

D B M S LAB
DBMS LAB
THIRD ASSIGNMENT ON DESIGN and SQL[CO1]

1. In an organization, number of departments exists. Each department has a name & unique code. Number of employees work in each department. Each employee has unique employee code. Detailed information like name, address, city, basic, date of join are also stored. In a leave register for each employee leave records are kept showing leave type (CL/EL/ML etc.), from-date, to-date. When an employee retires or resigns then all the leave information pertaining to him are also deleted. Basic salary must be within Rs.5000 to Rs.9000. A department can not be deleted if any employee record refers to it. Valid grades are A/B/C. Employee name must be in uppercase only. Default value for joining date is system date. Design & implement the tables with necessary constraints to support the scenario depicted above.

2. Try to violate the constraints that you have implemented in the table & note, what happens. [Try with suitable INSERT/UPDATE/DELETE instruction]

3. a) Create a table having empcode , Name, deptname, & basic From the existing tables along with the records of the employee who are in a particular department (say, d1) and with a basic Rs. 70000/-

b) From the existing table, add the employees with the basic salary greater than or equal to 7000/-

c) Alter the table to add a net pay column.

d) Replace net pay with $1.5 \times \text{Basic}$.

e) Try to remove the net pay column.

[It may require no. of steps]

4. In a library, for each book book-id, serial number (denotes copy number of a book), title, author, publisher and price are stored. Book-id and serial number together will be unique identifier for a book. Members are either student or faculty. Each member has unique member-id. Name, e-mail, address are also to be stored. Maximum number of books that a member can retain depends on member type. There may be other such parameters that depend on member type. Design should be flexible. For any transaction (book issue or return), members are supposed to place transactions slip. Each Transaction will have a unique id. User will submit member-id, book-id, and serial number (only for book return). Design and create the tables to store the book, member and transaction information. When a book is issued to a member a field like, To_Be_Returned_By has to be set as $\text{DT_Issue} + 7 \text{ days}$. At the time of book return, DT_Return will store the actual return date. While new book arrives, serial number will be last serial number for the Book-id +1. System should also keep track of the status of each physical book -- whether issued or available.

Design and create the tables. Populate the database maintaining the logic of transaction. May be issue/return will need update in multiple tables.

Write down the queries for the following:

a) Display total number of copies (irrespective of issued or not) for each book in the library and number of such copies issued

b) Find the members holding the books even after due date

c) Find the transaction details for delayed book returns and delay in terms of number of days.

d) Find the student members not making any transaction and do the same for faculty members.

e) Find the count of issue for each book (not the specific copy).