### DAY ☀️: \*\*Task – Normalization in SQL\*\*

```

Created by :

Name: Dhiraj Kr.

Profession: Data Scientist & GenAI Developer

```

---

### ✅ Unnormalized Table: `student\_records`

```sql

CREATE TABLE student\_records (

student\_id INT,

student\_name VARCHAR(50),

subjects VARCHAR(100),

pin\_code INT,

city\_name VARCHAR(50)

);

INSERT INTO student\_records VALUES

(1, 'Anjali', 'Math, Science', 110001, 'Delhi'),

(2, 'Rahul', 'English, History', 560001, 'Bangalore'),

(3, 'Priya', 'Math, English', 110001, 'Delhi');

```

---

### ⚡ Normalization Tasks

---

### ⚡ Task 1: Identify Violations in Unnormalized Table

\*\*Question:\*\* What problems exist in the `student\_records` table that violate 1NF?

The Subject Column Conain Two Value for One Unique Record < so For 1NF Lets Make the reord Unique first

---

### ⚡ Task 2: Convert to 1NF

\*\*Question:\*\* Break the `subjects` column into atomic values.

\*\*Table to Create:\*\* `student\_subjects(student\_id, student\_name, subject, pin\_code, city\_name)`

**Query : CREATE TABLE student\_records\_1NF (**

**student\_id INT,**

**student\_name VARCHAR(50),**

**subjects VARCHAR(100),**

**pin\_code INT,**

**city\_name VARCHAR(50)**

**);**

**INSERT INTO student\_records\_1NF VALUES**

**(1, 'Anjali', 'Math', 110001, 'Delhi'),**

**(1, 'Anjali', 'Science', 110001, 'Delhi'),**

**(2, 'Rahul', 'English', 560001, 'Bangalore'),**

**(2, 'Rahul', 'History', 560001, 'Bangalore'),**

**(3, 'Priya', 'Math', 110001, 'Delhi'),**

**(3, 'Priya', 'English', 110001, 'Delhi');**

---

### ⚡ Task 3: Convert to 2NF

\*\*Question:\*\* Create two separate tables:

- `students(student\_id, student\_name, pin\_code)`

- `subjects(student\_id, subject)`

Move data accordingly.

**Query:**

**CREATE TABLE students\_2NF (**

**student\_id INT,**

**student\_name VARCHAR(50),**

**pin\_code INT**

**);**

**INSERT INTO students\_2NF VALUES**

**(1, 'Anjali',110001),**

**(2, 'Rahul',560001),**

**(2, 'Rahul',560001),**

**(3, 'Priya',110001),**

**(3, 'Priya',110001);**

**CREATE TABLE subject\_2NF (**

**student\_id INT,**

**subjects VARCHAR(100)**

**);**

**INSERT INTO subject\_2NF VALUES**

**(1,'Math'),**

**(1,'Science'),**

**(2,'English'),**

**(2,'History'),**

**(3,'Math'),**

**(3,'English');**

---

### ⚡ Task 4: Identify Transitive Dependency

\*\*Question:\*\* In the `students` table, which column depends on a non-key column?

**THE PIN CODE and City Name**

---

### ⚡ Task 5: Apply 3NF

\*\*Question:\*\* Create a third table for city names:

- `cities(pin\_code, city\_name)`

Remove `city\_name` from `students`.

**Query CREATE TABLE students\_3NF (**

**student\_id INT,**

**student\_name VARCHAR(50),**

**pin\_code INT**

**);**

**INSERT INTO students\_3NF VALUES**

**(1, 'Anjali',110001),**

**(2, 'Rahul',560001),**

**(2, 'Rahul',560001),**

**(3, 'Priya',110001),**

**(3, 'Priya',110001);**

**CREATE TABLE subject\_3NF (**

**student\_id INT,**

**subjects VARCHAR(100)**

**);**

**INSERT INTO subject\_3NF VALUES**

**(1,'Math'),**

**(1,'Science'),**

**(2,'English'),**

**(2,'History'),**

**(3,'Math'),**

**(3,'English');**

**CREATE TABLE city\_name\_3NF(**

**pincode INT,**

**city\_name VARCHAR(100)**

**);**

**INSERT INTO city\_name\_3NF**

**VALUES**

**(110001, 'Delhi'),**

**(560001, 'Bangalore');**

---

### ⚡ Task 6: Insert Data into Normalized Tables

\*\*Question:\*\* Insert all data into the normalized `students`, `subjects`, and `cities` tables.

---

### ⚡ Task 7: Create Relationships

\*\*Question:\*\* Add primary and foreign key constraints in the 3NF structure.

**Qury:**

**#-----------------3NF----------------------------**

**CREATE TABLE students\_Normalized (**

**student\_id INT PRIMARY KEY,**

**student\_name VARCHAR(50),**

**pin\_code INT,**

**FOREIGN KEY (pin\_code) REFERENCES city\_name\_Normalized(pin\_code)**

**);**

**INSERT INTO students\_Normalized VALUES**

**(1, 'Anjali',110001),**

**(2, 'Rahul',560001),**

**(2, 'Rahul',560001),**

**(3, 'Priya',110001),**

**(3, 'Priya',110001);**

**CREATE TABLE subject\_Normalized (**

**student\_id INT,**

**subjects VARCHAR(100),**

**FOREIGN KEY (student\_id) REFERENCES students\_Normalized(student\_id)**

**);**

**INSERT INTO subject\_Normalized VALUES**

**(1,'Math'),**

**(1,'Science'),**

**(2,'English'),**

**(2,'History'),**

**(3,'Math'),**

**(3,'English');**

**CREATE TABLE city\_name\_Normalized(**

**pincode INT PRIMARY KEY,**

**city\_name VARCHAR(100)**

**);**

**INSERT INTO city\_name\_Normalized**

**VALUES**

**(110001, 'Delhi'),**

**(560001, 'Bangalore');**

---

### ⚡ Task 8: Retrieve Full Student Info

\*\*Question:\*\* Write a query to display student name, subject, and city using JOINs.

---

### ⚡ Task 9: Count Subjects per Student

\*\*Question:\*\* Write a query to display the number of subjects each student is enrolled in.

---

### ⚡ Task 10: Add a New Student

\*\*Question:\*\* Insert a new student `'Riya'` with `student\_id = 4`, subjects: `Math, Science`, pin: `500001`, city: `Hyderabad`.

Normalize and insert data correctly in all tables.

---

\*\*End of Document\*\* ✨