

1. **MCA-T103 ( Database Management System Practical ) Internal Examination 2022 April 28, 2022**  
**M.M. : 15 (Questions)+ 5M(Attendance regularity/ Response in class / Assignments) Duration 1:30 Hr**  
 Consider the below two tables for reference while trying to solve the SQL queries for practice.

Table – EmployeeDetails

EmpId	FullName	ManagerId	DateOfJoining	City
111	Amit Sharma	231	01/31/2014	Bangalore
333	Rajesh Vyas	654	01/30/2020	Mumbai
555	Kuldeep Tondon	543	27/11/2016	New Delhi

Table – EmployeeSalary

EmpId	Project	Salary	Variable
111	P1	8000	500
333	P2	10000	1000
555	P1	12000	0

1. SQL Query to fetch records that are present in one table but not in another table.
2. SQL query to fetch all the employees who are not working on any project.
3. SQL query to fetch all the Employees from EmployeeDetails who joined in the Year 2020.
4. Write an SQL query to fetch records from EmployeeDetails where city ends with character 'i'.
5. Write an SQL query to fetch only odd rows from the table.
6. Write SQL query to find the 3rd highest salary from a table without using the TOP/limit keyword.
7. Write an SQL query to fetch all those employees who work on Project other than P1.
8. Write an SQL query to fetch all the EmpIds which are present in either of the tables – 'EmployeeDetails' and 'EmployeeSalary'.
9. Write an SQL query to display both the EmpId and ManagerId together.
10. Write an SQL query to fetch project-wise count of employees sorted by
  - a. project's count in descending order.

2.

**MCA-T103 ( Database Management System Practical ) Internal Examination 2022 April 28, 2022**  
**M.M. : 15 (Questions)+ 5M(Attendance regularity/ Response in class / Assignments) Duration 1:30 Hr**

University Database

classroom(building, room number, capacity)

department(dept name, building, budget)

course(course id, title, dept name, credits)

instructor(ID, name, dept name, salary)

section(course id, sec id, semester, year, building, room number, time slot id)

teaches(ID, course id, sec id, semester, year)

student(ID, name, dept name, tot cred)

takes(ID, course id, sec id, semester, year, grade)

advisor(sID, i ID)

time slot(time slot id, day, start time, end time)

prereq(course id, prereq id)

3.2

Suppose you are given a relation grade points(grade, points) that provides a conversion from letter grades in the takes relation to numeric scores; for example, an "A" grade could be specified to correspond to 4 points, an "A-" to 3.7 points, a "B+" to 3.3 points, a "B" to 3 points, and so on. The grade points earned by a student for a course or section is defined as the number of credits for the course multiplied by the numeric points for the grade that the student received.

Write queries in SQL. Assumption that some grades might be null.

- a. Find the total grade points earned by the student with ID '12345', across all courses taken by the student.
- b. Find the grade point average (GPA) for the above student, that is, the total grade points divided by the total credits for the associated courses.
- c. Find the ID and the grade-point average of each student.



M.M. : 15 (Questions)+ 5M(Attendance regularity/ Response in class / Assignments) Duration 1:30 Hr

Figure 3.18 Banking database.

branch(branch name, branch city, assets)customer (ID, customer name, customer street, customer city)loan (loan number, branch name, amount)borrower (ID, loan number)account (account number, branch name, balance )depositor (ID, account number)

(g) Maximum loan giving branch with its customer names which has taken loan deposit in this branch

Consider the bank database of Figure 3.18, where the primary keys are underlined. Construct the following SQL queries for this relational database.

- Find the ID of each customer of the bank who has an account but not a loan.
- Find the ID of each customer who lives on the same street and in the same city as customer '12345'.
- Find the name of each branch that has at least one customer who has an account in the bank and who lives in

(d) Name of customer that is deposit 4600000 "Harrison".

4. MCA-T103 ( Database Management System Practical ) Internal Examination 2022 April 28, 2022

M.M. : 15 (Questions)+ 5M(Attendance regularity/ Response in class / Assignments) Duration 1:30 Hr

University Database

classroom(building, room number, capacity)department(dept name, building, budget)course(course id, title, dept name, credits)instructor(ID, name, dept name, salary)section(course id, sec id, semester, year, building, room number, time slot id)teaches(ID, course id, sec id, semester, year)student(ID, name, dept name, tot cred)takes(ID, course id, sec id, semester, year, grade)advisor(sID, i ID)time slot(time slot id, day, start time, end time)prereq(course id, prereq id)

Write the following inserts, deletes, or updates in SQL, using the university schema.

- Increase the salary of instructor in the Comp. Sci. department where salary is less than 5000 rs by 10%.
- Delete all courses that have never been offered (i.e., do not occur in the section relation).
- Insert every student whose total credit attribute is greater than 100 as an instructor in the same department, with a salary of \$10,000.

5. MCA-T103 ( Database Management System Practical ) Internal Examination 2022 April 28, 2022

M.M. : 15 (Questions)+ 5M(Attendance regularity/ Response in class / Assignments) Duration 1:30 Hr

Product:

Product ID	Product Name	Product Category
12	Bike ABC	Road Bike
13	Bike DEF	Mountain Bike
14	Bike GHI	Road Bike
15	Bike JKL	Touring Bike

Sales:

Product ID	Customer	Sales Amount
12	Joe	1000
13	Tom	2000
14	Joe	1500
12	Bill	1000

- Write a SQL statement that returns the distinct list of product categories from the Product table.
- Write a SQL statement that returns the total record count from the Sales table.
- Write a SQL statement that returns the Sum of Sales Amount grouped by Product Category having sales greater than 1500
- Write a SQL Statement that returns a list of products that do not appear the Sales table

c' Less than 1500



**University Database**

classroom(building, room number, capacity)  
 department(dept name, building, budget)  
 course(course id, title, dept name, credits)  
 instructor(ID, name, dept name, salary)  
 section(course id, sec id, semester, year, building, room number, time slot id)  
 teaches(ID, course id, sec id, semester, year)  
 student(ID, name, dept name, tot cred)  
 takes(ID, course id, sec id, semester, year, grade)  
 advisor(sID, i ID)  
 time slot(time slot id, day, start time, end time)  
 prereq(course id, prereq id)

3.1 Write the following queries in SQL, using the university schema.

- Find the titles of courses in the Comp. Sci. department that have 3 credits.
- Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.
- Find the highest salary of any instructor.
- Find all instructors earning the highest salary (there may be more than one with the same salary).
- Find the enrollment of each section that was offered in Fall 2017.
- Find the maximum enrollment, across all sections, in Fall 2017.
- Find the sections that had the maximum enrollment in Fall 2017

7.

**University Database**

classroom(building, room number, capacity)  
 department(dept name, building, budget)  
 course(course id, title, dept name, credits)  
 instructor(ID, name, dept name, salary)  
 section(course id, sec id, semester, year, building, room number, time slot id)  
 teaches(ID, course id, sec id, semester, year)  
 student(ID, name, dept name, tot cred)  
 takes(ID, course id, sec id, semester, year, grade)  
 advisor(sID, i ID)  
 time slot(time slot id, day, start time, end time)  
 prereq(course id, prereq id)

Write the following inserts, deletes, or updates in SQL, using the university schema.

- Increase the salary of each instructor in the Comp. Sci. department by 10%.
- Delete all courses that have never been offered (i.e., do not occur in the section relation).
- Insert every student whose tot cred attribute is greater than 100 as an instructor in the same department, with a salary of \$10,000.

8.

**Figure 3.17                  Insurance database**

person (driver id, name, address)  
 car (license plate, model, year)  
 accident (report number, year, location)  
 owns (driver id, license plate)  
 participated (report number, license plate, driver id, damage amount)

Consider the insurance database of Figure 3.17, where the primary keys are underlined. Construct the



- Find the total number of people who owned cars that were involved in accidents in 2017.
- Delete all year-2010 cars belonging to the person whose ID is '12345'.

**M.M. : 15 (Questions)+ 5M(Attendance regularity/ Response in class / Assignments) Duration 1:30 Hr**

employee (ID, person name, street, city)

works (ID, company name, salary)

company (company name, city)

manages (ID, manager id)

Consider the relational database of Figure 3.19, where the primary keys are underlined. Give an expression in SQL for each of the following queries.

- Find the ID, name, and city of residence of each employee who works for "First Bank Corporation".

b. Find the ID, name, and city of residence of each employee who works for "First Bank Corporation" and earns more than \$10000.

d. Find the ID of each employee who does not work for "First Bank Corporation".

d. Find the ID of each employee who earns more than every employee of "Small Bank Corporation".

e. Assume that companies may be located in several cities. Find the ID of each employee who works for a company located in a city that is not the headquarters of the company.

e. Assume that companies may be located in several cities. Find the name of each company that is located in every city in which "Small Bank Corporation" is located.

f. Find the name of the company that has the most employees (or companies, in the case where there is a tie for the most).

g. Find the name of each company whose employees earn a higher salary, on average, than the average salary at "First Bank Corporation".

1	classroom( <u>building</u> , <u>room number</u> , capacity)
---	---

department(dept name, building, budget)

course(course\_id, title, dept name, credits)

instructor(ID, name, dept name, salary)

teaches(ID, course id, sec id, semester, year, building, room number, time slot id)

teaches(ID, course id, sec id, semester, year)

student(ID, name, dept name, tot cred)

takes(ID, course id, sec id, semester, year, grade)

advisor(sID, i ID)

time slot(time slot id, day, start time, end time)

```
prereq(course id, prereq id)
```

