Date:- 30/9/25

Task-8: Normalizing Databases using Functional dependencies upto BCNF

Step-1:- Define the initial relational schema and functional dependencies.

StudentSlot (Student_ID, Student_Name, Email, Dept_ID, Dept_Name, Course_ID, Course_Name, Credits, Slot_ID, Slot_Type, Date, Venue)

1. Student_ID → Student_Name, Email, Dept_ID
2. Dept_ID → Dept_Name
3. Course_ID → Course_Name, Credits, Course_Type
4. Slot_ID → Slot_Type, Date, Venue
5. ~~Course~~ Student_ID, Slot_ID → Course_ID

Step-2:- Convert the relation to 1NF

* Identify and eliminate any repeating groups or arrays in the studentSlot relation
* Create separate tables if repeating groups exist

Step 3:- Convert to 2NF

* Ensure that each non-key attribute depends on the whole primary key.
* Move non-key attributes to separates relations if they depends on only part of the primary key

Proposed Decomposition

1. Student (Student_ID, Student_Name, Email, Dept_ID)
2. Department (Dept_ID, Dept_Name)
3. Course (Course_ID, Course_Name, Credits)
4. Slot (Slot_ID, Slot_Type, Date, Venue)
5. Student_slot_Course (Student_ID, Slot_ID, course_ID).

Step-4 - Convert to 3NF

* Remove transitive dependencies where a non-Key attribute depends on another non-Key attribute.

→ There is no transitive dependencies.

Step-5: Convert to BCNF

* Ensure every determinant is a candidate Key.
* Check for overlapping candidate Keys.
* Decompose relations to eliminate redundancy.

→ No decomposition needed.

Use Griffith Tool

1. Input relational schema and funtional dependencies
2. Griffith tool generates a dependency graph.
3. Analyze the graph to identify normal -igation issues.
4. Apply normalization rules to transform the schema
5. Verify the resulting schema meets BCNF criteria.

Griffith tool Steps

1. Create a new project in Griffith.
2. Define the relational schema and FDs.
3. Run the "Dependency Graph" tool.
4. Analyze the graph for normalization issues.
5. Apply transformations using the "normali -ze" tool.
6. Verify BCNF compliance using the "BCNF check" tool.

Normalized Schema

1. Student (Student_Id, Student_Name, Email)
2. Department (Dept_ID, Dept_Name)
3. Course (Course_ID, Course_Name, Credits)
4. Slot (Slot_ID, slot_Type, Date, Venue)
5. Student_slot_Course (Student_ID, slot_ID Course_ID).

Result: Thus performing normalizing database using functional dependencies has been executed & implemented successfully.

VEL TECH	
[illegible]	[illegible]
[illegible] ATTENDANCE (5)	5
[illegible] MODEL AND ANALYSIS (5)	5
[illegible] VIVA VOCE (5)	4
[illegible] RECORD (5)	4
TOTAL (20)	18
SIGN WITH DATE	17/10