WELCOME TO SQL

Basics of SQL

1) What is SQL?

- **SQL** stands for Structured Query Language.
- **SQL** is a standard language designed for managing data in relational database management system.
- **SQL** is a programming language specifically designed for storing, retrieving, managing or manipulating the data inside a relational database management system (RDBMS).

2) What is data in SQL?

- The information which we are storing inside the database is known as data in SQL.

3) What is Database?

- It is a place where we can store all the data permanently.

4) What is DBMS?

- DBMS Stands for database management System.
- It is a software which is used to store and organize the data in the form of a table.

5) What is table in SQL?

- An entity is known as table in SQL.
- Entity means collecting the data about any objects and stored in the form of tables.
- A table is a combination of multiple rows and multiple columns.
- For each and every data, we are using cells.
- Rows is known as tuple .
- Columns is known as field or attribute in sql.

6) What are the types of DBMS in sql?

❑ There are four types:

• HDBMS (Hierarchical DBMS)

• NDBMS (Network DBMS)

• ODBMS (Object DBMS)

• RDBMS (Relational DBMS)

7) HDBMS : It is a type of DBMS which helps to store and organize the data in the form of tree structure.

- Eg: Company details:

    CEO'S → MD → Branches → Branch Manager → Employees

8) **NDBMS :** It is a type of DBMS which helps to store and organize the data in the form of network.

- Eg: Google Map

9) ODBMS : It is a type of DBMS which helps to store and organize the data in the form of object.

- Eg: C++, Java

10) RDBMS : It is a type of DBMS which helps to store and organize the data in the form of table.

- Most popular database in market because of it's easy usage.

- E.F.Codd(Edgar Frank Codd) was the inventor of RDBMS Concept.

Chapter 1.

Constraints

1) What is Constraints?

- Basically it is condition or rules which we are applying to create one table record with data correctness.

- **Constraints** are used to limit the type of data that can go into a table.

2) Types of constraints in SQL?

a. Null
b. Not Null
c. Unique
d. Primary key
e. Foreign key
f. Check

a. Null :-

- Null means empty spaces.
- Based on one column if we are not going to pass some data, then this will be consider as null record in SQL.

b. Not Null :-

- It will always check whether the value is present or not inside one column.
- It will not consider any null records.

c. Unique :-

- It will check duplicate values or repeated data is present or not inside one column.

- If it will find any data repetitions inside one column, we can't consider this column as unique.

- But it will take multiple null constraints.

d. Primary key :-

- It is the combination of Not Null and Unique.
- It is used to identify the data uniquely inside one table.
- Only one primary key can be create based on one table record, creating more than one primary key will not be allowed by the server.
- Creating primary key is not mandatory but it is hightly recommended.

e. Foreign key :-

- It is also known as referential integrity constraints.
- We are using foreign key to create the relationship between one table record to another table record.
- We can create more than one foreign key.
- It will take null value and duplicate value both.
- To create foreign key, the column data should be present inside master table.

f. Check :-

- It is used to apply the condition to fetch one data inside one table record.

- It can be applied to a column in a table to limit the values that can be placed in a column.

- If you define a check constraint in a single column, it allows only certain values for this column.

Chapter 2:

Commands in SQL

1). Commands in SQL

❑ There are five commands in SQL

a) **Show :** We are using this command to check the default page sizes and line sizes inside our database.

Eg : Show pages;

Show lines;

b) Set: It is used to change the length of page size and line size inside the database.

• It is not for permanent changing.

Eg: set pages 200;

    set lines 190;

c) cl scr : The full form is clear screen.

- We are using this command to get one fresh page.

d) DESC : The full form is description.

- We are using this command to check the column records based on each and every table.

- It will display the column, constraints, Data-type and length(size) of the columns present in a table.

Eg: desc table-name;

    desc emp;

e) Ed :

- We are using this command to make change inside our database.

- We have to call ed inside the server.

- The editor will open.

- Inside the editor we are going to make change.

- After changing we have to cross/save it and the edited query will be store inside our database.

- To execute the edited query we are using /.

Chapter 3 :
Statements in SQL

1) **Statements in Sql / Subsets in Sql / Sub-languages in Sql :**

a) **DQL** [Data Query Language] → Select

b) **DDL** [Data Definition Language] → Create, Drop, Truncate, Rename, Alter

c) **DML** [Data Manipulation Language] → Insert, Update, Delete

d) **TCL** [Transition Control Language] → Commit, Savepoint, Rollback

a) DQL : It is used to read the data from the database.

b) DDL : Allows to perform various operations on the database.

c) DML : Allows to access and manipulate the data.

- It helps to insert, update, delete and retrieve data from the database.

c) TCL : Allows to control and Access to the database.

- There are 4 conditions are there to write the query based on HIREDATE Column.
- ❑ on → =
- ❑ From → >=
- ❑ Before → <
- ❑ After → >

Chapter 4 :

Distinct And Aliasing (Expression)

1) **Distinct :**

- It will remove the repetitions based on any specific column.

- And will always return the unique data based on any column.

- We can't call more than one distinct keyword inside one select statement.

Syntax :

distinct column-name;

2) Expression / Aliasing :

- If we want to rename the column name, we have to call aliasing concept.

- Whatever column-name we are going to define using aliasing the copy of the column-name will be created.

- There are 3-ways are there to rename the column-name.

➢ Using 'as' keyword between old column name to new column name.

➢ Using space (" ")between old column name to new column name.

➢ Using double quotes. (" ").

Chapter 5 :

Operators

1) **What is Operators ?**

- Operators are used to perform operation between operands.

2) **Types of Operator :**

i. Arithmetic Operator : +, -, *, /

ii. Relational Operator (Comparison)

                        : <, >, <=, >=, != (<>)

iii. Logical Operator   : And, Or, Not

iv. Special Operator   : Is, In, Like, Between

i. Arithmetic Operator :

- Arithmetic operators are used to perform arithmetic operation between the operands.
- Applied only for number type of data.

ii. Relational Operator

- Relational operators are used to perform comparison operation between the operands(column names).

iii. Logical Operator :

a. And : If we have multiple condition, based on all the condition, if any common record will be present it will return those common records as output.   If any one condition does not match, it will return no rows selected as output.

b. Or : If we have multiple condition, any one condition is satisfied, based on that condition that data will be displayed as output.

c. Not : It will return the opposite data as output.

iv. Special Operator :

a) IS : It is used to evaluate null value based on one specific column.

Syntax : WHERE column name IS Null;

b) IN : It is used to evaluate the multiple records based on any specific column.

Syntax : WHERE column –name IN (data);

c) LIKE : It is used for pattern matching.

The LIKE operator is used to search for a specified pattern in a column.

Syntax : WHERE *column name* LIKE *pattern*;

WILDCARDS IN LIKE OPERATOR

- We have two wildcards in like operator

1) "%" : it will take one or more characters as output
2) "_" : it will take only one single character as output

d) BETWEEN : It is used to get some data based on the range provided inside the parameter.

Syntax : column-name BETWEEN lowest-rage and highest-range

Chapter 6 :

Sorting Technique

❑ **<u>Sorting</u>**

- Sorting technique we are using to manually sorted the data inside one column.

- To sort the data we are using ascending (asc) / descending(desc) order.

- To call the Ascending / Descending we have to use order by statement.

- Ascending order is default inside order by statement.

- Order by clause can be applied for string type of data also.

- Inside the server all the columns are stored in the form of number starts from 1. so, if we wants to apply order by clause for any particular column, instead of passing the column-name, we can also call the number of that column. We will get the same output.

Syntax :

o **<u>Sort according to one column</u>**

   SELECT * FROM table_name

   ORDER BY column_name ASC|DESC ;

o **<u>Sort according to multiple columns</u>**

   SELECT * FROM table_name

   ORDER BY column1 ASC|DESC , column2   ASC|DESC ;

Chapter 7 :
Group Function / Grouping

❏**Group Function / Aggregate Function:**

- Group functions are also known as aggregate function.

- Group By statement are used to group the values by one or more columns.

- In SQL, Order by statement can be applied for Group functions also.

- In SQL, we are having 5 group functions.
  a. Max()   b. Min()   c. Sum()   d. Avg()
  e. Count()

a). Max() : Max function we are using to get the maximum value based on any specific column.

Syntax : max (Column-name) ;

b). Min() : Min function we are using to get the minimum value based on any specific column.

Syntax : min (Column-name);

c). Sum() : Sum function we are using to get the total value based on one specific column.

- Valid for number type of data.

Syntax : sum (Column-name);

d). Avg() : Avg function we are using to get the average value based on one specific column.

Syntax : Avg (Column-name);

e). Count() : Count function we are using to get the number of records present inside one column or one table data.

Syntax : Count (Column-name/*);

GROUP BY CLAUSE

❑Group by Clause :

- If we want to display some columns and also to display the columns based on group function, at that time we have to call Group by clause.

- Syntax :

> SELECT *column_name(s), group function*
> FROM *table_name*
> WHERE *condition*
> GROUP BY *column_name(s)*
> ORDER BY *column_name(s);*

❑ **Having Clause :**

- We are using "having" clause to apply the condition based on group functions.

- We are using "where" clause to apply the condition for non-group functions.

- Having clause we have to call after group by clause.

- Where clause we have to call before group by clause.

❑ **Syntax for Having Clause :**

SELECT *column_name(s), group function*

FROM *table_name*

WHERE *condition*

GROUP BY *column_name(s)*

HAVING *condition*

ORDER BY *column_name(s);*

Chapter : 8

Data-types

❑ **What is Data type ?**

- **A datatype** is defined as the type of data, which any column can store in SQL Server.

- While creating any table, in addition to specifying the column name, you also set the **Type of Data** it will store.

- **In SQL, we have six Data-types.**

a) Char

b) Varchar / Varchar2

c) Number (Precision , Scale)

d) Date

e) CLOB (Character Large Object)

f) BLOB (Binary Large Object)

a) Char

- It is used for fixed length data.
- It is a alpha-numeric data type.
- Alpha-Numeric means we can store number and Character both.
- It can store up to 2000 characters.

   Eg: Std-name char(15)

- If one student will come with the name length of more than 15, that data is not going to be properly stored inside the database.

- If it will be less than 15, our memory will be lost.
- As it is fixed length data type, in future we are not able to make any changes here.

b) Varchar / Varchar2 :
- Varchar is known as variable length data.
- It is also store alpha numeric data.
- We can store up to 2000 character in Varchar.
- And 4000 character in Varchar2.

  Eg: Std-name Varchar(30)

c) Number (Precision, Scale) :

- We can store only number type of data.
- <u>Precision</u> : It will store the number as well as the decimal number also.
- Total number of digits including decimal places.
- <u>Scale</u> : It will store only decimal number and the count will be start from right to left.
- Total number of decimal places.

<u>To call the number data-type, we have 3 rules</u>

1. If precision will be greater than scale
2. If precision will be less than scale
3. If precision will be same as scale

d) Date Data-type :

- Date data-type we are using to store date inside one column.

- All the date type of data will be store based on the default format- `dd-mon-yy'.

e) CLOB (Character Large Object) :

- We can store long character type of data inside character large object.

- It is not alpha-numeric. So we can't store number.

- It can store variable length character data which are too long.

- We can store the data upto 4GB.

e) BLOB (Binary Large Object) :

- We can store pdf file, audio file, video file and images.

- It is not alpha-numeric or character. So we can't store character or number.

- It can store binary data.

- We can store the data upto 4GB.

Difference between CHAR and VARCHAR

CHAR

- CHAR data-type is used to store character string of fixed length.

- If the length of the string is less than the fixed length, then our memory will be lost.

- It can use when we expect the data values in a column are of same length.

- Eg: char(10) :- store only 10 characters.

VARCHAR

- VARCHAR data-type is used to store character string of variable length.

- If the length of the string is less than the fixed length, it is stored as it is without wastage of memory.

- It can use when we expect the data values in a column are of variable length.

- Eg: varchar(10) :- it can store less than 10

Difference between CLOB and BLOB

BLOB

- Binary large object.
- This is used to store binary data.
- This stores the values in the form of binary streams.
- Using this you can stores files like images, gifs, videos and audio files.
- It can store the data upto 4GB.
- It is not alphanumeric or character.

CLOB

- Character large object.
- This is used to store long character data.
- This stores values in the form of character streams.
- Using this you can store files like text files, PDF and word documents, etc
- It can store the data upto 4GB.
- It is not alphanumeric.

Chapter : 8

STATEMENTS

DDL : <u>Create</u>

- We are using create statement to create the structure of a table.

- <u>Syntax</u> : create table table-name (
        column-name1 data-type(length),
        column-name2 data-type(length),
        ……………
        ……………
        column-name-n data-type(length)
         );

- To create any column as foreign key, we should follow this syntax.

Syntax : Column-name data-type(size) REFERENCES table-name(column-name)

Create a copy of table record :

- If we wants to create a copy of table we have to write this statement.

Syntax : create table new-table-name as select statement ;

DDL : <u>Drop</u>

- Drop is used to delete one table record from the database.

- Deleted tables present in the recycle-bin.

Syntax : Drop table table-name ;

<u>Purge</u> :

- Purge is used to permanently delete one table record from database.

- Tables can be deleted only if they are present in the recycle-bin.

Syntax : Purge table table-name ;

Flashback :

- Flashback is used to restore the table record inside our database then we have to call flashback statement.

Syntax : flashback table table-name to before drop;

- Delete & permanently delete from recycle-bin

Syntax : drop table table-name purge;

DDL : Truncate :

- Truncate statement also we are using to delete the table record from our database.

- Only table records will be deleted but the table structure (column-name, data-type and size) will be remain same.

- We are not able to restore the data.

Syntax : truncate table table-name ;

DDL : Rename :

- Rename is used to rename the table-name.

Syntax : rename old-table-name to new-table-name

DDL : Alter :

- By using alter statement, we can rename the table-name as well as column-name.

- And also we can add a column and also can drop a column.

- We can change the data-type and size also.

- <u>To Rename the table-name :</u>

Syntax : alter table table-name rename to new-table-name ;

- <u>To Rename the column-name :</u>

Syntax : alter table table-name rename column old-column-name to new-column-name ;

- <u>To add a column :</u>

Syntax : alter table table-name add column-name data-type(size) ;

- <u>To drop a column :</u>

Syntax : alter table table-name drop column column-name ;

- <u>To change the data-type and size :</u>

Syntax : alter table table-name modify(column-name data-type(size)) ;

DML : Insert

- We are using this statement to insert some value inside one table.

1. Insert column-name in to the table

Syntax : Insert into table-name(column1, column2, ………………, column-n) values (value1, value2, ………………, value-n) ;

2. Insert values in to the column

Syntax : Insert into table-name values(value1, value2, ………………, value-n) ;

DML : In data manipulation language, 3 statements are there

1. Insert

2. Update

3. Delete

- DML changes are not permanent changes for database.

- To make these changes as permanent, we are using TCL statement.

DML : Update

- It is used to change the records present in the column of the table.

- In condition, use the value which uniquely represents that record.

Syntax : update table-name set expression where condition ;

DML : delete

- With the help of delete statement, we can delete first few records or middle records or full table records based on some condition.

- But if we wants to delete the full table data using delete statement only records will be deleted but structure will be remain same.

- Syntax : delete from table-name where condition(?) ;

TCL :

- TCL statements we are using to permanently store the data inside the database.
- It is applied only for DML statements.
- DDL statements implicitly stored inside the database.
- In TCL, 3 statements are there
1. Commit
2. Rollback
3. Savepoint

TCL : commit

- It is used to store the data permanently inside the table.

Syntax : commit;

TCL : rollback

- Rollback we are using to get back the previous records from our database.

- We have to call rollback statement before calling commit statement because rollback will not work after calling commit statement.

- Syntax : rollback ;

TCL : savepoint

- Inside one table record, if we inserted new data but if we don't want to store all the new updates, then we have to call savepoint.

- Syntax : savepoint savepoint-name ;

- And inside rollback we have to call the savepoint-name.

- Syntax : rollback to savepoint-name ;

Chapter : 9

Sub- Queries

Sub-Query in SQL

- Sub-Query is a query which can be nested inside a main query.

- Sub-query is also called as a nested query.

- We are using sub-query to call multiple select statement inside one query.

- In subquery innermost query will be executed first. The result of the inner query will be work as a input for outer query.

- To write a sub-query, atleast 1 common column should be existing between the tables.

Syntax :

select * / …..

from table-name

where condition = (select * / …..

   from table-name

   OUTER QUERY   where condition) ;

INNER QUERY

Properties of Sub-Query

- Sub-query should not have order by clause.

- A sub-query should be placed in the right hand side of the comparison operator of the main query.

- It should be enclosed within the parenthesis.

- More than one sub-query can be included.

Chapter : 10
JOINS

Join :

- Join is used to call multiple table records inside one query.

- Join clause is used to combine rows from two or more tables, based on a related column between them.

Types of Joins :

- Inner join or Equi join
- Outer join :- a) Left outer join

  b) Right outer join

  c) Full outer join

- Self join
- Cartesian join or Cross join
- Natural join

Syntax-wise two types :

- Oracle syntax : if we will write one join condition using "Where" clause it will be known as Oracle Syntax.

- ANSI syntax : if we will write one join condition using "on" clause it will be known as ANSI Syntax.

Syntax :

Select column-name1, column-name2…….

from table1, table2….

Where join condition

a. Inner Join :

- It is also known as equi join.
- Equi join means with the help of equals to operator if we will write join condition, it will be consider as equi join.
- With the help of inner join we are able to get only the matching record inside both the table data.
- An inner join involves joining two tables where a common id/key exists in both.

b. outer Join :

- With the help of outer join we can call both matching records and unmatching records inside one query.

Left outer join :

- It will return matching record based on both table and un-matching record based on left side table.

- Syntax :

     select column-name

    From table-name left outer join tname

    On join condition

Right outer join :

- It will return matching record based on both table and un-matching record based on right side table.

- Syntax :

    select column-name

    From table-name right outer join t-name

    On join condition

Full outer join :

- It will return matching record with the un-matching record from both right and left side table.

- Syntax :

  select column-name

  From table-name full outer join t-name

  On join condition

c. cross join :

- It will give the data based on cross product.
- The cross join generate a output which is a combination of each row of the first table with the each row of the second table.
- Cross join also known as "cartesian join".

<u>SYNTAX</u> :

Select */ column-name from table-name

Cross join another-table-name;

c. Natural join :

- If we will call the query using natural join, the common column will come one time and it will behave like a primary key.

- Natural join is a kind of inner join only.

- But in inner join, if we have any common column inside multiple table which executing the query, the common column will be repeated multiple times.

SYNTAX :

Select */ column-name

Natural join another-table-name;

d. Self join :

- If we wants to apply join condition inside one table record, we have to call self join.
- A table joined with itself is known as self join.
- In self join, we need to temporarily rename the table name.

Syntax :

Select dummy-table1.column-name,

    dummy- table2.column-name……….

From main-table dummy-table1,

    main-table dummy-table2……..

Where join condition

Chapter : 11

SET OPERATORS

Set Operators :

- Set operators are used to combine two or more select statements and returns a single result.

- <u>2 – Types are there :</u>

  a. Union

  b. Union All

  c. Intersect

  d. minus

a. Union :

- It is used to combine the results of two or more select statements.

- Inside select statement number of column and data-type should be same.

- Repeated data will not come as output.

- Output will be displayed in alphabetical order.

b. Union all :

- It is used to combine the results of two or more select statements.

- Inside select statement number of column and data-type should be same.

- Union all is taking all the duplicate record from both select statement.

c. Intersect:

- It is used to combine two select statements but it only returns the records which are common from both select statements.

- Inside select statement number of column and data-type should be same.

b. Minus :

- It is used to combine the results of two select statements and returns in the final result which belongs to only first select statement but not belongs to second select statement.

- Number of column and data-type should be same.

Co-related Sub-Query:

- This is a special type of sub-query.

- In this sub-query, we are using join and sub-query concept both.

- Co-related sub-queries are used for row by row processing. Each sub-query is executed once for every row of outer query.

Chapter : 12

TYPES OF FUNCTIONS

a. Types of Functions :

1. Group Function

2. Character Function

3. Number Function

4. Special Function

5. Date Function

2. Character Function :

It is a function that takes one or more character values as parameters and returns either a character value or a number value.

Types of Character Function :

a. Upper()   b. lower()   c. length()   d. concat()

e. Replace()   f. substr()   g. Instr()

Dummy Table :

-if we don't have any data inside one table record, but if we wants to check , we can use dummy table.

- It is used to perform some independent operations which are not belong to to any existing tables.

- The name of the dummy table is "DUAL".

a. **Upper() :** it is used to convert all the data as uppercase.

Syntax : upper(column-name/data);

b. lower() : it is used to convert all the data to lowercase.

Syntax : lower(column-name/data);

c. length() : it is used to check the length of any particular column or data and returns the number of character in it.

Syntax : length(column-name/data);

d. concat() : ||

- If we wants to add or merge multiple column data with string, we have to call concat function.

- To concat column with string we are using "||" (double vertical symbol).

e. Replace() :

This function consists of 3 arguments.

Arg1→ represents main string

Arg2 → represents search string

Arg3 → represents replace data.

→ Inside the replace function, if we call 2 arguments only then it will remove the search string from the main string

e. substr(arg1, arg2, arg3) :

- Arg1 will be consider as main string.

- Arg2 will be consider as position string.

- Arg3 will be consider as number of data we want to display.

e. Instr(arg1, arg2, arg3, arg4) :

- The full form is Instr is In String.
- Arg1 will be consider as main string.
- Arg2 will be consider as search string.
- Arg3 will be consider as index position.
- Arg4 will be consider as number of occurences.

3. Types of Number Function :

There are 3 types :

a). Mod   b). Round   c). Trunc

a). Mod() : it is used to return the remainder value.

By using 2nd value, we are going to divide 1st value.

b) Round() :

- round() we are using to get the nearest decimal value.

- We are having 2 rules.

- If the decimal range will be 0-4, it will be return the previous integer value.

- If the decimal range will be 5-9, it will be return the next integer value.

c) Trunc ():

- It is used to eliminate the decimal value from one data and it will always return the actual integer data as output.

4) Date Function :

- There are two types :

a) SysDate   b) SysTimeStamp

a). SysDate

- Stands for System date.
- It returns both date and time. But by default – only date is displayed.
- The default format is, dd-mon-yy
- We can get the data after or before number of dates.

a). SysTimeStamp :

- Introduced from Oracle9i

- Returns date, time and timezone.

- It gives the fraction of millisecond.

- Fraction of millisecond get changes from every second to second.

5) Special Function :

- There are two types.

a) To-Char   b) NVL (Null Value Logic)

a) To-Char:

- It is used to change the default format of date.

b) NVL(arg1, arg2) :

- It substitutes a value for a null.
- It will return arg1, if arg1 will be actual data.
- It will return arg2, if arg1 will be null record.
- Means arg2 will be substituted, if arg1 will be null record.

Chapter : 13

NORMALIZATION

NORMALIZATION :

- Normalization is the process of splitting the bigger table into many small tables without changing its functionality.

Advantages :

1) It reduces redundancy.

2) Avoid anomalies problem.

Normalization is a step by step process. In each step, we have to perform some activities.

Types of Normalization :
1) 1NF → 1ST Normal form
2) 2NF → 2nd Normal form
3) 3NF → 3rd Normal form
4) BCNF → (Boyee-Codd Normal form)
5) 4NF → 4th Normal form
6) 5NF → 5th Normal form