DBMS LAB **EXERCISE 1** Date:03/08/2023

Q1. Read about the MySQL datatype and get familiar with how to use it.

Q2. Use the DDL commands performs the following operation:

i) Create a table called EMP with the following structure.

NAME	TYPE
EMPNO	NUMBER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
DEPTNO	NUMBER(3)
SAL	NUMBER(7,2)

Allow NULL for all columns except ename and job.

ii) Add a column experience to the EMP table. Experience numeric null allowed.
iii) Modify the column width of the job field of EMP table.
iv) Create dept table with the following structure.

Name	Type
DEPTNO	NUMBER(2)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

v). drop a column experience from the EMP table.

Q3. Use the DML commands performs the following operation:
i) Insert a single record into dept table.

ii) Insert more than a record into EMP table using a single insert command.

iii) Select employee name, job from the emp table

Q4. Use the DDL commands to i). Truncate the EMP table and drop the dept table.

Q5. Use the DCL commands to perform the following operation
i. Create a new user ‘dbuser’ on the localhost

ii. Create a new database mysampldb and use that database for the following exercises.

iii. Grant all privileges for the dbuser on the mysampledb

Q6. Use the DCL command to revoke privilege to the user. i) Create a new user ‘dbuser1’ on the localhost ii) Grant only select privileges for the dbuser1 on the EMP table iii) Revoke the select privileges for the dbuser1 on the EMP table.

EXERCISE 2

Date: 10/08/2023

1. The following are maintained by a book dealer.

AUTHOR(author_id:int , name:string , city:string , country:string)

PUBLISHER(publisher_id:int , name:string , city:string , country:string)

CATALOG(book_id:int , title:string , author_id:int , publisher_id:int , category_id:int , year:int , price:int)

CATEGORY(category_id:int , description:string)

ORDER_DETAILS(order_no:int , book_id:int , quantity:int)

i) Create the above tables by properly specifying the primary keys.

ii) Enter at least five tuples for each relation.

iii) Find the total number of authors present in author relation.

iv) Find the book which has maximum sale.

2. Consider the following table "Book":

Acc-no	Yr_pub	title
734216	1982	Algorithm design
237235	1995	Database systems
631523	1992	Compiler design
543211	1991	programming
376112	1992	Machine design

i) Select from the relation “Book” all the books whose year of publication is 1992.

ii) Select from the relation “Book” all the books whose Acc-no is greater than equal to 56782.

iii) List all the Title and Acc-no of the “Book” relation.

iv) Using ‘Rename operator’ to rename the ‘Acc-no’ and ‘Yr_pub’ into a ‘SERIAL NO’ and

‘YEAR’ in the ‘’Book’’ relation.

3. branch (branch_name, branch_city, assets)

customer (customer_name, customer_street, customer_city)

account (account_number, branch_name, balance)

loan (loan_number, branch_name, amount)

depositor (customer_name, account_number)

borrower (customer_name, loan_number)

i) Create the above tables by properly specifying the primary keys.

ii) Enter at least five tuples for each relation.

iii) Find all loans of over 12000rs.

iv)display the branch names for a given city.

v) display depositor name for a specific account number.

vi) display customer names whose names starts with specified character.

EXERCISE 3

Date: 17/08/2023

Create a table called EMP with the following structure.

Name	**Type**
EMPNO	**INT (6)**
EFNAME	**VARCHAR (20)**
ELNAME	**VARCHAR (20)**
JOB	**VARCHAR (10)**
DEPTNAME	**VARCHAR (10)**
DEPTNO	**INT (2)**
ECITY	**VARCHAR (10)**
SAL	**INT (7,2)**
WORKEXPERIENCE	**INT(10)**
MANAGERNAME	**VARCHAR (10)**
MANAGERNO	**INT (20)**

Create dept table with the following structure.

Name	**Type**
DEPTNO	**INT (2)**
DNAME	**VARCHAR (10)**
LOC	**VARCHAR (10)**
LOCID	**INT (5)**

Q1. Write SQL queries to implement the following

1. Implement the above schema enforcing primary key and foreign key constraints and insert 5 records into the table.
2. Write a query to display the last name, department number, and department name for all employees.
3. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.
4. Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission.
5. Display the employee last name and department name for all employees who have an "*a*" (lowercase) in their last names.
6. Display the employee last name and employee number along with their manager's name and manager number.
7. Write a query to display the last name, job, department number, and department name for all employees who work in Toronto.
8. Modify the query 6 and display all employees including king, who has no manager and order the result by employee number.
9. Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label.
10. Find the sum and average of salary from the EMP table
11. Find the employee who is having maximum year of experience.
12. Find the number of employees working.
13. Find the employee who is having very less work experience.
14. Find the employee who is getting very high salary.

Create a table called Depositor with the following structure.

Name Type

---------- -------------------

CUSNAME VARCHAR (20)

ACC NO VARCHAR (20)

Create Borrower table with the following structure.

Name Type

------------ ---------------------

CUSNAME VARCHAR (20)

LOAN NO VARCHAR (20)

Q2. Write SQL queries to implement the following

1. Implement the above schema enforcing primary key constraints and insert 5 records into the table.
2. Find the names of all customers who have both loan and account in the bank
3. Find the names of all customers who have only loan in the bank
4. Find the names of all customers who have either loan or account in the bank

EXERCISE 4

Date: 24/08/2023

Create a table called EMP with the following structure.

Name Type
---------- ---- ------------------
EMPLOYEE_ID INT
FIRST_NAME VARCHAR
LAST_NAME VARCHAR
EMAIL VARCHAR
PHONE_NUMBER VARCHAR
HIRE_DATE DATE
JOB_ID VARCHAR (like IT_PROG, AD_PRES)
COMMISSION_PCT FLOAT
MANAGER_ID INT
DEPARTMENT_ID INT
IT, ST *specifies a particular department name*
Department names should be *Administration, Marketing, Purchasing, Human Resources, Shipping, IT, Public Relations, Sales, Executive, Finance, Accounting, Treasury, Corporate Tax, Control And Credit, Shareholder Services, Manufacturing, Construction, Contracting, IT Support, IT Helpdesk, Government Sales, Retail Sales, Recruiting, Payroll*

Create dept table called DEPT with the following structure.

Name Type
----------- --------------------
DEPARTMENT_ID INT
DEPARTMENT_NAME VARCHAR
MANAGER_ID INT
LOCATION_ID INT

Create a location table called LOCA with the following structure.

Name Type
----------- ---------------------
LOCATION_ID INT
STREET_ADDRESS VARCHAR
POSTAL_CODE INT
CITY VARCHAR
STATA_PROVINCE VARCHAR
COUNTRY_ID INT

1. Display all the information of an employee whose id is any of the number **134, 159** and **183**. ***(use In)***
2. Write a query to display all the information of the employees who does not work in those departments where some employees work whose manager id within the range **100 and 200.** ***(use Not in & Between)***
3. Write a query to display all the information for those employees whose id is any id who earn the second highest salary. ***(use In & Max)***

4. Write a query in SQL to display all the information about those employees who earn second lowest salary of all the employees. ***(use distinct)***Write a query to get the details of employees who are managers. ***(use exists)***
5. Write a subquery that returns a set of rows to find all departments that do actually have one or more employees assigned to them. ***(use distinct)***
6. Write a query to display the employee name (first name and last name) and department for all employees for any existence of those employees whose salary is more than 3700. ***(use exists)***
7. List department id, a department name for all the departments in which there are no employees in the department. ***(use Not exists)***
8. Write a query to display the employee number and name (first name and last name) for all employees who work in a department with any employee whose name contains a **T.** ***(use In)***
9. Write a query to display the employee number, name (first name and last name), and salary for all employees who earn more than the average salary and who work in a department with any employee with a **J** in their name. ***(use avg & In)***
10. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is smaller than any salary of those employees whose job title is **IT_PROG.** ***(use any)***
11. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is smaller than any salary of those employees whose job title is **IT_PROG.**Exclude Job title **IT_PROG.** ***(use any)***
12. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is more than any salary of those employees whose job title is **IT_PROG**. Exclude job title **IT_PROG**. ***(use all)***
13. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is more than any average salary of any department. ***(use all & avg)***
14. Write a query in SQL to display the first and last name, salary, and department ID for all those employees who earn more than the average salary and arrange the list in descending order on salary. ***(use order by)***
15. Write a query to display all the information of the employees whose salary is within the range of smallest salary and **2500.** ***(use Between & min)***

EXERCISE 5

31/8/2023

1. Create given tables and perform JOIN operations on them

Create a table called STUDENT with the following structure.

Name	**type**
---------	------------
Roll number	integer type
Name	character type
Address	character type
Phone	int type
Age	int type

Create a table called StudentCourse with the following structure.

Name	**type**
---------	------------
CourseId	integer type
Roll number	integer type

Perform given JOIN operations on the above tables.

i. INNER JOIN
ii. LEFT JOIN
iii. RIGHT JOIN
iv. FULL JOIN
v. NATURAL JOIN
vi. THETA JOIN
vii. EQUI JOIN

2. **Customer**(Cust id : integer, cust_name: string)

Item(item_id: integer,item_name: string, price: integer)

Sale(bill_no: integer, bill_date: date, cust_id: integer, item_id: integer, qty_sold: integer)

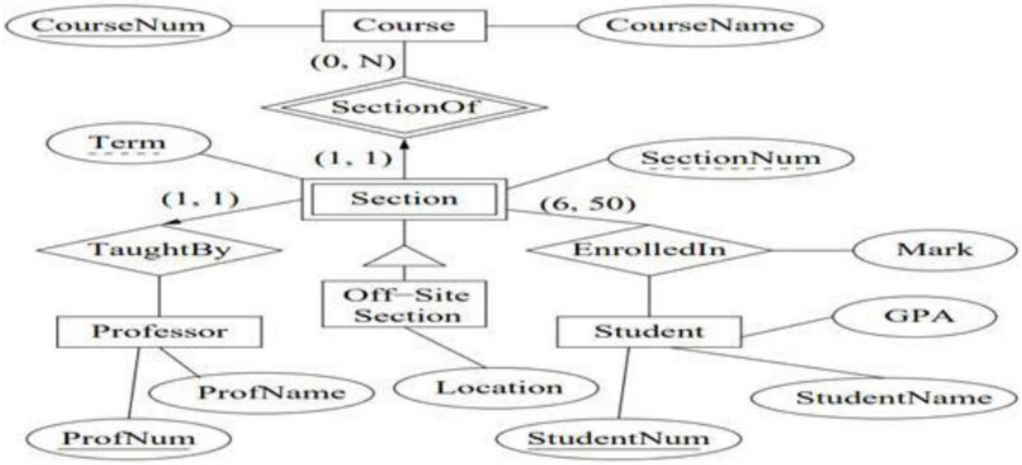
For the above schema, perform the following:

1. Create the tables with the appropriate integrity constraints and insert around 10 records in each of the tables
2. Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, and amount.
3. Create a view which lists the daily sales date wise for the last one week
4. Create a derived relation to get the top 5 products> by sales revenue in 2021 from the sale and Item tables
5. Classify the customers into 3 groups based on their purchases in 2021 and count the number of customers in each group using derived relation. Silver - < 10 k , Gold - > 10k and < 50 k, Platinum > 50k
6. Find the top 5 customer by their spending in year 2021 (use with clause)

EXERCISE 6

Date: 14/09/2023

1. Convert ER-diagram into relational database and to create the table for the relation by properly specifying the primary keys and foreign keys.



2. **Product (BarCode, PName, Price, QuantityInStock)**
   Sale (SaleID, DeliveryAddress, CreditCard)
   SaleItem (SaleID, BarCode, Quantity)

   Create a trigger called updateAvailableQuantity that updates the quantity in stock in the Product table, for every product sold. The trigger should be executed after each insert operation on the SaleItem table: for the product with the given barcode (the one inserted into SaleItem), update the available quantity in Product table to be the old quantity minus the sold quantity.

3. create the following tables with given attributes by specifying appropriate primary key and foreign keys. Tables should be created with necessary constraints which enables to perform on delete, on update cascade functions and self-referential integrity constraints.
   Employee(empNo, empName, jobPosition, managerId, salary)
   Department(department number, department name)
   Company(empNo, department number, joining date)

   Perform the following functions.
   i. On delete cascade
   ii. On update cascade
   iii. Self-referential integrity constraint.

EXERCISE 7

Date: 20/09/2023

Create a table called EMP with the following structure.

NAME	TYPE
EMPNO	NUMBER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
DEPTNO	NUMBER(3)
SAL	NUMBER(7,2)

Create DEPT table with the following structure.

Name	Type
DEPTNO	NUMBER(2)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

1. Create a procedure to display the details of an employee record from employee table for a given employee number.

2. Create a procedure to add details of a new employee into employee table

3. Write a procedure raise_sal which increases the salary of an employee. It accepts an employee number and salary increase amount. It uses the employee number to find the current salary from the EMPLOYEE table and update the salary.

4. Create a procedure to delete a record from employee table for a given employee name.

5. Write a function to display minimum salary of employees from the employee table.

6. Write a function to display the number of employees working in the Organization.

7. Write a function to display salary of an employee with the given employee number = 5.

8. Write a function average which takes DeptNo as input argument and returns the average salary received by the employee in the given department.

9. Write a procedure which takes the DeptNo =5 as input parameter and lists the names of all employees belonging to that department.

10. Write procedure that lists the highest salary drawn by an employee in each of the departments. It should make use of a named procedure dept_highest which finds the highest salary drawn by an employee for the given department.

11. Write a function that will display the number of employees with salary more than 30000.

12. Write a function that will display the count of the number of employees working in Mumbai.

EXERCISE 8

Date: 05/10/2023

1. Write a C Program to find Candidate Key from Functional Dependencies.
2. Write a C Program to find super Key from Functional Dependencies.

Test the program with the following set of functional dependencies:

i. Given R (X Y Z W) and FD= {XYZ → W, XY → ZW and X → YZW}

ii. Given R (X Y Z W) and FD= {X→Y, Y→Z, Z→X}

EXERCISE 9

Date: 12/10/2023

1.

Name	Type
---------- ----	------------------
EMPNO	NUMBER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
DEPT	VARCHAR2(10)
DEPTNO	NUMBER(3)
SAL	NUMBER(7,2)

Create a table called EMP with the following structure. Update any one attribute then show the result of following transaction operations.

These statements provide control over use of transactions:

- START TRANSACTION or BEGIN start a new transaction.
- COMMIT commits the current transaction, making its changes permanent.
- ROLLBACK rolls back the current transaction, canceling its changes.
- SET autocommit disables or enables the default autocommit mode for the current session.

By default, MySQL runs with autocommit mode enabled. To force MySQL not to commit changes automatically, you use the following statement:

SET autocommit = 0;

To disable autocommit mode implicitly for a single series of statements, use the `START TRANSACTION` statement:

2. Product (BarCode, PName, Price, QuantityInStock)

Sale (SaleID, DeliveryAddress, CreditCard)

SaleItem (SaleID, BarCode, Quantity)

(i). Create a trigger called updateAvailableQuantity that updates the quantity in stock in the Product table, for every product sold. The trigger should be executed after each insert operation on the SaleItem table: for the product with the given barcode (the one inserted into SaleItem), update the available quantity in Product table to be the old quantity minus the sold quantity.

(ii). Create a stored procedure called spInsertProduct that inserts a new product into the database, under some conditions. The stored procedure has as input parameters the barcode, the product name, price, and quantityInStock. The stored procedure should insert a row in the Product table only if the price is greater than 0 and the quantity is greater or equal to 0. If the conditions are not satisfied, the stored procedure just terminates (no errors generated)

(iii). Create a function called spreturn that returns the total price of a product by passing the quantity and barcode.

EXERCISE 10

Date: 19/10/2023

1. (i) Create a XML file which acts as a database with following nodes

<EmployeeDetails> as the root element
Create <Employee> element with following Child Nodes for atleast 5 employee details.
EmpNo
EName
Job
working Hours
Dept
DeptNo
Salary

1. **Create a xquery to list the salary > 30000**
2. **Get Employee numbers of employees whose last name starts with "S".**
3. **Get names of employees in the "Research" department.**
4. **Get employees who are managers work more than 8 hours**
5. **Display the salary in highest to lowest.**
6. **Display the Employee name in Alphabetical order.**

ii) Create a XML file which acts as a database with following nodes o XML file can be created in notepad with .xml extension and opened in any browser.

<Student Details> as the root element
Create <Student> element with following Child Nodes for atleast 5 employee details.
STUID
SName
Course
Dept
subject
Marks

1. **Create a xquery to list the Marks > 75**
2. **Find the Avg Mark of a Student.**
3. **Find the Total Marks of a Student.**
4. **Find the Min and Max Mark of a student in a subject.**

2. (i) Create a XML file which acts as a database with following nodes

<Flight Details> as the root element
Create <Flight> element with following Child Nodes for atleast 5 employee details.
FlNo
FlName
PilotName
From
To
Date

Departs Time
Arrives Time
Price

1. **Create a xquery to list the price of journey < 5000**
2. **Create a xquery to find the departs Time of the particular flight on a 4.12.2020 from a particular city.**
3. **Create a xquery to find the Flight Names handled by a particular Pilot.**
4. **Create a xquery to find out number of Flight journeys of a particular flight on 30.11.2020**
5. **Create a xquery to find Arrival Time of a particular flight on 25.11.2020 from a particular city.**

ii) Create a XML file which acts as a database with following nodes

<Employee Details> as the root element
Create <Employee> element with following Child Nodes for atleast 5 employee details.
EID
EName
Project
Job
Dept
DeptNo
salary

1. **Create a xquery to list the employees in Dept ='Human Resources'.**
2. **Create a xquery to find the Employee who works in particular project and salary > 50000.**
3. **Create a xquery to find the Total salary of Employees in a particular department.**
4. **Create a xquery to find the number of Employees working in a department.**
5. **Create a xquery to find the highest salary of a manager in particular department.**