Part 1

Consider a relation schema R(A, B, C, D, E, G) and the following sets of functional dependencies
F1 = {ABC → D, BC → EA, BCE → G }
F2 = {A→B, C→AD, AE→CG, BC → C}, AE, CE
F3 = {AC → B, BC → D, BD → E, AE → G, ED → A, DA → C},

A. For each set F1, F2 and F3 :-
Work out whether relation R with the respective set is in BCNF and show how you reached the answer.

B. For each set F1, F2 and F3 :-
Find all candidate keys of R and show how you reached the answer.

C. For each set F1, F2 and F3 :-
Consider partitioning R into 3 sub relations R1{A,B,C} , R2{D,E,G} , R3{B,C,D}. Is this
R2R3 -> DEGCB
R1 -> BC

decomposition lossless? Explain your answer.

Part 2

Goal
In this problem, you are required to translate the given entity relationship model that is provided to you in the solution to Assignment 1 (which is on the website), into relations for the relational model and produce MySQL DDL statements to create corresponding tables.
Note: The provided ER diagram is one of the possible solutions for Assignment 1.

Description

A. Use the methods presented in class to translate the ER diagram into relations. (50 points)
   - Translate the ER design into a set of relations.
   - Describe the primary key, not NULL and inclusion dependency constraints needed for the relational schema to capture and enforce the semantics of the ER design.

     Example:
     Relation:

employee(id, depId, name, level, award)

Not Null Attributes:

All non-key attributes are not NULL.

Inclusion constraints:

employee(depId) ⊆ department(id)

Person (Person_ID,Name,DOB,Gender)
Employee(Person_ID, Schedule,Employee_Type,Salary per hour)
Member(Person_ID, Membership_ID)

key: person_id for all relations

IND: employee(person_id) person(person_id)
member(person_id) is subset of person(person_id)

since, employee subclass relationship is disjoint & total we can eliminate employee
Trainer(Person_ID, Credentials) -> eliminate employee and put all attributes of employee in Trainer
Desk_Employee(Person_ID) -> eliminate desk_employee and put all attributes of employee in Trainer

Same can be done with member (person ID or member id?)
University _Affiliate(person_id, meme_id,Department)
Family(person_id,mem_id,Credit_Card),
related(person_id_univ, person_id_family)

IND: related(person_id_univ) Univ_aff (person_id)
related(person_id_family) family (person_id)

related(memeber_id_uni_aff,member_id_family)
key -> mem_id_family
IND: related(mem_id_uni_aff) is subset of university_aff(mem_id)
related(mem_id_family) is subset of family(mem_id)
family(mem_id) is subset of related(mem_id_family)

Same thing with University affiliate (total disjoint)
Non-Student(member_id, Member_Type,Credit_Card)
Student(member_id, Student_Type) -> key: member_id

IND: Non-Student(member_id) Member(member_id)

Student(member_id) Member(member_id)

Employee_Exit_Log(person_id,timestamp)
Entry_Log(person_id, Timestamp)
key : person_id, timestamp
IND: Employee_exit_log(person_id) person(person_id)
entry_log(person_id) is subset of person[person_id]

Events(Event_ID,Description,Start_time,End_time,Capacity)
key: event_id

Space(Space_ID, Description,Max_Capacity)
key: space_id

Equipment(Equipment_ID , Equipment_Type,Is_Available)
key_equipment_id

Hosted_in(event_id, space_id)
key -> event_id
Hosted_in(event_id) Events(event_id)
Hosted_in(space_id) Space(space_id)
Events(event_id) Hosted_in(event_id)

Contains(space_id, equipment_id)
key -> space_id
Contains(space_id) space(space_id)
Contains(equipment_id) Equipment(equipment_id)
Equipment(equipment_id) Contains(equipment_id)

Attends(membership_id, event_id)
Attends(membership_id) Member(membership_id)
Attends(event_id) Events(event_id)

Location Sensor(Sensor_ID,Coverage)
Equipment Sensor(Sensor_ID,Coverage)

Usage_Reading(equipment_id, membership_id, equipment_sensor_id, Timestamp)

IND:
Usage_Reading(equipment_id) Equipment(equipment_id)
Usage_Reading(membership_id) Member(membership_id)
Usage_Reading(equipment_sensor_id) Equipment_sensor(equipment_sensor_id)

Location_Reading(space_id, person_id, location_sensor_id, Timestamp)
key: space_id, person_id, location_sensor_id

IND:
Location_Reading(space_id) Space(space_id)
Location_Reading(person_id) Person(person_id)
Location_Reading(location_sensor_id) Locationt_sensor(Location_sensor_id)

B. Write SQL DDL statements for creating the tables corresponding to the relations you developed. Pick suitable data types for each attribute. Also include the appropriate referential integrity constraints and "NOT NULL" constraint while creating the tables. Execute your DDL statements on MySQL and make sure that all the statements execute without any error (we will be executing the DDL statements in the same order as in your submission).
(50 points)

Example:
CREATE TABLE Employee(
id INTEGER NOT NULL,
name VARCHAR(40) NOT NULL,

```
        depId VARCHAR(10) NOT NULL,
        award VARCHAR(40),
        PRIMARY KEY (id),
        FOREIGN KEY (depId) REFERENCES Department(depID)
);
```