5.1. Define the following terms as they apply to the relational model of data:

Domain: A domain D is a set of atomic values. By atomic we mean that each value in the domain is indivisible as far as the formal relational model is concerned.

Attribute: a column header is called an attribute. Each attribute A_i is the name of a role played by some domain D in the relation schema R.

n-tuple: Each n-tuple t is an ordered list of n values $t=<v_1,v_2,...,v_n>$,where each value v_i $(1 \le i \le n)$ is an element of dom (A_i) or is a special NULL value.

relation schema: A relation schema R, denoted by $R(A_1, A_2, ... , A_n)$, is made up of a relation name R and a list of attributes, $A_1, A_2, ... , A_n$.

relation state: A relation (or relation state) r of the relation schema $R(A_1, A_2, ... , A_n)$, also denoted by r(R), is a set of n-tuples $r = \{t_1, t_2, ... , t_m\}$.

degree of a relation: The degree (or arity) of a relation is the number of attributes n of its relation schema.

relational database schema: A relational database schema S is a set of relation schemas $S = \{R_1, R_2, ... , R_m\}$ and a set of integrity constraints IC.

relational database state: A relational database state DB of S is a set of relation states $DB = \{r_1, r_2, ... , r_m\}$ such that each r_i is a state of R_i and such that the r_i relation states satisfy the integrity constraints specified in IC.

5.2. Why are tuples in a relation not ordered?

A relation is defined as a set of tuples. Mathematically, elements of a set have no order among them; hence, tuples in a relation do not have any particular order

5.3. Why are duplicate tuples not allowed in a relation?

In the formal relational model, a *relation* is defined as a *set of tuples.* By definition, all elements of a set are distinct; hence, all tuples in a relation must also be distinct. This means that no two tuples can have the same combination of values for *all* their attributes.

5.4. What is the difference between a key and a super key?

A **key** k of a relation schema R is a super-key of R with the additional property that removing any attribute A from K leaves a set of attributes K′ that is not a super-key of R anymore. A key is a super-key but not vice versa. A super-key may be a key (if it is minimal) or may not be a key (if it is not minimal).

5.5. Why do we designate one of the candidate keys of a relation to be the primary key?

In general, a relation schema may have more than one key. In this case, each of the keys is called a candidate key . It is common to designate one of the candidate keys as the primary key of the relation. This is the candidate key whose values are used to identify tuples in the relation.

5.6. Discuss the characteristics of relations that make them different from ordinary tables and files.

- A table is simply a set of rows where a relation is a set that specifically does not allow duplicates.

- Ordering of tuples in a Relation -The tuples are not considered to be ordered, even though they appear to be in the tabular form.

- Ordering of attributes in a relation schema R and of values within each tuples - We consider the attributes in R(A1, A2, .., An) and the values in t=<v1, v2, .., vn> to be ordered.

- Values in a tuple - All values are considered to be indivisible(atomic). A special null value is used to represent values that are unknown or inapplicable to certain tuples.

5.7. Discuss the various reasons that lead to the occurrence of NULL values in relations.

The desired reasons that lead to the occurrence of null values in relations are as given below:

- When value of an attribute is not applicable for a tuple. it can be marked as NULL.

- When value for an attribute unknown for a tuple, it can be marked as NULL.

5.8. Discuss the entity integrity and referential integrity constraints. Why is each considered important?

The entity integrity constraint states that no primary key value can be NULL. This is because the primary key value is used to identify individual tuples in a relation. Having NULL values for the primary key implies that we cannot identify some tuples.

The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples in the two relations. Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an *existing tuple* in that relation

5.9. Define foreign key. What is this concept used for?
A primary key of one table that appears as an attribute in another table and acts to provide a logical relationship between the two tables

The conditions for a foreign key, given below, specify a referential integrity constraint between the two relation schemas R_1 and R_2. A set of attributes FK in relation schema R_1 is a foreign key of R_1 that references relation R_2 if it satisfies the following rules:

- The attributes in FK have the same domain(s) as the primary key attribute's PK of R_2; the attributes FK are said to reference or refer to the relation R_2.
- A value of FK in a tuple t_1 of the current state $r_1(R_1)$ either occurs as a value of PK for some tuple t_2 in the current state $r_2(R_2)$ *or is NULL*. In the former case, we have $t_1[\text{FK}] = t_2[\text{PK}]$, and we say that the tuple t_1 references or refers to the tuple t_2.

What are the six clauses in the syntax of the SQL retrieval query. Which are required and which are optional?

- SELECT<attribute list>
- FROM<table list>
- WHERE<condition>
- GROUP BY<grouping attribute (S)>

- HAVING<group condition >
- ORDER BY<attribute list>

  The SELECT and FROM clauses are the required clauses and the clauses like WHERE, GROUP BY, HAVING and ORDER BY are optional clauses.

Write SQL statements to do the following on the STUDENT/COURSE schema shown below. (actual SQL implementation is not required).

- Select Lname, Fname
  From EMPLOYEE,
  WORKS_ON, PROJECT
  Where Dno=5 and Ssn = Essn and Pno = Pnumber and Pname = 'ProductX' and Hours > 10.

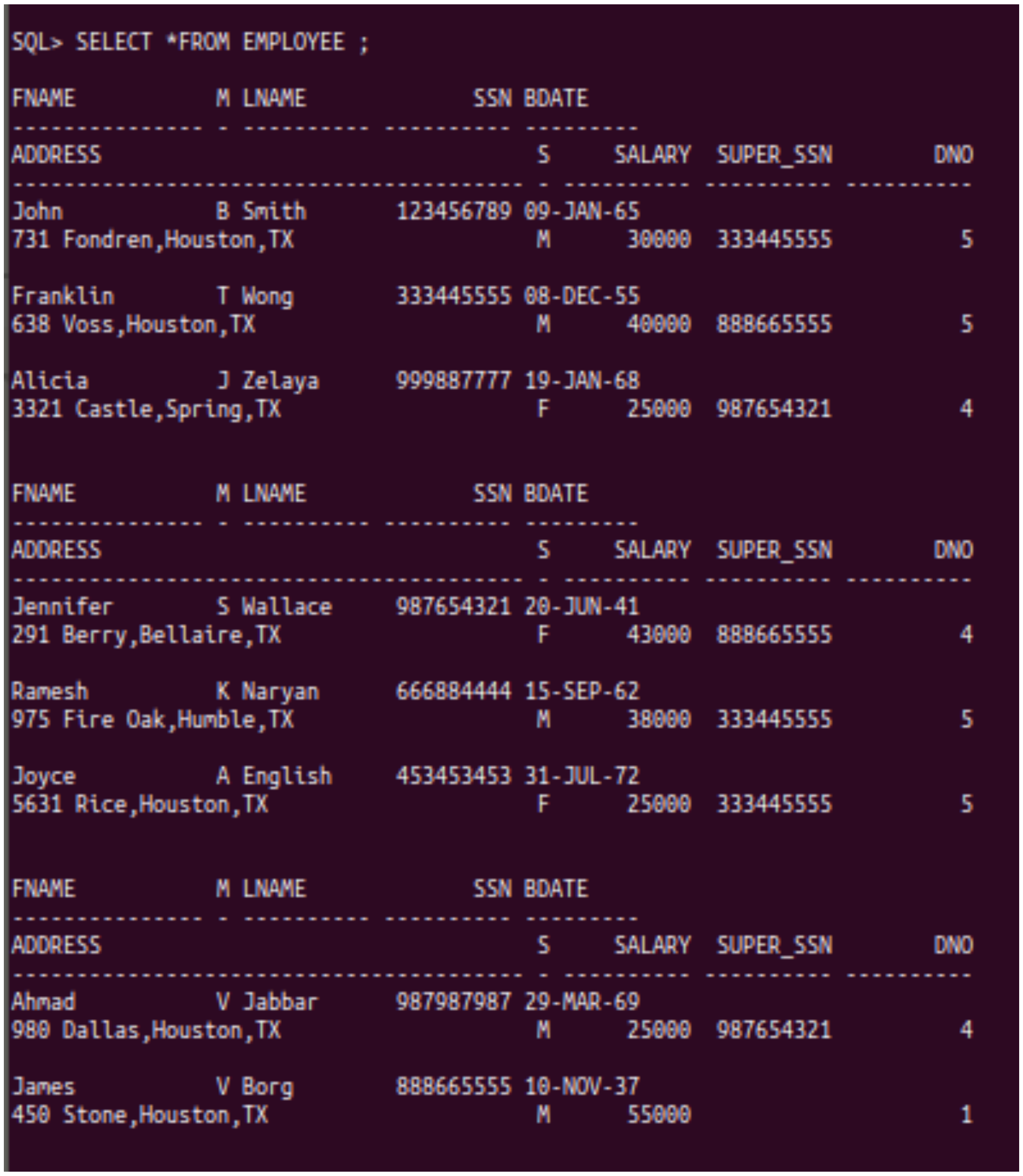
- Select Lname, Fname
  From Employee,
  DEPENDENT
  Where Ssn=Essn and Fname = Dependent_name.

- Select E.Lname, E.Fname
  from EMPLOYEE E, EMPLOYEE S
  Where S.Fname = 'Franklin' and
  S.Lname = 'Wong' and
  E.Super_ssn = S.Ssn.

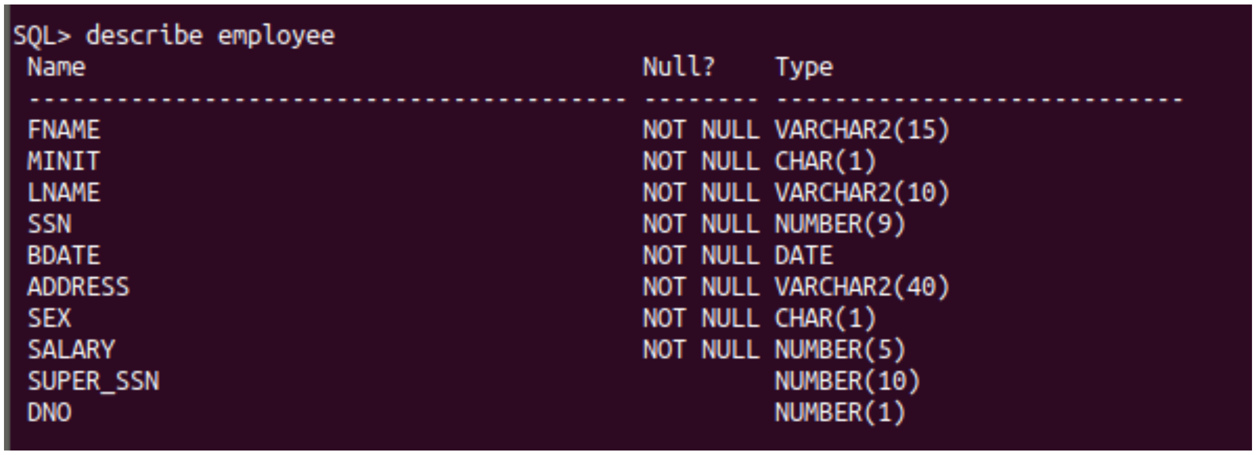
Write SQL statements to do the following on the STUDENT/COURSE schema shown below. (actual SQL implementation is not required).

INSERT into STUDENT values (‘Johnson’, 25, 1, ’Math’)
UPDATE Student set Class = 2 where name = ‘Smith’
INSERT into COURSE values (‘Knowledge Engineering’, ‘CS4390’, 3,’CS’)
DELETE from STUDENT where Name = ‘Smith’, and Student_number =17

On your Oracle DBMS, create and populate the EMPLOYEE table as shown below. Make sure to define the Bdate attribute with the ‘date’ type. To demonstrate your answer is correct, include screen shots from the output of the SQL “describe employee;” and “select * from employee;” commands.

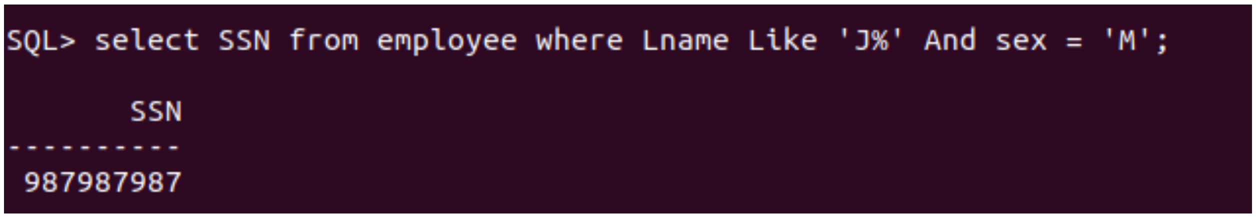


```
SQL> SELECT *FROM EMPLOYEE ;

FNAME           M LNAME            SSN BDATE
--------------- - --------- --------- ---------
ADDRESS                              S     SALARY  SUPER_SSN        DNO
------------------------------------ - ---------- ---------- ----------
John            B Smith      123456789 09-JAN-65
731 Fondren,Houston,TX               M      30000  333445555          5

Franklin        T Wong       333445555 08-DEC-55
638 Voss,Houston,TX                  M      40000  888665555          5

Alicia          J Zelaya     999887777 19-JAN-68
3321 Castle,Spring,TX                F      25000  987654321          4


FNAME           M LNAME            SSN BDATE
--------------- - --------- --------- ---------
ADDRESS                              S     SALARY  SUPER_SSN        DNO
------------------------------------ - ---------- ---------- ----------
Jennifer        S Wallace    987654321 20-JUN-41
291 Berry,Bellaire,TX                F      43000  888665555          4

Ramesh          K Naryan     666884444 15-SEP-62
975 Fire Oak,Humble,TX               M      38000  333445555          5

Joyce           A English    453453453 31-JUL-72
5631 Rice,Houston,TX                 F      25000  333445555          5


FNAME           M LNAME            SSN BDATE
--------------- - --------- --------- ---------
ADDRESS                              S     SALARY  SUPER_SSN        DNO
------------------------------------ - ---------- ---------- ----------
Ahmad           V Jabbar     987987987 29-MAR-69
980 Dallas,Houston,TX                M      25000  987654321          4

James           V Borg       888665555 10-NOV-37
450 Stone,Houston,TX                 M      55000                     1
```



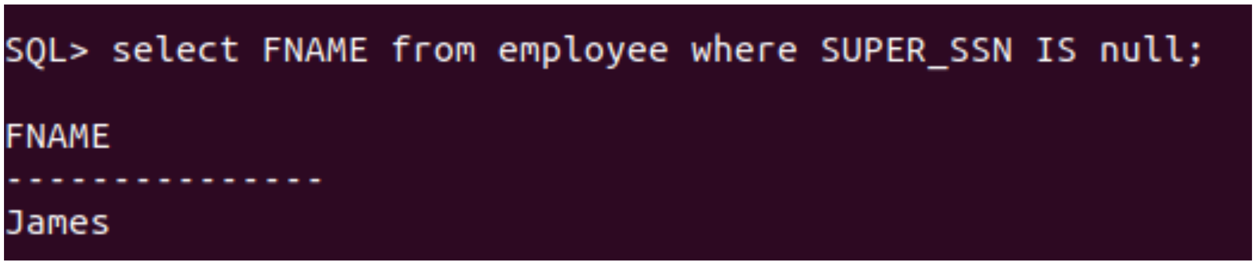
```
SQL> describe employee
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FNAME                                     NOT NULL VARCHAR2(15)
 MINIT                                     NOT NULL CHAR(1)
 LNAME                                     NOT NULL VARCHAR2(10)
 SSN                                       NOT NULL NUMBER(9)
 BDATE                                     NOT NULL DATE
 ADDRESS                                   NOT NULL VARCHAR2(40)
 SEX                                       NOT NULL CHAR(1)
 SALARY                                    NOT NULL NUMBER(5)
 SUPER_SSN                                          NUMBER(10)
 DNO                                                NUMBER(1)
```

Retrieve and display only the SSN values of male employees whose last name begins with the letter ‘J’.



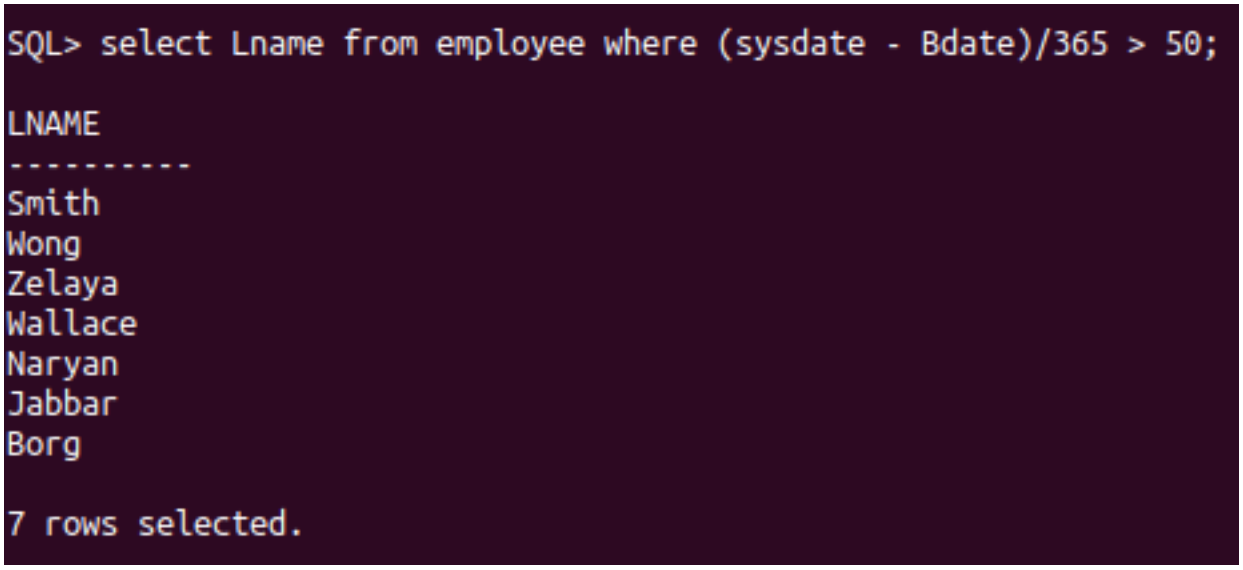
```
SQL> select SSN from employee where Lname Like 'J%' And sex = 'M';

       SSN
----------
 987987987
```

Retrieve and display only the employee’s first name only if he/she is a supervisor.



```
SQL> select FNAME from employee where SUPER_SSN IS null;

FNAME
---------------
James
```

Retrieve and display only the employee’s last name if he/she is older than 50 years.



```
SQL> select Lname from employee where (sysdate - Bdate)/365 > 50;

LNAME
----------
Smith
Wong
Zelaya
Wallace
Naryan
Jabbar
Borg

7 rows selected.
```