



Institute of Technology Tralee

Institiúid Teicneolaíochta Trá Lí

Database Concepts

Lab 04 – Database Creation using Scripts

In this lab you will learn:

- How to interpret a database schema
- How to build a database using an Oracle Script

Before You Start:

During this lab, you will be creating a new database in your tablespace. To ensure that there is no conflict with object names (table names, constraint names) you will need to remove any/all tables from your tablespace prior to generating the new database.

A script, *Demo_Clear.sql*, has been written to remove the tables you created during Lab03. This script contains the required SQL DROP TABLE statements to remove the previously created tables.

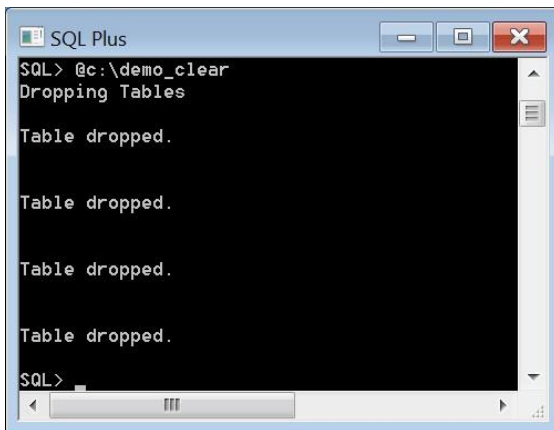
The script can be found on the X:\ drive at the following location:



Copy this Oracle script *Demo_Clear.sql* to your X:\Student\DBConcepts\Scripts folder. Place a copy on the C:\ drive for use during class.

Log on to the Oracle SQL*Plus environment and run *Clear_Demo.sql*.

To do this, enter @**Demo_Clear** at the SQL prompt.

A screenshot of a Windows-style window titled "SQL Plus". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is black with white text. The text shows a command prompt session where the user has entered "SQL> @c:\demo_clear" and the output is "Dropping Tables". This is followed by four lines of "Table dropped." indicating that four tables have been successfully removed. The prompt "SQL>" is visible at the bottom of the window.

```
SQL> @c:\demo_clear
Dropping Tables

Table dropped.

Table dropped.

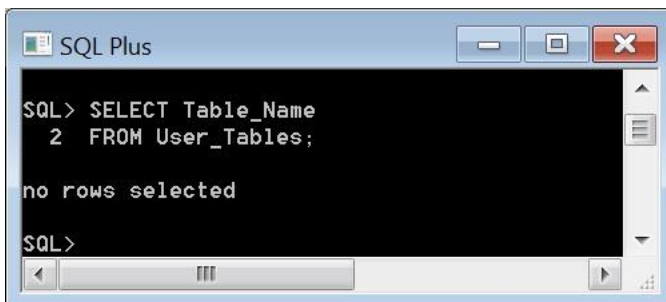
Table dropped.

Table dropped.

SQL>
```

Your tablespace should now be empty again.

Check that this is the case by selecting the *Table_Name* attribute from the system table *User_Tables*.

A screenshot of a Windows-style window titled "SQL Plus". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is black with white text. The text shows a command prompt session where the user has entered "SQL> SELECT Table_Name" and "2 FROM User_Tables;". The output is "no rows selected", indicating that the tablespace is empty. The prompt "SQL>" is visible at the bottom of the window.

```
SQL> SELECT Table_Name
2  FROM User_Tables;

no rows selected

SQL>
```

The Database Schema

Consider the following *database schema* which defines the database for the DVD rental system:

Schema: DVDSYS

Relation: Age_Ratings

Attributes: Age_Code char(3)
Description char(25)

Primary Key: Age_Code

Relation: Genres

Attributes: Genre_Code char(2)
Description char(15)

Primary Key: Genre_Code

Relation: Rates

Attributes: Rate_Code char(2)
Description char(15)
Rate numeric(3,2)

Primary Key: Rate_Code

Relation: DVDs

Attributes: DVD_Id numeric(4)
Title char(35)
Age_Code char(3)
Genre_Code char(2)
Rate_Code char(2)
Status char(1)

Primary Key: DVD_Id

Foreign Key: Age_Code REFERENCES Age_Ratings

Foreign Key: Genre_Code REFERENCES Genres

Foreign Key: Rate_Code REFERENCES Rates

Relation: **Members**
Attributes: Member_Id numeric(5)
 Surname char(20)
 Forename char(20)
 Street char(25)
 Town char(20)
 County char(20)
 Status char(1)
Primary Key: Member_Id

Relation: **Rentals**
Attributes: Rental_ID numeric(5)
 Member_Id numeric(5)
 DVD_Id numeric(4)
 Rate numeric(3,2)
 Rental_Date Date
 Due_Date Date
Primary Key: Rental_ID
Foreign Key: Member_Id REFERENCES Members
Foreign Key: DVD_Id REFERENCES DVDs

Building the Database

Create an Oracle script which builds the database defined in the database schema DVDSYS.

The script should contain:

- CREATE TABLE statements to create the database tables.
- Comments to explain the contents of the script (use the comment delimiter --).
- PROMPT statements to output progress of script execution.

DDL Statements

CREATE TABLE <table name>
(column1 data type/size [constraint],
column2 data type/size [constraint],
Column3 data type/size [constraint],
[CONSTRAINT <constraint name> PRIMARY KEY (att1, att2,...)]
[CONSTRAINT <constraint name> FOREIGN KEY (att1)][,.....]);

Ensure that you use meaningful names for table names and constraint names.
Use the following naming convention for primary key (PK) constraints:

pk_TableName (e.g. pk_Members)

Use the following naming convention for foreign key (FK) constraints:

fk_TableName_ReferencedTableName (e.g. fk_DVDs_Rates)

Remember: Constraint names must be unique.

It is possible to remove constraints (PK and/OR FK) from a database table *after* the database is live using the **ALTER TABLE / DROP CONSTRAINT** command. We will see this later in the semester.

Creating the Script File

Create an Oracle script (on your C:\ drive) to create the database described in the schema above.
Name this script *DVDSYS.sql*. (DO this from the Oracle SQL> prompt using the EDIT command).

SQL> ED C:\DVDSYS

Execute the script to create the database in your tablespace.

SQL> @C:\DVDSYS

Use the SQL*Plus DESC command and SQL SELECT statement to see the structure/data of the tables created.

IMPORTANT!

Test your script after each CREATE TABLE statement is entered.

You will need to execute/run your script many times as you correct errors you have made. Remember, you cannot create a table which already exists. To avoid this, your script should remove any existing table definitions before creating the tables.

To do this, include the required DROP TABLE statements before the CREATE TABLE statements.

DROP TABLE <table name>; (e.g. DROP TABLE Rentals;)

Execute the script to test it.

Referential integrity must be upheld at all times.

Where a relationship exists between two tables, the table doing the referencing (ie. The child table) must be dropped before the referenced table (i.e. the parent table).

The opposite applies when creating tables. The parent table must be created before the child table.

Changing the Database Definition

As systems evolve and are being tested, changes to a database definition may be required/requested.

For example:

- Increase the size of a column
- Reduce the size of a column
- Remove a column from a table
- Add a column to a table
- Define a constraint on a column or a table

Suppose the DBA has reviewed the DVDSYS database design and has made the following recommendations:

1. The attribute *Description* in the relation Genres should allow entries of up to **20** characters
2. The attribute *Rate* in the relations *Rates and Rentals* should accommodate values of numeric(4,2).
3. An attribute No_Out numeric(1) should be added to the table Members.
4. An attribute email char(25) should be added to the table Members.
5. Member Surname and Forename must never be NULL.

Do the following:

1. From the **SQL >** prompt, open the script for editing.

SQL> ED C:\DVDSYS

2. Make the required changes to the database definition and save the script.

File → Save (Exit)

3. Execute the script to implement the changes made.

SQL> @C:\DVDSYS

4. Describe the structures of the amended tables to ensure that the changes have been made.