**Padmakanya Multiple Campus**

**Bagbazaar, Kathmandu,**

**Nepal**

****

***Lab Report of DBMS SQL***

***2081***

**Submitted By: Submitted To:**

**Aakriti Neupane Mr. Mohan Singh Ayer**

**Bsc.CSIT (Fourth sem) Lecturer of DBMS**

**Section: A Signature:**

**Roll no: 25**

# Abstract

This lab report explores the fundamentals of Structured Query Language (SQL) and its practical applications in managing and manipulating relational databases. The lab focuses on executing various SQL queries, including data retrieval, insertion, updating, and deletion operations, on sample database tables. Key concepts such as SELECT statements, JOIN operations, aggregation functions, and subqueries are demonstrated to showcase how SQL can query and manipulate large datasets efficiently. The results highlight how SQL helps in efficiently querying and managing data in a database, and the importance of understanding basic SQL commands for database management tasks.

# Acknowledgment

I would like to express my sincere gratitude to Mr. Mohan Singh Ayer sir for his invaluable guidance and support throughout the SQL lab sessions. His expertise and encouragement significantly enhanced my understanding of SQL and its practical applications

My heartfelt thanks go to my peers for their collaboration, helpful discussions, and shared insights, which enriched my learning experience. I also acknowledge the resources, tutorials, and tools provided during the lab, which were essential for completing the exercises successfully. The experience gained in this lab has significantly contributed to my proficiency in using SQL for managing and manipulating relational databases, and I am truly thankful for the opportunity to explore these concepts in a hands-on setting.

Contents

[Abstract i](#_Toc185321829)

[Acknowledgment ii](#_Toc185321830)

[Background of Study 1](#_Toc185321831)

[Introduction 2](#_Toc185321832)

[LAB 1 3](#_Toc185321833)

[LAB 2 5](#_Toc185321834)

[LAB 3 8](#_Toc185321835)

[LAB 4 11](#_Toc185321836)

[LAB 5 18](#_Toc185321837)

[LAB 6 22](#_Toc185321838)

[LAB 7 25](#_Toc185321839)

[References 29](#_Toc185321840)

# Background of Study

The Bachelor of Science in Computer Science and Information Technology (BSc.CSIT), is a 4-year/8-semester course, designed to provide the students with all sorts of knowledge and skills in the fields of Information Technology and Computing. It fosters the skills that are essential for both computer professionals and IT specialists. The BSc.CSIT program provides the students with adequate theoretical and practical knowledge which will enable students to effectively participate in solving the complex problem of the IT industry.

In a CSIT, Database Management Systems (DBMS) is a key subject that provides knowledge about designing, managing, and interacting with databases.

The course begins by introducing the concept of databases, explaining how they differ from traditional file systems and why they are essential for data storage and retrieval in modern applications. Students are introduced to various data models, such as the relational model, which uses tables to represent data, and the Entity-Relationship (E-R) model, which helps in designing databases by defining entities, attributes, and relationships. The course covers Structured Query Language (SQL), a powerful tool for creating, managing, and querying relational databases, allowing students to perform operations like inserting, updating, and retrieving data.

A significant part of DBMS focuses on database design principles, including normalization, which reduces redundancy and ensures data integrity by organizing data into well-structured tables. Students also learn about transactions and the ACID properties (Atomicity, Consistency, Isolation, Durability), which guarantee the reliability and consistency of operations within a database. Concurrency control mechanisms, such as locking and deadlock prevention, are studied to understand how multiple transactions can be executed simultaneously without conflict.

The subject delves into advanced topics like database recovery techniques, which restore databases to a consistent state after failures, and indexing and hashing, which optimize data retrieval for faster query performance. Additionally, concepts like distributed databases, SQL databases, and data warehousing are explored to give students insight into modern database technologies used for large-scale, distributed, and unstructured data. The course also emphasizes database security, teaching methods to protect data from unauthorized access through encryption and access control mechanisms. Practical implementations, including writing SQL queries, designing database schemas, and working with tools like MySQL or Oracle, are integrated into the course, ensuring students develop hands-on skills alongside theoretical knowledge. This comprehensive approach prepares students for roles like database administrators, data analysts, or software developers in the future.

## Introduction

The primary goal of using SQL is to ensure efficient data management in databases, allowing businesses, organizations, and research to analyze and manipulate large sets of information with ease. SQL is commonly used in industries ranging from technology and healthcare to finance and education. The lab exercises in this report aim to demonstrate the practical implementation of SQL commands to perform operations such as data retrieval, data manipulation, joins, aggregations and so on.

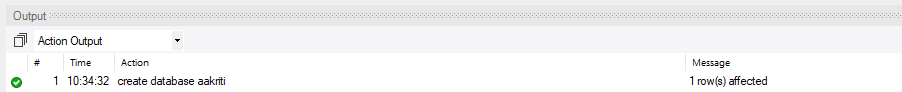
# LAB 1

Q.N.1. Create database as per your first\_name

Solution:

Query: create database aakriti;

Result:

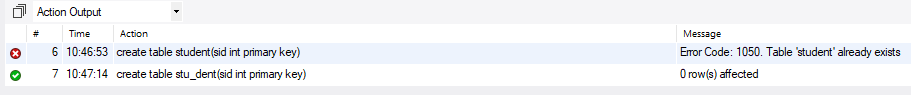


Q.N.2. Create Table Student with primary key sid.

Solution:

Query: create table stu\_dent(sid int primary key);

Result:

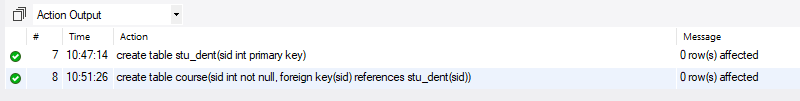


Q.N.3. Create table Course with foreign key sid.

Solution:

Query: create table course(sid int not null, foreign key(sid) references stu\_dent(sid));

Result:

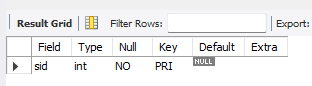


Q.N.4. Describe structure of table student.

Solution:

Query: describe stu\_dent;

Result:



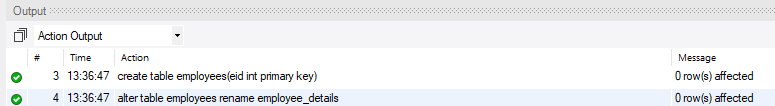
Q.N.5. Rename table employee with new name employee\_details.

Solution:

Query: create table employees(eid int primary key);

alter table employees rename employee\_details;

Result:



# LAB 2

Q.N.1. Write a query to display all employee who are from Bhaktapur.

Solution:

Query: select ename from employee where address="Bhaktapur" ;

Result:



Q.N.2 Create table employee with minimum 5 attributes and insert minimum 10 records.

Solution:

**Query**:

create table employees(eid int primary key, ename text,address text,salary integer, job text);

insert into employees(eid,ename,address,salary,job)

values

(1,"Aakriti","Chitwan",110000,"IT\_engineer"),

(2,"Prijma","Lalitpur",25000,"Security\_engineer"),

(3,"Prijal","Bhaktapur",10000,"Chemist"),

(4,"Sansar","Bharatpur",15000,"Manager"),

(5,"Kushal","Itahari",50000,"Doctor"),

(6,"Kamana","KTM",45000,"Nurse"),

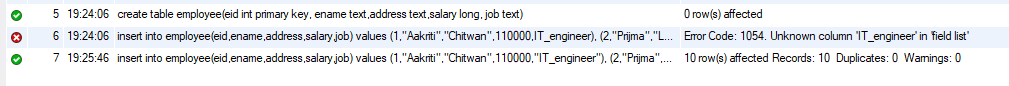
(7,"Pragesh","Raniban",100000,"Software\_Engineer"),

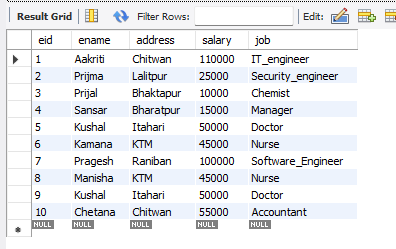
(8,"Manisha","KTM",45000,"Nurse"),

(9,"Kushal","Itahari",50000,"Doctor"),

(10,"Chetana","Chitwan",55000,"Accountant");

**Result**:





Q.N.3. Update table employee set new address kathmandu whose id is 2.

Solution:

**Query:** update employee set address="KTM" where eid ="2";

**Result:**



Q.N.4. Create table department with eid as a foreign key.

Solution:

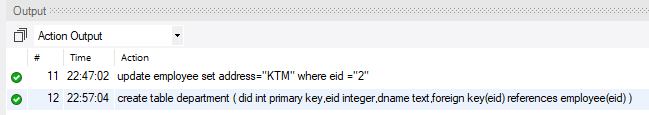
**Query:**

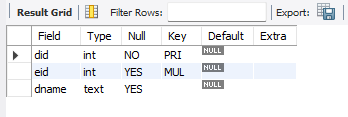
create table department

(did int primary key,eid integer,dname text,foreign key(eid) references employee(eid));

describe department;

**Result:**





Q.N.5. Write a query which will increase the salary of each department by Rs 1000.

Solution:

**Query:**

UPDATE employees

SET salary = salary + 1000;

**Result:**

****

# LAB 3

Q.N.1. Write a query to increase salary of employees by 10% whose salary is more than 50000.

Solution:

**Query:**

update employees

set salary=salary\*0.10 +salary

where salary>50000;

**Result:**

****

Q.N.2. Write a query to change the department id of employees whose old department id is 201. The new department id should be 501.

Solution:

**Query:**

update department

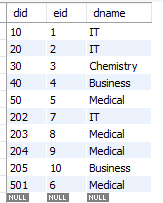
set did="501"

where did = 201;

/\*select\* from department;\*/

**Result:**

****

****

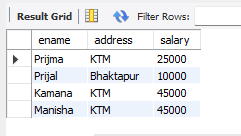
Q.N.3. Write a query to select name, address and salary of all employees who are from Kathmandu, Lalitpur and Bhaktapur.

Solution:

**Query:**

select ename, address, salary from employee where address IN("Lalitpur","KTM","Bhaktapur");

**Result:**



Q.N.4.Write a query to select name, department name and print employee details who are working in IT department.

Solution:

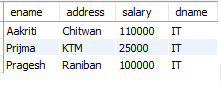
**Query:**

select e.ename,e.address,e.salary,d.dname from employee e

join department d ON e.eid=d.eid

where dname='IT' ;

**Result:**



Q.N.5. Create table **exam ,** marks and subject should be unique.

Solution:

**Query:**

create table exam (sid int primary key,subject text,marks integer,foreign key(sid) references student1(sid));

insert into exam(sid,subject,marks)

values

(1,"DBMS",90),

(2,"DSA",85),

(3,"MB",80),

(4,"Eco",89);

select \* from exam;

**Result:**

Q.N.6. Create table **customer** and **Orders** and orders table should have foreign key.

Solution:

**Query:**

create table customer (cid int primary key);

create table orders(cid integer, oid integer, order\_date integer,totalorder integer,foreign key(cid) references customer(cid));

**Result:**

****

****

Q.N.7. Create table **atm** where atmid should be greater than 10.

Solution:

**Query:** create table atm(atmid int primary key , check(atmid>10));

insert into atm(atmid) values(1),(11);

**Result:**

****

****

Q.N.8. Create table **bank** where default balance should be 1000.

Solution:

**Query:**

create table bank(bankid int primary key,balance int default '1000');

insert into bank(bankid,balance)

values

(107,20000),

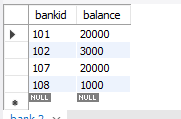
(108,default);

select \* from bank

**Result:**

****

****

****

# LAB 4

Q.1. Write a SQL query to retrieve empno, ename, job,salary of all employees in descending order of their salary.

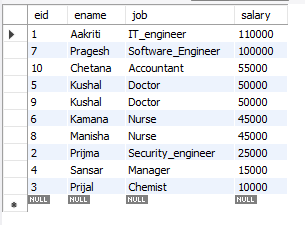
Solution:

Query:

select eid,ename, job, salary from employees

order by salary desc;

Result:



Q.2. Write a SQL query to retrieve all information of employee that belongs to department number 10 or 20.

Solution:

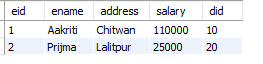
Query:

select e.eid,ename,address,salary,did from employees e

join department d on e.eid=d.eid

where did='10' or did='20';

Result:



Q.3. Write a query in SQL to list the employee who does not belongs department no 10.

Solution:

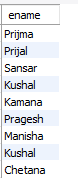
Query:

select ename from employees e

join department d on e.eid=d.eid

where did!='10';

Result:



Q.4. Write a SQL query to retrieve employee information whose salary is greater than average salary of all employee.

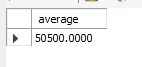
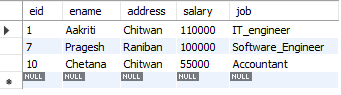
Solution:

Query:

/\*select avg(salary) as average from employees ;\*/

select \*from employees where salary(select avg(salary)from employees );

Result:

Q.5. Write a SQL query to display all employee who does not have any commission.

Solution:

Query:

alter table employee add column commission integer;

select ename from employee where commission= 'Null';

Result:



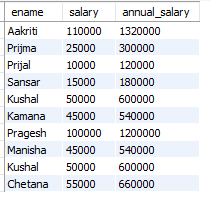


Q.6. Display empname and annual total salary of individual’s employee from employee table.

Solution:

Query: select ename, salary,(salary\*12)as annual\_salary from employees;

Result:



Q.7. Write a SQL query to display orders date and total orders amount using group by clause.

Solution:

Query: (column created at lab 3, question 6)

/\* alter table orders

MODIFY column order\_date DATE;

SELECT \* FROM orders;

DESCRIBE ORDERS;

insert into customer(cid) values (1),(2),(3);

insert into orders(cid,oid,order\_date,totalorder)

values

(1,101,"2080-1-1",15),

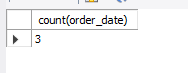
(2,102,"2080-1-1",20),

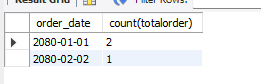
(3,103,"2080-2-2",10); \*/

SELECT count(order\_date) from orders;

select order\_date, count(totalorder) from orders group by order\_date;

Result:





Q.8. Write a SQL query to display information of employee whose name starts with A.

Solution:

Query: select \*from employees where ename like 'A%';

Result:



Q.9. Write a SQL query to find out total number of department in the given employee table.

Solution:

Query: select count(dname) as total\_dep from department;

Result:

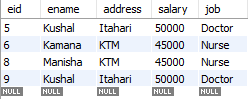


Q.10. Display all the information of employee whose salary is between 30000 AND 50000.

Solution:

Query: select \*from employees where salary not between 30000 and 50000;

Result:

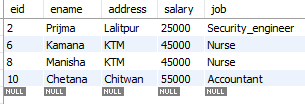


Q11. Write a SQL query to display information of employee whose name ends with A.

Solution:

Query: select \*from employees where ename like '%A';

Result:



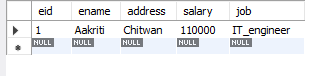
Q12. Write a SQL query to display information of employee whose name starts with A and ends

with I.

Solution:

Query: select \*from employees where ename like 'A%I';

Result:

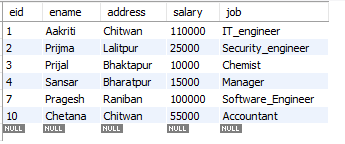


Q13. Display all the information of employee whose salary is not between 30000 AND 50000.

Solution:

Query: select \*from employees where salary not between 30000 and 50000;

Result:

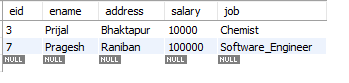


Q14. Display all the information of employee whose salary starts with 10.

Solution:

Query: select \*from employees where salary like '10%';

Result:

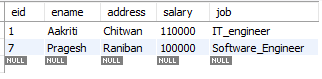


Q15. Display all the information of employee whose salary exactly have 6 digit.

Solution:

Query: select \*from employees where salary like '\_\_\_\_\_\_';

Result:



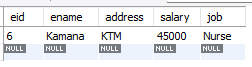
Q16.Write a query to display employee in which city name starts with ‘ka’ ends with ‘na’ and

contains multiple character between ‘ka’ and ‘na’.

Solution:

Query: select \*from employees where ename like 'ka%na';

Result:



# LAB 5

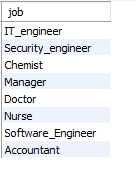
Q.1. Display different job levels of employees in employee table.

Solution:

Query:

select distinct(job) from employees;

Result:

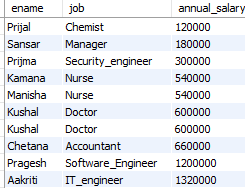


Q.2. Display empname, job, annual salary of employee using order by clause.

Solution:

Query: select ename,job,(salary\*12) as annual\_salary from employees order by annual\_salary;

Result:



Q.3. Display the maximum salary of employee using order by salary.

Solution:

Query: Select ename,salary from employees where salary=(select max(salary) from employees);

Result:



Q.4. Print the minimum salary of employee.

Solution:

Query: Select ename, salary from employees where salary=(select min(salary) from employees);

Result:



Q.5. Print the total salary of all employee

Solution:

Query: select sum(salary) as total\_salary from employees;

Result:

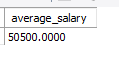


Q.6. Display the average salary of employee.

Solution:

Query: select avg(salary) as average\_salary from employees;

Result:



Q.7. Display branch and total marks of individual department(branch in my solution) from student table using group by clause.

Solution:

Query:

First creating table student and exam

create table student1(sid int primary key ,sname text ,branch text,location text,phone long, section text);

insert into student1(sid,sname,branch,location,phone,section)

values

(1,"Aakriti","IT","KTM","984111111","A"),

(2,"Prijal","IT","CTN","984222222","B"),

(3,"Prijma","Bio","Gorkha","904111111","A"),

(4,"Pragesh","Business","KTM","98433333","A");

select \* from student1;

create table exam (sid int primary key,subject text,marks integer,foreign key(sid) references student1(sid));

insert into exam(sid,subject,marks)

values

(1,"DBMS",90),

(2,"DSA",85),

(3,"MB",80),

(4,"Eco",89);

select \* from exam;

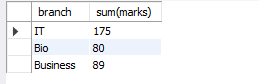
select branch,sum(marks) from student1 as s

join exam as e

on s.sid=e.sid

group by branch;

Result:



Q.8. Display branch and total marks of IT branch only from student table.

Solution:

Query:

select branch,sum(marks) from student1 as s

join exam as e

on s.sid=e.sid

where branch = 'IT';

Result:



Q.9. Create view name as viewIT on the basis of IT branch.

Solution:

Query:

CREATE VIEW viewIT AS

SELECT \* FROM student1

WHERE branch = 'IT';

Result:



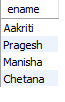
# LAB 6

Q.1. To find employee name containing exactly 7 characters use 7 instances of the \_pattern character.

Solution:

Query: select ename from employees where ename like '\_\_\_\_\_\_\_';

Result:



Q.2.. Display all the records from left table with matched records from right table. There are two tables given to you as employee and department.

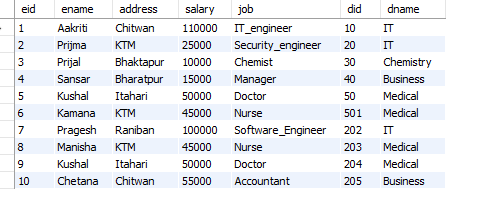
Query:

select e.eid,ename,address,salary,job,did,dname from employee as e

inner join department as d

on e.eid=d.eid;

Result:



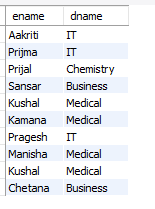
Q.3. Display employee name and department name from employee and department table using natural join.

Query:

select ename,dname from employee as e

natural join department as d

Result:



Q.4. Replace the table employee with table emp.

Query: alter table employee rename emp;

Result:

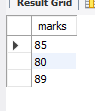


Q.5. Write a query for adding empno in emp table at first row of the table.

Q.6. Write a query to display marks details in which marks starts with 8.

Query: select marks from exam where marks like "8%";

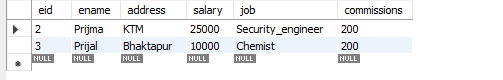
Result:



Q.7. Write a query to display employee details in which name contains letter ‘ik’ in between.

Query: select \* from employee where ename like "%ij%";

Result:

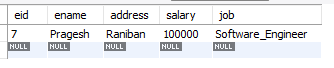


Q.8. Display all employees whose salary is either 30000,40000 or 100000.

Solution:

Query: select \* from employees where salary in ( 30000, 40000 ,100000);

Result:

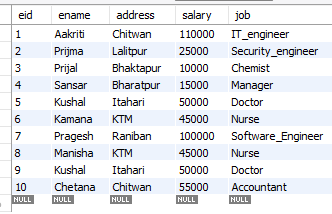


Q.9. Display all employees with the salary except between 40000 to 50000.

Solution:

Query: select \*from employees where salary not between 400000 and 500000;

Result:

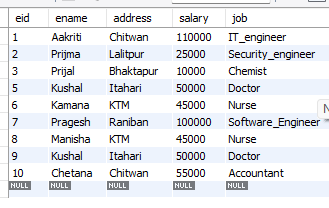


Q.10. Select all records from employee table where job is not manager.

Solution:

Query: select \* from employees where job! = "Manager";

Result:



# LAB 7

**Q1.**  Create table customer with this attributes (ID,Name,Age,Address,Salary,Primary key(ID)) and insert minimum 7 records in the table.

Query:

create table customers(cid int primary key,name text,age integer,salary int);

insert into customers(cid,name,age,salary)

values

(1,"A",20,2000),

(2,"B",21,3000),

(3,"C",22,4000),

(4,"D",23,000),

(5,"E",24,2000),

(6,"F",20,2000),

(7,"G",20,2000);

Result:

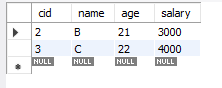


Q2. Write a SQL statement and display customer information customer salary greater than 2000.

Query:

select \* from customers where salary>2000;

Result:

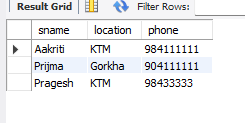


Q3. To display NAME, LOCATION, PHONE\_NUMBER of the students from student table whose section is A

Query:

select sname,location,phone from student1 where section='A';

Result:



Q4. Display employee’s name whose salary is greater than 100000 and age is less than 20.

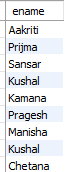
Query:

SELECT ename

FROM employees

WHERE salary>10000 AND age>25;

Result:



Q5. Update table customer and update salary by 10000, it means increase the salary of customer by 10000.

Query:

update customer

set salary=salary+10000;

Result:



Q6. Write a SQL statement and print employee\_id and salary whose employee\_id is 9.

Solution;

Query: select eid, salary from employees where eid = 9;

Result:

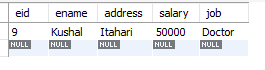


Q7. Write a SQL statement and print employee details whose salary is greater than 45000 and employee\_id is 9.

Solution:

Query: select \* from employees where salary>45000 and eid=9;

Result:



Q8. Write a SQL statement and print employee name and salary whose post is manager.

Solution:

Query: select ename, salary from employees where job='Manager';

Result:



Q9. Write a SQL statement to add joining\_date of employee in employee table.

Query:

alter table employees

add joining\_date date;

Solution:



Q10. Write a SQL statement to add email address after address of employee in employee table.

Query: alter table employees

add email\_add text after address;

Solution:



Q11. Update employee table for adding employee salary with 10000 in employee table.

Query:

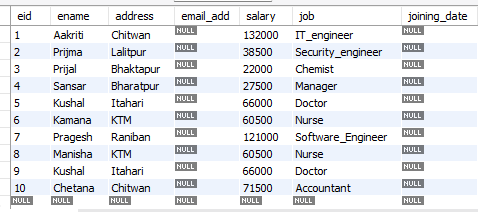
SET SQL\_SAFE\_UPDATES = 0;

UPDATE employees

SET salary=salary + 10000;

Solution:





# References

|  |  |
| --- | --- |
| [1] | tutorialspoint. [Online]. Available: https://www.tutorialspoint.com/index.htm. |
| [2] | "w3schools," [Online]. Available: https://www.w3schools.com/. |
| [3] | "stackoverflow," [Online]. Available: https://stackoverflow.com/. |
| [4] | I. C. Bhupendra Singh Saud, Database Management System, KEC Publication and Distribution Pvt.Ltd, 2077. |