Assignment 3 Hands-on – CREATE tABLE, INDEXES AND ddl StatementS

***laverty@rmu.edu***

Contents

[**Introduction 4**](#_Toc482452736)

[**How to complete Hand-on Tutorial Requirements and Review Questions 4**](#_Toc482452737)

[**Oracle Error Codes Summary 5**](#_Toc482452738)

[**Windows Snipping Tool 6**](#_Toc482452739)

[**1.0 Creating a Simple Faculty, Course and Student Database 8**](#_Toc482452740)

[**1.1 Introduction to DDL SQL 8**](#_Toc482452741)

[**1.2 Conceptual Design of a Student Registration Database 9**](#_Toc482452742)

[**1.4 Oracle Table Names 10**](#_Toc482452743)

[**1.5 SQL Column Names 11**](#_Toc482452744)

[**1.6 Overview of the CREATE TABLE Command 11**](#_Toc482452745)

[**Step 1 Use Windows Notepad, or other local editor to write and edit SQL command script. 13**](#_Toc482452746)

[**Step 2. Save AS or Save the SQL Command Script to a local storage device 13**](#_Toc482452747)

[**Step 3. Use WINSCP to copy the SQL Command Script to the mediaweb.rmu.edu Linux Server 14**](#_Toc482452748)

[**Step 4 - Use PUTTY to Execute the SQL Command Script at the SQL\*Plus command prompt. 15**](#_Toc482452749)

[**1.7 Create Table Oracle Error Messages (ORA) 16**](#_Toc482452750)

[**ORA-00955: name is already used by an existing object 16**](#_Toc482452751)

[**ORA-00922 Error Missing or Invalid Option 18**](#_Toc482452752)

[**ORA-00907 Error Missing Right Parenthesis 18**](#_Toc482452753)

[**ORA-00957 Error Duplicate Column Name 20**](#_Toc482452754)

[**1.8 Document Faculty Table - SQL DESCRIBE command 21**](#_Toc482452755)

[**1.9 Adding More Clauses and Columns to the Faculty Table 22**](#_Toc482452756)

[**1 9.1 NOT NULL Clause 22**](#_Toc482452757)

[**1.9.2 Primary Key Clause 24**](#_Toc482452758)

[**1.9.3 Primary Key Constraint 24**](#_Toc482452759)

[**ORA-00907 Error Missing Right Parenthesis 25**](#_Toc482452760)

[**ORA-00904 Error Invalid Column Name 26**](#_Toc482452761)

[**1.9.4 DEFAULT Clause/Option 26**](#_Toc482452762)

[**1.10 CREATE TABLE COURSE 27**](#_Toc482452763)

[**1.10.1 Coding a Foreign Key Clause 27**](#_Toc482452764)

[**1.10.1 Foreign Key Oracle Error Messages (ORA) 28**](#_Toc482452765)

[**ORA-00904 Error Invalid Column 28**](#_Toc482452766)

[**ORA-02267 Error Column Type Incompatible 28**](#_Toc482452767)

[**ORA-00942 Error Table or View Does Not Exist 29**](#_Toc482452768)

[**ORA-02449 Error - Unique/Primary Keys in Table Referenced by Foreign Keys 29**](#_Toc482452769)

[**1.11 CREATE TABLE STUDENT 30**](#_Toc482452770)

[**1.12 CREATE TABLE SCHEDULE 31**](#_Toc482452771)

[**1.13 Referential Integrity and Foreign Keys 32**](#_Toc482452772)

[**1.14 ON DELETE CASCADE 32**](#_Toc482452773)

[**1.15 Check Constraints 34**](#_Toc482452774)

[**1.16 CHECK Constraint Errors 34**](#_Toc482452775)

[**1.17 Multiple Column Check Constraints 35**](#_Toc482452776)

[**1.17 Code and Execute the create\_rmu2.sql script 36**](#_Toc482452777)

[**2.0 Creating Indexes 38**](#_Toc482452778)

[**2.1 CREATE INDEX Examples 38**](#_Toc482452779)

[**2.2 Types of Oracle Indexes 39**](#_Toc482452780)

[**2.2 Code and Execute the create\_rmu2\_indexes.sql script 39**](#_Toc482452781)

[**3. 0 Documenting Table Metadata (Information) 40**](#_Toc482452782)

[**1. Document the execution of the DESCRIBE FACULTY command below. 40**](#_Toc482452783)

[**2. Document the execution of the DESCRIBE COURSE command below. 40**](#_Toc482452784)

[**3. Document the execution of DESCRIBE STUDENT command below. 40**](#_Toc482452785)

[**4. Document the execution of the DESCRIBE SCHEDULE command below. 41**](#_Toc482452786)

[**4.0 Oracle Data Dictionary Concepts 42**](#_Toc482452787)

[**4.1 USER\_TABLES (TABS) 43**](#_Toc482452788)

[**4.1.1 Document Table Names from Oracle Data Dictionary 44**](#_Toc482452789)

[**4.2 USER\_CONSTRAINTS and USER\_CONS\_COLUMNS 45**](#_Toc482452790)

[**4.2.1 Code and Execute the show\_constraints.sql script 46**](#_Toc482452791)

[**4.2.2 Document User Constraints 47**](#_Toc482452792)

[**4.3 USER\_INDEXES (IND) and USER\_IND\_COLUMNS 49**](#_Toc482452793)

[**4.3.1 Code and Execute the show\_indexes.sql script 50**](#_Toc482452794)

[**4.3.2 Document Indexes 51**](#_Toc482452795)

[**5.0 ALTER TABLE - Modifying a Table Definition 51**](#_Toc482452796)

[**5.1 ALTER TABLE EXAMPLES 52**](#_Toc482452797)

[**6.0 TRUNCATE TABLE 53**](#_Toc482452798)

[**7.0 Making a COPY of a Table 54**](#_Toc482452799)

**Enter your Name Here 🡺 Puja Ghosh**

# Introduction

**Assignment 3 is organized into two documents: Assignment 3 – Theory and Assignment 3- Hands-on.**

**Assignment 3 – Theory will be allocated 30% of the assignment points**

**Assignment 3 – Hands-on will be allocated 70% of the Assignment points.**

## How to complete Hand-on Tutorial Requirements and Review Questions

This tutorial will provide directions and demonstration examples to guide the student to perform a hands-on requirement. Using these tutorial demonstration examples the student will be required to perform a similar hands-on task. The following is an example of typical tutorial hands-on requirement.

**Use a graphical snipping tool to document your successful logon with a display of your ISPF Primary Option Menu below.**

The directions specify the use of a graphical snipping tool, such as the Window's Snipping Tool. (Directions to use the Window's Snipping Tool will be presented next.) Any graphic snipping tool may be used to demonstrate that you have successfully completed the required hands-on task.

A grey or colored box will be provided after the requirement directions. You are required to provide a graphic image, e.g., using cut-and-paste, to document that the hands-on requirement was correctly completed.

Review questions are also be provided at the end of the tutorial. The following is an example of a review question format. Since type the answer in provided grey or colored box.

1. What is the purpose of a partitioned data set? Answer:

Type in the answer to the question into the grey or colored box.

**It is recommended that you use Table of Contents at the beginning of the tutorial to review and navigate to the concept presented in the review question. Students will find that using the document FIND tool or searching GOOGLE may also be valuable for researching the review question answer.**

## Oracle Error Codes Summary

**Important!**

The following error codes are discussed at least once in this assignment.

**ORA-00904 - Error Invalid Column Name**

**ORA-00907 - Error Missing Right Parenthesis**

**ORA-00922 - Error Missing or Invalid Option**

**ORA-00942 - Error Table or View Does Not Exist**

**ORA-00955 - Name Is Already Used by Existing Object**

**ORA-00957 - Error Duplicate Column Name**

**ORA-02267 - Error Column Type Incompatible (Foreign Key Relationship)**

**ORA-02449 - Error Unique/Primary Keys in Table Referenced by Foreign Keys**

## Windows Snipping Tool

(If necessary, consult your instructor for other graphical copying tools)

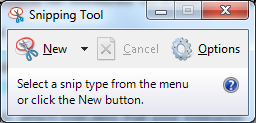
The following YouTube Videos will provide more directions for Windows 10, 8 and 7 Snipping Tool

* Windows 10 Tutorial | How to Use Snipping Tool - <https://www.youtube.com/watch?v=RbPAxcTf8h8>
* How To Use Snipping Tool & Print Screen | Windows 7 Tutorial - <https://www.youtube.com/watch?v=DQqwLkrqvmQ>

**Copying a z/OS Image into your Hands-on Requirement (Using Windows 7)**

**Step 1**. Click on the Window's **Start Button** located at the lower left-hand corner of your desktop. In the "**Search programs and file**" text box type the **key word Snipping**. The name of the Snipping Tool should appear. **Select the Snipping Tool entry.**

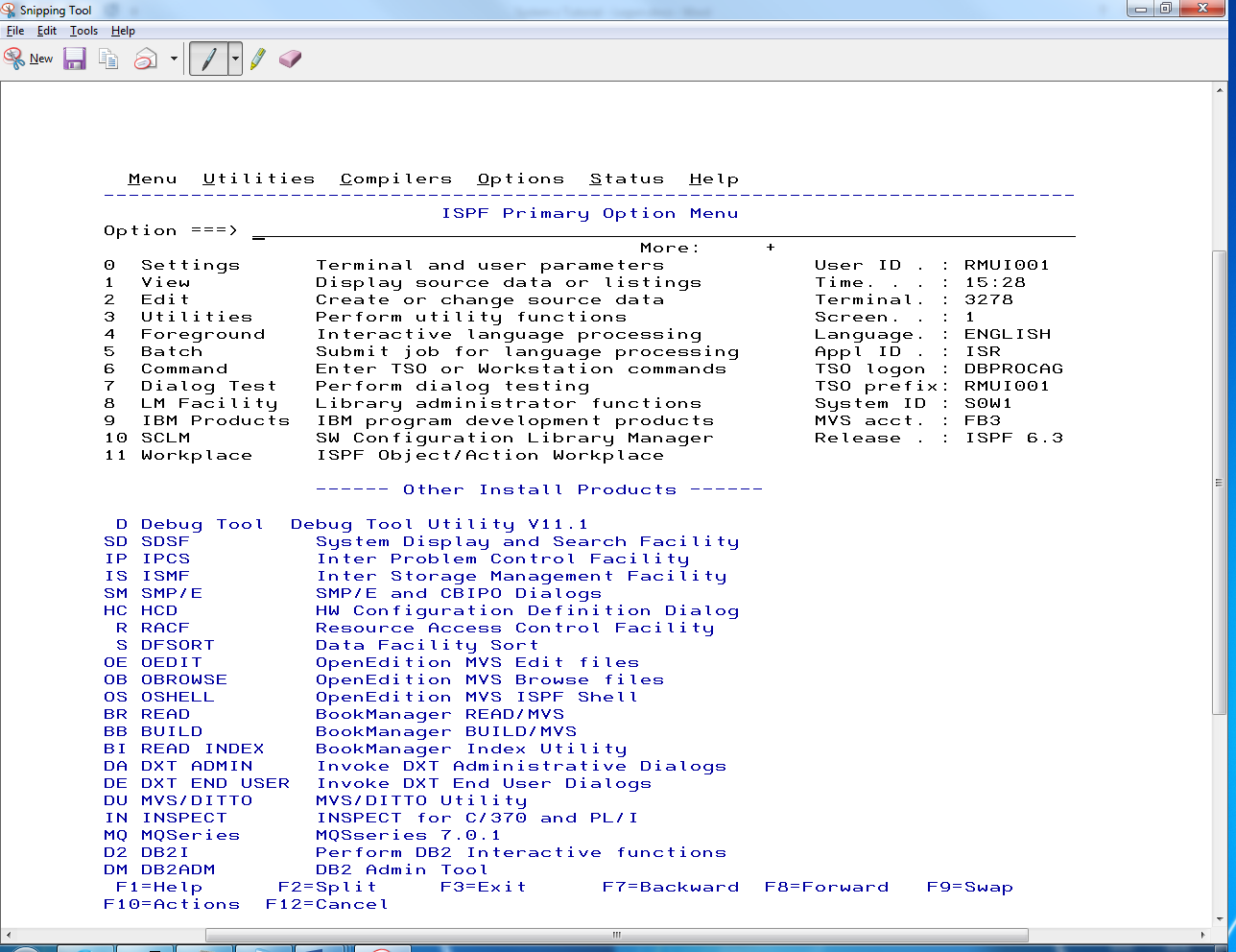
The following Snipping Tool Dialog Box will be displayed. It is recommended to drag this dialog box outside the area that you want to document before you continue, else you may copy the dialog box.



**Step 2.** Once you have z/OS information displayed in your 3270 emulator that you wish to document, click on the "**New**" icon to start the image selecting process. Your desktop will be "greyed" or dimmed, which indicates that the Snipping Tool utility is ready to select an image.

**Step 3.** Drag a copy box select tool using your right-mouse button around the image to be selected. Once you release the mouse button, a Snipping Tool windows box will display your selected image and a tool bar similar to the following.

***Snipping Tool Utility Window Example.***



**Step 4.** For this tutorial you should select the Snipping Tool ***Copy Button***/Icon which is displayed with two sheets of paper. Select the Copy Button. See previous red arrow.

**Step 5.** Position the cursor inside the grey or color Hands-on requirement box and left-click to select the position.

**Step 6.** Paste the image using Ctrl-V or the Edit/Paste drop down menu

# 1.0 Creating a Simple Faculty, Course and Student Database

## 1.1 Introduction to DDL SQL

In a previous assignment you were provided a DDL (Data Definition Language) Script which contained CREATE TABLE statements that you executed to create a simple project management database with sample data. You were not required to code any SQL DDL statements or to insert sample data to practice using simple SELECT statements.

In this requirement, you will be required to code CREATE TABLE, CREATE INDEX, ALTER TABLE, and INSERT scripts from sample SQL DDL statements to create a Faculty, Course and Student database which will be used in the Oracle DDL and Oracle DML handouts. While this assignment will use a copy-and-code approach to implement a sample database, your next assignment will provide a conceptual design of data base with no sample SQL statements. This means that you should get master your DDL statements and your SQL syntax errors in this assignment.

**Video - Oracle Database Concepts - Part 1** [**http://www.youtube.com/watch?v=dvpeTXM9jjs&feature=relmfu**](http://www.youtube.com/watch?v=dvpeTXM9jjs&feature=relmfu)

**Video - Oracle Database Concepts - Part 2 -** [**http://www.youtube.com/watch?v=0iF6BVp7mEw&feature=relmfu**](http://www.youtube.com/watch?v=0iF6BVp7mEw&feature=relmfu)

**Video - Creating an Oracle Table (Theory) -** [**http://www.youtube.com/watch?v=BjT4ptWHUjk&feature=fvwrel**](http://www.youtube.com/watch?v=BjT4ptWHUjk&feature=fvwrel)

**Video - Creating an Oracle Table- Hands-On (CREATE TABLE command) -**

[**http://www.youtube.com/watch?v=OKYi6nZ3AjA**](http://www.youtube.com/watch?v=OKYi6nZ3AjA)

**Video - Creating an Oracle Table using SQL Developer -** [**http://www.youtube.com/watch?v=dLr6PWbNagA&feature=fvwrel**](http://www.youtube.com/watch?v=dLr6PWbNagA&feature=fvwrel)

**Video - Difference between Char and Varchar2**

[**http://www.youtube.com/watch?v=0-0BmLO-78c&feature=relmfu**](http://www.youtube.com/watch?v=0-0BmLO-78c&feature=relmfu)

This requirement will present a tutorial on the CREATE TABLE command. Follow the steps. Later you will be required to document the tables that you created using the Oracle Administration Tables or Catalogs.

Videos are very good. It uses Oracle's SQL Developer. You may use and install Oracle SQL Developer instead of the using the Notepad, Putty, and WinSCP approach. But, most professionals recommend knowledge of using both development tools. Each development system has its unique advantages. It is the opinion of most professionals that 1) Basic knowledge of Putty, WinSCP and SQL\*Plus are requirements for all IS/IT professional, and 2) once a person is knowledgeable and comfortable using SQL Oracle's SQL Developer provides a IDE and tools that makes database management easier. The bottom line is that if you do not know SQL and databases, the tool doesn't matter.

Video - How to install Oracle SQL Developer in Windows 7 - Database Tutorial 49 - Oracle DBA tutorial

<https://www.youtube.com/watch?v=uI339bg4qKE>

Video - How to install SQL Developer in Window 8 - <https://www.youtube.com/watch?v=zQ3oEYojssM>

## 1.2 Conceptual Design of a Student Registration Database

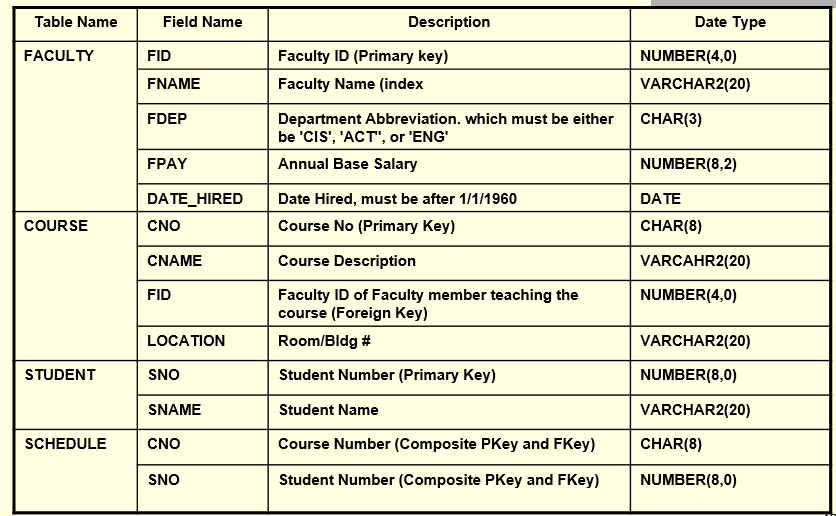
**Conceptual design requirements**

Conceptual design describes the major components of the of the database system:

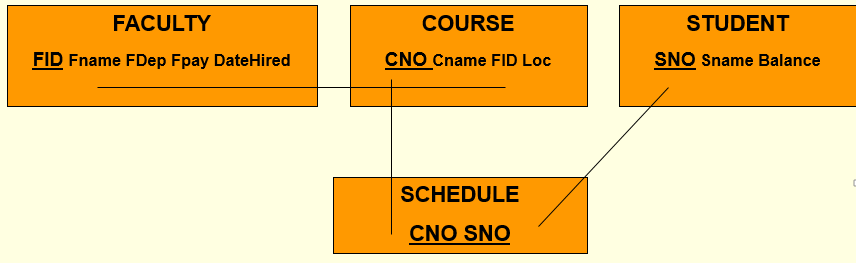
* Describe the major objects, tables, entities.
* Assign attributes, data fields, data elements to the major entities, and domain of values (data type and validation rules)
* Designate a Primary Key for each table
* Describe relationships between the major objects, tables, files, e.g., 1:1, 1:N, and N:M
* Describe any constraints on the relationships, e.g., foreign keys. For example, an object must exist before another.

Before we physically create various tables and columns we need to conceptually design the database.

* The conceptual design requirements are to create a data base that stores faculty, course, student and course registration information in four separate tables:
* **FACULTY** - Each row will store one-time information for one faculty member, uniquely identified by an arbitrary Faculty ID.
* **COURSE** - Each row will store information for one course section, taught by one faculty member, and uniquely identified by a Course Number. Course information is separated from the faculty information since one faculty member may teach more than one course.
* **STUDENT** - Each row will store information for one student uniquely identified by a Student Number.
* **SCHEDULE** - Each row will store information for one course selection for one student. Since a COURSE may have more than one student, and STUDENT may take more than one COURSE, this table will contain the primary keys (common fields) of COURSE and STUDENT tables.



**1.3 Bachman Diagram of the Student Registration Database**



## 1.4 Oracle Table Names

Every Oracle table must be named. The Table Name that you use should be (but not required) descriptive of the type of data you plan to enter into the table. In our example will store one-time information for a give faculty member in a table named faculty. Do not use the word TABLE or TBL in the table name. This is redundant. Use a singular term for the table name. Do not use unknown abbreviations. Other rules for naming tables are:

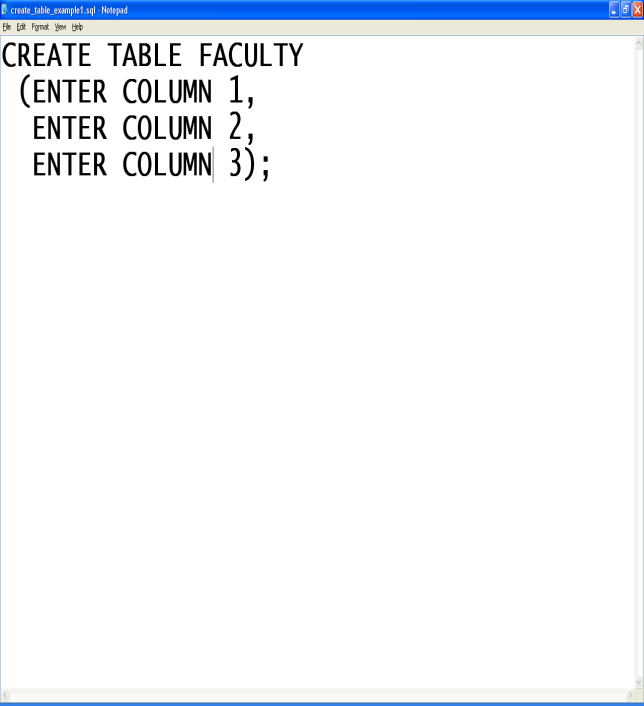
* Can be up to 30 characters long (length is rarely a problem). Oracle is case insensitive, unless you create the table with a name surrounded by double quotes. If you do use lower case letters, Oracle will always covert it to upper case when using various reports.
* Each table owned by an Oracle account must be unique
* Can include any combination of letters, digits (0-9), and the special characters $, # and \_ (underscore). The table name cannot include spaces. Use the underscore character to improve readability instead of a space.
* A table name must begin with an alphabetic character
* Must be unique within a table, but can be duplicated between tables.
* Must not be a SQL reserved word. If suspect that your error may be due to your use of a reserved word, it is recommended that you enter "ORACLE WORD\_IN\_QUESTION" into google.com. If you get a hit, most likely it is a reserved word.

## 1.5 SQL Column Names

A Column Name is a name of a column where you will store data, or values. The Column Name that you use should be (but not required) descriptive of the type of data you plan to enter into the column. In the first column, our example will store a Faculty ID number that will be named fid, an abbreviation that will make this field name easier to see during this presentation. Other rules for naming columns are:

* Can be up to 30 characters long (length is rarely a problem). Oracle is not case insensitive for column names.
* An Oracle table can have up to 254 columns
* Can include any combination of letters, digits (0-9), and the special characters $, # and \_ (underscore). The column name cannot include spaces. Use the underscore character to improve readability instead of a space.
* A column name must begin with an alphabetic character
* Must be unique within a table, but can be duplicated between tables.
* Must not be a SQL reserved word. If suspect that your error may be due to your use of a reserved word, it is recommended that you enter "ORACLE WORD\_IN\_QUESTION" into google.com. If you get a hit, most likely it is a reserved word.

## 1.6 Overview of the CREATE TABLE Command

Because of its many options and clauses, The SQL command CREATE TABLE can be quite complex.

The code for the CREATE TABLE SQL command may be written on one or more lines. The CREATE TABLE command is always terminated with a semicolon.

A set of parentheses are used to surround the column and constraint declarations.

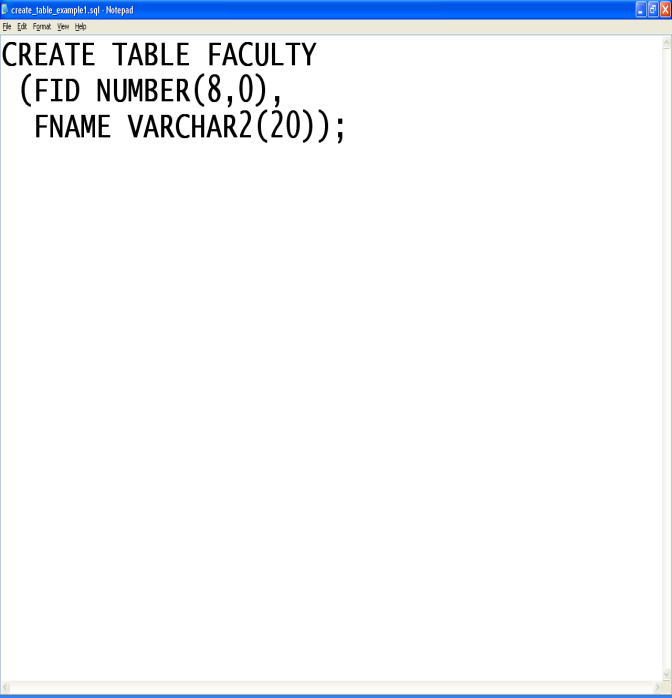
**The comma ends any column declaration, except the last one.**

You can code a column declaration on multiple lines, but each column and constraint declaration is ended by a comma.

**The semicolon ends any SQL statement.**

The comma means: *Get ready for another column declaration. I am ready for the next one****.***

***NOTICE Column 3 does NOT end with a comma since there is no column or constraint declarations coming.***

**Keep it Simple**

When an individual attempts to code their first CREATE TABLE statement, they start by coding column names, data types, options and clauses.

DO NOT code every column definition when you first attempt to code your fist table. Keep it simple!

In the following examples, the faculty table will first be created with two columns, using no options, primary keys or other constraints. We will then add additional column names and details to the examples.

You should follow the sequence this tutorial as written. A few extra steps will help you learn to develop complicated CREATE TABLE statements. Of course, many students rush and they pay the price with delays and confusion. This course's final grade will be determined by your understanding of the concepts and ability to implement SQL examples. There is no grade allocated to typing or effort for this course. If you expect a grade for effort, you should consider a different major.

**A Review of Using Notepad, Putty, Linux and WinSCP**

Some individuals prefer a full-screen editor like Notepad to make SQL command scripts. You will use this technique in a future assignment. In the previous assignment we copied the create\_project.sql script from our local computer to Linux using WinSCP. Then the create\_project.sql was executed at the SQL> command prompt. Coping complex Oracle application scripts to initialize databases for applications is a very frequent task. We will customize this process to assist us to learn more complex SQL statements. This process is a simple four-step process. While at first this process may seem cumbersome, most find the process faster and easier to use than using the SQL\*Plus editor.

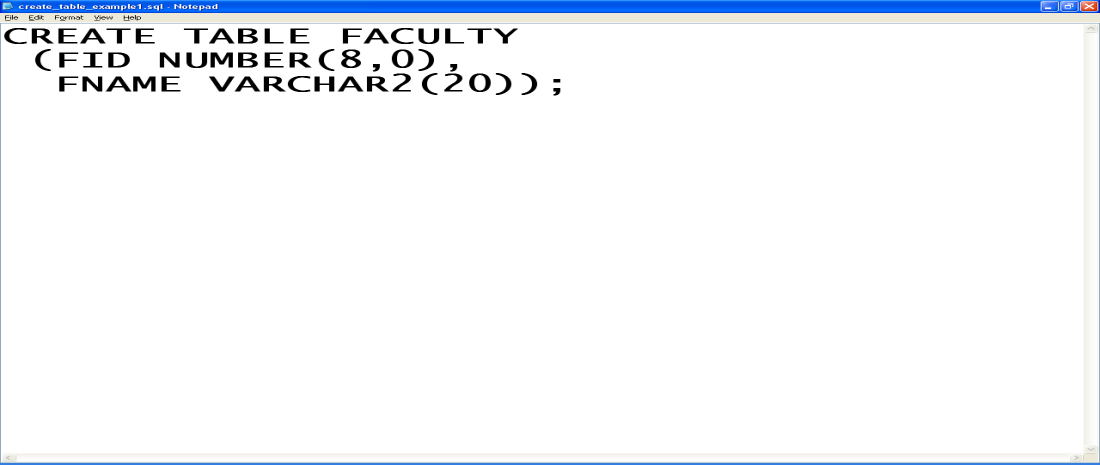
**1) Use Windows Notepad, or other local editor to write and edit SQL command script,**

**2) Save the SQL command script to a local storage device, e.g., Drive C, your jump drive, etc., using a file name with lower case letters, no embedded spaces and ends with the file extension .sql**

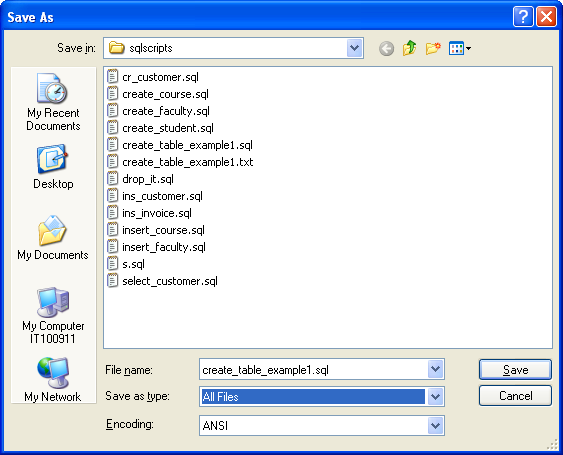
**3) Use WINSCP to copy the SQL command script from your local storage device to home directory on academics.rmu.edu**

**4) Use PUTTY to execute the SQL command script at the SQL> prompt.**

### Step 1 Use Windows Notepad, or other local editor to write and edit SQL command script.

******

### Step 2. Save AS or Save the SQL Command Script to a local storage device



**Use a file name with lower case letters**

**(Unix/Linux file names are case sensitive)**

**No embedded spaces  
 (use the underscore)**

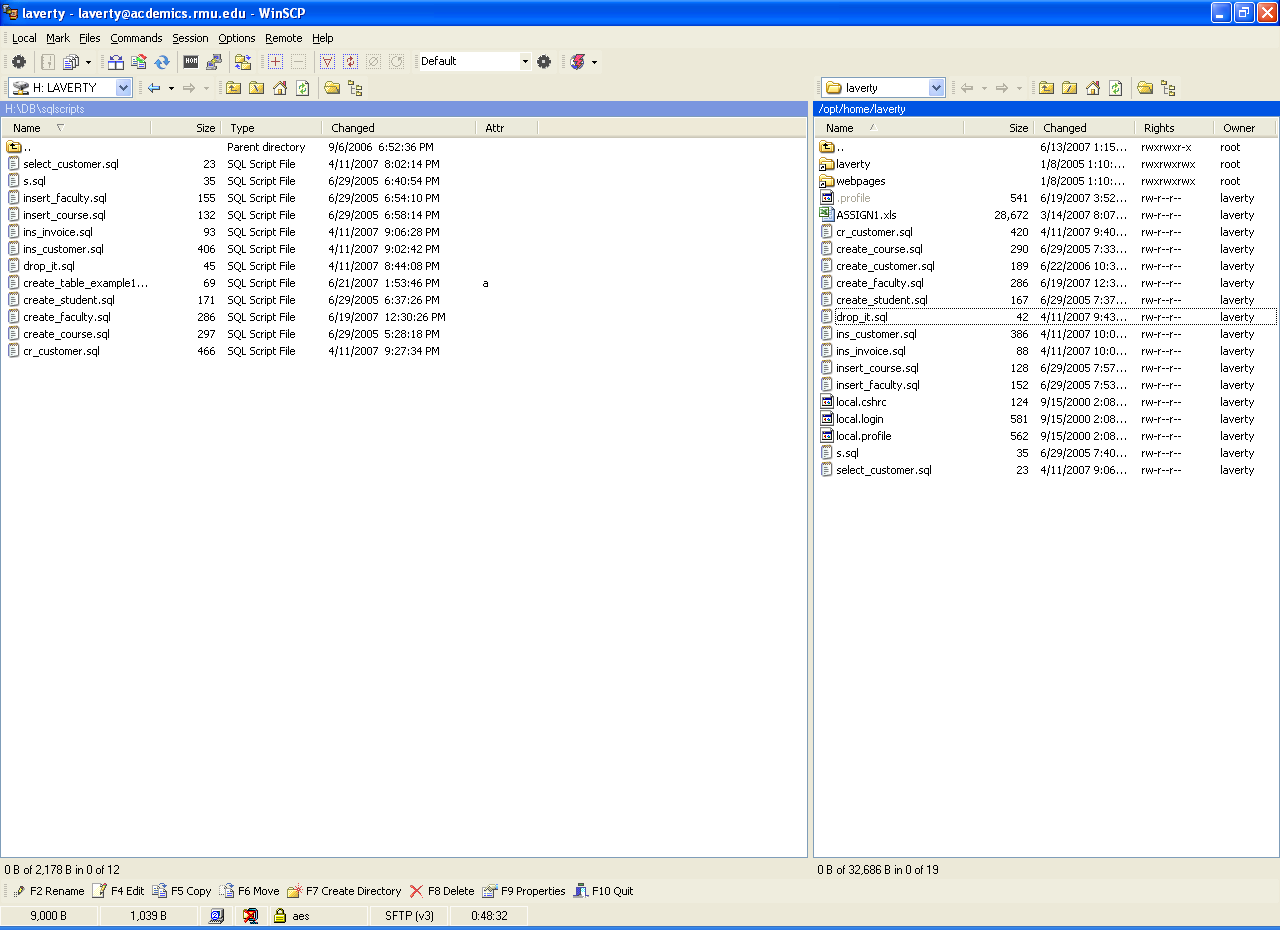
**End with the file extension .sql**

**(This is more descriptive and is not required)**

**Important!! Change the Save as Type to All Files**

**Notepad will automatically append .txt to your file name if you do not make this change or enclose your file name with double quotes. Your file name will be create\_table\_example\_1.sql.txt**

### Step 3. Use WINSCP to copy the SQL Command Script to the mediaweb.rmu.edu Linux Server

****

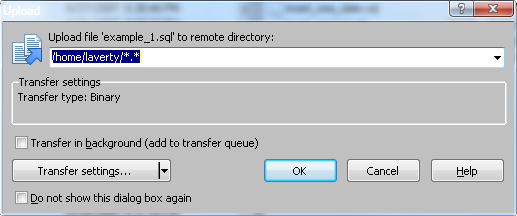
**WINSCP is a secured shell file copy utility. As with most FTP and SCP programs, there are two Window panes. In this example, the local drive folders and files is displayed on the left side and your home directory stored on the remote server, i.e., mediaweb.rmu.edu, is display on the right side.**

**Select the files you wish to copy on your local storage drive.**

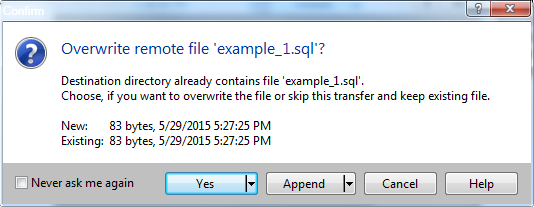
**Press F5 (copy) or right-click and select Copy**

**The default settings of this dialog boxes are appropriate for this application.**

**Click on OK to start the Process. You will be prompted if the file previously exists. You will be given an option to replace (Overwrite) the destination file.**



**Replacing an Existing File**



### Step 4 - Use PUTTY to Execute the SQL Command Script at the SQL\*Plus command prompt.

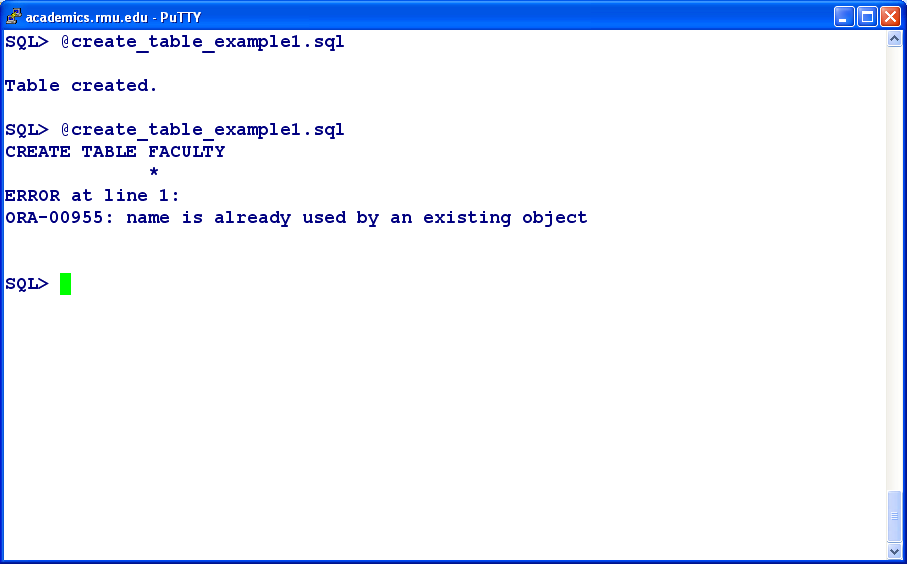


**To execute a SQL script at the SQL> prompt use must precede the file name with an @ symbol.**

**Notice that the ending semicolon is not used at the SQL>. The ending semicolon was written in the script.**

## 1.7 Create Table Oracle Error Messages (ORA)

### ORA-00955: name is already used by an existing object

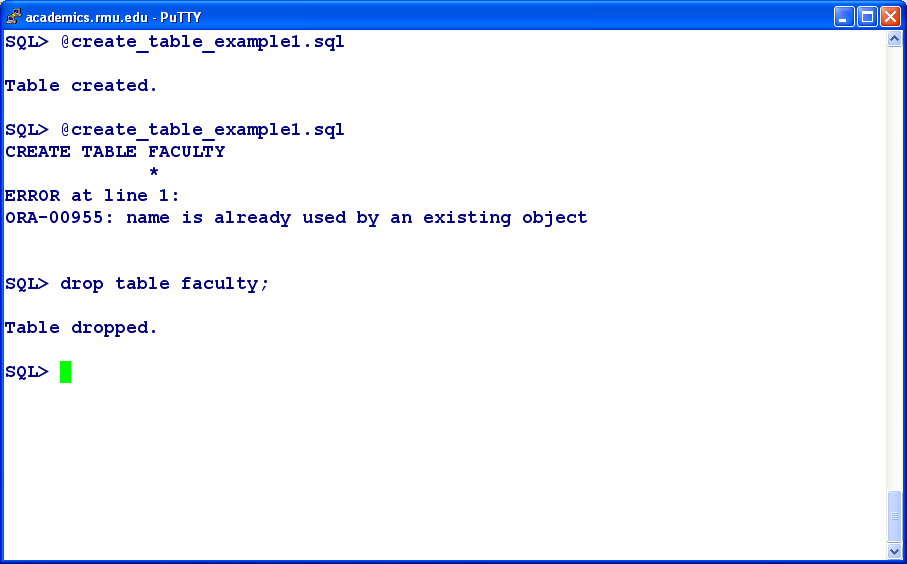
****

**Frequently we modify and improve a CREATE TABLE script, and wish to re-create the table a second time. As previously specified, every Oracle table name must be unique for a user.**

**The error message is clear. ORA-00955: name is already used by an existing object**

**In this example, the attempt to re-execute the improved script but the FACULTY TABLE already exists. (Tables, indexes and constraints are examples of Oracle Objects)**

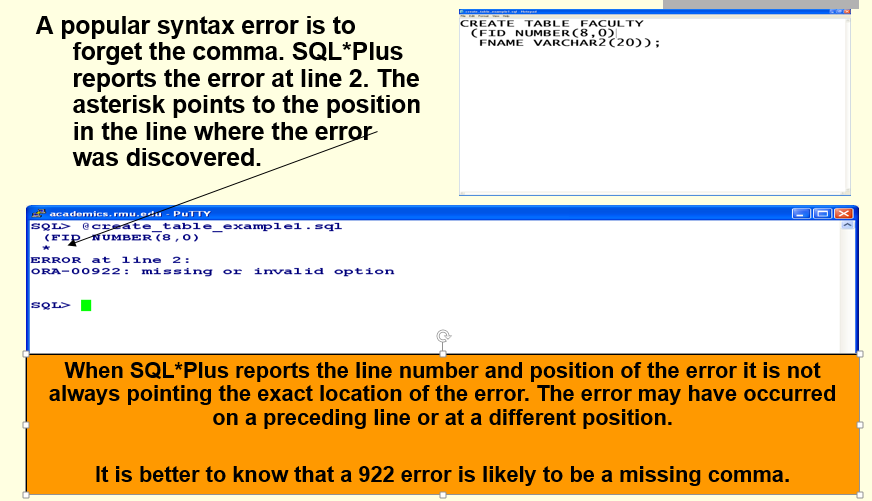
**DROP TABLE Command**



**The DROP TABLE command removes all previously stored data and the data structure of the FACULY Table.**

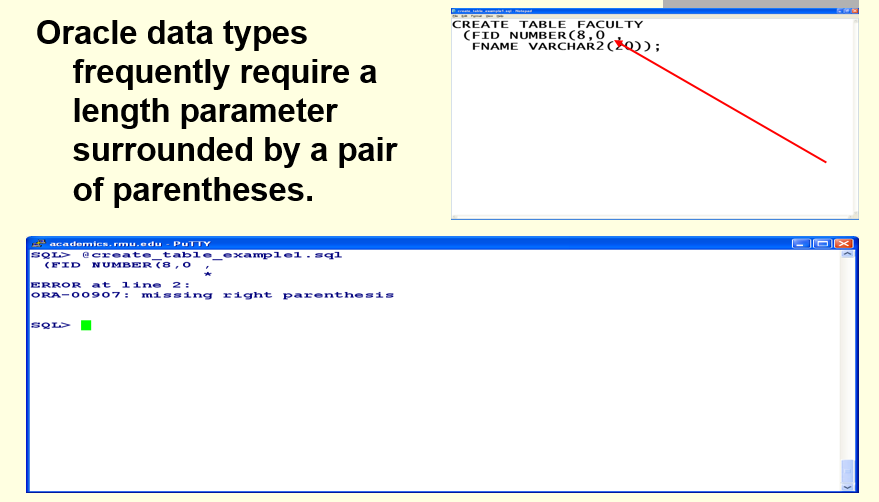
**Since we are executing the DROP TABLE directly at the SQL> command prompt and not using a script in this example, we DO NOT use the @ symbol, but we must use the ending semicolon to execute the DROP TABLE command.**

### ORA-00922 Error Missing or Invalid Option

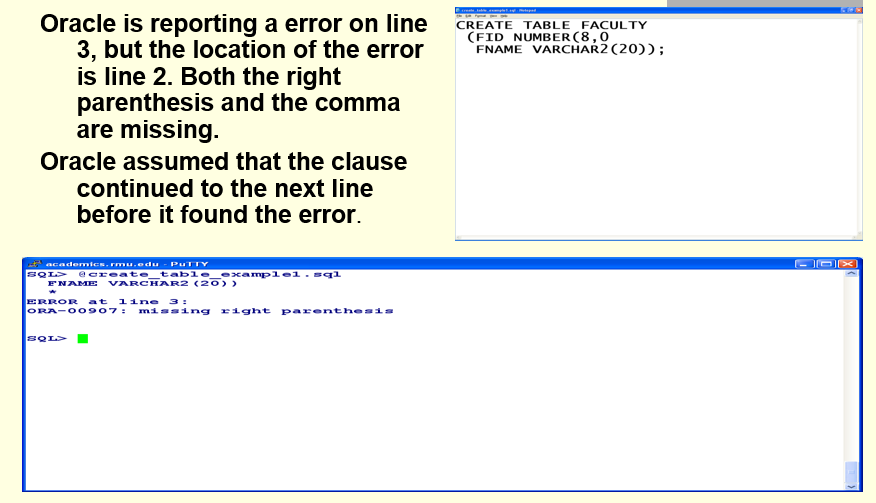


### ORA-00907 Error Missing Right Parenthesis

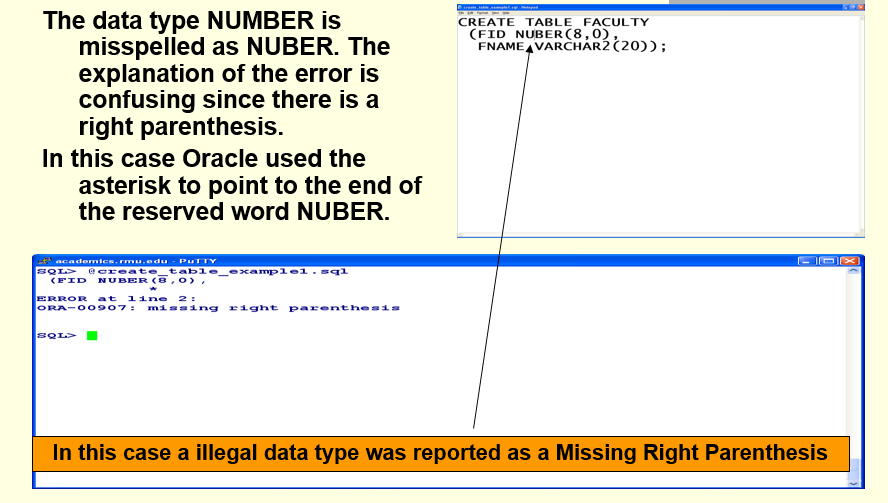
**Example 1**



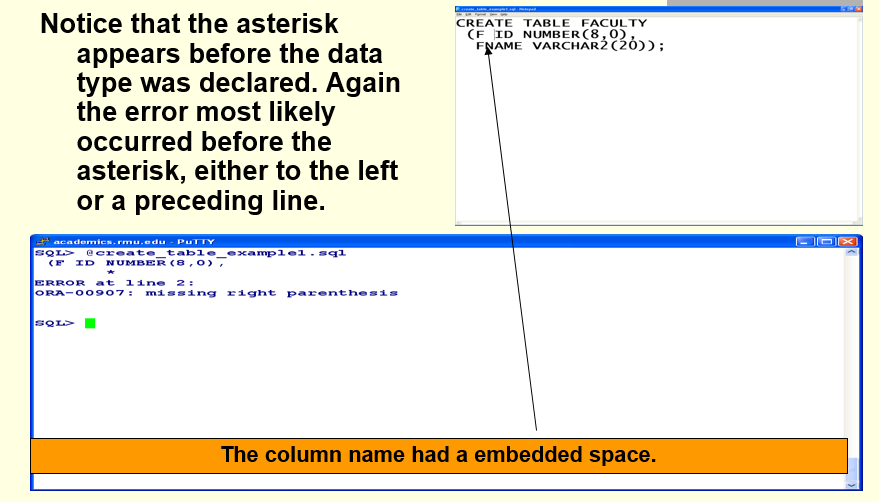
**Example 2**



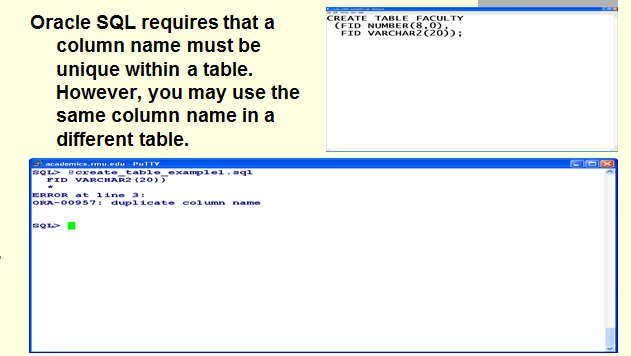
**Example 3**



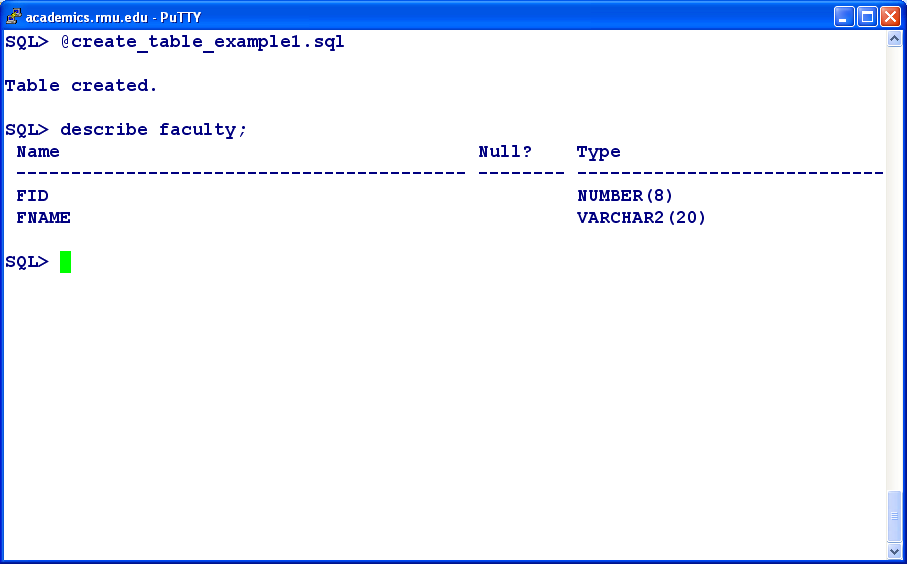
**Example 4**



### ORA-00957 Error Duplicate Column Name



## 1.8 Document Faculty Table - SQL DESCRIBE command



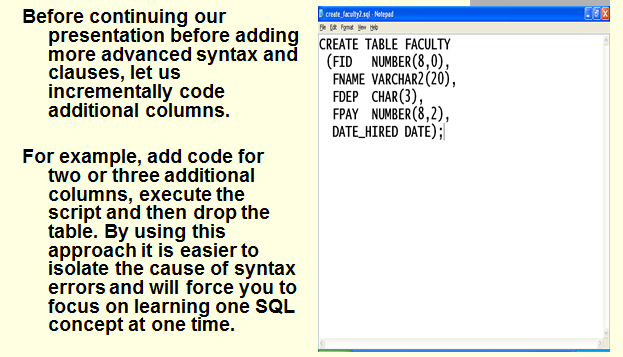
**To see a short summary of a table's column names and data types of a table use the describe table\_name command. The describe command may only display one table at a time.**

**A more detail display of the properties of your table will be presented later.**

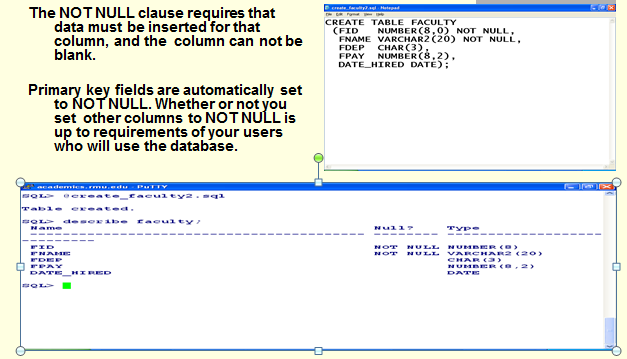
**If the describe command indicates that the table name was not found, either your misspelled the table name or the table has not been created yet.**

Using a Snipping Tool document the execution of your DESCRIBE FACULTY statement, Make sure that you include your Linux header at the top of the Putty screen to receive credit. You cm use the SQL\*Plus command clear screen to make your documentation neater. Paste you documentation below 🡺

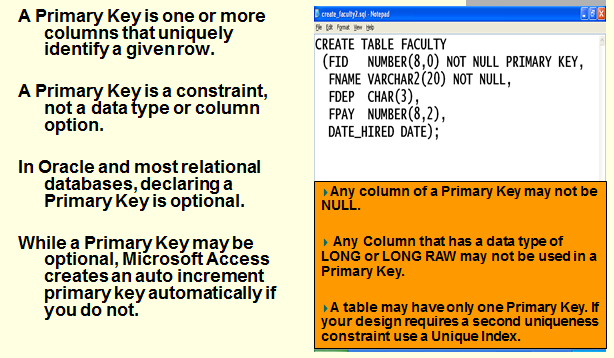
## 1.9 Adding More Clauses and Columns to the Faculty Table



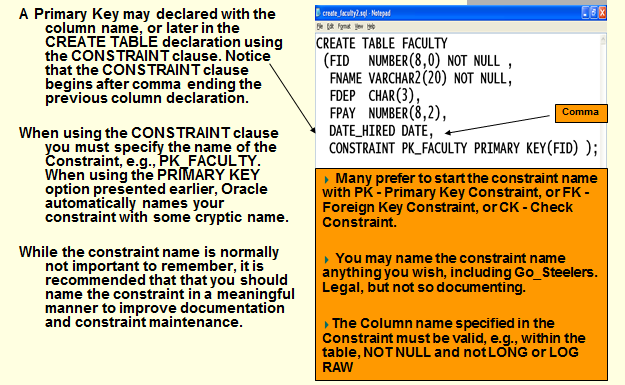
### 1 9.1 NOT NULL Clause



### 1.9.2 Primary Key Clause

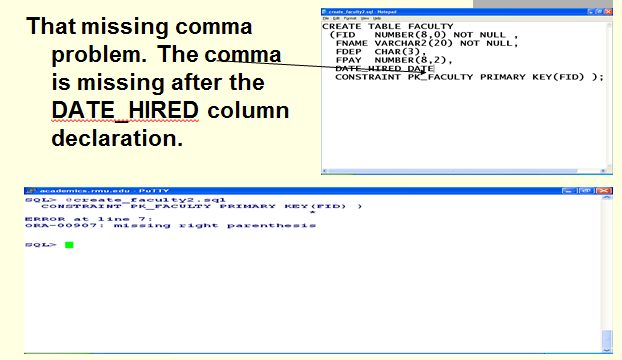


### 1.9.3 Primary Key Constraint

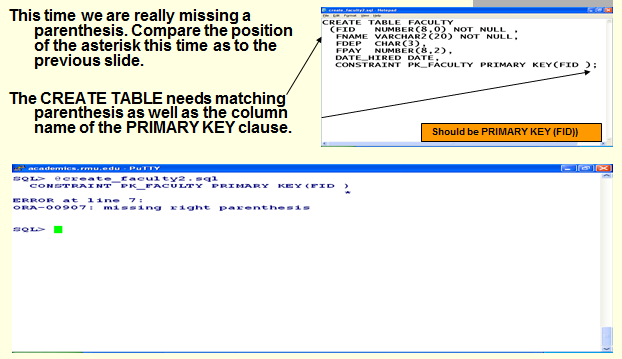


### ORA-00907 Error Missing Right Parenthesis

**Example 5**

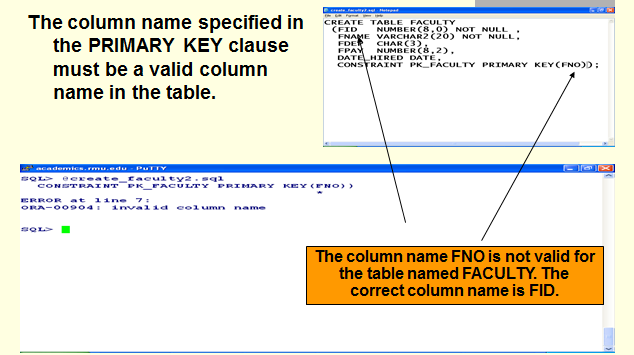


**Example 6**

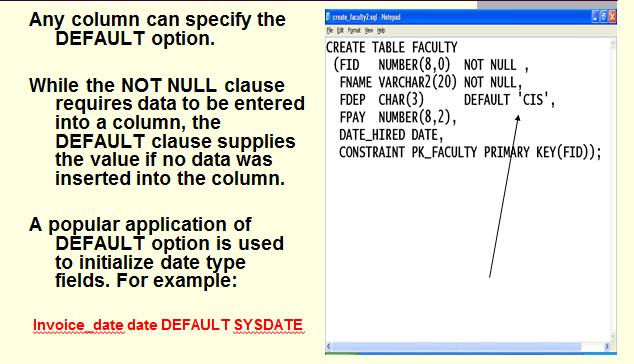


### ORA-00904 Error Invalid Column Name

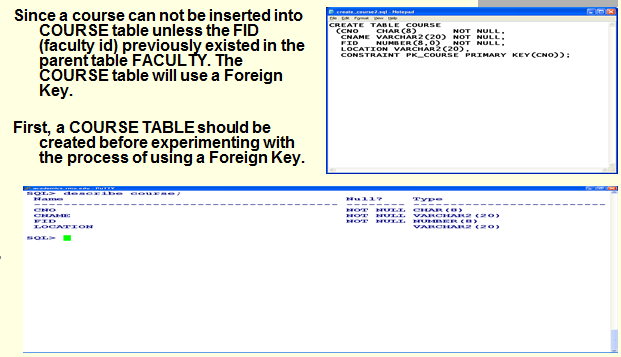
**Example 1**



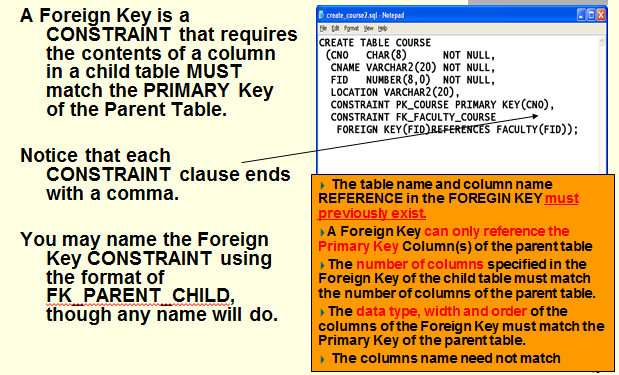
### 1.9.4 DEFAULT Clause/Option



## 1.10 CREATE TABLE COURSE



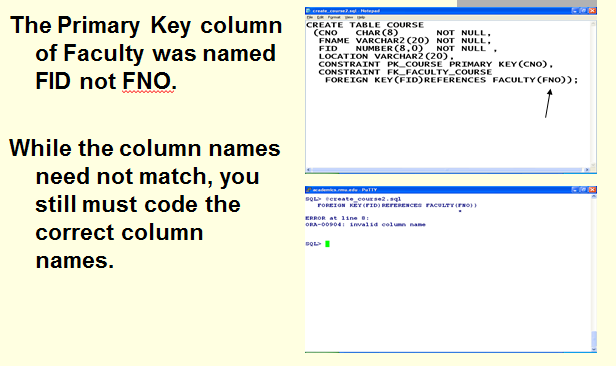
## 1.10.1 Coding a Foreign Key Clause



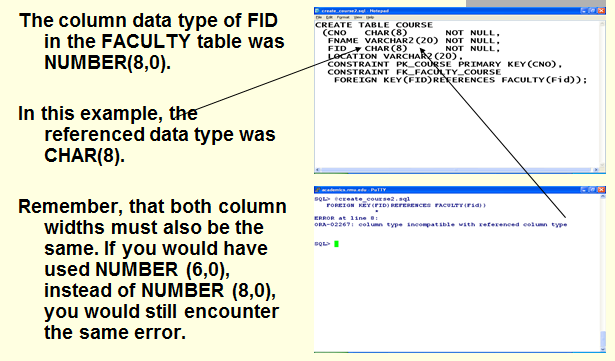
## 1.10.1 Foreign Key Oracle Error Messages (ORA)

### ORA-00904 Error Invalid Column

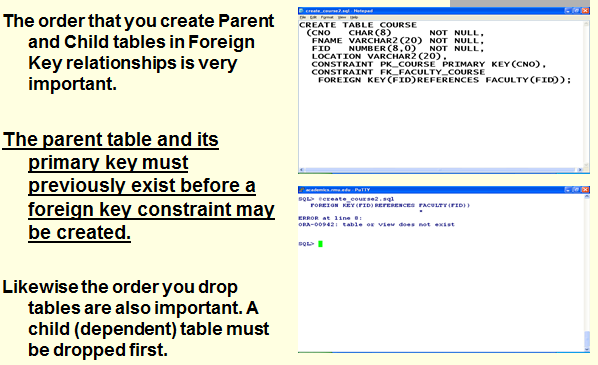
**Example 1**



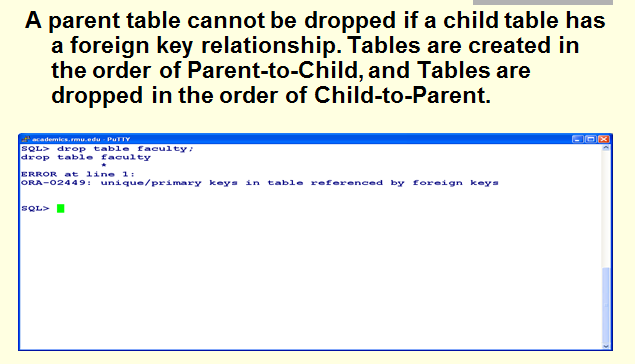
### ORA-02267 Error Column Type Incompatible



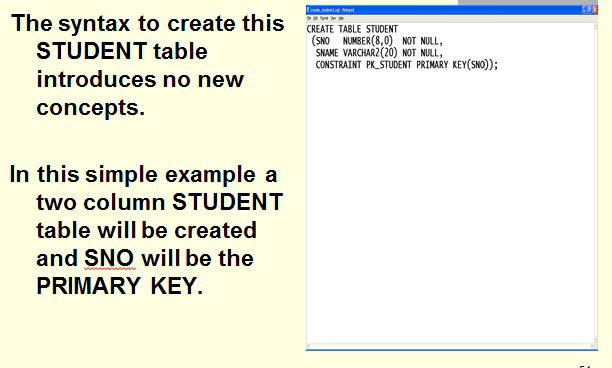
### ORA-00942 Error Table or View Does Not Exist



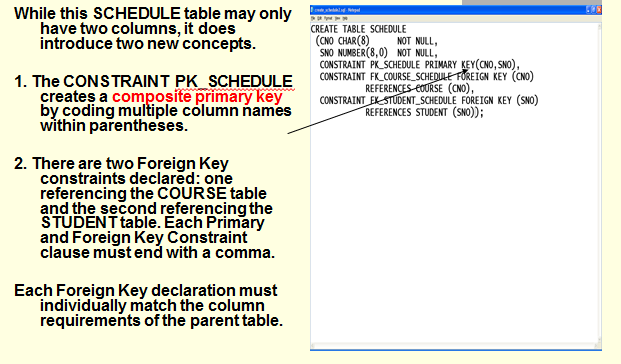
### ORA-02449 Error - Unique/Primary Keys in Table Referenced by Foreign Keys



## 1.11 CREATE TABLE STUDENT



## 1.12 CREATE TABLE SCHEDULE



## 1.13 Referential Integrity and Foreign Keys

**Video - Foreign Keys**

[**http://www.youtube.com/watch?v=KewZW5K4d7A&feature=fvwrel**](http://www.youtube.com/watch?v=KewZW5K4d7A&feature=fvwrel)

**Video - Creating Oracle Tables with Constraints -**

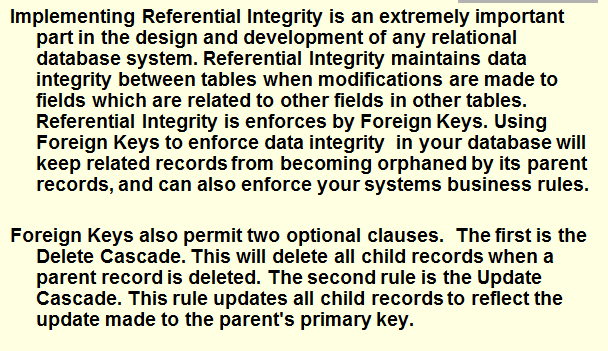
[**http://www.youtube.com/watch?v=SHIK49ROKXo&feature=fvwrel**](http://www.youtube.com/watch?v=SHIK49ROKXo&feature=fvwrel)

**Video - Check Constraints Part 1**

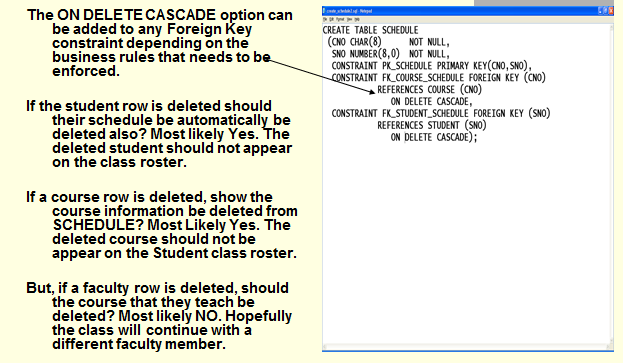
[**http://www.youtube.com/watch?v=PJ37ko6JBYA**](http://www.youtube.com/watch?v=PJ37ko6JBYA)

**Video- Check Constraints Part 2**

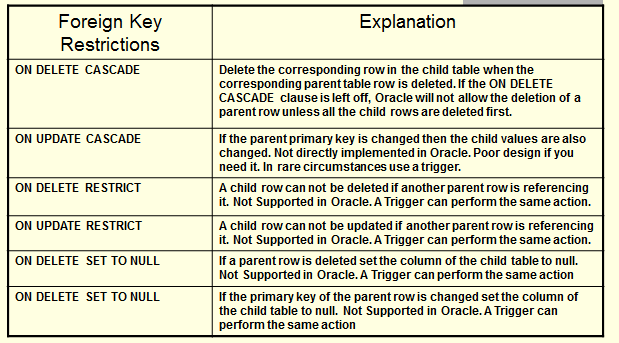
[**http://www.youtube.com/watch?v=9TYoGJW6Wk4&feature=relmfu**](http://www.youtube.com/watch?v=9TYoGJW6Wk4&feature=relmfu)



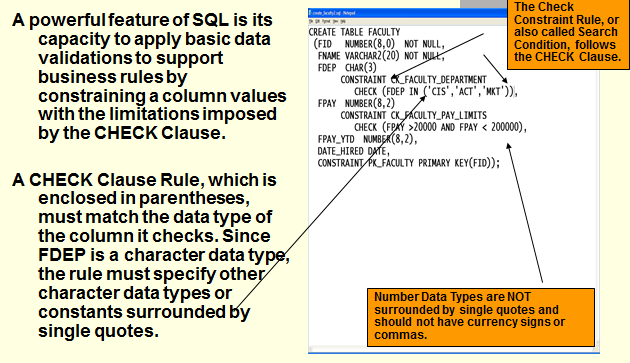
## 1.14 ON DELETE CASCADE



**Foreign Key Cascade Options**

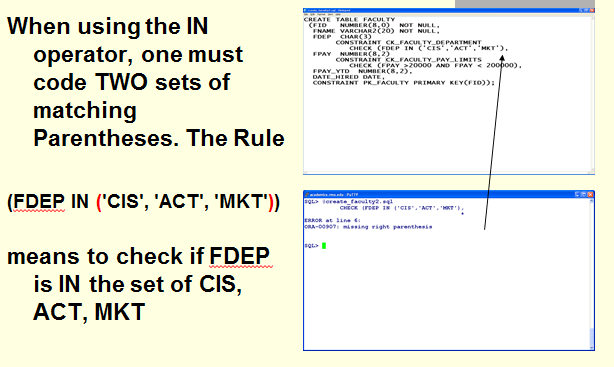


## 1.15 Check Constraints

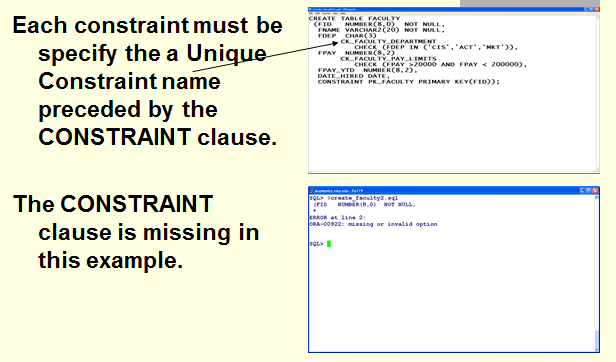


## 1.16 CHECK Constraint Errors

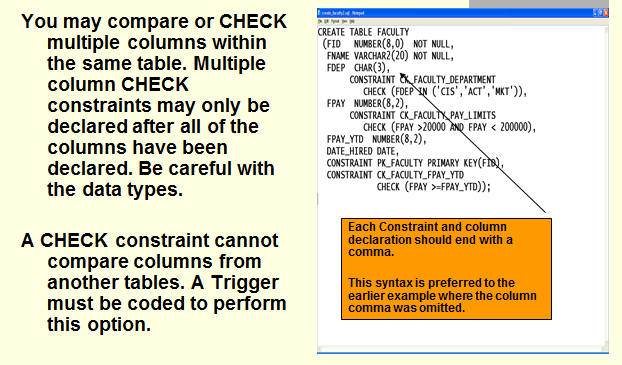
**Example 1**



**Example 2**

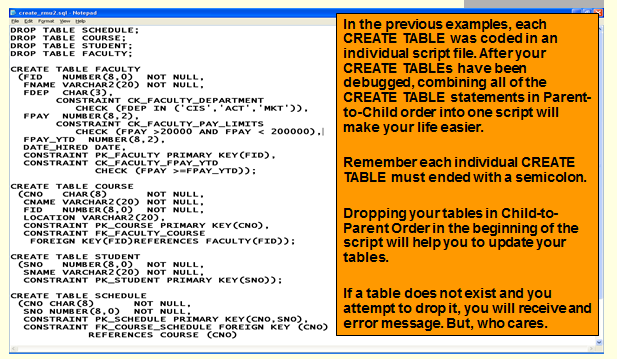


## 1.17 Multiple Column Check Constraints



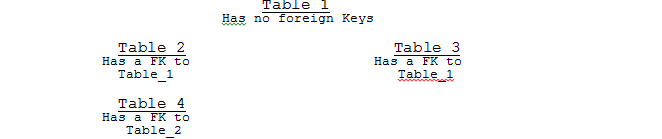
## 1.17 Code and Execute the create\_rmu2.sql script

Rarely, will one create multiple tables or objects one-at-a-time. This one-at-a-time approach is used to learn and test CREATE TABLE statements and other CREATE statements. Assuming you have successfully reached this point of the tutorial, you will copy your four previously tested CREATE TABLE SCRIPTS into one script. The installation process for any application that using a DBMS has a script similar to the create\_project.sql script you used in a previous assignment.



The DROP command is placed in the beginning script so that you drop any existing table. You may not CREATE a table if the table exists. If you do not want to execute the DROP command simply place two hyphens to comment the DROP command to prevent its execution, e.g.., - -DROP FACULTY

When creating tables that use Foreign Kys, the order that you create tables is important. Consider the following example:



When creating a Foreign Key, the referenced table and key must previously exist. This means that Table\_1 must exist before Table\_2 and Table\_3 is can be created. Both Table\_1 and Table\_2 must exist before Table\_4 can be created. Therefore, parent tables should be created before dependent or child tables.

Now, let’s consider the situation that all tables have been created, and the DBA discovers that Table\_1 must be totally recreated. Since Table\_1 already exists, then the ALTER TABLE command can be used to modify Table\_1.

But instead of altering Table\_1, let’s assume that you want to DROP TABLE Table\_1. You can’t. Table\_2 references Table\_1, and Table\_4 references Table\_2. You could temporarily alleviate the problem by dropping the Foreign Key Constraint of Table\_2 before Dropping Table\_1. A second solution one might consider is to DROP Table\_2 before DROPping Table\_1. But, you can’t DROP Table\_2 until the Foreign Key Constraint of Table\_4 has been removed or Table\_4 has been dropped.

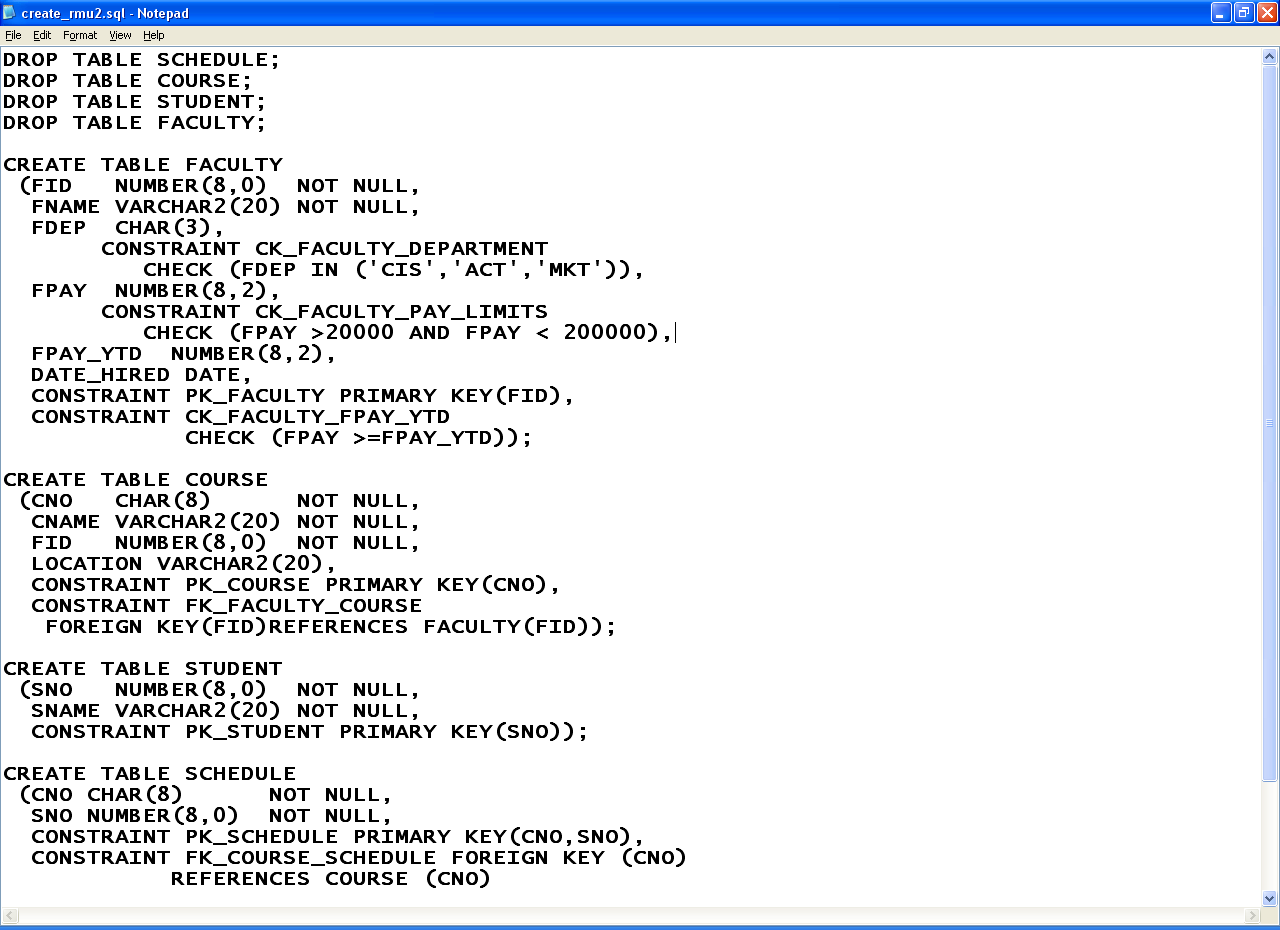
Therefore, let’s consider the following guidelines:

1. When creating a hierarchy of tables referenced by a foreign key, be careful that the parent table has been properly created before you create the dependent table (the one with the Foreign Key Constraint).

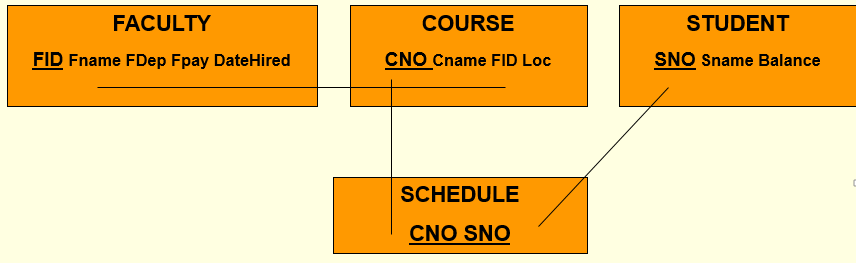
2. If the structure parent table must be changed, try to use the ALTER TABLE command, rather that DROPing and recreating the table.

3. If a parent table must be dropped then the foreign key constraint of the dependent table must be dropped or the dependent table itself must be dropped. When Creating Tables one starts at the top and then proceeds to the next lower level.

4. When dropping tables, one starts at the bottom and then proceeds to the next higher level.

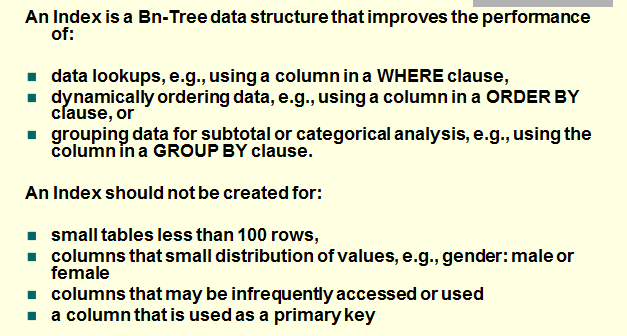


**Let's Review Our Conceptual Model**

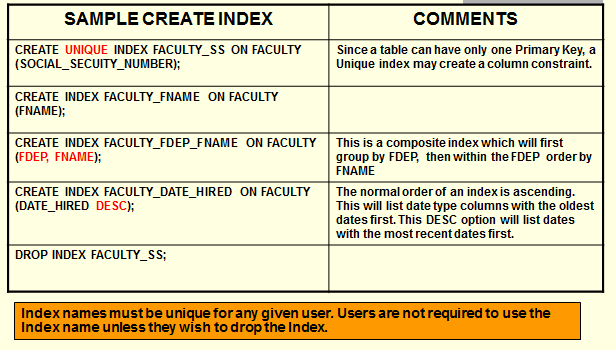


Do you notice that all three relationships have enforced by using a Foreign Key? Three Lines. Three Foreign keys.

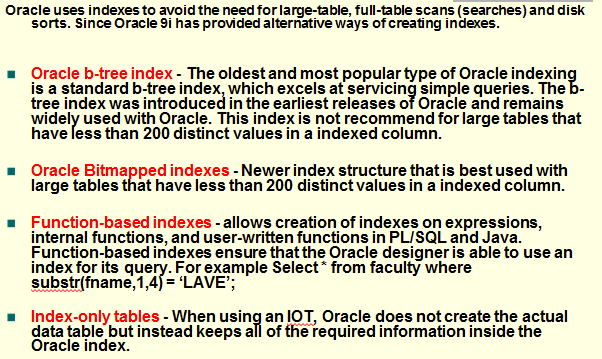
# 2.0 Creating Indexes



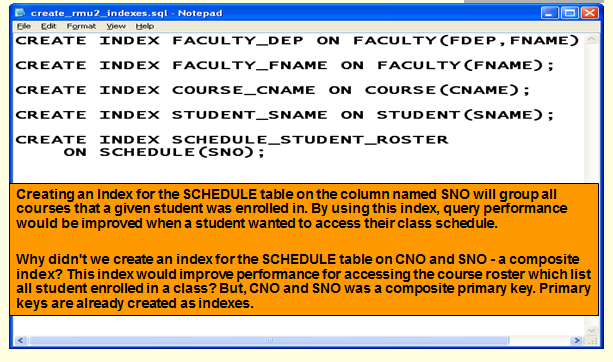
## 2.1 CREATE INDEX Examples



## 2.2 Types of Oracle Indexes



## 2.2 Code and Execute the create\_rmu2\_indexes.sql script



# 3. 0 Documenting Table Metadata (Information)

## Improving the Professional Appearance of the SQL Statement and Results using SQL\*Plus

The SQL\*Plus default LINESIZE is 80 characters. Your output line may exceed 80 characters and then wrap to the next line this making your documentation unreadable. Increase the LINESIZE at the SQL> prompt or by adding the following in your script, e.g., SET LINESIZE 130 [this is a SQL\*Plus command, not a SQL statement, and should not end with a semicolon

Setting the LINESIZE does not affect the font size in Putty. By default SQL\*Plus display 80 characters on a line not matter what the font size is in Putty. SQL\*Plus will force the display of the result of the SELECT statement is wrap to the next line at 80 characters. Increasing the SQL\*Plus LINESIZE may still be wrapped by Putty. You can decrease the font size in Putty.

## Improving the Professional Appearance of the SQL Statement and Results using PuTTY

**Change the point size (or font) under Window/Appearance**. Click CHANGE under Font Settings. A point size of 14 that is bolded is easier to read, but you may get word wrap problems. Clicking on OK only affects the current session.



**Video - Formatting SQL\*Plus Output -** [**https://www.youtube.com/watch?v=\_46MeUAFDO4**](https://www.youtube.com/watch?v=_46MeUAFDO4)

**Video - How-to: Change the PAGESIZE and LINESIZE in SQL\*Plus -** [**https://www.youtube.com/watch?v=hSQFcRt8kqI**](https://www.youtube.com/watch?v=hSQFcRt8kqI)

## 1. Document the execution of the DESCRIBE FACULTY command below.

**Enter the SQL\*Plus command Clear Screen.** Answer =>

***I must see your Putty Banner***

## 2. Document the execution of the DESCRIBE COURSE command below.

**Enter the SQL\*Plus command Clear Screen.** Answer =>

***I must see your Putty Banner***

## 3. Document the execution of DESCRIBE STUDENT command below.

**Enter the SQL\*PLUS command Clear Screen.** Answer =>

***I must see your Putty Banner***

## 4. Document the execution of the DESCRIBE SCHEDULE command below.

**Enter the SQL\*Plus command Clear Screen.** Answer =>

***I must see your Putty Banner***

# 4.0 Oracle Data Dictionary Concepts

The data dictionary is full of 'Metadata', information about what is going-on inside your database. Oracle’s data dictionary is a collection of system tables that contain information about the structure of database. The data dictionary is presented to us in the form of a number of views.

When you successfully execute a Create Table or Create Index statement the appropriate tables, columns, constraints and indexes are created. The information that describes these objects is stored in the data dictionary, e.g., the name of the tables, columns, data types, etc. In addition, information is also stored concerning the space utilization, performance, users, security and roles in the database.

A series of views are available that will provide access for three types of Oracle users: ALL (Public and User), USER, and DBA views. The DBA views include all objects, e.g., tables, indexes, etc., for all users and has access to more system performance information. The USER views will display all of the objects owned by the current user. The ALL views will display the USER- owned tables as well as access had been GRANTed to that User or the PUBLIC. You will NOT see information concerning other student's tables or Oracle objects.

## 4.1 USER\_TABLES (TABS)

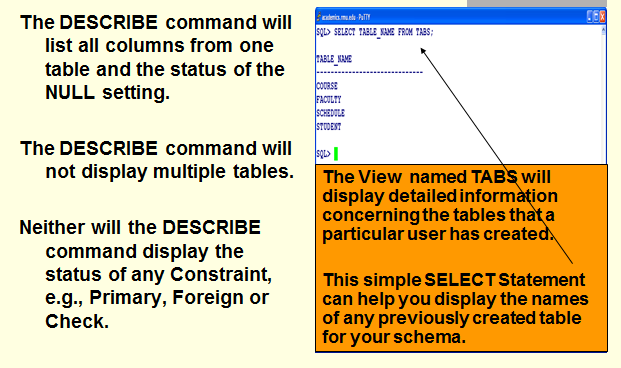
The USER\_TABLES or the TABS view displays information for all tables that the user has created. It is designed to provide one-time information. Since a table may have many columns, indexes or constraints this metadata is stored in different data dictionary system tables. Using the SELECT statement you can inquire about system administration of the tables. How many rows are in my tables? Which tables that up the most disk space?

|  |  |
| --- | --- |
| **USER\_TABLES (TABS)** | |
| **Column Name** | **Description** |
| **Table\_Name** | The Name of the table |
| **Num\_Rows** | Number of rows |
| **TableSpace\_Name** | An Oracle database is an operating system file. This is the name of the operating system file. A Table space can contain many tables, indexes or clusters. Tablespace has a fixed size and can only be increased by a DBA. Tablespace is added by attaching a second operating system file or by expanding the size of the original operating system file. The Create TableSpace command allows one or more files to be assigned immediately to the table space. |
| **Min\_Extents** | When a table is created an area is set aside for it. Extents are units of disk space measured in K that controls how a table will grow and use up space inside the Tablespace. Min\_Extents is the minimum size of the table. If min\_extents is 10K then a newly created table uses 10K of disk space even though nothing is stored inside it. |
| **Next\_Extent** | When rows are inserted to a point that the original min\_extents have been used, then Oracle can allocate more extents (disk space) to the table. Next\_extents controls how table will grow. |
| **Max\_Extents** | The maximum size of the table as measured in units of next\_extents. This number is not related to tablespace. Before a given table may grow to its max\_extents, Oracle may run out of tablespace. The total number of extents allocated to any given table is called a segment. Therefore, Tablespace contains tables, an each table have a physical unit of space assigned to it called a segment. A table segment was created by adding extents (units of disk space) to the table. |
| **Pct\_free**  **Pct\_used** | Percentage of the Segment used or free |
| **Blocks**  **Empty\_Blocks** | Physical measurement of disk space allocate to a segment. A block size varies from 512 characters to 2K depending upon the installation of Oracle into the operating system. |

For example, the following will display the number of rows stored in each table.

**Select Table\_name, Num\_rows, from tabs**

**SELECT TABLE\_NAME FROM TABS**



### 4.1.1 Document Table Names from Oracle Data Dictionary

**SELECT TABLE\_NAME FROM TABS;**

**Enter the SQL\*Plus command Clear Screen. Enter the SQL\*Plus command Clear Screen. Execute and Paste the Image for your SELECT TABLE\_NAME FROM TABS; below.** Answer =>

***I must see your Putty Banner***

## 4.2 USER\_CONSTRAINTS and USER\_CONS\_COLUMNS

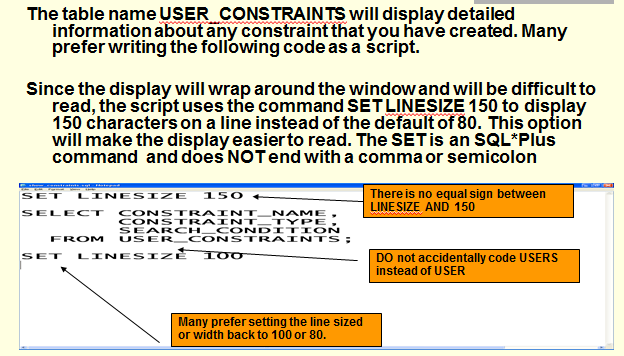
What are my Primary and Foreign Key Relationships? What are the Check Options stored with a table? There are two tables needed for this task. First, the view User\_Constraints is used to determine the type of constraint: Check Constraint (C) or NOT NULLS, Primary Key Constraint (P), Foreign Key Constraint (R), Unique Constraint (U), with the CHECK option (V) constraint. Once you know the type constraint you then can identify the name of table and the column involved.

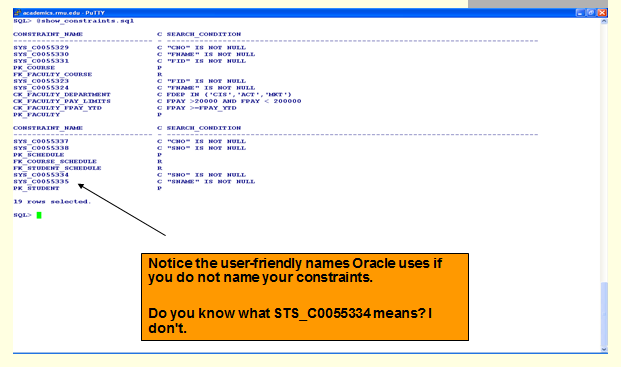
You will notice that these tables will specify the name of an OWNER. OWNER refers to the owner of the constraint, which may be different from the owner who created the table. Suppose the table COURSE has a primary key name COURSE\_PK based upon the column named CNO. In addition, the table has a foreign key constraint name FACULTY\_COURSE\_FK based upon column COURSE(FNO) which references FACULTY(FID)

|  |  |
| --- | --- |
| **USER\_CONSTRAINTS** | |
| **Column Name** | **Description** |
| **Owner** | The Owner of the constraint |
| **Constraint\_Name** | The name of the constraint. Using the above example there would be one row in this view named COURSE\_PK and another row name FACULTY\_COURSE\_FK |
| **Constraint\_Type** | Check Constraint (C) or NOT NULLS, Primary Key Constraint (P), Foreign Key Constraint (R), Unique Constraint (U), with the CHECK option (V) |
| **Table\_Name** | The name of the table associated with the constraint. In both cases this would be the table named COURSE |
| **Search\_Condition** | The search condition used with CHECK option, e.g., FPAY >10000 and FPAY< 90000. The previous example has not check option |
| **R\_Constraint\_Name** | The name of the constraint referenced by a FOREIGN KEY. Since a FOREIGN key must reference another table’s PRIMARY KEY, this column would only be used for FOREIGN KEY. In this case the PRIMARY KEY constraint name of the faculty table is FACULTY-PK |
| **R\_OWNER** | The owner of the constraint of the referenced table. Only used with FOREGIN KEY constraints |
| **Delete\_Rule** | Only used with Foreign Key tables. If the value is Cascade, if the parent row is deleted, i.e., the faculty row is deleted, then all dependent rows are deleted, i.e., all the courses taught by that faculty member. |
| **Status** | Enabled or Disabled |

|  |  |
| --- | --- |
| **USER\_CONS\_COLUMNS** | |
| **Column Name** | **Description** |
| **Owner** | The Owner of the constraint |
| **Constraint\_Name** | The name of the constraint. Using the above example there would be one row in this view named COURSE\_PK and another row name FACULTY\_COURSE\_FK |
| **Table\_Name** | The name of the table associated with the constraint. In both cases this would be the table named COURSE |
| **Column\_Name** | The name of the column associated with the constraint. The column CNO would be the name of the PRIMARY KEY constraint of the COURSE table and FNO would the name of the FOREIGN KEY constraint of the COURSE table |
| **Position** | The position of the column within the constraint definition. Since a constraint can have many columns (a composite key) then this system view must be separated from the constraint definition. |

### 4.2.1 Code and Execute the show\_constraints.sql script





Setting the LINESIZE does not affect the font size in Putty. By default SQL\*Plus display 80 characters on a line not matter what the font size is in Putty. SQL\*Plus will force the display of the result of the SELECT statement is wrap to the next line at 80 characters. Increasing the SQL\*Plus LINESIZE may still be wrapped by Putty. You can decrease the font size in Putty.

**Change the point size (or font) under Window/Appearance**. Click CHANGE under Font Settings. A point size of 14 that is bolded is easier to read, but you may get word wrap problems. Clicking on OK only affects the current session.



### 4.2.2 Document User Constraints

**Enter the SQL\*Plus command Clear Screen. Enter the SQL\*Plus command Clear Screen. Execute and Paste the Image for your show\_constraints.sql script below.** Answer =>

***I must see your Putty Banner***

The previous SQL script will not display the name of the column used in the primary key. The next example will

**Select C.Table\_Name, Column\_Name, Position**

**From User\_Constraints C, User\_Cons\_Columns Ccols**

**Where Constraint\_type = 'P' and**

**C.Constraint\_Name = Ccols.Constraint\_Name;**

To show all foreign key relationships for your tables you can use the following:

**Select Table\_Name, Constraint\_Name**

**From User\_Constraints**

**Where Constraint\_type = 'R';**

The previous SQL script will not display the name of the column used in the foreign key. Nor will it display information about the parent table. The next example will

**Select C.Table\_Name, Column\_Name, Position,**

**From User\_Constraints C, User\_Cons\_Columns Ccols**

**Where Constraint\_type = 'F' and**

**C.Constraint\_Name = Ccols.Constraint\_Name;**

{Need to be tested}

**Select C.Table\_Name, Column\_Name, Position, P.Table\_Name**

**From User\_Constraints C, User\_Constraints P, User\_Cons\_Columns Ccols**

**Where Constraint\_type = 'F' and**

**C.Constraint\_Name = Ccols.Constraint\_Name and**

**C.R\_Constaint\_Name = C.Constraint\_Name**

## 4.3 USER\_INDEXES (IND) and USER\_IND\_COLUMNS

The Primary key and Unique Constraints automatically have indexes created by Oracle. Other indexes may be created to provide better performance to access, order and group other columns. Similar to a constraint the column names are stored in a separate view because an index may have many columns.

Since an Index uses disk space, the USER\_INDEX view contains entries likes USER\_TABLES, i.e., tablespace, Initial\_Extent, Min\_Extents, Max\_Extents. This information will not be discussed here. When a table space is first created, by default have of the space is reserved for index space. You can however, change the allocation of table space between tables and indexes.

For sake of discussion consider the following SQL script:

CREATE INDEX FACULTY\_FNAME\_IND ON FACULTY(FNAME);

|  |  |
| --- | --- |
| **USER\_INDEXES (IND)** | |
| **Column Name** | **Description** |
| **Table\_Owner** | The Owner of the Table |
| **Index\_Name** | The name of the index. In this case the name of the index would be FACULTY\_FNAME\_IND |
| **Table\_Name** | The name of the table associated with the index. In this the tabvle name would be the table named FACULTY |
| **Uniqueness** | UNIQUE or NONUNIQUEe |
| **Status** | Enabled or Disabled |

|  |  |
| --- | --- |
| **USER\_IND\_COLUMNS** | |
| **Column Name** | **Description** |
| **Index\_Name** | The name of the index. In this case the name would be FACULTY\_FNAME\_IND |
| **Table\_Name** | The name of the table associated with the index. In this case it would be the table named FACULTY |
| **Column\_Name** | The name of the column associated with the index. In this case it would be column FNAME |
| **Position** | The position of the column within the index definition. Since an index can have many columns (a composite key) then this system view must be separated from the indexed definition. |
| **Column\_Length** | The indexed length of columns |

To display a list of indexes created

**Select Table\_Name, Index\_Name, Uniqueness**

**From USER\_INDEXES;**

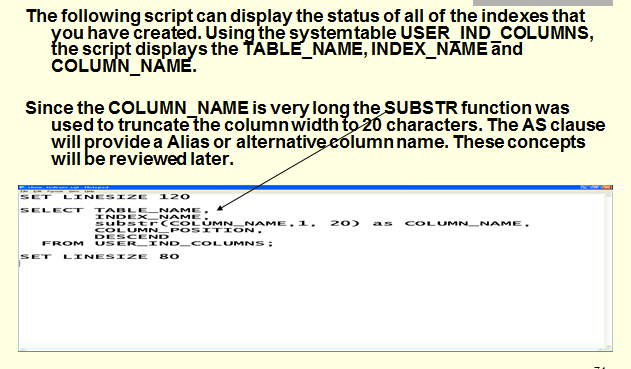
or

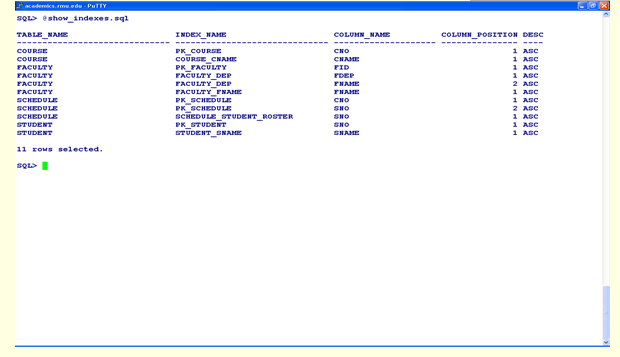
**Select Table\_Name, Index\_Name, Position, Column\_name**

**From USER\_INDEXES I, USER\_IND\_COLUMNS C**

**Where I.Index\_Name = C.Index\_Name;**

### 4.3.1 Code and Execute the show\_indexes.sql script



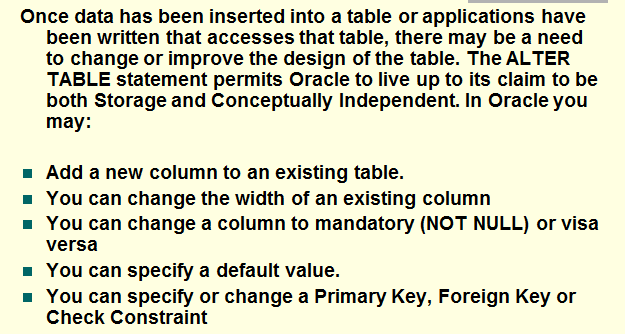


### 4.3.2 Document Indexes

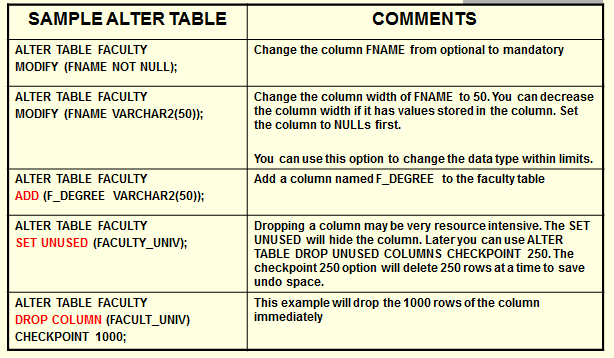
**Enter the SQL\*Plus command Clear Screen. Enter the SQL\*Plus command Clear Screen. Execute and Paste the Image for your show\_indexes.sql script below.** Answer =>

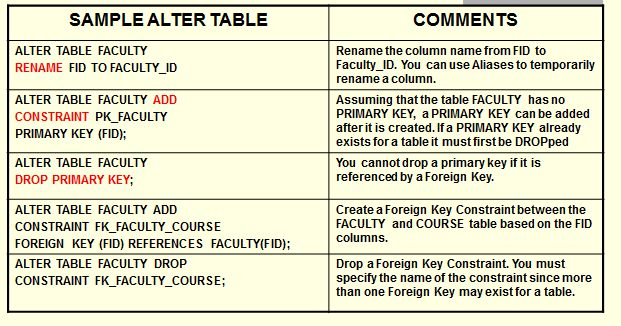
***I must see your Putty Banner***

# 5.0 ALTER TABLE - Modifying a Table Definition



## 5.1 ALTER TABLE EXAMPLES





**Adding a Column to an Oracle Table -**

[**http://www.youtube.com/watch?v=MwuF1dyK-oc&feature=fvwrel**](http://www.youtube.com/watch?v=MwuF1dyK-oc&feature=fvwrel)

**Altering an Oracle Table using SQL commands (Theory) -**

[**http://www.youtube.com/watch?v=xfvaQfpexRQ&feature=BFa&list=UL5D1dCiPV8Wg**](http://www.youtube.com/watch?v=xfvaQfpexRQ&feature=BFa&list=UL5D1dCiPV8Wg)

**Altering an Oracle Table using SQL commands (Hands-On) -**

[**http://www.youtube.com/watch?v=Qwq8HEA0lQI&feature=BFa&list=ULxfvaQfpexRQ**](http://www.youtube.com/watch?v=Qwq8HEA0lQI&feature=BFa&list=ULxfvaQfpexRQ)

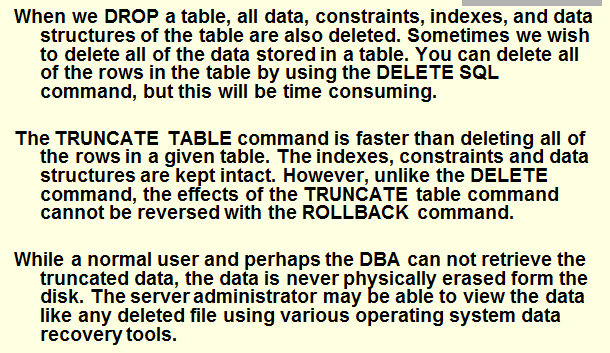
**Adding Foreign Keys**

[**http://www.youtube.com/watch?v=PTHHFOUobho&feature=fvwrel**](http://www.youtube.com/watch?v=PTHHFOUobho&feature=fvwrel)

**Creating a Sequence**

[**http://www.youtube.com/watch?v=tnQiCG6GJso&feature=fvwrel**](http://www.youtube.com/watch?v=tnQiCG6GJso&feature=fvwrel)

# 6.0 TRUNCATE TABLE



# 7.0 Making a COPY of a Table

