

Lab 8

Data Modification Language (DML), Transaction Control Language (TCL), Data Definition Language (DDL)

1. Learning Objectives

After the successful completion of this lab, you will be able to

- Explain the key concepts below:
 - DML Statements: INSERT, DELETE, UPDATE
 - Transaction Control Statements: COMMIT, ROLLBACK
 - DDL Statements: CREATE/DROP TABLE, CREATE INDEX/SEQUENCE
 - Database Constraints: Primary Key, Foreign Key
- Read SQL statements that use the key concepts above
- Write SQL statements that use the key concepts above

2. Tasks to Complete

Complete the questions about DML, DCL, and DDL statements on the **MGS Database** included in the later part of this document. These queries use the tables in **user mgs**.

NOTE: the links to online Oracle SQL Language references are available in the **Canvas Page: Links to Oracle SQL Language References**.

RELATED KNOWLEDGE POINTS

- **DML Statements**
 - Change the actual data stored in the relational database.
 - **INSERT:** insert a single row or multiple rows into a table
 - **Single row:**
`INSERT INTO table_name(column1, column2, ...)`
`VALUES(column1_value, column2_value, ...)`
 - **DELETE:** delete one or more existing rows from a table
 - `DELETE FROM table_name WHERE`
 - **UPDATE:** change selected columns in a table
 - `UPDATE table_name`
`SET column1 = new_value, column2 = new_value, etc`
`WHERE`
- **TCL Statements**
 - Determine at which point changes made to the actual data via DML statements in the current transaction should be made permanent or cancelled.
 - **COMMIT:** make permanent the DML changes in current transaction
 - **ROLLBACK:** cancel the DML changes made in current transaction
- **Database Constraints**
 - The rules about what data can be stored in the relational database
 - **PRIMARY KEY**
 - Specify that a single column or a combination of multiple columns in a table must be unique
 - Use keywords **PRIMARY KEY**
 - **Foreign Key**

- Specify data dependency between two or more tables
- Specify that data in a single column or a combination of columns in one table must refer to data in one or more other tables or NULL otherwise
- Use keyword REFERENCES
- Table level constraint, use keywords FOREIGN KEY
- Child table
 - A table that contains a foreign key constraint
 - Example: employee table
- Parent table
 - A table that is referenced by another table via a foreign key.
 - Example: department table
- Due to data dependency, operations on data and data structure must follow the Foreign key rules
 - Insert parent row before child row
 - Delete child row before parent row (default rule unless using ON DELETE CASCADE|SET NULL)
 - Can not update a column in a parent row if it's referenced by a child row
 - Create parent table before child table
 - Drop all child tables before parent table
-
- DDL Statements add or change the structure of the relational database
 - CREATE TABLE: add a new table with selected database constraints
 - DROP TABLE: remove an existing table (both structure and actual data)
 - CREATE INDEX: add an index for a column in a table
 - CREATE SEQUENCE: add a sequence to a table

3. Submission Requirements

WHEN

- See the lab canvas page for the exact due date.

WHAT

- A text file with the extension **.sql**, including all your SQL statements.
- The file is in the following format.

Mark each query based on the question number. Write your FULL name on the first page.

Sample:

```
--Lab8
--Your full name
```

```
--Q1
Your solution.
--Q2
Your solution.
...
```

HOW

- Submit your SQL script file by attaching it to the link **Lab 8** in folder **Assignments\Labs** on Canvas.

4. Grading

This assignment is graded **based on the CORRECTNESS of YOUR ANSWER.**

The point distribution among questions will be posted soon.

If your answer to a question is not 100% correct, you will get partial credits.

Exercises on Data Modification Language and Transaction Control Language on MGS Database

NOTE:

To verify whether a table has been modified correctly as you do these exercises, you can write and run an appropriate **SELECT** statement, as I showed you in the related lectures.

1. Write an **INSERT** statement that adds this row to the **Categories** table:

category_id: 5

category_name: Percussion

Code the **INSERT** statement as part of a script so that the **Oracle server makes this insertion permanent.**

NOTES:

- It's **REQUIRED** that you submit **TWO** SQL statements for this question.
- One is a **TCL** statement.

2. Write an **UPDATE** statement that changes the name of the musical instrument category with ID 5 to **String**.

NOTES:

- For this question, your query must choose the data based on only the data you are given: the ID.
- In a typical situation, you don't know the category name when writing this query unless you run another query to get it.

3. Write a **DELETE** statement that deletes the **String** category without knowing the category ID. Assume that category names are unique.

NOTES:

- For this question, your query must choose the data based on only the data you are given: the category name.
- In a typical situation, you don't know the category id when writing this query unless you run another query to get it.

4. Write an **INSERT** statement that adds the new row below to the **Products** table:

product_id: 11

category_id: 4

product_code: YDP162R
product_name: Yamaha Arius YDP162R Traditional Console Style Digital Piano
description: The best keyboard on the market. Offers excellent sound rendering that truly separates it from the rest of the pack.
list_price: 1599.99
discount_percent: 10
date_added: the date and time when this INSERT statement is executed

It's **REQUIRED** that your statement **USE a COLUMN LIST**.

HINT: See the knowledge points in Pages 1-2.

5. Write a DELETE statement that deletes the category with ID 4. When you execute this statement, it will produce an error since there are products in the Products table in this category.

To fix that, precede the DELETE statement with another DELETE statement that deletes all products in this category.

Code these **two DELETE statements** as part of a script so Oracle CANCELS the changes.

NOTES:

- It is **REQUIRED** that you submit **TWO DELETE statements in the CORRECT ORDER** (due to **FOREIGN KEY constraint**), then one transaction control statement so Oracle DB server cancels the deletions.
- **To code two or more statements in a script, you must end each statement with a semicolon.**

6. Write an INSERT statement that adds the row below to the **Customers** table:

customer_id: 9
email_address: john@yahoo.com
password: funny
first_name: John
last_name: Oliver

It's **REQUIRED** that your statement **USE a COLUMN LIST**.

HINT: See the knowledge points in Pages 1-2.

7. Write an UPDATE statement that changes the passwords of all customers to `hello`, which contains all lowercase characters.

HINT: character literals are enclosed within single quotes and are case sensitive.

8. Write an UPDATE statement that changes the passwords of the customers who placed more than one order to `excellent`, which contains all lowercase characters.

NOTE: You don't have prior knowledge of who placed more than one order when writing this query.

HINT: you need to use a Subquery and an aggregation function in your SQL statement.

9. Code a transaction control statement that cancels the changes made in the current transaction.

10. Open the script named `create_mgs_tables.sql` and run this script.

Rerun the script will make sure that the data in the database are restored for future labs.

NOTES:

- **IT'S IMPORTANT THAT YOU COMPLETE THIS STEP.**
- **OTHERWISE YOU MAY NOT COMPLETE FUTURE LABS CORRECTLY.**

Exercises on Data Definition Language

11. Create two tables with the following information:

Table **Event**:

Each event has a unique event ID like 123, a date, and a location when the event was held, and a menu ID of the menu that was served in the event. In each event, exactly one menu is provided.

Table **Menu**:

Each menu has a unique menu ID like 45, a description of the menu, and the type of the menu.

REQUIREMENTS:

- You **MUST LIST** your **CREATE TABLE** statements in the **CORRECT ORDER, MEANING**
 - The two tables can be actually created without any error when running the two statements in that listed order.
- You **MUST DEFINE NECESSARY CONSTRAINTS** such as primary key, foreign key, etc.

HINTS:

- One table is the parent table. The other table is a child table.
- Create the parent table before the child table.

12. Write two DDL statements to drop the two tables you created in 11.

REQUIREMENT

- You **MUST LIST** your **DROP TABLE** statements in the **CORRECT ORDER, MEANING**
 - The two tables can be actually dropped without any error when running the two statements in that listed order.

HINTS:

- One table is the parent table. The other table is a child table.
- Drop the child table before the parent table.

13. Write a DDL statement to create an index named **event_date_ix** for the **event date** in table **event**.

14. Write a DDL statement to create a sequence named **menu_id_seq**. This sequence will be used to auto-generate menu ids in table menu.

Note: You can use all default settings when creating the sequence.

15. Use the sequence menu_id_seq to insert the following row into table menu.

Menu id: next number in the sequence menu_id_seq

Description: Menu 1 for vegetarians

Type: Vegetarian