|  |
| --- |
| Department of Software Engineering  Mehran University of Engineering and Technology, Jamshoro |

|  |  |  |  |
| --- | --- | --- | --- |
| Course: SW222 – Database Management & Administration | | | |
| Instructor | Ms. Shafiya Qadeer | **Practical/Lab No.** | 11 |
| Date | 26-02-2021 | **CLOs** | 2 |
| Signature |  | **Assessment Score** | 2 Marks |

|  |  |
| --- | --- |
| Topic | To become familiar with Data retrieval, Data manipulation and transaction commands. |
| Objectives | * To become familiar with Select Statement, Insert, Update, Delete, Roll back, Commit |

|  |
| --- |
| Lab Discussion: Theoretical concepts and Procedural steps |

**Data retrieval**

Data retrieval, in database management, involves extracting the wanted data from a database. The two primary forms of the retrieved data are reports and queries.

In order to retrieve the desired data the user present a set of criteria by a query. Then the Database Management System (DBMS), software for managing databases, selects the demanded data from the database. The retrieved data may be stored in a file, printed, or viewed on the screen.

A query language, such as Structured Query Language (SQL), is used to prepare the queries. SQL is an American National Standards Institute (ANSI) standardized query language developed specifically to write database queries. Each DBMS may have its own language, but most relational DBMSs also support SQL.

Select statements are used to retrieve data from SQL tables. The Select statement illustrated below retrieves all of the columns and rows from the named table.

**SQL SELECT Statement**

The SELECT statement is used to select data from a database. The result is stored in a result table, called the result-set.

The six clauses of the SELECT statement:

There are six clauses that can be used in an SQL statement. These six clauses are SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY. Clauses must be coded in a specific sequence.

SELECT Select\_List

FROM Table\_List

[WHERE Conditions]

[GROUP BY Column\_List]

[HAVING Conditions]

[ORDER BY Column\_List]

• SELECT and FROM are required; the rest of these clauses are optional and serve to filter or limit, aggregate or combine, and control the sort.

• The WHERE clause is where you indicate a condition. This helps you filter unwanted data from the results. WHERE gives you a subset of the rows in a table.

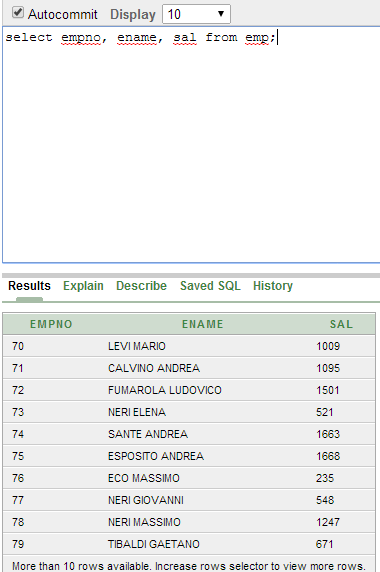
• GROUP BY allows you to group your data to achieve more meaningful results.

• HAVING puts a condition on your groups.

• ORDER BY orders your result rows. You can choose to order results by ASC (ascending) or DESC (descending) order. The default is ASC.

Example:

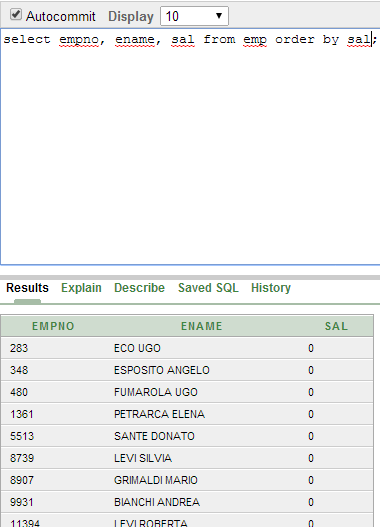
1.



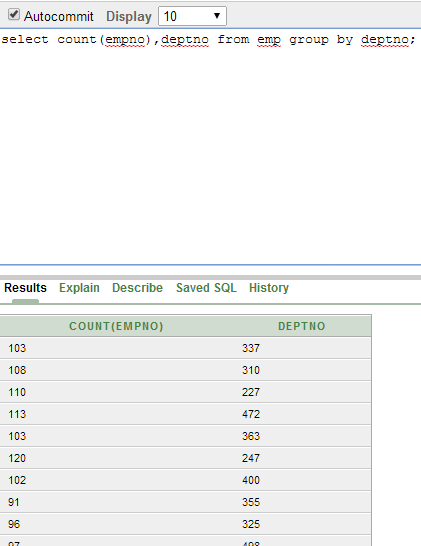
2)



3)



4)



**Data Manipulation Language (DML) Statements**

Data manipulation language (DML) statements access and **manipulate data** in existing schema objects. These statements do not implicitly **commit** the current transaction. The data manipulation language statements are:

**INSERT - insert data into a table**

**Syntax:**

The below syntax can be followed if the values for all the columns in the table is definite and known.

INSERT INTO table

VALUES (column1 value, column2 value,

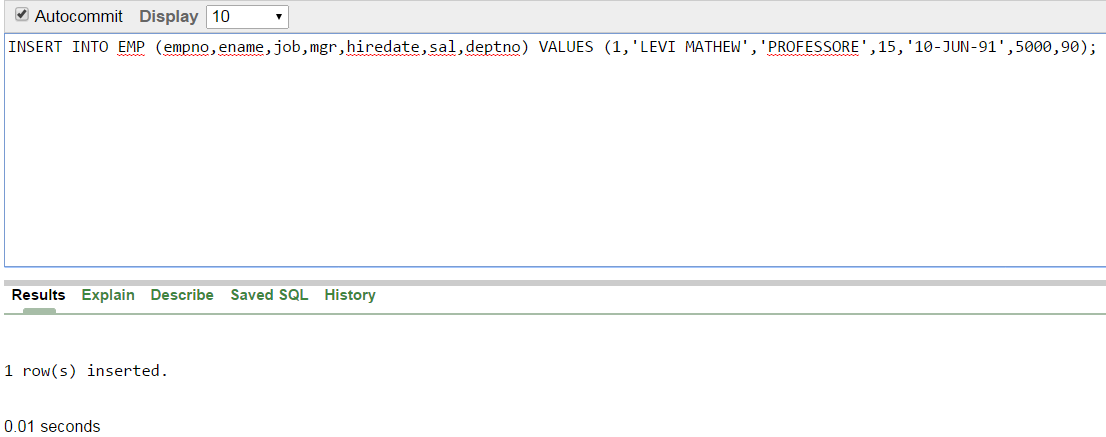
...);

The below syntax can be used if only few columns from the table have to be populated with a value. Rest of the columns can deduce their values either as NULL or from a different business logic.

INSERT INTO table (column1 name, column2 name, . . .)

VALUES (column1 value, column2 value, . . .);

**UPDATE - updates existing data within a table**



**Syntax:**

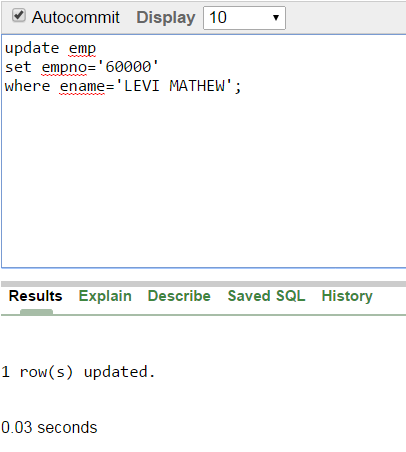
UPDATE table

SET column = value [, column = value ...]

[WHERE condition]

From the syntax,

The SET column = expression can be any combination of characters, formulas, or functions that will update data in the specified column name. The WHERE clause is optional, but if it is included, it specifies which rows will be updated. Only one table can be updated at a time with an UPDATE command.



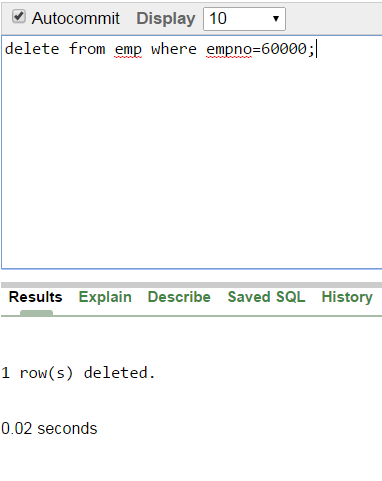
**DELETE - deletes all records from a table, the space for the records remain**

Syntax:

DELETE FROM table\_name

[WHERE condition];

The DELETE command deletes all rows in the table that satisfy the condition in the optional WHERE clause. Since the WHERE clause is optional, one can easily delete all rows from a table by omitting a WHERE clause since the WHERE clause limits the scope of the DELETE operation.



**Transaction Control Language (TCL)**

Transaction Control Language (TCL) commands are used to manage transactions in the database. These are used to manage the changes made to the data in a table by DML statements. It also allows statements to be grouped together into logical transactions.

**COMMIT command**

COMMIT command is used to permanently save any transaction into the database.

When we use any DML command like INSERT, UPDATE or DELETE, the changes made by these commands are not permanent, until the current session is closed, the changes made by these commands can be rolled back.

To avoid that, we use the COMMIT command to mark the changes as permanent.

Following is commit command's syntax,

COMMIT;

**ROLLBACK command**

This command restores the database to last commited state. It is also used with SAVEPOINT command to jump to a savepoint in an ongoing transaction.

If we have used the UPDATE command to make some changes into the database, and realise that those changes were not required, then we can use the ROLLBACK command to rollback those changes, if they were not commited using the COMMIT command.

Following is rollback command's syntax,

ROLLBACK TO savepoint\_name;

**SAVEPOINT command**

SAVEPOINT command is used to temporarily save a transaction so that you can roll back to that point whenever required.

Following is savepoint command's syntax,

SAVEPOINT savepoint\_name;

In short, using this command we can name the different states of our data in any table and then rollback to that state using the ROLLBACK command whenever required.

**Using Savepoint and Rollback**

Following is the table class,

|  |  |
| --- | --- |
| id | name |
| 1 | Abhi |
| 2 | Adam |
| 4 | Alex |

Lets use some SQL queries on the above table and see the results.

INSERT INTO class VALUES(5, 'Rahul');

COMMIT;

UPDATE class SET name = 'Abhijit' WHERE id = '5';

SAVEPOINT A;

INSERT INTO class VALUES(6, 'Chris');

SAVEPOINT B;

INSERT INTO class VALUES(7, 'Bravo');

SAVEPOINT C;

SELECT \* FROM class;

NOTE: SELECT statement is used to show the data stored in the table.

The resultant table will look like,

|  |  |
| --- | --- |
| Id | name |
| 1 | Abhi |
| 2 | Adam |
| 4 | Alex |
| 5 | Abhijit |
| 6 | Chris |
| 7 | Bravo |

Now let's use the ROLLBACK command to roll back the state of data to the savepoint B.

ROLLBACK TO B;

SELECT \* FROM class;

Now our class table will look like,

|  |  |
| --- | --- |
| id | name |
| 1 | Abhi |
| 2 | Adam |
| 4 | Alex |
| 5 | Abhijit |
| 6 | Chris |

Now let's again use the ROLLBACK command to roll back the state of data to the savepoint A

ROLLBACK TO A;

SELECT \* FROM class;

Now the table will look like,

|  |  |
| --- | --- |
| id | name |
| 1 | Abhi |
| 2 | Adam |
| 4 | Alex |
| 5 | Abhijit |

So now you know how the commands COMMIT, ROLLBACK and SAVEPOINT works.

|  |
| --- |
| Lab Tasks |

1. Display the name and job title for everyone that's in the table.
2. Display the employee name, job, and salary for everyone that does not belong to the job title "Tecnico".
3. Display all columns for everyone whose dept no: equals 108.
4. Select all the data of employees having hire date equals to 09-Jun-81.
5. Select all the data of employees that work in department 147 or department 50.
6. Select all the data of employees who are being managed by the manager having id equals to 1725.
7. Create a query that renames the ename column as employee\_name.
8. Create a query to display unique employee names from the emp table.
9. Create a query to display ename and job title as single column having column name as "information".( Hint: Use concatenation operator)
10. Create a query that selects ename and job title and uses the literal character strings to display the result like: **“John is a Professor.”**
11. Create an SQL command that will modify the hire date of all the employees having job title as PROFESSORE.(Set the new hire date = 20 Aug 1989).
12. Create two different forms of the INSERT command to add two different employees with an ID of 65798(and 65799) and name Andrew Lopez(Drew More) to the emp table.
13. Update the data of any two columns of newly created row.
14. Write a query to delete that newly created row.
15. Create 3 save points and show how it works along with commit and rollbacks.