SQL Questions

Part –A-

Employee(EmpId, Lname, Fname, PosId, Supervisor, HireDate, Salary, Commission, DeptId, QualId)
PosId: REFERENCES Position (PositionId)
DeptId: REFERENCES Dept (DeptId)
QualId: REFERENCES Qualification (QualId)
Supervisor: REFERENCES Employee(EmpId)
Position(PositionId,PosDesc)
Qualification(QualId, QualDesc)
Dept(DeptId, DeptName, Location, EmpId)
EmpId: REFERENCES employee (EmpId)
Dependent(EmpId, DependentId, DepDOB, Relation)
EmpId: REFERENCES Employee (EmpId)

a) Display all employee names and their department names.

b) Find all employees' full names (lastname, firstname format) with salary, and their supervisor's name with salary.

c) Find number of employees for each department and position. Display position description, department name and number of employees in the output.

d) Display each employee's name, department name, position description, and qualification description.

e) Display employee names and dependent information. Include those employees who has no dependent.

f) Find out the names and number of years worked along with their department names in descending order by number of years worked.

g) Who works in the same department in which John Smith works? Display first and last names of the employees in the output. John Smith should not appear in the output.

h) Find name of the supervisor for employee with lname 'SHAH'.

i) Find all employees in the SALES department. Display first name, last name and department name in the output.

j) Display departments (name) that their employees' have average salary less than the average salary of department 100.

k) Display the position (description) of the employee(s) with the highest average salary.

Part -B-

Movie (movieId, movieName, categoryId, duration, productionDate)
Theater(theaterId,theaterName, city, ticketPrice)
Category (categoryId, categoryName)
ShownAt(theaterId, movieId, startDate, endDate)

a) Find the name of all theaters that have shown a movie called 'Harry Potter'.

b) Find the name of the shortest movie. (i.e. Find the name of the movie with the minimum duration)

c) For each category, find the number of movies that have a duration greater than the average duration of all movies. List the categoryId and the number you calculated.

d) Find the name of the oldest (the movie with the smallest ProductionDate) and the latest (the movie with the largest ProductionDate) movie in the database.

e) Find the number of the movies that have been shown at the theater with the cheapest ticketPrice.

f) List movieName and categoryName for all movies produced in 2002. (Please remember that the ProductionDate is a date.)

g) List MovieName of all movies that were shown for longer than 30 days.

h) Find the number of different categories in the movie table.

i) What is the sql*plus command that is used to show the structure of the theater table.

j) Assume that Movie, Theater and Category tables are already created. Please create the ShownAt table.

- TheaterId must be number with 4 digits Foreign Key referencing the Theater Table,
- MovieId must be number 6 digits Foreign Key referencing the Theater Table,
- StartDate and EndDate must be date data type.
- Don't forget to declare the Primary Key.

k) Insert a new tuple into the Movie table with the following values:

- MovieId: *101*
- MovieName: 'Honey', CategoryId: 12,
- Duration: 100, ProductionDate: '02-JUN-2003'

l) Change the name of the category 302 with Drama.

m) Delete rows of the Theater table that has the minimum ticketPrice.

n) Delete all rows that have longest duration in the Movie table.

o) Increase the ticketprice %10 of the Theater that is showing the movie called 'Lord of the Rings'

Part -C-

Student (student_id, fname, lname, dob, city, cgpa, deptno)
Course (ccode, cname, noofcredits, deptno)
StudentCourse (student_id, ccode, grade)
Department (deptno, dname)

a) Assume that Student, Course and Department tables are already created.
Create '**StudentCourse**' table using the following datatypes.
Student_id must be number with 6 digits,
Ccode must be variable length string (varchar2) with at most 7 characters and
Grade must be variable length string with at most 2 characters.
All Primary Key and Foreign Keys must be declared clearly.

b) Insert the following row into Course table.
Ccode= 'IT451' , cname= 'COMPLETE DATABASE', noofcredits=3, deptno=35

c) Update Student table and change the students' cgpa from 2.00 to 2.50 whose department name is CMPE.

d) Delete the Course that has minimum noofcredits in ECONOMICS department.

e) List number of students who are taking a course that has the maximum noofcredits.

f) Find the department name that the students have cgpa greater than the average cgpa

g) Find the number of students for each grade that they are taking DBMS(cname).

h) Find the number of students for each department that they have grade 'B' in MATH211.

i) Find the number of students that they have born in January.

j) Find the name of the course that has maximum noofcredits.

Normalization Questions

1. Normalize the following table. Show all work and clearly indicate the primary and foreign keys.

R(elevator_no,building_no,building_name,capacity, staff_no, first_name, last_name, date_examined)

Functional Dependencies:

1. elevator_no → building_no,capacity
2. building_no→ building_name
3. staff_no→ first_name,last_name
4. elevator_no,staff_no→date_examined

Normalize table R up to BCNF.

2. Normalize the following table. Clearly indicate primary keys and foreign keys.

R(project_no, project_title, start_date, manager_no, manager_name, manager_salary, employee_no, employee_name, employee_salary, no_of_hours)

Functional Dependencies:

1- project_no→ project_title, start_date, manager_no
2- employee_no→ employee_name, employee_salary
3- manager_no → manager_name, manager_salary
4- project_no, employee_no→ no_of_hours

Normalize table R up to BCNF.

3. *Normalize table T given below up to 3NF (Third Normal Form).* Show your steps and indicate primary and foreign keys clearly.

T (A, B, C, D, E, F, G)

Functional Dependencies:

1. *C→D*
2. *D→E, F*

4. *Normalize table R given below up to BCNF (Boyce-Codd Normal Form).* Show your steps and indicate primary and foreign keys clearly.

R (A, B, C, D, E, F)

Functional Dependencies:

1. *A→ C*
2. *C→D*
3. *E, F →B*