Lab4

Objectives:

At the end of this lab, you should be able to

- Use DML: Insert, update, delete.
- Database Transactions: commit, rollback, and SavePoint

Data Manipulation Language (DML)

Data manipulation language (DML) is a core part of SQL. When you want to add, update, or delete data in the database, you execute a DML statement.

The INSERT Statement

```
INSERT INTO   table [(column [, column...])]
VALUES        (value [, value...]);
```

Example

```
INSERT INTO        dept (deptno, dname, loc)
  VALUES     (50, 'DEVELOPMENT', 'DETROIT');
```

Note: the column list is not required in the INSERT clause if it contains values for each column in the table.

```
INSERT INTO        dept   VALUES        (60, 'HR', 'DOHA');
```

Inserting Rows with Null Values:

Implicit method:

```
INSERT INTO     dept (deptno, dname )  VALUES(60, 'MIS');
```

Explicit method

```
INSERT INTO dept  VALUES (70, 'FINANCE', NULL);
```

Practice:

- Add the following employee to EMP table
  No : 200 Name: AHMED Salary: 4000 Hiredate: 1/1/2008
- Add the following department to DEPT table
  No : 80 Name: IT Location: None.

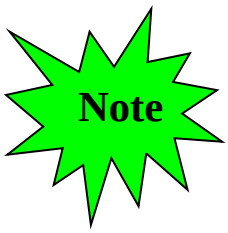
Prepared by :Dr. Saleh Alhazbi

The UPDATE Statement

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE      condition];
```

Example

UPDATE emp SET deptno = 20 WHERE empno = 7782;



- If you do not put a condition, update will be applied to all records in the table.
- If you attempt to update a record with a value that is tied to an integrity constraint, you will experience an error, for example try to update deptno 20 into 50 in the table dept

Practice:

- Write a SQL statement to change the location column of department table with 'Doha' for department number 80.
- Write a SQL statement to change salary of employee to 8000 whose ID is 7499, if the existing salary is less than 5000.

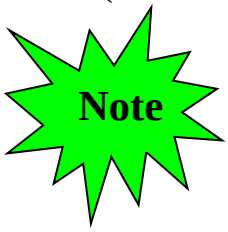
The DELETE Statement

```
DELETE [FROM]   table
[WHERE          condition];
```

Example

DELETE FROM department WHERE dname = 'DEVELOPMENT';

DELETE FROM employee WHERE deptno =
(SELECT deptno FROM dept WHERE dname ='SALES');



Integrity Constraint Error:

SQL> DELETE FROM dept WHERE deptno = 10;

Practice:

- Write a SQL statement to delete from the table all employees in the IT department (number 80).
- Write a SQL statement to delete from the table all employees who work in departments with a budget greater than or equal to $60,000.
- Write a SQL statement to delete from the table all employees.

Database Transactions

When Does a Transaction Start and End?

A transaction begins when the first executable SQL statement is encountered and terminates when one of the following occurs:

- A COMMIT or ROLLBACK statement is issued
- A DDL statement, such as CREATE, is issued
- A DCL statement is issued
- The user exits SQL*Plus
- A machine fails or the system crashes

Explicit Transaction Control Statements

You can control the logic of transactions by using the COMMIT, SAVEPOINT, and ROLLBACK statements.

Statement	Description
COMMIT	Ends the current transaction by making all pending data changes permanent
ROLLBACK [TO *SAVEPOINT name*]	ROLLBACK ends the current transaction by discarding all pending data changes; ROLLBACK TO SAVEPOINT rolls back the current transaction to the specified savepoint, thereby discarding the savepoint and any subsequent changes. If you omit this clause, the ROLLBACK statement rolls back the entire transaction.

Examples:

```
SQL> DELETE FROM EMP;
SQL>ROLLBACK;
SQL>UPDATE emp  SET  deptno = 10  WHERE empno = 7782;
SQL> SELECT ename,deptno from emp where empno=7782;
SQL>Rollback;
SQL>UPDATE emp  SET  deptno = 10  WHERE empno = 7782;
SQL>COMMIT;
SQL> SELECT ename,deptno from emp where empno=7782;
```

Implicit Transaction Processing

Status	Circumstances
Automatic commit	DDL statement or DCL statement is issued Normal exit from SQL*Plus, without explicitly issuing COMMIT or ROLLBACK
Automatic rollback	Abnormal termination of SQL*Plus or system failure

There is an environment variable called AUTOCOMMIT. By default it is set to OFF. A user can set it to ON or IMMEDIATE by typing.

SQL> set autocommit on
OR
SQL>set autocommit immediate

When AUTOCOMMIT is set to ON or IMMEDIATE, every DML statement is written to the DISK as soon as it is executed.

Example of savepoint:

Assuming that the t2 Table exists in the

```
SQL> select * from t2;
  NUM
 ----------
    20
SQL> insert into t2 values(30);
1 row created.
SQL> savepoint s1;
Savepoint created.
SQL>  insert into t2 values(40);
1 row created.
SQL> savepoint s2;
Savepoint created.
SQL> select * from t2;
  NUM
 ----------
    20
    30
    40
SQL> rollback to s1;
Rollback complete.
SQL> select * from t2;
  NUM
----------
    20
    30
```