UNIT-III

SQL & Functions

* SQL Basics:-

SQL:- SQL is Standard for "Structured Query language". It is a language used for query data present in relational database. (or)

SQL is Structured Query language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

→ SQL is the Standard language for Relational Database System.

→ All the RDBMS like MySQL, MS access, Oracle, Sybase, Informix, Postgres and SQL server.

→ SQL is a database language, it is used for database creation, deletion, fetching rows, and modifying rows.

→ SQL is based on relational algebra and tuple relation calculus.

History of SQL:-

1970 - Dr. Edgar F. Ted codd of IBM is known as the father of relational databases. He described a relational model for databases.

1974:- Structured Query language appeared.

1978: IBM worked to develop codd's ideas and released a product named System/R.

1986:- IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Sl/w which later come oracle.

SQL Data Types:-

The commonly used datatypes in SQL are,

1) char(size):- It is used to store character string data [Alpha numeric]. It is a fixed length character string. The size in brackets can determine the maximum number of characters that can hold. The char datatype can hold maximum of 256 characters (i.e; -128 to +128)

2) varchar (size) (or) varchar2 (size):- It is also used to store character string data i.e; alphanumeric data. It is a variable length character string, varchar stands for "character varying" or "varied character". varchar datatype can hold maximum of 2000 characters/bytes. The maximum length of varchar2 is 4000 bytes.

3) Int (or) Integer:- Integer datatype stores number data or numeric data. It does not contain any size in to it. The default size of INT datatype is 5.

4) Number(P,S):- It is used to store numbers (both fixed point and floating point). It stores zero, positive, Negative fixed and floating point numbers. Here p means precision and s means scale. The precision determine the maximum length of data. i.e; total number of digits in integer part and decimal part. The scale determines the number of positions in the floating part. If the scale is omitted it takes the default value as zero. The no. data typy can store Maximum of 38 digits precision.

5) Date:- This data type is used to store or represent Date and Time. The oracle format of Date is DD_MON-YY. while Entering date, we must keep value in single inverted comma(' ').

6) Long:- It is used to store long text character strings of size up to 2GB.

7) Raw/ long Raw:- It is used to store binary data or byte strings. i.e; digitized picture (or) image. The maximum length of Raw is 2000 bytes. And the maximum length of long Raw 2000 GB.

8) Real(or) Float(size):- It is used to store floating point number with user specified precision (decimal point) of atleast n digits.

Rules for Constructing SQL Statements:

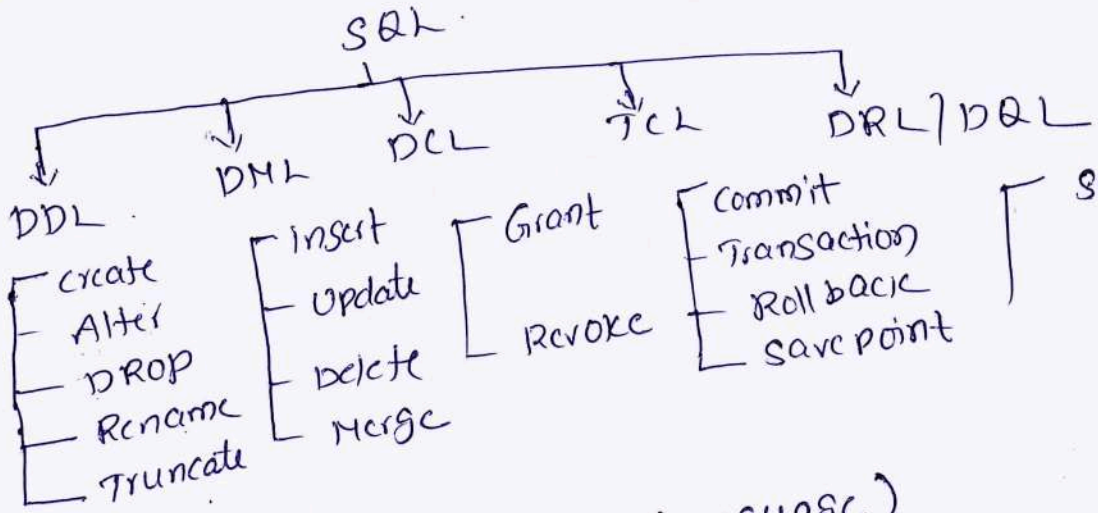
→ SQL statement must start with a verb.
→ The Attribute and Datatype must be separated by space.
→ The Attributes must be separated by comma(,).
→ The SQL statement must be terminated with a semicolon(;).
→ The SQL Command must be typed at prompt only i.e; SQL>

Importance of SQL:-

→ SQL is the primary database language used in thousands of applications.
→ SQL is the main, or almost the only language, used in all most all relational database management systems.
→ All market leading database management systems such as Oracle, MS SQL Server and DB2 support SQL.
→ Most popular free database management systems such as MySQL, SQLLite and postgreSQL also support SQL.
→ Most of the modern database applications for domains such as banking and telecom are written using SQL.
→ SQL can be used from various other language such as C, C++, C#, Java, Php, Python etc; for writing database applications.

* SQL statements / commands:-

SQL statements can be divided into 5 types.



1) DDL:- (Data Definition Language)

It is used for defining and modifying the data & its structure.
It is used to build & modify the structure of your tables and other objects in database.

(i) Create:- It is used to creating object in the database and creates a new table.

Syntax: SQL> Creati table <tablename> (column-name1 datatype(size), column-name2 datatype(size), ... column-namen datatype(size));

Ex:-(1) Employee table.

Create table Employee (Empid int, ename char(10), age int, city char(25), phone no varchar(20));

We can show our table that command is. Describe command (i.e; desc)

```
desc tablename;
```

desc employee;

O/p:- employee

empid	ename	age	city	phoneNO

2) Alter:-

It allows to alter & modify the structure of database and add additional columns.
Drop existing columns & even change datatype of columns.

(i) Alter --- Add:-

Syn:- Alter table <tablename> Add (new column name datatype(size));

Ex:- Alter table Employee Add (address varchar(20));

(ii) Alter --- modify:-

Syn:- Alter table <tablename> modify (column-name newdatatype(newsize));

Ex:- Alter table Employee modify (phoneno number(10));

(iii) Alter --- Rename:-

Syn:- Alter table <tablename> rename column old column name to new column name;

Ex:- Alter table Employee rename column ename to empname;

iv) Alter -- DROP:-

Syn:- Alter table <tablename> DROP column <columnname>;

Ex:- Alter table employee DROP column city;

(3) Rename:- It is used to rename an object and db table.

Syn:- Rename old tablename to new table name;

Ex:- Rename Employee to Emp;

(4) DROP:- It allows to remove Entire database objects from the database.

Syn:- DROP table <table-name>;

Ex:- DROP table Emp;

(5) Truncate:- It is used to delete all the rows from the table permanently.

Syn:- Truncate table <table-name>;

Ex: Truncate table Emp;

*Structure:- After creating the table we saw

desc tablename;

Inserting in the table values & we can see the table

Select * from tablename;

Ex:-(2) Student table;

(i) Create:- create table Student (Rollno number(10), name char(8), branch char(4));

(ii) Alter -- add:-

Alter table Student add (address varchar2(10));

(iii) Alter -- modify:-

Alter table Student modify (rollno varchar2(8));

(iv) Alter --- rename:-

Alter table Student rename column name to Stu_name;

v) Alter -- Drop:-

Alter table Student drop column branch;

vi) Rename:- Rename Student to std;

vii) Drop:- Drop table std;

viii) Truncate:- Truncate table std;

* DML:- (Data Manipulation Language):-

The SQL Commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL Statements. DML performs Read-only queries of data.

(i) insert:- It is used for insert data into a table, you can add / or more records into a single table.

Syn:- insert into <tablename> (column-name1 datatype(size), column-name2 datatype(size), --- column-namen datatype(size)) values ('value1', 'value2' ---- 'valuen');

(or)

→ Single row insertion:-

Syn:- insert into <table name> (col name1, col name2 -- colnamen) values ('value1', 'value2' --- 'valuen');

Ex:- insert into emp (eid, ename, city) values (1, 'ABC', 'PUNE');

1 row inserted.

→ Multiple rows insertion:-

Syn:- insert into <tablename> values (&'colname1', &'colname2' &'colnamen');

Ex:- insert into emp values (&eid, '&empname', '&city');

(ii) Update:- Modify the record present in existing table

syn:- Update <table name> set. column name = value where condition;

Ex:- Update emp set salary = 20,000 Where ename = 'ABC';

(iii) Delete:- It is used to delete some all records from the existing table.

Syn:- (i) Delete from tablename where condition;

Ex:- delete from Emp where empid = '001';

Syn (ii):- Delete * from <tablename>;

Ex:- Delete emp;

* Differences between Truncate, Drop, Delete:-

Truncate	Drop	Delete
1. DDL Command.	1. DDL command	1. DML command
2. Truncate removes all rows from a table	2. Drop removes the Entire table	2. removes matching rows.
3. The operation cannot be rolled back	3. can not be rolled back	3. can be rolled back
4. Permanently delete data from database	4. completely destroy table from database	4. temporarily delete data from database

* DCL (Data control language):-

Data control language (DCL) is used to control privileges in database. To perform any operation in the database, such as for creating tables, sequences or views, a user needs privileges.

privileges are of two types:

System:- This includes permissions for creating session, table etc. and all types of other system privileges.

Object:- This includes permissions for any command or query to perform any operation on the database tables.

DCL have two commands:-

(i) Grant:- Used to provide any user access privileges or other privileges for the database.

→ Allow a user to create table:-

To allow a user to create tables in the database.

```
Grant create table to username;
```

→ Grant all privilege to a user:- systemdba is a set of privilises which has all the permissions in it. If we want to provide all the privileges to any user. We can simply grant them the sysdba permission.

```
Grant Sysdba to username;
```

→ Grant permission to create any table:-

```
Grant create any table to username;
```

→ Grant permission to drop any table:-

```
Grant drop any table to username;
```

(ii) Revoke: Used to take back permissions from any user.

```
Revoke create table from username;
```

* Defining Constraints:-

(1) * Primary Key:- Primary key constraint uniquely identifies each record in a database. A primary key must contain unique value and it must not contain null value. Usually primary key is used to index the data inside the table.

→ Using Primary key constraint at Table Level:-

Ex: Create table student (s_id int primary key, name varchar(10) NOT NULL, Age int);

primary key

s_id	name	Age
101	A	18
102	B	19
103	C	20

→ Using Primary Key constraint at column level:-

Ex:- create table student (rollno number(10), name char(10), branch char(3), primary key(rollno));

By using alter table student:-

Alter table student Add primary key (S_id);

(2) * Foreign Key:- Foreign key is used to relate two tables. A foreign key provides a way of enforcing referential integrity with in SQL Server. Foreign key ensures values in one table must be present in another table.

Rules:- → NULL is allowed in foreign key.
→ The table being referenced is called the parent table.
→ The table with the foreign key is called child table.
→ The foreign key in child table references the primary key in the parent table.
→ This parent-child relationship enforces the rule which is known as "Referential integrity".

Ex:

Customer_Detail Table:

C_id	cust_name	address
101	Adam	Noida
102	Alex	Delhi
103	Stuart	Rohtak

Order_Detail Table

order_id	order_name	c_id
10	order1	101
11	order2	103
12	order 3	102

→ Using foreign key constraint at table level:-

```
Create table Order_Detail ( order_id int primary key,
                            Order_name char(10) NOT NULL,
                            C_id int foreign key References
                            customer_Detail (c_id));
```

→ Using foreign key constraint at column level:-

```
Alter table Order_Detail add foreign key (c_id) References
customer_Detail (c_id);
```

3) *check:-

Check constraint is used to restrict the value of a column between a range. It performs check on the values, before storing them into the database. Its like condition checking before saving data into a column.

→ Using check constraint at Table level:-

```
create table student ( s_id int NOT NULL check (S_id >0),
                       Name varchar(10) NOT NULL,
                       Age int );
```

→ Using check constraint at column level:

```
alter table Student Add check (S_id >0);
```

* NOT NULL:- NOT NULL constraint makes sure that a column does not hold NULL value. when we don't provide value for a particular column while inserting a record into a table, it takes NULL value by default. By specifying NULL constraint, we can be sure that a particular columns cannot have NULL values.

Ex:-

```
Create table Student (RollNo int NOT NULL, Stu_name
varchar (35) NOT NULL, Stu_age int NOT NULL,
Stu_Address varchar2(20), Primary Key (Roll_NO));
```

```
desc student;
```

Name	NULL	Type
Roll_NO	NOT NULL	number(38)
Stu_Name	NOT NULL	varchar(35)
Stu_age	NOT NULL	number(38)
Stu_Address		varchar2(20)

* UNIQUE:- UNIQUE constraint enforces a column or set of columns to have unique values. If a column has a unique constraint, it means that particular column can not have duplicate values in a table

Ex:- Create table Student (Roll_no int NOT NULL,
Stu_name varchar(25) UNIQUE, Stu_age int NOT NULL,
Stu_address varchar(30) UNIQUE);

* Default:- The default constraint provides a default value to a column when there is no value provided while inserting a record into a table.

Ex:- Create table Student (Roll_no int NOT NULL, Stu_name varchar(35) NOT NULL, Stu_age int NOT NULL, Exam_fee int default 1000, Stu_address varchar(35), primary key (Roll_no));

* Operators:- An operator is a reserved word or a character used primarily in an SQL statement's where clause to perform operations, such as comparisons and arithmetic operations. These operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement.

→ Arithmetic operators → +, -, *, /, %

→ comparison operators → =, !=, <, >, <>, >=, <=, !<, !>

→ logical operators

→ operators used to negate conditions

→ SQL Arithmetic operators

SQL logical operators:-

1) ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE, NOT, OR, ISNULL, UNIQUE

* IN operator:- The IN operator is used to compare a value to a list of literal values that have been specified.

The IN operator allows you to specify multiple values in a where clause.

Syntax:

```
Select column_name
from table_name
Where column_name IN (value1, value2,
                      ...valuen);
  (Or)
Select column_name
from table_name
Where column_name IN (Select statement);
```

Ex: 1) Select * from customers
Where country IN ('Germany', 'France', 'UK');

2) Select * from customers
where county NOT IN ('Germany', 'France', 'UK');

3) select * from customers
where country IN (select county from suppliers);

* SQL Functions :-

SQL functions are simply subprograms, which are commonly used and re-used throughout SQL database applications for processing or manipulating data.

SQL provides many built-in functions to perform operations on data. These functions are useful while performing mathematical calculations, string concatenations, sub-strings

There are two types of SQL functions, aggregate functions, and Scalar (non-aggregate) functions.

* Aggregate functions :- This function can produce a single value for an entire group or table. They operate on set of rows and return results based on groups of rows. Aggregate functions operate on many records and produce a summary. Works with GroupBy
  - SQL Count() :- Returns the number of rows
  - Sum() :- Returns the sum
  - Avg() :- Returns the average value
  - Max() :- Returns the largest value
  - Min() :- Returns the smallest value
  - First() :- Returns the first value
  - Last() :- Returns the last value

i) Avg() function :- Average returns average value after calculating it from values in a numeric column.

Syn:- Select Avg (column_name) from table_name;

Consider Emp table:

eid	name	age	salary
401	Anu	22	9000
402	shanc	29	8000
403	Rohan	34	6000
404	scott	44	10000
405	Tiger	35	8000

Query: To find average salary.

Command: `Select avg(salary) from Emp;`

Result:

avg(salary)
8200

(ii) Count() function: count returns the number of rows present in the table either based on some condition or without condition.

Syn: `Select count(column-name) from table-name;`

Query: To count employees.

Command: (i) `Select count(name) from Emp where salary=8000;`

Result:

Count(name)
2

Command: Select count(Distinct salary) from Emp;

count(distinct salary)
4

(iii) first() function:- first function returns first value of a selected column.

Syn:- `Select first(column_name) from table_name;`

Command: Select first(salary) from Emp;

Result:

first(salary)
9000

iv) Last() function:- Last function returns the return last value of the selected column.

Syn:- `Select last(column_name) from table_name;`

Command:- select last(salary) from Emp;

last(salary)
8000

v) Max() function:- MAX function returns maximum value from selected column of the table.

Syn:- `select Max(column_name) from table_name;`

Query:- To find the Maximum salary

Command: Select Max(salary) from Emp;

Result:-

MAX(salary)
10000

vi) SQL count(*):- The count(*) function returns the no. of records in a table.

Syn:- select count(*) from table_name;

(vii) MIN() function: MIN function returns minimum value from a selected column of the table.

Syn:- `Select Min(column_name) from table-name;`

Query:- To find minimum salary.

command:- Select Min(salary) from emp;

Result:-

Min(salary)
6000

viii) Sum() function:- Sum function returns total sum of a selected columns numeric values.

Syn:- `Select sum(column_name) from table-name;`

Query:- To find sum of salaries.

command:- Select sum(salary) from emp;

Result:-

sum(salary)
41000

* Built-in-functions:-

In SQL a built-in function is a piece of programming that takes zero or more inputs and returns a value. An example of a built-in functions is ABS() which when given a value calculates the absolute (non-negative) value of the number.

* Numaic functions:- This functions are used to perform operations on numbers and return numbers.

(i) Abs():- This function is used to convert a negative value to positive value.

Syn:- abs(NO)

(i) SQL> Select abs (100-120) from dual;

O/P:- 20.

(ii) SQL> Select abs(-243.45)

O/P:- 243.45

(ii) Sqrt():- This function returns the square root of a given number.

Syn:- sqrt(NO)

Ex:- SQL> Select sqrt(81) from dual;

O/P:- 9.

(iii) Mod():- This function returns the modulus.

Syn:- mod(dividend, divisor)

Ex:- Select mod(15,2) from dual;

O/P:- 1

iv) Power():- This function returns the power of given number as per the raised number specified.

Syn:- power(number, raised number)

Ex:- SQL> Select power(4,3) from dual;

O/P:- 64

v) Round():- This function is used to round off the required number of decimals.

Syn:- round (NO, NO. of decimals)

Ex:- SQL> select round (23.6789, 2) from dual;

O/P:- 23.68.

* String functions:- This functions are used to perform an operation on input string and return an output string.

(i) Upper():- This function converts a string into upper case.

Syn:- upper ('string')

Ex:- SQL> select upper ('hallo') from dual;

O/P:- HELLO.

(ii) Lower():- This function converts a string into lower case.

Syn:- lower ('string')

Ex:- SQL> select upper ('HELLO') from dual;

O/P:- hello.

(iii) Initcap():- This function converts a string into initial caps.

Syn:- initcap ('string')

Ex:- Select initcap ('computer education') from dual;

O/P:- Computer Education.

iv) ltrim():- This function is availed to remove the unnecessary spaces or characters available to a string

Syn:- ltrim ('string', 'unnecessary char')

Ex:- SQL> select ltrim('xyxyhello', 'xy')

O/P:- hello.

v) Rtrim():- This function is used to remove the unnecessary characters or spaces available on the right side of a string.

Syn:- rtrim ('string', 'unnecessary char')

Ex:- SQL> select rtrim ('computerxyzxy', 'xy') from dual;

O/P:- computerxyz

* Date functions:-

function	Description
NOW()	Returns the current date and time
curDate()	Returns the current date
curTime()	Returns the current time
Date()	Extracts the date part of a date or date / time expression
Extract()	Returns a single part of a date/time
DateADD()	Adds a specified time interval to a date.
DateSub()	Subtracts a specified time interval from a date.
DateDiff()	Returns the number of days between two dates
Date format()	Displays date / time data in different formats

SQL Server Date functions:-

Function	Description
GetDate()	Returns the current date and time
Datepart()	Returns a single part of a date/time
DateADD()	Adds or Subtracts a specified time interval from a date
DateDiff()	Returns the time between two dates
convert()	Displays date/time data in different formats.

SQL Date Data Types:-

→ Date - format YYYY-MM-DD
→ DateTime - format: YYYY-MM-DD HH:MM:SS
→ TimeStamp - format: YYYY-MM-DD HH:MM:SS
→ Year - format YYYY or YY

Example:- Assume we have the following "Orders" table:

OrderID	ProductName	OrderDate
1	Geitost	2008-11-11
2	Camembert pierrot	2008-11-09
3	Mozzarella di Giovanni	2008-11-11
4	Mascarpone Fabioli	2008-10-29

Query:- Select * from Orders where OrderDate = '2008-11-11';

Result:-

OrderID	ProductName	OrderDate
1	Geitost	2008-11-11
3	Mozzarella di Giovanni	2008-11-11

Example 2:-

1) ADDDate (date, interval expr unit),
ADDDate (expr, days)

→ SQL> Select Date_ADD ('1998-01-02', Interval 31 Day);

Date_ADD ('1998-01-02', Interval 31 Day)

1998-02-02

1 row in set (0.00 sec)

→ SQL> Select ADDDate ('1998-01-02', Interval 31 Day);

ADDDate ('1998-01-02', Interval 31 Day)

1998-02-02

1 row in set (0.00 sec)

2) ADDTime (expr1, expr2):-

ADDTime() adds expr2 to expr1 and returns the result.
The expr1 is a time or datetime expression, while the
expr2 is a time expression.

SQL> Select ADDTime ('1997-12-31 23:59:59.999999',
'1 1:1:1.000002');

Date_ADD('1997-12-31 23:59:59.999999', 11 1:1:1.000002')

1998-01-02 01:01:01.000001

1 row in set (0.00 sec)

(iii) CURDATE(): Returns the current date as a value in 'YYYY-MM-DD' or YYYYMMDD format, depending on whether the function is used in a string or in a numeric context.

SQL> Select CURDate();

CURDATE()
1997-12-15

1 row in set (0.00 sec)

iv) CURTIME():

SQL> Select CURTIME();

CURTIME()
23:50:26

1 row in set (0.00 sec)

v) SYSDATE(): - Returns the time at which the function executes.

Syn: sql> Select Sysdate from dual;

vi) Now(): Returns the current date and time.

Ex:- Select Now();

O/P:- 2017-01-13 08:03:52

vi) CURDATE(): Returns the current date

Ex:- Select CURDATE();

O/P:- 2017-01-13

vii) CURTIME(): Returns the current time.

Ex:- Select CURTIME();

O/P:- 08:05:15

viii) DATE(): Extracts the date part of a date or date/time expression.

Ex: Test table.

Id	Name	Birth Time
4120	Pratik	1996-09-26 16:44:15.581

Select Name, Date(BirthTime) AS BirthDate from Test;

O/P:-

Name	Birth Date
prathik	1996-09-26

ix) EXTRACT(): Returns a single part of a date/time.

Syn: Extract (unit from date);

There are several units that can be considered but only some are used such as: MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, etc. And 'date' is a valid date expression.

Ex: Test table

Id	Name	Birth Time
4120	Prathik	1996-09-26 16:44:15.581

Queries:-

→ Select Name, Extract (Day from BirthTime) AS BirthDay from Test;

O/P:-

Name	BirthDay
prathik	26

→ Select Name, Extract (Year from BirthTime) AS Birthyear from Test;

O/P:-

Name	Birthyear
prathik	1996

→ Select Name, Extract (Second from BirthTime) AS Birthsecond from Test;

O/P:-

Name	Birthsecond
prathik	58

(x) Date_ADD():- Adds a specified time interval to a date

Syn:- DATE_ADD (date, INTERVAL expr type);

Ex:- Test table

Id	Name	BirthTime
4120	prathik	1996-09-26 16:44:15.58

Queries:- Select Name, Date_ADD(BirthTime, INTERVAL 1 Year) AS BirthTime_Modified from Test;

O/P:-

Name	BirthTime_Modified
prathik	1997-09-26 16:44:15.58

→ Select Name, Date_ADD (BirthTime, INTERVAL 30 DAY) As BirthDayModified from Test;

O/P:-

Name	BirthDayModified
Prathik	1996-10-26 16:44:15.58)

→ Select Name, Date_ADD (BirthTime, INTERVAL 4 HOUR) AS BirthHourModified from Test;

O/P:-

Name	BirthSecond
prathik	1996-10-26 20:44:15.581

xi) DATE_SUB():- Subtracts a specified time interval from a date.

xii) DATEDIFF():- Returns the number of days between two dates.

Syn:- DATEDIFF (date1, date2);

Ex:- Select DATE_DIFF ('2017-01-13', '2017-01-03') AS DATEDIFF;

O/P:- DATEDiff
10.

xiii) DATE_FORMAT():- Displays date/time data in different formats.

Syn:- DATE_FORMAT (date, format);

date is a valid date and format specifies the output format for the date/time. The formats that can be used are:

→ %.a - Abbreviated weekday name (Sun-Sat)
→ %.b - Abbreviated monthname (Jan-Dec)
→ %.c - Month, numeric (0-12)
→ %.D - Day of month with English suffix (0th, 1st, 2nd, 3rd)
→ %.d - Day of month, numeric (00-31)
→ %.e - Day of month, numeric (0-31)
→ %.f - Microseconds (000000-999999)
→ %.H - Hour (00-23)
→ %.h - Hour (01-12)
→ %.I - Hour (01-12)
→ %.i - Minutes, numeric (00-59)
→ %.j - Day of year (001-366)
→ %.k - Hour (0-23)
→ %.l - Hour (1-12)
→ %.M - Month name (January - December)
→ %.m - month, numeric (00-12)
→ %.p - AM or PM
→ %.r - Time, 12-hour (hh:mm:ss followed by AM or PM)
→ %.S - seconds (00-59)
→ %.s - seconds (00-59)
→ %.T - Time, 24-hour (hh:mm:ss)
→ %.U - week (00-53) where Sunday is the first day of week
→ %.y - year, numeric, two digits
→ %.Y - year, numeric, four digits.

ex:- DATE_FORMAT(NOW(), '%d %b %y')

Result:- 13 Jan 17

* Set Operations in SQL:-

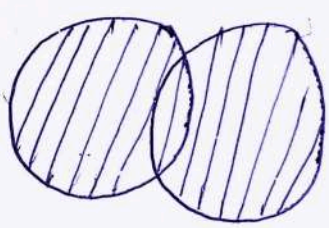
SQL supports few set operations which can be performed on the table data. These are used to get meaningful results from data stored in the table, under different special conditions.

4 different types of SET operations.

1. Union
2. Union ALL
3. INTERSECT
4. MINUS.

1. UNION Operation:- UNION is used to combine the results of two or more SELECT statements. However it will eliminate duplicate rows from its result set. In case of union, number of columns and datatype must be same in both the tables, on which UNION operation is being applied.

Syn:- Select column_name from table1 UNION Select column_name from table2;



Example of UNION:- The first table,

ID	Name
1	abhi
2	adam

The second table,

ID	Name
2	adam
3	Chester

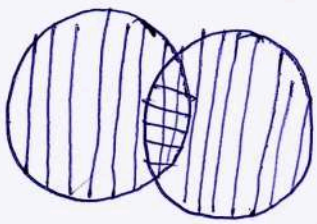
UNION SQL query will be

```
Select * from first
UNION
Select * from second;
```

Result:-

ID	Name
1	abhi
2	adam
3	chester

2) UNION ALL:- This operation is similar to union - But it also shows the duplicate rows.



Example:-

The first table,

ID	Name
1	abhi
2	adam

The second table

ID	Name
2	adam
3	chester

Union all query:-

```
Select * from First
UNION ALL
Select * from second;
```

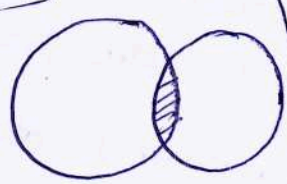
Result:-

ID	Name
1	abhi
2	adam
2	adam
3	chester

Syn:-

```
Select column_name from table1
UNION ALL
Select column_name from table2;
```

3) INTERSECT:- Intersect operation is used to combine two select statements, but it only returns the records which are common from both select statements. In case of intersect the number of columns and datatype must be same.



Syn:-

```
Select col_name from table1
INTERSECT
Select col_name from table2;
```

Ex:- The first table

ID	Name
1	abhi
2	adam

The second table

ID	Name
2	adam
3	chester

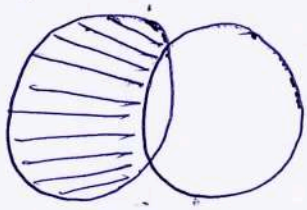
Query:-

```
Select * from first
INTERSECT
Select * from second;
```

Result:-

ID	Name
2	adam

4) MINUS :- The Minus operation combines results of two Select statements and return only those in the final result, which belongs to the first set of the result.



Ex:- The first table,

ID	Name
1	abhi
2	adam

The second table.

ID	Name
2	adam
3	chester

Query:- Select * from first
Minus
Select * from second;

Result:-

ID	Name
1	abhi

Syn:-

```
Select column_name from table1
MINUS
select column_name from table2;
```

* Differences b/w Count(*) and count?

→ Count(*) function return all the rows including duplicates.

Count(*) also considers Nulls and duplicates.

→ Count(column name) does not because count(column name) excludes null values.

Examples:- Student Table.

ID	Name	Description
1	Noon	
2	Rahi	
3		
4	Noon	Smart

→ Select count(*) from student;

count(*)
4

→ Select count(name) from student;

count(Name)
3

* Scalar (non-aggregate) functions:-

Scalar functions return a single value from an input value. Non-aggregate functions operate on each record independently.

Ucase():- Ucase function is used to convert value of string column to upper case characters.

Syn:- Select Ucase(column_name) from table-name;

Emp table:

eid	name	age	salary
401	anu	22	9000
402	Shane	29	8000
403	rohan	34	6000
404	scott	44	10000
405	Tiger	35	8000

SQL query for using Ucase is,

```
Select Ucase(name)
from Emp;
```

Result:

Ucase(name)
ANU
SHANE
ROHAN
SCOTT
TIGER

LCASE():- LCASE function is used to convert value of string columns to lowercase characters.

Syn:- Select Lcase (column_name) from table-name;

Ex:- Select Lcase(name) from emp;

Result:-

LCASE(name)
anu
shane
rohan
scott
tiger

MID() function:- MID function is used to extract substrings from column values of string type in a table.

Syn:- Select MID(column_name, start, length) from tablename;

Ex:- Select MID(name, 2, 2) from emp;

Result:-

MID(name, 2, 2)
nu
ha
oh
co
ig

Round():- Round function is used to round a numeric field to number of nearest integer. It is used on decimal point values.

Syn:- Select ROUND (column_name, decimals) from tablename;

Ex:- emp table:

eid	name	age	salary
401	any	22	9000.67
402	Shane	29	8000.98
403	rohan	34	6000.45
404	scott	44	10000
405	tiger	35	8000.01

Query:- Select Round(salary) from emp;

Result:

Round(salary)
9000
8001
6000
10000
8000

* **String functions (continuous):**

(i) **ASCII():** This function is used to find the ASCII value of a character.

Syn:- Select ascii('t');

O/P:- 116

(ii) CHAR_LENGTH(): DOES not work for SQL Server.

Use LEN() for SQL server. This function is used to find the length of a word.

Syn:- Select char_length ('Hello!');

O/P: 6

(iii) CHARACTER_LENGTH():- Doesn't work for SQL Server.

Use LEN() for SQL server. This function is used to find the length of a line.

Syn: Select character_length ('geeks for geeks');

O/P:- 15

iv) concat(): This function is used to add two words or strings

Syn:- Select 'Geeks' || ' ' || 'for Geeks' from dual;

O/P:- Geeks for Geeks

(v) concat_WS(): This function is used to add two words or strings with a symbol as concatenating symbol.

Syn:- Select Concat_ws ('_', 'geeks', 'for', 'geeks');

O/P:- geeks_for_geeks

vi) FIND_IN_SET(): This function is used to ~~display a number in the given format~~ find a symbol from a set of symbols.

Syn:. select find_in_set ('b', 'a,b,c,d,e,f');

O/P:- 2

vii) format():. This function is used to display a number in the given format.

syn:. format ("0.981", "percent");

O/P:. '98.10%'

viii) INSERT():. This function is used to insert the data into a database.

syn:. INSERT INTO database (geek_id, geek_name).
values (5000, 'abc');

O/P:. Successfully updated.

ix) LEFT():. This function is used to select a substring from the left of given size or characters.

syn:. Select Left ('geeksforgeeks.org', 5);

O/P:. geeks

x) LENGTH():- This function is used to find the length of a word.

syn: LENGTH ('GeeksforGeeks');

O/P:- 13.

xi) Repeat():. This function is used to write the given string again and again till the number of times mentioned

Syn:. Select Repeat ('geeks', 2);

O/P: geeksgeeks

xii) Replace(): This function is used to cut the given string by removing the given substring -

Syn: Replace ('123geeks123', '123');

O/P: geeks -

Syn: Replace (Text, search_string, replacement string)

Ex: Select Replace ('Data Management', 'Data', 'Database') from dual;

O/P: Database Management

xiii) Reverse(): This function is used to reverse a string.

Syn:- Select Reverse ('geeksforgeeks.org');

O/P: 'gro.skeegrofskeeg'

xiv) STRCMP(): This function is used to compare 2 strings.

→ If string1 and string2 are the same, the STRCMP function will return 0.

→ If string1 is smaller than string2, the STRCMP function will return -1.

→ If string1 is larger than string2, the STRCMP function will return 1.

Syn: Select STRCMP ('google.com', 'geeksforgeeks.com');

O/P:- -1

xv): SUBSTR(): This function is used to find a substring from the a string from the given position.

Syn:- SUBSTR ('geeksforgeeks', 1, 5);

O/P: 'geeks'

* Numeric functions (Continuous):

→ ACOS():- It returns the cosine of a number.

Syn:- Select ACOS(0.25);

O/P:- 1.31811607165 2818

→ ASIN():- It returns the arc sine of a number

Syn:- Select ASIN(0.25);

O/P:- 0.25268025514207865

→ ATAN():- It returns the arc tangent of a number.

Syn:- Select ATAN(2.5);

O/P:- 1.190289949682 5317

→ ceil():- It returns the smallest integer value that is greater than or equal to a number

Syn:- Select ceil(25.75);

O/P:- 26.

→ COS():- It returns the cosine of a number

Syn:- Select cos(30);

O/P:- 0.154251449887 58405

→ COT():- It returns the cotangent of a number

syn:- Select cot(6);

O/P:- -3.436353004180128

→ DIV():- It is used for integer division.

syn:- Select 10 DIV 5;

O/P:- 2

→ Floor():- It returns the largest integer value that is less than or equal to a number.

Syn:. Select floor(25.75);

O/P:- 25

→ Greatest():. It returns the greatest value in a list of expressions.

Syn:. Select Greatest(30,2,36,81,125);

O/P:. 125

→ least():. It returns the smallest value in a list of expressions

Syn:. Select least(30,2,36,81,125);

O/P:- 2

→ LN():. It returns the natural logarithm of a number.

Syn:. Select LN(2);

O/P:- 0.693147180559453

→ log10():. It returns the base-10 logarithm of a number.

Syn:. Select log(2);

O/P:. 0.693147180559453

→ log2():. It returns the base-2 logarithm of a number

Syn:. Select log2(6);

O/P:. 2.584962500721156

→ PI():. It returns the value of PI displayed with 6 decimal places.

Syn:. Select PI();

O/P:. 3.141593

→ Rand():- It returns a random number.

Syn:. Select Rand();

O/P:- 0.33623238684258644.