TASK-8 Normalizing Database using Functional Dependencies upto BCNF. | 30/9/25

Tool: Griffith / ALM: Jigsaw

Step 1: Identify Student booking Attributes

we'll assume Student slot booking table like-terms:

Student_Booking (St_ID, St_name, Dept_id, Dept_name, C_Id, C_name, slot_Id, slot_time, Room_no, F_Id, F_name)

Each Student books 2 course slot handled by 2 faculty in 2 department.

Step 2: Define Functional dependence (FDs)

1) Student_id → Student_name, Dept_Id
(Each student belongs to one department)

2) Dept_Id → Dept_name.
(Each department has one name)

3) C_Id → C_name, Dept_id, F_Id
(Each course belongs to one department and has one personality)

4) F_Id → F_name (Each faculty has one name)

5) Slot_Id → Slot_time, Room no, C_Id.
(Each slot has time, room, is assigned to one course)

Step 3: Convert to INF (First Normal Form)

Rule: No repeating groups or multi-valued attributes.

Student-booking (Student-ID, St_name, dept-ID)

Dept_name (C-ID, C-Name, Slot-ID, Slot-Time, Room-no, Faculty-Id, F-name)

Step 4: Convert to 2NF.

Rule: Every Non-key attribute depend on whole primary key.

If we take (St-Id, Slot-ID) is composite primary key then some Non-key of Root key.

Non-key Attribute	Depends on	Action
Student_name, Dept_Id	St_ID	Move to Student
Dept-name	Dept_Id	Move to department
C-name, F_Id, Dept_Id	C_Id	Move to course
F_name	F_Id	Move to Faculty
Slot_time, Room_no, C_Id.	S_Id	Move to Slot.

Step 5: Convert to 3NF.

- Dept_Id -> Dept_name.
- F_Id -> F_Name.
- Course_Id -> F_Id and F_Id -> F_Name.
- Student_Id -> Dept_Id and Dept_Id -> Dept_name.

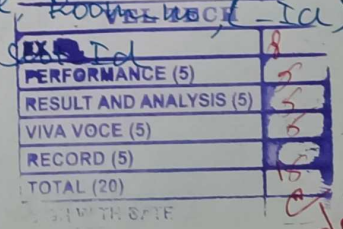
Step 6: Convert to BCNF.

For every dependency (X -> Y) (Candidate key)

Table	Functional dependencies	Candidate key	BCNF
Student	St_Id -> St Name, Dept-ID	Student_ID	✓
Department	Dept-ID -> Dept_name	Dept_Id	✓
Faculty	F_ID -> F_Name	Faculty-ID	✓
course	C_Id -> C_Name, F_Id Dept_Id.	course_ ID	✓
Slot	Slot-ID -> Slot_Time, Room-no, C_Id	Slot_Id	✓
Booking	(Student_Id, Slot_Id) -> C)	Student_Id, Slot_ID	✓

Step:7: Final normalised schema.

1) Student (Student_Id, St_name, Dept_ID)
2) Dept (Dept_Id, Dept_name)
3) Faculty (F_Id, F_name)
4) Course (C_Id, C_name, F_Id, Dept_Id)
5) Slot (Slot_Id, Slot_Time, Room_no, C_Id)
6) Booking (Student_Id, Slot_Id)



Result:

Thus the normalising database using functional dependencies upto BCNF has been completed successfully.