Time: 1:30hrs   **Marks: 40**

Instructions:
- All queries must be written using MySQL.
- Ensure to use correct syntax and test your queries before submission.
- Marks are distributed as per the sub-questions.
- Mention the table structures, relationships, and logic in your queries.

Section A: (10 Marks)
1. (4 Marks)
   Create at least three tables (Pets, Trainers, TrainingSessions) and insert a minimum of five records into each table. Define the column specifications appropriately.

Pets Table:
- o PetID (INT, PRIMARY KEY, AUTO_INCREMENT)
- o PetName (VARCHAR(50), NOT NULL)
- o PetType (VARCHAR(50), NOT NULL)
- o OwnerName (VARCHAR(100), NOT NULL)

Trainers Table:
- o TrainerID (INT, PRIMARY KEY, AUTO_INCREMENT)
- o TrainerName (VARCHAR(100), NOT NULL)
- o Specialization (VARCHAR(50), NOT NULL)

TrainingSessions Table:
- o SessionID (INT, PRIMARY KEY, AUTO_INCREMENT)
- o PetID (INT, FOREIGN KEY REFERENCES Pets(PetID))
- o TrainerID (INT, FOREIGN KEY REFERENCES Trainers(TrainerID))
- o SessionDate (DATE, NOT NULL)

2. (2 Marks)
   Write SQL queries to create foreign key relationships between the Pets, Trainers, and TrainingSessions tables.
3. (2 Marks)
   Write a query to fetch all pets who are either dogs or have been trained by a trainer specialized in 'obedience training'.
4. (2 Marks)
   Write a query to fetch the first 5 records from the Pets table.

Section B: (15 Marks)
5. (2 Marks)
   Write a query to display the number of pets per trainer.

6. (2 Marks)
   Write a query to display the TrainerName and SessionDate of the most expensive training session.
7. (3 Marks)
   Write a query to create a view named TrainingSummary that shows the PetName, TrainerName, and SessionDate. Update the session date for a pet through this view and ensure the change reflects in the base table.
8. (2 Marks)
   Write a query to perform a LEFT JOIN between Pets and TrainingSessions, retrieving all pets and their respective training sessions, even if some pets haven't attended any sessions.
9. (2 Marks)
   Write a query to create a temporary table showing training sessions of pets with the type 'Dog'. Drop the temporary table after use.
10. (4 Marks)
    Write a stored procedure named GetTrainerDetails that takes a TrainerID as input and returns the TrainerName and the number of sessions they have conducted.

Section C: (15 Marks)
11. (4 Marks)
    Write a MySQL program using a WHILE loop to track the weight progress of pets after every training session.
12. (4 Marks)
    Write a stored function named SESSION_COST that takes the SessionID as input and calculates the total cost of the training session based on the number of hours.
13. (2 Marks)
    Write a query to find trainers who have conducted more than 5 sessions using the HAVING clause.
14. (2 Marks)
    Write a trigger that logs any updates made to the session date in a SessionLog table whenever a session is updated.
15. (3 Marks)
    Write a subquery to find the TrainerName and SessionDate of the second most expensive training session in the TrainingSessions table.