References: `SQL Language Reference`

Learning Objective:

1. Using the DELETE command
2. Creating database views
3. GRANTing object privileges

Topic: The effects of Deleting a record

We will attempt to remove all of the food order number 3001 from the **order_list** table.

First, query the table to show the records.

1. Remove the records with order number 3001 from the **order_list** table.
   ```
   DELETE FROM order_list

   WHERE order_no = [insert order_no here];
   ```
   Check that the records have been deleted.

2. What do you think would happen if the following is used:
   ```
   DELETE FROM order_list;
   ```

   ____

   Proceed to execute the above statement. Query the table to show its contents.

*Recall that when you logged in to the Oracle server, a SESSION is created for you (refer practical 1). NONE of the updates that you are doing will be updated to the database unless you actually issue a command to update the database. Only the data in your session (a copy of the actual data from the database) are updated.

To reverse the changes done using the INSERT, UPDATE, DELETE commands use:

```
ROLLBACK;
```

Now, query the table to see its contents.

If you want to update the changes you've made to the database, issue the **COMMIT** command.

ROLLBACK cannot reverse any changes that has been **COMMIT**ted.

3. Now, repeat Step 1, then issue the COMMIT command.
   Now, issue a ROLLBACK command. Query the table to see its contents.

   What happened? Why?

   ____

   ____

   ____

4. Try to delete all the staff, customer and category records.
   The commands would be:

   ____

   ____

   ____

   What happened? Why?

   ____

   ____

   ____

   ____

   ____

Topic: Creating a VIEW

If you need to display each menu item and the respective category description, the command would be:

```
select M.menu_id, M.description, M.price,
       C.description
from menu M, category C
where M.cat_id = C.cat_id;
```

If you need to associate the menu item and the respective category quite often in your queries, you may want to create a view.

```
CREATE VIEW MENU_CATEGORY as
select M.menu_id, M.description, M.price,
       C.description
from menu M, category C
where M.cat_id = C.cat_id;
```

Now, to view the menu item and the respective category, just issue the command

```
select * from menu_category;
```

5. You can also create a view to simplify or hide details of a table.

```
CREATE VIEW short_menu as

select menu_id, description, price
from menu;
```

Now, to view the most menu details, just issue the command

```
select * from short_menu;
```

Can you create a view from another view?

____

____

6. Now create a view to show customer information and the menu items they have ordered. (Hint: refer to the ERD of the restaurant system to determine how many tables to use.)

____

____

____

Topic: The GRANT command

As part of database security control, the DBA can assign system and object privileges to the different classification of users (please read or Google to find out more).

The non-DBA user can control privileges on objects (tables, views, stored procedures, etc) that they owned.

Let's use the ***STAFF*** table for practice. Recall in Practical 5 that you have inserted some of your own detail into the staff table.

Currently, only you can query the table.

```
select * from Staff;
```

All students can only see their own records in the STAFF table.

Give another student the privilege to query your STAFF table:

```
GRANT Select
ON STAFF
TO <login username>;

e.g. Assuming the owner of STAFF is user_DIA88
GRANT Select
ON STAFF
TO user_DIA99;
```

Now, user_DIA99 can execute the following query:

```
    select *
    from user_DIA88.Staff;
```

What is the result?

____

____

Experiment with GRANTING Insert, Update, Delete.

How do you remove the privilege that you have given to other users?

____

____