

UNIT - 1.
Q.N.1.

Define data and their importance? Give real life example.

Data is any kind of information which may include any particular information on any particular category. Those information can be used for any website, application or any other client side particular used to store for future purpose. The most common information is User information which includes user personal address, contact details, address, banking information etc. These data are very critical because they can be very helpful to manipulate update retain customer for any kind of business platforms. Example: we give our information to sim registrar company like ncell/ntc/smart cell and so on meanwhile when they have new updates offers they send text to all the customers and customer knows about the service and buy the service.

Q.N.2

Differentiate between Database and Database Management System.

Difference between Database and DBMS is given below:

Category	Database	DBMS
1. Definition & Storage	A collection of related pieces of data, whose purpose is to solve the data management needs is called a Database. Besides computers, databases can even be maintained in physical ledgers, books or papers.	Database Management System are very complex software which store the data on the secondary storage devices and which are used to change the databases. In a database management system all the records are maintained on a computer.
2. Data Retrieval	The data retrieval from the databases can be done manually, through queries (C, C++, Java etc.)	We can retrieve the data from the DBMS through queries written in SQL. The retrieval of information is very quick.
3. Speed	When SQL is not used to retrieve information, it can be very slow.	
4. Access	The databases are designed	The database management system is

Category	Database	DBMS
	for accessing the data at different times.	designed to access the data at the real time.
5.	Data Manipulation	<p>Very little information can be changed at a time.</p> <p>In the DBMS, a lot of data can be manipulated at one time.</p>

Q.N.3. List out the major advantages and disadvantages of DBMS. How data are stored in database?

The advantages of a database management system (DBMS) are:

1. Improved data sharing

DBMS helps to create an environment in which end users have better access managed data, which makes it possible for end users to respond quickly to changes in their environment.

2. Improved data security

The more users access the data, the greater the risks of data security breaches. A DBMS provides a framework for better data privacy and security.

3. Better data integration

Wider access to well-managed data promotes an integrated view of the organization's operations and a clearer view. It becomes much easier to see how data integration of the company affect other segments.

The Disadvantages of a database management system (DBMS) are:

1. Increased costs

one of the disadvantages of dbms is it require complex hardware and software and highly skilled personnel to operate. The cost of maintaining the hardware, software, and personnel is high.

2. Management complexity

Database systems interface with many different technologies and have a significant impact on a company's resources and culture and it becomes hard for management.

3. Maintenance

To maximize the efficiency of the database system, you must keep your system up to date. Therefore, you must perform frequent updates and maintenance and apply the latest components.

4. Frequent upgrade/replacement cycles

DBMS requires upgrade by adding new functionality, Such new features often come bundled in new upgrade versions of the software.

Q.N.4

Describe and define the key properties of a database system. Give an organizational example of the benefits of each property.

The following are the key properties of a database system:

- Atomicity.
- Consistency.
- Isolation.
- Durability.

Atomicity:

Atomicity in database ensures that the transactions are indivisible and irreducible where transactions either commit or abort. If a part of transaction fails then entire transaction fails.

Example: In an e-commerce website if it fails to have transition between product and cart page entire buy process fails.

Consistency:

Any change in the values of a database at particular instance are consistent with changes to other values. Consistency on any transaction acts as a predicate for the data which serves as

- Precondition.
- Postcondition.
- Transformation condition.

EXAMPLE: logic applied to one table may be linked to another so changing the logic or applying changes should be first made clear and should only be applied.

Isolation:

Transaction in database ensures that the working transaction will not be changed or affected by any other transaction. In other words, modifications or updates made on one transaction is not viewed or changed by any other transaction.

Example: In an e-commerce website there should be isolation between product chosen from cart page by each individual and should not be changed once selected should be isolated.

Durability:

Durability of the databases states that “once transaction has been committed, should remain in the same status” even in the case of failures such as • Power loss. • Database crash, etc.

Example: An e-commerce website should be durable if any kind of issues arises with the crash of system or process. Should also be durable in case like if user exits in middle of process should take back to selected product for his/her next visit to system.

Q.N.5

Explain Distributed and relational database system with example.

A **distributed database** is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.

- It synchronizes the database time to time in certain interval.
- It is used in application areas where large volumes of data are used and accessed by many users at a same time.

A **relational database** is a type of a digital database based on the relation of data. It uses a structure that allows us to identify and access data in relation to another piece of data.

The data tables used in a relational database store information about the related objects. Each row holds a record with a unique identifier known as a key and each column contains the attributes of the data. Each record assigns a value to each feature, making relationships between data points easy to identify.

Q.N.6

List and explain different database models.

ANS:

The following are the different types of database models:

- Relational Model
- Hierarchical Model
- Network Model
- Object-oriented Model

Relational Model

In this type of database model Information related to a particular type is stored in rows of a table. Relationships are created by dividing data into entity and attributes. Relationship is maintained by storing a common field that may be between primary key and foreign key for simpler example

Hierarchical Model

By the term 'hierarchical' it is clear that it will have a order like parent –child relation for common understanding, with a single root Element. Hierarchy starts from the Root data, and expands like a tree, adding child nodes to the parent nodes.

Network Model

This model is conceived as a flexible way of representing data and their relationships. This model has multiple parent elements. Accessing the data is also easier and fast as data can be related from multiple nodes. This database model is many-to-many data relationships

Object-oriented Model

An object oriented model is a system in which information is represented in the form of objects as used in object-oriented programming. This database model provides features of Object Oriented Design like class, encapsulation, inheritance, polymorphism.

Q.N.7

What is database architecture? Explain different database architecture.

Using the programming languages to design a particular type of software for businesses or organizations which involves on the design, development, implementation and maintenance is called database architecture.

The architecture of a DBMS can be seen as either single tier or multi-tier. The tiers are classified as follows :

1. **1-tier architecture**
2. **2-tier architecture**
3. **3-tier architecture**
4. **n-tier architecture**

1-tier architecture:

One-tier architecture involves putting all of the required components for a software application or technology on a single server or platform (keeps all of the elements of an application, including the interface, Middleware and back-end data, in one place).

2-tier architecture:

The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server.

3-tier architecture:

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.

N-tier architecture:

N-tier architecture would involve dividing an application into three different tiers. These would be the

1. logic tier,
2. The presentation tier, and
3. The data tier.

It is the physical separation of the different parts of the application as opposed to the usually conceptual or logical separation of the elements in the model-view-controller (MVC) framework.

UNIT - 2

Q.N.1

What is E-R diagram? Describe different symbols use in the E-R diagram.

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

Following are the main components and its symbols in ER Diagrams:

- **Rectangles:** This Entity Relationship Diagram symbol represents entity types
- **Ellipses :** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes



ER-Diagram Symbols

Q.N.2

List the advantages and shortcoming of E-R diagram.

The following are the advantage of ER diagram:

Conceptually it is very simple: ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.

Better visual representation: ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.

Effective communication tool: It is an effective communication tool for database designer.

Highly integrated with relational model: ER model can be easily converted into relational model by simply converting ER model into tables.

Easy conversion to any data model: ER model can be easily converted into another data model like hierarchical data model, network data model and so on.

The following are the disadvantage of ER diagram:

Loss of information content: Some information be lost or hidden in ER model

Limited relationship representation: ER model represents limited relationship as compared to another data models like relational model etc.

No representation of data manipulation: It is difficult to show data manipulation in ER model.

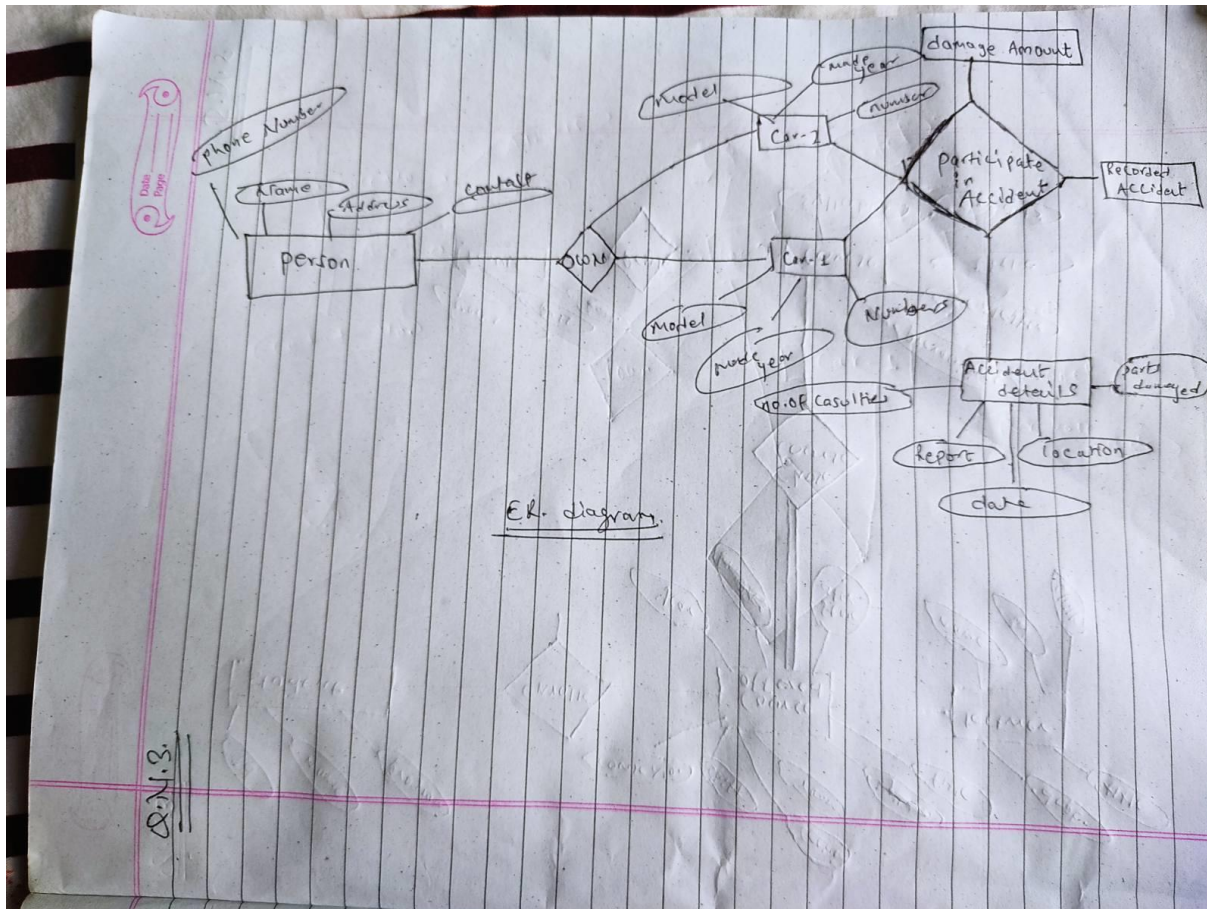
Popular for high level design: ER model is very popular for designing high level design

Q.N.3

Construct an E-R diagram for a car-insurance company whose customers

Own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

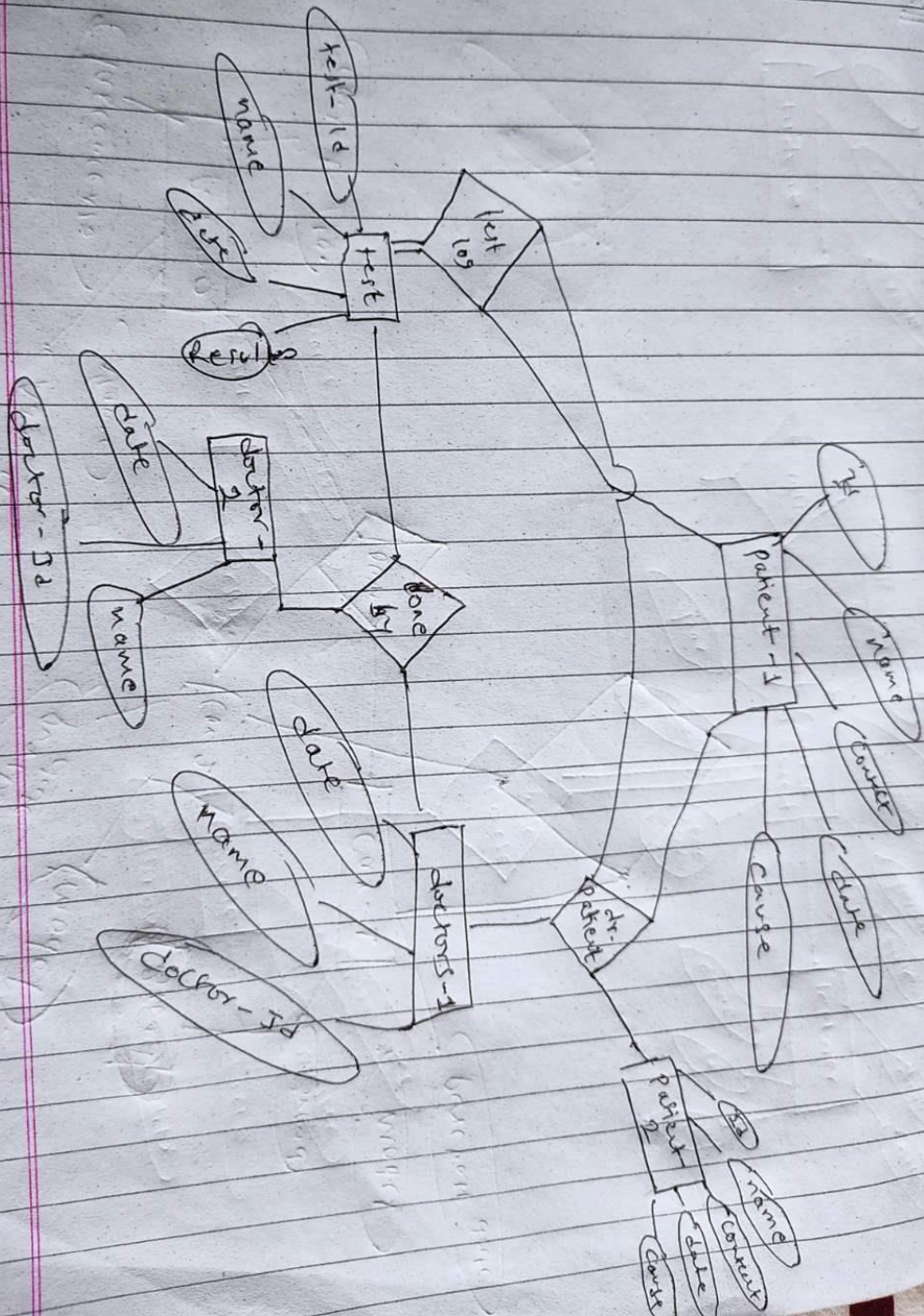
Ans:



Q.N.4

Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted

Q 11.4.

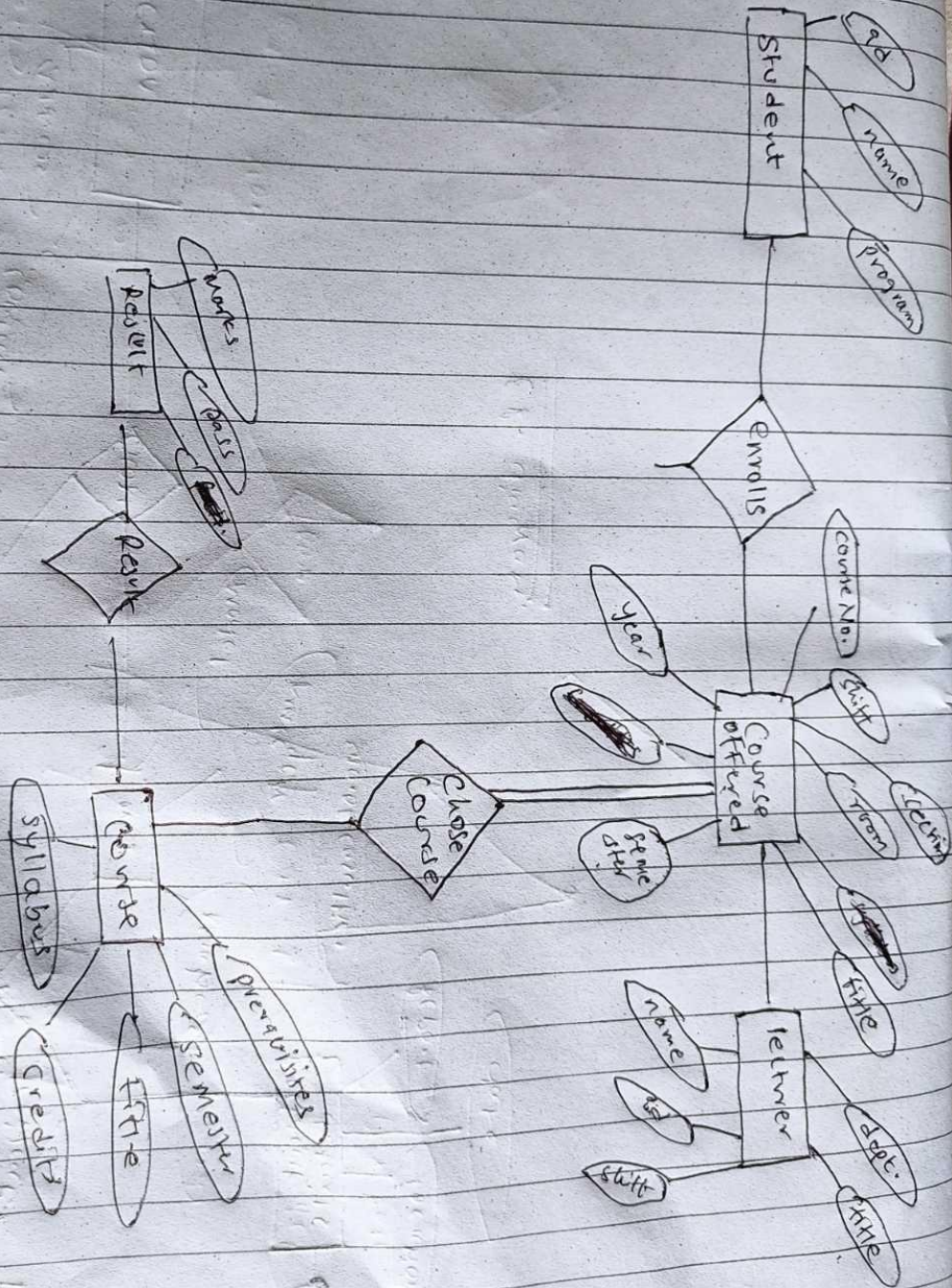


Q.N.5.

A university registrar's office maintains data about the entities

Make a suitable ER diagram.

A.N.S.



Unit-3

Q.N.1

What is Relational Algebra? Explain its usages.

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either unary or binary.

They accept relations as their input and yield relations as their output. Relational algebra is performed recursively on a relation and intermediate results are also considered relations.

The relational algebra is very important for several reasons:

1. it provides a formal foundation for relational model operations.
2. and perhaps more important, it is used as a basis for implementing and optimizing queries in the query processing and optimization modules that are integral parts of relational database management systems (RDBMSs)
3. some of its concepts are incorporated into the SQL standard query language for RDBMS.

Q.N.2

Explain different types of operation in relational algebra with example.

The fundamental operations of relational algebra are as follows –

- Select
- Project
- Union
- Set different
- Cartesian product
- Rename

Select Operation (σ)

It selects tuples that satisfy the given predicate from a relation.

Notation – $\sigma_p(r)$

Where σ stands for selection predicate and r stands for relation. p is propositional logic formula which may use connectors like and, or, and not. These terms may use relational operators like $=, \neq, \geq, <, >, \leq$.

For examples –

Example 1

$\sigma_{\text{topic} = \text{"Database"}}(\text{Tutorials})$

Output - Selects tuples from Tutorials where topic = 'Database'.

Example 2

$\sigma_{\text{topic} = \text{"Database"} \text{ and } \text{author} = \text{"mentor"}}(\text{Tutorials})$

Output - Selects tuples from Tutorials where the topic is 'Database' and 'author' is mentor.

Example 3

$\sigma_{\text{sales} > 50000}(\text{Customers})$

Output - Selects tuples from Customers where sales is greater than 50000

$\sigma_{\text{subject} = \text{"database"}}(\text{Books})$

Output – Selects tuples from books where subject is 'database'.

$\sigma_{\text{subject} = \text{"database"} \text{ and } \text{price} = 450}(\text{Books})$

Output – Selects tuples from books where subject is 'database' and 'price' is 450.

$\sigma_{\text{subject} = \text{"database"} \text{ and } \text{price} = 450 \text{ or } \text{year} > 2010}(\text{Books})$

Output – Selects tuples from books where subject is 'database' and 'price' is 450 or those books published after 2010.

Project Operation (Π)

It projects column(s) that satisfy a given predicate.

Notation – $\Pi_{A_1, A_2, A_n}(r)$

Where A_1, A_2, A_n are attribute names of relation **r**.

Duplicate rows are automatically eliminated, as relation is a set.

This operator helps you to keep specific columns from a relation and discards the other columns.

For example –

$\Pi_{\text{subject, author}}(\text{Books})$

Selects and projects columns named as subject and author from the relation Books.

EXAMPLE-2-

$\Pi_{\text{CustomerName, Status}}(\text{Customers})$

Here, the projection of CustomerName and status will be displayed.

Union Operation (\cup)

It performs binary union between two given relations and is defined as –

$r \cup s = \{ t \mid t \in r \text{ or } t \in s \}$

Notation – $r \cup s$

Where **r** and **s** are either database relations or relation result set (temporary relation).

For a union operation to be valid, the following conditions must hold –

- **r**, and **s** must have the same number of attributes.
- Attribute domains must be compatible.
- Duplicate tuples are automatically eliminated.

$\Pi_{\text{author}}(\text{Books}) \cup \Pi_{\text{author}}(\text{Articles})$

Output – Projects the names of the authors who have either written a book or an article or both.

Syntax of Union Operator (\cup) : `table_name1 \cup table_name2`

Set Difference ($-$)

The result of set difference operation is tuples, which are present in one relation but are not in the second relation.

Notation – $r - s$

Finds all the tuples that are present in r but not in s .

$\Pi_{\text{author}}(\text{Books}) - \Pi_{\text{author}}(\text{Articles})$

Output – Provides the name of authors who have written books but not articles.

$\Pi \text{ Student_Name}(\text{STUDENT}) - \Pi \text{ Student_Name}(\text{COURSE})$

To select those student names that are present in STUDENT table but not present in COURSE table.

Cartesian Product (\times)

Combines information of two different relations into one.

Notation – $r \times s$

Where r and s are relations and their output will be defined as –

$r \times s = \{ q \mid q \in r \text{ and } t \in s \}$

$\sigma_{\text{author} = \text{'peter'}}(\text{Books} \times \text{Articles})$

Output – Yields a relation, which shows all the books and articles written by peter.

Rename Operation (ρ)

The results of relational algebra are also relations but without any name. The rename operation allows us to rename the output relation. 'rename' operation is denoted with small Greek letter **rho** ρ .

$\rho_{(a/b)}R$ will rename the attribute 'b' of relation by 'a'.

Notation – $\rho_x(E)$

Where the result of expression E is saved with name of x .

Q.n.3

What is Relational Calculus? Explain different approach used here.

Relational calculus is a non-procedural query language, and instead of algebra, it uses mathematical predicate calculus. The relational calculus is not the same as that of differential and integral calculus in mathematics but takes its name from a branch of symbolic logic termed as predicate calculus. When applied to databases, it is found in two forms.

- Tuple relational calculus
- Domain relational calculus

Tuple relational calculus

The tuple relational calculus, you will have to find tuples for which a predicate is true. The calculus is dependent on the use of tuple variables. A tuple variable is a variable that 'ranges over' a named relation: i.e., a variable whose only permitted values are tuples of the relation.

Domain relational calculus

the domain relational calculus, you will also use variables, but in this case, the variables take their values from domains of attributes rather than tuples of relations.

Q.n.4

What is Natural, Theta and Equi join? Explain its types.

Join operation is essentially a cartesian product followed by a selection criterion. Join operation denoted by \bowtie . JOIN operation also allows joining variously related tuples from different relations.

Various forms of join operation are:

Inner Joins:

- Theta join
- EQUI join
- Natural join

Theta Join:

The general case of JOIN operation is called a Theta join. It is denoted by symbol θ . Theta join can use any conditions in the selection criteria.

EQUI join:

When a theta join uses only equivalence condition, it becomes a equi join. EQUI join is the most difficult operations to implement efficiently using SQL in an RDBMS and one reason why RDBMS have essential performance problems.

NATURAL JOIN (⋈)

Natural join can only be performed if there is a common attribute (column) between the relations. The name and type of the attribute must be same.

Q.N.5

Explain union with example.

The Union is a binary set operator in DBMS. It is used to combine the result set of two select queries. Thus, It combines two result sets into one. In other words, the result set obtained after union operation is the collection of the result set of both the tables. But two necessary conditions need to be fulfilled when we use the union command. These are:

1. Both SELECT statements should have an equal number of fields in the same order.
2. The data types of these fields should either be the same or compatible with each other.

The syntax for the union operation is as follows:

```
SELECT (column_names) from table1 [WHERE condition] UNION SELECT (column_names) from table2 [WHERE condition];
```

Example:

```
SELECT color_name FROM colors_a UNION ALL SELECT color_name FROM colors_b;
```

The returned values for the above query is the complete color list of table a and table b.

Unit 4

Q.N.1

What is Normalization?

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalization is to eliminate redundant (repetitive) data and ensure data is stored logically.

Q.N.2

Explain different form of normalization with example.

NF (First Normal Form) Rules

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

- Each table cell should contain a single value.
- Each record needs to be unique.

Example:

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212 9900012222

To make the table complies with 1NF we should have the data like this:

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212
102	Jon	Kanpur	9900012222

Second normal form (2NF)

A table is said to be in 2NF if both the following conditions hold:

- Table is in 1NF (First normal form)
- No non-prime attribute is dependent on the proper subset of any candidate key of table.

An attribute that is not part of any candidate key is known as non-prime attribute.

teacher_id	subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40

Candidate Keys: {teacher_id, subject}

Non prime attribute: teacher_age

The table is in 1 NF because each attribute has atomic values. However, it is not in 2NF because non prime attribute teacher_age is dependent on teacher_id alone which is a proper subset of candidate key. This violates the rule for 2NF as the rule says “**no** non-prime attribute is dependent on the proper subset of any candidate key of the table”.

To make the table complies with 2NF we can break it in two tables like this:

Teacher_details table:

Teacher_id	teacher_age
111	38
222	38
333	40

teacher_subject table:

teacher_id	subject
111	Maths
111	Physics
222	Biology
333	Physics

Third Normal form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

- Table must be in 2NF
- Transitive functional dependency of non-prime attribute on any super key should be removed.

An attribute that is not part of any candidate key is known as non-prime attribute.

Example:

emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Urrapakkam

Super keys: {emp_id}, {emp_id, emp_name}, {emp_id, emp_name, emp_zip}...so on

Candidate Keys: {emp_id}

Non-prime attributes: all attributes except emp_id are non-prime as they are not part of any candidate keys.

To make this table complies with 3NF we have to break the table into two tables to remove the transitive dependency:

employee table:

emp_id	emp_name	emp_zip
1001	John	282005
1002	Ajeet	222008
1006	Lora	282007

employee_zip table:

emp_zip	emp_state	emp_city	emp_district
282005	UP	Agra	Dayal Bagh
222008	TN	Chennai	M-City
282007	TN	Chennai	Urrapakkam

Q.N.3
ANS:
Design an appropriate tables to store the above data.

Student Details					Book Details					Borrow Details		Library Details		
Student	DOB	Gender	Class	Score	Book	Page	Score	Author	Type	Taken date	Brought date	Amount	Date	Remarks
Ram Kumar	1/1/ 1990	Male	12A	0	Hard Times	293	25	Charles Dickens	Science fiction					
Hari Das	6/1/ 1991	Male	12B	0	Editha	155	80	William Dean Howells	Art					
Gita Nepal	12/20 /1990	Female	11A	0	Show Boar, Database		25,40	Edna Ferber, Connolly	Art, Technol ogy					

To design the above table appropriately, we use different forms of normalization i.e., 1NF, 2NF and 3NF.

First, we make the table in first normal form

Student Details					Book Details					Borrow Details		Library Details		
Student	DOB	Gender	Class	Score	Book	Page	Score	Author	Type	Taken date	Brought date	Amount	Date	remarks
Ram Kumar	1/1/ 1990	Male	12A	0	Hard Times	293	25	Charles Dickens	Scienc e fiction					
Hari Das	6/1/	Male	12B	0	Editha	155	80	William	Art					

	1991							Dean Howells						
Gita Nepal	12/20 /1990	Female	11A	0	Show Boar		25	Edna Ferber	Art					
Gita Nepal	12/20 /1990	Female	11A	0	Show Boar		25	Edna Ferber	Techn ology					
Gita Nepal	12/20 /1990	Female	11A	0	Show Boar		25	Connolly	Art					
Gita Nepal	12/20 /1990	Female	11A	0	Show Boar		25	Connolly	Techn ology					
Gita Nepal	12/20 /1990	Female	11A	0	Show Boar		40	Edna Ferber	Art					
Gita	12/20	Female	11A	0	Show Boar		40	Edna	Techn					

Nepal	/1990							Ferber	ology					
Gita	12/20	Female	11A	0	Show Boar		40	Connolly	Art					
Nepal	/1990													
Gita	12/20	Female	11A	0	Show Boar		40	Connolly	Techn					
Nepal	/1990								ology					
Gita	12/20	Female	11A	0	Database		25	Edna	Art					
Nepal	/1990							Ferber						
Gita	12/20	Female	11A	0	Database		25	Edna	Techn					
Nepal	/1990							Ferber	ology					
Gita	12/20	Female	11A	0	Database		25	Connolly	Art					
Nepal	/1990													
Gita	12/20	Female	11A	0	Database		25	Connolly	Techn					
Nepal	/1990								ology					
Gita	12/20	Female	11A	0	Database		40	Edna	Art					
Nepal	/1990							Ferber						
Gita	12/20	Female	11A	0	Database		40	Edna	Techn					
Nepal	/1990							Ferber	ology					
Gita	12/20	Female	11A	0	Database		40	Connolly	Art					
Nepal	/1990													
Gita	12/20	Female	11A	0	Database		40	Connolly	Techn					
Nepal	/1990								ology					

Now we perform second normal form:

Table 1:

Student Details

Student	DOB	Gender	Class	Score
Ram Kumar	1/1/1990	Male	12A	0
Hari Das	6/1/1991	Male	12B	0
Gita Nepal	12/20/1990	Female	11A	0
Gita Nepal	12/20/1990	Female	11A	0
Gita Nepal	12/20/1990	Female	11A	0
Gita Nepal	12/20/1990	Female	11A	0

[illegible]

Table 2:

Book Details					Borrow Details		Library Details		
Book	Page	Score	Author	Type	Taken date	Brought date	Amount	Date	Remarks
Hard Times	293	25	Charles Dickens	Science fiction					
Editha	155	80	William Dean Howells	Art					
Show Boar		25	Edna Ferber	Art					
Show Boar		25	Edna Ferber	Technology					
Show Boar		25	Connolly	Art					
Show Boar		25	Connolly	Technology					
Show Boar		40	Edna Ferber	Art					
Show Boar		40	Edna Ferber	Technology					
Show Boar		40	Connolly	Art					
Show Boar		40	Connolly	Technology					
Database		25	Edna Ferber	Art					
Database		25	Edna Ferber	Technology					
Database		25	Connolly	Art					
Database		25	Connolly	Technology					
Database		40	Edna Ferber	Art					
Database		40	Edna Ferber	Technology					
Database		40	Connolly	Art					
Database		40	Connolly	Technology					

Unit-4 question answer has been copied as I was absent for this lecture and I have no note for this unit.