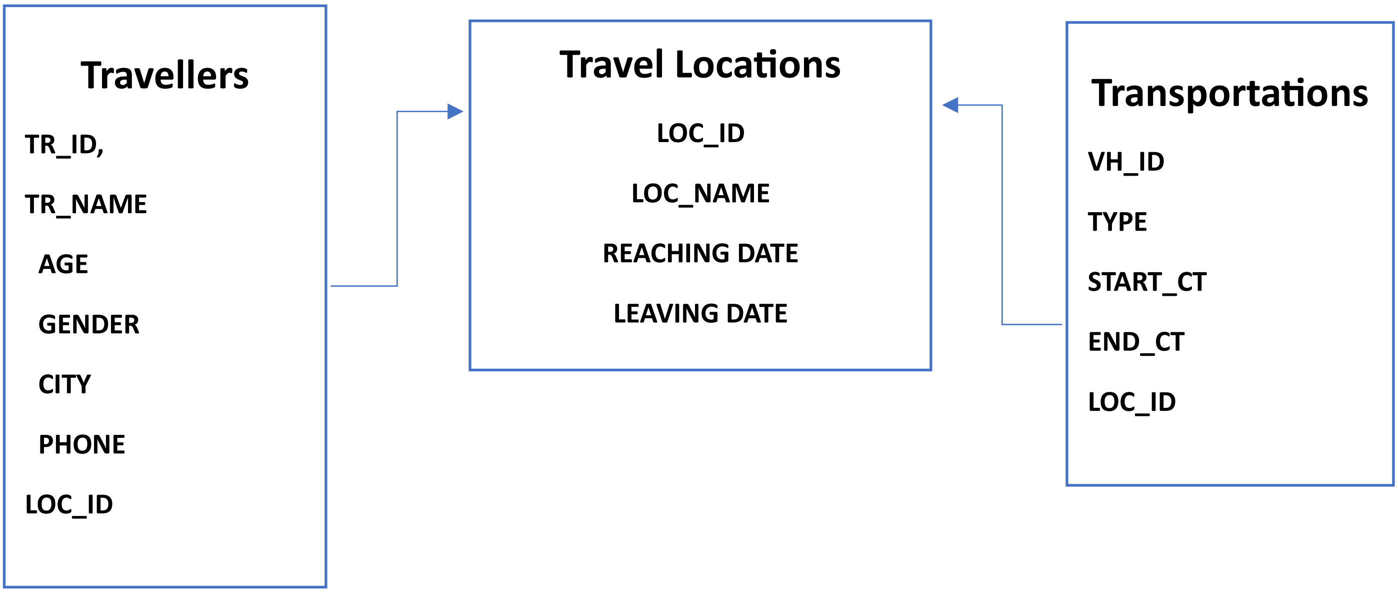
Travel Itinerary Management

Submitted by- Chetan Hemant Chuadhari

Project Details: This project displays how can we create, edit and manage a travel itinerary using **Structured Query Language (SQL).** It will help in efficient management of the complete tour and is useful for co-ordinating between various travellers, location and their stay duration.

ER DIAGRAM



1. Create Database

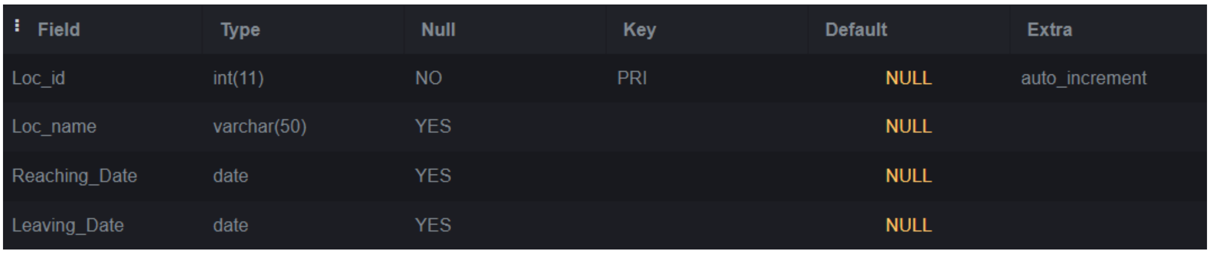
Command

```
CREATE Database Travel_Plan;
```

2. Create Table **Travel_Locations**

Commands

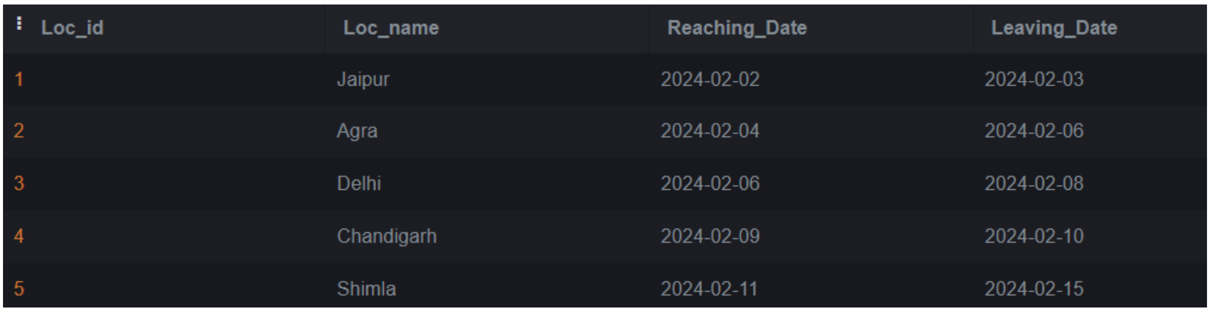
```
CREATE TABLE Travel_Locations(

Loc_id INT PRIMARY KEY AUTO_INCREMENT,

Loc_name VARCHAR(50),

Reaching_Date DATE,

Leaving_Date DATE);
```



Field	Type	Null	Key	Default	Extra
Loc_id	int(11)	NO	PRI	NULL	auto_increment
Loc_name	varchar(50)	YES		NULL	
Reaching_Date	date	YES		NULL	
Leaving_Date	date	YES		NULL	

Insert the values in Travel_Locations Table

```
INSERT INTO Travel_Locations (loc_id,loc_name,reaching_date,leaving_date)

VALUES (1,'Jaipur','2024-02-02','2024-02-03'),

(2,'Agra','2024-02-04','2024-02-06'),

(3,'Delhi','2024-02-06','2024-02-08'),

(4,'Chandigarh','2024-02-09','2024-02-10'),

(5,'Shimla','2024-02-11','2024-02-15');
```

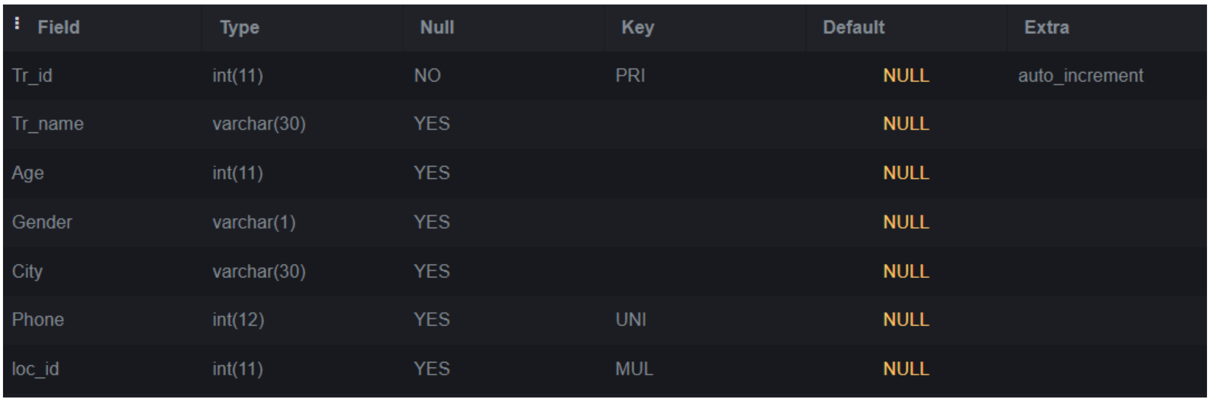


Loc_id	Loc_name	Reaching_Date	Leaving_Date
1	Jaipur	2024-02-02	2024-02-03
2	Agra	2024-02-04	2024-02-06
3	Delhi	2024-02-06	2024-02-08
4	Chandigarh	2024-02-09	2024-02-10
5	Shimla	2024-02-11	2024-02-15

3. CREATE TABLE **Travellers**

commands

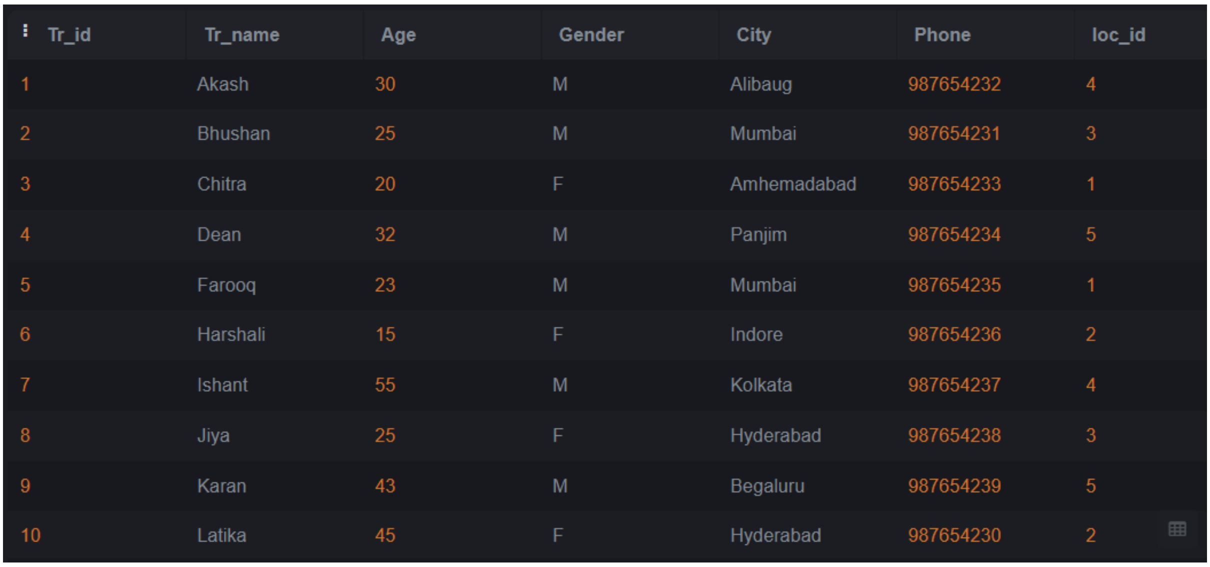
```
CREATE table Travellers(
  Tr_id INT PRIMARY KEY AUTO_INCREMENT,
  Tr_name VARCHAR(30),
  Age INT,
  Gender VARCHAR(1),
  City VARCHAR(30),
  Phone INT(12) UNIQUE,
  loc_id INT,
FOREIGN KEY(Loc_id)REFERENCES Travel_Locations(Loc_id));
```



Field	Type	Null	Key	Default	Extra
Tr_id	int(11)	NO	PRI	NULL	auto_increment
Tr_name	varchar(30)	YES		NULL	
Age	int(11)	YES		NULL	
Gender	varchar(1)	YES		NULL	
City	varchar(30)	YES		NULL	
Phone	int(12)	YES	UNI	NULL	
loc_id	int(11)	YES	MUL	NULL	

Insert values in Travellers

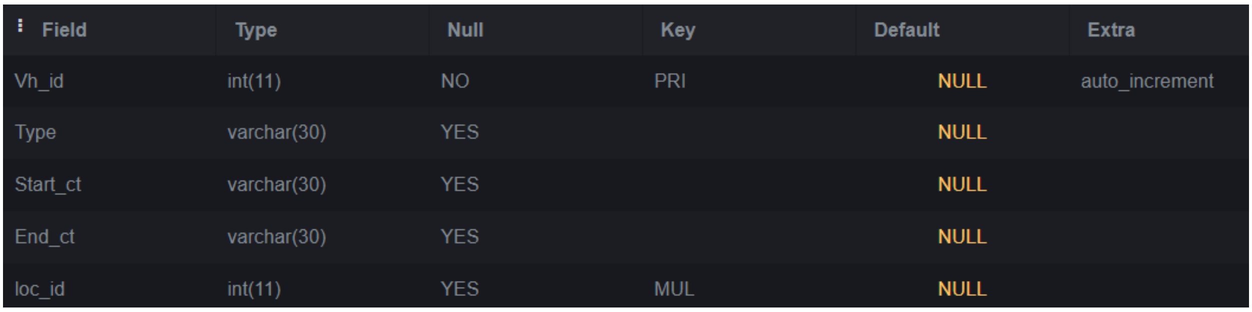
```
INSERT INTO Travellers (Tr_id,Tr_name,Age,Gender,City,Phone, loc_id)
VALUES (1,'Akash', 30,'M','Alibaug',987654232,4),
(2,'Bhushan',25,'M','Mumbai',987654231,3),
(3,'Chitra',20,'F','Amhemadabad',987654233,1),
(4,'Dean',32,'M','Panjim',987654234,5),
(5,'Farooq',23,'M','Mumbai',987654235,1),
(6,'Harshali',15,'F','Indore',987654236,2),
(7,'Ishant',55,'M','Kolkata',987654237,4),
(8,'Jiya',25,'F','Hyderabad',987654238,3),
(9,'Karan',43,'M','Begaluru',987654239,5),
(10,'Latika',45,'F','Hyderabad',987654230,2);
```



Tr_id	Tr_name	Age	Gender	City	Phone	loc_id
1	Akash	30	M	Alibaug	987654232	4
2	Bhushan	25	M	Mumbai	987654231	3
3	Chitra	20	F	Amhemadabad	987654233	1
4	Dean	32	M	Panjim	987654234	5
5	Farooq	23	M	Mumbai	987654235	1
6	Harshali	15	F	Indore	987654236	2
7	Ishant	55	M	Kolkata	987654237	4
8	Jiya	25	F	Hyderabad	987654238	3
9	Karan	43	M	Begaluru	987654239	5
10	Latika	45	F	Hyderabad	987654230	2

4. CREATE TABLE **Transportation**

```
CREATE table Transportation(
  Vh_id INT PRIMARY KEY AUTO_INCREMENT,
  Type VARCHAR(30),
  Start_ct VARCHAR(30),
  End_ct VARCHAR(30),
  loc_id INT,
FOREIGN KEy(Loc_id)REFERENCES Travel_Locations(Loc_id));
```

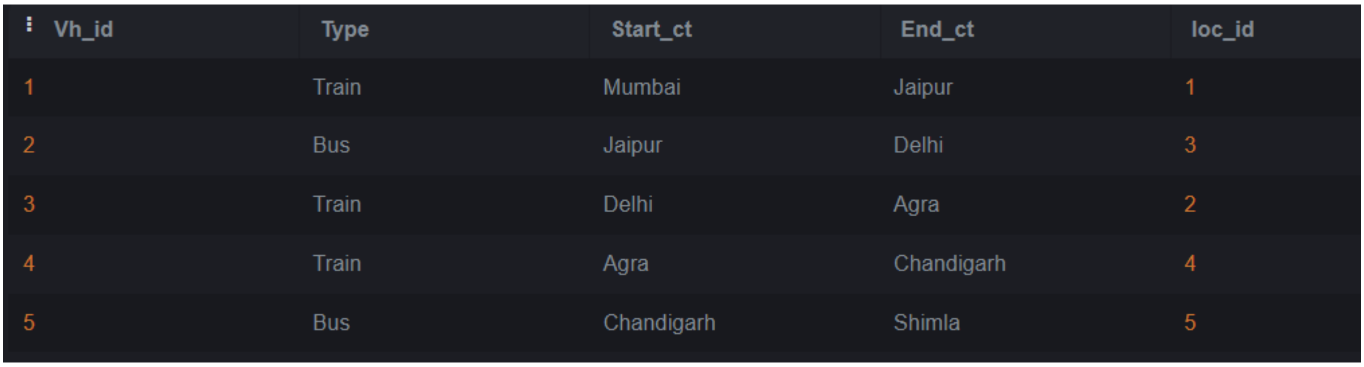


Field	Type	Null	Key	Default	Extra
Vh_id	int(11)	NO	PRI	NULL	auto_increment
Type	varchar(30)	YES		NULL	
Start_ct	varchar(30)	YES		NULL	
End_ct	varchar(30)	YES		NULL	
loc_id	int(11)	YES	MUL	NULL	

INSERT VALUE IN Transportation

Commands

```
INSERT INTO Transportation (vh_id,type,start_ct,end_ct,loc_id)
VALUES (1,'Train','Mumbai','Jaipur',1),
(2,'Bus','Jaipur','Delhi',3),
(3,'Train','Delhi','Agra',2),
(4,'Train','Agra','Chandigarh',4),
(5,'Bus','Chandigarh','Shimla',5);
```



Vh_id	Type	Start_ct	End_ct	loc_id
1	Train	Mumbai	Jaipur	1
2	Bus	Jaipur	Delhi	3
3	Train	Delhi	Agra	2
4	Train	Agra	Chandigarh	4
5	Bus	Chandigarh	Shimla	5

Here are some important query’s which help of we can do easily

Some changes in our table / data.

How to retrieve table

DESC TABLE NAME;

How to retrieve all records from table

SELECT * FROM TABLENAME;

Retrieve one particular or specific column from table

SELECT COLUMN NAME FROM TABLE NAME;

ALTER TABLE

ALTER TABLE Travellers

ADD COLUMN Food Type VARCHAR(30);

CHANGE COLUMN

ALTER TABLE Travellers

CHANGE COLUMN Phone Ph VARCHAR(30);

MODIFY COLUMN

ALTER TABLE Travellers

MODIFY COLUMN Ph VARCHAR(30) UNIQUE;

WHERE CONDITION

```
**SELECT COLUMN NAME FROM TABLE
WHERE CONDITION;**

**SELECT * FROM** TRAVEL_LOCATIONS **WHERE Loc_id = 4;**
```

LOGICAL OPERATORS

IN OPERATOR

```
SELECT * FROM Travellers

WHERE city IN (‘Mumbai’,’Hyderabad’);
```

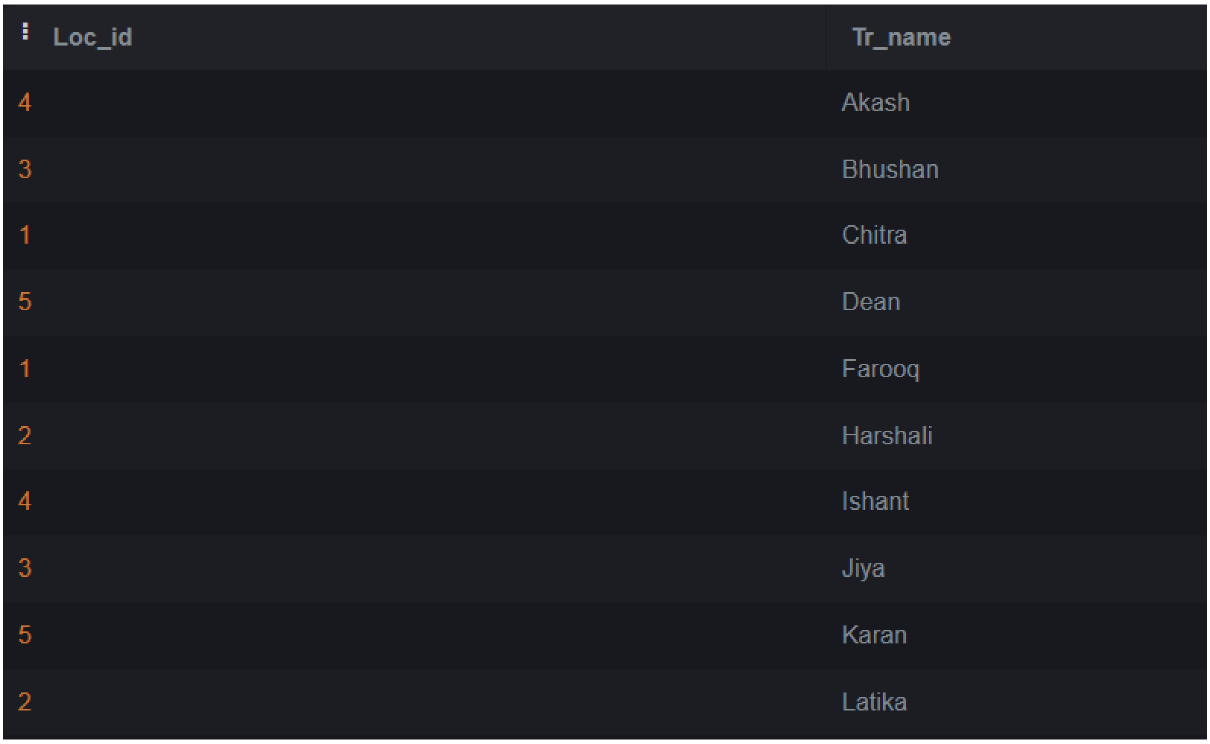
IN OPERATOR

```
SELECT * FROM Travellers

WHERE city IN (‘Ahemadabad’);
```

JOINS

INNER JOIN

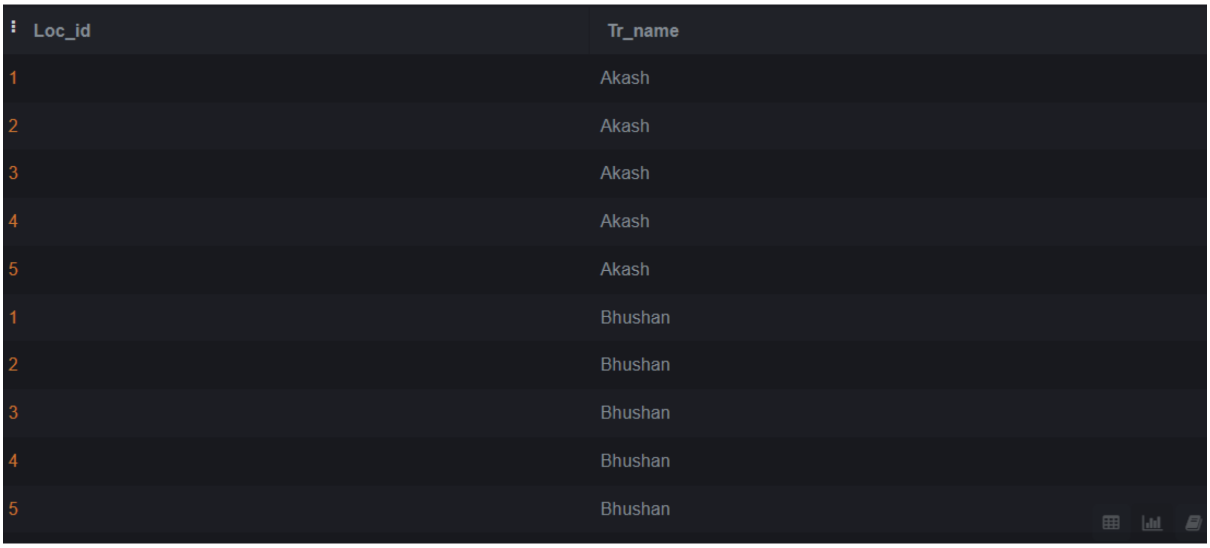
```
SELECT Travel_Locations.Loc_id,Travellers.Tr_name

FROM Travel_Locations INNER join Travellers

ON Travel_Locations.Loc_id=Travellers.loc_id;
```



Loc_id	Tr_name
4	Akash
3	Bhushan
1	Chitra
5	Dean
1	Farooq
2	Harshali
4	Ishant
3	Jiya
5	Karan
2	Latika

NON EQUI JOIN

```
SELECT Travel_Locations.Loc_id,Travellers.Tr_name
FROM Travel_Locations INNER join Travellers
ON Travel_Locations.Loc_id!=Travellers.Tr_name;
```



Loc_id	Tr_name
1	Akash
2	Akash
3	Akash
4	Akash
5	Akash
1	Bhushan
2	Bhushan
3	Bhushan
4	Bhushan
5	Bhushan

OUTER JOIN HAVE THREE PART

- **LEFT JOIN / LEFT OUTER JOIN** = return all row from the left table
- **RIGHT JOIN /RIGHT OUTER JOIN** = return all row from the right table
- **FULL JOIN / FULL OUTER JOIN =** return all row from the both table

1. **UNION =** this operator combine the result sets of multiple queries and remove duplicate row from the final result
   SELECT COLUMNS FROM TABLE 1 UNION SELECT COLUMNS FROM TABLE 2;
2. **UNION ALL =** operator, on the other hand, combines the result sets of multiple queries without removing duplicates.
   SELECT columns FROM table1 UNION ALL SELECT columns FROM table2;
3. Use **UPDATE** from modify existing records
   UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;

Constraints in SQL

● NOT NULL - Restricts NULL value from being inserted into a column.

● CHECK - Verifies that all values in a field satisfy a condition.

● DEFAULT - Automatically assigns a default value if no value has been specified for the field.

● UNIQUE - Ensures unique values to be inserted into the field.

● PRIMARY KEY - Uniquely identifies each record in a table.

● FOREIGN KEY - Ensures referential integrity for a record in another table.

delete and truncate

4. The `DELETE` command is used to remove specific rows from a table based on a specified condition

5. The `TRUNCATE` command is used to remove all rows from a table in a more efficient way compared to `DELETE`.