Mahatma Gandhi Mission's College of Engineering and Technology
Kamothe, Navi Mumbai

Assignment -3

Subject-DBMS Div-A & B Sem-IV Class –SE

Q.No	Question
Q1.Fill in the blanks	
a)	The relational model is based on the core concept of relation or table.
b)	The first attempt at a large implementation of Codd's relational model was relational.
c)	In the RDBMS terminology, a record is called a row.
d)	Degree of a table means the number of column in a table.
e)	The number of attributes in a relation is called the degree of the relation.
Q2. Choose Correct Options	
a)	Which of the following is not a relational algebra function? a) Select b) Project [✓ c) Manipulate] d) Union
b)	How is the left outer join symbol represented in relational algebra? [✓ a) ⟕] b) ⟖ c) ⟗ d) ⋈
c)	Which of the following is used to denote the selection operation in relational algebra? a) Pi (Greek) [✓ b) Sigma (Greek)] c) Lambda (Greek) d) Omega (Greek)
d)	The ___________ operation, denoted by −, allows us to find tuples that are in one relation but are not in another. a) Union [✓ b) Set-difference] c) Difference d) Intersection
e)	Which is a join condition contains an equality operator: [✓ a) Equijoins] b) Cartesian c) Natural d) Left
Q3. state whether the following statements are true or false (Give Reasons)	
a)	There exists a division operator in Relational Algebra [✓ a) True] b) False
b)	Projection is used for selecting tuples a) True

b) False

c) Union selects common data in both table.

a) True

b) False

Q4. Name the following or define or design the following

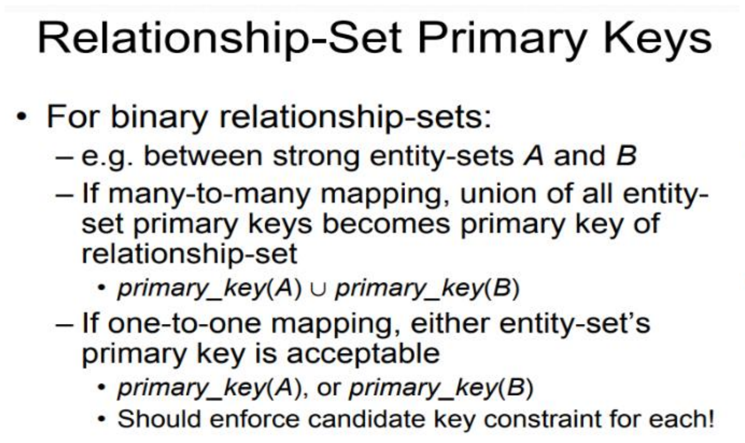
a)
Ans:

Define Relational model

The relational data model provides conceptual tools to design the database schema of the relational database. The relational model describes the data, relationship between that data, data sematic and constraints on the data in the relational database.

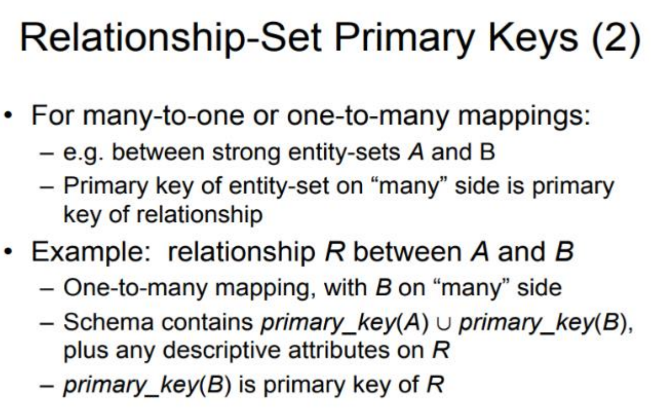
b)
Ans:

List the types of keys.



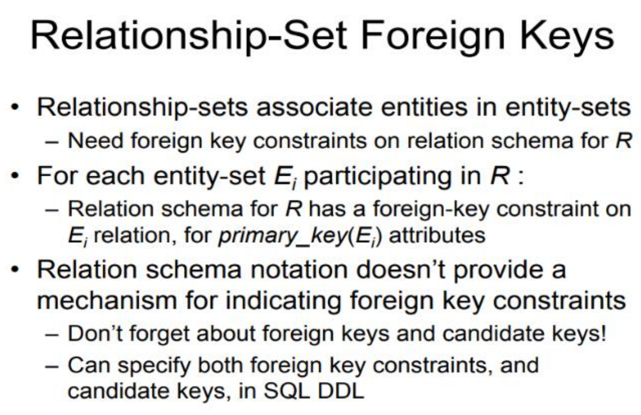
Relationship-Set Primary Keys

- For binary relationship-sets:
  - e.g. between strong entity-sets *A* and *B*
  - If many-to-many mapping, union of all entity-set primary keys becomes primary key of relationship-set
    - *primary_key*(*A*) ∪ *primary_key*(*B*)
  - If one-to-one mapping, either entity-set's primary key is acceptable
    - *primary_key*(*A*), or *primary_key*(*B*)
    - Should enforce candidate key constraint for each!



Relationship-Set Primary Keys (2)

- For many-to-one or one-to-many mappings:
  - e.g. between strong entity-sets *A* and B
  - Primary key of entity-set on "many" side is primary key of relationship
- Example: relationship *R* between *A* and *B*
  - One-to-many mapping, with *B* on "many" side
  - Schema contains *primary_key*(*A*) ∪ *primary_key*(*B*), plus any descriptive attributes on *R*
  - *primary_key*(*B*) is primary key of *R*



Relationship-Set Foreign Keys

- Relationship-sets associate entities in entity-sets
  - Need foreign key constraints on relation schema for *R*
- For each entity-set E_i participating in *R* :
  - Relation schema for *R* has a foreign-key constraint on E_i relation, for *primary_key*(E_i) attributes
- Relation schema notation doesn't provide a mechanism for indicating foreign key constraints
  - Don't forget about foreign keys and candidate keys!
  - Can specify both foreign key constraints, and candidate keys, in SQL DDL

c)
Ans:

List the Unary operators in relational algebra

Relational Algebra Operations -

1. Select Operator (σ) –

Select Operator is denoted by sigma (σ) and it is used to find the tuples (or rows) in a relation (or table) which satisfy the given condition. If you understand little bit of SQL then you can think of it as a where clause in SQL, which is used for the same purpose.

Syntax of Select Operator (σ)

σ Condition/Predicate(Relation/Table name)

2.Project Operator (∏) –

Project operator is denoted by ∏ symbol and it is used to select desired columns (or attributes) from a table (or relation). Project operator in relational algebra is similar to the Select statement in SQL.

Syntax of Project Operator (∏) -

∏ column_name1, column_name2, ..., column_nameN(table_name)

Q5. Answer the following questions in brief (20 to 30 words)

a)
Ans:

Explain Keys in DBMS

- Keys play a very important role in DBMS. They are crucial for the arrangement of tables in the database.
- Keys uniquely identify records or a combination of records from huge database tables.
- Database keys are also useful in establishing a relationship between one table with other tables.

b)
Ans:

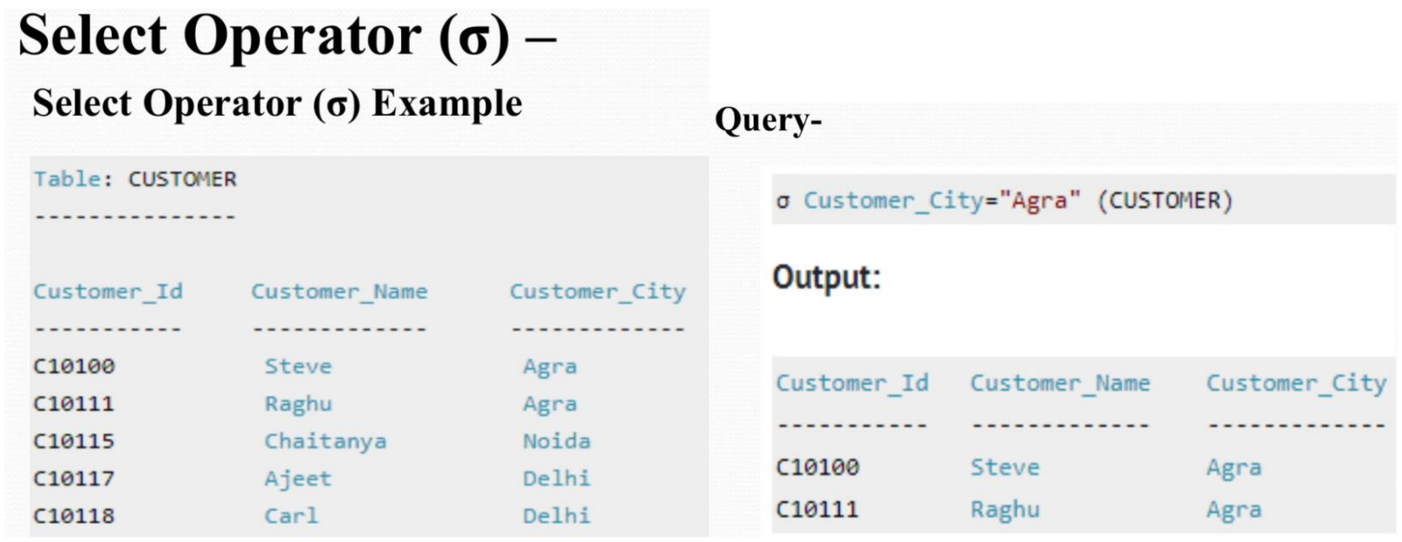
Explain selection , projection and rename operators in relational algebra

1. Select Operator (σ) –

Select Operator is denoted by sigma (σ) and it is used to find the tuples (or rows) in a relation (or table) which satisfy the given condition. If you understand little bit of SQL then you can think of it as a where clause in SQL, which is used for the same purpose.

Syntax of Select Operator (σ)

σ Condition/Predicate(Relation/Table name)



Select Operator (σ) –

Select Operator (σ) Example

```
Table: CUSTOMER
---------------

Customer_Id     Customer_Name     Customer_City
-----------     -------------     -------------
C10100           Steve             Agra
C10111           Raghu             Agra
C10115           Chaitanya         Noida
C10117           Ajeet             Delhi
C10118           Carl              Delhi
```

Query-

```
σ Customer_City="Agra" (CUSTOMER)
```

Output:

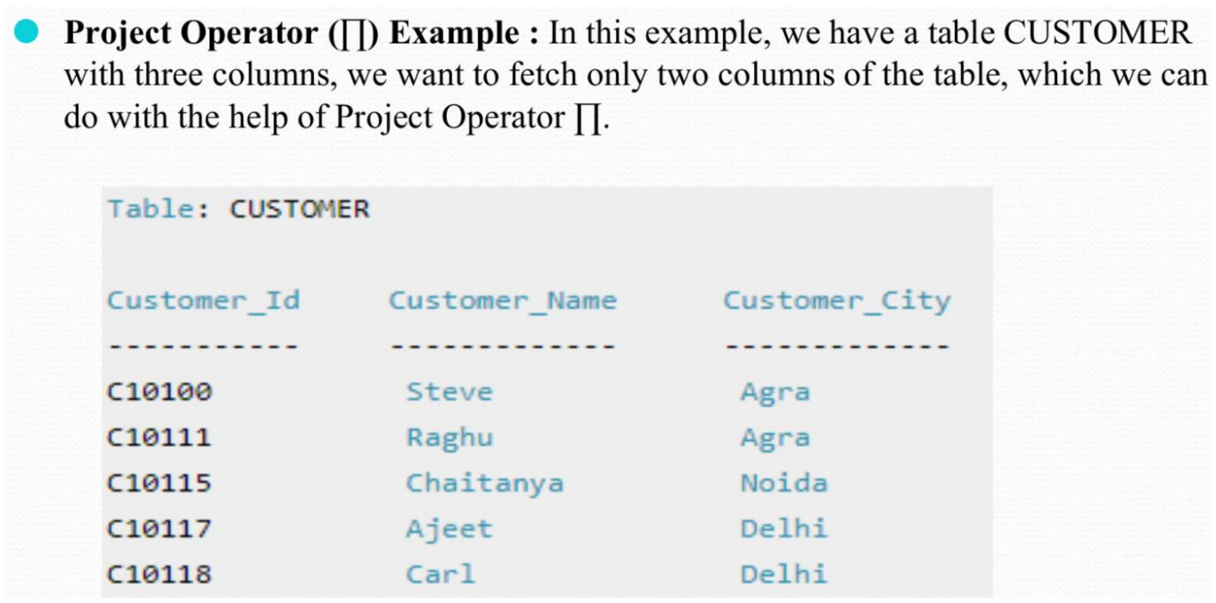
```
Customer_Id     Customer_Name     Customer_City
-----------     -------------     -------------
C10100          Steve             Agra
C10111          Raghu             Agra
```

2.Project Operator (∏) –

Project operator is denoted by ∏ symbol and it is used to select desired columns (or attributes) from a table (or relation). Project operator in relational algebra is similar to the Select statement in SQL.

Syntax of Project Operator (∏) -

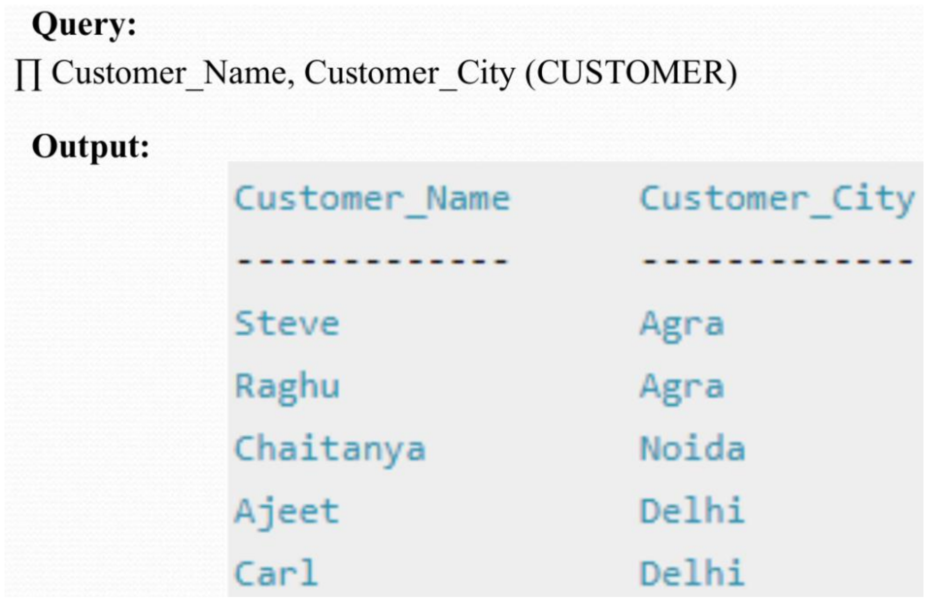
∏ column_name1, column_name2, ..., column_nameN(table_name)



- **Project Operator (∏) Example :** In this example, we have a table CUSTOMER with three columns, we want to fetch only two columns of the table, which we can do with the help of Project Operator ∏.

Table: CUSTOMER

Customer_Id	Customer_Name	Customer_City
C10100	Steve	Agra
C10111	Raghu	Agra
C10115	Chaitanya	Noida
C10117	Ajeet	Delhi
C10118	Carl	Delhi



Query:

∏ Customer_Name, Customer_City (CUSTOMER)

Output:

Customer_Name	Customer_City
Steve	Agra
Raghu	Agra
Chaitanya	Noida
Ajeet	Delhi
Carl	Delhi

Rename (ρ):

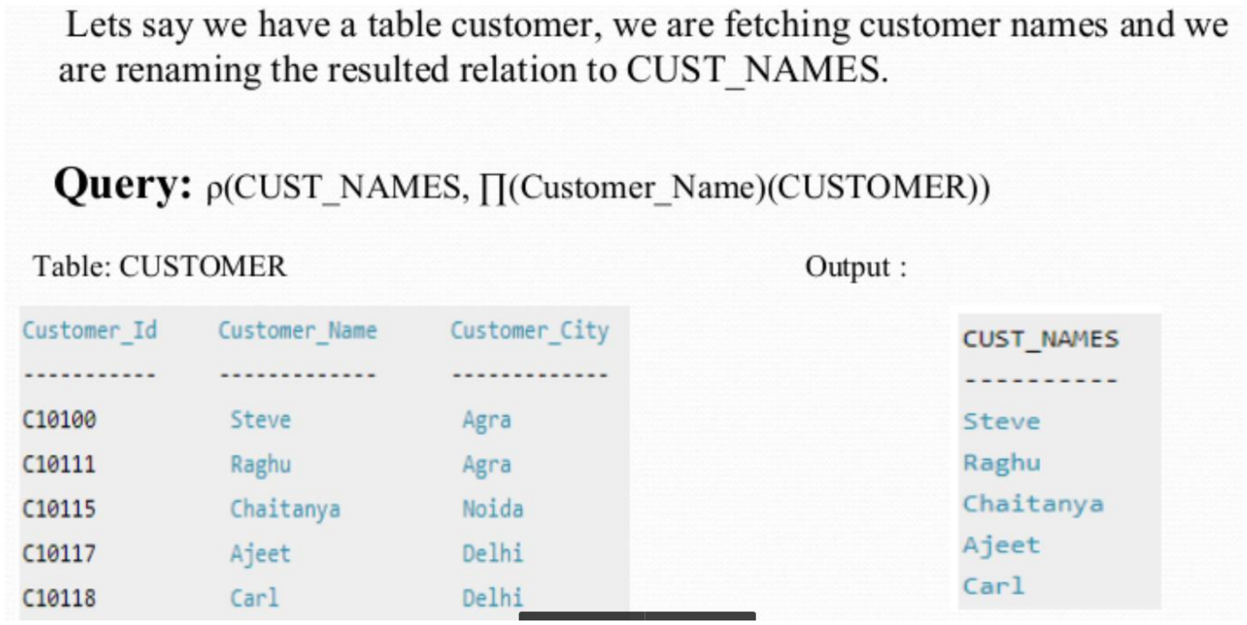
Rename (ρ) operation can be used to rename a relation or an attribute of a relation.

Rename (ρ) Syntax:

ρ(new_relation_name, old_relation_name)

Rename (ρ) Example

Let’s say we have a table customer, we are fetching customer names and we are renaming the resulted relation to CUST_NAMES.



Lets say we have a table customer, we are fetching customer names and we are renaming the resulted relation to CUST_NAMES.

Query: ρ(CUST_NAMES, ∏(Customer_Name)(CUSTOMER))

Table: CUSTOMER

```
Customer_Id     Customer_Name     Customer_City
-----------     -------------     -------------
C10100           Steve             Agra
C10111           Raghu             Agra
C10115           Chaitanya         Noida
C10117           Ajeet             Delhi
C10118           Carl              Delhi
```

Output :

```
CUST_NAMES
----------
Steve
Raghu
Chaitanya
Ajeet
Carl
```

c)
Ans:

Difference between Natural join and Inner join

SN	Natural Join	Inner Join
1.	It joins the tables based on the same column names and their data types.	It joins the tables based on the column ON clause explicitly.
2.	It always returns unique columns in the result set.	It returns all the attributes of bot duplicate columns that match the ON c
3.	If we have not specified any condition in this join, it returns the records based on the common columns.	It returns only those rows that exist in b

4.	The syntax of natural join is given below:
	SELECT [column_names | *]
	FROM table_name1
	NATURAL JOIN table_name2;
		The syntax of inner join is given below:
		SELECT [column_names | *]
		FROM table_name1
		INNER JOIN table_name2
		ON table_name1.column_
		table_name2.column_name;

. Q6. Answer the following questions in brief (50 to 70 words)

a)
Ans:

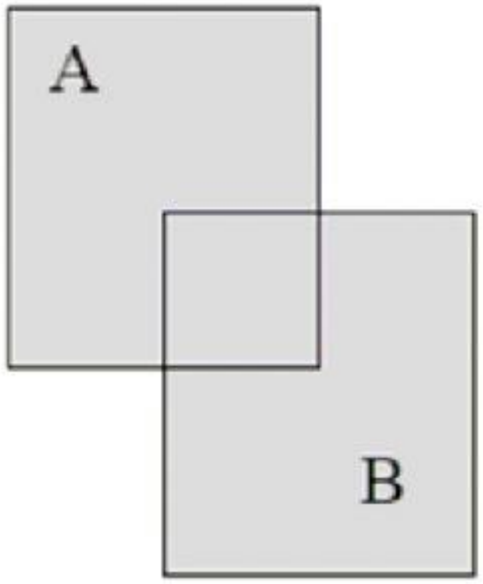
Explain SET operators in detail

The set operators are availed to combine information of similar type from one or more than one table. The set operators look similar to SQL joins although there is a big difference. SQL joins tends to combine columns from different tables, whereas SQL set operators combine rows from distinct queries. There are distinct types of set operators in SQL. Set Operator in SQL are same as of DBMS. This article will cover all the set operations in SQL like Union, Union All, Intersect, Minus with Examples.

- **Union**
- **Union all**
- **Intersect**
- **Minus**

1. Union

This set operator is used to combine the outputs of two or more queries into a single set of rows and columns having different records.



R=UNION(A,B)

Example Of UNION

Table A

Roll No.	**Name**
234	Mark
235	Steve
236	Harry

Table B

Roll No.	Name
236	Harry
237	James
238	Jessica

UNION Set Operator SQL Query

```
SQL> SELECT * FROM A

UNION
SELECT * FROM B
```

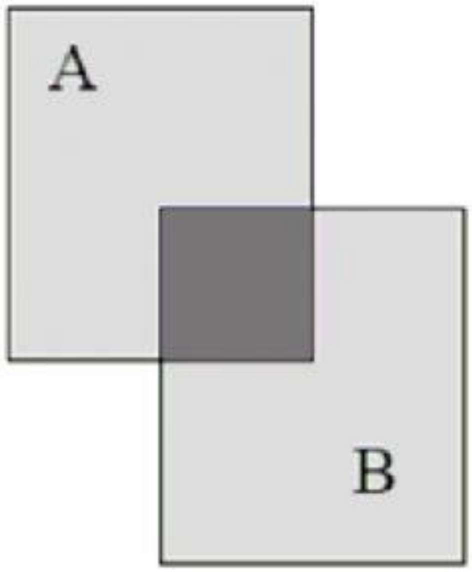
Result of the above UNION Operator will be

Roll No.	Name
234	Mark
235	Steve
236	Harry
237	James
238	Jessica

Also See: Pagination in SQL Server, MySQL and Oracle with Examples

2. Union All

This set operator is used to join the outputs of two or more queries into a single set of rows and columns without the removal of any duplicates.



R=UNION ALL(A,B)

Example of UNION ALL

Table A

Roll No.	**Name**
234	Mark
235	Steve
236	Harry

Table B

Roll No.	**Name**
236	Harry
237	James
238	Jessica

UNION ALL Set Operator SQL Query

SQL> SELECT * FROM A

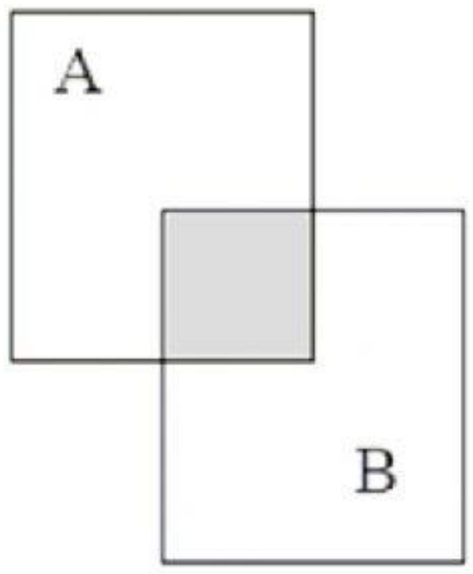
UNION ALL

SELECT * FROM B
Result of the above UNION ALL Operator will be

Roll No.	Name
234	Mark
235	Steve
236	Harry
236	Harry
237	James
238	Jessica

3. INTERSECT

This set operator is availed to retrieve the information which is common in both tables. The number of columns and data type must be same in intersect set operator.



R=INTERSECT(A,B)

Example on INTERSECT Operator
Table A

Roll No.	Name
234	Mark

235	Steve
236	Harry

Table B

Roll No.	Name
236	Harry
237	James
238	Jessica

INTERSECT Set Operator Query

SQL> SELECT * FROM A

INTERSECT

SELECT * FROM B

Result of the above INTERSECT Operator will be

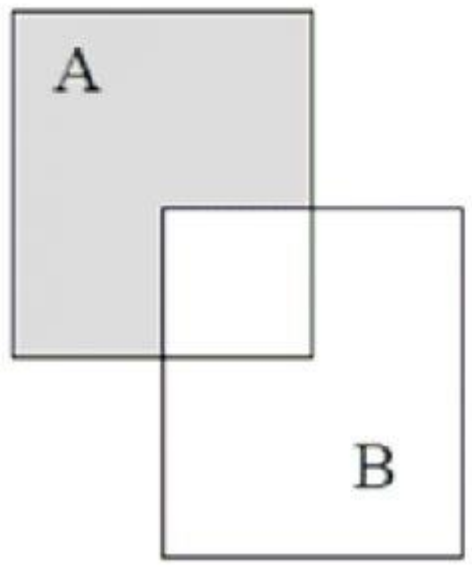
Roll No.	Name
236	Harry

NOTE: MYSQL does not support INTERSECT set operator

Also Read: Explain RDBMS in Detail

4. MINUS

This set operator is availed to retrieve the information of one table which is not available in another table.



R=MINUS(A,B)

Example of MINUS Operator

Table A

Roll No.	**Name**
234	Mark
235	Steve
236	Harry

Table B

Roll No.	**Name**
236	Harry
237	James
238	Jessica

MINUS Set Operator SQL Query

```
SQL> SELECT * FROM A

MINUS
```

SELECT * FROM B
Result of the above MINUS Operator will be

Roll No.	Name
234	Mark
235	Steve

Important points of set operators:

- INTERSECT operator and UNION operator is commutative.
- Performance wise, UNION ALL shows higher performance as compared to UNION as a result of resources aren't wasted in filtering duplicates and sorting the result set.
- Set operators are the part of subqueries.
- Set operators cannot be utilized to choose statements containing TABLE assortment expressions.
- The LONG, BLOB, CLOB, BFILE, VARRAY, or nested table aren't allowable to be used in Set operators. For an update, a clause isn't allowed with the set operators.

So it was all about **Set Operators in SQL with examples**, if you have any query then please comment below and tell us.

b) Ans:

Explain Outer Join in details

In a relational DBMS, we follow the principles of normalization that allows us to minimize the large tables into small tables. By using a select statement in Joins, we can retrieve the big table back. Outer joins are of following three types.

1. Left outer join
2. Right outer join
3. Full outer join

Creating a database : Run the following command to create a database.

```
Create database testdb;
```

Using the database : Run the following command to use a database.

```
use testdb;
```

Adding table to the database : Run the following command to add tables to a database.

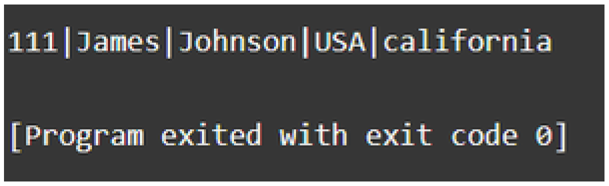
```
CREATE TABLE Students (

   StudentID int,

   LastName varchar(255),

   FirstName varchar(255),

   Address varchar(255),

   City varchar(255)

);
```

Inserting rows into database :

```
INSERT INTO students (
StudentID,
LastName,
FirstName,
Address,
City
)
VALUES
(
111,
'James',
 'Johnson',
 'USA',
 california
);
```

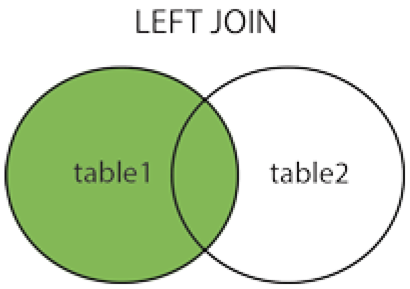
Output of database :
Type the following command to get output.

```
SELECT  * FROM students;
```



```
111|James|Johnson|USA|california

[Program exited with exit code 0]
```

Types of outer join :
1.**Left Outer Join** : The left join operation returns all record from left table and matching records from the right table. On a matching element not found in right table, NULL is represented in that case.

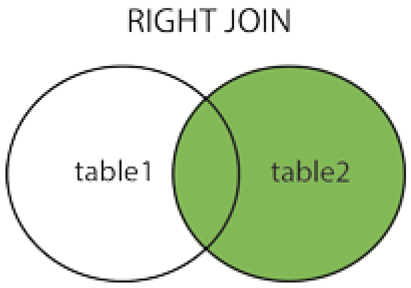


Syntax :

```
SELECT column_name(s)
FROM table1
LEFT JOIN Table2
```

```
ON Table1.Column_Name=table2.column_name;
```

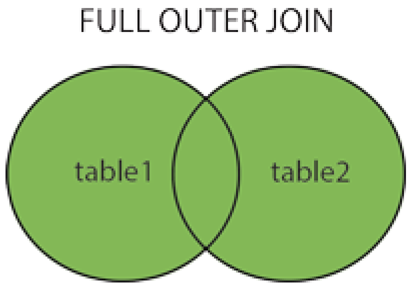
2. Right Outer Join : The right join operation returns all record from right table and matching records from the left table. On a matching element not found in left table, NULL is represented in that case.



Syntax :

```
SELECT column_name(s)

FROM table1

RIGHT JOIN table2

ON table1.column_name = table2.column_name;
```

3. Full Outer Join : The full outer Join keyword returns all records when there is a match in left or right table records.



Syntax:

```
SELECT column_name
FROM table1
FULL OUTER JOIN table2
ON table1.columnName = table2.columnName
WHERE condition;
```

Example :

Creating 1st Sample table students.

```
CREATE TABLE students (

 id INTEGER,

 name TEXT NOT NULL,

 gender TEXT NOT NULL

);

-- insert some values
```

```
INSERT INTO students VALUES (1, 'Ryan', 'M');

INSERT INTO students VALUES (2, 'Joanna', 'F');

INSERT INTO students Values (3, 'Moana', 'F');
```

Creating 2nd sample table college.

```
CREATE TABLE college (
 id INTEGER,
 classTeacher TEXT NOT NULL,
 Strength TEXT NOT NULL
);
-- insert some values
INSERT INTO college VALUES (1, 'Alpha', '50');
INSERT INTO college VALUES (2, 'Romeo', '60');
INSERT INTO college Values (3, 'Charlie', '55');
```

Performing outer join on above two tables.

```
SELECT College.classTeacher, students.id
FROM College
FULL OUTER JOIN College ON College.id=students.id
ORDER BY College.classTeacher;
```

The above code will perform a full outer join on tables students and college and will return the output that matches the id of college with id of students. The output will be class Teacher from college table and id from students table. The table will be ordered by class Teacher from college table.

Class Teacher	Id
Alpha	1
Romeo	2
Charlie	3

c)
Ans:

Write difference between cartein product and Natural join

Cross-join is SQL 99 join and Cartesian product is Oracle Proprietary join. A cross-join that does not have a 'where' clause gives the Cartesian product. Cartesian product result-set contains the number of rows in the first table, multiplied by the number of rows in second table.

Q7. Think and Answer

a) Explain foreign key with example

Ans:

In simpler words, a foreign key is a set of attributes that references a candidate key. For example, a table called TEAM may have an attribute, MEMBER_NAME, which is a foreign key referencing a candidate key, PERSON_NAME, in the PERSON table. Since MEMBER_NAME is a foreign key, any value existing as the name of a member in TEAM must also exist as a person's name in the PERSON table; in other words, every member of a TEAM is also a PERSON.

b) Explain steps for mapping ER diagram into Relational database

Ans:

ER Model, when conceptualized into diagrams, gives a good overview of entity-relationship, which is easier to understand. ER diagrams can be mapped to relational schema, that is, it is possible to create relational schema using ER diagram. We cannot import all the ER constraints into relational model, but an approximate schema can be generated.

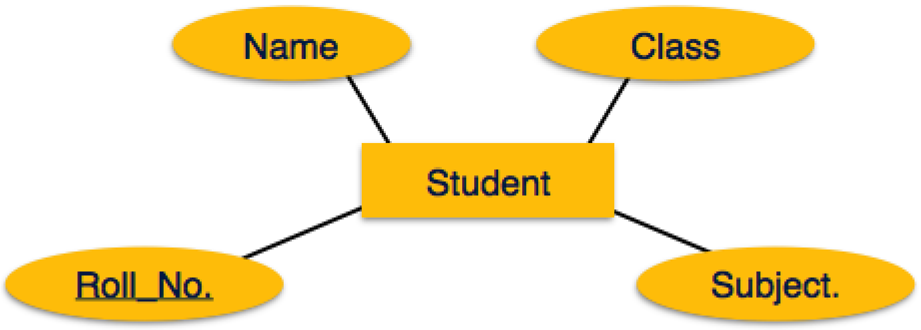
There are several processes and algorithms available to convert ER Diagrams into Relational Schema. Some of them are automated and some of them are manual. We may focus here on the mapping diagram contents to relational basics.

ER diagrams mainly comprise of −

- Entity and its attributes
- Relationship, which is association among entities.

Mapping Entity

An entity is a real-world object with some attributes.

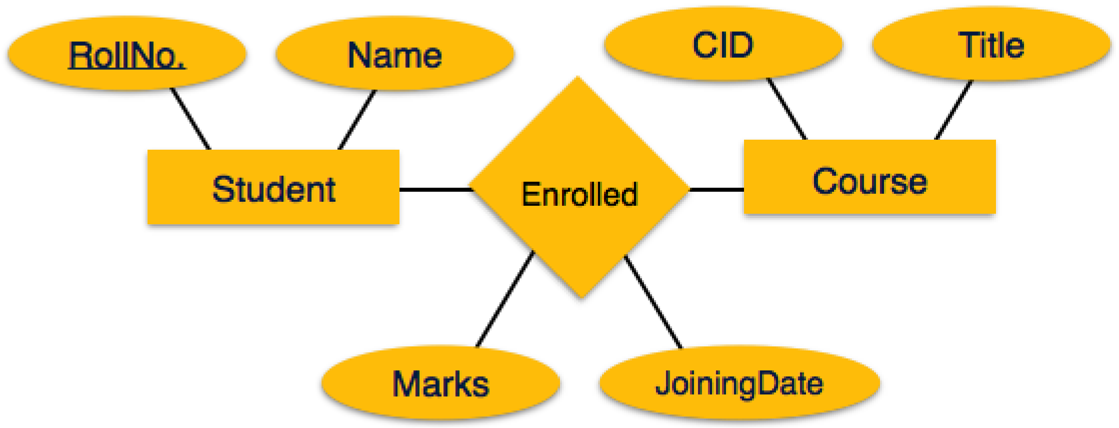


Mapping Process (Algorithm)

- Create table for each entity.
- Entity's attributes should become fields of tables with their respective data types.
- Declare primary key.

Mapping Relationship

A relationship is an association among entities.

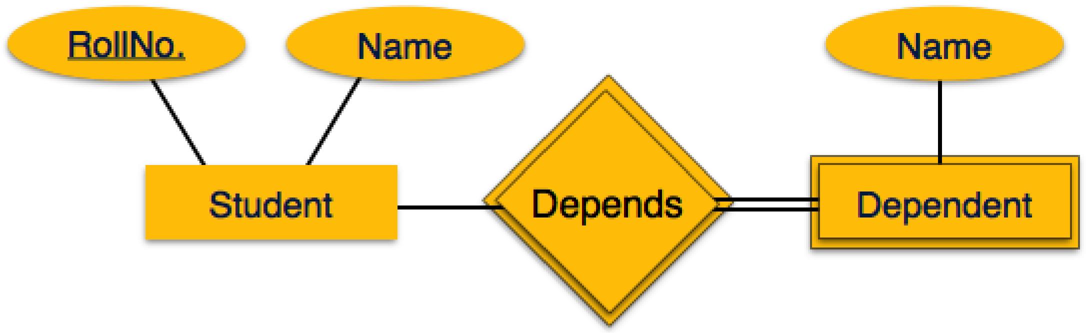


Mapping Process

- Create table for a relationship.
- Add the primary keys of all participating Entities as fields of table with their respective data types.
- If relationship has any attribute, add each attribute as field of table.
- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

Mapping Weak Entity Sets

A weak entity set is one which does not have any primary key associated with it.

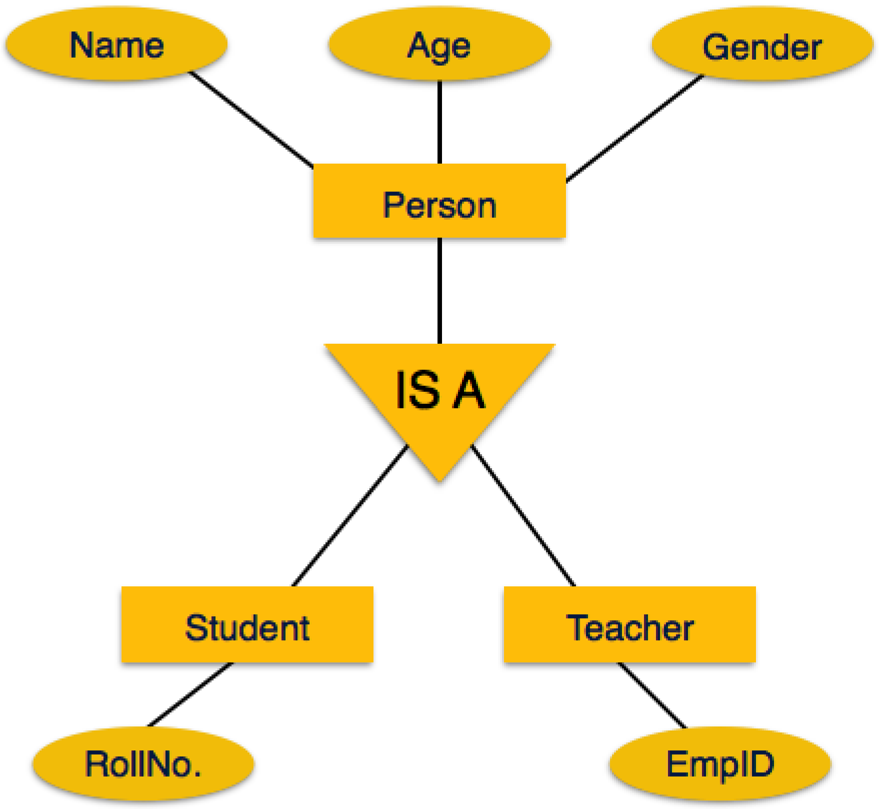


Mapping Process

- Create table for weak entity set.
- Add all its attributes to table as field.
- Add the primary key of identifying entity set.
- Declare all foreign key constraints.

Mapping Hierarchical Entities

ER specialization or generalization comes in the form of hierarchical entity sets.

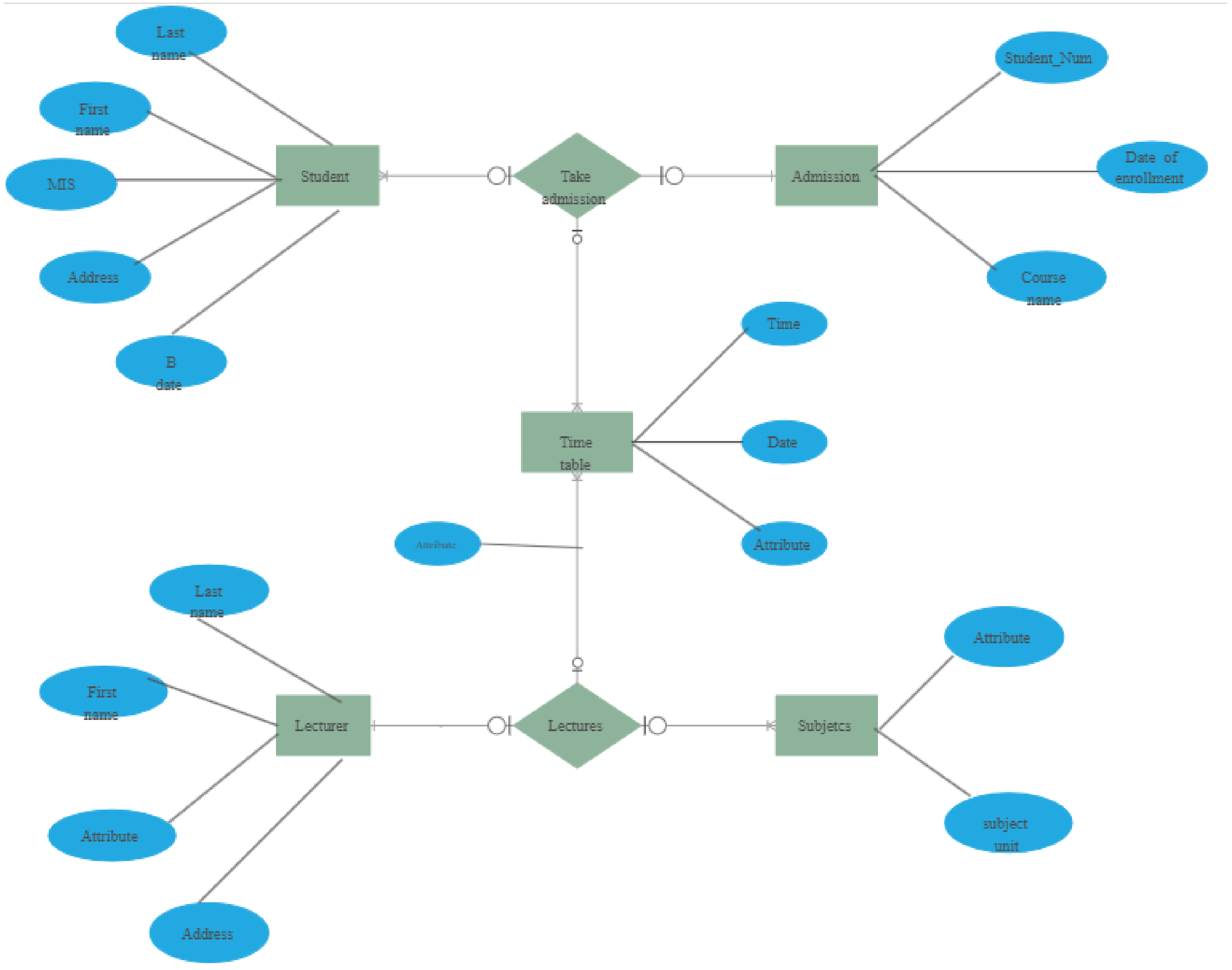


Mapping Process

- Create tables for all higher-level entities.
- Create tables for lower-level entities.
- Add primary keys of higher-level entities in the table of lower-level entities.
- In lower-level tables, add all other attributes of lower-level entities.
- Declare primary key of higher-level table and the primary key for lower-level table.
- Declare foreign key constraints.

Q8. **My Ideas**

a) **Ans:**	Draw college management system ER diagram which includes aggregation generalization and specialization and convert into relational database



b) Consider the following relational for database

Student(ssn,name,subject,DOB)

Course(Courseid,name,dept)

Enroll(ssn,coursed,semsester,grade);

Book_issued(coursed,semester,isbn)

Text(isbn,title,publisher,author)

1)write the query to select all course available in institute

2)find various book title and authors for semester less than 8

Ans: Select * from institute course where course-

	Name; Table Course Name Comp Civil Mech IT

Course Name
Comp
Civil
Mech
IT