Experiment 1:

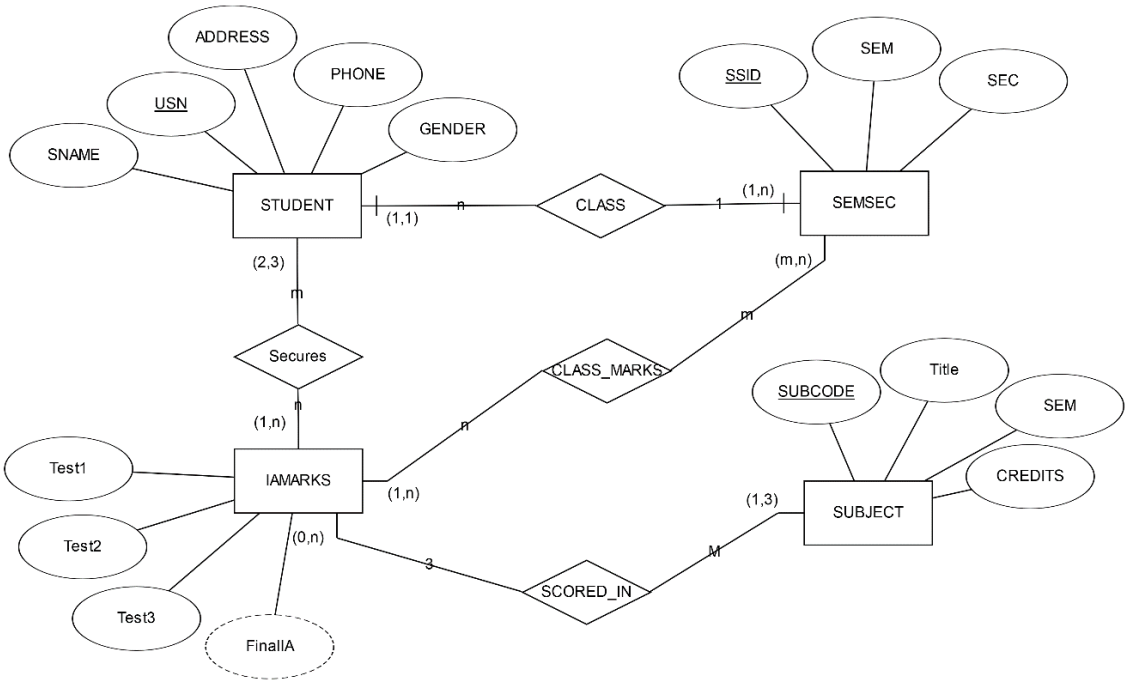
Consider following databases and draw ER diagram and convert entities and relationships to relation table for a given scenario.

1. **COLLEGE DATABASE:**
   STUDENT (*USN, SName, Address, Phone, Gender*)
   SEMSEC (*SSID, Sem, Sec*)
   CLASS (*USN, SSID*)
   SUBJECT (*Subcode, Title, Sem, Credits*)
   IAMARKS (*USN, Subcode, SSID, Test1, Test2, Test3, FinalIA*)

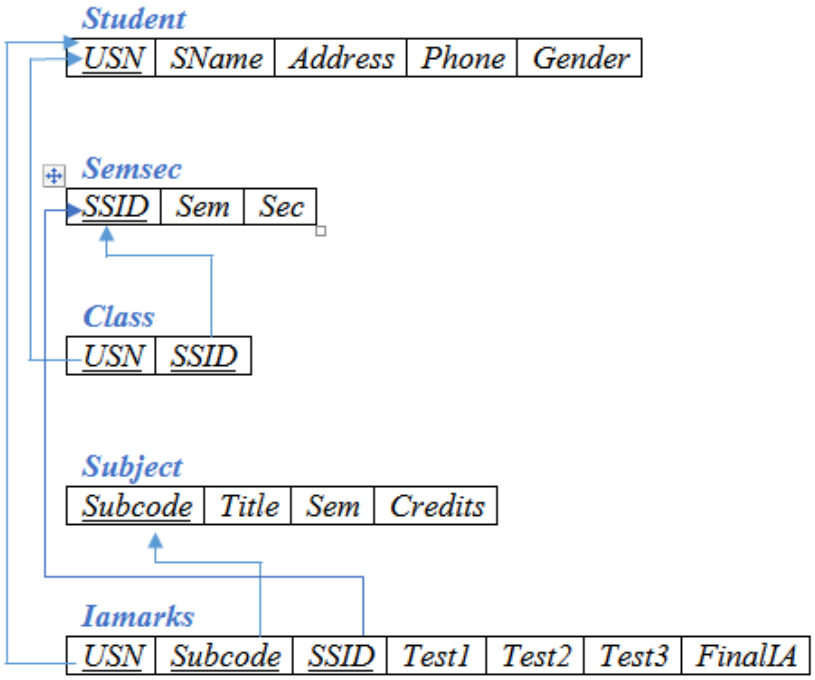
2. **COMPANY DATABASE:**
   EMPLOYEE (*SSN, Name, Address, Sex, Salary, SuperSSN, DNo*)
   DEPARTMENT (*DNo, DName, MgrSSN, MgrStartDate*)
   DLOCATION (*DNo,DLoc*)
   PROJECT (*PNo, PName, PLocation, DNo*)
   WORKS_ON (*SSN, PNo, Hours*)

SOLUTION:

College Database: E-R Diagram

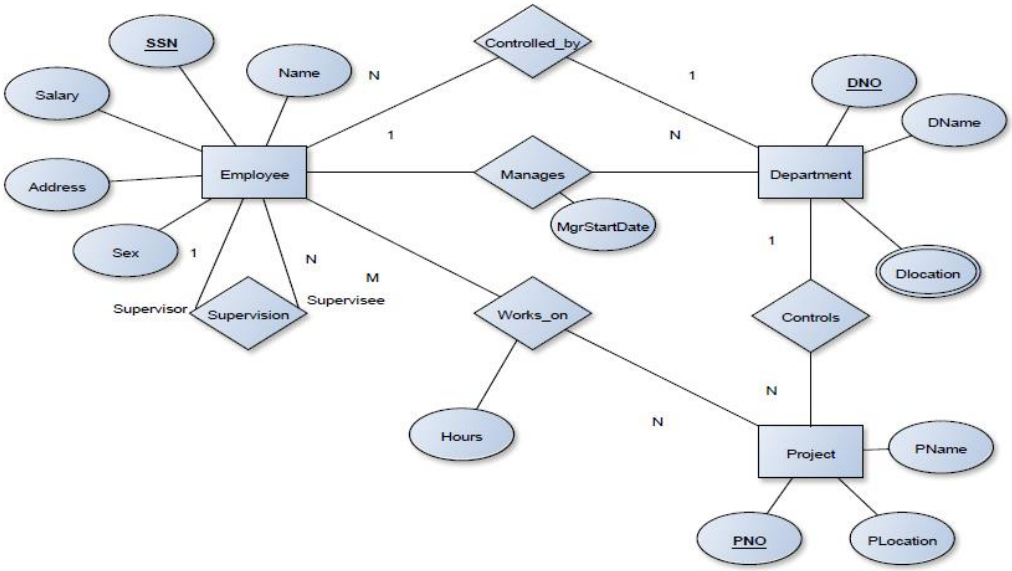


Mapping entities and relationships to relation table (Schema Diagram)

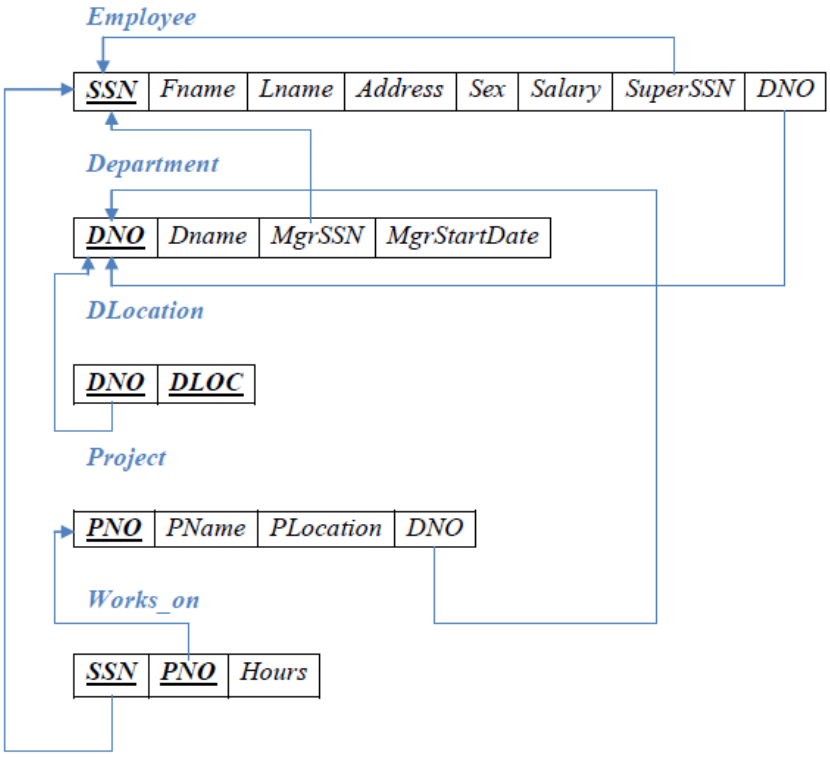


COMPANY DATABASE:

E-R Diagram



Schema Diagram



Experiment 4

Consider Dept table

DEPTNO	DNAME	LOC

Perform the following:

1. Rename the table dept as department
2. Add a new column PINCODE with not null constraints to the existing table DEPT
3. All constraints and views that reference the column are dropped automatically, along with the column.
4. Rename the column DNAME to DEPT_NAME in dept table
5. Change the data type of column loc as CHAR with size 10
6. Delete table

SOLUTION:

Create Table

SQL> CREATE TABLE DEPT(DEPTNO INTEGER, DNAME VARCHAR(10),LOC VARCHAR(4), PRIMARY KEY(DEPTNO));

1. Rename the table dept as department

   SQL> ALTER TABLE DEPT RENAME TO DEPARTMENT;
   Table altered.

2. Add a new column PINCODE with not null constraints to the existing table DEPT

   SQL> ALTER TABLE DEPARTMENT ADD(PINCODE NUMBER(6) NOT NULL);

   Table altered.

   SQL> DESC DEPARTMENT;

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPTNO                                    NOT NULL NUMBER(38)
 DNAME                                              VARCHAR2(10)
 LOC                                                VARCHAR2(4)
 PINCODE                                   NOT NULL NUMBER(6)
```

3. All constraints and views that reference the column are dropped automatically, along with the column.

   SQL> ALTER TABLE DEPARTMENT DROP column LOC CASCADE CONSTRAINTS;

   Table altered.

   SQL> desc dept

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(38)
DNAME		VARCHAR2(10)
PINCODE	NOT NULL	NUMBER(6)

4. Rename the column DNAME to DEPT_NAME in dept table

   SQL> ALTER TABLE DEPT RENAME COLUMN DNAME TO DEPT_NAME ;

   Table altered.
   SQL> DESC DEPARTMENT;

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(38)
DEPT_NAME		VARCHAR2(10)
LOC		VARCHAR2(4)
PINCODE	NOT NULL	NUMBER(6)

5. Change the datatype of colunm loc as CHAR with size 10

   SQL> ALTER TABLE DEPARTMENT MODIFY LOC CHAR(10) ;
   Table altered.
   SQL> DESC DEPARTMENT;

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(38)
DEPT_NAME		VARCHAR2(10)
LOC		CHAR(10)
PINCODE	NOT NULL	NUMBER(6)

6. Delete table

   SQL> DROP TABLE DEPARTMENT;
   Table dropped.

Experiment 5A

Consider Employee table

EMPNO	EMP_NAME	DEPT	SALARY	DOJ	BRANCH
E101	Amit	roduction	45000	12-Mar-00	Bangalore
E102	Amit	HR	70000	03-Jul-02	Bangalore
E103	sunita	lanagemer	120000	11-Jan-01	mysore
E105	sunita	IT	67000	01-Aug-01	mysore
E106	mahesh	Civil	145000	20-Sep-03	Mumbai

Perform the following

1. **Display all the fields of employee table**
2. **Retrieve employee number and their salary**
3. **Retrieve average salary of all employee**
4. **Retrieve number of employee**
5. **Retrieve distinct number of employee**
6. **Retrieve total salary of employee group by employee name and count similar names**
7. **Retrieve total salary of employee which is greater than >120000**
8. **Display name of employee in descending order**
9. **Display details of employee whose name is AMIT and salary greater than 50000;**

1. Display all the fields of employee table

SQL> select * from employee;

EMPNO	EMP_NAME	DEPT	SALARY	DOJ	BRANCH
E101	Amit	Production	45000	12-MAR-00	Bangalore
E102	Amit	HR	70000	03-JUL-02	Bangalore
E103	sunita	Management	120000	11-JAN-01	mysore
E105	sunita	IT	67000	01-AUG-01	mysore
E106	mahesh	Civil	145000	20-SEP-03	Mumbai

2. Retrieve employee number and their salary

SQL> select empno, salary from employee;

EMPNO	SALARY
E101	45000
E102	70000
E103	120000
E105	67000
E106	145000

3. Retrieve average salary of all employee

```
SQL> select  avg(salary) from employee;

AVG(SALARY)
-----------
    89400
```

4. Retrieve number of employee

```
SQL> select count(*) from employee;

 COUNT(*)
----------
       5
```

5. Retrieve distinct number of employee

```
SQL> select count(DISTINCT emp_name) from employee;
COUNT(DISTINCTEMP_NAME)
-----------------------
                     3
```

6. Retrieve total salary of employee group by employee name and count similar names

```
SQL> SELECT EMP_NAME, SUM(SALARY),COUNT(*) FROM EMPLOYEE
  2  GROUP BY(EMP_NAME);

EMP_NAME                       SUM(SALARY)   COUNT(*)
------------------------------ -----------   ----------
mahesh                         145000        1
sunita                         187000         2
Amit                           115000        2
```

7. Retrieve total salary of employee which is greater than >120000

```
SQL> SELECT EMP_NAME, SUM(SALARY) FROM EMPLOYEE
  2  GROUP BY(EMP_NAME)
  3  HAVING SUM(SALARY)>120000;

EMP_NAME                       SUM(SALARY)
------------------------------ -----------
mahesh                         145000
sunita                         187000
```

8. Display name of employee in descending order

```
SQL> select emp_name from employee
  2  order by emp_name desc;
```

```
EMP_NAME
------------------------------
sunita
sunita
mahesh
Amit
Amit
```

9. Display details of employee whose name is AMIT and salary greater than 50000;

```
SQL> select * from employee
  2  where emp_name='Amit' and salary>50000;

EMPNO      EMP_NAME        DEPT      SALARY      DOJ          BRANCH
E102        Amit           HR        70000       03-JUL-02    Bangalore
```

Viva Questions

1. What is SQL?

Structured Query Language

2. What is database?

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

3. What is DBMS?

It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

4. What is a Database system?

The database and DBMS software together is called as Database system.

5. Advantages of DBMS?

Redundancy is controlled.

Unauthorized access is restricted.

Providing multiple user interfaces.

Enforcing integrity constraints.

Providing backup and recovery.

6. Disadvantage in File Processing System?

Data redundancy & inconsistency.

Difficult in accessing data.

Data isolation.

Data integrity.

Concurrent access is not possible.

Security Problems.

7. Describe the three levels of data abstraction?

There are three levels of abstraction:

Physical level: The lowest level of abstraction describes how data are stored.

Logical level*:* The next higher level of abstraction, describes what data are stored in database and what relationship among those data.

View level:The highest level of abstraction describes only part of entire database.

8. Define the "integrity rules"

There are two Integrity rules.
Entity Integrity:States that ―Primary key cannot have NULL value‖

Referential Integrity:States that ―Foreign Key can be either a NULL value or should be Primary Key value of other relation.

9. What is extension and intension?

Extension - It is the number of tuples present in a table at any instance. This is time dependent.
Intension -It is a constant value that gives the name, structure of table and the constraints laid on it.

10. What is Data Independence?

Data independence means that ―the application is independent of the storage structure and access strategy of data‖. In other words, The ability to modify the schema definition in one level should not affect the schema definition in the next higher level.
Two types of Data Independence:

- Physical Data Independence: Modification in physical level should not affect the logical level.
- Logical Data Independence: Modification in logical level should affect the view level.

NOTE: Logical Data Independence is more difficult to achieve

11. What is a view? How it is related to data independence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that direct represents the view instead a definition of view is stored in data dictionary. Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

12. What is Data Model?

A collection of conceptual tools for describing data, data relationships data semantics and constraints.

13. What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

14. What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

15. What is an Entity?

It is an 'object' in the real world with an independent existence.

16. What is an Entity type?

It is a collection (set) of entities that have same attributes.

17. What is an Entity set?

It is a collection of all entities of particular entity type in the database.

18. What is an Extension of entity type?

The collections of entities of a particular entity type are grouped together into an entity set.

19. What is an attribute?

It is a particular property, which describes the entity.

20. What is a Relation Schema and a Relation?

A relation Schema denoted by R(A1, A2, …, An) is made up of the relation name R and the list of attributes Ai that it contains. A relation is defined as a set of tuples. Let r be the relation which contains set tuples (t1, t2, t3, ...,tn). Each tuple is an ordered list of n-values t=(v1,v2, ..., vn).

21. What is degree of a Relation?

It is the number of attribute of its relation schema.

22. What is Relationship?

It is an association among two or more entities.

23. What is Relationship set?

The collection (or set) of similar relationships.

24. *What is Relationship type?*

Relationship type defines a set of associations or a relationship set among a given set of entity types.

25. What is degree of Relationship type?

It is the number of entity type participating.

26. What is DDL (Data Definition Language)?

A data base schema is specified by a set of definitions expressed by a special language called DDL.

27. What is VDL (View Definition Language)?

It specifies user views and their mappings to the conceptual schema.

28. What is SDL (Storage Definition Language)?

This language is to specify the internal schema. This language may specify the mapping between two schemas.

29. What is Data Storage - Definition Language?

The storage structures and access methods used by database system are specified by a set of definition in a special type of DDL called data storage- definition language.

30. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organized by appropriate data model.

 Procedural DML or Low level: DML requires a user to specify what data are needed and how to get those data.

 Non-Procedural DML or High level: DML requires a user to specify what data are needed without specifying how to get those data.

31. What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

32. What is Relational Algebra?

It is a procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation.

33. What is Relational Calculus?

It is an applied predicate calculus specifically tailored for relational databases proposed by E.F. Codd. E.g. of languages based on it are DSL, ALPHA, QUEL.

34. What is normalization?

It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

- Minimizing redundancy
- Minimizing insertion, deletion and update anomalies.

35. What is Functional Dependency?

A Functional dependency is denoted by X Y between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if t1[X] = t2[X] then they have t1[Y] = t2[Y]. This means the value of X component of a tuple uniquely determines the value of component Y.

36. When is a functional dependency F said to be minimal?

- Every dependency in F has a single attribute for its right hand side.
- We cannot replace any dependency X A in F with a dependency Y A where Y is a proper subset of X and still have a set of dependency that is equivalent to F.
- We cannot remove any dependency from F and still have set of dependency that is equivalent to F.

37. What is Multivalued dependency?

Multivalued dependency denoted by X Y specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation r of R: if two tuples t1 and t2 exist in r such that t1[X] = t2[X] then t3 and t4 should also exist in r with the following properties

- t3[x] = t4[X] = t1[X] = t2[X]
- t3[Y] = t1[Y] and t4[Y] = t2[Y]
- t3[Z] = t2[Z] and t4[Z] = t1[Z]

where [Z = (R-(X U Y))]

38. What is Lossless join property?

It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

39. What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

40. What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency X Y is fully functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

41. What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

42. What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD X A either of the following is true

- X is a Super-key of R.
- A is a prime attribute of R.

In other words, if every non prime attribute is non-transitively dependent on primary key.

43. What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies additional constraints that for every FD X A, X must be a candidate key.

44. What is 4NF?

A relation schema R is said to be in 4NF if for every Multivalued dependency X Y that holds over R, one of following is true

- X is subset or equal to (or) XY = R.
- X is a super key.

45. What is 5NF?

A Relation schema R is said to be 5NF if for every join dependency {R1, R2, ...,Rn} that holds R, one the following is true

- Ri = R for some i.
- The join dependency is implied by the set of FD, over R in which the left side is key of R.