

Mechi Multiple Campus

(Tribhuvan University)

Bhadrapur, Jhapa



Lab Report of Database Management System (CACS-255)

Faculty of Humanities & Social Sciences

Tribhuvan University

Kritipur, Nepal

Submitted By

Name: Santosh Bhandari

Roll No: 58

Submitted To

Mechi Multiple Campus

Department of Bachelor in Computer

Bhadrapur, Jhapa, Nepal

Certificate from the supervisor

This is to certify that the Lab Report entitled "**Database Management System**" is an academic work done by "**Santosh Bhandari**" submitted in the partial fulfillment of the requirements for the degree of **Bachelor of Computer Application** at Faculty of Humanities and Social Science, Tribhuvan University under my guidance and supervision. To the best of my knowledge, the work performed by him in the Lab report is his own creation.

Signature of the Supervisor:-

Name:-

Designation:-

Date:-

Acknowledgement

I would like to express my special thanks of gratitude to my Database Management System teacher “Mr. Sunil Sharma” and to the whole faculty member of humanities and social science who has provided me this opportunity. I am really grateful to sir Sunil Sharma for this kind of support in this report.

We are making this report as it compulsory required by Mechi Multiple Campus. It is compulsory report that should be submitted to the college in order to get practical marks. For this report our Database Management System teacher has motivated us. We are also guided by him to score good marks in practical. He has suggested how to make the report. He was the main source for us to make the report ready for Mechi Multiple Campus. In this report, we the student of BCA were engaged and I would like to express my deep thankful particularly to all of them. We are doing this report to be very helpful for coming days.

Thank You.

Table of Content

S.N	Title	Page No.
1.	Labsheet 1	1-2
2.	Labsheet 2	3-8
3.	Labsheet 3	9-12
4.	Labsheet 4	13-15
5.	Labsheet 5	16-18
6.	Labsheet 6	19-20
7.	Labsheet 7	21-22
8.	Labsheet 8	23-26
9.	Labsheet 9	27-28
10.	Labsheet 10	29-30
11.	Labsheet 11	31-32
12.	Labsheet 12	33-43

LABSHEET -2

1. Create a Database named Students_Info.

`CREATE DATABASE Students_Info;`

2. Create the table as given below.

Student

SNO	Sname	Saddress	Sage

`CREATE TABLE Student (SNO int, Sname varchar(50), Saddress varchar(50), Sage int);`

Output

SNO	Sname	Saddress	Sage

3. Add new column SDOB to the above table.

`ALTER TABLE Student ADD SDOB DATE;`

Output

SNO	Sname	Saddress	Sage	SDOB

4. Remove the column Sage from the above table.

`ALTER TABLE Student DROP COLUMN Sage;`

Output

SNO	Sname	Saddress	SDOB

5. Change the Column Name SNO to Student_id.

ALTER TABLE Student CHANGE SNO Student_id int;

Output

Student_id	Sname	Saddress	SDOB

6. Change the datatype of student_id to varchar.

ALTER TABLE Student MODIFY Student_id varchar(20);

Output

Student_id	Sname	Saddress	SDOB

7. Change the name of the existing table to KCL_Students.

ALTER TABLE Student RENAME TO KCL_Students;

Output

KCL_Students

Student_id	Sname	Saddress	SDOB

LABSHEET-2

1. Create the Table as given below.

eid	ename	eaddress	egender	eage	esalary
1	Niraj	Ktm	M	21	12000
2	Faroj	Dkh	M	22	5000
3	Suraj	Ktm	M	20	10500
4	Sujeena	Lpt	F	19	7500
5	Sandhya	NULL	F	20	19000
6	Prerana	Pkh	F	18	13000
7	Subodh	NULL	M	24	9800
8	Samir	NDJ	M	23	8500
9	Samipa	Brt	F	22	6700
10	Uday	NULL	M	25	11300

```
CREATE TABLE Employee(eid int, ename varchar(50), eaddress varchar(50),
egender char(2), eage int, esalary decimal(10,2));
```

Output

eid	ename	eaddress	egender	eage	esalary

2. Write an SQL command to insert the data as given below.

```
INSERT INTO Employee VALUES(1, "Niraj", "Ktm", "M", 21, 12000),
(2, "Faroj", "Dkh", "M", 22, 5000),
(3, "Suraj", "Ktm", "M", 20, 10500),
(4, "Sujeena", "Lpt", "F", 19, 7500),
(6, "Prerana", "Pkh", "F", 18, 13000),
(8, "Samir", "NDJ", "M", 23, 8500),
(9, "Samipa", "Brt", "F", 22, 6700);
```

```
INSERT INTO Employee(eid, ename, egender, eage, esalary)
VALUES (5, "Sandhya", "F", 20, 19000),
(7, "Subodh", "M", 24, 9800),
(10, "Uday", "M", 25, 11300);
```

Output

eid	ename	eaddress	egender	eage	esalary
1	Niroj	Ktm	M	21	12000
2	Fayoj	Dkh	M	22	5000
3	Suraj	Ktm	M	20	10500
4	Sujeena	Lpt	F	19	7500
6	Drenana	Dkh	F	18	13000
8	Samir	NDS	M	23	80500
9	Samipa	Brt	F	22	6700
5	Sandhya	[NULL]	F	20	14000
7	Subodh	[NULL]	M	23	8800
10	Vday	[NULL]	M	25	11300

3. Retrieve name and address of all Employees.

SELECT ename, eaddress FROM Employee;

Output

ename	eaddress
Niroj	Ktm
Fayoj	Dkh
Suraj	Ktm
Sujeena	Lpt
Drenana	Dkh
Samir	NDS
Samipa	Brt
Sandhya	[NULL]
Subodh	[NULL]
Vday	[NULL]

4. Retrieve name and address of these employees who lives in ktm and whose salary is above 7900.

SELECT ename, eaddress FROM Employee WHERE eaddress = "Ktm" AND esalary > 7900;

Output

ename	eaddress
Niroj	Ktm
Suraj	Ktm

5. Remove Employees Whose age is less than or equal to 20.
~~DELETE FROM Employee WHERE eage <= 20;~~
Output

eid	ename	eaddress	egender	eage	esalary
1	Niroj	Ktm	M	21	12000
2	Farms	Dkh	M	22	5000
8	Samer	NDJ	M	23	8500
9	Samipa	Brt	F	22	6700
7	Subodh	[NULL]	M	23	9800
10	Vday	[NULL]	M	25	11300

6. Retrieve information of those Students Who lives in ktm and Whose age is above 22 or Whose salary is greater than 10500.

~~SELECT * FROM Employee WHERE eaddress = "ktm" AND (eage > 22 OR esalary > 10500);~~

Output

eid	ename	eaddress	egender	eage	esalary
1	Niroj	Ktm	M	21	12000

7. Show those employees Whose address is not currently known.

~~SELECT * FROM Employee WHERE eaddress IS NULL;~~

Output

eid	ename	eaddress	egender	eage	esalary
7	Subodh	[NULL]	M	23	9800
10	Vday	[NULL]	M	25	11300

8. Change the column eid to EMP-ID.

~~ALTER TABLE Employee CHANGE eid EMP-ID int;~~

Output

EMP-ID	ename	eaddress	egender	eage	esalary

9. Change the datatype of emp_id from integer to varchar.

ALTER TABLE Employee MODIFY emp_id varchar(10);

Output

EMP-ID	ename	eaddress	egender	eage	esalary

10. Increase the age of employee Niraj by 2.

UPDATE Employee SET eage=eage+2 WHERE ename = "Niraj";

Output

EMP-ID	ename	eaddress	egender	eage	esalary
1	Niraj	KTM	M	23	12000
2	Firoj	Dkh	M	22	5000
8	Samar	NDS	M	23	8500
9	Samgpa	Brt	F	22	6700
7	Subodh	[NULL]	M	23	9800
10	Vday	[NULL]	M	25	11300

11. Delete those employee whose address is Brt or salary is greater than 15000.

DELETE FROM Employee WHERE eaddress = "Brt" OR esalary > 15000;

Output

EMP-ID	ename	eaddress	egender	eage	esalary
1	Niraj	KTM	M	23	12000
2	Firoj	Dkh	M	22	5000
8	Samar	NDS	M	23	8500
7	Subodh	[NULL]	M	23	9800
10	Vday	[NULL]	M	25	11300

12. Increase the Salary of all Employees by 7 percentages.

UPDATE Employee SET esalary = esalary + esalary * 0.07;

Output

EMP-ID	ename	eaddress	egender	eage	esalary
1	Niroj	KTM	M	23	12840
2	Firoj	Dkh	M	22	5350
8	Samir	NDJ	M	23	9095
7	Subodh	[NULL]	M	23	1086
10	Uday	[NULL]	M	25	12091

13. Now Increase the salary of employee by 5% who earns more than 15000, else increase it by 10%.

UPDATE Employee SET esalary = CASE
WHEN esalary > 15000 THEN esalary + esalary * 0.05
ELSE esalary + esalary * 0.1 END;

Output

EMP-ID	ename	eaddress	egender	eage	esalary
1	Niroj	KTM	M	23	14124
2	Firoj	Dkh	M	22	5885
8	Samir	NDJ	M	23	10004.5
7	Subodh	[NULL]	M	23	11534.6
10	Uday	[NULL]	M	25	13300.1

14. Display employee whose salary is in the range 6999 and 11998.

SELECT * FROM Employee WHERE esalary BETWEEN 6999 AND 11998;

Output

EMP-ID	ename	eaddress	egender	eage	esalary
8	Samir	NDJ	M	23	10004.5
7	Subodh	[NULL]	M	23	11534.6

15. Display employees who do not live in ktm.

`SELECT * FROM Employee WHERE address != "ktm";`

Output

EMP-ID	ename	address	egender	eage	esalary
2	Froy	Pkh	M	22	5885
8	Samir	NPJ	M	23	10004.5

16. Update the database such that Uday now lives in NPJ.

`UPDATE Employee set address = "NPJ" WHERE ename = "Uday";`

Output

EMP-ID	ename	address	egender	eage	esalary
1	NPJ	Ktm	M	23	14124
2	Froy	Pkh	M	22	5885
8	Samir	NPJ	M	23	10004.5
7	Subadh	[NULL]	M	23	11534.5
10	Uday	NPJ	M	25	13300.0

LABSHEET - 3

Create the table for Employee (eid, ename, branch-name, salary)

Eid	ename	Branch-name	Salary
10	Saroj	KTM	13000
11	Bikash	LPT	15000
13	Rabi	KTM	18000
16	Subash	LPT	15000
19	Santhosh	BKT	9500
20	Ajay	KTM	11000
21	Kiran	BKT	14500
22	Bipin	LPT	7500

CREATE TABLE Employee (eid int, ename varchar(25), Branch-name varchar(25), salary decimal (10,2));

Output

Eid	ename	Branch-name	Salary

1. Find the total sum salaries of all employees.

SELECT SUM(salary) FROM Employee;

Output

SUM(salary)
101500

2. Find the highest Salary, Column name Should be displayed as Highest-Salary.

SELECT MAX(salary) as Highest-Salary FROM Employee;

Output

Highest-Salary
18000

3. Find the average Salary of Employees.

SELECT AVG(salary) FROM Employee;

Output

AVG(Salary)
12667.5

4. Display the total No of Employees.

```
SELECT COUNT(*) FROM Employee;
```

Output

COUNT(*)
8

5. Display the total values appearing in the field 'branch-name'.

```
SELECT DISTINCT branch_name FROM Employee;
```

Output

Branch_name
KTM
LPT
BKT

6. Display the total No of branches appearing in the 'Employee' relation.

```
SELECT COUNT(branch_name) FROM Employee;
```

Output

COUNT(branch_name)
3

7. Sort the Name of Employees in ascending order and display the first name of Employee.

```
SELECT ename FROM Employee ORDER BY ename ASC;
```

Output

ename
Ajay
Bikash
Bipin
Kiran
Rabi
Santhosh
Saroj

8. Find the name of employees who has the highest Salary.

`SELECT ename FROM Employee WHERE salary = (SELECT MAX(salary) FROM Employee);`

Output

ename
Rabi

9. Find the name of employee who has the lowest Salary.

`SELECT ename FROM Employee WHERE salary = (SELECT MIN(salary) FROM Employee);`

Output

ename
Santhosh
Bipin

10. Display the employee whose salary is above average salary.

`SELECT ename FROM Employee WHERE salary > (SELECT AVG(salary) FROM Employee);`

Output

ename
Sonos
Bikash
Rabi
Subash
Kiran

11. Find the name of employee whose salary is same as that of 'Bipin'.

`SELECT ename FROM Employee WHERE salary = (SELECT salary FROM Employee WHERE ename = 'Bipin') AND ename <> 'Bipin';`

Output

ename
Santhosh

12. Display the Total Sum Salaries of all Employees at each branch Separately.

SELECT sum(Salary) FROM Employee GROUP BY branch_name;

Output

SUM(Salary)
22,000
42000
37,500

13. Display the branch name where the total salary of each branch must be greater than 50000.

SELECT branch_name, sum(Salary) FROM Employee GROUP BY branch_name HAVING sum(Salary) > 50000;

Output

branch_name	SUM(Salary)

15

LABSHEET-4

College

S-id	S-name	S-address	Age	Phone
1	Ram	Kathmandu	21	5562236
2	Hari	Pokhara	25	5465222
3	Sita	Kathmandu	23	4635544
4	Ramhari	Bhairawa	25	4566855
5	Rameshwor	Bhutwal	19	6542534
6	Harka	Kathmandu	31	5302530
7	Sidaram	Pokhara	28	5554560

1. Display name, address and phone of all Students whose name start with 'R'.

SELECT S-name, S-address, phone FROM College WHERE S-name LIKE "R%";

Output

S-name	S-address	phone
Ram	Kathmandu	5562236
Ramhari	Bhairawa	4566855
Rameshwor	Bhutwal	6542534

2. List out id and name of all Students whose name's second letter is 'a'.

SELECT S-id, S-name FROM College WHERE S-name LIKE "-a%";

Output

S-id	S-name
1	Ram
2	Hari
4	Ramhari
5	Rameshwor
6	Harka

3. Select name and phone of all Students whose name ends with letter 'm'.

SELECT S-name, phone FROM College WHERE S-name LIKE "%,m";

Output

S-name	phone
Ram	5562236
Sidaram	5554560

4. Print address of all students whose name's second letter is 'a' and second last letter is 'r'.

`SELECT s-name, s-address FROM college WHERE s-name LIKE '_-a%r';`

Output

s-name	s-address
Han	Pokhara
Ramhari	Bhairawa

5. Display name of all students whose name contains exactly four letters.

`SELECT s-name FROM college WHERE LENGTH(s-name)=4;`

Output

s-name
Han
Sita

6. Display Information of those students whose name start with letter 'H' and contains exactly four letters.

`SELECT * FROM college WHERE s-name LIKE "H---";`

Output

s-id	s-name	s-address	age	phone
2	Han	Pokhara	25	54659222

7. Display information of those students whose name start with letter 'H' or 'S' and contains exactly four letters.

`SELECT * FROM college WHERE s-name LIKE "H---" OR s-name LIKE "S---";`

Output

s-id	s-name	s-address	age	phone
2	Han	Pokhara	25	54659222
3	Sita	Pokhara	23	4635544

8. Display information of those students whose name start with letter 'S' and ends with 'a' and contains exactly four letters.

`SELECT * FROM college WHERE s-name LIKE "S--A";`

Output

S-id	S-name	S-address	age	phone
3	Sita	Pokhara	23	4635544

9. Display name address and age of all students who lives in kathmandu and are atmost 20 years old.

```
SELECT S-name, S-address, age FROM College WHERE S-address = "kathmandu"
AND age >= 20;
```

Output

S-name	S-address	age
Ram	kathmandu	21
Harka	kathmandu	31

10. Display information about all students in Ascending order of age.

```
SELECT * FROM College ORDER BY age;
```

Output

S-id	S-name	S-address	age	phone
5	Rameshwar	Bhutwai	19	6542534
1	Ram	kathmandu	21	5562236
3	Sita	Pokhara	23	4635544
2	Hari	Pokhara	25	5665444
4	Ramhari	Bharawa	25	4566855
7	Sitaram	Pokhara	28	5554560
6	Harka	kathmande	31	5302530

11. Select distinct (not repeated) address from table [i.e. no address should be repeated in list].

```
SELECT DISTINCT S-address FROM College;
```

Output

S-address
Kathmandu
Pokhara
Bharawa
Bhutwai

LABSHEET-5

1. Create Database named "BANK".

```
CREATE DATABASE BANK;
```

2. Create table named Employee (emp-no, emp-name, salary, Dept-no, Job-id).

```
CREATE TABLE Employee (
    emp-no varchar(10) PRIMARY KEY,
    emp-name varchar(25),
    salary decimal(10,2),
    dept-no varchar(10),
    job-id varchar(10),
    FOREIGN KEY (dept-no) REFERENCES Department(dept-no),
    FOREIGN KEY (job-id) REFERENCES Job(job-id));
```

Output

emp-no	emp-name	salary	dept-no	job-id

3. Create table named Department (Dept-no, Dept-name, location).

```
CREATE TABLE Department (dept-no varchar(10) PRIMARY KEY, dept-name
varchar(25), location varchar(25));
```

Output

dept-no	dept-name	location

4. Create table named Job (Job-id, title, duration).

```
CREATE TABLE Job (job-id varchar(10) PRIMARY KEY, title varchar(25),
duration int);
```

Output

job-id	title	duration

5. Insert at least 7 records in each relation.

```
INSERT INTO Job VALUES ("J440", "Worker", 8), ("J441", "Operator", 8),
("J442", "Supervisor", 10), ("J443", "Programmer", 8), ("J444", "Designer", 8);
```

```
INSERT INTO Department VALUES ("A201", "Marketing", "First-Floor"),
("B203", "Finance", "First-Floor"), ("A202", "Sales", "Second-Floor"), ("B204",
"Purchase", "Third-Floor");
```

```
INSERT INTO Employee VALUES ("E101", "Ayush", 50000, "A201", "J444"),
("E102", "Bikash", 75000, "A202", "J441"), ("E103", "Kiran", 90000, "B203", "J442"),
("E104", "Maya", 55000, "A201", "J443"), ("E105", "Neethu", 75000, "B204", "J441"),
("E106", "Monal", 50000, "A202", "J444"), ("E107", "Ayesha", 55000, "B203", "J443");
```

6. Insert 2 records in Employee table. [Insert only ID, name and salary].

```
INSERT INTO Employee (emp_no, emp_name, salary) VALUES ("E108", "Uttam", 90000),
("E109", "Krishna", 50000);
```

7. Insert 2 records in Department. [Insert dept-no and Dept-name].

```
INSERT INTO Department (dept_no, dept_name) VALUES ("C205", "Management"),
("C206", "Training");
```

8. List all the employees and their salary, who are earning more than 10000 and less than 15000.

```
SELECT emp_name, salary FROM Employee WHERE salary > 10000 AND salary < 15000.
```

Output

emp_name	Salary

9. Display name of all employees, whose department name is "Marketing".

SELECT emp-name FROM Employee,Department WHERE Employee.dept-no = Department.dept-no AND Department.dept-name = "Marketing";

Output

emp-name
Ayush
Maya

10. Display salary and name of all employee, whose salary is more than 10000 per month in ascending order in the reference salary.

SELECT emp-name, salary FROM Employee WHERE salary > 10000 ORDER BY salary ASC;

Output

emp-name	Salary
Ayush	50000
Manoj	52000
Krishna	50000
Maya	55000
Ayesha	55000
Bikash	75000
Nitesh	95000
Kiran	90000
Uttam	90000

LABSHEET- 6

1. Display name and salary of employee who is getting maximum salary among all employees.

`SELECT emp_name, salary FROM Employee WHERE salary = (SELECT MAX(salary) FROM Employee);`

Output

emp_name	Salary
Kiran	50000
Uttam	50000

2. Display name, id and salary of employees who is getting minimum salary among all employees.

`SELECT emp_no, emp_name, salary FROM Employee WHERE salary = (SELECT MIN(salary) FROM Employee);`

Output

emp_no	emp_name	Salary
E101	Ayush	50000
E106	Monoj	50000
E109	Krushna	50000

3. Display total number of employee currently working.

`SELECT COUNT(emp_no) FROM Employee WHERE job_id IS NOT NULL;`

Output

COUNT(emp_no)
7

4. Display total number of employee currently working with heading named Total_Employee.

`SELECT COUNT(emp_no) AS Total_Employee FROM Employee WHERE job_id IS NOT NULL;`

Output

Total_Employee
7

5. Display total expenses in salary for Employee with heading named Total-expenses.

SELECT SUM(salary) AS Total-Expenses FROM Employee;

Output

Total-Expenses
590000

6. Display Average Salary of all Employees.

SELECT AVG(salary) FROM Employee;

Output

AVG(salary)
65555.556

7. Display average salary of those employees whose department_id is 'A201'.

SELECT AVG(salary) FROM Employee WHERE dep_no = "A201";

Output

AVG(salary)
52500

8. Display average salary of those employee whose job_id is 'J441'.

SELECT AVG(salary) FROM Employee WHERE job_id = "J441";

Output

AVG(salary)
75000

9. Group all department with department_id, maximum, average and minimum salary.

SELECT dep_no, MAX(salary), AVG(salary), MIN(salary) FROM Employee GROUP BY ~~salary~~ dep_no;

Output

dep_no	MAX(salary)	AVG(salary)	MIN(salary)
[NULL]	90000	70000	50000
A201	55000	52,500	50000
A202	95000	62500	50000
B203	90000	72500	50000
B204	75000	75000	75000

LADSHET - 7

1. Display name of those employees who works in department named 'Sales'.

```
SELECT emp_name FROM Employee, Department WHERE Employee.dept_no = Department.dept_no AND Department.dept_name = "Sales";
```

Output

emp_name
Bikash
Manoj

2. Display name of those employees who works in department named "purchase" and earns more than 15000 per month.

```
SELECT emp_name FROM Employee, Department WHERE Employee.dept_no = Department.dept_no AND Department.dept_name = "purchase" AND salary > 15000;
```

Output

emp_name
Nitesh

3. Display name, salary and department id of all employees who is working as supervisor.

```
SELECT emp_name, salary, dept_no FROM Employee, Job WHERE Employee.job_id = Job.job_id AND Job.title = "Supervisor";
```

Output

emp_name	salary	dep_no
Kiran	90000	B203

4. Display name, salary and location of those employees who work in department named "purchase".

```
SELECT emp_name, salary, location FROM Employee, Department WHERE Employee.dept_no = Department.dept_no AND Department.dept_name = "purchase";
```

Output

emp_name	salary	location
Nitesh	75000	Third-floor

5. Display name salary and location of those employees who works in department named "Purchase" and whose job is "operator".

```
SELECT emp_name, salary, location FROM Employee, Department, Job WHERE
Employee.dept_no = department.dept_no AND Employee.job_id = Job.job_id
AND Department.dept_name = "Purchase" AND Job.title = "Operator";
```

Output

emp-name	Salary	location
Nitesh	75000	Third-Floor

6. Display total Number of Employee Working department wise.

```
SELECT dept_no, COUNT(*) FROM Employee GROUP BY dept_no;
```

Output

dep-no	COUNT(*)
[NULL]	2
A201	2
A202	2
B203	2
B204	1

LABSHEET-8

1. CREATE Database named "HOSPITAL".

CREATE DATABASE HOSPITAL;

2. Create table named Patient (P_id integer primary key, P-name character, dob character, address character, Status, doc_id, des_id).

CREATE TABLE Patient (P_id int PRIMARY KEY,
 P-name varchar(25) NOT NULL,
 dob varchar(10) NOT NULL,
 address varchar(25),
 Status varchar(25),
 doc_id int,
 des_id int,
 FOREIGN KEY(doc_id) REFERENCES Doctor(doc_id),
 FOREIGN KEY(des_id) REFERENCES Disease(des_id));

Output

P_id	P-name	dob	address	Status	doc_id	des_id

3. Create table named Doctor (doc_id primary key integer, doc_name , address, speciality, enroll_date, experience)

CREATE TABLE Doctor (doc_id int PRIMARY KEY,
 doc_name varchar(25),
 speciality varchar(25),
 enroll_date date,
 experience int);

Output

doc_id	doc_name	speciality	enroll-date	experience

4. Create table named Disease (des_id integer primary key, des_name, causes, recovery_date, doc_id)

```
CREATE TABLE Disease (des_id int PRIMARY KEY,
                     des_name varchar(25),
                     causes varchar(25),
                     recovery_date int,
                     doc_id int,
                     FOREIGN KEY (doc_id) REFERENCES Doctor(doc_id));
```

Output

des_id	des_name	causes	recovery_date	doc_id

5. Create table named Hospital (doc_id, salary, dept_id)

```
CREATE TABLE Hospital (doc_id int,
                      salary decimal (10,2),
                      dept_id int,
                      FOREIGN KEY (doc_id) REFERENCES Doctor(doc_id),
                      FOREIGN KEY (dept_id) REFERENCES Department(dept_id));
```

Output

doc_id	salary	dept_id

6. Create table named department (dept_id integer primary key, dept_name, location, position)

```
CREATE TABLE Department (
    dept_id int PRIMARY KEY,
    dept_name varchar(25),
    location varchar(25),
    position varchar(25));
```

Output

dept_id	dept_name	location	position

7. Insert at least 8 records in each relation.

```
INSERT INTO Patient VALUES (100, "Hari", "2060", "KTM", "Normal", 205, 302),
(102, "Shyam", "2050", "BTM", "ICU", 201, 305), (103, "Surya", "2057", "BRT", "Ventilator",
204, 303), (104, "Madan", "2040", "LPT", "Normal", 203, 304) (105, "Basu", "2035",
"PKH", "ICU", 202, 301), (106, "Manoj", "2054", "BTM", "Normal", 205, 302), (107, "Krishna",
"2052", "KTM", "Ventilator", 203, 303);
```

```
INSERT INTO DoctorValues (201, "Fulmatti", "Heart", "2075/10B", 10), (202, "Ranash",
"Liver", "2071/113", 12), (203, "Harka Bhador", "Bone", "2078/3/10", 3), (204,
"Rashana", "Pregnancy", "2075/10B", 10), (205, "Prayatka", "Eye", "2075/10B",
10);
```

```
INSERT INTO DiseaseValues (301, "Jaundice", "Liver", 10, 202), (302, "Bardhi",
"Eye", 3, 205), (303, "Heart Attack", "Heart", 1, 301), (304, "Bone Break", "Bone",
5, 203), (305, "Brah", "Brah", 1, 204);
```

```
INSERT INTO Department VALUES (501, "Surgery", "First Floor", "First"), (502,
"Cardiology", "First Floor", "First"), (503, "Radiology", "First-Floor", "First"),
(504, "Ophthalmology", "Second Floor", "Second"), (505, "Pregnancy", "Second Floor",
"Third");
```

```
INSERT INTO Hospital VALUES (201, 125000, 502), (202, 115000, 501),
(203, 100000, 503), (204, 85000, 505), (205, 95000, 504);
```

8. Insert 2 records in Doctor table. [Insert only ID, name and specialty]

INSERT INTO Doctor(doc_id, doc_name, specialty) VALUES (206, "Hitesh", "Brain"), (207, "Rahul", "Digestion");

9. Insert 2 records in Department table. [Insert only Dept_id and Dept_name].

INSERT INTO Department (dept_id, dept_name) VALUES (SD6, "Account"), (SD7, "Medicine");

LABSHEET - 9

1. Display name and Salary of doctor who is getting maximum salary.

```
SELECT Doctor.doc-name, Hospital.salary FROM Doctor, Hospital WHERE Doctor.doc_id = Hospital.doc_id AND Hospital.salary = (SELECT MAX(salary) FROM Hospital);
```

Output

doc-name	Salary
Fulmati	25000

2. Display total number doctors currently working with heading named Total_Doctor.

```
SELECT COUNT(DISTINCT doc_id) AS Total_Doctor FROM Patient;
```

Output

Total_Doctor
5

3. Display name of doctors currently working in Department named Surgery.

```
SELECT Doctor.doc-name FROM Doctor, Department, Hospital WHERE Doctor.doc_id = Hospital.doc_id AND Hospital.dept_id = Department.dept_id AND Department.dept-name = "Surgery";
```

Output

doc-name
Ramesh

4. Display name and salary of doctor who is getting minimum salary.

```
SELECT Doctor.doc-name, Hospital.salary FROM Doctor, Hospital WHERE Doctor.doc_id = Hospital.doc_id AND Hospital.salary = (SELECT MIN(salary) FROM Hospital);
```

Output

doc-name	Salary
Piyaka	75000

5. List out all doctors who are giving treatment to patient suffering from bird flu.

```
SELECT DISTINCT Doctor.doc-name FROM Doctor, Patient, Disease WHERE
Doctor.doc-id = Patient.doc-id AND Patient.des-id = Disease.des-id AND
Disease.des-name = "Bird Flu";
```

Output

doc-name
Priyaka

6. List all regular patients.

```
SELECT * FROM Patients;
```

Output

p_id	p-name	dob	address	Status	doc_id	des_id
100	Hari	2060	KTM	Normal	205	302
102	Shyam	2050	BTM	ICU	201	305
103	Sita	2057	BRT	Ventilator	204	303
104	Madan	2040	LPT	Normal	203	304
105	Banu	2035	DKH	ICU	202	301
106	Manoj	2054	BTM	Normal	205	302
107	krishna	2002	KTM	Ventilator	203	303

7. What are the cause and recovery date for Jaundice?

```
SELECT causes, recovery-date FROM Disease WHERE des-name = "Jaundice";
```

Output

causes	recovery-date
Liver	10

8. Delete position column from department column.

```
ALTER TABLE Department DROP COLUMN position;
```

Output

dept_id	dept_name	location

LABSHEET-10

1. Increase salary of all employees by 16% of their salary.

UPDATE Hospital SET Salary = Salary + Salary * 0.16;

Output

doc_id	Salary	dept_id
201	145000	S02
202	133400	S01
203	116000	S03
204	98600	S05
205	87000	S04

2. Decrease salary of all doctors by 10% of their salary who name is "Fulmatr".

UPDATE Hospital SET Salary = Salary - Salary * 0.1 WHERE doc_id = (SELECT doc_id FROM Doctor WHERE doc_name = "Fulmatr");

Output

doc_id	Salary	dept_id
201	130500	S02
202	123400	S01
203	116000	S03
204	98600	S05
205	87000	S04

3. Decrease salary of all doctors by 15% of their salary who name starts from "Ra".

UPDATE Hospital, Doctor SET Salary = Salary - Salary * 0.15 WHERE Hospital.doc_id = Doctor.doc_id AND Doctor.doc_name LIKE "Ra%";

Output

doc_id	Salary	dept_id
201	130500	S02
202	120060	S01
203	116000	S03
204	88740	S05
205	87000	S04

4. Increase salary of all doctors by 5000 whose salary is less than 1lakh but decrease all other's by 7500.

UPDATE Hospital SET Salary = CASE
 WHEN Salary < 100000 THEN Salary + 50000
 ELSE Salary - 75000
 END;

Output

doc_id	Salary	dept_id
201	55,500	S02
202	45060	S01
203	41000	S03
204	138740	S05
205	137000	S04

5. Display name of all Doctors, whose name is not in Patient list.

SELECT docname FROM Doctor WHERE doc_name NOT IN (SELECT D-name FROM Patient);

Output

docname
Furmati
Ramesh
Karla Bahadur
Rashana
Dnyaka
Hitesh
Rahul

Labsheet - 11

1. Display name of any 3 patient who are getting treatment Under Dr. Rasthana.

```
SELECT Patient.p-name FROM Patient, Doctor WHERE Patient.doc_id = Doctor.doc_id AND Doctor.doc_name = "Rasthana" LIMIT 3;
```

Output

p-name
Sanya

2. List out name of any 5 doctor who are heart Specialist and has more than 12 years experience.

```
SELECT docname FROM Doctor WHERE speciality = "Heart" AND experience >= 11 LIMIT 5;
```

Output

docname
Fulmati

3. Display name of all doctor who are not working in Surgery Department.

```
SELECT docname FROM Doctor, Department, Hospital WHERE Doctor.doc_id = Hospital.doc_id AND Department.dept_id = Hospital.dept_id AND Department.dept_name <> "Surgery";
```

Output

docname
Fulmati
Hanca Bhadur
Rasthana
Myaka

4. Display name of any 3 doctors who are not working in Surgery department.

```
SELECT doc_name FROM Doctor, Department, Hospital WHERE
Doctor.doc_id = Hospital.doc_id AND Department.dept_id = Hospital.dept_id AND
Department.dept_name <> "Surgery" LIMIT 3;
```

Output

doc_name
Fulmety
Harka Shadur
Rashana

5. SELECT name of all doctors who is getting more than average salary and his/her name doesn't end with letter 'esh'.

```
SELECT Doctor.doc_name FROM Doctor, Hospital WHERE Doctor.doc_id =
Department.doc_id AND Doctor.doc_name NOT LIKE "%esh" AND
Hospital.salary > (SELECT AVG(salary) FROM Hospital);
```

Output

doc_name
Rashana
Dnyaka

LABSHEET-12

Create the following Tables:

location

location-1d	Regional-group
122	New York
123	Dallas
124	Chicago
167	Boston

Department

department-1d	Name	location-1d
10	Accounting	122
20	Research	124
30	Sales	123
40	Operation	167

Job

Job-1d	Function
667	Clerk
668	Staff
669	Analyst
670	Salesperson
671	Manager
672	President

Employee

employee-1d	Last-name	First-name	Middle-name	Job-1d	Manager-1d	Hire-Date
7369	Smith	John	Q	667	7902	17-Dec-84
7499	Allen	Kevin	J	670	7698	20-Feb-85
7505	Doyle	Jean	K	671	7839	4-Apr-85
7506	Demos	Lynn	S	671	7839	15-May-85
7507	Baker	Leslie	D	671	7839	10-Jun-85
7521	Wang	Cynthia	D	670	7698	22-Feb-85

Salary	Comm	Department-1d
800	NULL	20
1600	300	30
2850	NULL	30
2750	NULL	30
2200	NULL	40
2500	500	30

1. List all the employee details.

SELECT * FROM Employee;
Output

employee-1d	Last-name	First-name	Middle-name	Job-1d	Manager-1d	Hire-Date	Salary	Comm	Department-1d
7369	Smith	John	Q	667	7902	17-Dec-84	800	NULL	20
7499	Allen	Kevin	J	670	7698	20-Feb-85	1600	300	30
7505	Doyle	Jean	K	671	7839	4-Apr-85	2850	NULL	30
7506	Demos	Lynn	S	671	7839	15-May-85	2750	NULL	30

7507	Bakir	Ledzie	D	671	7839	10-Jun-85	2200	NULL	40
7521	Worke	Cynthia	D	670	7698	22-Feb-85	1250	500	30

2. List all the department Details.

SELECT * FROM Department;

Output

department_id	name	location_id
10	Accounting	122
20	Research	124
30	Sales	123
40	Operations	167

3. List all Job details.

SELECT * FROM Job;

Output

Job_id	Function
667	Clerk
668	Staff
669	Analyst
670	Salesperson
671	Manager
672	President

4. List all the locations.

SELECT * FROM Location;

Output

location_id	regional_group
122	New York
123	Dallas
124	Chicago
167	Boston

5. List out first name, last name, salary, commission for all employees.

SELECT first_name, last_name, salary, comm FROM Employee;

Output

first-name	last-name	Salary	comm [NULL]
John	Smith	800	
Karen	Allen	1600	300
Jean	Doyle	2650	[NULL]
Lynn	Dorothy	2750	[NULL]
Leslie	Baker	2200	[NULL]
Cynthia	Wark	1250	300

6. List out employee_id, last_name, department_id for all employees and rename employee_id as "ID of the employee", last_name as "Name of Employee", department_id as "department ID".

SELECT employee_id AS "ID of the Employee", last_name AS "Name of the Employee", department_id AS "Department ID" FROM Employee;

Output

ID of the Employee	Name of the Employee	Department ID
7369	Smith	20
7499	Allen	30
7505	Doyle	30
7506	Dorothy	30
7507	Baker	40
7521	Wark	30

7. List out the employees annual salary with their name only.

SELECT first_name, salary FROM Employee;

Output

first_name	salary
John	800
Karen	1600
Jean	2650
Lynn	2750
Leslie	2200
Cynthia	1250

8. List the details about 'SMITH'.

`SELECT * FROM Employee WHERE last_name = "Smith";`

Output

employee_id	last_name	first_name	middle_name	job_id	manager_id	hire_date	salary	comm	department_id
7369	Smith	John		667	7902	12-Dec-84	800	NULL	20

9. List out the employees who are working in department 20.

`SELECT first_name FROM Employee WHERE department_id = 20;`

Output

first_name
John

10. List out the employees who are working in department 10 or 20.

`SELECT first_name FROM Employee WHERE department_id IN (10, 20);`

Output

first_name
John

11. Find out the employees who are not working in department 10 or 30.

`SELECT first_name FROM Employee WHERE department_id NOT IN (10, 30);`

Output

first_name
John
Leagle

12. List out the employees whose name start with "S" and end with "H".

`SELECT first_name, last_name FROM Employee WHERE last_name LIKE "S%H";`

Output

first_name	last_name
John	Smith

13. List out the employees who are not receiving Commission.

SELECT first_name FROM Employee WHERE comm IS NULL;

Output

first_name
John
Jean
Lynn
Leslie

14. List out the employee id, last name in ascending order based on the employee id.

SELECT employee_id, last_name FROM Employee ORDER BY employee_id ASC;

Output

employee_id	last_name
7369	Smith
7499	Allen
7505	Doyle
7506	Dennis
7507	Baker
7521	Wark

15. List out the employee id, name in descending order based on salary column.

SELECT employee_id, last_name FROM Employee ORDER BY salary DESC;

Output

employee_id	last_name
7369	Smith
7499	Allen
7505	Doyle
7506	Dennis
7507	Baker
7521	Wark

16. List out the employee details according to their last_name in ascending order and then on department_id in descending order.

SELECT first_name, last_name, department_id FROM Employee ORDER BY last_name ASC, department_id DESC;

Output

first_name	last_name	department_id
Kevin	Allen	30
Leanne	Baker	40
Lynn	Dennise	30
Jean	Doyle	30
John	Smith	20
Lynne	Wark	30

17. How many employees who are working in different departments were in the organization.

SELECT department_id, count(*) FROM Employee GROUP BY department_id;

Output

department_id	count(*)
20	1
30	4
40	1

18. List out the department wise maximum salary, minimum salary, average salary of the employees.

SELECT department_id, MAX(salary), MIN(salary), AVG(salary) FROM Employee GROUP BY department_id;

Output

department_id	MAX(salary)	MIN(salary)	AVG(salary)
20	800	800	800
30	2850	1250	2112.5
40	2200	2200	2200

19. List out the job wise maximum salary, minimum salary, average salaries of the employees.

SELECT job_id, MAX(salary), MIN(salary), AVG(salary) FROM Employee GROUP BY job_id;

Output

job_id	MAX(salary)	MIN(salary)	Avg(salary)
669	800	800	800
670	1600	1250	1425
671	2850	2200	2600

20. List out the department id having at least four employees.

`SELECT department_id, count(*) FROM Employee GROUP BY department_id HAVING count(*) >= 4;`

Output

department_id	count(*)
30	4

21. Display the employee who got the maximum salary.

`SELECT first_name FROM Employee WHERE salary = (SELECT MAX(salary) FROM Employee);`

Output

first_name
Jean

22. Display the employees who are working in sales department.

`SELECT first_name FROM Employee WHERE department_id = (SELECT department_id FROM Department WHERE name = "Sales");`

Output

first_name
Kevin
Jean
Lynn
Cynthia

23. Display the employees who are working as "clerk".

`SELECT first_name FROM Employee WHERE job_id = (SELECT job_id FROM Job WHERE function = "clerk");`

Output

first_name
John

24. ~~Display~~ Display the employees who are working in "New York".

`SELECT first_name FROM Employee, Department, Location WHERE Employee.department_id = Department.department_id AND Department.location_id = Location.location_id AND Location.region_group = "New York";`

25. Update the employees salaries, who are working as Clerk on the basis of 10%.

`UPDATE Employee SET salary = salary + salary * 0.1 WHERE job_id = (SELECT job_id FROM Job WHERE function = 'Clerk');`

26. Delete the employees who are working in account department.

`DELETE FROM Employee WHERE department_id = (SELECT department_id FROM Department WHERE name = 'Accounting');`

27. Display the second highest salary drawing employee details.

`SELECT first_name, salary FROM Employee ORDER BY salary DESC LIMIT 1;`

Output

first_name	Salary
Lynn	2750

28. List out the employees who earn more than every employee in department 30.

`SELECT first_name FROM Employee WHERE salary > ALL (SELECT salary FROM Employee WHERE department_id = 30);`

29. List out the employees who earn more than the lowest salary in department 30.

`SELECT first_name FROM Employee WHERE salary > ANY (SELECT salary FROM Employee WHERE department_id = 30);`

Output

first_name
Kevin
Jean
Lynn
Terrie

30. Find out who department has no employees.

`SELECT name FROM Department d WHERE NOT EXISTS (SELECT * FROM Employee e WHERE d.department_id = e.department_id);`

Output

name
Accounting

31. List out employees with their department names.

SELECT first_name, Department.name FROM Employee, Department WHERE Employee.department_id = Department.department_id;

Output

first-name	name
John	Research
Kevan	Sales
Jean	Sales
Lynn	Sales
Leslie	Operations
Cynthia	Sales

32. Display employees with their designations (Jobs)

SELECT first_name, Job.function FROM Employee, Job WHERE Employee.job_id = Job.job_id;

Output

first_name	function
John	clerk
Kevan	Salesperson
Jean	Manager
Lynn	Manager
Leslie	Manager
Cynthia	Salesperson

33. Display the employee with their department name and regional groups.

SELECT first_name, Department.name, Location.regional_group FROM Employee, Department, Location WHERE Employee.department_id = Department.department_id AND Department.location_id = Location.location_id;

Output

first-name	name	regional-group
John	Research	Chicago
Kevan	Sales	Dallas
Jean	Sales	Dallas
Lynn	Sales	Dallas
Leslie	Operations	Dallas
Cynthia	Sales	Boston

34. How many employees who are working in different departments and display with department name.

`SELECT Department.name, COUNT(*) FROM Employee, Department WHERE Employee.department_id = Department.department_id AND Department.name GROUP BY Department.name;`

Output

name	Count(*)
Operations	1
Research	1
Sales	4

35. How many Jobs in the organization with Designations.

`SELECT COUNT(DISTINCT Employee.job_id) FROM Employee, Job WHERE Employee.job_id = Job.job_id AND Job.function IS NOT NULL;`

Output

COUNT(DISTINCT Employee.job_id)
3

36. How many employees Working in "New York"

`SELECT COUNT(*) FROM Employee, Department, Location WHERE Employee.department_id = Department.department_id AND Department.location_id = Location.location_id AND Location.location_id = "New York";`

Output

COUNT(*)
0

37. List out the distinct jobs in Sales and Account Departments.

`SELECT DISTINCT Department.name, Job.function FROM Job, Employee, Department WHERE Employee.department_id = Department.department_id AND Employee.job_id = Job.job_id AND Department.name IN ("Sales", "Accounting");`

Output

name	function
Sales	Salesperson
Sales	manager

38. List out the All jobs in Sales AND Accounting Departments.

```
SELECT Department.name, Job.function FROM Job J, Employee e, Department d WHERE e.department_id = d.department_id AND Employee.job_id = J.job_id AND d.name IN ("Sales", "Accounting");
```

Output

name	function
Sales	Salesperson
Sales	Manager
Sales	Manager
Sales	Salesperson

39. List out the common jobs in Research and Accounting Departments in ascending orders

```
SELECT dname, function FROM Job j, Employee e, Department d WHERE e.department_id = d.department_id AND e.job_id = j.job_id AND d.name IN ("Research", "Accounting") ORDER BY function ASC;
```

Output

name	function
Research	Clerk