13. Display the list of ECE 3rd year students whose birthday is 20th October.

```
SQL> SELECT *
  2  FROM STUDENT
  3  WHERE SEMESTER = 'SEM5' AND DEPTCODE='ECE' AND to_char (BIRTHDATE, 'DD')=20 AND to_char (BIRTHDATE, 'MM')=10;

    ROLLNO NAME           ADDRESS                          PHONENO  YEAROFADM DEPT SEME BIRTHDAT
---------- -------------- ------------------------------ ---------- ---------- ---- ---- --------
      2001 SANCHITA SINGH  Tegharia, Kolkata             9999110011       2017 ECE  SEM5 20-10-00
```

15. Display the students name, department, semester whose age is more than 20 year.

```
SQL> SELECT NAME, DEPTCODE, SEMESTER
  2  FROM STUDENT
  3  WHERE (MONTHS_BETWEEN(SYSDATE , BIRTHDATE)/12) > 20;
```

Column Name	Data Type	Size	Constraints
SubjectCode	Varchar2	6	Not null, Primary key
SubjectName	Varchar2	15	Not null
DeptCode	Varchar2	3	Foreign key Department(DeptCode)
Semester	Varchar2	4	Not Null

Solution:

CREATE TABLE SUBJECT (SubjectCode VARCHAR2(6) NOT NULL PRIMARY KEY, SubjectName VARCHAR2(25) NOT NULL, DeptCode VARCHAR2(3), Semester VARCHAR2(4) NOT NULL, foreign key (DeptCode) REFERENCES DEPARTMENT(DEPTCODE));

iii. Change the address of the student whose roll number is given.

```
SQL> Select ADDRESS from STUDENT where ROLLNO='1001';

ADDRESS
------------------------------
VIP Road, Kolkata
```

```
Update STUDENT set ADDRESS='Amsterdam' where ROLLNO='1001';
```

```
SQL> Update STUDENT set ADDRESS='Amsterdam' where ROLLNO='1001';

1 row updated.

Commit complete.
SQL> Select ADDRESS from STUDENT where ROLLNO='1001';

ADDRESS
------------------------------
Amsterdam
```

v. Increase the total number of student allocation of CSE by 15.

Update DEPARTMENT set STUDENTALLOTED=STUDENTALLOTED+15 where DEPTCODE='CSE';

x. Delete the student with given roll number from student table

DELETE FROM STUDENT WHERE ROLLNO = 3005;

vii. Add a default phone number to all CSE 5th semester students who do not have phone number.

Update STUDENT set PHONENO=0123456789 where PHONENO is NULL and DEPTCODE='CSE' and SEMESTER='SEM5';

xi. Display the students name, deptcode, semester who admitted before 2017 in alphabeti order of name.

select NAME,DEPTCODE,SEMESTER from STUDENT where YEAROFADM<2017 order by NAME;

xiv. Delete phone number of a given student.

update STUDENT set PHONENO=NULL where PHONENO=8902443764;

(i) Add constraint : DeptCode of Student is foreign key with reference to DeptCode in Department.

alter table STUDENT add foreign key(DEPTCODE) references DEPARTMENT(DEPTCODE);

select table_name,constraint_name,constraint_type,search_condition from all_constraints where table_name='STUDENT' and owner='B2_2024';

(ii) Add an attribute ***Block*** to the department table (data type - Char).

alter table DEPARTMENT add Block Char;

(iii) Insert data to this attribute.

update DEPARTMENT set BLOCK='C' where DEPTCODE='IT' or DEPTCODE='CSE';

update DEPARTMENT set BLOCK='B' where DEPTCODE='ECE' or DEPTCODE='EE';

(iv) Increase the width of the column subject code of SUBJECT table to 8.

alter table SUBJECT modify SUBJECTCODE VARCHAR(8);

(v) Delete primary key of table of Subject.

```
SQL> alter table SUBJECT drop primary key;
```

(vi) Add primary key to the table subject.

```
SQL> alter table SUBJECT add primary key(SUBJECTCODE);
```

(vii) Make Sub_code of Result table foreign key with respect to Subject code of SUBJECT table.

alter table RESULT add foreign key(SUB_CODE) references SUBJECT(SUBJECTCODE);

(viii) Add constraint that marks cannot be negative.

```
SQL> alter table RESULT add constraint CHK_RES check(MARKS>=0);
```

(ix) Alter the table SUBJECT and add check constraint such DeptCode is either CSE,IT, MCA, EE, ECE

alter table SUBJECT add constraint chk_dept check(DEPTCODE='CSE' or DEPTCODE='IT' or DEPTCODE='MCA' or DEPTCODE='EE' or DEPTCODE='ECE');

(x) Add column college phone number and add phone number 25739607 to each student.

alter table STUDENT add COLLEGE_PHONE_NO NUMBER(10) DEFAULT 987654321;

(xi) Create a table FACULTY (FID VARCHAR2(4), Name VARCHAR2(20)); Faculty id should start with 'F'.

```
create table FACULTY(FID VARCHAR2(4), NAME VARCHAR2(20), CONSTRAINT CHK_FAC CHECK(FID LIKE 'F%'));
```

(xiv) Delete the column pass marks from Result.

alter table RESULT drop COLUMN PASS_MARKS;

ii. Display the marks for all available subjects for the students who born in JUNE.

```
select MARKS from RESULT where ROLLNO IN (select ROLLNO from STUDENT
where to_char(BIRTHDATE, 'MM')=06);
```

iii. Show the student details of 5th semester Computer science and engineering Department.

```
select * from STUDENT where SEMESTER='SEM5' and DEPTCODE IN (select
DEPTCODE from DEPARTMENT where DEPTCODE='CSE');
```

v. Display the department name for which some student get more than 90.

```
select DEPTNAME from DEPARTMENT where DEPTCODE IN (select DEPTCODE from
STUDENT where ROLLNO IN (select ROLLNO from RESULT where MARKS>90));
```

xii. Find the student details of CSE students whose marks are not available in result.

```
select * from STUDENT where DEPTCODE='CSE' and ROLLNO NOT IN (select
ROLLNO from RESULT);
```

xiii. Find the subject name, subject code which is not included in result.

```
select SUBJECTNAME, SUBJECTCODE from SUBJECT where SUBJECTCODE NOT IN
(select SUB_CODE from RESULT);
```

i. Display student names along with the department name.

```
select STUDENT.Name, DEPARTMENT.DeptName from STUDENT INNER JOIN
DEPARTMENT on STUDENT.DEPTCODE = DEPARTMENT.DEPTCODE;
```

ii. Display student names, subject code and marks of students who took admission in the year 2017 and read subject CS502.

```
SELECT NAME, SUB_CODE, MARKS FROM STUDENT, RESULT WHERE STUDENT.YEAROFADM
= 2017 AND STUDENT.ROLLNO = RESULT.ROLLNO AND RESULT.SUB_CODE='CS502';
```

iii. Find the name of the students who got marks between 70 to 89 in CS501.

```
SELECT STUDENT.NAME FROM STUDENT INNER JOIN RESULT ON STUDENT.ROLLNO =
RESULT.ROLLNO INNER JOIN SUBJECT ON RESULT.SUB_CODE = SUBJECT.SUBJECTCODE
WHERE SUBJECT.SUBJECTCODE = 'CS501' AND RESULT.MARKS BETWEEN 70 AND 90;
```

iv. Find the student name, roll no who appeared in the examination for paper DBMS.

```
SELECT STUDENT.NAME, STUDENT.ROLLNO FROM STUDENT INNER JOIN RESULT ON
STUDENT.ROLLNO = RESULT.ROLLNO INNER JOIN SUBJECT ON RESULT.SUB_CODE =
SUBJECT.SUBJECTCODE WHERE SUBJECT.SUBJECTNAME='Data Base Systems';
```

vi. Show student names, department name for 5 th semester students who born in MAY.

```
SELECT NAME, DEPTNAME FROM DEPARTMENT, STUDENT WHERE STUDENT.SEMESTER =
'SEM5' AND EXTRACT(MONTH FROM STUDENT.BIRTHDATE) = 5 AND STUDENT.DEPTCODE
= DEPARTMENT.DEPTCODE;
```

x. Show student names, department name, subject name and subject wise marks for which the student get more than 90.

```
SELECT NAME, DEPTNAME, SUBJECTNAME, MARKS FROM STUDENT , DEPARTMENT,
SUBJECT, RESULT WHERE MARKS > 90 AND STUDENT.ROLLNO = RESULT.ROLLNO AND
STUDENT.DEPTCODE = DEPARTMENT.DEPTCODE AND RESULT.SUB_CODE =
SUBJECT.SUBJECTCODE;
```

xi. Add column HOD in DEPARTMENT. Insert faculty id in place of HoD.

```
ALTER TABLE DEPARTMENT ADD HOD VARCHAR2(4);
UPDATE DEPARTMENT SET HOD = 'F01' WHERE DEPTCODE='CSE';
UPDATE DEPARTMENT SET HOD = 'F02' WHERE DEPTCODE='ECE';
```

xiv. Find the names of all faculties whose HOD name is given.

```
SELECT NAME, DEPARTMENT.DEPTNAME FROM FACULTY, DEPARTMENT WHERE
FACULTY.FID = DEPARTMENT.HOD;
```

xv. Display subject name, subject code, marks including all subjects.[Outer join]

```
SELECT SUBJECT.SUBJECTNAME, SUBJECT.SUBJECTCODE, RESULT.MARKS FROM
SUBJECT LEFT OUTER JOIN RESULT ON SUBJECT.SUBJECTCODE = RESULT.SUB_CODE;
```

xvi. Display students name, subject code, marks. Include all students. [Outer join]

```
SELECT STUDENT.NAME, RESULT.SUB_CODE, RESULT.MARKS FROM STUDENT LEFT
OUTER JOIN RESULT ON STUDENT.ROLLNO = RESULT.ROLLNO;
```

2. Find the number of students whose marks of any subject is available.

```
SELECT COUNT (*) AS AVAILABLE FROM STUDENT, RESULT WHERE STUDENT.ROLLNO =
RESULT.ROLLNO;
```

3. Find total number of faculty.

```
SELECT COUNT(*) AS TOTAL_FACULTY FROM FACULTY;
```

4. Find the 5th semester student(s) who got maximum marks in a subject.

```
SELECT S.NAME, R.MARKS, SU.SUBJECTNAME FROM STUDENT S, RESULT R, SUBJECT
SU WHERE S.ROLLNO = R.ROLLNO AND S.SEMESTER = 'SEM5' AND R.SUB_CODE =
SU.SUBJECTCODE AND SU.SUBJECTNAME = 'Discrete Mathematics' AND R.MARKS =
(SELECT MAX(MARKS) FROM RESULT WHERE SUB_CODE = SU.SUBJECTCODE);
```

5. Find the roll number of a student who got maximum marks in CS501.

```
SELECT ROLLNO , MARKS FROM RESULT WHERE MARKS = (SELECT MAX(MARKS) FROM
RESULT WHERE SUB_CODE ='CS501');
```

6. Display average marks of CS502.

```
SELECT AVG(MARKS) FROM RESULT WHERE RESULT.SUB_CODE ='CS502';
```

7. Find the number of students in each department with their department code.

```
SELECT DEPTCODE, COUNT(*) AS NUMBER_OF_STUDENTS FROM STUDENT GROUP
BY(DEPTCODE);
```

8. Find the number of students in each department with their department name.

```
SELECT DEPARTMENT.DEPTCODE, COUNT(*) AS NUMBER_OF_STUDENTS,
DEPARTMENT.DEPTNAME FROM STUDENT, DEPARTMENT WHERE DEPARTMENT.DEPTCODE =
STUDENT.DEPTCODE GROUP BY DEPARTMENT.DEPTCODE, DEPARTMENT.DEPTNAME;
```

9. Find the Department with more than three faculty.

```
SELECT D.DEPTNAME FROM DEPARTMENT D WHERE (SELECT COUNT(TEACHER) FROM
SUBJECT S WHERE S.DEPTCODE = D.DEPTCODE) >= 3;
```

14. Find the second highest marks of the result table.

```
SELECT MAX(MARKS) AS SECOND_HIGHEST FROM RESULT WHERE MARKS NOT IN
(SELECT MAX(MARKS) FROM RESULT);
```

10. Find the student name and roll no who get more than 80 in at least two

```
SELECT NAME, ROLLNO FROM STUDENT WHERE (SELECT
COUNT(*) FROM RESULT R, STUDENT S WHERE
S.ROLLNO = R.ROLLNO AND R.MARKS > 80) >= 2;
```

11. Find the student name and roll no who get more than 70 in average.

```
SELECT S.NAME, S.ROLLNO FROM STUDENT S WHERE (SELECT AVG(MARKS) FROM
RESULT R WHERE R.ROLLNO = S.ROLLNO) > 70;
```

12. Display number of subject semester wise in dept CSE.

```
SELECT SEMESTER, COUNT(*) FROM SUBJECT WHERE DEPTCODE='CSE' GROUP
BY(SEMESTER);
```

```
6 rows selected.

SEME  COUNT(*)
```

13. Find the department name with maximum number of student.

```
SELECT DEPTNAME, STUDENTALLOTED FROM DEPARTMENT WHERE STUDENTALLOTED =
(SELECT MAX(STUDENTALLOTED) FROM DEPARTMENT);
```

15. Find the students name who got highest marks, subjectwise.

```
SELECT S.SUBJECTNAME, ST.NAME, MAX(R.MARKS) AS HIGHEST_MARKS FROM SUBJECT
S, RESULT R, STUDENT ST WHERE S.SUBJECTCODE = R.SUB_CODE AND R.ROLLNO =
ST.ROLLNO AND R.MARKS = (SELECT MAX(R2.MARKS) FROM RESULT R2 WHERE
R2.SUB_CODE = S.SUBJECTCODE) GROUP BY S.SUBJECTNAME, ST.NAME;
```

(Creation, Manipulate and Dropping of Views)

1. Create view V_MARKS with roll no, student name, subject code, marks of all CSE students.

```
CREATE VIEW V_MARKS AS SELECT STUDENT.ROLLNO , STUDENT.NAME,
SUBJECT.SUBJECTCODE FROM STUDENT, SUBJECT, RESULT WHERE STUDENT.ROLLNO =
RESULT.ROLLNO AND RESULT.SUB_CODE = SUBJECT.SUBJECTCODE AND
STUDENT.DEPTCODE = 'CSE';
```

```
SQL> select * from V_MARKS;
```

2. Create view V_CLASSTAKEN with subject code, subject name, Faculty code, faculty name where the subject is taught by that faculty.

```
CREATE VIEW V_CLASSTAKEN AS SELECT SU.SUBJECTCODE, SU.SUBJECTNAME,
SU.TEACHER, F.NAME FROM SUBJECT SU, FACULTY F WHERE SU.TEACHER = F.FID;
```

3. Create view V_SUBJECT with subject_code, subject_name, teacher name where number of student in the subject is more than 1.

```
CREATE VIEW V_SUBJECT AS SELECT S.SUBJECTCODE, S.SUBJECTNAME, F.NAME FROM
SUBJECT S, FACULTY F WHERE S.TEACHER = F.FID AND (SELECT COUNT(*) FROM STUDENT
ST WHERE ST.SEMESTER = S.SEMESTER AND ST.DEPTCODE = S.DEPTCODE) > 1;
```

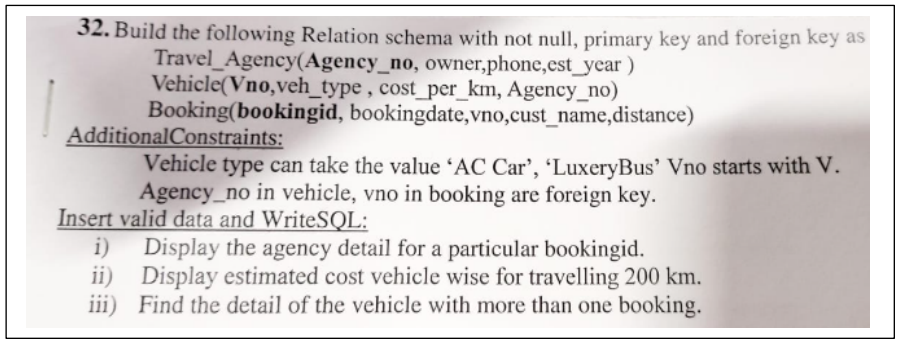
4. Create view V_STUDENT with student name, roll number, department who appeared in at least one examination.

```
CREATE VIEW V_STUDENT AS SELECT S.NAME , S.ROLLNO, S.DEPTCODE
FROM STUDENT S WHERE (SELECT COUNT (*) FROM RESULT R WHERE
R.ROLLNO=S.ROLLNO) >= 1;
```

5. Create view V_DEPT with dept code, semester, number of student with more than 75 marks in at least one subject.

```
CREATE VIEW V_DEPT AS SELECT S.DEPTCODE, S.SEMESTER, COUNT(*) AS
STUDENT_COUNT FROM STUDENT S WHERE (SELECT COUNT(*) FROM RESULT R WHERE
S.ROLLNO = R.ROLLNO AND R.MARKS > 75)>1 GROUP BY S.DEPTCODE, S.SEMESTER;
```

32. TRAVEL AGENT


32. Build the following Relation schema with not null, primary key and foreign key as
Travel_Agency(**Agency_no**, owner,phone,est_year)
Vehicle(**Vno**,veh_type , cost_per_km, Agency_no)
Booking(**bookingid**, bookingdate,vno,cust_name,distance)
AdditionalConstraints:
Vehicle type can take the value 'AC Car', 'LuxeryBus' Vno starts with V.
Agency_no in vehicle, vno in booking are foreign key.
Insert valid data and WriteSQL:
i) Display the agency detail for a particular bookingid.
ii) Display estimated cost vehicle wise for travelling 200 km.
iii) Find the detail of the vehicle with more than one booking.

```
CREATE TABLE TravelAgency (
    Agency_no INT PRIMARY KEY,
    owner VARCHAR(50) NOT NULL,
    phone VARCHAR(15) NOT NULL,
    est_year INT NOT NULL
);

CREATE TABLE Vehicle (
    Vno VARCHAR(10) PRIMARY KEY,
    veh_type VARCHAR(20) NOT NULL,
    cost_per_km DECIMAL(10,2) NOT NULL,
    Agency_no INT,
    FOREIGN KEY (Agency_no) REFERENCES TravelAgency(Agency_no)
);

CREATE TABLE Booking (
    bookingid INT PRIMARY KEY,
    bookingdate DATE NOT NULL,
    vno VARCHAR(10),
    cust_name VARCHAR(50) NOT NULL,
    distance DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (vno) REFERENCES Vehicle(Vno)
);
```

INSERT

```
INSERT INTO TravelAgency (Agency_no, owner, phone, est_year)
VALUES (1, 'John Doe', '123-456-7890', 2000);

INSERT INTO TravelAgency (Agency_no, owner, phone, est_year)
VALUES (2, 'Jane Smith', '987-654-3210', 1995);


INSERT INTO Vehicle (Vno, veh_type, cost_per_km, Agency_no)
VALUES ('V001', 'AC Car', 10.50, 1);

INSERT INTO Vehicle (Vno, veh_type, cost_per_km, Agency_no)
VALUES ('V002', 'Luxury Bus', 15.75, 1);

INSERT INTO Vehicle (Vno, veh_type, cost_per_km, Agency_no)
VALUES ('V003', 'AC Car', 11.00, 2);

INSERT INTO Vehicle (Vno, veh_type, cost_per_km, Agency_no)
VALUES ('V004', 'Luxury Bus', 16.25, 2);


INSERT INTO Booking (bookingid, bookingdate, vno, cust_name, distance)
VALUES (1001, TO_DATE('30-05-2024', 'DD-MM-YYYY'), 'V001', 'Alice', 150);

INSERT INTO Booking (bookingid, bookingdate, vno, cust_name, distance)
VALUES (1002, TO_DATE('30-05-2024', 'DD-MM-YYYY'), 'V003', 'Bob', 200);

INSERT INTO Booking (bookingid, bookingdate, vno, cust_name, distance)
VALUES (1003, TO_DATE('30-05-2024', 'DD-MM-YYYY'), 'V002', 'Charlie', 250);

INSERT INTO Booking (bookingid, bookingdate, vno, cust_name, distance)
VALUES (1004, TO_DATE('30-05-2024', 'DD-MM-YYYY'), 'V004', 'David', 180);

INSERT INTO Booking (bookingid, bookingdate, vno, cust_name, distance)
VALUES (1005, TO_DATE('30-05-2024', 'DD-MM-YYYY'), 'V001', 'Emma', 300);
```

SQL

i) Display the agency detail for a particular bookingid.

```
SELECT ta.*
FROM TravelAgency ta
JOIN Vehicle v ON ta.Agency_no = v.Agency_no
JOIN Booking b ON v.Vno = b.vno
WHERE b.bookingid = 1002;
```

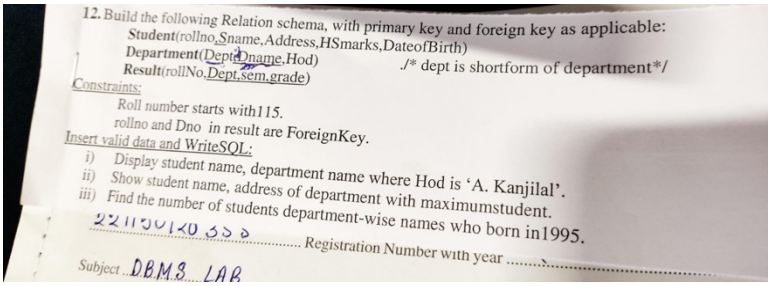
ii) Display estimated cost vehicle wise for traveling 200 km.

```
SELECT v.Vno, v.veh_type, (v.cost_per_km * 200) AS estimated_cost
FROM Vehicle v;
```

iii) Find the detail of the vehicle with more than one booking.

```
SELECT v.*
FROM Vehicle v
WHERE v.Vno IN (
    SELECT b.vno
    FROM Booking b
    GROUP BY b.vno
    HAVING COUNT(*) > 1
);
```

12. Build the Relation Schema


12. Build the following Relation schema, with primary key and foreign key as applicable:
Student(rollno,Sname,Address,HSmarks,DateofBirth)
Department(Deptid,Dname,Hod) ./* dept is shortform of department*/
Result(rollNo,Dept,sem,grade)
Constraints:
Roll number starts with115.
rollno and Dno in result are ForeignKey.
Insert valid data and WriteSQL:
i) Display student name, department name where Hod is 'A. Kanjilal'.
ii) Show student name, address of department with maximumstudent.
iii) Find the number of students department-wise names who born in1995.
Registration Number with year
Subject DBMS LAB

Constraints:
Roll number starts with 115.
rollno and Dno in result are Foreign Key.

i)Student(rollno,Sname,Address,HSmarks,DateofBirth)

```
CREATE TABLE Student (
  rollno VARCHAR2(10) PRIMARY KEY CHECK (rollno LIKE '115%'),
  Sname VARCHAR2(25) NOT NULL,
  Address VARCHAR2(30) NOT NULL,
  HSmarks INTEGER,
  DateofBirth DATE NOT NULL
);
```

ii) Department(Deptiname, Hod)

```
CREATE TABLE Department (
  Dept VARCHAR2(3) PRIMARY KEY,
  Dname VARCHAR2(50),
  Hod VARCHAR2(50)
);
```

iii) Result(rollNo,Dept, sem,grade)

```
CREATE TABLE Result (
  rollNo VARCHAR2(10),
  Dept VARCHAR2(3),
  sem NUMBER,
  grade VARCHAR2(1),
  FOREIGN KEY (rollNo) REFERENCES Student(rollno),
  FOREIGN KEY (Dept) REFERENCES Department(Dept)
);
```

INSERT

```
INSERT INTO Student VALUES ('11501', 'John Doe', 'Address1', 85, TO_DATE('1995-06-
01', 'YYYY-MM-DD'));
INSERT INTO Student VALUES ('11502', 'Jane Doe', 'Address2', 90, TO_DATE('1995-07-
01', 'YYYY-MM-DD'));
INSERT INTO Student VALUES ('11503', 'Bob Smith', 'Address3', 80, TO_DATE('1996-08-
01', 'YYYY-MM-DD'));
INSERT INTO Student VALUES ('11504', 'Alice Johnson', 'Address4', 95, TO_DATE('1995-
09-01', 'YYYY-MM-DD'));


INSERT INTO Department VALUES ('CSE', 'Computer Science and Engineering', 'A.
Kanjilal');
INSERT INTO Department VALUES ('ECE', 'Electronics and Communication Engineering',
'B. Chatterjee');
INSERT INTO Department VALUES ('ME', 'Mechanical Engineering', 'C. Das');


INSERT INTO Result VALUES ('11501', 'CSE', 1, 'A');
INSERT INTO Result VALUES ('11501', 'CSE', 2, 'B');
INSERT INTO Result VALUES ('11502', 'ECE', 1, 'A');
INSERT INTO Result VALUES ('11502', 'ECE', 2, 'A');
INSERT INTO Result VALUES ('11503', 'ME', 1, 'B');
INSERT INTO Result VALUES ('11504', 'CSE', 1, 'A');
```

SQL Queries:

i) Display student name, department name where Hod is 'A. Kanjilal':

```
SELECT S.Sname, D.Dname
FROM Student S
JOIN Result R ON S.rollno = R.rollNo
JOIN Department D ON R.Dept = D.Dept
WHERE D.Hod = 'A. Kanjilal';
```

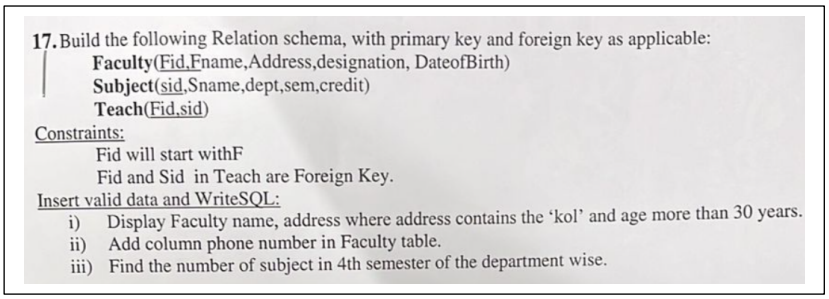
ii) Show student name, address of department with maximum students:

```
SELECT S.Sname, S.Address
FROM Student S
JOIN (
  SELECT Dept
  FROM Result
  GROUP BY Dept
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS max_dept ON S.rollno IN (
  SELECT rollNo
  FROM Result
  WHERE Dept = max_dept.Dept
);
```

iii) Find the number of students department-wise names who born in 1995:

```
SELECT D.Dname, COUNT(*) AS num_students
FROM Student S
JOIN Result R ON S.rollno = R.rollNo
JOIN Department D ON R.Dept = D.Dept
WHERE YEAR(S.DateofBirth) = 1995
GROUP BY D.Dname;
```

17. Build the Relation Schema


17. Build the following Relation schema, with primary key and foreign key as applicable:
Faculty(Fid,Fname,Address,designation, DateofBirth)
Subject(sid,Sname,dept,sem,credit)
Teach(Fid,sid)
Constraints:
Fid will start withF
Fid and Sid in Teach are Foreign Key.
Insert valid data and WriteSQL:
i) Display Faculty name, address where address contains the 'kol' and age more than 30 years.
ii) Add column phone number in Faculty table.
iii) Find the number of subject in 4th semester of the department wise.

```
CREATE TABLE Faculty (
    Fid VARCHAR(10) PRIMARY KEY,
    Fname VARCHAR(50),
    Address VARCHAR(100),
    Designation VARCHAR(50),
    DateofBirth DATE,
    CHECK (Fid LIKE 'F%')
);

CREATE TABLE Subject (
    Sid VARCHAR(10) PRIMARY KEY,
    Sname VARCHAR(50),
    Dept VARCHAR(50),
    Sem INT,
    Credit INT
);

CREATE TABLE Teach (
    Fid VARCHAR(10),
    Sid VARCHAR(10),
    FOREIGN KEY (Fid) REFERENCES Faculty(Fid),
    FOREIGN KEY (Sid) REFERENCES Subject(Sid)
);
```

ii) Insert Valid Data

```
INSERT INTO Faculty (Fid, Fname, Address, Designation, DateofBirth) VALUES
('F001', 'John Doe', '123 Kol Street', 'Professor', '1970-05-15'),
('F002', 'Jane Smith', '456 Kol Avenue', 'Associate Professor', '1985-08-20'),
('F003', 'Alice Johnson', '789 Kol Blvd', 'Lecturer', '1990-03-22');

INSERT INTO Subject (Sid, Sname, Dept, Sem, Credit) VALUES
('S101', 'Mathematics', 'Science', 4, 3),
('S102', 'Physics', 'Science', 4, 4),
('S103', 'Chemistry', 'Science', 4, 3),
('S201', 'History', 'Arts', 4, 3),
('S202', 'Geography', 'Arts', 4, 4);

INSERT INTO Teach (Fid, Sid) VALUES
('F001', 'S101'),
('F002', 'S102'),
('F003', 'S103'),
('F001', 'S201'),
('F002', 'S202');
```

iii) SQL Queries

i) Display Faculty name, address where address contains 'kol' and age more than 30 years

```
SELECT Fname, Address
FROM Faculty
WHERE Address LIKE '%Kol%'
AND (MONTHS_BETWEEN(SYSDATE, DATEOFBIRTH) / 12) > 30;
```

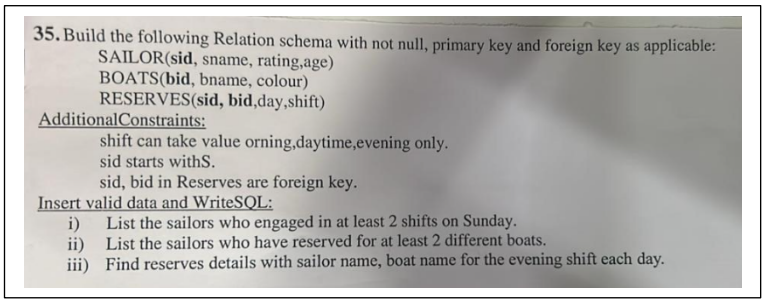
ii) Add column phone number in Faculty table

```
ALTER TABLE Faculty
ADD PhoneNumber VARCHAR(15);
```

iii) Find the number of subjects in 4th semester of the department wise

```
SELECT Dept, COUNT(*) as SubjectCount
FROM Subject
WHERE Sem = 4
GROUP BY Dept;
```

35. Creating the Tables with Constraints


35. Build the following Relation schema with not null, primary key and foreign key as applicable:
SAILOR(**sid**, sname, rating,age)
BOATS(**bid**, bname, colour)
RESERVES(**sid, bid**,day,shift)
AdditionalConstraints:
shift can take value orning,daytime,evening only.
sid starts withS.
sid, bid in Reserves are foreign key.
Insert valid data and WriteSQL:
i) List the sailors who engaged in at least 2 shifts on Sunday.
ii) List the sailors who have reserved for at least 2 different boats.
iii) Find reserves details with sailor name, boat name for the evening shift each day.

```
CREATE TABLE SAILOR (
    sid VARCHAR2(10) NOT NULL,
    sname VARCHAR2(50) NOT NULL,
    rating NUMBER,
    age NUMBER,
    CONSTRAINT pk_sailor PRIMARY KEY (sid),
    CONSTRAINT chk_sid CHECK (sid LIKE 'S%')
);

CREATE TABLE BOATS (
    bid VARCHAR2(10) NOT NULL,
    bname VARCHAR2(50) NOT NULL,
    colour VARCHAR2(20),
    CONSTRAINT pk_boats PRIMARY KEY (bid)
);

CREATE TABLE RESERVES (
    sid VARCHAR2(10) NOT NULL,
    bid VARCHAR2(10) NOT NULL,
    day VARCHAR2(10) NOT NULL,
    shift VARCHAR2(10) NOT NULL,
    CONSTRAINT pk_reserves PRIMARY KEY (sid, bid, day, shift),
    CONSTRAINT fk_reserves_sailor FOREIGN KEY (sid) REFERENCES SAILOR(sid),
    CONSTRAINT fk_reserves_boats FOREIGN KEY (bid) REFERENCES BOATS(bid),
    CONSTRAINT chk_shift CHECK (shift IN ('morning', 'daytime', 'evening'))
);
```

ii. Inserting Valid Data

```
INSERT INTO SAILOR (sid, sname, rating, age) VALUES ('S1', 'John', 5, 25);
INSERT INTO SAILOR (sid, sname, rating, age) VALUES ('S2', 'Paul', 3, 30);
INSERT INTO SAILOR (sid, sname, rating, age) VALUES ('S3', 'Mike', 7, 35);

INSERT INTO BOATS (bid, bname, colour) VALUES ('B1', 'Boat A', 'Red');
INSERT INTO BOATS (bid, bname, colour) VALUES ('B2', 'Boat B', 'Blue');
INSERT INTO BOATS (bid, bname, colour) VALUES ('B3', 'Boat C', 'Green');

INSERT INTO RESERVES (sid, bid, day, shift) VALUES ('S1', 'B1', 'Sunday', 'morning');
INSERT INTO RESERVES (sid, bid, day, shift) VALUES ('S1', 'B1', 'Sunday', 'evening');
INSERT INTO RESERVES (sid, bid, day, shift) VALUES ('S2', 'B2', 'Sunday', 'daytime');
INSERT INTO RESERVES (sid, bid, day, shift) VALUES ('S3', 'B3', 'Monday', 'evening');
INSERT INTO RESERVES (sid, bid, day, shift) VALUES ('S2', 'B1', 'Monday', 'morning');
INSERT INTO RESERVES (sid, bid, day, shift) VALUES ('S1', 'B2', 'Monday', 'evening');
```

iii. Writing SQL Queries

i) List the sailors who engaged in at least 2 shifts on Sunday:

```
SELECT s.sid, s.sname
FROM SAILOR s
JOIN RESERVES r ON s.sid = r.sid
WHERE r.day = 'Sunday'
GROUP BY s.sid, s.sname
HAVING COUNT(r.shift) >= 2;
```

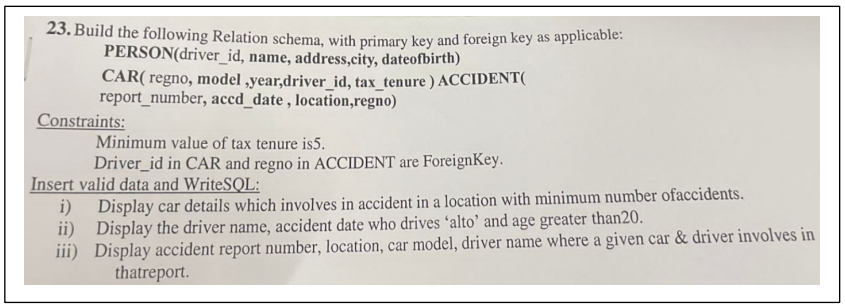
ii) List the sailors who have reserved for at least 2 different boat

```
SELECT s.sid, s.sname
FROM SAILOR s
JOIN RESERVES r ON s.sid = r.sid
GROUP BY s.sid, s.sname
HAVING COUNT(DISTINCT r.bid) >= 2;
```

iii) Find reserves details with sailor name, boat name for the evening shift each day:

```
SELECT s.sname, b.bname, r.day
FROM RESERVES r
JOIN SAILOR s ON r.sid = s.sid
JOIN BOATS b ON r.bid = b.bid
WHERE r.shift = 'evening';
```

23. Creating the Tables with Constraints



23. Build the following Relation schema, with primary key and foreign key as applicable:

PERSON(driver_id, **name, address,city, dateofbirth)**

CAR(regno, **model ,year,driver_id, tax_tenure) ACCIDENT(**
report_number, **accd_date , location,regno)**

Constraints:

Minimum value of tax tenure is5.

Driver_id in CAR and regno in ACCIDENT are ForeignKey.

Insert valid data and WriteSQL:

i) *Display car details which involves in accident in a location with minimum number ofaccidents.*

ii) *Display the driver name, accident date who drives 'alto' and age greater than20.*

iii) *Display accident report number, location, car model, driver name where a given car & driver involves in thatreport.*

```
CREATE TABLE PERSON (
    driver_id VARCHAR2(10) NOT NULL,
    name VARCHAR2(50) NOT NULL,
    address VARCHAR2(100),
    city VARCHAR2(50),
    dateofbirth DATE,
    CONSTRAINT pk_person PRIMARY KEY (driver_id)
);

CREATE TABLE CAR (
    regno VARCHAR2(10) NOT NULL,
    model VARCHAR2(50) NOT NULL,
    year NUMBER(4) NOT NULL,
    driver_id VARCHAR2(10) NOT NULL,
    tax_tenure NUMBER NOT NULL,
    CONSTRAINT pk_car PRIMARY KEY (regno),
    CONSTRAINT fk_car_driver FOREIGN KEY (driver_id) REFERENCES PERSON(driver_id),
    CONSTRAINT chk_tax_tenure CHECK (tax_tenure >= 5)
);

CREATE TABLE ACCIDENT (
    report_number VARCHAR2(10) NOT NULL,
    accd_date DATE NOT NULL,
    location VARCHAR2(100) NOT NULL,
    regno VARCHAR2(10) NOT NULL,
    CONSTRAINT pk_accident PRIMARY KEY (report_number),
    CONSTRAINT fk_accident_car FOREIGN KEY (regno) REFERENCES CAR(regno)
);
```

ii. Inserting Valid Data

```
INSERT INTO PERSON (driver_id, name, address, city, dateofbirth) VALUES ('D1', 'John
Doe', '123 Elm St', 'Springfield', TO_DATE('1990-05-15', 'YYYY-MM-DD'));
INSERT INTO PERSON (driver_id, name, address, city, dateofbirth) VALUES ('D2', 'Jane
Smith', '456 Oak St', 'Springfield', TO_DATE('1985-08-22', 'YYYY-MM-DD'));
INSERT INTO PERSON (driver_id, name, address, city, dateofbirth) VALUES ('D3', 'Mike
Johnson', '789 Pine St', 'Shelbyville', TO_DATE('2000-01-10', 'YYYY-MM-DD'));
```

```
INSERT INTO CAR (regno, model, year, driver_id, tax_tenure) VALUES ('R1', 'Alto', 2018,
'D1', 5);
INSERT INTO CAR (regno, model, year, driver_id, tax_tenure) VALUES ('R2', 'Civic',
2019, 'D2', 6);
INSERT INTO CAR (regno, model, year, driver_id, tax_tenure) VALUES ('R3', 'Corolla',
2020, 'D3', 7);
```

```
INSERT INTO ACCIDENT (report_number, accd_date, location, regno) VALUES ('A1',
TO_DATE('2023-05-10', 'YYYY-MM-DD'), 'Main St', 'R1');
INSERT INTO ACCIDENT (report_number, accd_date, location, regno) VALUES ('A2',
TO_DATE('2023-05-11', 'YYYY-MM-DD'), 'Elm St', 'R2');
INSERT INTO ACCIDENT (report_number, accd_date, location, regno) VALUES ('A3',
TO_DATE('2023-05-12', 'YYYY-MM-DD'), 'Main St', 'R3');
INSERT INTO ACCIDENT (report_number, accd_date, location, regno) VALUES ('A4',
TO_DATE('2023-05-13', 'YYYY-MM-DD'), 'Main St', 'R1');
```

iii. Writing SQL Queries

i) Display car details which involves in accident in a location with minimum number of accidents:

```
SELECT c.*
FROM CAR c
JOIN ACCIDENT a ON c.regno = a.regno
WHERE a.location = (
    SELECT location
    FROM ACCIDENT
    GROUP BY location
    ORDER BY COUNT(*) ASC
    FETCH FIRST 1 ROWS ONLY
);
```

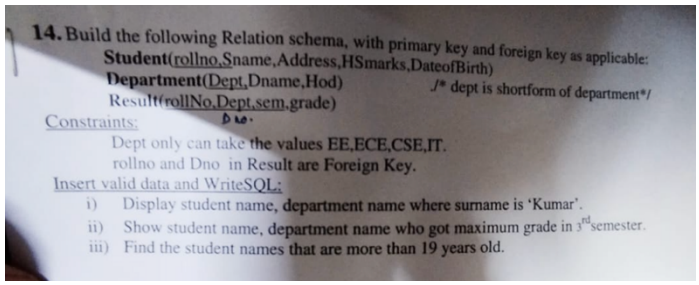
ii) Display the driver name, accident date who drives 'alto' and age greater than 20

```
SELECT p.name, a.accd_date
FROM PERSON p
JOIN CAR c ON p.driver_id = c.driver_id
JOIN ACCIDENT a ON c.regno = a.regno
WHERE c.model = 'Alto'
  AND (EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM p.dateofbirth)) > 20;
```

iii) Display accident report number, location, car model, driver name where a given car & driver involves in that report:

```
SELECT a.report_number, a.location, c.model, p.name
FROM ACCIDENT a
JOIN CAR c ON a.regno = c.regno
JOIN PERSON p ON c.driver_id = p.driver_id
WHERE c.regno = 'given_regno'
  AND p.driver_id = 'given_driver_id';
```

14. Creating the Tables with Constraints


14. Build the following Relation schema, with primary key and foreign key as applicable:
Student(rollno,Sname,Address,HSmarks,DateofBirth)
Department(Dept,Dname,Hod) /* dept is shortform of department*/
Result(rollNo,Dept,sem,grade)
Constraints:
Dept only can take the values EE,ECE,CSE,IT.
rollno and Dno in Result are Foreign Key.
Insert valid data and WriteSQL:
i) Display student name, department name where surname is 'Kumar'.
ii) Show student name, department name who got maximum grade in 3rd semester.
iii) Find the student names that are more than 19 years old.

```
create table student1053(
    rollno VARCHAR2(10) NOT NULL PRIMARY KEY,
    Sname VARCHAR2(25) NOT NULL,
    Address VARCHAR2(30) NOT NULL,
    HSmarks Integer,
    DateOfBirth Date NOT NULL
);

create table department (
    Dept VARCHAR2(3) PRIMARY KEY,
    Dname VARCHAR2(30),
    HOD VARCHAR2(30),
    CONSTRAINT Dept_check CHECK (Dept IN ('EE', 'CSE','ECE', 'IT'))
);

CREATE TABLE Result (
  rollNo VARCHAR2(10),
  Dept VARCHAR2(3),
  sem NUMBER,
  grade VARCHAR2(10),
  FOREIGN KEY (rollNo) REFERENCES student1053(rollNo),
  FOREIGN KEY (Dept) REFERENCES department(Dept)
);
```

ii. Inserting Valid Data

```
Insert into student1053 values('1030', 'Surya Kumar', 'DumDum, Kolkata', 40,
TO_DATE('1922-02-24', 'YYYY-MM-DD'));

Insert into student1053 values('1040', 'Ankita Dutta', 'Newtown, Kolkata', 90,
TO_DATE('2002-04-23', 'YYYY-MM-DD'));

Insert into student1053 values('1020', 'Gaylord Dey', 'Noida, Delhi', 60,
TO_DATE('2012-01-05', 'YYYY-MM-DD'));

Insert into student1053 values('1010', 'Gay Kumar', 'sugardaddy, Chennai', 50,
TO_DATE('2010-04-13', 'YYYY-MM-DD'));
```

iii. Writing SQL Queries

i) Display student name, department name where surname is 'Kumar'.

```
select S.Sname, D.Dname from
student1053 S JOIN result R ON
S.RollNo = R.RollNo
JOIN Department D ON R.Dept = D.Dept
WHERE S.Sname LIKE '%Kumar';
```

ii) Show student name, department name who got maximum grade in 3"semester.

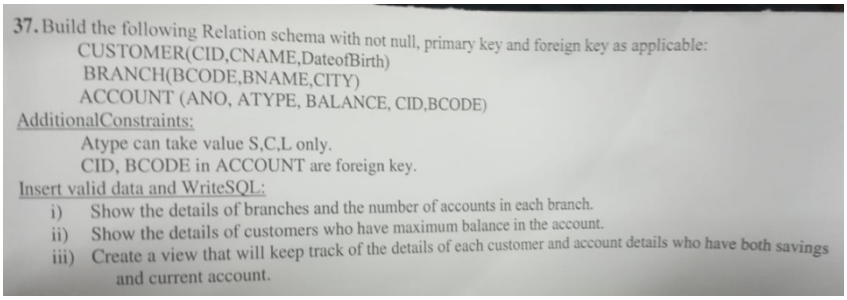
```
select S.Sname, D.Dname
from student1053 S

JOIN Result R ON S.rollno = R.rollno
JOIN Department D ON R.Dept = D.Dept
WHERE R.sem = 3 AND R.grade = (Select MAX(grade) FROM Result where sem = 3);
```

iii) Find the student names that are more than 19 years old.

```
select Sname from student1053
WHERE DateofBirth < ADD_MONTHS(SYSDATE, -19*12);
```

37. Creating the Tables


37. Build the following Relation schema with not null, primary key and foreign key as applicable:
CUSTOMER(CID,CNAME,DateofBirth)
BRANCH(BCODE,BNAME,CITY)
ACCOUNT (ANO, ATYPE, BALANCE, CID,BCODE)
AdditionalConstraints:
Atype can take value S,C,L only.
CID, BCODE in ACCOUNT are foreign key.
Insert valid data and WriteSQL:
i) Show the details of branches and the number of accounts in each branch.
ii) Show the details of customers who have maximum balance in the account.
iii) Create a view that will keep track of the details of each customer and account details who have both savings and current account.

```
CREATE TABLE CUSTOMER (
    CID VARCHAR2(10) NOT NULL,
    CNAME VARCHAR2(50) NOT NULL,
    DOB DATE NOT NULL,
    CONSTRAINT pk_customer PRIMARY
KEY (CID)
);

CREATE TABLE BRANCH (
    BCODE VARCHAR2(10) NOT NULL,
    BNAME VARCHAR2(50) NOT NULL,
    CITY VARCHAR2(50) NOT NULL,
    CONSTRAINT pk_branch PRIMARY KEY (BCODE)
);

CREATE TABLE ACCOUNT (
    ANO VARCHAR2(10) NOT NULL,
    ATYPE VARCHAR2(1) NOT NULL,
    BALANCE NUMBER NOT NULL,
    CID VARCHAR2(10) NOT NULL,
    BCODE VARCHAR2(10) NOT NULL,
    CONSTRAINT pk_account PRIMARY KEY (ANO),
    CONSTRAINT fk_account_customer FOREIGN KEY (CID) REFERENCES CUSTOMER(CID),
    CONSTRAINT fk_account_branch FOREIGN KEY (BCODE) REFERENCES BRANCH(BCODE),
    CONSTRAINT chk_atype CHECK (ATYPE IN ('S', 'C', 'L'))
);
```

ii. Inserting Valid Data

```
INSERT INTO CUSTOMER (CID, CNAME, DOB) VALUES ('C1', 'Alice', TO_DATE('1990-01-01',
'YYYY-MM-DD'));
INSERT INTO CUSTOMER (CID, CNAME, DOB) VALUES ('C2', 'Bob', TO_DATE('1985-02-02',
'YYYY-MM-DD'));
INSERT INTO CUSTOMER (CID, CNAME, DOB) VALUES ('C3', 'Charlie', TO_DATE('2000-03-03',
'YYYY-MM-DD'));

INSERT INTO BRANCH (BCODE, BNAME, CITY) VALUES ('B1', 'Main Branch', 'New York');
INSERT INTO BRANCH (BCODE, BNAME, CITY) VALUES ('B2', 'Uptown Branch', 'New York');
INSERT INTO BRANCH (BCODE, BNAME, CITY) VALUES ('B3', 'Downtown Branch', 'Los
Angeles');

INSERT INTO ACCOUNT (ANO, ATYPE, BALANCE, CID, BCODE) VALUES ('A1', 'S', 5000, 'C1',
'B1');
INSERT INTO ACCOUNT (ANO, ATYPE, BALANCE, CID, BCODE) VALUES ('A2', 'C', 3000, 'C1',
'B1');
INSERT INTO ACCOUNT (ANO, ATYPE, BALANCE, CID, BCODE) VALUES ('A3', 'L', 1500, 'C2',
'B2');
INSERT INTO ACCOUNT (ANO, ATYPE, BALANCE, CID, BCODE) VALUES ('A4', 'S', 2000, 'C3',
'B3');
INSERT INTO ACCOUNT (ANO, ATYPE, BALANCE, CID, BCODE) VALUES ('A5', 'C', 2500, 'C3',
'B3');
```

iii. Writing SQL Queries

i) Show the details of branches and the number of accounts in each branch:

```
SELECT b.BCODE, b.BNAME, b.CITY, COUNT(a.ANO) AS num_accounts
FROM BRANCH b
LEFT JOIN ACCOUNT a ON b.BCODE = a.BCODE
GROUP BY b.BCODE, b.BNAME, b.CITY;
```

ii) **Show the details of customers who have the maximum balance in the account:**

```
SELECT c.CID, c.CNAME, c.DOB, a.BALANCE
FROM CUSTOMER c
JOIN ACCOUNT a ON c.CID = a.CID
WHERE a.BALANCE = (
    SELECT MAX(BALANCE)
    FROM ACCOUNT
);
```

iii) **Create a view that will keep track of the details of each customer and account details who have both savings and current accounts:**

```
CREATE VIEW customer_savings_current AS
SELECT c.CID, c.CNAME, c.DOB, a.ANO, a.ATYPE, a.BALANCE, a.BCODE
FROM CUSTOMER c
JOIN ACCOUNT a ON c.CID = a.CID
WHERE a.ATYPE IN ('S', 'C')
  AND c.CID IN (
      SELECT CID
      FROM ACCOUNT
      WHERE ATYPE = 'S'
  )
  AND c.CID IN (
      SELECT CID
      FROM ACCOUNT
      WHERE ATYPE = 'C'
  );
```

Question No 30: Shuvechha Nandi-11500121037 **30**

ANS.

Create Table:

1)orders:

```
create table orders(orderno integer primary key,odate date,custno varchar2(8),ord_amt
integer,constraint chk_ord check (ord_amt > 0 and custno like 'C%'));
```

2)warehouse:

```
create table warehouse(warehouseno integer primary key,city varchar2(20) NOT NULL);
```

3)shipment:

```
create table shipment(orderno integer,warehouseno integer,ship_date date,primary
key(orderno,warehouseno));
```

Value insertion in table:

1)Orders:

```
INSERT INTO orders VALUES (1001,TO_DATE('01-01-2023','DD-MM-YYYY'), 'C001', 500);
INSERT INTO orders VALUES (1002,TO_DATE('15-02-2023','DD-MM-YYYY'), 'C002', 650);
INSERT INTO orders VALUES (1003,TO_DATE('17-03-2023','DD-MM-YYYY'), 'C003', 750);
INSERT INTO orders VALUES (1004,TO_DATE('19-04-2023','DD-MM-YYYY'), 'C004', 1000);
INSERT INTO orders VALUES (1005,TO_DATE('30-05-2023','DD-MM-YYYY'), 'C005', 450);
```

2)Warehouse:

```
INSERT INTO warehouse VALUES(1,'New York');
INSERT INTO warehouse VALUES(2,'Canada');
INSERT INTO warehouse VALUES(3,'Chikago');
INSERT INTO warehouse VALUES(4,'Pune');
INSERT INTO warehouse VALUES(5,'Kolkata');
```

3)Shipment:

```
INSERT INTO shipment VALUES(1001, 1, TO_DATE('05-01-2023','DD-MM-YYYY'));
INSERT INTO shipment VALUES(1002, 2, TO_DATE('06-02-2023','DD-MM-YYYY'));
INSERT INTO shipment VALUES(1001, 3, TO_DATE('12-03-2023','DD-MM-YYYY'));
INSERT INTO shipment VALUES(1001, 4, TO_DATE('19-04-2023','DD-MM-YYYY'));
INSERT INTO shipment VALUES(1005, 5, TO_DATE('20-05-2023','DD-MM-YYYY'));
```

Display:

```
SQL> SELECT * FROM orders;

   ORDERNO ODATE            CUSTNO      ORD_AMT
---------- ---------------- -------- ----------
      1001 01-JAN-23        C001            500
      1002 15-FEB-23        C002            650
      1003 17-MAR-23        C003            750
      1004 19-APR-23        C004           1000
      1005 30-MAY-23        C005            450
```

```
SQL> SELECT * FROM warehouse;

WAREHOUSENO CITY
----------- --------------------
          1 New York
          2 Canada
          3 Chikago
          4 Pune
          5 Kolkata
```

```
SQL> select * from shipment;

   ORDERNO WAREHOUSENO SHIP_DATE
---------- ----------- -----------------
      1001           1 05-JAN-23
      1002           2 06-FEB-23
      1001           3 12-MAR-23
      1001           4 19-APR-23
      1005           5 20-MAY-23
```

First quey:

SELECT * FROM orders WHERE ord_amt= (SELECT MAX(ord_amt) FROM orders);

```
SQL> SELECT * FROM orders WHERE ord_amt= (SELECT MAX(ord_amt) FROM orders);

   ORDERNO ODATE            CUSTNO      ORD_AMT
---------- ---------------- -------- ----------
      1004 19-APR-23        C004           1000
```

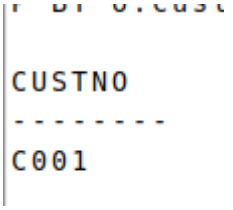
Second Query:

select o.custno, o.orderno, o.odate , o.ord_amt from orders o, shipment s, warehouse w where w.warehouseno = s.warehouseno and o.orderno = s.orderno and w.city='Pune';

```
CUSTNO       ORDERNO ODATE                   ORD_AMT
-------- ---------- ------------------ ----------
C001           1001 01-JAN-23                 500
```

Third Query:

SELECT o.custno FROM orders o , shipment s where o.orderno = s.orderno GROUP BY o.custno HAVING COUNT(DISTINCT s.warehouseno) = 3;



```
CUSTNO
--------
C001
```

3

Question No. 3

Build following Relational Schema:

1. Employee(Eid,Ename.Address,City,Doj,Salary)

```
CREATE TABLE Employee1016 (
        Eid VARCHAR2(5) PRIMARY KEY,
        Ename VARCHAR2(30) NOT NULL,
   Address VARCHAR2(25) NOT NULL,
   City VARCHAR2(10) NOT NULL,
   DOB date,
   salary integer NOT NULL,
   CHECK(salary >7500)
);
```

2. Project(Pid,Location,Pname.Mng,Client,Branch)

```
CREATE TABLE PROJECT1016(
   Pid VARCHAR2(5) PRIMARY KEY,
   Location VARCHAR2(30) NOT NULL,
   Pname VARCHAR2(25) NOT NULL,
   Mng VARCHAR2(25) NOT NULL,
   Client VARCHAR2(25) NOT NULL,
   Branch VARCHAR2(25) NOT NULL
);
```

3. Work(Eid,Pid)

```
CREATE TABLE WORKS1016(
   Eid VARCHAR2(5),
   Pid VARCHAR2(5),
   Foreign Key (Eid) REFERENCES Employee1016(Eid),
   Foreign Key (Pid) REFERENCES PROJECT1016(Pid)
);
```

Insert Data:

```
INSERT INTO Employee1016 VALUES ('E1','Subrata Dey','Saltlake','Kolkata',TO_DATE('20-01-2001','DD-MM-YYYY'),8000);
INSERT INTO Employee1016 VALUES ('E2','Pankaj Roy','Ultadanga','Kolkata',TO_DATE('20-05-2002','DD-MM-YYYY'),9000);
INSERT INTO Employee1016 VALUES ('E3','Sayak Roy','Dankuni','Howrah',TO_DATE('25-05-2001','DD-MM-YYYY'),10000);
INSERT INTO Employee1016 VALUES ('E4','Parthiv Dey','Haldiram','Kolkata',TO_DATE('27-01-2004','DD-MM-YYYY'),8700);
INSERT INTO Employee1016 VALUES ('E5','Sankalpa Ghosh','Belgachia','Kolkata',TO_DATE('20-05-2006','DD-MM-YYYY'),8730);
```

```
SQL> select * from Employee1016;

EID   ENAME                          ADDRESS                   CITY       DOJ         SALARY
----- ------------------------------ ------------------------- ---------- -------- ----------
E2    Pankaj Roy                     Ultadanga                 Kolkata    20-05-02       9000
E3    Sayak Roy                      Dankuni                   Howrah     25-05-01      10000
E4    Parthiv Dey                    Haldiram                  Kolkata    27-01-04       8700
E1    Subrata Dey                    Saltlake                  Kolkata    20-01-01       8000
E5    Sankalpa Ghosh                 Belgachia                 Kolkata    20-05-06       8730
```

INSERT INTO PROJECT1016 VALUES ('P1','Kolkata','Bridge','Rachit Dey','KMDA','Kolkata');
INSERT INTO PROJECT1016 VALUES ('P2','Howrah','Plant','Adrija Saha','Bank','Howrah');
INSERT INTO PROJECT1016 VALUES ('P3','Habra','Cooling Plant','Sounak Dey','IDFC','Habra');
INSERT INTO PROJECT1016 VALUES ('P4','Kolkata','Building','Souvik Dey','KMDA','Kolkata');
INSERT INTO PROJECT1016 VALUES ('P5','Haldia','Bridge','Sayak Dey','KMDA','Kolkata');

```
SQL> select * from PROJECT1016;

PID   LOCATION                       PNAME                  MNG                    CLIENT                 BRANCH
----- ------------------------------ ---------------------- ---------------------- ---------------------- --------
P1    Kolkata                        Bridge                 Rachit Dey             KMDA                   Kolkata
P2    Howrah                         Plant                  Adrija Saha            Bank                   Howrah
P3    Habra                          Cooling Plant          Sounak Dey             IDFC                   Habra
P4    Kolkata                        Building               Souvik Dey             KMDA                   Kolkata
P5    Haldia                         Bridge                 Sayak Dey              KMDA                   Kolkata
```

INSERT INTO WORKS1016 VALUES('E1','P1');
INSERT INTO WORKS1016 VALUES('E2','P2');
INSERT INTO WORKS1016 VALUES('E3','P3');
INSERT INTO WORKS1016 VALUES('E4','P4');
INSERT INTO WORKS1016 VALUES('E5','P5');

```
SQL> select * from WORKS1016;

EID   PID
----- -----
E1    P1
E2    P2
E3    P3
E4    P4
E5    P5
```

QUESTION 1:
Display the eid,employee name with surname ‘Roy’.

Code:
select e.Eid,e.Ename from Employee1016 e where e.Ename like '%Roy';
Output:

```
SQL> select  e.Eid,e.Ename from Employee1016 e where e.Ename like '%Roy';

EID   ENAME
----- ------------------------------
E2    Pankaj Roy
E3    Sayak Roy
```

Question 2:
Find the eid,employee name whose project location is ‘Kolkata’.

Code:
select e.Eid, e.Ename from Employee1016 e,PROJECT1016 p,WORKS1016 w where e.Eid = w.Eid and p.Pid = w.Pid and p.Location='Kolkata';

Output:

```
SQL> select e.Eid, e.Ename from Employee1

EID   ENAME
----- ------------------------------
E1    Subrata Dey
E4    Parthiv Dey
```

Question 3:
Show the number of employee works in each project with projectname who born in January.

Code:
select P.Pname,count(*)
from PROJECT1016 P, WORKS1016 W, Employee1016 E
where W.Pid = P.Pid and E.Eid = W.Pid and extract(month from DOB) = 1
group by P.Pname;

Output:

```
PNAME                     COUNT(*)
------------------------- ----------
Bridge                             1
Building                           1
```

Constraints:
1. Salary of employees must be more than Rs. 7500.
2. Eid and Pid in Works1016 are Foreignkey.

8

CREATE TABLE EMPLOYEE1030(Eid varchar2(10) primary key, Ename varchar2(50),Address varchar2(50), City varchar2(50), Doj varchar2(50),salary number(20));

```
SQL> DESC EMPLOYEE1030
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EID                                       NOT NULL VARCHAR2(10)
 ENAME                                              VARCHAR2(50)
 ADDRESS                                            VARCHAR2(50)
 CITY                                               VARCHAR2(50)
 DOJ                                                VARCHAR2(50)
 SALARY                                             NUMBER(20)
```

CREATE TABLE PROJECT1030(Pid varchar2(10) primary key, Location varchar2(50),Pname varchar2(20), Mng varchar2(50), Client varchar2(50),Branch varchar2(50));

```
SQL> DESC PROJECT1030
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                       NOT NULL VARCHAR2(10)
 LOCATION                                           VARCHAR2(50)
 PNAME                                              VARCHAR2(20)
 MNG                                                VARCHAR2(50)
 CLIENT                                             VARCHAR2(50)
 BRANCH                                             VARCHAR2(50)
```

CREATE TABLE WORKS1030(Eid varchar2(10) references EMPLOYEE1030(Eid), Pid varchar2(10) references PROJECT1030(Pid), primary key(Eid, Pid));

```
SQL> DESC WORKS1030;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EID                                       NOT NULL VARCHAR2(10)
 PID                                       NOT NULL VARCHAR2(10)
```

DISPLAY EMPLOYEE1030

```
SQL> SELECT * FROM EMPLOYEE1030;

EID        ENAME                                    ADDRESS                                  CITY                                     DOJ                                      SALARY
---------- ---------------------------------------- ---------------------------------------- ---------------------------------------- ---------------------------------------- ----------
E001       Ajit Dey                                 Baguiati                                 Kolkata                                  02-06-2020                                20000
E002       Srijita Bhatt                            Garia                                    Kolkata                                  03-07-2010                                30000
E003       Soumik Saha                              Behala                                   Kolkata                                  12-08-2010                                40000
E004       Soukya Dhar                              Chinar Park                              Kolkata                                  15-10-2007                                50000
```

DISPLAY PROJECT1030

```
SQL> SELECT * FROM PROJECT1030;

PID       LOCATION                                           PNAME              MNG                                                CLIENT                                             BRANCH
--------- -------------------------------------------------- ------------------ -------------------------------------------------- -------------------------------------------------- -----
---------
P001      Kolkata                                            CHAT APP           Srijita Bhatt                                      Rohit                                              Kolkata
P002      Bangalore                                          BLOOD CHECK WEBSITE Ajit Dey                                          Srijita                                            Kolkata
P003      Bangalore                                          TRAFFIC MANAGEMENT Soumik Saha                                        Ajit                                               Delhi
```

DISPLAY WORKS1030

```
QL> SELECT * FROM WORKS10

ID         PID
---------- ----------
001        P001
002        P001
003        P002
004        P002
```

queries

1. SELECT CITY,COUNT(*) FROM EMPLOYEE1030 GROUP BY CITY HAVING COUNT(EID)>3;

```
SQL> SELECT CITY,COUNT(*) FROM EMPLOYEE1030 GROUP BY CITY HAVING COUNT(EID)>=3;

CITY                                                COUNT(*)
-------------------------------------------------- ----------
Kolkata                                                     4
```

2. SELECT PNAME FROM PROJECT1030 WHERE PID NOT IN(SELECT PID FROM WORKS1030);

```
PNAME
--------------------
TRAFFIC MANAGEMENT

SQL>
```

3. SELECT EID, ENAME FROM EMPLOYEE1030 WHERE CITY = 'Kolkata'AND EID IN(SELECT EID FROM WORKS1030);

```
EID        ENAME
---------- --------------------------------------------------
E001       Ajit Dey
E002       Srijita Bhatt
E003       Soumik Saha
E004       Soukya Dhar

SQL>
```

25

CREATE TABLE CUSTOMER(custno INT NOT NULL, cname VARCHAR2(100) NOT NULL, city VARCHAR2(100) NOT NULL, PRIMARY KEY(custno));

```
SQL> DESC CUSTOMER
 Name
      Null?     Type
 ----------------------------------------------------------------------------------------------------------------------------------
----- -------- ----------------------------------------------------------------------------------------------------------------------
 CUSTNO
      NOT NULL NUMBER(38)
 CNAME
      NOT NULL VARCHAR2(100)
 CITY
      NOT NULL VARCHAR2(100)
```

CREATE TABLE ORDER0020(orderno INT NOT NULL, odate DATE NOT NULL, custno INT NOT NULL, PRIMARY KEY(orderno), FOREIGN KEY(custno) REFERENCES CUSTOMER(custno));

```
Name
     Null?     Type
 ----------------------------------------
----- -------- ------------------------
 ORDERNO
      NOT NULL NUMBER(38)
 ODATE
      NOT NULL DATE
 CUSTNO
      NOT NULL NUMBER(38)
```

CREATE TABLE ITEM(itemno VARCHAR2(10) NOT NULL, unitprice INT NOT NULL CHECK(unitprice>0), PRIMARY KEY(itemno), check(itemno like 'It%'));

```
SQL> DESC ITEM
 Name
      Null?     Type
 ----------------------------------------------------
----- -------- ---------------------------------------
 ITEMNO
      NOT NULL VARCHAR2(10)
 UNITPRICE
      NOT NULL NUMBER(38)
```

CREATE TABLE ORDER_ITEM(orderno INT NOT NULL, itemno VARCHAR2(10) NOT NULL, quantity INT NOT NULL, PRIMARY KEY(orderno,itemno), FOREIGN KEY(orderno) REFERENCES ORDER1(orderno), FOREIGN KEY(itemno) REFERENCES ITEM(itemno));

```
SQL> DESC ORDER_ITEM
 Name
      Null?     Type
 -----------------------------------
----- -------- --------------------
 ORDERNO
      NOT NULL NUMBER(38)
 ITEMNO
      NOT NULL VARCHAR2(10)
 QUANTITY
      NOT NULL NUMBER(38)
```

INSERT INTO CUSTOMER VALUES(1101,'Ashis Kumar', 'Kolkata');
INSERT INTO CUSTOMER VALUES(1102,'Kunal Kumar', 'Bangalore');
INSERT INTO CUSTOMER VALUES(1103,'Shubham Shaw','Chennai');

```
SQL> SELECT * FROM CUSTOMER;

    CUSTNO CNAME                                                    CITY
---------- -------------------------------------------------------- ----------------------------
----------------------------------------
      1101 Ashis Kumar                                              Kolkata
      1102 Kunal Kumar                                              Bangalore
      1103 Shubham Shaw                                             Chennai
```

INSERT INTO ORDER0020 VALUES(5001,to_date('01-02-2024','DD-MM-YYYY'),1101);
INSERT INTO ORDER0020 VALUES(5002,to_date('01-03-2024','DD-MM-YYYY'),1102);
INSERT INTO ORDER0020 VALUES(5003,to_date('02-03-2024','DD-MM-YYYY'),1103);

```
   ORDERNO ODATE        CUSTNO
---------- -------- ----------
      5001 01-02-24       1101
      5002 01-03-24       1102
      5003 02-03-24       1103
```

INSERT INTO ITEM VALUES('It01',150);
INSERT INTO ITEM VALUES('It02',50);

```
ITEMNO      UNITPRICE
---------- ----------
It01              150
It02               50
```

INSERT INTO ORDER_ITEM VALUES(5001,'It01',10);
INSERT INTO ORDER_ITEM VALUES(5002,'It02',12);
INSERT INTO ORDER_ITEM VALUES(5003,'It01',2);

```
   ORDERNO ITEMNO          QUANTITY
---------- ---------- ----------
      5001 It01                10
      5002 It02                12
      5003 It01                 2
```

SELECT DISTINCT c2.custno, c2.cname, c2.city FROM CUSTOMER c1 JOIN "ORDER0020" o1 ON c1.custno = o1.custno JOIN ORDER_ITEM oi1 ON o1.orderno =oi1.orderno JOIN ORDER_ITEM oi2 ON oi1.itemno = oi2.itemno JOIN "ORDER0020" o2 ON oi2.orderno = o2.orderno JOIN CUSTOMER c2 ON o2.custno =c2.custno WHERE c1.cname = 'Ashis Kumar' AND c1.custno <> c2.custno;

```
    CUSTNO CNAME                                                                                                CITY
---------- ---------------------------------------------------------------------------------------------------- ----------------------------------------------------------------------------------------------------
      1103 Shubham Shaw                                                                                         Chennai
```

SELECT o.orderno, SUM(oi.quantity * i.unitprice) AS total_amount,c.cname FROM "ORDER1" o JOIN ORDER_ITEM oi ON o.orderno = oi.orderno JOIN Item i ON oi.itemno = i.itemno JOIN CUSTOMER c ON o.custno = c.custno GROUP BY o.orderno, c.cname;

```
   ORDERNO TOTAL_AMOUNT CNAME
---------- ------------ ----------------------
      5001         1500 Ashis Kumar
      5002          600 Kunal Kumar
      5003          300 Shubham Shaw
```

ALTER TABLE CUSTOMER ADD phone VARCHAR2(10);

```
SQL> ALTER TABLE CUSTOMER ADD phone VARCHAR2(10);

Table altered.
```

```
SQL> DESC CUSTOMER
 Name
      Null?     Type
 --------------------------------
----- -------- ----------------
 CUSTNO
      NOT NULL NUMBER(38)
 CNAME
      NOT NULL VARCHAR2(100)
 CITY
      NOT NULL VARCHAR2(100)
 PHONE
ARCHAR2(10)
```