Relational Schema:-

1. student( roll\_no, student\_name, phone\_no, email\_id, password)
2. hostel( hostel\_no, room\_no, roll\_no, general\_secretary\_roll\_no, cultural\_councilor\_roll\_no, technical\_councilor\_roll\_no, sports\_councilor\_roll\_no)

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 anamolies.

Here hostel\_no and room\_no are the primary keys, so if there are 100 rooms in a hostel then there will be 100 entries in the table and the information like general secretary etc. will be repeated 100 times.

Foreign Key Constraints:

roll\_no REFERENCES student

general\_secretary\_roll\_no REFERENCES student

cult\_councilor\_roll\_no REFERENCES student

tech\_councilor\_roll\_no REFERENCES student

sports\_councilor\_roll\_no REFERENCES student

1. club( club\_name, genre, manager\_roll\_no, convener\_roll\_no)

Foreign Key Constraints:

manager\_roll\_no REFERENCES student

convener\_roll\_no REFERENCES student

1. joins( roll\_no, club\_name)

Foreign Key Constraints:

roll\_no REFERENCES student

club\_name REFERENCES club

1. sports\_activity( sports\_activity\_name, description, sports\_institute\_secretary\_roll\_no)

Foreign Key Constraints:

sports\_institute\_secretary\_roll\_no REFERENCES student

1. technical\_activity( technical\_activity\_name, description, technical\_institute\_secretary\_roll\_no)

Foreign Key Constraints:

technical\_institute\_secretary\_roll\_no REFERENCES student

1. cultural\_activity( cultural\_activity\_name, description, cultural\_institute\_secretary\_roll\_no)

Foreign Key Constraints:

cultural\_institute\_secretary\_roll\_no REFERENCES student

1. likes\_cultural\_activity( cultural\_activity\_name, roll\_no)

Foreign Key Constraints:

roll\_no REFERENCES student

cultural\_activity\_name REFERENCES cultural\_activity

1. likes\_technical\_activity( technical\_activity\_name, roll\_no)

Foreign Key Constraints:-

roll\_no REFERENCES student

technical\_activity\_name REFERENCES technical\_activity

1. likes\_sports\_activity( sports\_activity\_name, roll\_no)

Foreign Key Constraints:

roll\_no REFERENCES student

sports\_activity\_name REFERENCES sports\_activity

1. venue( venue\_name, location, phone\_no, capacity)
2. 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:-

1. 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.
2. 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

1. technical\_club( technical\_activity\_name, club\_name)

Foreign Key Constraints:

Technical\_activity\_name REFERENCES technical\_activity

club\_name REFERENCES club

1. cultural\_club( cultural\_activity\_name, club\_name)

Foreign Key Constraints:

cultural\_activity\_name REFERENCES cultural\_activity

club\_name REFERENCES club

O/O conversion

1. sports\_type( sports\_event\_name, start\_date, sports\_activity\_name)

Foreign Key Constraints:

sports\_event\_name, start\_date REFERENCES sports\_events

sports\_activity\_name REFERENCES sports\_activity

1. technical\_type( technical\_event\_name, start\_date, technical\_activity\_name)

Foreign Key Constraints:

technical\_event\_name, start\_date REFERENCES technical\_events

technical\_activity\_name REFERENCES technical\_activity

1. cultural\_type( cultural\_event\_name, start\_date, cultural\_activity\_name)

Foreign Key Constraints:

cultural\_event\_name, start\_date REFERENCES cultural\_events

cultural\_activity\_name REFERENCES cultural\_activity