sytaddr AS SELECT * FROM spraddr WHERE 1 = 2;
CREATE TABLE sytpers AS SELECT * FROM spbpers WHERE 1 = 2;
SPOOL OFF
```

247

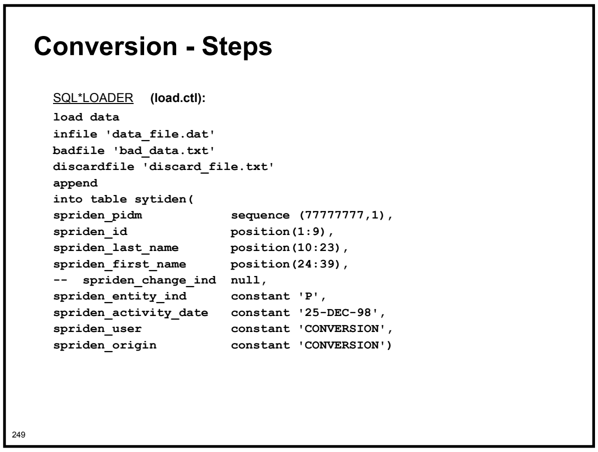


Conversion - Steps

- **Alter temporary tables** **(alter_temp.sql):**

```
SPOOL alter_tables
ALTER TABLE sytiden MODIFY spriden_pidm null;
ALTER TABLE sytaddr MODIFY spraddr_pidm null;
ALTER TABLE sytpers MODIFY spbpers_pidm null;
SPOOL OFF
```

248



Conversion - Steps

SQL*LOADER **(load.ctl):**

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

249

Conversion - Steps

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

250

Conversion - Steps

- **Decode SPBPERS_SEX (decode_sex.sql):**

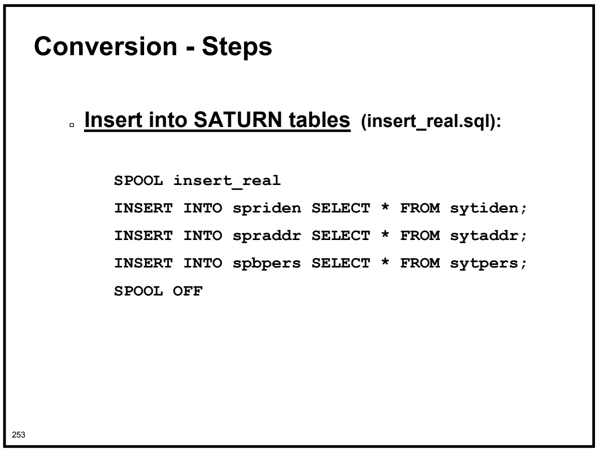
```
SPOOL decode
UPDATE sytpers
SET spbpers_sex = decode
  (spbpers_sex,'1','F','2','M','N');
SPOOL OFF
```

251

Conversion - Steps

```
  SELECT   spriden_id,
           substr(spriden_last_name,1,15)||
           ', '||spriden_first_name,
           spriden_change_ind IND,
           spriden_entity_ind ENT, spriden_activity_date,
           spriden_pidm, spraddr_pidm, spbpers_pidm,
           spraddr_street_line1, spraddr_city,
           spraddr_stat_code, spraddr_zip, spbpers_sex,
           spbpers_birth_date
    FROM   sytiden, sytaddr, sytpers
   WHERE   spriden_pidm = spraddr_pidm
     AND   spriden_pidm = spbpers_pidm
ORDER BY   spriden_pidm;
```

252

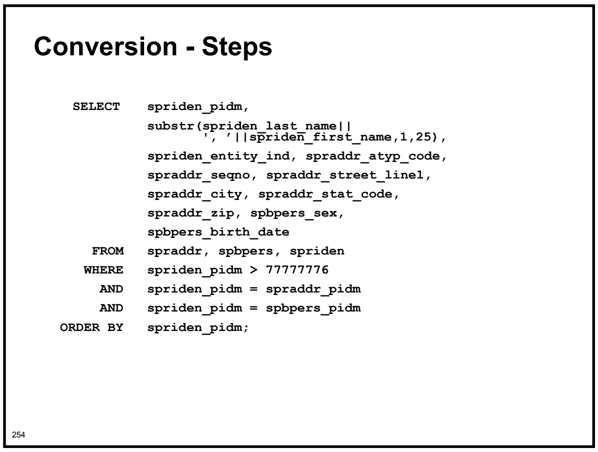


Conversion - Steps

- **Insert into SATURN tables** (insert_real.sql):

```
SPOOL insert_real
INSERT INTO spriden SELECT * FROM sytiden;
INSERT INTO spraddr SELECT * FROM sytaddr;
INSERT INTO spbpers SELECT * FROM sytpers;
SPOOL OFF
```

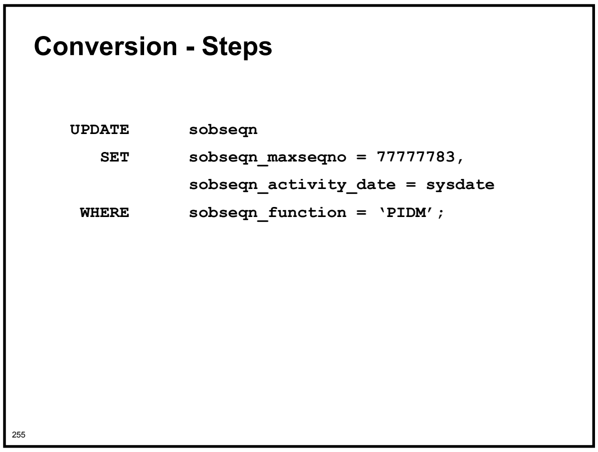
253



Conversion - Steps

```
  SELECT   spriden_pidm,
           substr(spriden_last_name||
                 ', '||spriden_first_name,1,25),
           spriden_entity_ind, spraddr_atyp_code,
           spraddr_seqno, spraddr_street_line1,
           spraddr_city, spraddr_stat_code,
           spraddr_zip, spbpers_sex,
           spbpers_birth_date
    FROM   spraddr, spbpers, spriden
   WHERE   spriden_pidm > 77777776
     AND   spriden_pidm = spraddr_pidm
     AND   spriden_pidm = spbpers_pidm
ORDER BY   spriden_pidm;
```

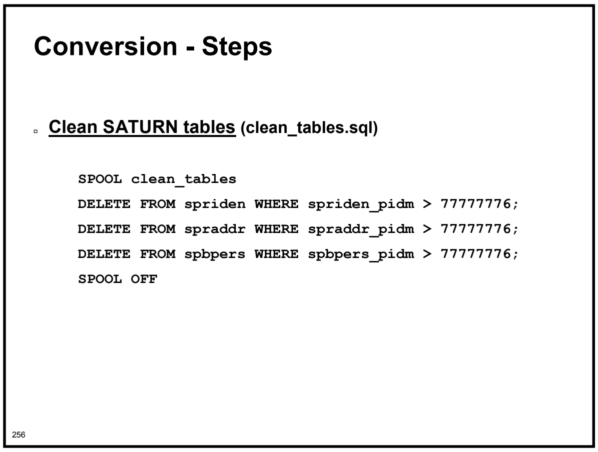
254



Conversion - Steps

```
UPDATE    sobseqn
   SET    sobseqn_maxseqno = 77777783,
          sobseqn_activity_date = sysdate
 WHERE    sobseqn_function = 'PIDM';
```

255

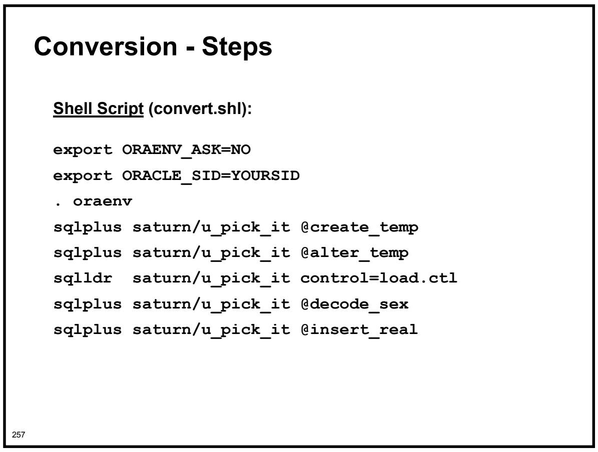


Conversion - Steps

- **Clean SATURN tables** **(clean_tables.sql)**

```
SPOOL clean_tables
DELETE FROM spriden WHERE spriden_pidm > 77777776;
DELETE FROM spraddr WHERE spraddr_pidm > 77777776;
DELETE FROM spbpers WHERE spbpers_pidm > 77777776;
SPOOL OFF
```

256

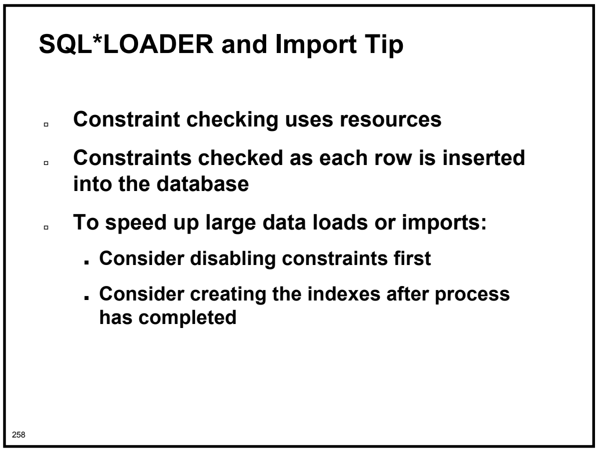


Conversion - Steps

Shell Script **(convert.shl):**

```
export ORAENV_ASK=NO
export ORACLE_SID=YOURSID
. oraenv
sqlplus saturn/u_pick_it @create_temp
sqlplus saturn/u_pick_it @alter_temp
sqlldr  saturn/u_pick_it control=load.ctl
sqlplus saturn/u_pick_it @decode_sex
sqlplus saturn/u_pick_it @insert_real
```

257



SQL*LOADER and Import Tip

- **Constraint checking uses resources**
- **Constraints checked as each row is inserted into the database**
- **To speed up large data loads or imports:**
  - **Consider disabling constraints first**
  - **Consider creating the indexes after process has completed**

258

Conversion – Summary of Steps

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

259

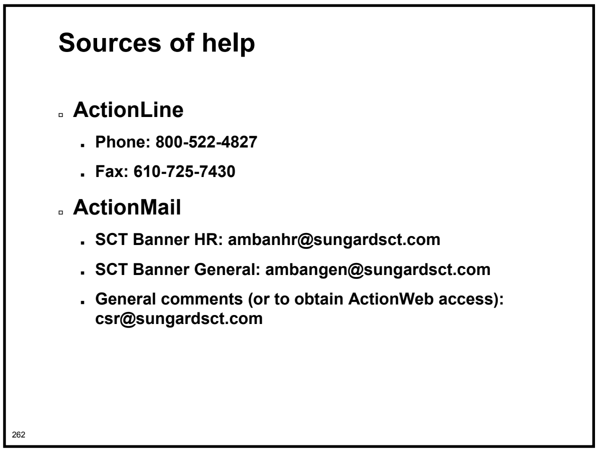
Conversion – Summary of Steps

- Translate codes using DECODE, if necessary
- Check temporary table data
- Insert temporary table data into Banner tables
- Check data in Banner (in forms and tables)
- Update sobseqn
- Use Shell Script

260

Help??

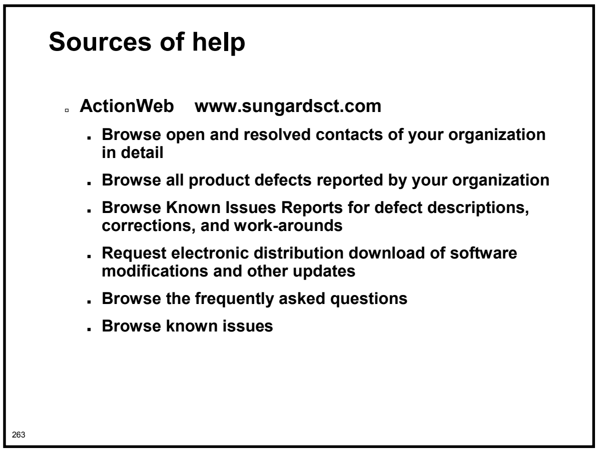
261



Sources of help

- ActionLine
  - Phone: 800-522-4827
  - Fax: 610-725-7430
- ActionMail
  - SCT Banner HR: ambanhr@sungardsct.com
  - SCT Banner General: ambangen@sungardsct.com
  - General comments (or to obtain ActionWeb access): csr@sungardsct.com

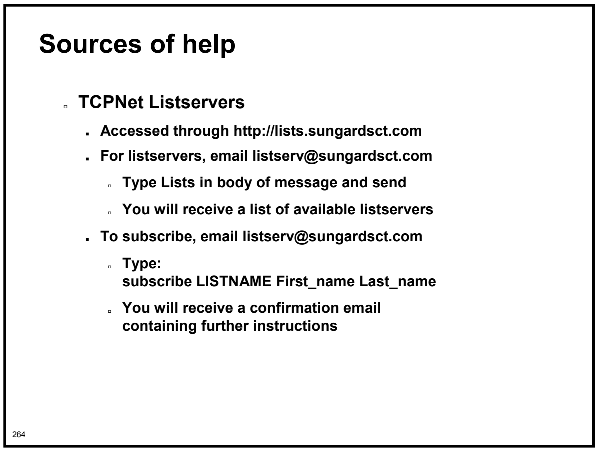
262



Sources of help

- ActionWeb www.sungardsct.com
  - Browse open and resolved contacts of your organization in detail
  - Browse all product defects reported by your organization
  - Browse Known Issues Reports for defect descriptions, corrections, and work-arounds
  - Request electronic distribution download of software modifications and other updates
  - Browse the frequently asked questions
  - Browse known issues

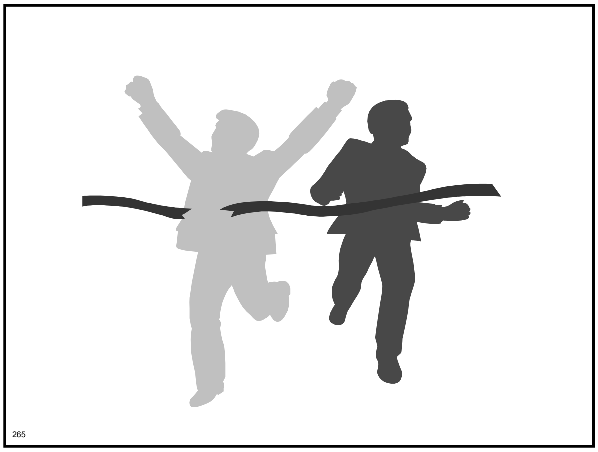
263



Sources of help

- TCPNet Listservers
  - Accessed through http://lists.sungardsct.com
  - For listservers, email listserv@sungardsct.com
    - Type Lists in body of message and send
    - You will receive a list of available listservers
  - To subscribe, email listserv@sungardsct.com
    - Type:
      subscribe LISTNAME First_name Last_name
    - You will receive a confirmation email
      containing further instructions

264


265