SQL Programming Lecture 2

Department of Information Technology
Rajiv Gandhi College of Engineering and Research, Nagpur.

SQL | Comments

- Comments can be written in the following three formats:
- Single line comments.
- Multi line comments
- In line comments
- **Single line comments:**
- Comments starting and ending in a single line are considered as single line comments.
- Line starting with '–' is a comment and will not be executed.
- Syntax:

```
-- single line comment
-- another comment
SELECT * FROM Customers;
```

SQL | Comments

- **Multi line comments**
- Comments starting in one line and ending in different line are considered as multi line comments.
- Line starting with '/*' is considered as starting point of comment and are terminated when '*/' is encountered.
- Syntax:

```
/* multi line comment another comment*/
SELECT * FROM Customers;
```

SQL | Comments

- **In line comments**
- In line comments are an extension of multi line comments, comments can be stated in between the statements and are enclosed in between '/*' and '*/'.
- Syntax:

```
SELECT * FROM /* Customers; */
```

Examples

- **Multi line comment ->**

```
/* SELECT * FROM Students;
SELECT * FROM STUDENT_DETAILS;
 SELECT * FROM Orders;
*/ SELECT * FROM Articles;
```

- **In line comment ->**

```
SELECT * FROM Students;
SELECT * FROM /* STUDENT_DETAILS;
SELECT * FROM Orders;
 SELECT * FROM */ Articles;
```

DQL (Data Query Language)

- DML statements are used for performing queries on the data within schema objects.
- The purpose of DQL Command is to get some schema relation based on the query passed to it.
- **Example of DQL:** SELECT_

SQL | SELECT

- The SELECT Statement in SQL is used to retrieve or fetch data from a database.
- With this statement, we specify the columns that we want to be displayed in the query result.
- Syntax

```
SELECT column1, column2, columnN
FROM table_name
 WHERE <Condition>;
```

- Here, column1, column2... are the fields of a table whose values you want to fetch. If you want to fetch all the fields available in the field, then you can use the following syntax.

```
SELECT * FROM table_name;
```

Example: Consider the CUSTOMERS table having the following records

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmadabad	20000.00
2	Khilan	25	Kota	15000.00
3	Kaushik	23	Delhi	20000.00
4	Chaitali	25	Mumbai	65000.00
5	Hardik	27	Bhopal	85000.00
6	Rahul	22	Pune	45000.00
7	Muffy	24	Indore	10000.00

The following code is an example, which would fetch the ID, Name and Salary fields of the customers available in CUSTOMERS table.

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS;
```

ID	NAME	SALARY
1	Ramesh	20000.00
2	Khilan	15000.00
3	Kaushik	20000.00
4	Chaitali	65000.00
5	Hardik	85000.00
6	Rahul	45000.00
7	Muffy	10000.00

The following code is an example, which would fetch all the fields of the CUSTOMERS table

```
SELECT * FROM CUSTOMERS;
```

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmadabad	20000.00
2	Khilan	25	Kota	15000.00
3	Kaushik	23	Delhi	20000.00
4	Chaitali	25	Mumbai	65000.00
5	Hardik	27	Bhopal	85000.00
6	Rahul	22	Pune	45000.00
7	Muffy	24	Indore	10000.00

The following code is an example, which would fetch all the fields of the CUSTOMERS table

```
SELECT * FROM CUSTOMERS;
```

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmadabad	20000.00
2	Khilan	25	Kota	15000.00
3	Kaushik	23	Delhi	20000.00
4	Chaitali	25	Mumbai	65000.00
5	Hardik	27	Bhopal	85000.00
6	Rahul	22	Pune	45000.00
7	Muffy	24	Indore	10000.00

Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- DML commands :
  - INSERT
  - UPDATE
  - DELETE

INSERT

- The INSERT statement is a SQL query. It is used to insert data into the row of a table.
- **Syntax:**

```
INSERT INTO TABLE_NAME
(col1, col2, col3,.... col N)
VALUES (value1, value2, value3, .... valueN);
```

Or

```
INSERT INTO TABLE_NAME
VALUES (value1, value2, value3, .... valueN);
```

INSERT

- **For example:**

  INSERT INTO Student (ROLL_NO, NAME, SUBJECT) VALUES (1,“Raju", "DBMS");

  Or

- INSERT INTO StudentVALUES (1,“Raju", "DBMS");

- **Queries Sample Table: Student**

ROLL_NO	NAME	SUBJECT
1	Raju	DBMS

UPDATE

- This command is used to update or modify the value of a column in the table.
- **Syntax:**

  UPDATE table_name

  SET [column_name1= value1,...column_nameN = valueN]

  [WHERE CONDITION]

- **For example:**

  UPDATE students

  SET Name = ‘Aman' WHERE ROLL_NO = ‘1'

ROLL_NO	NAME	SUBJECT
1	Aman	DBMS

DELETE

- It is used to remove one or more row from a table.
- **Syntax:**

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

- **For example:**

```
DELETE FROM Student
WHERE NAME="Aman";
```

ROLL_NO	NAME	SUBJECT

DCL(Data Control Language)

- DCL commands deals with the rights, permissions and other controls of the database system
- .**Examples of DCL commands:**
- **GRANT**-gives user's access privileges to database.
- **REVOKE**-withdraw user's access privileges given by using the GRANT command.

Grant Command

- **Allow a User to create table**
- To allow a user to create tables in the database, we can use the below command,

GRANT CREATE TABLE TO username;

- **Allow a User to create session**
- When we create a user in SQL, it is not even allowed to login and create a session until and unless proper permissions/priviliges are granted to the user.
- Following command can be used to grant the session creating priviliges.

GRANT CREATE SESSION TO username;

Grant Command

- **To take back Permissions**
- if you want to take back the privileges from any user, use the REVOKE command.

  REVOKE CREATE TABLE FROM username

TCL(TRANSACTION CONTROL LANGUAGE)

- TCL commands deals with the transaction within the database.
- **Examples of TCL commands:**
- **COMMIT**– commits a Transaction.
- **ROLLBACK**– rollbacks a transaction in case of any error occurs.
- **SAVEPOINT**–sets a savepoint within a transaction.
- **SET TRANSACTION**–specify characteristics for the transaction.

SQL Data Types

- Data types mainly classified into three categories for every database.
- String Data types
- Numeric Data types
- Date and time Data types

MySQL String Data Types

- **CHAR(Size)**
- **VARCHAR(Size)**
- **BINARY(Size)**
- **VARBINARY(Size)**
- **TEXT(Size)**
- **TINYTEXT**
- **MEDIUMTEXT**
- **LONGTEXT**
- **ENUM(val1, val2, val3,…)**

MySQL Numeric Data Types

- **INT(size)**
- **INTEGER(size)**
- **FLOAT(size, d)**
- **DOUBLE(size, d)**
- **DECIMAL(size, d)**
- **DEC(size, d)**

MySQL Date and Time Data Types

- **DATE**
- **DATETIME(fsp)**
- **YEAR**
- **TIME(fsp)**