Lab Manual for Introduction to Database Systems

Lab-02: SQL DML (INSERT, UPDATE, DELETE, SELECT INTRODUCTION)

Table of Contents

Lab 2: SQL DML (INSERT, UPDATE, DELETE, SELECT INTRODUCTION)

1. Introduction

In this lab you will learn about the INSERT, UPDATE, DELETE, SELECT, and INTRODUCTION. INSERT statement is used to insert records in the database, UPDATE statement is used to update records of tables. Delete statement is used to delete records from tables. Throughout this lab you will learn how to use these statements.

Relevant Lecture Material

a) Text Book: Java: Text Book: Database Systems, A practical approach to design, implementation and management by Thomas Connolly, Carolyn Begg, Addison Wesley , Fifth Edition,
   1. Read URL:
      i. https://www.studytonight.com/dbms/

2. Activity Time boxing

Table 1: Activity Time Boxing

Task No.	Activity Name	Activity time	Total Time
6.2	Walkthrough Tasks	30mins	60mins
7	Practice tasks	20 to 30mins for each task	50mins
8	Evaluation Task	60mins for all assigned task	40mins

3. Objective of the experiment

- To get familiar with inserting, updating and deleting records from tables.
- To get familiar with SELECT statement.

4. Concept Map

In this section, a brief overview of the concepts is presented, those will be used in this lab afterwards. The examples in the following sections are based on "Oracle.db" database.
All of the operation is performed on Oracle database. First we will import this database then we have used it. The database is available on data server. The procedure of Import and export are given in Appendix.

4.1. Inserting Data using INSERT INTO Statement

There are multiple ways to insert new records using the INSERT INTO statement.

4.1.1. Inserting single record

To insert single record with all column values, the syntax is:

```
INSERT INTO tableName VALUES (column1Value, ..., columnNValue);
```

You need to list the values in the same order in which the columns are defined in the CREATE TABLE, separated by commas. For columns of string data type (CHAR, VARCHAR), enclosed the value with a pair of single quotes (or double quotes). For columns of numeric data type (INT, DECIMAL, FLOAT, DOUBLE), simply place the number.

For Examples:

To insert a record in a jobs table, job table contain following attribute:

- Job id (varchar(50))
- Job title (varchar(50))
- Min Salary (int)
- Max Salary (int)

INSERT INTO jobs VALUES (‘Se_PROG’, ‘Senior Programmer‘, 1000, 2000);

4.1.2. Inserting multiple records

Multiple records can also be inserted in one INSERT INTO statement:

```
INSERT INTO tableName VALUES
  (row1FirstColumnValue, ..., row1lastColumnValue),
  (row2FirstColumnValue, ..., row2lastColumnValue),
  ... ;
```

For Example: To in record of two Jobs:

INSERT INTO jobs VALUES
(‘Web_PROG’, ‘Web Programmer‘, 1000, 2000),
(‘AND_PROG’, ‘Android Programmer‘, 2000, 3000);

4.1.3. Inserting Single Record with Specific Columns

To insert single record with some specific column values, the syntax is:

```
INSERT INTO tableName
        (column1Name, …, columnNName)
        VALUES
        (column1Value, ..., columnNValue);
```

To insert a record in a specific column:

For Example:

```
INSERT INTO jobs (job_id,job_title,min_salory)
VALUES ('IOS_PROG', 'IOS Developer', 2000);
```

4.1.4. Inserting Multiple Record with Specific Columns

Multiple records can also be inserted in one INSERT INTO statement:

```
INSERT INTO tableName
        (column1Name, …, columnNName)
        VALUES
        (column1Value, ..., columnNValue),
        (column1Value, ..., columnNValue)
        … ;
```

To in multiple record in a specific column:

For Example:

```
INSERT INTO jobs (job_id,job_title,min_salory)
VALUES ('C_PROG', 'C_Sharp Programmar', 2000),
VALUES ('J_PROG', 'Java Programmer', 3000);
```

Note:

- A column defined with NOT NULL constraint and no DEFAULT constraint, must have a value i.e. it cannot be NULL.

4.1.Modifying Data using UPDATE Statement

To modify existing data, use UPDATE ... SET statement, with the following syntax:

```
UPDATE tableName SET columnName = {value|NULL|DEFAULT}, ... WHERE criteria;
```

For Examples:

To modify *max_salory* of all jobs to "25000", the following statement can be written:

```
UPDATE jobs SET max_salary = 25000;
```

To modify *min_salory to* 5000 and *max_salory* to 10000, against all jobs who's job_id is IT_PROG, the following statement can be written:

```
UPDATE jobs
SET min_salary = 5000, max_salary = 10,000
WHERE job_id = “IT_PROG”;
```

4.2.Deleting Rows using DELETE FROM Statement

Use the DELELE FROM statement to delete row(s) from a table, with the following syntax:

```
DELETE FROM tableName WHERE criteria;
```

For Examples:
To delete all jobs the following statement can be written:

```
DELETE from jobs;
```

To delete all jobs of IT_PROG, the following statement can be written:

```
DELETE from jobs WHERE job_id = “IT_PROG”;
```

Note: Deleted data cannot be recovered.

5.1. Reading the data from database using SELECT statement.

The most common, and important task is to query a database for a subset of data that meets your needs with the SELECT statement. The output of the select statement is a two-dimensional table, known as the ResultSet. Please note that, to use the SELECT statement a database must be selected first using USE statement, see section 6.3.3 of Lab 01.

The SELECT command has the following syntax:

```
SELECT column1Name, column2Name, ... FROM tableName;
```

For example,
To select all rows from the table *jobs* for the columns *job_id*, and *job_title*
mysql> SELECT name, price FROM products;

job_id	job_title
AC_ACCOUNT	Public Accountant
AC_MGR	Accounting Manager

5. Homework before Lab

You must solve the following problems at home before the lab.

5.2.Task-2

Describe the difference between where and having clause.

6. Procedure & Tools

In this section, procedure of the tasks and setup of required tools is defined.

6.1.Starting MySQL CLI) [Expected time = 5mins]

Refer to Lab 1 sec 6.2.

6.2.Walkthrough Task [Expected time = 30mins]

This section defines the tasks with examples to better understand the concepts and experiments we are going to perform in this lab.

6.2.1. Inserting Data

Insert a record of jobs as shown in *Figure 1*.

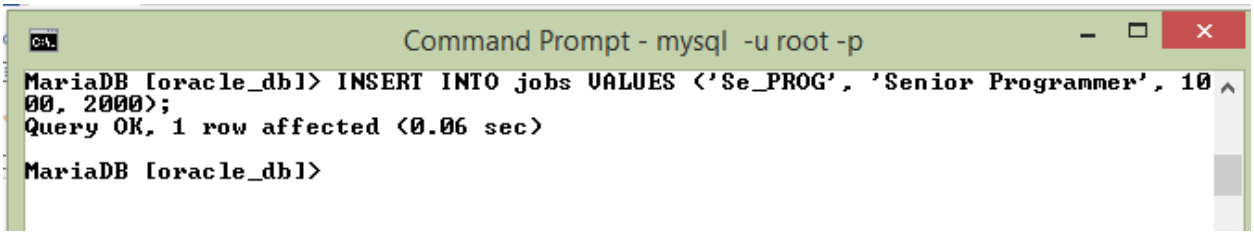

Figure 1: Inserting a record of an jobs

6.2.2. Updating data

Update max_salary of all jobs to 50000, see *Figure 2*.

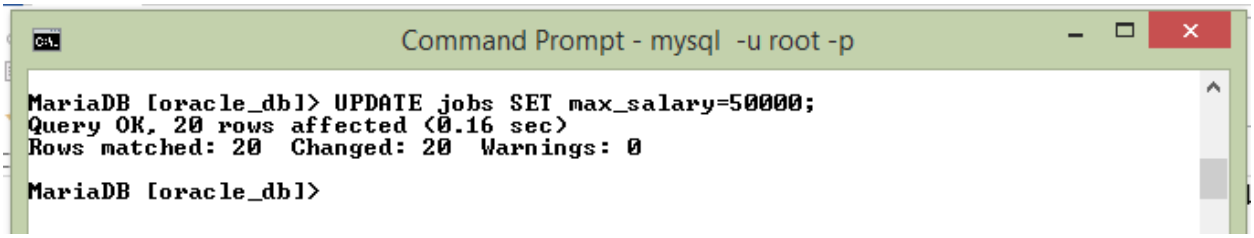

Figure 2: Update *max_salary* of all jobs to 50000

Update *max_salary* to 100,000 of jobs whose *job_id* is 'IT_PROG', see Figure 3.

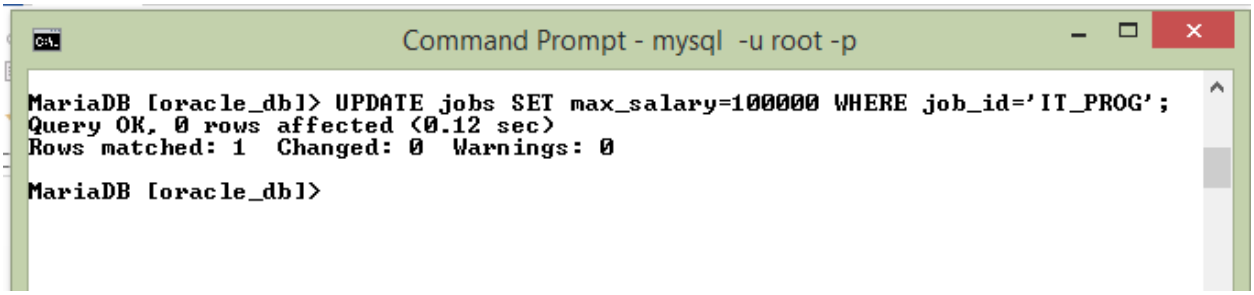

Figure 3: Update *max_salary of jobs_id is IT_PROG*

CAUTION: If the WHERE clause is omitted in the UPDATE statement, ALL ROWS will be updated. Hence, it is a good practice to issue a SELECT query, using the same criteria, to check the result set before issuing the UPDATE. This also applies to the DELETE statement in the following section.

6.2.3. Deleting Rows

Delete all jobs of the job_id is Se_PROG, See Figure 4.

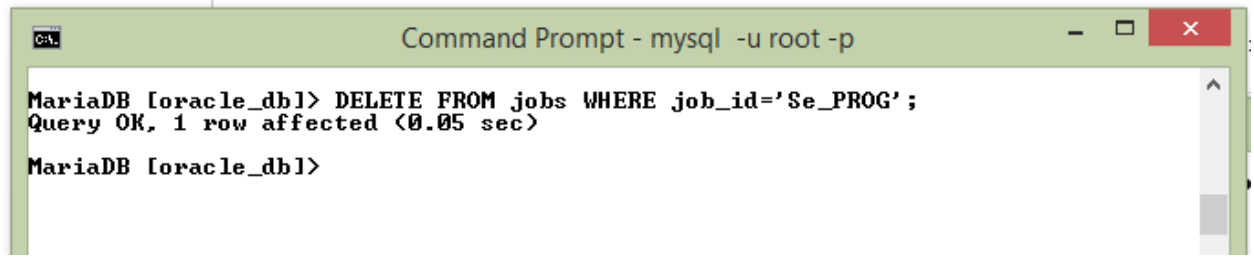

Figure 4: Deleting jobs

6.2.4. Select statement for all rows and all columns.

To select all rows and all columns of jobs, execute the following query, see Figure 4.

```
SELECT * from jobs;
```

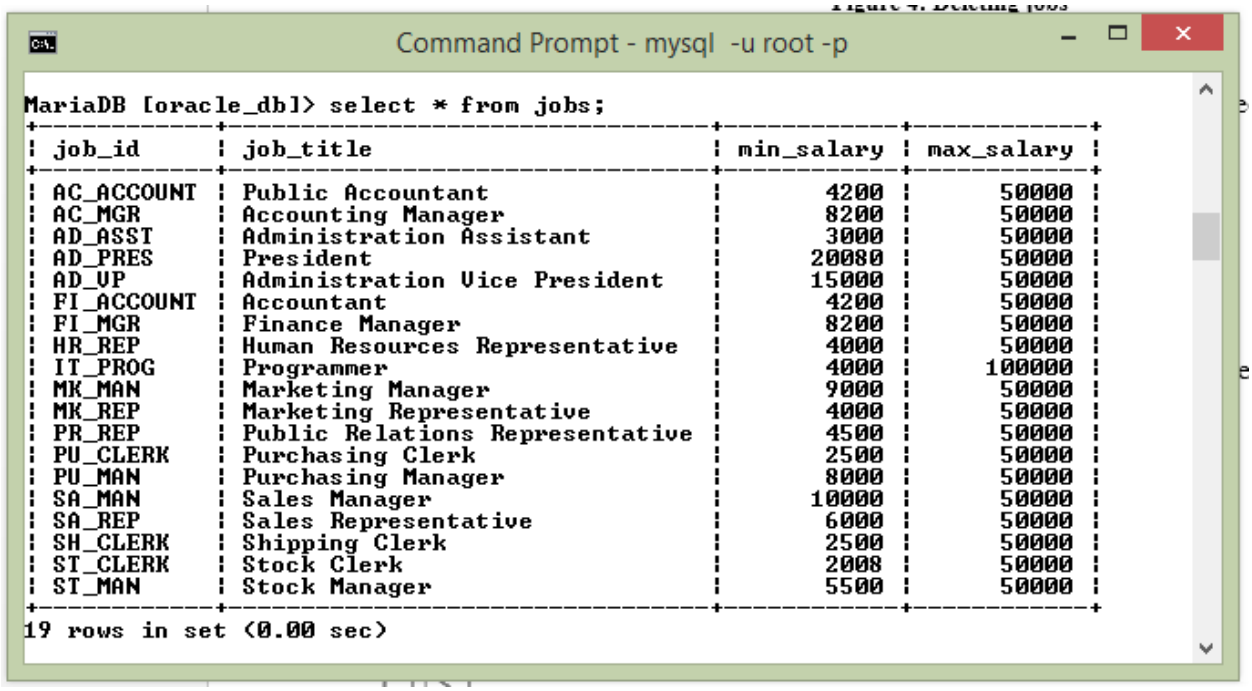

Figure 5: Select all rows for all columns

6.2.5. Select command for specific column.

To select all rows and jobs_id and job_title of jobs table, execute the following query, see Figure 5.

SELECT job_id, job_title from jobs

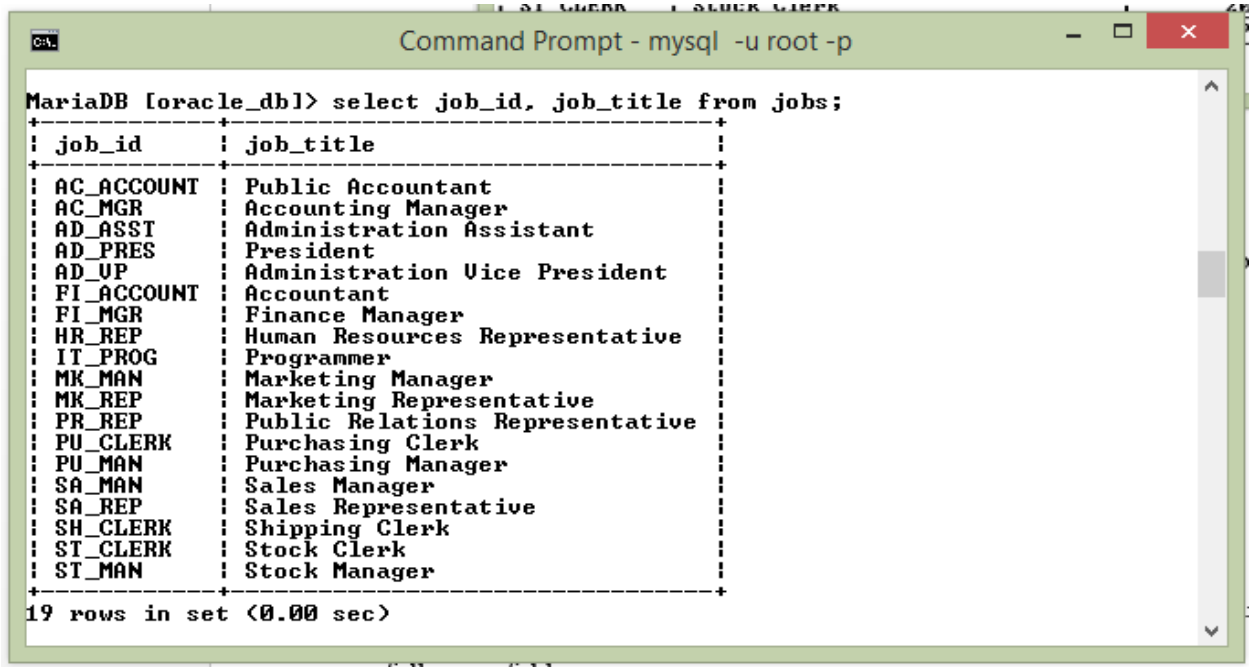

Figure 6: Select Statement for Specific Column.

7. Practice Tasks

This section will provide more practice exercises which you need to finish during the lab. You need to finish the tasks in the required time. For practice task you will use the given schema. When you finish them, put these tasks in the following folder:
\\fs\assignments$

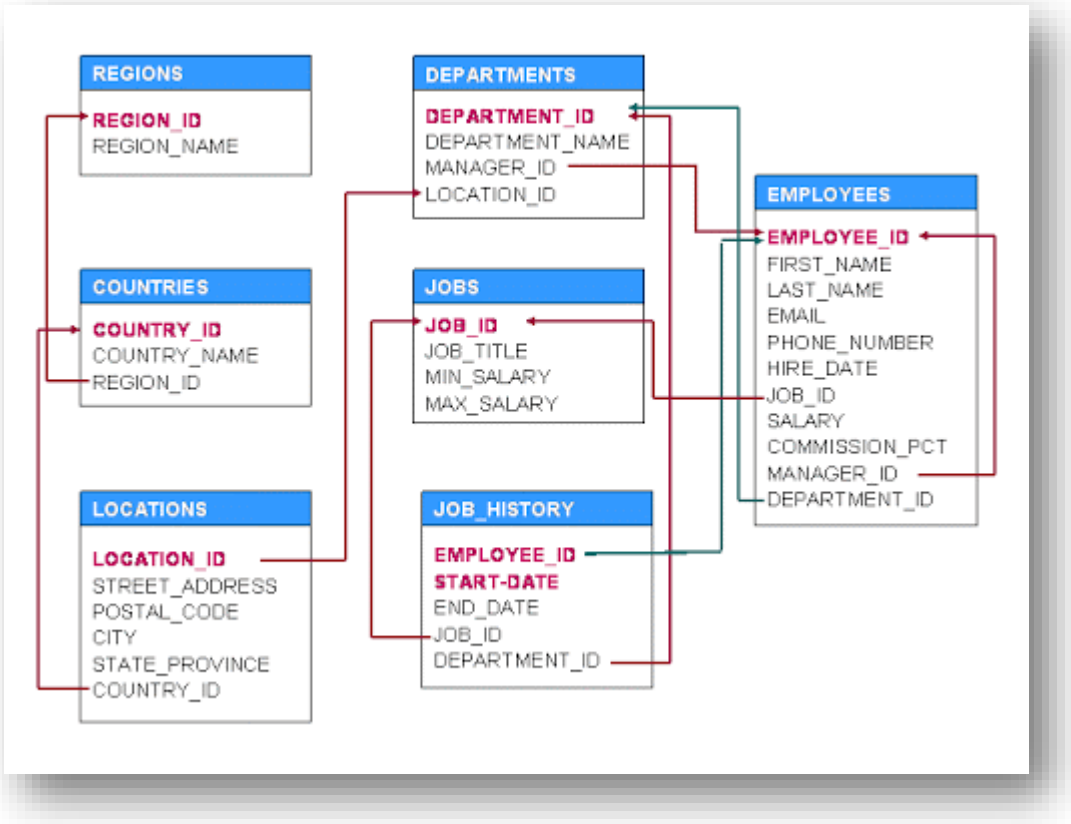

Figure 7: HR Database

7.1.Practice Task 1 **[Expected time = 15mins]**

Use the "HR" database (Database file is present in \\fs\lectures$), write down following SQL queries.

1. Insert a new job with the following information
    a. Job id is "P_PROG"
    b. Job title is "Python Programmer"
    c. Min Salary is 60000
    d. Max Salary is 100000
2. Update the name of Department whose department_id is 60, the new name is Information Technology.
3. Update the minimum and maximum salary of the above inserted record, the new Minimum salary is 20,000 and max salary is 40,000.
4. Add 5000 to maximum salary of all jobs whose id is FI_MGR.

5. Delete the inserted new record of location table.
6. Select all the records of Countries table.
7. Select all the records of job_history table.
8. Select all the records of regions table.
9. Select location_id and streetaddress of location table.
10. Show all employees whose job_id is IT_PROG.
11. Show all location information including location_id, street_address etc of US country.
12. Show first_name, last_name, email of employee whose job_id is FI_MGR.

7.2.Out comes

After completing this lab, student will be able to insert, update, and delete data from a database. They will also be able to create complex queries involving sorting, getting distinct records, and limiting the number of records to be fetched.

8. Evaluation Task (Unseen) [Expected time = 60mins for two tasks]

The lab instructor will give you unseen task depending upon the progress of the class.

9. Evaluation criteria

The evaluation criteria for this lab will be based on the completion of the following tasks. Each task is assigned the marks percentage which will be evaluated by the instructor in the lab whether the student has finished the complete/partial task(s).

Table 3: Evaluation of the Lab

Sr. No.	Task No	Description	Marks
1	6	Procedures and Tools	05
2	7	Practice tasks and Testing	15
3	8	Evaluation Tasks (Unseen)	80

10. Further Reading

This section provides the references to further polish your skills.

10.1. Books

Text Book

Database Systems, A practical approach to design, implementation and management by Thomas Connolly, Carolyn Begg, Addison Wesley , Fifth Edition,

10.2. Slides

The slides and reading material can be accessed from the folder of the class instructor available at \\fs\lectures$\

11. REFERENCES:

11.1. SQL-99 Complete, Really, by Peter Gulutzan & Trudy Pelzer.

- More examples for the SELECT statement: http://dev.mysql.com/doc/mysql/en/select.html
- MySQL operators: http://dev.mysql.com/doc/mysql/en/non-typed_operators.html

Appendix

11.2. Exporting and Importing databases

A database, created on MySQL server can be exported and imported to the same or other server. Database is exported to an SQL dump file. SQL dump file contains database structure as well as the data. This helps in the following:

- Taking and restoring backup.
- Moving the database to another server.

11.3. Exporting the data to SQL dump file

A database can be exported to SQL dump file. To export MySQL database MySQLDUMP utiltiy is used. Write following statement in command line:

```
MySQLDUMP -u username -p dbname > dbexport.sql
```

For Example,
MySQLDUMP -u root -p *oracle_db* > Oracle.sql

The file *Oracle.sql* should now contain the database *oracle_db*.

11.4. Importing the data from SQL dump file

A database can be imported into MySQL server from SQL dump file. To import MySQL database, write following statement in command line:

```
MySQL -u username -p dbname < dbexport.sql
```

For Example,
MySQL -u root -p *oracle_db* < Oracle.sql

The *oracle_db* database should now contain the data that is in the Oracle.sql file.