

DATE:

Database Management System

[MPTDQ]

Name: Sabiha Naaz Salamullah Khan

Stream: SYBCA

Faculty: Prof. Archana Patil

College: Brin's Sangha, Kurla East,
Mumbai.

INDEX

Practical No.	Activities	Date	Sign
1.	To draw ER Model and Relational Model for a given database.	21-08-23	①
2.	Create one-to-many Relationship between Manager and Employee Relations.	04-11-23	
3.	Determine the functional dependencies. Remove partial dependency and transitive dependencies in given table.	04-11-23	
4.	Create of Database and table DDL command	24-08-23	
5.	Simple SQL Query -1- DML command.	31-08-23	
6.	Simple SQL Query-2- SQL functions	31-08-23	
7.	Advanced SQL queries using Set Operations.	04-10-23	
8.	Advanced SQL query using set query query	19-10-23	
9.	Advanced SQL queries using JOINS.		
10.	Advanced SQL queries using PL-SQL.	09-11-23	

Practical I (a) :-

Aim: To draw ER Model and Relational Model for a given database.

a. Case Study 1: List the data requirements for the database of the company which keeps track of the company employee, department and projects. The database designers provide following descriptions.

Relational Model:-

Relation - Department

D-No.	D-Name	Start Date	D-Location
10	Director	12-07-2021	Chembur
11	Manager	2-1-2023	Kurla

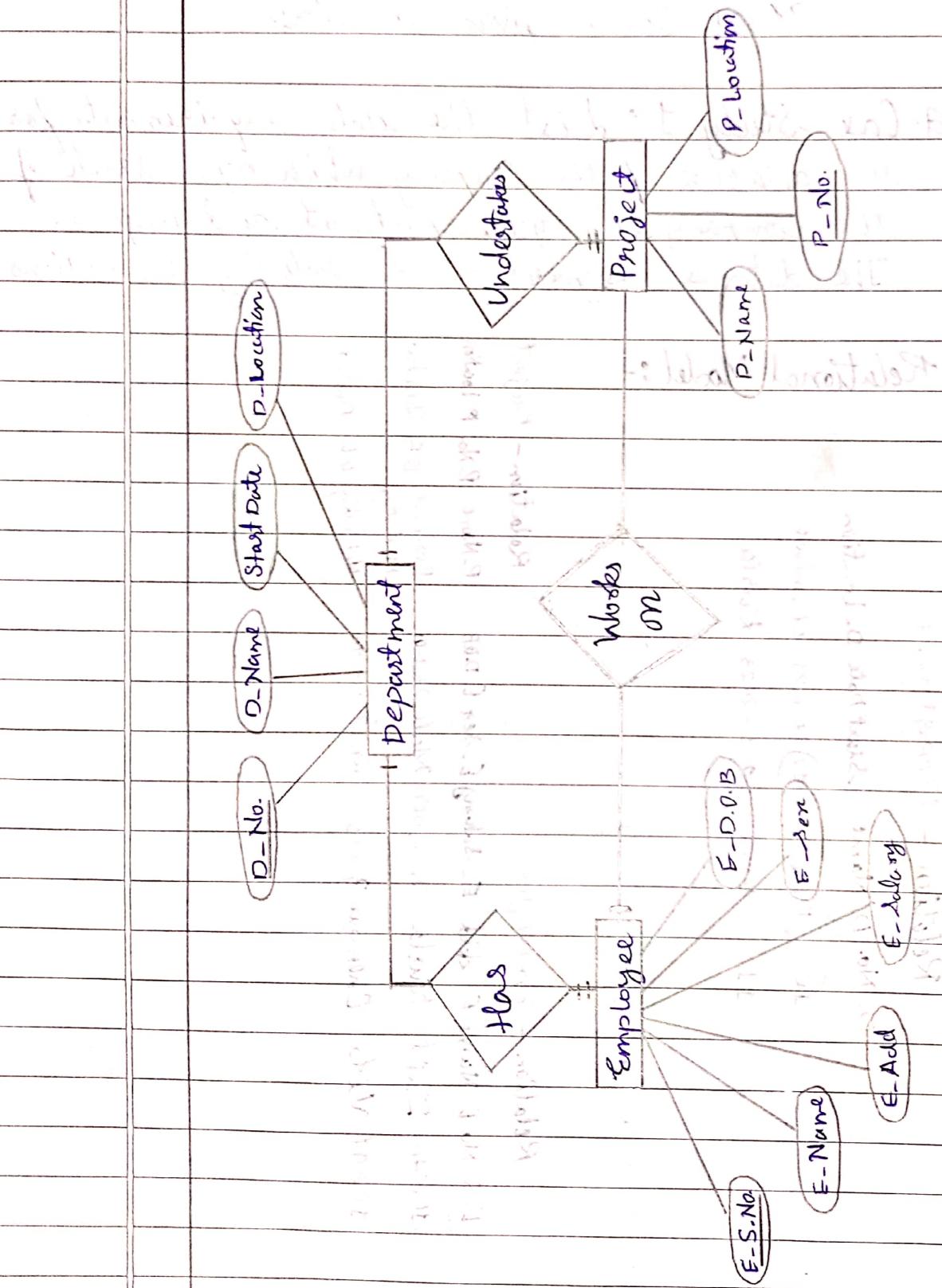
Relation - Employee

E-S-No	E-Name	E-Dob	E-Salary	E-Age
102046	Zaid	Kurla	2,00,000	Male 23-9-91
102047	Vida	Chembur	3,00,000	Female 5-4-94

Relation - Project

P-No.	P-Name	P-Location
01	Business	Chembur
02	Marketing	Kurla
03	Networking	Mumbai
04	Design	Karachi
05	Dades	Delhi

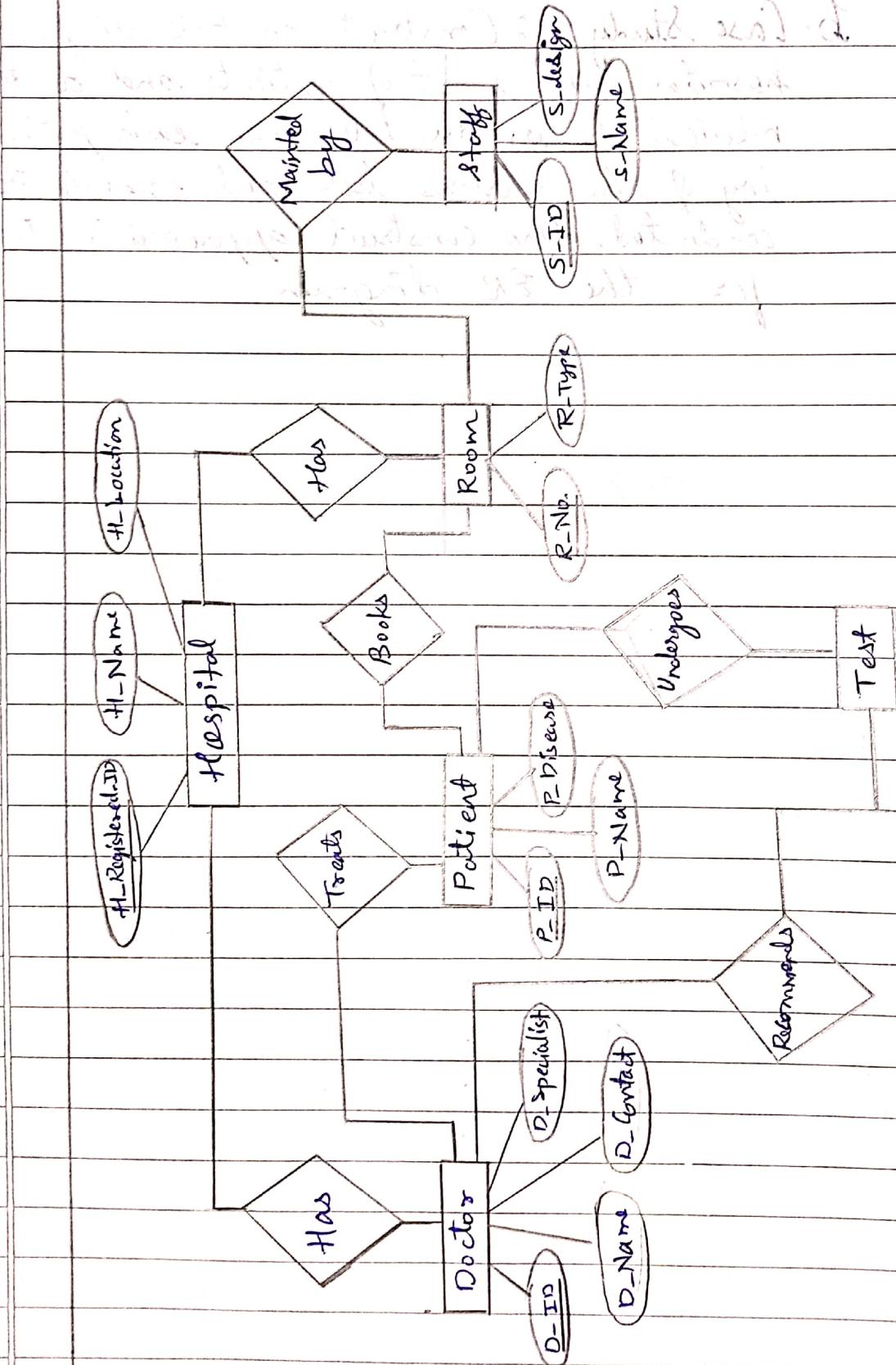
ER Model :-



Practical 1(b):

b. Case Study 2: Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examination conducted. Also construct appropriate tables for the ER diagram

ER Model :-



Practical 2 :

Aim : Create one-to-many Relationship between Manager and Employee Relations.

Question : Create following relations with the given fields.

a) EMPLOYEE

EmpId (PK),

EmpName (Should be in the upper case)

Department (Should be Finance, Purchase or Sales).

Salary
Mgrid.

b) MANAGER

Mgrid

MgrName,

No. of Employees controlled.

Program:

Create table EMPLOYEE (

EmpId INT PRIMARY KEY,

EmpName VARCHAR(50) UPPER,

Department VARCHAR(20),

Salary DECIMAL (10, 2),

Mgrid INT,

FOREIGN KEY (MgrId) REFERENCES.
MANAGER (MgrId)
);

Create table MANAGER (
MgrId INT PRIMARY KEY,
MgrName VARCHAR(50),
No.ofEmployeesControlled INT
);

a. Display details of all those employees whose salary is higher than Rs. 50

Select * from EMPLOYEE Where Salary > 50 ;

b. Display the details of employees who are working in Purchase department.

Select * from EMPLOYEE Where Department = 'Purchase' ;

Practical 3 :

Aim: Determine the functional dependencies.

Question: Remove partial dependency and transitive dependencies in given table. (i.e. convert it into 3NF).

Student = (RollNo, Course_Code,
Course_Name, Fees)

Answer:

Student_data :

RollNo	Name
123	Ravi
123	Ravi
123	Ravi
124	Sumit
124	Sumit
125	Trypta
125	Trypta
125	Trypta

DATE:

Course data :

RollNo	Course_code	CourseName
123	C102	C
123	C103	C++
123	C104	OOPS
124	C102	C
124	C103	C++
125	C102	C
125	C103	C++
125	C104	OOPS

Fee data :

Course_code	CourseName	Fees
123	C	2500
123	C++	1200
123	OOPS	3200
124	C	2500
124	C++	1200
125	C	2500
125	C++	1200
125	OOPS	3200.

Practical 4 (a):

Quesn: Creation of Database and table - DDL
COMMAND:

Question: Create a table called EMP with the following structure.

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

Allow NULL for all columns except ENAME and JOB.

Program:-

Create table EMP01(empno number(6),
 ename varchar(20) NOT NULL,
 job varchar(10) NOT NULL,
 deptno number(3),
 sal number(7, 2));

a. Add a column experience to the emp table.
 Experience numeric NULL allowed.

alter table EMP01 ADD experience int;

b. Modify the column width of the job field of emp table.

`alter table EMP01 MODIFY job varchar(20);`

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
101	SCOTT	ANALYST	7782	1980-12-17	3000	0
102	KING	MANAGER	0	1980-11-17	5000	0
103	ADAMS	CLERK	101	1981-01-12	1200	0
104	JONES	MANAGER	201	1981-04-01	2900	0
105	WILLIAMS	CLERK	101	1982-02-17	1200	0
106	MARTIN	SALESREP	101	1981-09-28	1200	1400
107	BLAKE	MANAGER	201	1981-05-01	2500	0
108	CANTRELL	CLERK	101	1982-02-01	1200	0

2. Modify column job of emp with data type.

-> mapping

(a) Insert employees into table

EMP01 and alter column

job type set to char(10)

and change width

of job column to 10

• insert employees into table

emp01 from oracle database

the existing job name did not exist

Practical 4 (b) :

Question : Create dept table with the following structure.

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

DEPTNO as the primary key.

Program :-

Create table DEPT01 (deptno number(2) primary key,
Dname varchar(10),
LOC varchar(10));

a. Create the emp1 table with ename and empno.

Create table EMPNEW1 as select empno, ename from
EMP01;

b. Drop a column experience to the emp table.

alter table EMP01 drop column experience;

Practical 5(a):

Aim: Simple SQL Query - I - DML COMMAND

Question: Write syntax of all DML Command.

Answer:

Here are the syntaxes for Data Manipulation Language (DML) commands in SQL:

1. SELECT :- Retrieve data from a database.

Syntax:

SELECT DISTINCT column_name(s) FROM
table name.

2. INSERT :- Add new records into a table.

Syntax:

INSERT INTO table_name [(Col1, Col2, ...)]
VALUES (Expression1, Expression2, ...)

3- UPDATE :- Modify existing records in a table.

Syntax :-

UPDATE table-name

SET Col1 = expression1, Col2 = expression2, ...
[WHERE expression]

4. DELETE :- Remove records from a table.

Syntax :-

DELETE FROM table-name [WHERE expression]

Practical 5(b):

Question: Create database where create following tables:

Emp(EMPNO int, ENAME VARCHAR(20), JOB
VARCHAR(10), DEPNO int, SAL numeric(7,2))

allow NULL for all columns except
ename and job.

Dept(DEPTNO int, DNAME VARCHAR(10),
LOC VARCHAR(10))

Deptno as the primary key.

Insert at least 8 records into tables and
solve the following given queries using
Emp and Dept Table.

Program:

Create table emp(EMPNO int,
ENAME varchar(20) NOT NULL,
JOB varchar(10) NOT NULL,
DEPTNO int,
SAL numeric(7,2));

Create table dept (DEPTNO int PRIMARY KEY,
 DNAME varchar(10),
 LOC varchar (10));

insert into emp values(101, 'Shanti Singhania',
 'Manager', 1052, 50000);

insert into emp values(102, 'Yahya Khan',
 'Programmer', 1056, 70000);

insert into emp values(103, 'Sudha Murti',
 'ASP', 1056, 60000);

insert into emp values(104, 'Priya Gaitwad',
 'Clerk', 1052, 10000);

insert into emp values(105, 'Anuj Kapadia',
 'Lecturer', 1059, 35000);

insert into emp values(106, 'Parth Goyal',
 'ASP', 1056, 55000);

insert into emp values(107, 'Arti Singh',
 'Manager', 1052, 55000);

insert into emp values(108, 'Shubham Gupta',
 'Lecturer', 1059, 30000);

a. Create the emp1 table with ename and empno,
 add constraints to check the empno value
 while entering (i.e) empno > 100.

Create table emp1 as select EMPNO, ENAME from emp where EMPNO > 100;
--

b. Update the emp table to set the salary of all employees to Rs.15000/- who are working as ASP.

```
update emp set SAL = 15000  
where JOB = 'ASP';
```

c. Delete only those who are working as lecturer.

```
delete from emp  
where JOB = 'Lectures';
```

d. List the records in the emp table orderby salary in ascending order.

```
select ENAME, SAL from emp  
ORDER BY SAL ASC;
```

e. Display total salary spent for each job category.

```
select sum(SAL) from emp;
```

f. Add constraints to the emp table that empno as the primary key and deptno as the foreign key.

Alter table emp
ADD CONSTRAINT pk-emp PRIMARY KEY
(EMPNO),
ADD CONSTRAINT fk-emp-dept FOREIGN KEY
(DEPTNO) REFERENCES Dept (DEPTNO);

Q. Add columns DOB to the emp table.

alter table emp ADD DOB date ;

Practical 6 (a) :

Aim: Simple SQL Query 2: SQL Functions.

Question: List all the aggregate functions with example.

Answer:

Aggregate functions in DBMS perform calculations on sets of values and return a single result.

Here is the list of all aggregate functions with example :-

- Consider Table Student as follows :-

RollNo	Name	Amount	Percentage
1021	Sachin	15000	55.50
1022	Sandesh	20000	73.20
1023	Kavita	15000	75.00
1024	Ravi	25000	60.80
1025	Kunal	18000	63.45

1. AVG() :- Returns the average value.

Example:-

SELECT AVG(Amount) FROM Student;

2. COUNT :- Returns the number of rows.

Example :

SELECT COUNT (Amount) FROM student;

OR

SELECT COUNT(*) FROM student;

3. FIRST() :- Returns the first value.

Example :

SELECT FIRST (Name) FROM Student;

4. LAST() :- Returns the last value.

Example :

SELECT LAST (Name) FROM Student;

5. MAX() :- Returns the largest value.

Example :

SELECT MAX (Amount) FROM student;

6. MIN() :- Returns the smallest value.

Example :

```
SELECT MIN(Amount) FROM student;
```

7. SUM() :- Returns the sum.

Example :

```
SELECT SUM(Amount) FROM student;
```

8. NOW() :- Returns the current system date and time.

Example :

```
SELECT RollNo, amount, NOW() as Date  
FROM student WHERE RollNo = 1021
```

9. ROUND() :- Used to round a numeric field to the number of decimals specified.

Example :

```
SELECT RollNo, ROUND(Percentage, 0) as Percentage  
FROM student WHERE RollNo = 1023.
```

10. FORMAT() :- Used to format how a field is to be displayed.

Example :-

```
SELECT RollNo, Amount, FORMAT(NOW(), 'YYYY-MM-DD') as Date  
FROM Student WHERE RollNo = 1024.
```

Practical 6 (b) :

Question: List all the string functions with example.

Answer:

String functions in DBMS are mainly used to perform different types of string manipulation.

- Consider the following Students Table for string operation:

Roll No	Last Name	First Name	Address	City
1	Pawar	Ramesh	Bytco	Nashik
2	Patil	Sachin	Juhu	Mumbai
3	Shah	Nayana	Nashik Road	Nashik
4	Hussein	Zakir	Cidco	Nashik
5	Jain	Kavita	Shivajinagar	Pune
6	Mishra	Ravi	Pune Station	Pune

1. Substring :- Used to grab a piece of the stored data

Example:

SELECT SUBSTR(Last-Name, 3)

FROM Students WHERE Last-Name = 'Hussein';

2. TRIM :- Used to remove particular prefix or suffix from a string.

Example :-

1. SELECT TRIM('Sample');

2. SELECT LTRIM(' Sample');

3. SELECT RTRIM('Sample');

3. LENGTH :- Used to get the length of the string.

Example :-

SELECT Length(Last_Name) FROM Students
WHERE Last_Name = 'Pawar';

4. Replace :- Replace all occurrences of string 2 in the string 1 with string 3.

Example :-

SELECT REPLACE('Last-Name', 't-l', 'tel')
FROM Students WHERE Last_Name = 'Patil';

5. CONCATENATE :- Combines the results from several fields.

Example :-

```
SELECT CONCAT(Last_Name, First_Name)  
FROM students WHERE Last_Name = 'Hussein';
```

6. LEFT :- Returns left part of a string with the specified number of characters.

Example :-

```
SELECT LEFT('Last_Name', 3)  
FROM students WHERE Last_Name = 'Shah';
```

7. RIGHT :- Returns right part of a string with the specified number of characters.

Example :-

```
SELECT RIGHT('Last_Name', 3)  
FROM students WHERE Last_Name = 'Shah';
```

8. **REPLICATE** :- Repetts string for specified number of times.

Example :

```
SELECT REPLICATE('Last_Name', 2)
FROM Students WHERE Last_Name = 'Shah';
```

9. **REVERSE** :- Returns reverse of a string.

Example :

```
SELECT REVERSE('Last_Name')
FROM Students WHERE Last_Name = 'Shah';
```

10. **UPPER** :- Converts string to Uppercase.

Example :

```
SELECT UPPER('Last_Name')
FROM Students WHERE Last_Name = 'Pawar';
```

11. **LOWER** :- Converts string to Lowercase.

Example :

```
SELECT LOWER('Last_Name')
FROM Students WHERE Last_Name = 'Shah';
```

12. UNICODE :- Returns Unicode standard integer value.

Example :

```
SELECT UNICODE ('Last-Name')
FROM Students WHERE Last-Name = 'Shah';
```

13. STUFF :- Deletes a specified length of characters
and inserts string at a specified starting index.

Example :

```
SELECT STUFF('Last-Name', 3, 3, 'tel')
FROM Students WHERE Last-Name = 'Patil';
```

14. MID :- Function is used to extract characters
from a Text Field.

Example :

```
SELECT MID(First-Name, 1, 3) FROM Students
WHERE First-Name = 'Ramesh';
```

Practical 6 (c) :

Question: Using above Emp table solve the following queries:

Program:

Select * from emp;

Table:

EMPNO	ENAME	JOB	DEPTNO	SAL
101	Shanti Singhania	Manager	1052	50000
102	Yahya Khan	Programmer	1056	70000
103	Sudha Mwiti	ASP	1056	60000
104	Priya Gaikwad	Clerk	1052	30000
105	Anuj Kapadia	Lecturer	1059	35000
106	Parth Goyal	ASP	1056	55000
107	Aarti Singh	Manager	1052	55000
108	Shubham Gupta	Lecturer	1059	30000

- a. Display all the details of the records whose employee name starts with 'A'.

Select * from emp where ENAME LIKE 'A%';

b. Display all the details of the records whose employee name does not start with 'A'.

select * from emp where ENAME NOT LIKE '%.A%';

c. Calculate the total and average salary amount of the emp table.

Select sum(SAL), avg(SAL) from emp;

d. Determine the max and min salary and rename the column as max_salary and min_salary.

Select max(SAL) as max_salary, min(SAL) as min_salary from emp;

e. Find how many job titles are available in employee table.

Select count(JOB) from emp;

f. Count the total records in the emp table

Select count(*) from emp;

Practical 7(a):

Aim: Advanced SQL queries using set operations.

Question: List all the set operators.

Answer:

SQL set operators allows to combine results from two or more SELECT statements. It combines rows from different queries.

The following are the set operators :-

1. UNION Operator :- Combines the result sets of two or more SELECT statements into a single result set, eliminating duplicate rows.

2. UNION ALL Operator :- SELECTs only distinct values by default. To allow duplicate values, use UNION ALL.

3. INTERSECT Operator :- Returns only the rows that are common to the result sets of two or more SELECT statements.

DATE:

4. MINUS or EXCEPT Operator :- Returns rows that are present in the first result set but not in the second result set.

Practical 7 (b) :

Questions: Using above Emp table solve the following queries.

Program:

Select * from emp;

Table:

EMPNO	ENAME	JOB	DEPTNO	SAL
101	Shwanti Singhania	Manager	1052	50000
102	Yahya Khan	Programmer	1056	70000
103	Sudha Murti	ASP	1056	60000
104	Priya Gaikwad	Clerk	1052	10000
105	Arij Kapadia	Lecturer	1059	35000
106	Divyanshu Goyal	ASP	1056	55000
107	Darti Singh	Manager	1052	55000
108	Shubham Gupta	Lecturer	1059	30000

a. Display all the dept numbers available with the dept and emp tables avoiding duplicates

```
SELECT . . . DEPTNO from emp
UNION
SELECT DEPTNO from Dept;
```

b. Display all the dept numbers available with the dept and emp tables.

```
SELECT DEPTNO from emp  
UNION ALL  
SELECT DEPTNO from dept;
```

c. Display all the dept numbers available in emp and not in dept tables and vice versa.

```
i. SELECT DEPTNO from emp  
EXCEPT  
SELECT DEPTNO from dept;
```

```
ii. SELECT DEPTNO from dept  
EXCEPT  
SELECT DEPTNO from emp;
```

Practical 8 :

Xim: Advanced SQL queries using Sub query.

Question: Using above Emp table solve the following queries.

Program:

```
select * from emp;
```

Table:

EMPNO	ENAME	JOB	DEPTNO	SAL
101	Shweta Singhania	Manager	1052	50000
102	Tahya Khan	Programmer	1056	70000
103	Sudha Murti	ASP	1056	60000
104	Priya Gaikwad	Clerk	1052	10000
105	Anuj Kapadia	Lecturer	1059	35000
106	Arjun Goyal	ASP	1056	55000
107	Darti Singh	Manager	1052	55000
108	Shubham Gupta	Lecturer	1059	30000

- Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with 'M'.

Select ENAME, SAL from emp
where SAL > (select min(SAL) from emp)
AND JOB LIKE 'M%';

b. Issue a query to find all the employees
who work in the same job as Arjun.

Select * from emp
where JOB = (select JOB from emp
where ENAME = 'Arjun Goyal');

c. Issue a query to display information about
employees who earn more than any
employee in dept.

Select * from emp
where SAL = (select MAX(SAL)
from emp);

Practical 9(a) :

Ques: Advanced SQL query using JOINS.

Ques: What is joins? List types of joins with syntax.

Answer:

- A join is a database operation that combines rows from two or more tables based on a related column between them.
- Joins are used to retrieve data from multiple tables by matching records that have a common key or attribute.
- There are several types of joins in DBMS including :-

1. INNER JOIN : This join returns all records from multiple tables that satisfy the specified join condition.

Syntax:

SELECT columns FROM table1

INNER JOIN table2 ON condition1

INNER JOIN table3 ON condition2 ;

2. OUTER JOIN : This join returns all records from both tables that satisfy the join condition.

- We can categorize the OUTER JOIN further into three types :

i. LEFT OUTER JOIN : This join retrieves all the records from the left table and matching rows from the right table.

Syntax :

```
SELECT column-lists
FROM table1
LEFT [OUTER] JOIN table2
ON table1.column = table2.column ;
```

ii. RIGHT OUTER JOIN : This join retrieves all the records from the right-hand table and matched rows from left-hand table.

Syntax :

```
SELECT column-lists
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column ;
```

iii FULL OUTER JOIN : This join returns a result that includes all rows from both tables.

Syntax :

```
SELECT column-lists  
FROM table1  
FULL [OUTER] JOIN table2  
ON table1.column = table2.column ;
```

Practical 9 (b):

Questions Using above Emp table solve
the following queries.

Program:

Select * from emp;

Table:

EMPNO	ENAME	JOB	DEPTNO	SAL
101	Shanti Singhania	Manager	1052	50000
102	Yahya Khan	Programmer	1056	40000
103	Sudha Murti	ASP	1056	60000
104	Priya Gaikwad	Clerk	1052	10000
105	Arijit Kapadia	Lecturer	1059	35000
106	Ajum Goyal	ASP	1056	55000
107	Aarti Singh	Manager	1052	55000
108	Shubham Gupta	Lecturer	1059	30000

- a. Display the employee details, departments that the departments are same in both the emp and dept.

Select emp.EMPNO, emp.ENAME, dept.DNAME From emp INNER JOIN dept ON emp.DEPTNO = dept.DEPTNO ;

b. Display all the employees and the departments implementing a left outer join.

```
Select emp.ENAME, emp.DEPTNO, dept.DNAME
From emp
LEFT OUTER JOIN dept
ON emp.DEPTNO = dept.DEPTNO;
```

c. Display the employee name and department name in which they are working implementing a right outer join.

```
Select emp.ENAME, emp.DEPTNO, dept.DNAME
From emp
RIGHT OUTER JOIN dept
ON emp.DEPTNO = dept.DEPTNO;
```

d. Display the employee name and department name in which they are working implementing a full outer join.

```
Select emp.ENAME, emp.DEPTNO, dept.DNAME
From emp
FULL OUTER JOIN dept
ON emp.DEPTNO = dept.DEPTNO;
```

Practical 10 (a):

Aim: Advanced SQL queries using PL-SQL

Question: Solve the following PL-SQL programs

a. Write a pl/sql program to swap two numbers.

Program:

declare

a int := 198;

b int := 564;

c int := a;

begin

dbms_output.put_line('a: '||a);

dbms_output.put_line('b: '||b);

a := b;

b := c;

dbms_output.put_line('a: '||a);

dbms_output.put_line('b: '||b);

end;

/

Output:

a: 198

b: 564

a: 564

b: 198

Practical 10(b) :

Question: b. Write a pl/sql program to find the largest of two numbers.

Program :

declare

 a int ;

 b int ;

begin

 a := 10 ;

 b := 38 ;

if a > b then

 dbms_output.put_line('a is the largest number :
 '||a);

else

 dbms_output.put_line('b is the largest
 number : '||b);

end if ;

end;

DATE:

Output:

b is the largest number: 38