GIRIJANANDA CHOWDHURY UNIVERSITY, GUWAHATI(ASSAM)

Introduction To Database System Lab

LAB MANUAL

COURSE CODE: BCS23213P

B.TECH 5TH SEMESTER

Prepared by:-

MRIDUSMITA BARUAH
Laboratory Instructor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEEING
GUWAHATI-781017

GIRIJANANDA CHOWDHURY UNIVERSITY, GUWAHATI(ASSAM)

BCS23213P	INTRODUCTION TO DATABASE SYSTEM LAB	L	T	P	C
		0	0	4	2

Prerequisite: Basic programming skills

Course Objectives:

1. To provide students with hands-on experience to design and create databases, write SQL queries, and perform data manipulation tasks, enabling them to translate theoretical knowledge into practical skills.
2. To gain proficiency in setting up and configuring database management systems, managing user access, and implementing security measures
3. To enhance students' abilities to optimize database queries and improve system performance

Course Outcome:

After Successful completion of the course, students will be able to

1 : Make use of Data Definition Language(DDL) statement using SQL for Creating, Deleting and Modifying Schemas.

2 :Make use of Data Manipulation Language(DML) statement using to SQL Retrieve, Store,Modify,Delete, Insert and Update data in Database.

3 : Apply the concept of trigger using SQL.

4 : Apply the concept of procedure using SQL.

Experiments

1. Write statements for creation/alteration/view of relational database schema along with necessary integrity constraints. Insert tuples into the created tables.
2. Write SQL statements for selection, updation and deletion of data
3. Write SQL statements for joins, grouping, nested sub query.
4. Write statements for creating and using stored procedures.
5. Write statements for creating and using triggers.

Total Lab hours	**30 hours**

Reference Books

1.	"Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2.	"Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
3.	"Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
4.	"Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

GIRIJANANDA CHOWDHURY UNIVERSITY, GUWAHATI(ASSAM)

LIST OF EXPERIMENTS:

Q1.Create a student database in Mysql and under this database create one table namely studentrecords containing field rollno,name,address.
a)Write an sql query to retrieve all students records
b)Write an sql query to retrieve the address of the students
c) Write an sql query to retrieve the name of the student whose roll no is 1.
c)Write an sql query to update a student records whose address is Guwahati.

Q2. Consider the following relational schema
Book(ISBN,book_title,category,page_count,price,year_of_publication)
a)Find the book_title and price of the books having page_count greater than 600.
b)Display price and ISBN of books whose category is either "novel","language" or "religious book"
c)Change the column name book_title to book_name
d)calculate total price of the books published in the year 1990.
e)Display the details of all the books except the book whose publication year is 1992 and arrange them in ascending order of price.
f)Find the total number of books.
g)Display the name of the books having price between 300 to 800.
h)Delete all the details of the book whose ISBN is 102.

Q3. Create an employee table
Enter employee name, designation and basic pay of 5 employees

1. Find out gross salary and net salary where

   DA=60% of Basic pay
   TA =20% of Basic pay
   HR=10% of Basic pay
   TAX=15% of Basic pay
   GROSS=Basic+DA+TA+HR
   NET=GROSS-TAX
2. Find the total no of employees.
3. Find the total no of employees having gross salary between 5000 to 15000.
4. Find out the maximum gross salary.
5. Display the details of the employee who gets maximum gross salary.

Q4. Consider the following relational schema

Employee (Emp_no, Name, Salary, design, dept_id, DOJ)
Department (Dept_id, DName, loc, DOE)

(a) Display the name of the employees working in marketing dept.
(b) Display the details of the employee joined in the month of July.
(c) Display the details of the employee who gets maximum salary.
(d) Count the no of employees in each dept.

Q5. Consider the following relational schema

Student (Rollno, Name, Address, DOB, C_id)
Course (C_id, Cname, Dur, Fees)

(a) Display rollno,name,cname,fees of each student
(b) Count the no of students in each course

Q6. Consider the following relational schema

Books(book_id,b_name,author,purchase_date,cost)
Members(member_id,m_name,address,phone,birthdate)
Issue_return(book_id,member_id,issue_date,return_date)

(a)Find the author of the books that have not been issued.
(b)Display the member_id and no of books issued to that (Assume that if a book in Issue_Return relation does not have a return_date then it is issued)
(c)Find the book that has been issued the minimum no of times.
(d)Display the names and author of the books that have been issued at any time to a member whose name begins with "Ra".
(e)Display the name and Cost of those books that have been issued to any member whose date of birth is less than 01-01-1989 but not been issued to any member having the birth date equal to or greater than 01-01-1989.

Q7.Consider the following relational schema

Teacher(teacher_id,teacher_name,address)
Student(name,phone,dob,s_id)
Course(c_id,cname,credit,teacher_id)
Result(s_id,c_id,mark)

(a) Find the name of the students whose results are not declared in any course
(b) Find the teachers who are teaching more than one course
(c) Display the name and marks of those students who were born before 1-1-1989 and score more than 80 marks in any course
(d) Find the details of students securing pass marks in more than 3 course
(e) Find the total no of credits earned by a students whose id is 10.
(f) Find name of the students who got maximum overall marks.
(g) Display the name and marks of those students who scored more than 80 marks in any subject.
(h) Find the details of the students securing less than 30 marks in more than 3 subjects.

Q8. Consider the following relational schema

Customer(C_id, Name , Address)
Item(i_code , Name , Price)
Purchase (P_id ,C_id , I_code, qty , pdate)
(a) Find the name of the customer who has done maximum purchase.
(b) Display the name of the item that has been purchased maximum no of times in the month of Feb.
(c) Display the name of the customer who didn't purchase any item.

Q9. Emp(emp_no,name,salary,supervisor_no,dept_code)
Dept(dept_code, dept_name)

i) employees who get more salary than their supervisor
ii) Department name and total number of employees in each Department.
iii) Name and department of employee(s) who earn maximum salary.

Q10. Write a procedure to accept an emp_id and display the employee details .

Q11. Write a procedure to accept an emp_id and return the employee salary.

Q12. Create a trigger insert on employee table so that:
Whenever a new record is inserted then the emp_id and date of insertion is stored in another table called new_rec.

Q13. Create a trigger delete on employee table so that:
whenever a record is deleted the emp_id and date of deletion is stored in another table called old_rec.

Q14. Create a trigger update on employee table so that:
whenever employee's salary is updated the emp_id , old salary and updated salary is stored in another table called update_info.

GIRIJANANDA CHOWDHURY UNIVERSITY, GUWAHATI(ASSAM)

Q1.Create a student database in Mysql and under this database create one table namely studentrecords containing field rollno,name,address.
a)Write an sql query to retrieve all students records
b)Write an sql query to retrieve the address of the students
c) Write an sql query to retrieve the name of the student whose roll no is 1.
c)Write an sql query to update a student records whose address is Guwahati.

MySql Queries:

```
CREATE DATABASE student;
USE student;

CREATE TABLE studentrecords (
  rollno INT PRIMARY KEY,
  name VARCHAR(100),
  address VARCHAR(100)
);
```

a) Retrieve all student records

```
SELECT * FROM studentrecords;
```

b) Retrieve address of all students

```
SELECT address FROM studentrecords;
```

c) Retrieve name of student whose roll no is 1

```
SELECT name FROM studentrecords WHERE rollno = 1;
```

d) Update address of student whose address is 'Guwahati'

```
UPDATE studentrecords SET address='Guwahati' WHERE rollno=2;
```

Q2. Consider the following relational schema
Book(ISBN,book_title,category,page_count,price,year_of_publication)
a)Find the book_title and price of the books having page_count greater than 600.
b)Display price and ISBN of books whose category is either “novel”,”language” or “religious book”
c)Change the column name book_title to book_name
d)calculate total price of the books published in the year 1990.
e)Display the details of all the books except the book whose publication year is 1992 and arrange them in ascending order of price.
f)Find the total number of books.
g)Display the name of the books having price between 300 to 800.
h)Delete all the details of the book whose ISBN is 102.

MySql Queries:

```
CREATE TABLE Book (
  ISBN INT PRIMARY KEY,
  book_title VARCHAR(255),
  category VARCHAR(100),
  page_count INT,
  price DECIMAL(10,2),
  year_of_publication INT
);
```

a)SELECT book_title, price FROM Book WHERE page_count > 600;

b)SELECT ISBN, price FROM Book WHERE category IN ('novel', 'language', 'religious book');

c)ALTER TABLE Book RENAME COLUMN book_title TO book_name;

d)SELECT SUM(price) FROM Book WHERE year_of_publication = 1990;

e)SELECT * FROM Book WHERE year_of_publication <> 1992 ORDER BY price;

f)SELECT COUNT(*) FROM Book;

g)SELECT book_name FROM Book WHERE price BETWEEN 300 AND 800;

h)DELETE FROM Book WHERE ISBN = 102

Q3. Create an employee table
Enter employee name, designation and basic pay of 5 employees

1. Find out gross salary and net salary where

   DA=60% of Basic pay
   TA =20% of Basic pay
   HR=10% of Basic pay
   TAX=15% of Basic pay
   GROSS=Basic+DA+TA+HR
   NET=GROSS-TAX
2. Find the total no of employees.
3. Find the total no of employees having gross salary between 5000 to 15000.
4. Find out the maximum gross salary.
5. Display the details of the employee who gets maximum gross salary.

MySql Queries:

```
CREATE TABLE Employee (
  emp_id INT PRIMARY KEY AUTO_INCREMENT,
  emp_name VARCHAR(100),
  designation VARCHAR(100),
  basic_pay DECIMAL(10,2)
);
INSERT INTO Employee (Emp_Name, Designation, Basic_Pay) VALUES
('Ravi Kumar', 'Manager', 12000),
('Anita Sharma', 'Clerk', 8000),
('Rajesh Singh', 'Accountant', 10000),
('Pooja Patel', 'Assistant', 7000),
('Amit Das', 'Salesman', 6000);
```

1. SELECT Emp_ID,Emp_Name,Designation,Basic_Pay,(0.6 * Basic_Pay) AS DA,(0.2 * Basic_Pay) AS TA,(0.1 * Basic_Pay) AS HR,(0.15 * Basic_Pay) AS TAX,(Basic_Pay + (0.6*Basic_Pay) + (0.2*Basic_Pay) + (0.1*Basic_Pay)) AS GROSS_SALARY, ((Basic_Pay + (0.6*Basic_Pay) + (0.2*Basic_Pay) + (0.1*Basic_Pay)) - (0.15*Basic_Pay)) AS NET_SALARY FROM Employee;

2. SELECT COUNT(*) AS Total_Employees FROM Employee;

3. SELECT COUNT(*) AS Employees_In_Range FROM Employee WHERE (Basic_Pay + (0.6*Basic_Pay) + (0.2*Basic_Pay) + (0.1*Basic_Pay)) BETWEEN 5000 AND 15000;

4. SELECT MAX(Basic_Pay + (0.6*Basic_Pay) + (0.2*Basic_Pay) + (0.1*Basic_Pay)) AS Max_Gross_Salary FROM Employee;

5. SELECT Emp_ID,Emp_Name, Designation,Basic_Pay,(Basic_Pay + (0.6*Basic_Pay) + (0.2*Basic_Pay) + (0.1*Basic_Pay)) AS GROSS_SALARY FROM Employee WHERE (Basic_Pay + (0.6*Basic_Pay) + (0.2*Basic_Pay) + (0.1*Basic_Pay)) = (SELECT MAX(Basic_Pay + (0.6*Basic_Pay) + (0.2*Basic_Pay) + (0.1*Basic_Pay)) FROM Employee);

Q4. Consider the following relational schema

Employee (Emp_no, Name, Salary, design, dept_id, DOJ)
Department (Dept_id, DName, loc, DOE)

(a) Display the name of the employees working in marketing dept.
(b) Display the details of the employee joined in the month of July.
(c) Display the details of the employee who gets maximum salary.
(d) Count the no of employees in each dept.

MySql Querirs:

a) SELECT e.Name FROM Employee JOIN Department d ON e.Dept_id = d.Dept_id WHERE d.DName = 'Marketing';

b) SELECT * FROM Employee WHERE MONTH(DOJ) = 7;

c) SELECT * FROM Employee WHERE Salary = (SELECT MAX(Salary) FROM Employee);

d) SELECT d.DName, COUNT(e.Emp_no) AS Total_Employees FROM Employee e JOIN Department d ON e.Dept_id = d.Dept_id GROUP BY d.DName;

Q5. Consider the following relational schema
Student (Rollno, Name, Address, DOB, C_id)
Course (C_id, Cname, Dur, Fees)
(a) Display rollno,name,cname,fees of each student
(b) Count the no of students in each course

MySql Queries:

a) SELECT s.Rollno, s.Name, c.Cname, c.Fees FROM Student s JOIN Course c ON s.C_id = c.C_id;

b) SELECT c.Cname, COUNT(s.Rollno) AS Total_Students FROM Student s JOIN Course c ON s.C_id = c.C_id GROUP BY c.Cname;

Q6. Consider the following relational schema

Books(book_id,b_name,author,purchase_date,cost)
Members(member_id,m_name,address,phone,birthdate)
Issue_return(book_id,member_id,issue_date,return_date)

(a)Find the author of the books that have not been issued.
(b)Display the member_id and no of books issued to that (Assume that if a book in Issue_Return relation does not have a return_date then it is issued)
(c)Find the book that has been issued the minimum no of times.
(d)Display the names and author of the books that have been issued at any time to a member whose name begins with "Ra".
(e)Display the name and Cost of those books that have been issued to any member whose date of birth is less than 01-01-1989 but not been issued to any member having the birth date equal to or greater than 01-01-1989.

MySql Queries:

a)SELECT DISTINCT b.author FROM Books b WHERE b.book_id NOT IN (SELECT DISTINCT book_id FROM Issue_Return);

b)SELECT member_id,COUNT(book_id) AS No_of_Books_Issued FROM Issue_Retur WHERE return_date IS NULL GROUP BY member_id;

c)SELECT b.book_id, b.b_name, COUNT(ir.book_id) AS Times_Issued FROM Books b JOIN Issue_Return ir ON b.book_id = ir.book_id GROUP BY b.book_id, b.b_name HAVING COUNT(ir.book_id) = (SELECT MIN(BookCount)FROM (SELECT COUNT(book_id) AS BookCount FROM Issue_Return GROUP BY book_id AS SubQuery);

d)SELECT DISTINCT b.b_name, b.author FROM Books b JOIN Issue_Return ir ON b.book_id = ir.book_id JOIN Members m ON ir.member_id = m.member_id WHERE m.m_name LIKE 'Ra%';

e) SELECT DISTINCT b.b_name,.cost FROM Books JOIN Issue_Return ir ON b.book_id = ir.book_id JOIN Members m ON ir.member_id = m.member_id WHERE m.birthdate < '1989-01-01' AND b.book_id NOT IN (SELECT ir2.book_id FROM Issue_Return ir2 JOIN Members m2 ON ir2.member_id = m2.member_id WHERE m2.birthdate >= '1989-01-01');

Q7.Consider the following relational schema

Teacher(teacher_id,teacher_name,address)
Student(name,phone,dob,s_id)
Course(c_id,cname,credit,teacher_id)
Result(s_id,c_id,mark)

(a) Find the name of the students whose results are not declared in any course
(b) Find the teachers who are teaching more than one course
(c) Display the name and marks of those students who were born before 1-1-1989 and score more than 80 marks in any course
(d) Find the details of students securing pass marks in more than 3 course
(e) Find the total no of credits earned by a students whose id is 10.
(f) Find name of the students who got maximum overall marks.
(g) Display the name and marks of those students who scored more than 80 marks in any subject.
(h) Find the details of the students securing less than 30 marks in more than 3 subjects.

MySql Queries:

a) SELECT s.name FROM Student s WHERE s.s_id NOT IN (SELECT DISTINCT r.s_id FROM Result r);

b) SELECT t.teacher_name, COUNT(c.c_id) AS No_of_Course FROM Teacher t JOIN Course c ON t.teacher_id = c.teacher_id GROUP BY t.teacher_name HAVING COUNT(c.c_id) > 1;

c) SELECT s.name, r.mark FROM Student s JOIN Result r ON s.s_id = r.s_id WHERE s.dob < '1989-01-01' AND r.mark > 80;

d) SELECT s.* FROM Student s JOIN Result r ON s.s_id = r.s_id WHERE r.mark >= 40 GROUP BY s.s_id HAVING COUNT(r.c_id) > 3;

e) SELECT SUM(c.credit) AS Total_Credits FROM Result r JOIN Course c ON r.c_id = c.c_id WHERE r.s_id = 10 AND r.mark >= 40;

f) SELECT s.name FROM Student JOIN Result r ON s.s_id = r.s_id GROUP BY s.s_id, s.name HAVING SUM(r.mark) = (SELECT MAX(TotalMarks)FROM (SELECT SUM(mark) AS TotalMarks FROM Result GROUP BY s_id) AS SubQuery);

g) SELECT s.name, r.mark FROM Student s JOIN Result r ON s.s_id = r.s_id WHERE r.mark > 80;

h) SELECT s.* FROM Student s JOIN Result r ON s.s_id = r.s_id WHERE r.mark < 30 GROUP BY s.s_id HAVING COUNT(r.c_id) > 3;

Q8. Consider the following relational schema

Customer(C_id, Name , Address)
Item(i_code , Name , Price)
Purchase (P_id ,C_id , I_code, qty , pdate)
(a) Find the name of the customer who has done maximum purchase.
(b) Display the name of the item that has been purchased maximum no of times in the month of Feb.
(c) Display the name of the customer who didn't purchase any item.

MySql Queries:

a)SELECT c.Name FROM Customer JOIN Purchase p ON c.C_id = p.C_id GROUP BY c.C_id, c.Name HAVING SUM(p.qty) = (SELECT MAX(TotalQty) FROM (SELECT SUM(qty) AS TotalQty FROM Purchase GROUP BY C_id) AS SubQuery);

b)SELECT i.Name FROM Item JOIN Purchase p ON i.i_code = p.i_code WHERE MONTH(p.pdate) = 2 GROUP BY i.i_code, i.Name HAVING SUM(p.qty) = (SELECT MAX(TotalQty FROM (SELECT SUM(qty) AS TotalQty FROM Purchase WHERE MONTH(pdate) = 2 GROUP BY i_code) AS SubQuery);

c) SELECT c.Name FROM Customer c WHERE c.C_id NOT IN (SELECT DISTINCT C_id FROM Purchase);

Q9. Emp(emp_no,name,salary,supervisor_no,dept_code)
Dept(dept_code, dept_name)

i) employees who get more salary than their supervisor
ii) Department name and total number of employees in each Department.
iii) Name and department of employee(s) who earn maximum salary.

MySql Queries:

i) SELECT e.name AS Employee, e.salary AS Employee_Salary, s.name AS Supervisor, s.salary AS Supervisor_Salary FROM Emp e JOIN Emp s ON e.supervisor_no = s.emp_no WHERE e.salary > s.salary;
ii) SELECT d.dept_name, COUNT(e.emp_no) AS Total_Employees FROM Dept d LEFT JOIN Emp e ON d.dept_code = e.dept_code GROUP BY d.dept_name;
iii) SELECT e.name AS Employee, d.dept_name FROM Emp e JOIN Dept d ON e.dept_code = d.dept_code WHERE e.salary = (SELECT MAX(salary) FROM Emp);

Q10. Write a procedure to accept an emp_id and display the employee details

MySql Queries:

```
DELIMITER $$

CREATE PROCEDURE GetEmployeeDetails(IN emp_id INT)
BEGIN
   SELECT emp_no, name, salary, supervisor_no, dept_code
   FROM Emp
   WHERE emp_no = emp_id;
END $$

DELIMITER ;
```

Q11. Write a procedure to accept an emp_id and return the employee salary.

MySql Queries:

```
DELIMITER $$

CREATE PROCEDURE GetEmployeeSalary(
   IN emp_id INT,
   OUT emp_salary DECIMAL(10,2)
)
BEGIN
   SELECT salary INTO emp_salary
   FROM Emp
   WHERE emp_no = emp_id;
END $$

DELIMITER ;
```

Q12. Create a trigger insert on employee table so that:
Whenever a new record is inserted then the emp_id and date of insertion is stored in another table called new_rec.

MySql Queries:

```
CREATE TABLE new_rec (
   emp_no INT,
   insertion_date DATETIME
);
DELIMITER $$
CREATE TRIGGER after_emp_insert
AFTER INSERT ON Emp
FOR EACH ROW
BEGIN
   INSERT INTO new_rec(emp_no, insertion_date)
   VALUES (NEW.emp_no, NOW());
END $$
DELIMITER ;
INSERT INTO Emp(emp_no, name, salary, supervisor_no, dept_code)
VALUES (105, 'Alice', 50000, 102, 1);

SELECT * FROM new_rec;
```

Q13. Create a trigger delete on employee table so that:
whenever a record is deleted the emp_id and date of deletion is stored in another table called old_rec.

MySql Queries:

```
CREATE TABLE old_rec (
   emp_no INT,
   deletion_date DATETIME
);
DELIMITER $$

CREATE TRIGGER after_emp_delete
AFTER DELETE ON Emp
FOR EACH ROW
BEGIN
   INSERT INTO old_rec(emp_no, deletion_date)
   VALUES (OLD.emp_no, NOW());
END $$
DELIMITER ;

DELETE FROM Emp
WHERE emp_no = 105;

SELECT * FROM old_rec;
```

Q14. Create a trigger update on employee table so that:
whenever employee's salary is updated the emp_id , old salary and updated salary is stored in another table called update_info.

MySql Queries:

```
CREATE TABLE update_info (
   emp_no INT,
   old_salary DECIMAL(10,2),
   new_salary DECIMAL(10,2),
   update_date DATETIME
);
DELIMITER $$

CREATE TRIGGER after_salary_update
AFTER UPDATE ON Emp
FOR EACH ROW
BEGIN
   -- Only insert if salary is actually changed
   IF OLD.salary <> NEW.salary THEN
      INSERT INTO update_info(emp_no, old_salary, new_salary, update_date)
      VALUES (OLD.emp_no, OLD.salary, NEW.salary, NOW());
   END IF;
END $$
DELIMITER ;
UPDATE Emp SET salary = 60000 WHERE emp_no = 101;

SELECT * FROM update_info;
```