How to create Data base?

```
CREATE DATABASE DB

CREATE TABLE DB.person1 (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  name VARCHAR(45) NOT NULL,
  lastname VARCHAR(45) NOT NULL,
  PRIMARY KEY (id));

  CREATE TABLE DB.breeds (
  id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nameOfBreed varchar(20) UNIQUE
  );
```

HOW TO GET INFORMATION FROM TABLE?

```
SELECT*FROM DB.person1;
```

How to fill the table with information?

```
INSERT INTO DB.breeds(id,nameOfBreed)VALUES
(1,'Beagle'),
(2,'Pug'),
(3,'French Bulldog');
```

How to add new rows to a table?

```
INSERT INTO DB.breeds(nameOfBreed)VALUES
('Pudel'),
('Labrador'),
('Doberman');
```

How to change a specific field in a table?

```
UPDATE DB.breeds
SET nameOfBreed = 'Bulldog'
WHERE nameOfBreed = 'French Bulldog';
```

How to add a column to a table?

```
ALTER TABLE DB.breeds

ADD COLUMN `age` INT NOT NULL;
```

How to delete a column?

```
ALTER TABLE DB.breeds
DROP COLUMN `age`;
```

How to delete a row in a column?

```
DELETE FROM DB.breeds WHERE id=4;
```

How to delete a table?

```
DROP TABLE DB.person1;
```

How to delete a database?

```
DROP DATABASE DB;
```

Creating a new database and tables in it:

```
create Database HotelsDB;
USE HotelsDB;

CREATE TABLE Cities
(
cityID int primary key not null auto_increment,
cityName varchar(20)
);
```

Contains a foreign key — refers to the Cities table (to show in which city the hotel is located):

```
CREATE TABLE HOTELS
(
hotelID  int primary key not null auto_increment,
hotelName varchar(20),
starsCounts int not null,
creationDate date,
```

```
cityId int,
foreign key (cityId) references Cities (cityID)
);
```

Contains a foreign key — refers to the HOTELS table (to show which hotel the room belongs to):

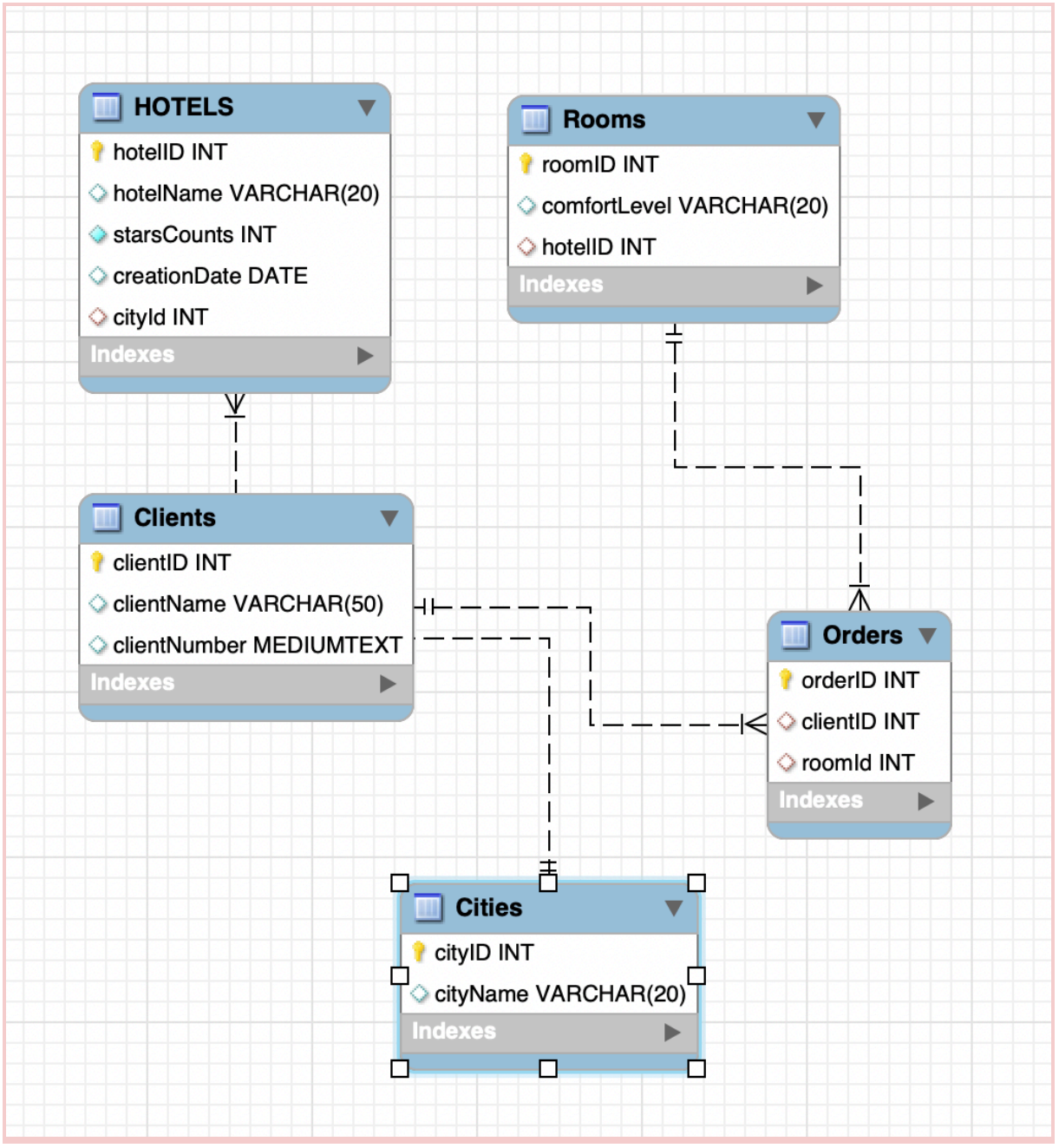
```
CREATE TABLE Rooms
(
roomID int primary key not null auto_increment,
comfortLevel varchar(20),
hotelID int,
foreign key (hotelID) references Hotels(hotelID)
)
CREATE TABLE Clients
(
clientID int primary key not null auto_increment,
clientName varchar(50),
clientNumber long
);
```

Contains 2 foreign keys:

```
Create table Orders
(
orderID int primary key not null auto_increment,
clientID int,
roomId int,
foreign key (clientID) references Clients(clientID),
foreign key (roomId) references Rooms(roomID)
);
```

To view the diagram :click Database → Reverse Engineer → select the connection you are using → select the database → Execute.

DIAGRAM:



FILLING TABLES:

Filling the Cities table:

```
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Vinnytsia');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Lutsk');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Dnipro');
```

```
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Donetsk');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Zhytomyr');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Uzhhorod');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Zaporizhzhia');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Ivano-Frankivsk');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Kyiv');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Kropyvnytskyi');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Luhansk');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Lviv');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Mykolaiv');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Odessa');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Poltava');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Rivne');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Sumy');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Ternopil');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Kharkiv');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Kherson');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Khmelnytskyi');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Cherkasy');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Chernivtsi');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Chernihiv');
INSERT INTO `HotelsDB`.`Cities` (`cityName`) VALUES ('Crimea');
```

Filling the 'Hotels' table – hotel name and number of stars:

```
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Ukraine', '2');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Fairmont Grand Hotel ', '5');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Radisson', '5');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Star', '1');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Calm', '3');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Sun', '1');
```

```
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Peace', '5');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Sunflower', '2');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('Sofia', '4');
INSERT INTO `HotelsDB`.`HOTELS` (`hotelName`, `starsCounts`) VALUES ('TopHotel', '4');
```

Filling the 'Rooms' table:

```
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES (' lux', '10');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('cheap', '13');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('comfort', '14');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('lux', '12');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('cheap', '19');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('comfort', '18');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('lux', '16');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('cheap', '15');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('comfort', '12');
INSERT INTO `HotelsDB`.`Rooms` (`comfortLevel`, `hotelID`) VALUES ('lux', '17');
```

Filling the 'Clients' table:

```
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Olha Lima', '+380987744638');
```

```
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Orest Pundyk', '+380976633547');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Ostap Lana', '+380956647836');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Natalia Bober', '+380985566437');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Robert Obertov', '+380953388765');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Inna Ivanovna', '+380963388654');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Lilia Markovych', '+380973355623');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Anna Nilson', '+380963344217');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Katerina Gerasina', '+380988844536');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Nazar Redka', '+380986633241');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Alie Myrna', '+380964455387');
INSERT INTO `HotelsDB`.`Clients` (`clientName`, `clientNumber`) VALUES ('Darina Rosa', '+380986644328');
```

Filling the 'Orders' table:

```
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('1', '1');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('2', '2');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('3', '2');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('4', '3');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('5', '4');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('6', '5');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('7', '6');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('8', '6');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('9', '7');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('10', '8');
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('11', '9');
```

```
INSERT INTO `HotelsDB`.`Orders` (`clientID`, `roomId`) VALUES ('12', '10');
```

FILLING COLUMNS:

Filling the creationDate column:

```
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2002-09-21' WHERE `hotelID` = 10;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2001-08-22' WHERE `hotelID` = 11;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2003-03-24' WHERE `hotelID` = 12;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2020-08-25' WHERE `hotelID` = 13;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2019-10-26' WHERE `hotelID` = 14;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2017-11-21' WHERE `hotelID` = 15;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2005-06-05' WHERE `hotelID` = 16;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2019-09-07' WHERE `hotelID` = 17;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2020-12-12' WHERE `hotelID` = 18;
UPDATE `HotelsDB`.`Hotels` SET `CreationDate` = '2005-10-13' WHERE `hotelID` = 19;
```

Fill the city id in the Hotels table:

```
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '1' WHERE (`hotelID` = '10');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '2' WHERE (`hotelID` = '11');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '3' WHERE (`hotelID` = '12');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '4' WHERE (`hotelID` = '13');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '5' WHERE (`hotelID` = '14');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '6' WHERE (`hotelID` = '15');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '7' WHERE (`hotelID` = '16');
```

```
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '8' WHERE (`hotelID` = '17');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '9' WHERE (`hotelID` = '18');
UPDATE `HotelsDB`.`Hotels` SET `cityId` = '10' WHERE (`hotelID` = '19');
```

SELECT queries

To get the entire table:

```
SELECT*FROM Clients;
```

To get specific columns from a table:

```
SELECT clientName,clientNumber FROM Clients;
```

Search by name:

```
SELECT clientName,clientNumber FROM Clients WHERE clientName =
'Orest Pundyk';
```

Search for clients whose name starts with a specific letter:

```
SELECT clientName,clientNumber FROM Clients WHERE clientName LIKE
'O%';
```

Search for clients whose name ends with a specific letter:

```
SELECT clientName,clientNumber FROM Clients WHERE clientName LIKE
'%a';
```

Search for clients by letters within the name:

```
SELECT clientName,clientNumber FROM Clients WHERE clientName LIKE
'%lia%';
```

Search by phone number, specifying which digits the phone number contains:

```
SELECT clientName,clientNumber FROM Clients WHERE clientNumber
LIKE '%6437%';
```

Working with numbers: select phone numbers that are greater or less than a specific number:

```
SELECT clientName,clientNumber FROM Clients WHERE clientNumber > 1;
SELECT clientName,clientNumber FROM Clients WHERE clientNumber >
3809876543245678;
SELECT clientName,clientNumber FROM Clients WHERE clientNumber <
3809876543245678;
SELECT clientName,clientNumber FROM Clients WHERE clientNumber <
3809876543245678 AND clientNumber>1;
SELECT clientName,clientNumber FROM Clients WHERE clientNumber =
380987744638 OR clientNumber=380976633547;
```

Select all phone numbers except one:
SELECT clientName,clientNumber FROM Clients WHERE clientNumber <> 380987744638;

AGGREGATE FUNCTIONS:

COUNT

1) To see how many entries are in the table:

SELECT COUNT(*)FROM Cities;
SELECT COUNT(*)FROM Clients;

2)To see how many entries are in a column:
SELECT COUNT(ClientName)FROM Clients;

3) To see how many names in the column start with the letter O:
SELECT COUNT(ClientName)FROM Clients WHERE ClientName LIKE "O%";
4)To see how many names in the column end with the letter A:
SELECT COUNT(ClientName)FROM Clients WHERE ClientName LIKE "%a";

MIN

5)To select the minimum hotel rating from the hotel rating column in the Hotels table:
SELECT Min(StarsCounts)FROM Hotels;

6) To select the hotel whose name contains the minimum number of letters (compared to others):

SELECT Min(HotelName)FROM Hotels;

7)To select which hotel was created first:
SELECT Min(CreationDate)FROM Hotels;

MAX:

8)To select the maximum rating among hotels in the Hotels table:

```
SELECT Max(StarsCounts)FROM Hotels;
```

AVG:

9)To find the arithmetic average of numbers (the average of the digits in the phone number):

```
SELECT AVG (ClientNumber)FROM Clients;
```

SUM:

10) To find the sum of all phone numbers:

```
SELECT Sum(ClientNumber)FROM Clients;
```

To show all hotels with the minimum number of stars:

```
SELECT *
FROM Hotels
WHERE StarsCounts = 1;
```

To see how many rooms there are in a hotel with a specific ID:

```
SELECT COUNT(*)
FROM Rooms
WHERE HotelID = 12;
```

Adding new columns:

```
ALTER TABLE `HotelsDB`.`Clients`
ADD COLUMN `age` INT NULL DEFAULT NULL AFTER `clientNumber`,
ADD COLUMN `salary` VARCHAR(45) NULL AFTER `age`;
```

Changing column:

```
ALTER TABLE Clients
MODIFY COLUMN salary DECIMAL(10,2) NOT NULL;
```

Filling in the age and salary columns:

```
UPDATE `HotelsDB`.`Clients` SET `age` = '22', `salary` = '10000' WHERE (`clientID` = '1');
UPDATE `HotelsDB`.`Clients` SET `age` = '22', `salary` = '500' WHERE (`clientID` = '2');
UPDATE `HotelsDB`.`Clients` SET `age` = '27', `salary` = '10000' WHERE (`clientID` = '3');
UPDATE `HotelsDB`.`Clients` SET `age` = '26', `salary` = '2000' WHERE (`clientID` = '4');
UPDATE `HotelsDB`.`Clients` SET `age` = '30', `salary` = '2000' WHERE (`clientID` = '5');
UPDATE `HotelsDB`.`Clients` SET `age` = '23', `salary` = '5000' WHERE (`clientID` = '6');
UPDATE `HotelsDB`.`Clients` SET `age` = '24', `salary` = '2000' WHERE (`clientID` = '7');
UPDATE `HotelsDB`.`Clients` SET `age` = '45', `salary` = '3000' WHERE (`clientID` = '8');
UPDATE `HotelsDB`.`Clients` SET `age` = '12', `salary` = '700' WHERE (`clientID` = '9');
UPDATE `HotelsDB`.`Clients` SET `age` = '30', `salary` = '700' WHERE (`clientID` = '10');
UPDATE `HotelsDB`.`Clients` SET `age` = '28', `salary` = '800' WHERE (`clientID` = '11');
UPDATE `HotelsDB`.`Clients` SET `age` = '29', `salary` = '5000' WHERE (`clientID` = '12');
```

Grouping and sorting data

GROUP BY

1)Group people by salary to see how many people receive the same salary.

```
SELECT COUNT(salary),salary FROM Clients GROUP BY salary;
```

2)Group people by age.

SELECT COUNT(age),age FROM Clients GROUP BY age;

GROUP BY +HAVING:

SELECT COUNT(age),age FROM Clients GROUP BY age HAVING age>22;

SELECT COUNT(age),age FROM Clients GROUP BY age HAVING age>=22;

SELECT COUNT(age),age,clientName FROM Clients HAVING clientName LIKE '%a%';

SELECT COUNT(age),age,clientName FROM Clients HAVING clientName LIKE '%a%';

ORDER BY

From largest to smallest — DESC
From smallest to largest — ASC

Sorting alphabetically:
1)SELECT*FROM HotelsDB.Clients ORDER BY clientName;

Sort in reverse alphabetical order:
2)SELECT*FROM HotelsDB.Clients ORDER BY clientName DESC;

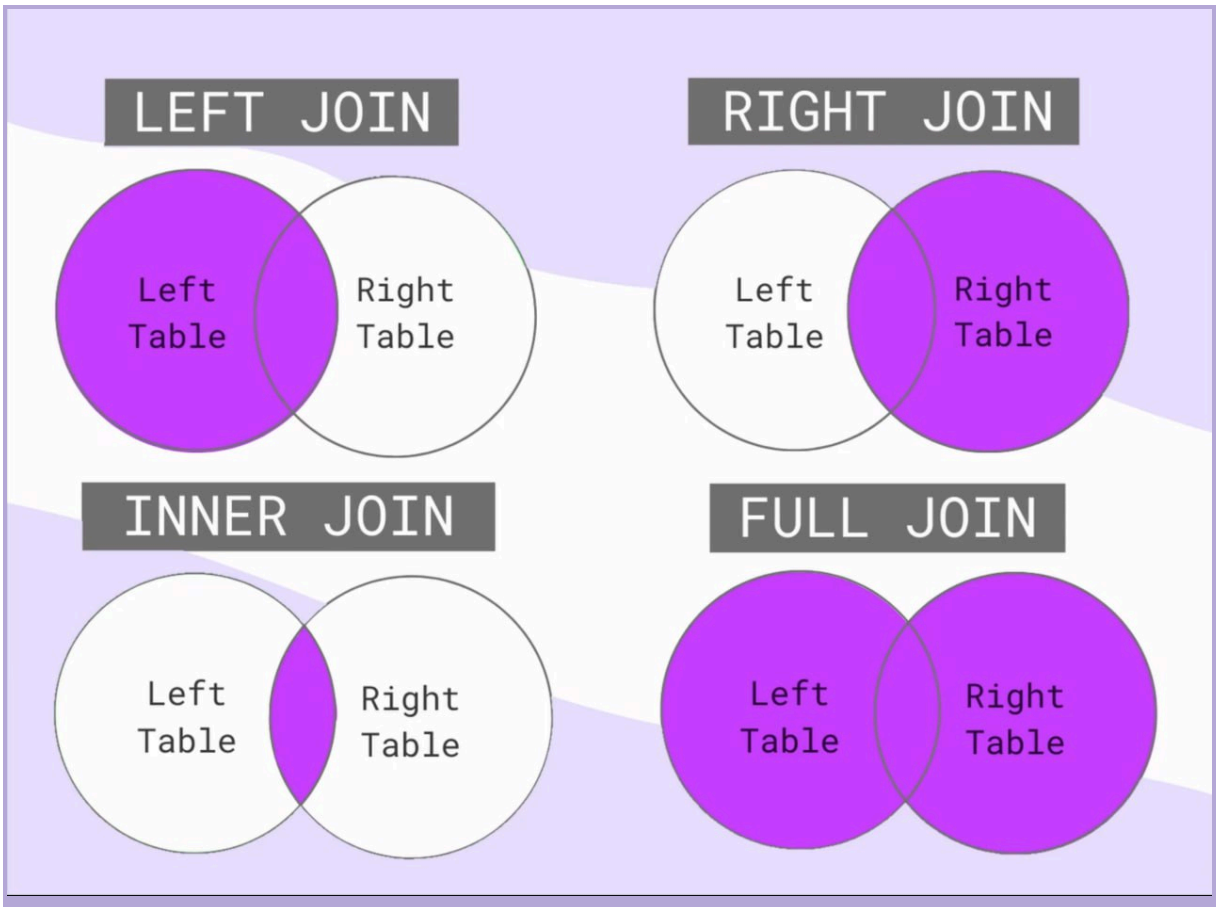
Sort salaries in ascending order:
3)SELECT*FROM HotelsDB.Clients ORDER BY salary ASC;

Sort salaries in descending order
4)SELECT*FROM HotelsDB.Clients ORDER BY salary DESC;

To find out the TOP 5 highest salaries:
5)SELECT*FROM HotelsDB.Clients ORDER BY salary DESC LIMIT 5;

6)To find out the TOP 5 highest salaries starting from the 4th person :
SELECT*FROM HotelsDB.Clients ORDER BY salary DESC LIMIT 5 OFFSET 3;

Joining tables



- LEFT JOIN – returns all rows from the left table, and from the right table only those that have a match in the left table.
- RIGHT JOIN – returns all rows from the right table, and from the left table only those that have a match in the right table.
- INNER JOIN – returns only the rows that are common to both tables (have a match).
- FULL JOIN (FULL OUTER JOIN) – returns all rows from both tables, regardless of whether they have a match, filling in NULL where there is no match.

Practical exercises:

1)INNER JOIN:
SELECT*FROM Cities JOIN Hotels ON Cities.cityID = Hotels.cityId;

2)LEFT JOIN:
SELECT*FROM Cities LEFT JOIN Hotels ON Cities.cityID = Hotels.cityId;

3)RIGHT JOIN:
SELECT*FROM Cities RIGHT JOIN Hotels ON Cities.cityID = Hotels.cityId;

INSERT INTO `HotelsDB`.`Hotels` (`hotelID`, `hotelName`, `starsCounts`, `cityId`, `CreationDate`) VALUES ('20', 'NON', '2', '1', '2004-01-01');

4)FULL JOIN:
SELECT*FROM Cities RIGHT JOIN Hotels ON Cities.cityID = Hotels.cityId;
UNION ALL
SELECT*FROM Cities LEFT JOIN Hotels ON Cities.cityID = Hotels.cityId
WHERE Cities.cityID is NULL;