

DBMS – ASSIGNMENT 3 (LAB TEST)

M.Tech. (CS), First Year, 2025–2026

Date: April 17, 2026

Total: 5 marks

SUBMISSION INSTRUCTIONS

1. Submit all the solutions in a single file.
2. Naming convention for your submission file (assuming CS25xx is your roll number): CS25xx-assign3 (.rtf, .docx, .doc, .pdf, .tex, etc.).
3. To submit a solution file (say CS25xx-assign3.pdf), ensure that it is not password protected and mail to <assignisik@gmail.com> with the subject line as follows: DBMS M.Tech. (CS) 2025-26 CS25xx Assignment 3.

NOTE: All the solutions of this Lab Test must only include SQL queries. They should be written in a generalized way (as applicable). The time and space efficient queries will receive better marks.

Q1. Consider the following schema representing the login details of employees in a company.

- LOGIN = $\langle \text{userid} : \text{integer}, \text{username} : \text{string}, \text{logindate} : \text{date} \rangle$

Note that there is no primary key in the aforementioned relation. Write the following queries in SQL.

- (i) Find the name of the user who logged in last.
- (ii) Find the names of the users who logged in exactly twice at least on a single day.
- (iii) Find the names of the users who logged in for two or more consecutive days.

Q2. Consider the following schema that summarizes the company details found in a placement brochure of an institute.

- PLACEMENT = $\langle \underline{\text{companyid}} : \text{integer}, \text{companyname} : \text{string}, \text{email} : \text{string} \rangle$

The primary key corresponding to the aforementioned relation is underlined. Write the following queries in SQL.

- (i) Find the names of all the companies who do not have a valid email ID (having a local part and domain part before and after '@', respectively).
- (ii) Find the names of all the companies who have multiple email IDs included with comma separation.
- (iii) Find all the duplicate email IDs present.

Q3. Consider the following schema that denotes details of students who are involved in a group study helping each other as instructors. Note that students who do not have any instructor are included with the instructor id as *null*.

- STUDY = $\langle \underline{studentid} : integer, name : string, age : integer, instructorid : integer \rangle$

The primary key corresponding to the aforementioned relation is underlined. Write the following queries in SQL.

- Find the names of the students whose surnames start with a vowel.
- Find the names of the students who have the second highest age.
- Find the names of the students who are older than their instructors.

Q4. Consider the following schema that represents the American employees who have worked for the companies Facebook and X till date. Note that every American has a unique social security number denoted by ssn below.

- FACEBOOK = $\langle \underline{employeeid} : integer, ssn : integer, position : string, salary : real \rangle$
- X = $\langle \underline{employeeid} : integer, ssn : integer, position : string, salary : real \rangle$

The primary keys corresponding to the aforementioned relations are underlined. Write the following queries in SQL.

- Find the social security numbers of all the employees who worked in both Facebook and X.
- Find the social security numbers of all the employees who worked in multiple positions for Facebook but never worked in X.

Q5. Consider the following schema that represents the list of natural numbers in ascending order. Note that some of the numbers might have missed out from the table.

- NATURAL = $\langle number : integer \rangle$

Note that there is no primary key in the aforementioned relation. Write the following queries in SQL.

- Find the maximum number present.
- Find the median of all the numbers present.
- Find the missing numbers in the sequence.

Q6. Create a trigger to set control before the insert operations on a table. In case an insert operation happens on it, a separate log table will keep a record of the inserted tuple along with the time of insertion.

Q7. Show the usage of a cursor to collect a list of names, which are separated by comma, from a table with the following schema.

- UNIVERSITY = $\langle \underline{studentid} : integer, studentname : string, address : string \rangle$