Welcome to the

Java Course

Module 4 – Day 04

Content of the course

- Introduction to Database Theory and SQL Basics
- Database connection with Java
- Table Management and Relationships
- Advanced SQL Queries
- **Data Normalization**
- Views, subqueries and nested queries

Event Manager Project - Step 3

- Create a Java project for the Event Manager
- Create Java classes for the tables you created in the database
- Create a main program that will allow the user to view all existing events

Data normalization

Data Normalization

Database normalization is the process of organizing data in a database to minimize redundancy and dependency by dividing large tables into smaller ones and defining relationships between them.

First normal form (1NF)

- A table is in 1NF if each column contains **atomic values** (indivisible values).

- It ensures that each cell in the table contains only a single value.

Example 1NF

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721, 9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 1

Conversion to first normal form

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721	HARYANA	
1	RAM	9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 2

Second normal form (2NF)

- A table is in 2NF if it is in 1NF and every non-key attribute is fully functionally **dependent** on the primary key.

- It eliminates partial dependencies where part of the primary key determines other attributes' values.

Example 2NF

(Employee_ID, Project_ID) forms the **composite primary key**. However, there's a partial dependency because **Employee_Name depends only on Employee_ID**, not the entire composite key. Similarly, Project_Name depends only on Project_ID, not the entire composite key.

Employee_ID	Project_ID	Employee_Name	Project_Name	Hours_Worked
1	101	Alice	Project A	20
1	102	Alice	Project B	30
2	101	Bob	Project A	25
2	103	Bob	Project C	40

Example 2NF

Employee_ID	Employee_Name
1	Alice
2	Bob

Project_ID	Project_Name
101	Project A
102	Project B

Employee_ID	Project_ID	Hours_Worked
1	101	20
1	102	30
2	101	25
2	103	40

Third normal form 3NF

- A table is in 3NF if it is in 2NF and has no transitive dependencies.

- It eliminates **transitive dependencies** where attributes don't depend on the key.

a *deduct b and b deduct* ***c****, then a deduct c*

Example 3NF

For each **employee_id**, we can deduct their **home_state**, but this is a transitive dependency. The name of the state (**home_state**) can be deducted from the **state_code**, and the **state_code** can be deducted from the **employee_id**.

`employees` **Table**

employee_id	name	state_code	home_state
E001	Alice	26	Michigan
E002	Bob	56	Wyoming
E003	Alice	56	Wyoming

Example 3NF

Instead, we separate the state data to its own table

`employees` **Table**

employee_id	name	state_code
E001	Alice	26
E002	Bob	56
E003	Alice	56

`states` **Table**

state_code	home_state
26	Michigan
56	Wyoming

Now YOUR TURN !

Let’s do the exercises

Event Manager Project - Step 4

- New feature: Allow the user to buy a ticket. The user should be requested:
  - his personal details.
  - which event he wants to attend
  - which ticket type he would like to buy

- Is your database in 3NF? No need to make the updates, just think about what would be needed to convert it to 3NF.