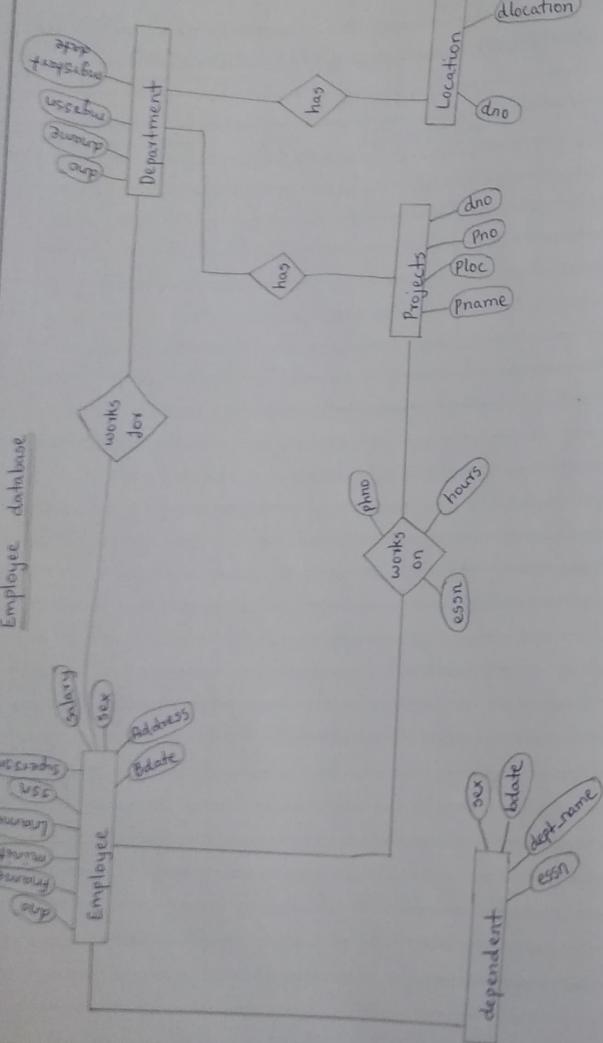


Employee database



DATE..... PAGE NO..... EXP. NO.....

PART - A

1. Draw E-R diagram and convert entities and relationships to relation table for a given scenario. Two assignments shall be carried out i.e. consider two different scenarios (e.g. bank, college)

Consider the Company database with following schema.

EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO)

DEPARTMENT (DNAME, DNUMBER, MGRSSN, MRSSTARTDATE)

DEPT LOCATIONS (DNUMBER, DLOCATION)

PROJECT (PNAME, PNUMBER, PLOCATION, DNUM)

WORKS-ON (ESSN, PNO < HOURS)

DEPENDENT (ESSN, DEPENDENT-NAME, SEX, BDATE, RELATIONSHIP)

```
CREATE TABLE department
(
```

```
    dname VARCHAR (20),
    dno VARCHAR (10),
    mgrssn VARCHAR (10),
    mrsstartdate DATE,
    CONSTRAINT dept_pk PRIMARY KEY (dno)
);
```

```
CREATE TABLE Employee
(
```

```
    fname VARCHAR (20),
    minit VARCHAR (1),
    lname VARCHAR (20),
    ssn VARCHAR (10),
    bdate DATE,
```

dname	dno	mgrssn	mgstart date
-------	-----	--------	--------------

Output : Table created.

fname	minit	lname	ssn	bdate	address	gender	salary	superssn	dno
-------	-------	-------	-----	-------	---------	--------	--------	----------	-----

Output : Table created.

dno	dloc
-----	------

Output : Table created.

pname	Pno	Ploc	dno
-------	-----	------	-----

Output : Table created

ssn	Pno	no-hrs
-----	-----	--------

Output : Table created.

ssn	dep-name	gender	bdate	relationship
-----	----------	--------	-------	--------------

Output : Table created.

DATE.....

PAGE NO.....

EXP. NO.....

```

address VARCHAR (50),
gender VARCHAR (1),
salary NUMBER (15,2),
superssn VARCHAR (20),
dno VARCHAR (10),
CONSTRAINT emp-pk PRIMARY KEY (ssn),
CONSTRAINT emp-fmk FOREIGN KEY (dno) REFERENCES department (dno)
);

```

```

CREATE TABLE dept-loc
(
dno VARCHAR (10),
dloc VARCHAR (20),
CONSTRAINT dloc-pk PRIMARY KEY (dno,dloc),
CONSTRAINT dloc-fmk FOREIGN KEY (dno) REFERENCES department (dno)
);

```

```

CREATE TABLE project
(
pname VARCHAR (20),
pno VARCHAR (10),
ploc VARCHAR (20),
dno VARCHAR (20),
CONSTRAINT proj-pk PRIMARY KEY (pno),
CONSTRAINT proj-fmk FOREIGN KEY (dno) REFERENCES department (dno)
);

```

DATE.....

PAGE NO.....

EXP. NO.

CREATE TABLE works_on

(

ssn VARCHAR(10),

pno VARCHAR(10),

no_hrs NUMBER(5),

CONSTRAINT w_pk PRIMARY KEY (ssn, pno),

CONSTRAINT w_frnk_ssn FOREIGN KEY (ssn)

REFERENCES employee (ssn),

CONSTRAINT w_frnk_pno FOREIGN KEY (pno)

REFERENCES project (pno)

);

CREATE TABLE dependent

(

ssn VARCHAR(10),

dept_name VARCHAR(30),

gender VARCHAR(1),

bdate DATE,

relationship VARCHAR(20),

CONSTRAINT dep_frnk FOREIGN KEY (ssn) REFERENCES

employee (ssn)

);

```
INSERT INTO department VALUES ('fbyname', 'fdno', 'fmgrssn',
    'fmgrstartdate');
```

Enter value for dname : Design

Enter value for dno : d001

Enter value for mgrssn : 512

Enter value for mgrstartdate : 14 - Jan - 1993

Output: 1 row created.

SQL > /

Enter value for dname : HR

Enter value for dno : d002

Enter value for mgrssn : 659

Enter value for mgrstartdate : 25 - May - 1995

Output: 1 row created.

```
SQL > INSERT INTO employee VALUES ('fname', 'fssn', 'fbdate', 'faddress',
    'fgender', 'f salary', 'fdno');
```

Enter value for name : Amu

Enter value for ssn : e101

Enter value for bdate : 18 - JULY - 2000

Enter value for address : CHENNAI

Enter value for gender : Female

Enter value for salary : 25000

Enter value for dno : d001

Output: 1 row created

SQL > /

Enter value for name : Abi

Enter value for ssn : e247

Enter value for bdate : 25 - AUG - 2002

Enter value for address : BLR

Enter value for gender : Male

DATE..... PAGE NO..... EXP. NO.....

2. Perform the following :

a. Viewing all databases, Creating Tables (with and without Constraints),

Inserting / Updating / Deleting Records in a Table, Saving (commit) and Undoing (rollback).

INSERT & SELECT QUERIES :

```
INSERT INTO department VALUES ('fbyname', 'fdno',
    'fmgrssn', 'fmgrstartdate');
```

SELECT * FROM department;

```
INSERT INTO employee VALUES ('fname', 'finit', 'flname',
    'fssn', 'fbdate', 'faddress', 'fgender', fsalary, 'fsuper
    ssn', 'fdno');
```

SELECT * FROM employee;

DESC dept_loc;

```
INSERT INTO dept_loc VALUES ('fdno', 'fdloc');
```

SELECT * FROM dept_loc;

```
INSERT INTO project VALUES ('fpname', 'fplc', 'fdno', fpr)
```

SELECT * FROM project;

```
INSERT INTO works_on VALUES ('fssn', 'fpno', &no-hrs);
```

SELECT * FROM works-on;

```
INSERT INTO dependent VALUES ('fssn', '&dep-name', 'f
    gender', 'fbdate', 'f relationship');
```

SELECT * FROM dependent;

UPDATE QUERY :

UPDATE employee

SET address = 'BLR'

WHERE ssn = 'e101';

Enter value for salary : 15000
Enter value for dno : d002

Output : 1 row created

SQL > INSERT INTO dept_loc VALUES ('dno', 'dloc');

Enter value for dno : d001
Enter value for dloc : Second floor

Output : 1 row created.

SQL > /

Enter value for dno : d002
Enter value for dloc : first floor

Output : 1 row created

SQL > INSERT INTO project VALUES ('pname', 'pno', 'ploc', 'dno');

Enter value for pname : Create template
Enter value for pno : p001
Enter value for ploc : 005
Enter value for dno : d001

Output : 1 row created

SQL > /

Enter value for pname : Coding
Enter value for pno : p002
Enter value for ploc : 006
Enter value for dno : d002

Output : 1 row created.

DATE.....

PAGE NO.....

EXP. NO.....

DELETE QUERY :

DELETE FROM employee WHERE ssn = 'e247';

SAVING AND UNDOING :

SAVEPOINT P1;

SAVEPOINT P2;

ROLLBACK TO P1;

COMMIT;

SQL > INSERT INTO works_on VALUES ('fssn', 'fpno', fno-hrs);

Enter value for ssn: e101

Enter value for pno: P001

Enter value for no-hrs: 12

Output: 1 row created

SQL > /

Enter value for ssn: e247

Enter value for pno: P002

Enter value for no-hrs: 14

Output: 1 row created

SQL > INSERT INTO dependent VALUES ('fssn', 'f dep-name', 'f gender',
'f bdate');

Enter value for ssn: e101

Enter value for dep-name: A

Enter value for gender: female

Enter value for bdate: 10-JUN-2000

Output: 1 row created

SQL > /

Enter value for ssn: e247

Enter value for dep-name: B

Enter value for gender: Male

Enter value for bdate - 19- Sept - 2001

Output: 1 row created

Output : 1 row updated.

Output : 1 row deleted.

Output : Savepoint created.

Output : Savepoint created.

Output : Rollback complete.

Output : Commit Complete.

DATE.....

PAGE No.....

EXP. No.....

Output : Table altered.

Output : Table created.

Output : Table truncated.

Output : Table dropped.

3. Perform the following:

a. Altering a Table , Dropping / Truncating / Renaming Tables , Backing up / Restoring a Database .

ALTERING A TABLE :

ALTER TABLE employee ADD phno NUMBER(10);

ALTER TABLE employee DROP COLUMN phno;

RENAMING A TABLE :

CREATE TABLE employee_copy AS SELECT * FROM employee;

TRUNCATE COMMAND :

TRUNCATE TABLE employee ;

DROP COMMAND :

DROP TABLE employee ;

Output: Table created.

```
SQL > INSERT INTO student VALUES ('fno', 'fname', 'fprg',
    fmarks);
```

Enter value for fno : 1

Enter value for name: Abi

Enter value for prg: BBA

Enter value for marks: 56

Output: 1 row created.

```
SQL > /
```

Enter value for fno: 2

Enter value for name: Anu

Enter value for prg: BBA

Enter value for marks: 76

Output: 1 row created.

```
SQL > /
```

Enter value for fno: 3

Enter value for name: Anu

Enter value for prg: BCA

Enter value for marks: 88

Output: 1 row created.

```
SQL > /
```

Enter value for fno: 4

Enter value for name: Pringa

Enter value for prg: BA

Enter value for marks: 78

Output: 1 row created.

DATE.....

PAGE No.....

EXP. NO.....

4. for a given set of relation schemas , create tables and perform the following Simple Queries , Simple Queries with Aggregate functions , Queries with Aggregate functions (group by and having clause).

CREATE TABLE Student

(

fno VARCHAR (10) PRIMARY KEY,
name VARCHAR (20),
prg VARCHAR (10),
marks VARCHAR (3)

);

INSERT INTO student VALUES ('fno', 'fname', 'fprg',
 fmarks);

SELECT *

FROM Student;

SELECT COUNT (*) AS NUMBER_STUDENT
FROM Student;

SELECT MAX(marks)

FROM Student;

SELECT MIN(marks)

FROM Student;

SELECT SUM(marks) TOTAL_MARKS
FROM Student;

SQL > /

Enter value for rno: 5

Enter value for name: kubra

Enter value for prg: BCA

Enter value for marks: 86

Output: 1 row created.

RNO	NAME	PRG	MAR
1	Abi	BBA	56
2	Anu	BBA	76
3	Amu	BCA	88
4	Priya	BA	78
5	kubra	BCA	86

Output: NUMBER - STUDENT
5

Output: MAX
88

Output: MIN
56

Output: TOTAL - MARKS
384

Output: AVERAGE - STUDENTS
76.8

DATE.....

PAGE NO.....

EXP. NO.....

SELECT AVG(marks) AVERAGE - STUDENTS
FROM student;

SELECT prg, COUNT(rno) AS NUMBER - OF - STUDENTS
FROM student
GROUP BY prg;

SELECT prg, COUNT(rno) AS NUMBER - OF - STUDENTS
FROM student
GROUP BY prg
HAVING COUNT(rno) > 1;

Output : PRG
BCA
BA
BBA

NUMBER - OF - STUDENTS
2
1
2

Output : PRG
BCA
BBA

NUMBER - OF - STUDENTS
2
2

DATE.....

PAGE No.....

Output: Table created.

Output: Table created

```
SQL > INSERT INTO department_e4 VALUES ('fdname', 'fdno',
  'fmgrssn', 'mgrstartdate');
```

Enter value for dname : HR

Enter value for dno : d001

Enter value for mgrssn : 234

Enter value for mgrstartdate : 15 - JUN - 1997

Output: 1 row created.

```
SQL > /
```

Enter value for dname : Research

Enter value for dno : d002

Enter value for mgrssn : 791

Enter value for mgrstartdate : 22 - JAN - 2001

Output: 1 row created.

```
SQL > /
```

Enter value for dname : Accounts

Enter value for dno : d003

Enter value for mgrssn : 823

Enter value for mgrstartdate : 09 - MAY - 2000

Output: 1 row created.

```
SQL > /
```

Enter value for dname : Accounts

Enter value for dno : d004

Enter value for mgrssn : 602

DATE.....

PAGE NO.....

EXP. No.....

5. Execute the following queries
a. How the resulting salaries if every employee working on the 'Research' Departments is given a 10% raise.

b. Find the sum of the salaries of all employees of the 'Accounts' department , as well as the maximumsalary , the minimum salary and the average salary in this department .

```
CREATE TABLE department_e4
(
```

 dname VARCHAR (20),

 dno VARCHAR(10) PRIMARY KEY,

 mgrssn VARCHAR (10),

 mgrstartdate DATE,

);

```
CREATE TABLE employee_e6
(
```

 ename VARCHAR (20),

 ssn VARCHAR(10) PRIMARY KEY,

 address VARCHAR (20),

 salary NUMBER (10),

 dno VARCHAR (10)

);

```
INSERT INTO department_e4 VALUES ('fdname', 'fdno',
  'fmgrssn', 'mgrstartdate');
```

```
SELECT *
```

```
FROM department_e4;
```

Enter value for mgrstartdate : 23-AUG-1999

Output : 1 row created.

Output	DNAME	DNO	MGRSSN	MGRSTARTDATE
	HR	d001	234	15-JUN-97
	Research	d002	791	22-JAN-01
	Accounts	d003	823	09-MAY-00
	Accounts	d004	602	23-AUG-99

SQL > INSERT INTO employee_e6 VALUES ('fename', 'fssn',
 'faddress', f salary , 'fdno');

Enter value for ename : Anu

Enter value for ssn : 112

Enter value for address : Blr

Enter value for salary : 42000

Enter value for dno : d001

Output : 1 row created.

SQL > /

Enter value for ename : Abi

Enter value for ssn : 113

Enter value for address : Chennai

Enter value for salary : 50000

Enter value for dno : d002

Output : 1 row created.

SQL > /

Enter value for ename : Binu

Enter value for ssn : 114

Enter value for address : kalyan

Enter value for dno : 55000

DATE.....

PAGE NO.....

EXP. NO.....

INSERT INTO employee_e6 VALUES ('fename', 'fssn',
 'faddress', f salary , 'fdno');

SELECT * FROM employee_e6;

UPDATE employee_e6 SET salary = salary * 1.1
 WHERE dno in (SELECT dno

FROM department_e4 WHERE
 dname = 'Research');

SELECT sum(salary) , max(salary) , min(salary)
 FROM employee_e6
 WHERE dno in (SELECT dno

FROM department_e4
 WHERE dname = 'Accounts');

Enter value for dno : d003

Output : 1 row created

SQL > /

Enter value for ename : Amu

Enter value for ssn : 115

Enter value for address : kothanur

Enter value for salary : 41000

Enter value for dno : d004

Output : 1 row created.

<u>Output</u> :	<u>ENAME</u>	<u>SSN</u>	<u>ADDRESS</u>	<u>SALARY</u>	<u>DNO</u>
	Anu	112	Bt4	42000	d001
	Abi	113	Chennai	50000	d002
	Binu	114	Kalyan	55000	d003
	Amu	115	Kothanur	41000	d004

Output : 1 row updated.

<u>Output</u> :	<u>SUM(SALARY)</u>	<u>MAX(SALARY)</u>	<u>MIN(SALARY)</u>
	96000	55000	41000

Output : Table created

Output : Table created

Output : Table created

Enter value for ename : Abi

Enter value for ssn : 456

Enter value for bdate : 12 - Jan - 1990

Enter value for address : Bls

Enter value for salary : 42000

Enter value for dno : d001

Output : 1 row created.

SQL > /

Enter value for ename : Tinu

Enter value for ssn : 781

Enter value for bdate : 22 - May - 1993

Enter value for address : Chennai

Enter value for salary : 45000

Enter value for dno : d002

Output : 1 row created.

SQL > /

Enter value for ename : Anu

Enter value for ssn : 981

Enter value for bdate : 26 - Aug - 1990

Enter value for address : Bls

Enter value for salary : 46000

Enter value for dno : d003

DATE.....

PAGE No.....

EXP. No.....

6. Execute the following queries
a. For each project , retrieve the project number , the project name and the number of employee who work there on that project . (use GROUP BY)
b. Retrieve the name of employees who born in the year 1990.

CREATE TABLE employee - p

(

ename VARCHAR (20),
ssn VARCHAR (10) PRIMARY KEY,
bdate DATE,
address VARCHAR (50),
salary NUMBER (15, 2),
dno VARCHAR (10)

);

CREATE TABLE project - o

(

pname VARCHAR (20),
pno VARCHAR (10) PRIMARY KEY,
plc VARCHAR (20),
dno VARCHAR (10)

);

CREATE TABLE works-on - o

(

ssn VARCHAR (10) PRIMARY KEY,
pno VARCHAR (10),
no-hrs NUMBER (5)

);

Output : 1 row created.

Enter value for pname : Coding
Enter value for pno : p001
Enter value for plac : goa
Enter value for dno : d001

Output : 1 row created.

SQL > /
Enter value for pname : Design
Enter value for pno : p002
Enter value for plac : andhra
Enter value for dno : d002

Output : 1 row created.

SQL > /
Enter value for pname : Editing
Enter value for pno : p003
Enter value for plac : chennai
Enter value for dno : d003

Output : 1 row created.

Enter value for ssn : 456
Enter value for pno : p001
Enter value for no-hrs : 12

Output : 1 row created.

SQL > /
Enter value for ssn : 781
Enter value for pno : p002
Enter value for no-hrs : 14

DATE.....

PAGE NO.....

EXP. NO.....

INSERT INTO employee-p VALUES ('fename', 'fssn',
'fbdate', 'faddress', & salary, 'fdno');

INSERT INTO project-o VALUES ('fpname', 'fpno',
'fplace', 'fdno');

INSERT INTO works-on-o VALUES ('fssn', 'fpno',
& no-hrs);

SELECT p.pno, p.pname, COUNT(*) AS "NO_OF_EMP"
FROM project-o P, Works-on-o W WHERE
P.pno = W.pno
GROUP BY p.pno, p.pname;

SELECT ename, bdate
FROM employee-p
WHERE bdate like '%.90';

Output : 1 row created

SQL > /

Enter value for ssn : 981

Enter value for pno : p003

Enter value for no_hrs : 15

Output : 1 row created

	PNAME	NO_OF_EMP
p001	Coding	1
p002	Design	1
p003	Editing	1

	ENAME	BDATE
Abi		12-JAN-90
Anu		26-AUG-90

DATE.....

PAGE No.....

EXP NO.....

Output :

ENAME	SSN	ADDRESS	SALARY	DNO
Anu	112	Bly	42000	d001
Abi	113	Chennai	55000	d002
Binu	114	Kalyan	55000	d003
Amu	115	Kothanur	41000	d004
Shalini	116	Bly	45000	d002
Kubra	117	Kalyan	46000	d002
Manasa	118	Gubbi Gross	42000	d002
Ben	119	Kothanur	60000	d002
Megha	120	Gubbi Gross	40000	d002
Kavya	121	Gubbi Gross	60000	d002

Output :

DNO
d002

COUNT(ENAME) ---
6

DATE.....

PAGE NO.....

EXP. NO.....

7. For each Department that has more than five employees, retrieve the department number and number of employees who are making salary more than 40000.

SELECT * FROM employee_e6,

SELECT dno, count(ename)

FROM employee_e6 WHERE (salary > 40000)

GROUP BY dno

HAVING COUNT(ename) > 5;

Output : Table created

Output : View created

Output : ENAME ----- SALARY ----- DNO ----- DNAME

Abi	42000	d001	bca
Tina	45000	d002	bcom

Output : View created.

Output : View dropped.

DATE.....

PAGE NO.....

EXP. NO.....

8. For a given set of relation tables perform the following:
Creating views (with and without check option), Dropping views , Selecting from a view.

CREATE TABLE department_o

(

 dname VARCHAR (15),
 dno VARCHAR (10) PRIMARY KEY,
 mgrssn VARCHAR (20),
 mgrstartdate VARCHAR (20)

);

CREATE VIEW EMP_DEPTI AS (SELECT e.ename , e.salary ,
e.dno , d.dname FROM employee - p e , department - o d
WHERE e.dno = d.dep_no);

SELECT * FROM EMP_DEPTI;

CREATE VIEW EMP_DEPTI AS (SELECT e.ename , e.salary ,
e.dno , d.dname FROM employee - p e , department - o d
WHERE e.dno = d.dep_no) with check option ;

DROP VIEW EMP_DEPTI;