Laboratory 5
Getting Started with Oracle APEX

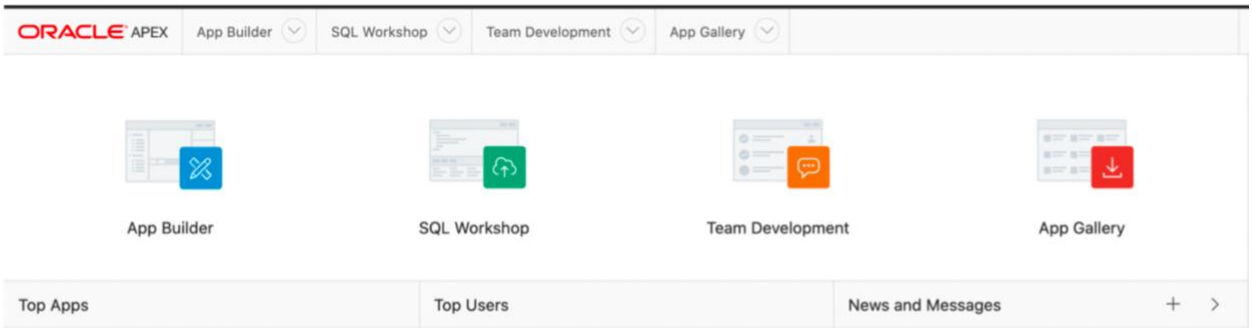
References
Lecture Notes: Topic 6 – Elmasri and Navathe, 2017: Chapter 6

From this lab onwards, you will use Oracle Application Express (APEX) to work on your SQL skills. Your APEX credentials have been released through LMS. If you do not have your credentials yet, check the latest announcement section on LMS or contact your lecturer.

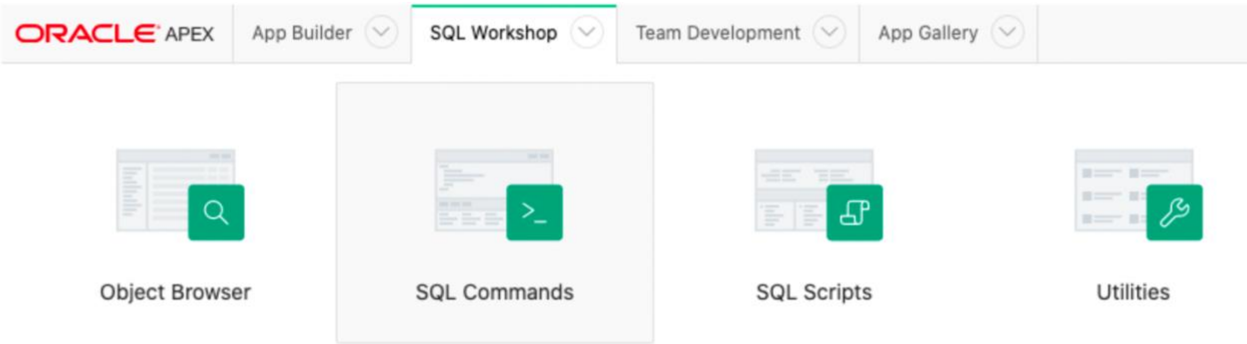
To login to Oracle Application Express:
1. Go to https://iacademy2.oracle.com
2. Enter workspace as provided.
3. Enter your student username as provided.

4. Enter your **password** as provided. Upon first access, you will be prompted to change your password (Please store the new password somewhere safe for future references).

After you login to your APEX account, please click on "**SQL Workshop**":



Then click on "**SQL Commands**" to get started:



From this point onwards, your lab tutor will help you get started.

Helpful Tips:

This page contains some helpful tips regarding structure of useful queries. However, they are not actual queries. You can refer to this page anytime during the lab. Please get started with Exercise 1 in page 4:

1. Create a table:

```
CREATE TABLE tableName(
   AttributeName_1 type,
   AttributeName_2 type, …
   AttributeName_5 type,
   …
PRIMARY KEY(attributeName_1),
FOREIGN KEY(attributeName_5)
REFERENCES tableName(matching-attributeName-in-tableName)
);
```

2. Insert a record:

```
INSERT INTO tableName
(attributeName_1, attributeName_2, …, attributeName_N)
VALUES
(value_1, value_2, …, value_N);
```

3. Drop column from a table:

```
ALTER TABLE tableName
   DROP (attributeName);
```

4. Add attributes to a table:

```
ALTER TABLE tableName
   ADD (attributeName1 type, attributeName2 type);
```

5. Change attribute type in a table:

```
ALTER TABLE tableName
   MODIFY attributeName type;
```

6. Update attribute’s value:

```
UPDATE tableName
SET attributeName = newValue
WHERE attributeName = searchValue;
```

7. Delete record from a table:

```
DELETE FROM tableName
WHERE attributeName = searchValue;
```

8. Drop a table:

```
DROP TABLE tableName;
```

9. Retrieve the table from the recyclebin:

```
FLASHBACK TABLE tableName TO BEFORE DROP;
```

10. Drop a table without considering the integrity constraint:

```
DROP TABLE tableName CASCADE CONSTRAINTS;
```

11. Display the tables in recyclebin:

```
SHOW RECYCLEBIN;
```

12. Drop a table from recyclebin

```
PURGE TABLE tableName;
```

13. Drop all tables from recyclebin:

```
PURGE RECYCLEBIN;
```

Discussion Questions

1. What is SQL?

2. What is Oracle?

Exercise 1 – Creating Tables, Adding Constraints and Inserting Data

(Note: Please make sure to clear the SQL Worksheet before you type and run the new SQL statement(s)**)**

1. At the SQL Worksheet type the following statement and write down your observations.

   ```
   SELECT *
   FROM TAB;
   ```

   TAB is a pre-defined table that stores, amongst other things, information regarding the tables in your database.

2. Then type the following SQL statement:

   ```
   CREATE TABLE Venues
   (venueID           VARCHAR2(6)     NOT NULL,
    venueName         VARCHAR2(50),
    venueAddress      VARCHAR2(70),
    venueCapacity     NUMBER(4)       NOT NULL,
    costPerDay        NUMBER(7,2),
    venueManager      VARCHAR2(50),
    managerPhoneNo    VARCHAR2(15),
   PRIMARY KEY (venueID));

   --YOU DO NOT NEED TO TYPE THIS
   --Things to note: The primary key is the attribute that
   --uniquely identifies each record
   --NOT NULL stipulates that this attribute MUST have a value
   --when a record is inserted into the table
   --As you may have guessed, the double dash -- signifies
   --comments
   ```

3. Clear your SQL Worksheet and type the following statement and write down your observations.

   ```
   SELECT *
   FROM TAB;
   ```

4. Clear your SQL Worksheet and type the following statement and write down your observations.

   ```
   DESCRIBE Venues;
   ```

5. Clear your SQL Worksheet and type the following SQL statement.

```
--NOTE: The format of Sean O'Riley below is NOT a typo!
--First try without the extra quote and see what happens.

INSERT INTO Venues
(venueID, venueName, venueAddress,  venueCapacity,
costPerDay, venueManager, managerPhoneNo)
VALUES
('V00001', 'Town Hall', '15 High St, Local Town', 800, 650.00,
'Sean O''Riley', '9333 2498');
```

Once you have successfully added the above row (Oracle will tell you when you are successful – "1 row(s) inserted"), clear your SQL Worksheet and try inserting the next record.

```
INSERT INTO Venues
(venueID, venueName, venueAddress, venueCapacity, costPerDay,
venueManager, managerPhoneNo)
VALUES
('V00001', 'Lyndhurst Street Community Centre', '12 Lyndhurst
   St, Local Town', 170, 310.00, 'Kylie Ong', '9333 1212');
```

What happens? Why?

6. Note that if you are entering values for all the attributes in a table, as in 5 above, you do not need to include the attribute names. Clear your SQL Worksheet and type the following SQL statement.

```
INSERT INTO Venues
VALUES
('V00003', 'Local Town Community Theatre', '146 Main Rd, Local
Town', 650, 1500.00, 'James McPhee', '9333 8569');
```

7. If you are only entering partial information you MUST specify the attributes and ensure that all NOT NULL fields have a value. Clear your SQL Worksheet and type the following SQL statement.

```
INSERT INTO Venues
(venueID, venueName, venueAddress, costPerDay, venueManager)
VALUES
('V00004', 'Glass Street Scout Hall', '6 Grange Rd, Local
Town', 20.00);
```

Any problems?

Fix the problem and try again.

8. Please create a table call `Event` in the database, table `Event` includes the following attributes, and the primary key is `eventId`, foreign key is `venueId`, please define these attributes appropriately.

```
Events(eventId, eventName, eventDescription,
venueCapacityRequired, cateringRequired, venueId)
```

9. How do you ensure `cateringRequired` in the table above is only given the values 'y' or 'n'?

10. What is the SQL statement to display all the records in the Venue table? Try it and see.

11. Please insert one record into event table by yourself.

12. Clear your SQL Worksheet and type the following SQL statement.

```
INSERT INTO Events
(eventID, eventName, eventDescription, venueCapacityRequired,
cateringRequired, venueID)
VALUES
('E00002', 'Little Athletic''s Trivia Night', 'Fundraiser for
the local little athletics club', 250, 'n', 'V00009');
```

What happens? Think carefully about the significance of referential integrity.

Exercise 2 – Altering Tables and Modifying Data

1. Please drop attribute `venueManager` from table `Venues` without recreating the table.

2. Type the following statements and write down your observations.

```
DESCRIBE Venues;

SELECT * FROM Venues;
```

3. Please add three attributes: `streetAddress`, `addressCity`, `addressPostcode` into table `Venues` without recreating the table.

4. Type the following statements and write down your observations. Take particular note of the order in which the attributes are listed. What impact will this have on future insert statements?

```
DESCRIBE Venues;
```

5. Please change the type of attribute `addressCity` to `VARCHAR2(200)` without recreating the table.

6. Type the following statements and write down your observations.

```
DESCRIBE Venues;
```

7. Type the following statements and write down your observations.

```
ALTER TABLE Venues
MODIFY venueName   VARCHAR2(5);

DESCRIBE Venues;
```

8. Write the SQL statement to update the street address of the venues with id 'V00001' to '12 Lyndhurst St'.

   Display the data stored in the `Venues` table.

9. Write the SQL statement to delete the venue with `venueId` 'V00002' from the `Venues` table.

Display the data stored in the `Venues` table.

10. Write the SQL statement to delete the venue with `venueId` 'V00001' from the `Venues` table.

What happens? Display the data stored in the `Venue` table.

Display the data stored in the `Venues` table.

Exercise 3 – Deleting Tables

1. Type the following statement and write down your observations.

   ```
   DROP TABLE Venues;
   ```

   What happens?

2. Correct the code to drop the table `Venues` that has a referential integrity constraint with another table?

3. Type the following statements and write down your observations.

   ```
   SELECT * FROM TAB;
   ```

4. When you issue the drop table command the table is placed in the recycle bin, thereby allowing you to retrieve it if necessary. If you want to retrieve the table, what command you need to type?

5. When you selected the tables in the database after dropping the `Venues` table you will have noticed that the representation of the `Venues` table does not make it easy to identify. What command you need to type to identify `Venues` table in recycle bin?

6. To delete a table from the recycle bin you need to "purge" it. Please type code to delete the table `Venues` from recycle bin. After that, type the following statements and write down your observations.

   ```
   PURGE TABLE Venue;
   SELECT * FROM TAB;
   ```

7. Please type the code to delete all the tables in your recycle bin.

A word on style…

The commands in SQL are not case sensitive. The mixed use of upper- and lower-case in this lab is for readability only. HOWEVER, it is suggested you follow a similar format when writing your SQL statements.

For more practice try the following from your text book (Elmasri and Navathe, 2014):

Q4.10, Q4.11, Q4.12, Q4.13 and Q4.14 (pages 107 – 109)