MODULE III:
Basic SQL reports and commands – Datatypes and notations – String functions – Data functions – Unions – Joins – DDL – DML – DLL.

SQL Data Types

Data types are used to represent the nature of the data that can be stored in the database table. For example, in a particular column of a table, if we want to store a string type of data then we will have to declare a string data type of this column.

Data types mainly classified into three categories for every database.

- String Data types
- Numeric Data types
- Date and time Data types

Data Types

Oracle String data types

CHAR(size)	It is used to store character data within the predefined length. It can be stored up to 2000 bytes.
NCHAR(size)	It is used to store national character data within the predefined length. It can be stored up to 2000 bytes.
VARCHAR2(size)	It is used to store variable string data within the predefined length. It can be stored up to 4000 byte.
VARCHAR(SIZE)	It is the same as VARCHAR2(size). You can also use VARCHAR(size), but it is suggested to use VARCHAR2(size)
NUMBER(p, s)	It contains precision p and scale s. The precision p can range from 1 to 38, and the scale s can range from -84 to 127.
FLOAT(p)	It is a subtype of the NUMBER data type. The precision p can range from 1 to 126.

DATE	It is used to store a valid date-time format with a fixed length. Its range varies from January 1, 4712 BC to December 31, 9999 AD.
BLOB	It is used to specify unstructured binary data. Its range goes up to 232-1 bytes or 4 GB.

SQL Operator Symbols	**Operators**
**	Exponentiation operator
+, -	Identity operator, Negation operator
*, /	Multiplication operator, Division operator
+, -, ||	Addition (plus) operator, subtraction (minus) operator, String Concatenation operator
=, !=, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN	Comparison Operators
NOT	Logical negation operator
&& or AND	Conjunction operator
OR	Inclusion operator

SQL ALL Operator

The ALL operator in SQL compares the specified value to all the values of a column from the sub-query in the SQL database.

This operator is always used with the following statement:

1. SELECT,
2. HAVING, and
3. WHERE.

SELECT Emp_Id, Emp_Name FROM Employee_details WHERE Emp_Salary > ALL (SELECT Emp_Salary FROM Employee_details WHERE Emp_City = Jaipur)

SQL AND Operator

The **AND operator** in SQL would show the record from the database table if all the conditions separated by the AND operator evaluated to True. It is also known as the conjunctive operator and is used with the WHERE clause.

SELECT Emp_Id, Emp_Name FROM Employee_details WHERE Emp_Salary > ALL (SELECT Emp_Salary FROM Employee_details WHERE Emp_City = Jaipur)

SQL OR Operator

The OR operator in SQL shows the record from the table if any of the conditions separated by the OR operator evaluates to True. It is also known as the conjunctive operator and is used with the WHERE clause.

SELECT * FROM Employee_details WHERE Emp_Salary = 25000 OR Emp_City = 'Delhi';

SQL BETWEEN Operator

The **BETWEEN operator** in SQL shows the record within the range mentioned in the SQL query

SELECT * FROM Employee_details WHERE Emp_Salary BETWEEN 30000 AND 45000;

SQL IN Operator

The **IN operator** in SQL allows database users to specify two or more values in a WHERE clause. This logical operator minimizes the requirement of multiple OR conditions.

SELECT * FROM Employee_details WHERE Emp_Id NOT IN (202,205);

SQL NOT Operator

The **NOT operator** in SQL shows the record from the table if the condition evaluates to false. It is always used with the WHERE clause.

Oracle String Function

The Oracle String functions manipulate the character strings more effectively.

The following table indicates each of the functions with a brief description:

Functions	**Description**
ASCII()	The ASCII() function returns the numeric value of given character.
CHR()	The Oracle CHR() function returns all the characters of the given ASCII code.
CONCAT()	The Oracle CONCAT() function returns a string by concatenating all the arguments.
INSTR4()	The Oracle INSTR4() function returns the location of the substring using UCS4 code points from the given string.
LENGTH()	The Oracle LENGTH() function returns the size of the given string.
LOWER()	The Oracle LOWER() function returns the given string into the lower case.
LPAD()	The LPAD() returns the left padded to the given length.
LTRIM()	The LTRIM() function returns the string by removing given characters from the left side.
NCHR()	The NCHR() function returns the character which is based on the number_code in the national character set.
REGEXP_INSTR()	The Oracle REGEXP_INSTR() function is used for pattern matching in a string.
REGEXP_SUBSTR()	The REGEXP_SUBSTR()function is used for pattern matching in substring from a string.
REPLACE()	The Oracle REPLACE() function is used to replace the sequence of character with another character in the given string.
RTRIM()	The Oracle RTRIM() function returns the string by removing given characters from the right side.
SUBSTR()	The Oracle SUBSTR() function is used to extract the substring from the given string.
TRIM()	The Oracle TRIM() function is used to remove the specified character from the head of the string or tail of the string.
UPPER()	The Upper() function is used to convert the given string into uppercase.
VSIZE()	The Oracle VSIZE() function returns the number of bytes of the given expression.

DATE Function in SQL:

The SYSDATE function in Oracle is used to return the current date and time set for the operating system on which the database resides. The data type of the returned value is DATE, and the format returned depends on the value of the NLS_DATE_FORMAT initialization parameter. The function requires no arguments. In distributed SQL statements, this function returns the date and time set for the operating system of your local database. You cannot use this function in the condition of a CHECK constraint.

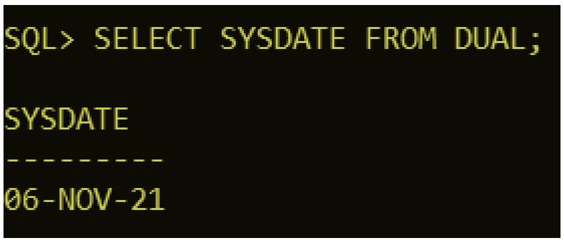
Example:

SELECT SYSDATE FROM DUAL;

Example:

SELECT SYSDATE+10 FROM DUAL;

Output:

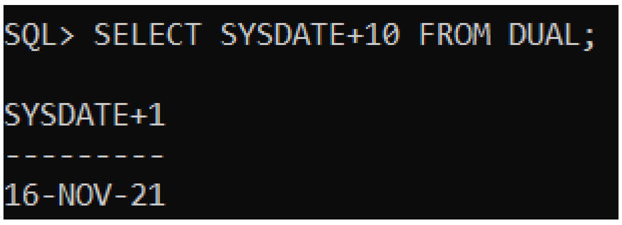


```
SQL> SELECT SYSDATE FROM DUAL;

SYSDATE
---------
06-NOV-21
```

Example:

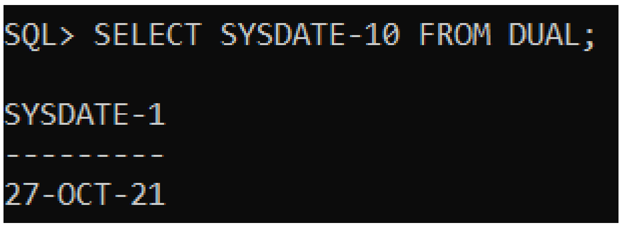
SELECT SYSDATE-10 FROM DUAL;

Output:



```
SQL> SELECT SYSDATE+10 FROM DUAL;

SYSDATE+1
---------
16-NOV-21
```

SYSDATE Date Function in Oracle



```
SQL> SELECT SYSDATE-10 FROM DUAL;

SYSDATE-1
---------
27-OCT-21
```

What is the Union clause?

MySQL Union clause allows us to combine two or more relations using multiple SELECT queries into a single result set. By default, it has a feature to remove the duplicate rows from the result set.

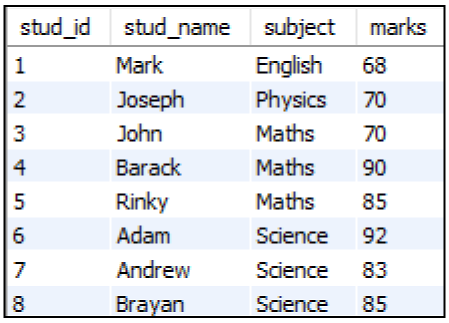
Union clause in MySQL must follow the rules given below:

- The order and number of the columns must be the same in all tables.
- The data type must be compatible with the corresponding positions of each select query.
- The column name in the SELECT queries should be in the same order.

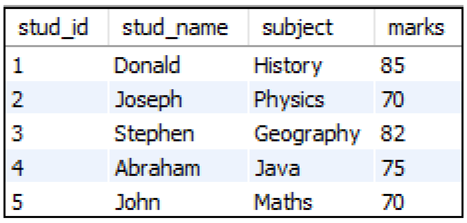
```
SELECT column_name(s) FROM table_name1
UNION
SELECT column_name(s) FROM table_name2;
```

Union Example

Suppose our database has the following tables: **"Student1"** and **"Student2"** that contains the data below:



stud_id	stud_name	subject	marks
1	Mark	English	68
2	Joseph	Physics	70
3	John	Maths	70
4	Barack	Maths	90
5	Rinky	Maths	85
6	Adam	Science	92
7	Andrew	Science	83
8	Brayan	Science	85

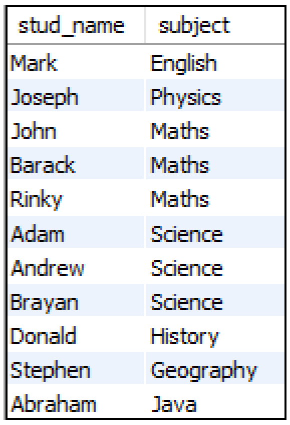


stud_id	stud_name	subject	marks
1	Donald	History	85
2	Joseph	Physics	70
3	Stephen	Geography	82
4	Abraham	Java	75
5	John	Maths	70

The following statement produces output that contains all student names and subjects by combining both tables.

```
SELECT stud_name, subject FROM student1    UNION    SELECT stud_name, subject
FROM student2;
```

After the successful execution, we will get the output that contains all unique student names and subjects:



stud_name	subject
Mark	English
Joseph	Physics
John	Maths
Barack	Maths
Rinky	Maths
Adam	Science
Andrew	Science
Brayan	Science
Donald	History
Stephen	Geography
Abraham	Java

What is the Join clause?

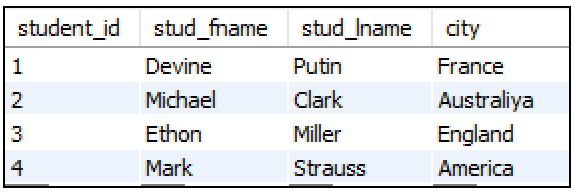
Join in MySQL is used with SELECT statement to **retrieve data** from multiple tables. It is performed whenever we need to fetch records from more than one tables. It returns only those records from the tables that match the specified conditions.

Syntax

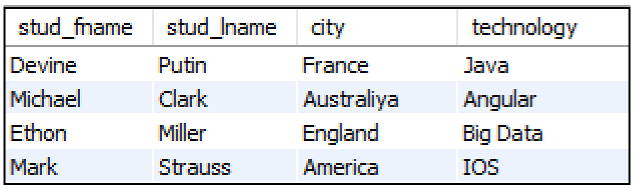
```
SELECT column_name(s) FROM table_name1

JOIN table_name2  ON conditions;
```

Join Example

Suppose our database has the following tables: **"Students"** and **"Technologies"** that contains the data below:



student_id	stud_fname	stud_lname	city
1	Devine	Putin	France
2	Michael	Clark	Australiya
3	Ethon	Miller	England
4	Mark	Strauss	America

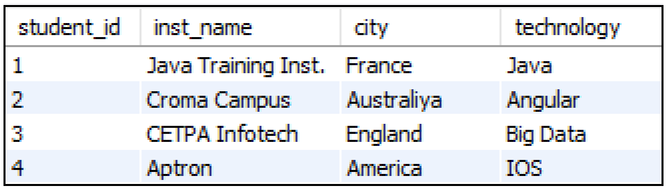


stud_fname	stud_lname	city	technology
Devine	Putin	France	Java
Michael	Clark	Australiya	Angular
Ethon	Miller	England	Big Data
Mark	Strauss	America	IOS

We can fetch records from both tables using the following query:

```
SELECT students.stud_fname, students.stud_lname, students.city, technologies.techn
ology
FROM students    JOIN technologies    ON students.student_id = technologies.tech
_id;
```

We will get the following output:



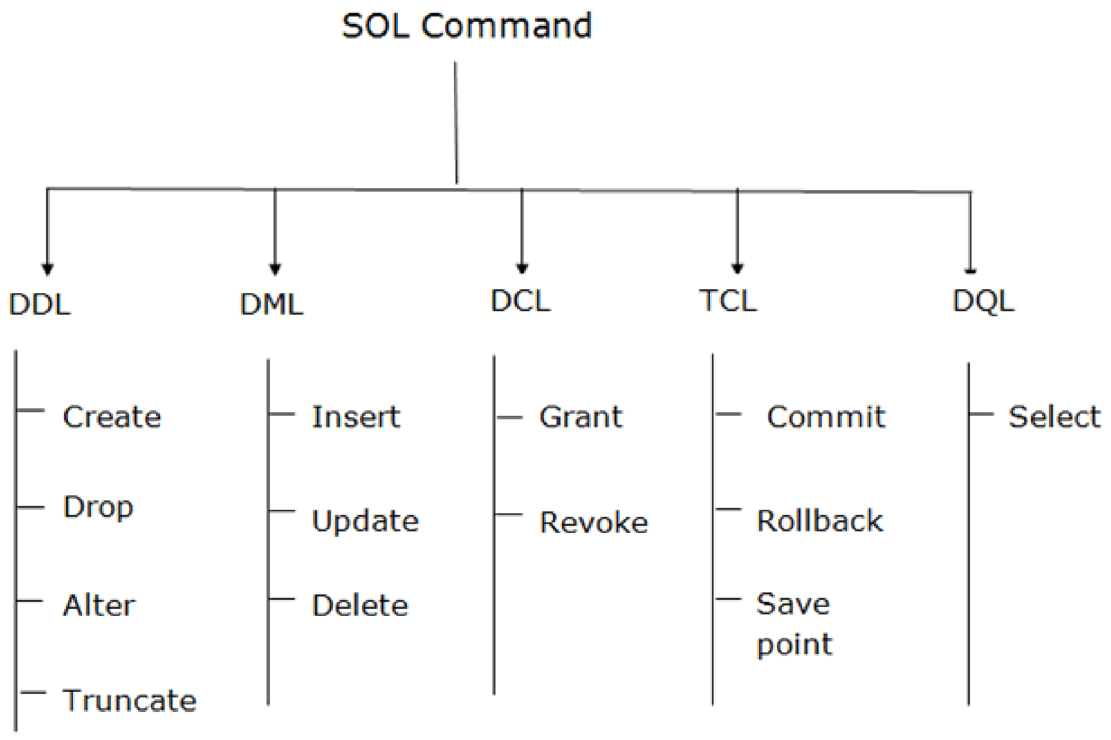
student_id	inst_name	city	technology
1	Java Training Inst.	France	Java
2	Croma Campus	Australiya	Angular
3	CETPA Infotech	England	Big Data
4	Aptron	America	IOS

SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



1. Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

a. CREATE It is used to create a new table in the database.

Example:

```
CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB
DATE);
```

b. DROP: It is used to delete both the structure and record stored in the table.

Syntax

```
DROP TABLE table_name;
```

Example

```
DROP TABLE EMPLOYEE;
```

c. ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

Syntax:

To add a new column in the table

```
ALTER TABLE table_name ADD column_name COLUMN-definition;
```

To modify existing column in the table

XAMPLE

```
ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));
```

d. TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.

Syntax:

TRUNCATE TABLE table_name;

Example:

TRUNCATE TABLE EMPLOYEE;

2. Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

a. 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);
```

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

For example:

```
INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");
```

b. 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 User_Name = 'Sonoo'
WHERE Student_Id = '3'
```

c. DELETE: It is used to remove one or more row from a table.

Syntax:

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

For example:

```
DELETE FROM javatpoint
WHERE Author="Sonoo";
```

3. Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- Grant
- Revoke

a. Grant: It is used to give user access privileges to a database.

Example

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

b. Revoke: It is used to take back permissions from the user.

Example

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

4. Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the database.

Syntax:

```
COMMIT;
```

Example:

```
DELETE FROM CUSTOMERS
WHERE AGE = 25;
COMMIT;
```

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

```
ROLLBACK;
```

Example:

```
DELETE FROM CUSTOMERS
WHERE AGE = 25;
ROLLBACK;
```

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

```
SAVEPOINT SAVEPOINT_NAME;
```

5. Data Query Language

DQL is used to fetch the data from the database.

It uses only one command:

- SELECT

a. SELECT: This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

Syntax:

```
SELECT expressions
FROM TABLES
WHERE conditions;
```

For example:

```
SELECT emp_name
FROM employee
WHERE age > 20;
```