# SQL – Structured Query Language

1. SQL is a standard language for storing, manipulating and retrieving data in database.
2. SQL keywords are not case sensitive eg . SELECT, select, Select
3. Semicolon is the standard way to separate each SQL statement in database system. Eg. Select \* from TN;

## What SQL can do?

1. SQL can insert records in database.
2. SQL can update records in database.
3. SQL can delete records from database.
4. SQL can create new database.
5. SQL can create new tables in database.
6. SQL can retrieve data from a database.

**Database** – A database in SQL server is made up of a collection of tables that stores specific set of data.

**Table**

Table contains rows and columns, where the rows are known as records and column are known as field.

|  |  |  |  |
| --- | --- | --- | --- |
| SR.no | Student name | Mobile number | City |
| 1 | ghy | 947 | pune |
| 2 | aijd | 947 | goa |
| 3 | iud | 749 | delhi |

## SQL has categorized into different types of commands.

1. DDL – Data definition language
2. DML – Data Manipulation Language
3. DQL – Data Query language
4. DCL – Data control language
5. TCL – Transaction control language
6. **DDL – Data definition language**

All DDL commands are auto-commited that means it permanently save all the changes in database.

The basic DDL command in SQL are Create, Alter , Drop , Truncate

|  |  |
| --- | --- |
| Command | What it does |
| Create table | It creates new table |
| DROP table | It deletes the entire table |
| Alter table | Modifies the existing table |
| Truncate | Deletes the data inside a table but not the table itself ( Column will remain same) |

1. **DML – Data manipulation Language**

DML command are used to modify the database.

The DML commands in SQL are Insert, Update and Delete

|  |  |
| --- | --- |
| Command | What is does |
| Insert | Add new information to the database/table |
| Update | Modifies the information currently stored in database/table |
| Delete | Delete information from the database/table |

1. **DQL – Data Query language**

DQL is used to fetch the data from database.

DQL comprises only one command SELECT

|  |  |
| --- | --- |
| Command | What it does |
| Select | It retrieves the data from the database. |

1. **DCL – Data Control Language**

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

Grant – It is used to give user permission access to a database.

Revoke – It is used to take back the permission from the user.

1. **TCL – Transaction Control Language**

TCL commands there are 2 commands Commit and Rollback

Commit – Commit command is used to save all the transaction to the database.

Rollback – Rollback command is used to undo transaction that have not already been saved to the database.

**Data types in SQL**

Character

Char(20) – fixed length

Varchar(255) – variable length

Numbers – INT , BIGINT

**Values** – The values command specified the value of an insert into statement.

Create table velocity(

studentid int,

Fname varchar(255),

Lname varchar(255),

Mockmarks int,

City Varchar(251)

);

insert into velocity(studentid,Fname,Lname,Mockmarks,City)

values(1,'shantanu','sharma',16,'Indore');

insert into velocity(studentid,Fname,Lname,Mockmarks,City)

values(2,'Sonali','Ahirwar',15,'Jabalpur');

insert into velocity(studentid,Fname,Lname,Mockmarks,City)

values(3,'Mukesh','shukla',16,'Rewa');

insert into velocity(studentid,Fname,Lname,Mockmarks,City)

values(4,'shubham','Rahate',13,'Akola');

insert into velocity(studentid,Fname,Lname,Mockmarks,City)

values(5,'Prajakta','Narvekar',17,'Ratnagiri');

insert into velocity(studentid,Fname,Lname,Mockmarks,City)

values(6,'Kirti','Gathade',18,'Chandrapur');

select \* from Velocity;

**DROP** – To delete the entire table

Syntax – Drop table TN;

Drop table Velocity;

**Alter** – Modifiy the existing table

Syntax – Alter table TN add CN datatype;

Alter table Velocity add Pincode int;

**Update** – To modify the table

Syntax – Update Velocity set pincode=value;

Update Velocity set pincode=451881;

**Delete** – To delete the particular information from the table.

Syntax – delete from TN where CN=value;

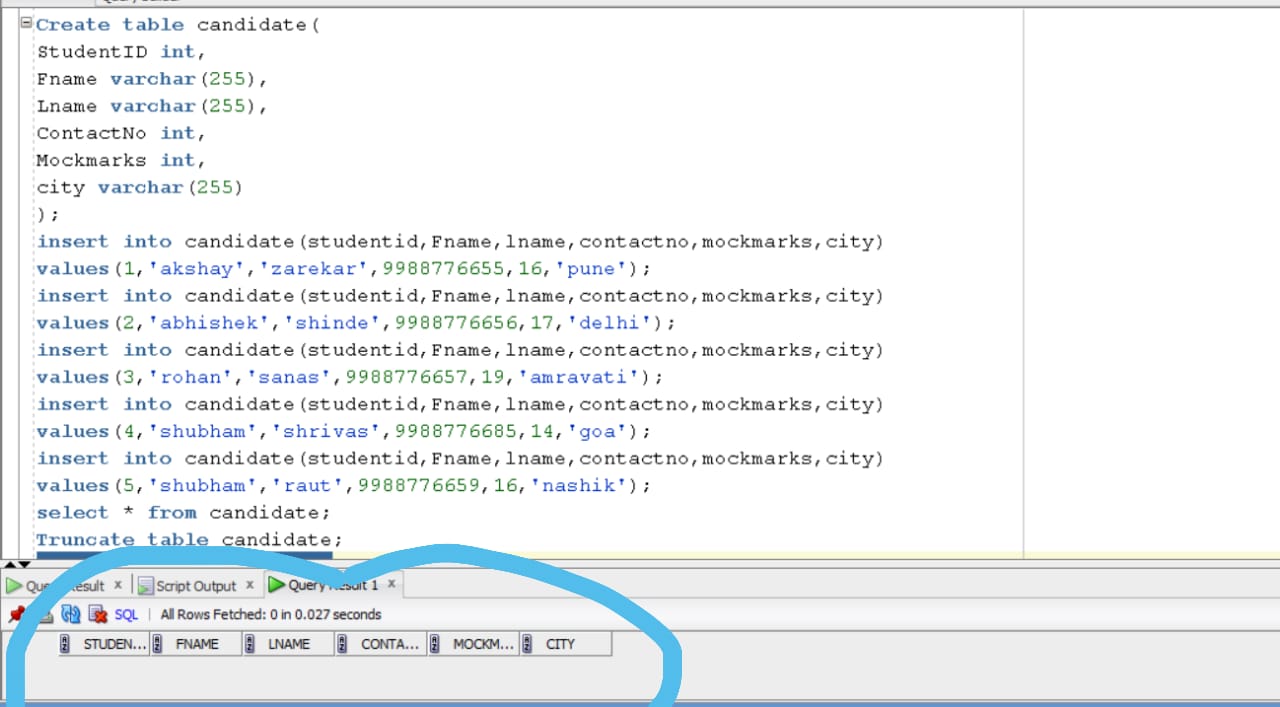
Delete from velocity where pincode=451881;

Delete updated column – syntax - alter table TN drop column CN;

**Truncate** – deletes the data inside a table

Syntax – Truncate table TN;

Truncate table velocity;



**Where Clause**

It is used to specify a condition while fetching the data from a single table.

Syntax – Select \* from TN where CN=value;

Select \* from velocity where mockmarks=16;

Logical operators – And, or, Not

1. **And** – It is used to filter the record with more than one condition and both the conditions should be true.

Syntax – Select \* from TN where CN1=value and CN2=value;

Select \* from velocity where mockmarks=16 and city='Indore'; (both the condition should be true)

1. **Or** – It displays a record on the basis of condition and it is separated by OR operator. In this one condition should be true.

Syntax- Select \* from TN where CN1=value or CN2=value;

Select \* from velocity where mockmarks=16 and city='Akola';

1. **Not** – If we don’t want a particular data in a table we are using NOT operator

Syntax – Select \* from TN where not CN=value;

Select \* from velocity where not fname=’Shubham’;

## Arithmetic operators

1. **Addition** (+) – It is used to perform addition operation on the data .

Syntax – select CN+value from TN;

Select mockmarks+2 from velocity;

1. **Substraction** (-) - It is used to perform substraction operation on the data .

Syntax – select CN-value from TN;

Select mockmarks-2 from velocity;

1. **Multiplication** (\*) - It is used to perform multiplication operation on the data .

Syntax – select CN\*value from TN;

Select mockmarks\*2 from velocity;

1. **Division** (/) - It is used to perform division operation on the data .

Syntax – select CN/value from TN;

Select mockmarks/2 from velocity;

1. **Modulus** – (mod/ %) – It is used to get reminder when one data is divided with another.

Syntax – select mod(CN,value) from TN;

Select mod(mockmarks,3) from velocity;

## Comparison Operator

A comparison operator is used to compare the two values and test whether they are same.

1. **Equal to** (=) – The = symbol used to filter the results that are equal to a certain value.

Syntax – Select \* from TN where CN=value;

Select \* from Velocity where Mockmarfks=16;

1. **Not equal to** (!=) - The != symbol used to filter the results that are not equal to a certain value.

Syntax – Select \* from TN where CN!=value;

Select \* from Velocity where Mockmarks!=16;

1. **Greater than** (>) – This > symbol is used to filter the results where a column value is greater than the required value.

Syntax – Select \* from TN where CN>value;

Select \* from Velocity where Mockmarks>16;

1. **Less than** (<) – This < symbol is used to filter the results where a column value is less than the required value.

Syntax – Select \* from TN where CN<value;

Select \* from Velocity where Mockmarks<16;

1. **Greater than or equal to** (>=) – This >= symbol is used to filter the results where a column value is greater than or equal to the required value.

Syntax – Select \* from TN where CN>=value;

Select \* from Velocity where Mockmarks>=16;

1. **Less than or equal to** (<=) – This <= symbol is used to filter the results where a column value is less than or equal to the required value.

Syntax – Select \* from TN where CN<=value;

Select \* from Velocity where Mockmarks<=16;

**Between Operator**

It is used to select the values within the given range with a where clause.

Syntax – select \* from TN where CN between value 1 and value2;

Select \* from velocity where mockmarks between 15 and 17;

**Not Between Operator** – It will fetch the data except the applied conition.

Syntax – select \* from TN where CN not between value 1 and value2;

Select \* from velocity where mockmarks not between 15 and 17;

**Like Operator**

In like operator where clause is used to compare a value to similar values using wildcard operators.

There are 2 wildcards used in conjunction with the like operator.

1. Percent sign **(%)** – matches one or more characters
2. Underscore **(\_)** – matches one character

Syntax – Select \* from TN Where CN like pattern;

Select \* from Velocity Where fname like ‘a%’

|  |  |
| --- | --- |
| **Like Operator** | **Description** |
| Where CN like ‘a%’ | Finds the values that starts with ‘a’ |
| Where CN like ‘%a’ | Finds the values that ends with ‘a’ |
| Where CN like ‘%ah%’ | Finds the value that have ‘ah’ in any position |
| Where CN like ‘\_a%’ | Finds the value that have ‘a’ in the second position. |
| Where CN like ‘a\_%’ | Finds any value that start with ‘a’ and are at least of 2 characters. |
| Where CN like ‘a%y’ | Finds the value that starts with ‘a’ and ends with ‘y’. |

**Not like operator**

Sometimes we want to get the records that doesn’t match the like pattern in that case we can use not like operator.

Syntax – Select \* from TN where CN not like pattern;

select \* from velocity where fname not like 'S%';

**Distinct Keyword**

It is used to show the unique values from the table.

Syntax – Select distinct CN from TN;

Select distinct Mockmarks from velocity;

**Top Clause/Rownum Clause**

It is used to limit the number of rows returned by a query.

Select \* from TN where Rownum contition;

Select \* from Velocity where Rownum<=3;

**In operator**

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

Syntax – Select \* from TN where CN condition;

Select \* from velocity where mockmarks in (13,15,17);

**Order By clause**

It is used to sort the records in your result set in ascending and descending order.

Syntax – Select \* from TN order by CN Asc/Desc;

Select \* from velocity order by mockmarks asc;

Select \* from velocity order by mockmarks desc;

## Aggregate Function

1. **Count** – Count is used to count the records from the table.

Syntax – Select count(CN) from Velocity;

select count(fname) from velocity;

1. **AVG** – It shows the avg value from the table

Syntax – Select avg(CN) from Velocity;

select avg(mockmarks) from velocity;

1. **Sum** – It is used for addition of the value;

Syntax – Select sum(CN) from Velocity;

select sum(mockmarks) from velocity;

1. **Max** – It shows the max value

Syntax – Select max(CN) from Velocity;

select max(mockmarks) from velocity;

1. **Min** – It shows the Min value

Syntax – Select min(CN) from Velocity;

select min(mockmarks) from velocity;

**Group by**

To get the result of table in group we are using group by clause.

The group by statement is often used with aggregate function.

**Having clause**

The having clause places the condition in the groups, defined by the group by clause.

Select CN1,max(CN2) from TN group by CN1 having max(CN2)>=value;

Select Fname,max(mockmarks) from velocity group by fname having max(mockmarks)>=15;

## Union

1. It is used to combine the result set of two or more than two oracle select statement.
2. It combines both select statement and removes duplicate values.
3. It gives unique vales.

One column should be same in both the tables.

Syntax – Select CN from TN1 union Select CN from TN2;

## Union All

1. It allows duplicate values and it combines all the data from both the tables.

Syntax - Select CN from TN1 union all Select CN from TN2;.

## SQL table Constraint

SQL constraint are used to specify the values for data in a table. Constraint can be used when the table is created with the create table statement.

**Types of constraint**

1. Not null
2. Unique
3. Primary key
4. Foreign key
5. Check constraint
6. Default constraint

**Not null** – It does not allow the user to insert null values. If we want to make any field as mandatory then we will use not null constraint.

Syntax- Create table velocity(

studentid int,

Fname varchar(255) not null,

Lname varchar(255));

**Unique** – It does not accepts duplicate values it only accepts unique vales.

Syntax- Create table velocity(

studentid int unique,

Fname varchar(255) ,

Lname varchar(255));

**Primary key** – It is a combination of not null and unique. It does not allows duplicate vales. It does not allows null values. In each table only one primary key is used.

Syntax- Create table velocity(

studentid int primary key,

Fname varchar(255) ,

Lname varchar(255));

**Foreign key** – It is used for mapping the relationship between two or more tables. Primary key of one table acts as foreign key of another table. It can accept multiple null values and multiple duplicate values.

Syntax- Create table Order1(

Orderid int,

Oname Varchar(255),

Customerid int, foreign key(Customerid) references customer(Customerid)

);

Create table customer(

Customerid int primary key,

Cname Varchar(255),

Lname Varchar(255),

Csalary int,

City varchar(255)

);

Insert into customer(Customerid,Cname,Lname,Csalary,City)

Values (1,'Rahul','Gadbail',50000,'Akola');

Insert into customer(Customerid,Cname,Lname,Csalary,City)

Values (2,'Vijay','Shinde',40000,'Pune');

Insert into customer(Customerid,Cname,Lname,Csalary,City)

Values (3,'Sonali','Ahirwar',70000,'Mumbai');

Insert into customer(Customerid,Cname,Lname,Csalary,City)

Values (4,'Mukesh','Shukla',65000,'Delhi');

Insert into customer(Customerid,Cname,Lname,Csalary,City)

Values (5,'Kirti','Gathade',37000,'Goa');

Select \* from customer;

drop table customer;

Create table Order1(

Orderid int,

Oname Varchar(255),

Customerid int, foreign key(Customerid) references customer(Customerid)

);

Insert into Order1(Orderid,oname,Customerid)

Values (98,'Nokia',2);

Insert into Order1(Orderid,oname,Customerid)

Values (99,'Apple',4);

Insert into Order1(Orderid,oname,Customerid)

Values (101,'Oneplus',6);

Insert into Order1(Orderid,oname,Customerid)

Values (102,'MI',8);

Select \* from Order1;

drop table order1;

**Check Constraint**

1. It is used to restrict the value of column between the range.
2. We can apply check constraint on a particular column that we want while creating a table.

Syntax - Create table Order1(

Orderid int, Check(Orderid<100),

Oname Varchar(255),

Customerid int

);

**Default constraint**

1. It is used to set a default value.
2. If we are not6 specifying any value for the column then it will take the default vale.

Syntax - Create table customer(

Customerid int primary key,

Cname Varchar(255),

Lname Varchar(255),

Csalary int default 20000,

City varchar(255)

);

**Aliases** – Aliases are temporary name given to column or table for the purpose of a particular SQL query.

Syntax – Select CN AS C from TN;

Select Cname AS C from customer;

## SQL Joins

Joins are used to retrieve the data from multiple tables.

A SQL sever join is performed whenever two or more tables are joined in a SQL statement.

**Types of Join**

1. Inner join
2. Left join
3. Right join
4. Full join

**Inner Join** – It is the most common type of join. It gives matching record value from both the table.

Syntax – Select CN1,CN2,CN3(second) from TN1 inner join TN2 on TN1.CN=TN2.CN;

**Left join** – It returns all the records from the left table and matching records from right table.

Syntax – Select CN1,CN2,CN3(second) from TN1 Left join TN2 on TN1.CN=TN2.CN;

**Right join** – It returns all the records from the Right table and matching records from left table.

Syntax – Select CN1,CN2,CN3(second) from TN1 Right join TN2 on TN1.CN=TN2.CN;

**Full join** - It returns all the records from both the tables.

Syntax – Select CN1,CN2,CN3(second) from TN1 Full join TN2 on TN1.CN=TN2.CN;

## Interview Questions

**Q- How to find the highest salary from the table?**

Select max(salary) from TN;

**Q- How to find the second highest salary?**

select max(Csalary) from Customer where Csalary not in (select max(Csalary) from Customer);

**Q- How to find the 3rd highest salary?**

select max(Csalary) from Customer where Csalary < (select max(Csalary) from Customer where Csalary not in (select max(Csalary) from Customer));

Q- What all are the join in SQL?

Q- Constraint is SQL?

Q- Aggregate functions?

Q- Different types of SQL commands?

Q- Difference between delete, drop and truncate?

Q- Difference between Union and Union all?

Q- Different operators in SQL?

**The End**