Student Activities Database Management System

Team Members

- Mridul Ravi Jain 110040083
- Abhishek Gupta 110040067
- Dhanesh Kumar 110050021
- Abhilash Gupta 110050058

Relational Schema

1) student (roll_no, student_name, phone_no, email_id, pasword)

   Functional Dependency: email_id →roll_no

2) hostel (hostel_no, room_no, roll_no, general_secretary_roll_no, cultural_councilor_roll_no, technical_councilor_roll_no, sports_councilor_roll_no, hostel_name, phone_no, cult_points, tech_points, sports_points)

   Deviation: Here we have merged the *resides_in* relationship (many to one from students to hostel) with hostel which should have been merged with students.

   Problems: The main problem associated with this is the redundancy in information which will further lead to insertion and update anomalies.

   Here hostel_no and roll_no are the primary keys, so if there are 100 students in a hostel then there will be 100 entries in the table and the information like general secretary etc. will be repeated 100 times. This will introduce redundancy in the schema.

   Foreign Key Constraints:

   roll_no REFERENCES student

   general_secretary_roll_no REFERENCES student

   cultural_councilor_roll_no REFERENCES student

   technical_councilor_roll_no REFERENCES student

   sports_councilor_roll_no REFERENCES student

   Functional Dependency:

   hostel_no →general_secretary_roll_no, cultural_councilor_roll_no, cultural_points, technical_points, technical_councilor_roll_no, sports_councilor_roll_no, sports_points, hostel_name, phone_no

   general_secretary_roll_no → hostel_no

   cultural_councilor_roll_no → hostel_no

   technical_councilor_roll_no → hostel_no

   sports_councilor_roll_no → hostel_no

3) club (club_name, genre, manager_roll_no, convener_roll_no)

Foreign Key Constraints:

manager_roll_no REFERENCES student

convener_roll_no REFERENCES student

Functional Dependency:

manager_roll_no → club_name

convener_roll_no → club_name

4) joins (roll_no, club_name)

Foreign Key Constraints:

roll_no REFERENCES student

club_name REFERENCES club

5) sports_activity (sports_activity_name, description, sports_institute_secretary_roll_no)

Foreign Key Constraints:

sports_institute_secretary_roll_no REFERENCES student

Functional Dependency:

sports_institute_secretary_roll_no → sports_activity_name

6) technical_activity (technical_activity_name, description, technical_institute_secretary_roll_no)

Foreign Key Constraints:

technical_institute_secretary_roll_no REFERENCES student

Functional Dependency:

technical_institute_secretary_roll_no → technical_activity_name

7) cultural_activity (cultural_activity_name, description, cultural_institute_secretary_roll_no)

Foreign Key Constraints:

cultural_institute_secretary_roll_no REFERENCES student

Functional Dependency:

cultural_institute_secretary_roll_no → cultural_activity_name

8) likes_cultural_activity (cultural_activity_name, roll_no)

Foreign Key Constraints:

roll_no REFERENCES student

cultural_activity_name REFERENCES cultural_activity

9) likes_technical_activity (technical_activity_name, roll_no)

Foreign Key Constraints:

roll_no REFERENCES student

technical_activity_name REFERENCES technical_activity

10) likes_sports_activity (sports_activity_name, roll_no)

Foreign Key Constraints:

roll_no REFERENCES student

sports_activity_name REFERENCES sports_activity

11) venue (venue_name, location, phone_no, capacity)

12) staff (staff_name, phone_no, email_id, venue_name)

Deviation: have merged the many-many relationship *manages* (from staff to venue) in staff.

Problems: There are 2 problems associated with this merging:-

a) If a staff member manages many venues, his details will be stored many times, which will cause redundancy as well as insertion and update anamolies.
b) Since venue_name is a part of the primary key, therefore it cannot take NULL values, hence if a staff member does manage any venue, then his details cannot be stored in staff relation.

Foreign Key Constraints:

venue_name REFERENCES venue

13) technical_club (technical_activity_name, club_name)

Foreign Key Constraints:

technical_activity_name REFERENCES technical_activity

club_name REFERENCES club

14) cultural_club (cultural_activity_name, club_name)

Foreign Key Constraints:

cultural_activity_name REFERENCES cultural_activity

club_name REFERENCES club

15) events (event_name, start_date, end_date, start_time, duration)

16) sports_events (sports_event_name, start_date)

Foreign Key Constraint: sports_event_name, start_date REFERENCES events

17) technical_events (technical _event_name, start_date)

Foreign Key Constraint: technical_event_name, start_date REFERENCES events

18) cultural_events (cultural _event_name, start_date)

Foreign Key Constraint: cultural _event_name, start_date REFERENCES events

19) academic_events (academic _event_name, start_date)

Foreign Key Constraint: academic _event_name, start_date REFERENCES events

20) academic_talk (academic_event_name, start_date, speaker)

Foreign Key Constraint: academic _event_name, start_date REFERENCES academic_events

21) hostel_position (event_name, start_date, winner, first_runnerup, second_runnerup, first_point, second_point, third_point)

Foreign Key Constraint: event_name, start_date REFERENCES events

winner REFERENCES hostel

first_runnerup REFERENCES hostel

second_runnerup REFERENCES hostel

22) happens_at (event_name, start_date, venue_name)

Foreign Key Constraint: event_name, start_date REFERENCES events

venue_name REFERENCES venue

23) organised_by (event_name, start_date, club_name)

Foreign Key Constraint: event_name, start_date REFERENCES events

club_name REFERENCES club

24) winner (event_name, start_date, roll_no, competition_name)

Foreign Key Constraint: event_name, start_date REFERENCES events

roll_no REFERENCES student

25) sports_type (sports_event_name, start_date, sports_activity_name)

Foreign Key Constraint: sports_event_name, start_date REFERENCES sports_events

sports_activity_name REFERENCES sports_activity

26) cultural_type (cultural_event_name, start_date, cultural_activity_name)

Foreign Key Constraint: cultural_event_name, start_date REFERENCES cultural_events

cultural_activity_name REFERENCES cultural _activity

27) technical_type (technical_event_name, start_date, technical_activity_name)

Foreign Key Constraint: technical_event_name, start_date REFERENCES technical_events

technical_activity_name REFERENCES technical_activity

Normalization

1) student(roll_no, student_name, phone_no, email_id, pasword):

Functional Dependency:

email_id →roll_no

email_id uniquely identifies a tuple in the relation and therefore is a superkey for the schema. Thus relation ***student*** **is in BCNF and also in 3NF.**

2) hostel (hostel_no, room_no, roll_no, general_secretary_roll_no, cultural_councilor_roll_no, technical_councilor_roll_no, sports_councilor_roll_no, hostel_name, phone_no, cult_points, tech_points, sports_points)

Functional Dependency:

hostel_no →general_secretary_roll_no, cultural_councilor_roll_no, cultural_points, technical_points, technical_councilor_roll_no, sports_councilor_roll_no, sports_points, hostel_name, phone_no

But in this functional dependency *hostel_no* is not a superkey (As two students with different roll nos can have the same hostel). Therefore, relation ***hostel*** **is not in BCNF**. Also, attributes such as *cult_points, tech_points, sports_points* are not contained in any candidate key for relation *hostel.* Hence, relation ***hostel*** **is not in 3NF.**
Normalization: Since this relation is not in BCNF, we normalize it by decomposing it into following two relations:

- **hostel** (hostel_no , general_secretary_roll_no, cultural_councilor_roll_no, cultural_points, technical_points, technical_councilor_roll_no, sports_councilor_roll_no, sports_points, hostel_name, phone_no)
- **resides_in** (hostel_no, roll_no,room_no)

Now both these relations are in BCNF.

3) club (club_name, genre, manager_roll_no, convener_roll_no)

Functional Dependency:

manager_roll_no → club_name

convener_roll_no → club_name

manager_roll_no is a superkey for the above relation. Similarly convener_roll_no is also a superkey. None of the functional dependencies violates the conditions for BCNF. Hence relation ***club*** **is in BCNF and in 3NF.**

4) joins (roll_no, club_name)

No functional dependencies. Hence relation ***joins*** **is in BCNF and 3NF.**

5) sports_activity (sports_activity_name, description, sports_institute_secretary_roll_no)

Functional Dependency:

sports_institute_secretary_roll_no → sports_activity_name

sports_institute_secretary_roll_no is a superkey for the above relation. It satisfies the condition for conditions for BCNF. Hence relation ***sports_activity*** **is in BCNF and in 3NF.**

6) technical_activity (technical_activity_name, description, technical_institute_secretary_roll_no)

Functional Dependency:

technical_institute_secretary_roll_no → technical_activity_name

technical_institute_secretary_roll_no is a superkey for the above relation. It satisfies the condition for conditions for BCNF. Hence relation ***technical_activity*** **is in BCNF and in 3NF.**

7) cultural_activity (cultural_activity_name, description, cultural_institute_secretary_roll_no)

Functional Dependency:

cultural_institute_secretary_roll_no → cultural_activity_name

cultural_institute_secretary_roll_no is a superkey for the above relation. It satisfies the condition for conditions for BCNF. Hence relation ***cultural_activity*** **is in BCNF and in 3NF.**

8) likes_cultural_activity (cultural_activity_name, roll_no)

No functional dependencies. Hence relation ***likes_cultural_activity*** **is in BCNF and 3NF.**

9) likes_technical_activity (technical_activity_name, roll_no)

No functional dependencies. Hence relation ***likes_ technical_activity*** **is in BCNF and 3NF.**

10) likes_sports_activity (sports_activity_name, roll_no)

No functional dependencies. Hence relation ***likes_ sports_activity*** **is in BCNF and 3NF.**

11) venue (venue_name, location, phone_no, capacity)

No functional dependencies here. Hence relation ***venue*** **is in BCNF and 3NF.**

12) staff (staff_name, phone_no, email_id, venue_name)

No functional dependencies here. Hence relation ***staff*** **is in BCNF and 3NF.** This relation was a result of merging a many-many relationship *manages* (from staff to venue) in staff.

There were 2 problems associated with this merging:-

a) If a staff member manages many venues, his details will be stored many times, which will cause redundancy as well as insertion and update anomalies.
b) Since venue_name is a part of the primary key, therefore it cannot take NULL values, hence if a staff member does manage any venue, then his details cannot be stored in staff relation.

We see that there is a multivalued functional dependency on staff:

staff_name →→ *venue_name*

The above functional dependency is a non-trivial multivalued functional dependency and *staff_name* is not a superkey. Hence, the relation schema *staff* with *venue_name* as a multivalued attribute is **not in 4NF**. So we normalize it to 4NF by decomposing it into:

- ***staff*** (staff_name, phone_no, email_id)
- ***manages*** (staff_name, venue_name)

Here relation manages has Foreign Key Constraints:

staff_name REFERENCES staff
venue_name REFERENCES venue

13) technical_club (technical_activity_name, club_name)

No functional dependencies. Hence relation ***likes_ sports_activity*** **is in BCNF and 3NF.**

14) cultural_club (cultural_activity_name, club_name)

No functional dependencies. Hence relation ***likes_ sports_activity*** **is in BCNF and 3NF.**

15) events (event_name, start_date, end_date, start_time, duration)

No functional dependencies. Hence relation ***sports_events*** **is in BCNF and 3NF.**

16) sports_events (sports_event_name, start_date)

No functional dependencies. Hence relation ***sports_events*** **is in BCNF and 3NF.**

17) technical_events (technical_event_name, start_date)

No functional dependencies. Hence relation ***technical_events*** **is in BCNF and 3NF.**

18) cultural_events (cultural_event_name, start_date)

No functional dependencies. Hence relation ***cultural_events*** **is in BCNF and 3NF.**

19) academic_events (<u>academic_event_name, start_date</u>)

No functional dependencies. Hence relation ***academic_events*** **is in BCNF and 3NF.**

20) academic_talk (<u>academic_event_name, start_date,</u> speaker)

No functional dependencies. Hence relation ***academic_talk*** **is in BCNF and 3NF.**

21) hostel_position (<u>event_name, start_date</u>, winner, first_runnerup, second_runnerup, first_point, second_point, third_point)

No functional dependencies. Hence relation ***hostel_position*** **is in BCNF and 3NF.**

22) happens_at (<u>event_name, start_date, venue_name</u>)

No functional dependencies. Hence relation ***happens_at*** **is in BCNF and 3NF.**

23) organised_by (<u>event_name, start_date, club_name</u>)

No functional dependencies. Hence relation ***organised_by*** **is in BCNF and 3NF.**

24) winner (<u>event_name, start_date, roll_no</u>, competition_name)

No functional dependencies. Hence relation ***winner*** **is in BCNF and 3NF.**

25) sports_type (<u>sports_event_name, start_date, sports_activity_name</u>)

No functional dependencies. Hence relation ***sports_type*** **is in BCNF and 3NF.**

26) cultural_type (<u>cultural_event_name, start_date, cultural_activity_name</u>)

No functional dependencies. Hence relation ***cultural_type*** **is in BCNF and 3NF.**

27) technical_type (<u>technical_event_name, start_date, technical_activity_name</u>)

No functional dependencies. Hence relation ***technical_type*** **is in BCNF and 3NF.**