SQL Query

**SQL**

Structural query language

2 types of databses -->>

1. Relational
2. Non relational

**Most asked interview question answers.**

1. What is the difference between SQL and MySQL?

|  |  |
| --- | --- |
| ****SQL**** | ****MySQL**** |
| SQL is a standard language which stands for Structured Query Language based on the English language | MySQL is a database management system. |
| SQL is the core of the relational database which is used for accessing and managing database | MySQL is an RDMS (Relational Database Management System) such as SQL Server, Informix etc. |

1. What are the different subsets of the SQL?

**DDL** Data Defintion Langauge It allows to perform CREATE, ALTER, DELETE

**DML** Data Manipulation Langauge To access and manipuilate the data. Insert, update, delete, retrieve the data.

**DCL** It allows you to access the data Example grant, rewoke access permisions.

1. What do you mean by DBMS? What are its different types?

DBMS -> Data base management system

It is the application software which interact with the user, application and database itself to capture and analyse the data.

Relational database management system

Non relational database system.

There are 2 types of DBMS

1. Relation :- Stores data in a series of tables with relation ship defined between them.
2. Non-relational:- Stores the data data in forms of document, key-value, graphs, columunar

4) What are the examples of relational databses?

Relational databses uses SQL.

-- will add a comment

What are the relational databases management systems

1. mysql
2. SQL server
3. Oracle

Each databses has different flavour of the sql. But base is common.

5) What are the example of the non relational databse?

1) Mongodb

2) Cassandra

3) CouchDB

6) What is OLTP?

OLTP is the online transaction processing.

It is a type of database system that is optimized for the large number of shorts, transaction operational.

They are characterized by the concurrent, complex transaction relationship, and short transaction lifetimes.

For example e-commerce, banking

1. Types of the relationships in SQL?

One to one

One to many/ Many to one

Many to many

Self referencing relationships

1. What is use of SELECT statement?

SELECT statement is used to query the database and retrive data from one or more tables.

1. Common clauses used with the **SELECT** statement in query?

**WHERE** :- To filter the record depend on the certain criteria

**ORDER BY** :- Sort the table by asc, des or with specific fields

**GROUP BY** :- To group entry with identicaldata and may used with aggretion methods to obtain summarised databse tables.

**HAVING** :- To filter the record with GROUP BY clause. WHERE can not filter aggregate data HAVING can do it.

1. What is the difference between SQL and NO SQL databases?

|  |  |
| --- | --- |
| **SQL** | **No-SQL** |
| 1. SQL is a relational database management system. | 1. While No-SQL is a non-relational or distributed database management system. |
| 2. The query language used in this database system is a structured query language. | 2. The query language used in the No-SQL database systems is a non-declarative query language. |
| 3. The schema of SQL databases is predefined, fixed, and static. | 3. The schema of No-SQL databases is a dynamic schema for unstructured data. |
| 4. These databases are vertically scalable. | 4. These databases are horizontally scalable. |
| 5. The database type of SQL is in the form of tables, i.e., in the form of rows and columns. | 5. The database type of No-SQL is in the form of documents, key-value, and graphs. |
| 6. It follows the ACID model. | 6. It follows the BASE model. |
| 7. Complex queries are easily managed in the SQL database. | 7. NoSQL databases cannot handle complex queries. |
| 8. This database is not the best choice for storing hierarchical data. | 8. While No-SQL database is a perfect option for storing hierarchical data. |
| 9. All SQL databases require object-relational mapping. | 9. Many No-SQL databases do not require object-relational mapping. |
| 10. Gauges, CircleCI, Hootsuite, etc., are the top enterprises that are using this query language. | 10. Airbnb, Uber, and Kickstarter are the top enterprises that are using this query language. |
| 11. SQLite, Ms-SQL, Oracle, PostgreSQL, and MySQL are examples of SQL database systems. | 11. Redis, MongoDB, Hbase, BigTable, CouchDB, and Cassandra are examples of NoSQL database systems. |

1. What are union, minus, intersect command?

**UNION** is used to combine result of 2 tables Which also removes the duplicate entries.

**Minus** return row from the first query but not from the second query.

**INTERSACT** Combine the result of the 2 queries into single row. Few conditions may be satisfied like both SELECT query have a same amount of query.

1. where i can store databases?
2. There are several options available for storing databases, depending on your needs and resources:
3. Local Server: You can store your database on a local server or a dedicated server located on-premises. This option provides full control over the database and can offer faster performance due to the proximity of the data to the users. However, it may require ongoing maintenance and upgrades, and may not be as scalable as other options.
4. Cloud-based Database: You can store your database on a cloud-based platform such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). This option provides greater scalability, flexibility, and availability compared to a local server. You can also choose from different types of cloud-based databases, such as relational databases, NoSQL databases, or object-based databases, depending on your needs.
5. Managed Database Services: Many cloud providers also offer managed database services, such as Amazon RDS, Azure SQL Database, or Google Cloud SQL. These services take care of database administration tasks such as backups, patching, and monitoring, allowing you to focus on application development. Managed database services are often more expensive than self-hosted databases, but they can save time and resources.
6. Hybrid Approach: You can also use a combination of local and cloud-based databases to balance performance, scalability, and cost. For example, you can store frequently accessed data on a local server and less frequently accessed data on a cloud-based database.

Questions ans answers

What re SQL joins?

or

Types of joins?

or

Inner join and outer join In SQL

tblCountry Inner join result tblstate

StateId CountryId CountryName

1 1 Maharastra

2 1 Punjab

3 2 Kathmandu

4 N California

CountryId Country StateId CountryId StateName

1 india 1 1 Maharastra

1 india 2 1 Punjab

2 Nepal 3 2 Kathmandu

CountryId Country

1 India

2 Nepal

3 Srilanka

SQL query will be

select \* from tblCountry

inner join tblstate

on tblCountry.Countryid=tblState.Countryid

**Outer join** are of 3 types

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

StateId CountryId CountryName

1 1 Maharastra

2 1 Punjab

3 2 Kathmandu

4 N California

CountryId Country StateId CountryId StateName

1 india 1 1 Maharastra

1 india 2 1 Punjab

2 Nepal 3 2 Kathmandu

4 Nepal N N N

CountryId Country

1 India

2 Nepal

3 Srilanka

This is left side of left statement so it is considered as a left table. Same tblstate is right side

select \* from tblCountry left join tblstate

on tblCountry.Countryid=tblState.Countryid

Right join

StateId CountryId CountryName

1 1 Maharastra

2 1 Punjab

3 2 Kathmandu

4 N California

CountryId Country StateId CountryId StateName

1 india 1 1 Maharastra

1 india 2 1 Punjab

2 Nepal 3 2 Kathmandu

N N 4 N California

CountryId Country

1 India

2 Nepal

3 Srilanka

select \* from tblCountry right join tblstate

on tblCountry.Countryid=tblState.Countryid

Full outer join

CountryId Country

1 India

2 Nepal

3 Srilanka

StateId CountryId CountryName

1 1 Maharastra

2 1 Punjab

3 2 Kathmandu

4 N California

CountryId Country StateId CountryId StateName

1 india 1 1 Maharastra

1 india 2 1 Punjab

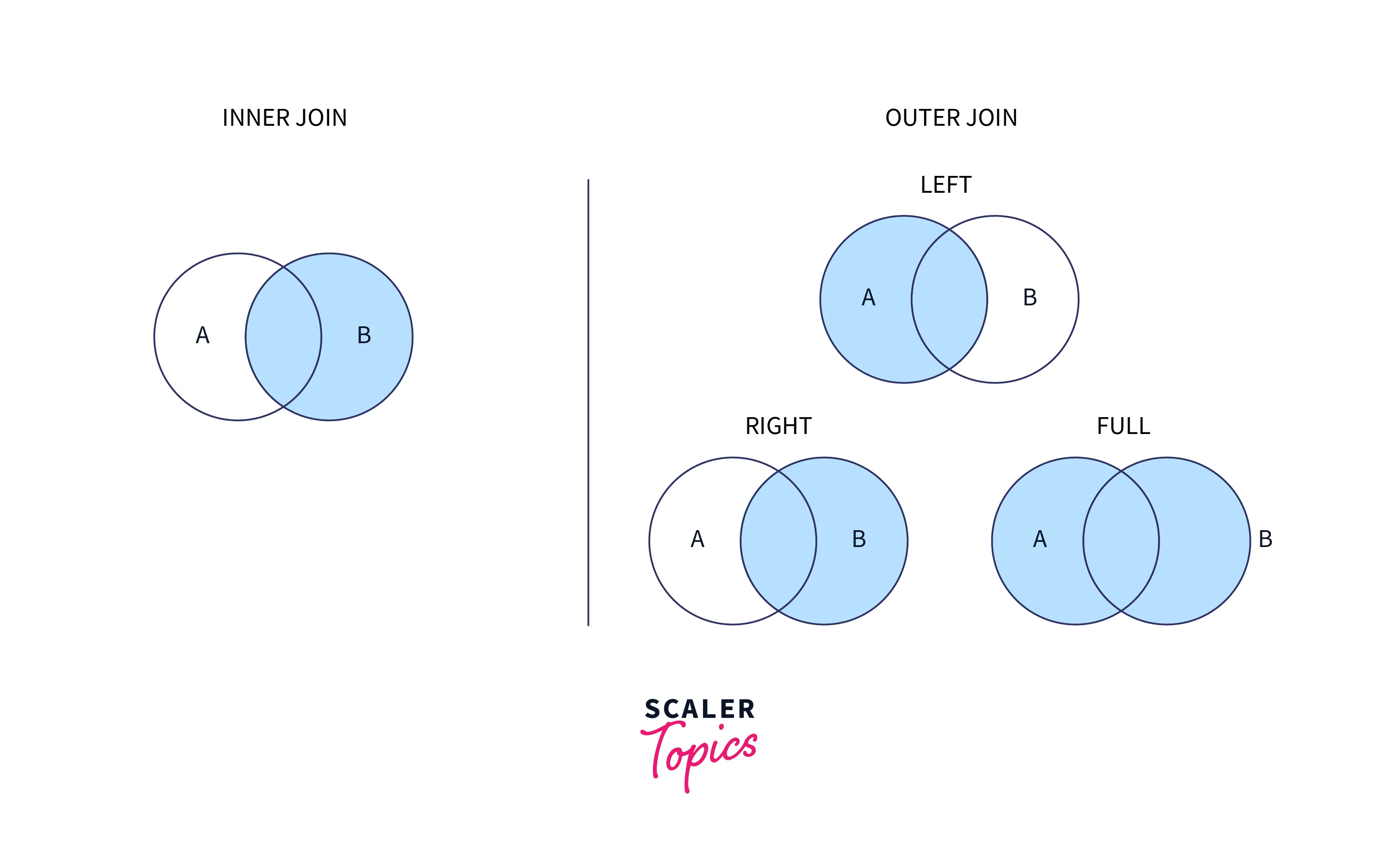
2 Nepal 3 2 Kathmandu

4 Srilanka N N N

N N 4 N California

select \* from tblCountry full outer join tblstate

on tblCountry.Countryid=tblState.Countryid



Referance

<https://www.youtube.com/watch?v=KTvYHEntvn8&t=49s>

---------------------------------------------------------------------------------------------------------------------------

**Things need to know about SQL.**

1. SQL and SQUEL both are same.
2. SQL is not the case sensitive langauge
3. Spaces new lines are ignored by the langauge

Things need to remember

Note that in SQL only column name is not case sensitive

Data inside the column is case sensitive.

USE // This will use to select the database name from our stored databases

SELECT \* //This will select all the columns

FROM customers //

To do a qury sequence is very important

1. select
2. from
3. where
4. order by

Create the SQL text file

DROP DATABASE IF EXISTS `store`;

CREATE DATABASE `store`;

USE `store`;

CREATE TABLE `products` (

  `product\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `name` varchar(50) NOT NULL,

  `quantity\_in\_stock` int(11) NOT NULL,

  `unit\_price` decimal(4,2) NOT NULL,

  PRIMARY KEY (`product\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `products` VALUES (1,'Foam Dinner Plate',70,1.21);

INSERT INTO `products` VALUES (2,'Pork - Bacon,back Peameal',49,4.65);

INSERT INTO `products` VALUES (3,'Lettuce - Romaine, Heart',38,3.35);

INSERT INTO `products` VALUES (4,'Brocolinni - Gaylan, Chinese',90,4.53);

INSERT INTO `products` VALUES (5,'Sauce - Ranch Dressing',94,1.63);

INSERT INTO `products` VALUES (6,'Petit Baguette',14,2.39);

INSERT INTO `products` VALUES (7,'Sweet Pea Sprouts',98,3.29);

INSERT INTO `products` VALUES (8,'Island Oasis - Raspberry',26,0.74);

INSERT INTO `products` VALUES (9,'Longan',67,2.26);

INSERT INTO `products` VALUES (10,'Broom - Push',6,1.09);

CREATE TABLE `shippers` (

  `shipper\_id` smallint(6) NOT NULL AUTO\_INCREMENT,

  `name` varchar(50) NOT NULL,

  PRIMARY KEY (`shipper\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `shippers` VALUES (1,'Hettinger LLC');

INSERT INTO `shippers` VALUES (2,'Schinner-Predovic');

INSERT INTO `shippers` VALUES (3,'Satterfield LLC');

INSERT INTO `shippers` VALUES (4,'Mraz, Renner and Nolan');

INSERT INTO `shippers` VALUES (5,'Waters, Mayert and Prohaska');

CREATE TABLE `customers` (

  `customer\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `first\_name` varchar(50) NOT NULL,

  `last\_name` varchar(50) NOT NULL,

  `birth\_date` date DEFAULT NULL,

  `phone` varchar(50) DEFAULT NULL,

  `address` varchar(50) NOT NULL,

  `city` varchar(50) NOT NULL,

  `state` char(2) NOT NULL,

  `points` int(11) NOT NULL DEFAULT '0',

  PRIMARY KEY (`customer\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `customers` VALUES (1,'Babara','MacCaffrey','1986-03-28','781-932-9754','0 Sage Terrace','Waltham','MA',2273);

INSERT INTO `customers` VALUES (2,'Ines','Brushfield','1986-04-13','804-427-9456','14187 Commercial Trail','Hampton','VA',947);

INSERT INTO `customers` VALUES (3,'Freddi','Boagey','1985-02-07','719-724-7869','251 Springs Junction','Colorado Springs','CO',2967);

INSERT INTO `customers` VALUES (4,'Ambur','Roseburgh','1974-04-14','407-231-8017','30 Arapahoe Terrace','Orlando','FL',457);

INSERT INTO `customers` VALUES (5,'Clemmie','Betchley','1973-11-07',NULL,'5 Spohn Circle','Arlington','TX',3675);

INSERT INTO `customers` VALUES (6,'Elka','Twiddell','1991-09-04','312-480-8498','7 Manley Drive','Chicago','IL',3073);

INSERT INTO `customers` VALUES (7,'Ilene','Dowson','1964-08-30','615-641-4759','50 Lillian Crossing','Nashville','TN',1672);

INSERT INTO `customers` VALUES (8,'Thacher','Naseby','1993-07-17','941-527-3977','538 Mosinee Center','Sarasota','FL',205);

INSERT INTO `customers` VALUES (9,'Romola','Rumgay','1992-05-23','559-181-3744','3520 Ohio Trail','Visalia','CA',1486);

INSERT INTO `customers` VALUES (10,'Levy','Mynett','1969-10-13','404-246-3370','68 Lawn Avenue','Atlanta','GA',796);

CREATE TABLE `order\_statuses` (

  `order\_status\_id` tinyint(4) NOT NULL,

  `name` varchar(50) NOT NULL,

  PRIMARY KEY (`order\_status\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `order\_statuses` VALUES (1,'Processed');

INSERT INTO `order\_statuses` VALUES (2,'Shipped');

INSERT INTO `order\_statuses` VALUES (3,'Delivered');

CREATE TABLE `orders` (

  `order\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `customer\_id` int(11) NOT NULL,

  `order\_date` date NOT NULL,

  `status` tinyint(4) NOT NULL DEFAULT '1',

  `comments` varchar(2000) DEFAULT NULL,

  `shipped\_date` date DEFAULT NULL,

  `shipper\_id` smallint(6) DEFAULT NULL,

  PRIMARY KEY (`order\_id`),

  KEY `fk\_orders\_customers\_idx` (`customer\_id`),

  KEY `fk\_orders\_shippers\_idx` (`shipper\_id`),

  KEY `fk\_orders\_order\_statuses\_idx` (`status`),

  CONSTRAINT `fk\_orders\_customers` FOREIGN KEY (`customer\_id`) REFERENCES `customers` (`customer\_id`) ON UPDATE CASCADE,

  CONSTRAINT `fk\_orders\_order\_statuses` FOREIGN KEY (`status`) REFERENCES `order\_statuses` (`order\_status\_id`) ON UPDATE CASCADE,

  CONSTRAINT `fk\_orders\_shippers` FOREIGN KEY (`shipper\_id`) REFERENCES `shippers` (`shipper\_id`) ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `orders` VALUES (1,6,'2019-01-30',1,NULL,NULL,NULL);

INSERT INTO `orders` VALUES (2,7,'2018-08-02',2,NULL,'2018-08-03',4);

INSERT INTO `orders` VALUES (3,8,'2017-12-01',1,NULL,NULL,NULL);

INSERT INTO `orders` VALUES (4,2,'2017-01-22',1,NULL,NULL,NULL);

INSERT INTO `orders` VALUES (5,5,'2017-08-25',2,'','2017-08-26',3);

INSERT INTO `orders` VALUES (6,10,'2018-11-18',1,'Aliquam erat volutpat. In congue.',NULL,NULL);

INSERT INTO `orders` VALUES (7,2,'2018-09-22',2,NULL,'2018-09-23',4);

INSERT INTO `orders` VALUES (8,5,'2018-06-08',1,'Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis.',NULL,NULL);

INSERT INTO `orders` VALUES (9,10,'2017-07-05',2,'Nulla mollis molestie lorem. Quisque ut erat.','2017-07-06',1);

INSERT INTO `orders` VALUES (10,6,'2018-04-22',2,NULL,'2018-04-23',2);

CREATE TABLE `order\_items` (

  `order\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `product\_id` int(11) NOT NULL,

  `quantity` int(11) NOT NULL,

  `unit\_price` decimal(4,2) NOT NULL,

  PRIMARY KEY (`order\_id`,`product\_id`),

  KEY `fk\_order\_items\_products\_idx` (`product\_id`),

  CONSTRAINT `fk\_order\_items\_orders` FOREIGN KEY (`order\_id`) REFERENCES `orders` (`order\_id`) ON UPDATE CASCADE,

  CONSTRAINT `fk\_order\_items\_products` FOREIGN KEY (`product\_id`) REFERENCES `products` (`product\_id`) ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `order\_items` VALUES (1,4,4,3.74);

INSERT INTO `order\_items` VALUES (2,1,2,9.10);

INSERT INTO `order\_items` VALUES (2,4,4,1.66);

INSERT INTO `order\_items` VALUES (2,6,2,2.94);

INSERT INTO `order\_items` VALUES (3,3,10,9.12);

INSERT INTO `order\_items` VALUES (4,3,7,6.99);

INSERT INTO `order\_items` VALUES (4,10,7,6.40);

INSERT INTO `order\_items` VALUES (5,2,3,9.89);

INSERT INTO `order\_items` VALUES (6,1,4,8.65);

INSERT INTO `order\_items` VALUES (6,2,4,3.28);

INSERT INTO `order\_items` VALUES (6,3,4,7.46);

INSERT INTO `order\_items` VALUES (6,5,1,3.45);

INSERT INTO `order\_items` VALUES (7,3,7,9.17);

INSERT INTO `order\_items` VALUES (8,5,2,6.94);

INSERT INTO `order\_items` VALUES (8,8,2,8.59);

INSERT INTO `order\_items` VALUES (9,6,5,7.28);

INSERT INTO `order\_items` VALUES (10,1,10,6.01);

INSERT INTO `order\_items` VALUES (10,9,9,4.28);

CREATE TABLE `sql\_store`.`order\_item\_notes` (

  `note\_id` INT NOT NULL,

  `order\_Id` INT NOT NULL,

  `product\_id` INT NOT NULL,

  `note` VARCHAR(255) NOT NULL,

  PRIMARY KEY (`note\_id`));

INSERT INTO `order\_item\_notes` (`note\_id`, `order\_Id`, `product\_id`, `note`) VALUES ('1', '1', '2', 'first note');

INSERT INTO `order\_item\_notes` (`note\_id`, `order\_Id`, `product\_id`, `note`) VALUES ('2', '1', '2', 'second note');

Above will create the database like this

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| customer\_id | first\_name | last\_name | birth\_date | phone | address | city | state | points |
| 1 | Babara | MacCaffrey | 1986-03-28 | 781-932-9754 | 0 Sage Terrace | Waltham | MA | 2273 |
| 2 | Ines | Brushfield | 1986-04-13 | 804-427-9456 | 14187 Commercial Trail | Hampton | VA | 947 |
| 3 | Freddi | Boagey | 1985-02-07 | 719-724-7869 | 251 Springs Junction | Colorado Springs | CO | 2967 |
| 4 | Ambur | Roseburgh | 1974-04-14 | 407-231-8017 | 30 Arapahoe Terrace | Orlando | FL | 457 |
| 5 | Clemmie | Betchley | 1973-11-07 |  | 5 Spohn Circle | Arlington | TX | 3675 |
| 6 | Elka | Twiddell | 1991-09-04 | 312-480-8498 | 7 Manley Drive | Chicago | IL | 3073 |
| 7 | Ilene | Dowson | 1964-08-30 | 615-641-4759 | 50 Lillian Crossing | Nashville | TN | 1672 |
| 8 | Thacher | Naseby | 1993-07-17 | 941-527-3977 | 538 Mosinee Center | Sarasota | FL | 205 |
| 9 | Romola | Rumgay | 1992-05-23 | 559-181-3744 | 3520 Ohio Trail | Visalia | CA | 1486 |

To see all the table value

USE sql\_store;

SELECT \*

FROM customers

To see the single row

USE sql\_store;

SELECT customer\_id

FROM customers

|  |
| --- |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |

To see the 2 row customer id and first row

USE sql\_store;

SELECT customer\_id, first\_name

FROM customers

|  |  |
| --- | --- |
| customer\_id | first\_name |
| 1 | Babara |
| 2 | Ines |
| 3 | Freddi |
| 4 | Ambur |
| 5 | Clemmie |
| 6 | Elka |
| 7 | Ilene |
| 8 | Thacher |
| 9 | Romola |
| 10 | Levy |

To select column

USE sql\_store;

SELECT \*

FROM customers

WHERE customer\_id = 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| customer\_id | first\_name | last\_name | birth\_date | phone | address | city | state | points | customer\_id |
| 1 | Babara | MacCaffrey | 1986-03-28 | 781-932-9754 | 0 Sage Terrace | Waltham | MA | 2273 | 1 |

To select the customer id 1 with first name

USE sql\_store;

SELECT customer\_id, first\_name

FROM customers

WHERE customer\_id = 1

|  |  |
| --- | --- |
| customer\_id | first\_name |
| 1 | Babara |

To sort the name on specific coulum ORDER BY used

USE sql\_store;

SELECT \*

FROM customers

ORDER BY first\_name

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| customer\_id | **first\_name** | last\_name | birth\_date | phone | address | city | state | points | customer\_id |
| 4 | **Ambur** | Roseburgh | 1974-04-14 | 407-231-8017 | 30 Arapahoe Terrace | Orlando | FL | 457 | 4 |
| 1 | **Babara** | MacCaffrey | 1986-03-28 | 781-932-9754 | 0 Sage Terrace | Waltham | MA | 2273 | 1 |
| 5 | **Clemmie** | Betchley | 1973-11-07 |  | 5 Spohn Circle | Arlington | TX | 3675 | 5 |
| 6 | **Elka** | Twiddell | 1991-09-04 | 312-480-8498 | 7 Manley Drive | Chicago | IL | 3073 | 6 |
| 3 | **Freddi** | Boagey | 1985-02-07 | 719-724-7869 | 251 Springs Junction | Colorado Springs | CO | 2967 | 3 |
| 7 | **Ilene** | Dowson | 1964-08-30 | 615-641-4759 | 50 Lillian Crossing | Nashville | TN | 1672 | 7 |
| 2 | **Ines** | Brushfield | 1986-04-13 | 804-427-9456 | 14187 Commercial Trail | Hampton | VA | 947 | 2 |
| 10 | **Levy** | Mynett | 1969-10-13 | 404-246-3370 | 68 Lawn Avenue | Atlanta | GA | 796 | 10 |
| 9 | **Romola** | Rumgay | 1992-05-23 | 559-181-3744 | 3520 Ohio Trail | Visalia | CA | 1486 | 9 |
| 8 | **Thacher** | Naseby | 1993-07-17 | 941-527-3977 | 538 Mosinee Center | Sarasota | FL | 205 | 8 |

Arithmatics operation in SQL

or

Write a program to add 10 points for all the customer present in database.

select last\_name, first\_name, points, points + 10

FROM customers

Can do Addition, substraction, division, multiplication etc on similar way

|  |  |  |  |
| --- | --- | --- | --- |
| **last\_name** | **first\_name** | **points** | **points + 10** |
| MacCaffrey | Babara | 2273 | 2283 |
| Brushfield | Ines | 947 | 957 |
| Boagey | Freddi | 2967 | 2977 |
| Roseburgh | Ambur | 457 | 467 |
| Betchley | Clemmie | 3675 | 3685 |
| Twiddell | Elka | 3073 | 3083 |
| Dowson | Ilene | 1672 | 1682 |
| Naseby | Thacher | 205 | 215 |
| Rumgay | Romola | 1486 | 1496 |
| Mynett | Levy | 796 | 806 |

After arithmetic operation ca change the name like point + 10 will be discount\_factor or dicount factor using the following code.

or

What is use of AS keyword in SQL

select last\_name, first\_name, points, points + 10 AS dicount\_factor

FROM customers

|  |  |  |  |
| --- | --- | --- | --- |
| last\_name | first\_name | points | dicount\_factor |
| MacCaffrey | Babara | 2273 | 2283 |
| Brushfield | Ines | 947 | 957 |
| Boagey | Freddi | 2967 | 2977 |
| Roseburgh | Ambur | 457 | 467 |
| Betchley | Clemmie | 3675 | 3685 |
| Twiddell | Elka | 3073 | 3083 |
| Dowson | Ilene | 1672 | 1682 |
| Naseby | Thacher | 205 | 215 |
| Rumgay | Romola | 1486 | 1496 |
| Mynett | Levy | 796 | 806 |

or use a space instead of underscore like discount\_factor ill be discount factor

select last\_name, first\_name, points, points + 10 AS "dicount factor"

FROM customers

|  |  |  |  |
| --- | --- | --- | --- |
| last\_name | first\_name | points | dicount factor |
| MacCaffrey | Babara | 2273 | 2283 |
| Brushfield | Ines | 947 | 957 |
| Boagey | Freddi | 2967 | 2977 |
| Roseburgh | Ambur | 457 | 467 |
| Betchley | Clemmie | 3675 | 3685 |
| Twiddell | Elka | 3073 | 3083 |
| Dowson | Ilene | 1672 | 1682 |
| Naseby | Thacher | 205 | 215 |
| Rumgay | Romola | 1486 | 1496 |
| Mynett | Levy | 796 | 806 |

select last\_name, first\_name, points, points + 10 AS 'dicount factor'

FROM customers

String can be written in a single or double quote.

SQL Statements

1. SELECT
2. SELECT DISTINCT

Query to create a table

CREATE TABLE Customers (

CustomerID int,

CustomerName varchar(255),

ContactName varchar(255),

Address varchar(255),

City varchar(255),

PostalCode varchar(255),

Country varchar(255)

);

INSERT INTO Customers (CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES

(1, 'Alfreds Futterkiste', 'Maria Anders', 'Obere Str. 57', 'Berlin', '12209', 'Germany'),

(2, 'Ana Trujillo Emparedados y helados', 'Ana Trujillo', 'Avda. de la Constitución 2222', 'México D.F.', '05021', 'Mexico'),

(3, 'Antonio Moreno Taquería', 'Antonio Moreno', 'Mataderos 2312', 'México D.F.', '05023', 'Mexico'),

(4, 'Around the Horn', 'Thomas Hardy', '120 Hanover Sq.', 'London', 'WA1 1DP', 'UK'),

(5, 'Berglunds snabbköp', 'Christina Berglund', 'Berguvsvägen 8', 'Luleå', 'S-958 22', 'Sweden');

|  |  |  |
| --- | --- | --- |
| Sr | Command | Description |
| 1 | SQL SELECT  SELECT \* FROM table\_name; | SELECT \* FROM Customers; |
| 2 | SELECT column1, column2, ... FROM table\_name; | Select the few raw instead of full table  SELECT CustomerID, CustomerName, ContactName  FROM Customers; |
| 3 | SELECT DISTINCT column1, column2, ... FROM table\_name; | The SELECT DISTINCT statement is used to return only distinct (different) values.  SELECT DISTINCT ContactName, Address, City, PostalCode, Country  FROM Customers;  -- To check disctict works or not  INSERT INTO Customers (CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  VALUES  (6, 'MIMInap snabbköp', 'Christina Berglund', 'Berguvsvägen 8', 'Luleå', 'S-958 22', 'Sweden');  Then use below query  SELECT DISTINCT ContactName, Address, City, PostalCode, Country  FROM Customers;  It will return only differnet columns. Same columns will be omitted |
| 4 | DISTINCT WITH COUNT | lists the number of different (distinct) customer countries:  SELECT DISTINCT COUNT( DISTINCT Country) FROM Customers; |
|  | Operator Description for WHERE  = Equal  < Less than  > Greater than  <= Less than or equal  >= Greater than or equal  != Not equal to  BETWEEN Between a certain range | |
| 5 | SELECT column1, column2, ... FROM table\_name WHERE =; | The WHERE clause is used to filter records.  SELECT \* FROM Customers  where customerid = 5  SELECT \* FROM CUstomers  where city = 'London' ; |
| 6 | SELECT column1, column2, ... FROM table\_name WHERE <=digit; | SELECT \* FROM Customers  where customerid <= 3 ; |
| 7 | SELECT column1, column2, ... FROM table\_name WHERE != Value; | Aprt from not equal all the raw datawill shown  SELECT \* FROM Customers  where city != 'London' ; |
| 8 | SELECT column1, column2, ... FROM table\_name WHERE BETWEEN Value1 and value2; | Between a certain range  SELECT \* FROM Customers  where customerid between 1 and 3 ; |
| 9 | SELECT column1, column2, ... FROM table\_name WHERE Value LIKE 'L%'; | Search for a pattern  SELECT \* FROM Customers  WHERE City LIKE 'L%';  Above query will return all the city starting from L |
| 10 | SELECT column1, column2, ... FROM table\_name WHERE Column IN ('Value1','Value2'); | To specify multiple possible values for a column  SELECT \* from customers  where city IN ('London', 'Luleå');  This will return where all the rows contains London and Luleå. |
| 11 | Or | SELECT \* FROM customers  WHERE city = 'London' or city = 'Luleå'; |
| 12 | AND | Will return where city is london and country is UK  SELECT \* FROM customers  WHERE city = 'London' and country = 'UK'; |
| 13 | NOT | Will return all the resulte expect from UK  SELECT \* FROM customers  WHERE NOT country = 'UK'; |

SQL Database

|  |  |  |
| --- | --- | --- |
| Sr | Command | Description |
| 1 | CREATE DATABASE testDB; | Creates databases called “tesetDB” |
| 2 | DROP DATABSE testDB; | Completely remove all the data from testDB |
| 3 | BACKUP DATABASE testDB TO DISK = 'D:\backups\testDB.bak'; | Take backup to disk path  Only first time |
| 4 | BACKUP DATABASE testDB TO DISK = 'D:\backups\testDB.bak' WITH DIFFERENTIAL; | A differential back up reduces the back up time (since only the changes from last backup are backed up). |
| 5 | CREATE TABLE table\_name (     column1 datatype,     column2 datatype,     column3 datatype,    .... ); | Creates new table  CREATE TABLE Persons (     PersonID int,     LastName varchar(255),     FirstName varchar(255),     Address varchar(255),     City varchar(255) ); |
| 6 | INSERT INTO table\_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...); | INSERT INTO persons (lastname, firstname,address,city) VALUES ('yagnik','pokal','102/1 Raj apartment','Ahmedabad'); |
| 7 | CREATE TABLE new\_table\_name AS     SELECT column1, column2,...     FROM existing\_table\_name     WHERE ....; | Create a table from existing project  CREATE TABLE personsclone AS  SELECT lastname, firstname, address  FROM persons; |
| 8 | DROP TABLE table\_name; | Permenently delete the table  DROP TABLE personsclone; |
| 9 | TRUNCATE TABLE table\_name; | Only delete value/data inside table, it keep columnas it is  TRUNCATE TABLE personsclone; |
| 10 | ALTER TABLE table\_name ADD column\_name datatype; | ALTER TABLE use to add, delete, modify cloumns in table.  ALTER TABLE persons ADD Email varchar(255);  --SELECT \* FROM persons;  Above query will add a new columns called Email  ALTER TABLE persons  ADD Dateofbirth date; |
| 11 | ALTER TABLE table\_name DROP column\_name datatype; | ALTER TABLE use to add, delete, modify cloumns in table.  ALTER TABLE persons DROP Email;  Above query will delete a new columns called Email |
| 12 | ALTER TABLE table\_name RENAME COLUMN old\_name to new\_name; | Rename the column name  ALTER TABLE persons RENAME COLUMN firstname TO first\_name; |
| 13 | ALTER TABLE table\_name ALTER COLUMN column\_name datatype; |  |
| 14 | CREATE TABLE table\_name (     column1 datatype constraint,     column2 datatype constraint,     column3 datatype constraint,     .... ); | Create table constraint |
|  |  | ALTER TABLE use to add drop various constraints on existing table |
| 15 | * [NOT NULL](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value * [UNIQUE](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different * [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table * [FOREIGN KEY](https://www.w3schools.com/sql/sql_foreignkey.asp) - Prevents actions that would destroy links between tables * [CHECK](https://www.w3schools.com/sql/sql_check.asp) - Ensures that the values in a column satisfies a specific condition * [DEFAULT](https://www.w3schools.com/sql/sql_default.asp) - Sets a default value for a column if no value is specified * [CREATE INDEX](https://www.w3schools.com/sql/sql_create_index.asp) - Used to create and retrieve data from the database very quickly | |
| 16 | CREATE TABLE table\_name (     column1 datatype NOT NULL,     column2 datatype NOT NULL,     column3 datatype NOT NULL, ); | NOT NULL means value must not be NULL  CREATE TABLE Persons ( ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255) NOT NULL, Age int );  --Not null can also be used in alter table byt there have some confusion for postgres |
| 17 | CREATE TABLE table\_name (     column1 datatype UNIQUE,     column2 datatype NOT NULL,     column3 datatype NOT NULL, ); | CREATE TABLE persons (  ID int UNIQUE,  LastName varchar(255) NOT NULL  FirstName varchar(255) NOT NULL  );  --Check the uniq valu taken by sql or not  INSERT INTO persons (ID,LastName,FirstName)  VALUES (1,'yagnik', 'Pokal');  INSERT INTO persons (ID,LastName,FirstName)  VALUES (1,'Bharat', 'Pokal');  ERROR: duplicate key value violates unique constraint "persons\_id\_key" DETAIL: Key (id)=(1) already exists. SQL state: 23505  ALTER TABLE Persons ADD UNIQUE (ID);  TO drop uniq constraint  ALTER TABLE Persons DROP CONSTRAINT UC\_Person; |
| 18 | CREATE TABLE table\_name (     column1 datatype PRIMARY KEY,     column2 datatype NOT NULL,     column3 datatype NOT NULL, ); | Primary key is the combination of unique and NOT NULL.  CREATE TABLE Persons (  ID int PRIMARY KEY, --Added primary key  FirstName varchar(255),  LastName varchar(255)  );  --Another way to write above query  CREATE TABLE Persons (  ID int NOT NULL,  FirstName varchar(255),  LastName varchar(255),  PRIMARY KEY (ID)  );  --Adding the one raw  INSERT INTO Persons (ID, FIrstName, LastName)  VALUES (1, 'Yagnik', 'Pokal');  --Checking UNIQUE  INSERT INTO Persons (ID, FIrstName, LastName)  VALUES (1, 'Bharat', 'Pokal');  ERROR: duplicate key value violates unique constraint "persons\_pkey"  DETAIL: Key (id)=(1) already exists.  SQL state: 23505  --Checking NULL  INSERT INTO Persons (ID, FIrstName, LastName)  VALUES (NULL,'Yagnik', 'Pokal');  ERROR: null value in column "id" of relation "persons" violates not-null constraint  DETAIL: Failing row contains (null, Yagnik, Pokal).  SQL state: 23502  Can also used multiple column in primary key  CREATE TABLE Persons (  ID int NOT NULL,  FirstName varchar(255),  LastName varchar(255),  PRIMARY KEY (ID, FIrstName)--Multiple column in primary key  ); |
| 19 | Alter table with primary key | Alter table with primary key  CREATE TABLE Persons (  ID int NOT NULL,  FirstName varchar(255),  LastName varchar(255)  );  ALTER TABLE Persons ADD PRIMARY KEY (ID); |
| 20 | Drop primary key  ALTER TABLE Table\_name  DROP CONSTRAINT Key\_Name; | Delete the primary key  ALTER TABLE persons  DROP CONSTRAINT persons\_pkey;  Note that the name of the key must be same while writing above sql.  Can check the name of primary key  tables-->>persons-->>constraints  WIll shows persons\_pkey |
| 21 | The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.  A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.  The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.  The FOREIGN KEY constraint prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the parent table.  Properties :   1. Parent that is being referenced has to be unique/Primary Key. 2. Child may have duplicates and nulls. 3. Parent record can be deleted if no child exists. 4. Master table cannot be updated if child exists. 5. Must reference PRIMARY KEY in primary table. 6. Foreign key column and constraint column should have matching data types. 7. Records cannot be inserted in child table if corresponding record in master table do not exist. 8. Records of master table cannot be deleted if corresponding records in child table exits. | |
| 22 | CREATE TABLE Orders (     OrderID int NOT NULL,     OrderNumber int NOT NULL,     PersonID int,     PRIMARY KEY (OrderID),     FOREIGN KEY (PersonID) REFERENCES Persons(PersonID) ); | drop table if exists orders;  drop table if exists persons;  create table persons(  PersonID int,  LastName varchar(255),  FirstName varchar(255),  Age int,  primary key (personid)  );  insert into persons (personid, lastname, firstname, age)  values (1, 'Hansen', 'Ola', 30);  insert into persons (personid, lastname, firstname, age)  values (2, 'Svendson', 'Tove', 23);  insert into persons (personid, lastname, firstname, age)  values (3 ,'Pettersen' ,'Kari', 20);  --select \* from persons  CREATE TABLE Orders (  OrderID int NOT NULL,  OrderNumber int NOT NULL,  PersonID int,  PRIMARY KEY (OrderID),  FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  );  /\*CREATE TABLE orders (  OrderID serial PRIMARY KEY,  OrderNumber integer NOT NULL,  PersonID integer REFERENCES Persons(PersonID)  );\*/  insert into orders (OrderID, OrderNumber, PersonID)  values (1, 77895, 3);  insert into orders (OrderID, OrderNumber, PersonID)  values (2, 44678, 3);  insert into orders(OrderID, OrderNumber, PersonID)  values (3, 22456, 2);  insert into orders(OrderID, OrderNumber, PersonID)  values (4, 24562, 1);  --select \* from orders;  --select \* from persons;  To check wether the foreign key is created or not go to the Tables-->>orders-->>Constraints  It shows orders\_personid\_fkey which is foreign key |
|  | Drop foreign key  ALTER TABLE Orders  DROP CONSTRAINT Name\_of\_key; | Can drop the foreign key with following SQL.  ALTER TABLE Orders  DROP CONSTRAINT orders\_personid\_fkey;  Note that name of key can be checked by going to tables-->> orders -->> constraints |
|  | CHECK constraint  CREATE TABLE Persons ( ID int , LastName varchar(255), FirstName varchar(255), Age int CHECK (Age>=18) ); | SQL query for CHECK constraint  drop table if exists persons;  create table persons (  Firstname varchar(255),  Lastname varchar(255),  ID int primary key,  age int check (age>=18)  );  insert into persons (firstname, lastname,id, age)  values ('yagnik','pokal',1,29);  insert into persons (firstname, lastname,id, age)  values ('bharat','pokal',2,16);--This qury will give error that age check not matching |
|  | Drop check constraint  ALTER TABLE Tablename DROP CONSTRAINT Key\_Name; | alter table persons  drop constraint persons\_age\_check; |
|  | Alter table with check constraint and multiple check constraints  ALTER TABLE TableName ADD CONSTRAINT CHeckkey CHECK (Age>=18 AND City='Sandnes'); | alter table with add constraint and multiple values  drop table if exists persons;  create table persons (  Firstname varchar(255),  Lastname varchar(255),  ID int primary key,  age int  );  alter table persons  add constraint person\_age\_check check (age>=18 and Firstname='yagnik');  insert into persons (firstname, lastname,id, age)  values ('yagnik','pokal',1,29);  insert into persons (firstname, lastname,id, age)  values ('bharat','pokal',2,29); |
|  | DEFAULT statement  CREATE TABLE PERSONS (  ID int NOT NULL UNIQUE,  Name varchar(255),  Pincode varchar(255) default ('380015')  ); | drop table if exists persons;  CREATE TABLE PERSONS (  ID int NOT NULL UNIQUE,  Name varchar(255),  Pincode varchar(255) default ('380015')  );  insert into persons (ID, name, pincode) values (1, 'Yagnik', '360025');  insert into persons (ID, name) values (2, 'Bharat');  select \* from persons; |
|  | Get time and date with default statement  CREATE TABLE PERSONS (  ID int NOT NULL UNIQUE,  Name varchar(255),  Pincode varchar(255) default ('380015'),  OrderDate date DEFAULT CURRENT\_DATE,  Timenow time DEFAULT CURRENT\_TIME  ); | drop table if exists persons;  CREATE TABLE PERSONS (  ID int NOT NULL UNIQUE,  Name varchar(255),  Pincode varchar(255) default ('380015'),  OrderDate date DEFAULT CURRENT\_DATE,  Timenow time DEFAULT CURRENT\_TIME  );  insert into persons (ID, name, pincode) values (1, 'Yagnik', '360025');  insert into persons (ID, name) values (2, 'Bharat');  select \* from persons; |
|  | SQL default on alter table  ALTER TABLE persons  ALTER COLUMN Name SET DEFAULT 'Yagnik'; | SQL DEFAULT on ALTER TABLE  drop table if exists persons;  CREATE TABLE PERSONS (  ID int NOT NULL UNIQUE,  Name varchar(255),  Pincode varchar(255) default ('380015'),  OrderDate date DEFAULT CURRENT\_DATE,  Timenow time DEFAULT CURRENT\_TIME  );  ALTER TABLE persons  ALTER COLUMN Name SET DEFAULT 'Yagnik';  insert into persons (ID, name, pincode) values (1, 'Yagnik', '360025');  insert into persons (ID, name) values (2, 'Bharat');  select \* from persons; |
|  | Drop the DEFAULT  ALTER TABLE Persons  ALTER COLUMN Timenow DROP DEFAULT; | Drop the default statement in table.  drop table if exists persons;  CREATE TABLE PERSONS (  ID int NOT NULL UNIQUE,  Name varchar(255),  Pincode varchar(255) default ('380015'),  OrderDate date DEFAULT CURRENT\_DATE,  Timenow time DEFAULT CURRENT\_TIME  );  ALTER TABLE Persons  ALTER COLUMN Timenow DROP DEFAULT;  insert into persons (ID, name, pincode) values (1, 'Yagnik', '360025');  insert into persons (ID, name) values (2, 'Bharat');  select \* from persons; |

---------------------------------------------------------------------------------------------------------------------------