DT228/2, DT282/2 Databases I
Getting Started with SQL

Objectives

The objectives of this lab are to:

- To learn how to create a connection to the DIT Oracle server
- To allow you to start familiarising yourself with SQL developer
- To allow you to become familiar with using SQL to create tables and insert data
- To create a data model in ERWin from SQL code.

Contents

1. Library Example from Lecture

1.1 Create Tables and Populate them In SQL Developer

1. As you have not set up any tables yet, you will need to create some.

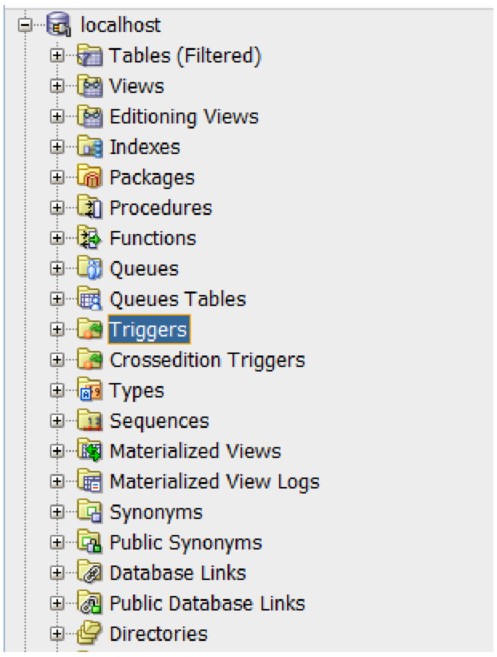
   Download the file L3-Library.sql from Webcourses and open it with a text editor (Notepad++ or whichever editor you prefer).
   This file includes commands to delete any tables relevant to this example that exist in your schema, recreate them and insert some data into them.
   Cut and paste the contents of this file into the SQL Worksheet window.

2. You need now to get the DBMS to execute these instructions.

   There are two buttons you can use to get the DBMS to do this
   The **first** (green arrow) will **run a single instruction** (the one you have highlighted in the window).
   The **second** (page with green arrow at the bottom) will **run all the instructions** currently in the window (this is the one you need to use here).
   Once you have run the script you will get output appearing in the window below.
   Browse through both the instructions and the output.
   NOTE: You will get some error notifications for this script, the start of the script includes instructions to cleardown i.e. delete all the tables etc to ensure the structures are correct.

3. Clear both the worksheet and the output window (use the button)

   In the left-hand window, **click on the + sign beside Tables**- this will display a list of tables that exist.



If no tables appear then you need to refresh your connection (sometimes it can be a little slow) then click on refresh connection button under the connections tab.

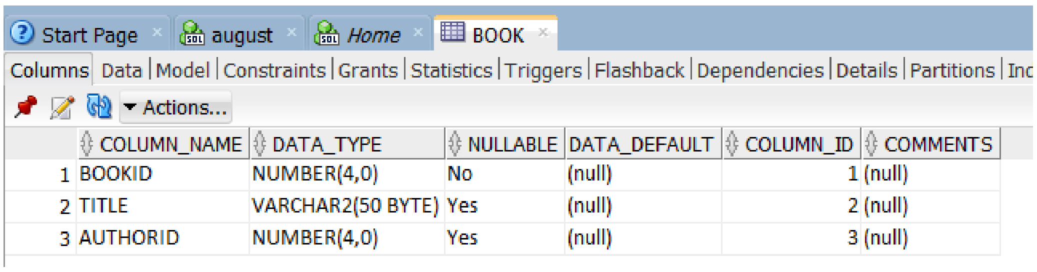
In LiveSQL (If using)

In the SQL WorkSheet window Paste in the SQL statements and click Run on the top right.

Once the tables have been created select Schema from the right hand menu and double-click the name of a table

1.2 Browsing the data

1. Double-click the **Book** table and the following window will be displayed on the right.



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BOOKID	NUMBER(4,0)	No	(null)	1	(null)
2	TITLE	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	AUTHORID	NUMBER(4,0)	Yes	(null)	3	(null)

Using the tabs at the top you can look at the structure of the table (and change it using the buttons displayed in that tab), the data, constraints etc, you can even generate the SQL to create the table in SQL tab. Play around with these and see what you can do.
Try to insert some data into the database and generate the SQL.

2. Go back to your SQL Worksheet where you created the tables (find the tab named with your connection).

   Clear all the commands if you haven't already.
   To view all the rows in the book table using SQL
   select * from book;
   To describe the structure of the book table:
   describe book;

   To execute these statements, click anywhere on the line and then click the green arrow.

3. Look at the data and structure of at least the other tables for our library example – use both the SQL Developer interface (or Schema in LiveSQL) and a SQL command.

Note: If you are not clear on how to do any of the following then make sure you ask your lab supervisor.

4. Write and execute a Select statement to return the first_name, last_name, street_address for all borrowers.

   HINT: select ______, ______, ____ from ____;

5. Change the select statement you created for 4. so that in the output you show FIRSTNAME, LASTNAME and STREETADDRESS in the output instead of the column names.

6. Return[1] the transactionid, borrowerid, bookid, transaction_date and transaction_type for all transactions.

   HINT: select _____, _____, _____, _____, _____ from _____;

7. Amend the select statement you created for 6. to only include transactions of type 1 and 2

   HINT: select _____, _____, _____, _____, _____ from _____ WHERE ___ ;

[1] This means write a select statement – get used to this terminology

8. To save your work

   Use either the File menu or the Save icon to save.
   Create a directory on your U drive which you will use to store the work for this module. Save the script in there.

1.3 Inserting Data

Try executing the following SQL statements.

They should result in errors.

Make sure you understand the errors before moving on to the next part of the lab.

If you are unsure of what the error means or why the error is happening then make sure you ask your lab supervisor.

1. INSERT INTO BORROWER VALUES (103, 'Sheppard', 'Sam', '70 Oaks Avenue', 'Mytown, MA 01234');

2. INSERT INTO TRANSACTION VALUES (100, 107, 104, SYSDATE, 1);

3. INSERT INTO BOOK (bookid, title, authorID) VALUES (120, 'Catcher in the Rye', 107);

4. INSERT INTO BOOK (bookid, title, authorID) VALUES (133, 'Cloud Atlas');

2. Creating Tables from Scratch Peter's Pets

From the following description of Peter's pet store, you are going to work out the details of the tables you need to create, the constraints you need to implement and put together the create statements and the insert statements you need to create and populate the data structures. Peter sells small animals (cats and small dogs), rodents (hamsters, guinea pigs and white mice) and also keeps a range of reptiles (lizards and snakes) to members of the public.

In order to manage all the information on animal sales in Peter's Pets a database containing 4 tables with information on animals, species, customers and sales of animals is needed. The following is a relational schema:

- Species (speciesCode, speciesName, speciesPrice)
- Animal (animalID, animalName, speciesCode)
- Customer (custID, custName, custEmail)
- AnimalSale (animalID, custId, saleDate)

Any attribute that is underlined above is part of the primary key of the relation (table) it appears in.

All identifiers (the ID Columns) should be numeric capable of holding up to 6 digits; all names (all columns with name in their name) should be alphanumeric capable of holding up to 30 characters; all date columns should be of type date.

Species Table: Primary Key: speciesCode. Holds details on all species Peters Pet's handles (e.g. dog, cat, hamster etc). Each species has a unique id (speciesCode), a name (speciesName) and a price (speciesPrice). Price should capable of holding values up to 999.99 (Remember Scale and Precision).

Animal Table: Primary Key: animalID. Holds details on each animal. Each animal has a unique id (animalID), a name which it is given for the duration of its stay in the shop (animalName). Each animal belongs to one species (speciesCode) (e.g. dog, cat, hamster etc.) which must exist in the Species table.

Customer Table: Primary Key: custID. Holds details on each customer who purchases an animal: a unique ID (custID), a name (custName) and email (custEmail) (alphanumeric 50 characters).

AnimalSale Table: Primary Key: (composite) animalID and custID. A customer can buy many animals and some customers regularly return to buy new pets. Details of the pets bought by each customer are recorded in the AnimalSales table. Details stored are the ID of the customer who bought the animal (custID), the ID of the animal of the animal purchased (animalID) and the date on which the animal was purchased (saleDate).

You are required to insert the following data. You need to generate the IDs for Animal, Species and Customer.

				Sold To		
AnimalName	**SpeciesID**	**Species**	**Price**	**CustomerID**	**Customer Name and Email**	**Date of Sale**
Tiny	1	Dog	9.99	1	D. Smith, dsmith@yahoo.co.uk	01 Jun 2017
Prince	1	Dog	9.99	2	B. Byrne, bb@gmail.com	11 Jun 2017
CJ	2	Cat	10.20	2	B. Byrne, bb@gmail.com	12 Mar 2017
Sid	2	Cat	10.20	3	X. Dobbs	04 Sep 2017
Sid	3	Snake	20.00	2	B. Byrne, bb@gmail.com	04 Sep 2017
Danger	4	Mouse	5.00			
Bonnie	1	Dog	9.99			

To insert a date value you enclose it in single quotes and the easiest way is to follow DD Mon YYYY format e.g. '01 Jun 2016'. Where no customer details or date of sale is included for an animal it means it has not been sold yet.

2.1 Write Create Table Statements

1. Write Drop table statements to drop each of the tables you intend to create.
2. Create the basic structures (tables with all columns of correct datatypes).
3. Identify your Primary keys.

2.2 Add Foreign Key Constraints

1. Work out your foreign keys and add your foreign key constraints.
2. Rerun your drops and your new creates to make sure they work.
3. (Hint: Pay attention to the order in which these need to happen now)

2.3 Insert Your Data

1. Work out your insert statements and run to test that your structures work.
2. Rerun your drops and creates.
3. Run your insert statements to make sure they work. (Hint: Pay attention to the order in which your inserts need to happen)

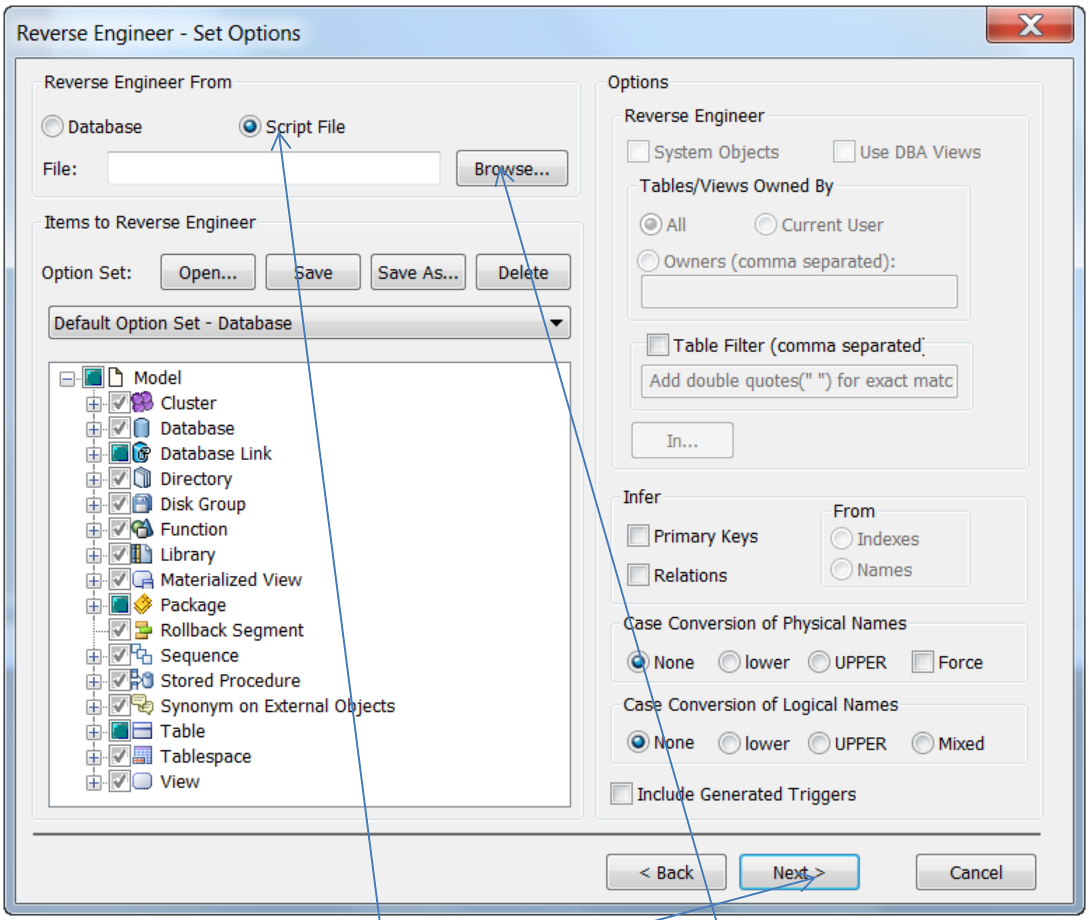
4. **Remember to include a commit after your inserts.**

2.4 Test your structures

1. Attempt to insert a row into each table that violates entity integrity.
2. Attempt to insert a row into each table that violates referential integrity.
3. Find the names of all animals and their species ID.
4. Find the names of all dogs
5. Find the dates of all animal sales;
6. Fine the animalIDs of all animals sold between 01 MAR 2017 and 30 JUN 2017.

3. Reverse Engineering a Data Model from SQL

1. Save your SQL to create the tables for Peters Pets as a .sql file.
2. Open ERWIN and create a logical/physical model.
3. Change your view to Physical/
4. From the Actions menu choose Reverse Engineer, choose logical/physical from the first dialog. The following screen will appear


Reverse Engineer - Set Options

Reverse Engineer From
Database [✓ Script File]
File:
Browse...

Items to Reverse Engineer
Option Set: Open... Save Save As... Delete
Default Option Set - Database

- Model
  - ✓ Cluster
  - ✓ Database
  - Database Link
  - ✓ Directory
  - ✓ Disk Group
  - ✓ Function
  - ✓ Library
  - ✓ Materialized View
  - Package
  - ✓ Rollback Segment
  - ✓ Sequence
  - ✓ Stored Procedure
  - ✓ Synonym on External Objects
  - Table
  - ✓ Tablespace
  - ✓ View

Options
Reverse Engineer
☐ System Objects ☐ Use DBA Views
Tables/Views Owned By
[✓ All] Current User
Owners (comma separated):
☐ Table Filter (comma separated)
Add double quotes(" ") for exact matc
In...

Infer
☐ Primary Keys
☐ Relations
From
Indexes
Names

Case Conversion of Physical Names
[✓ None] lower UPPER ☐ Force

Case Conversion of Logical Names
[✓ None] lower UPPER Mixed

☐ Include Generated Triggers

< Back Next > Cancel

5. Click on the Script File Radio button and then click Browse. Navigate to where you have saved the SQL file and select it.
6. Click Next and ERwin will generate a data model from your Create sql.