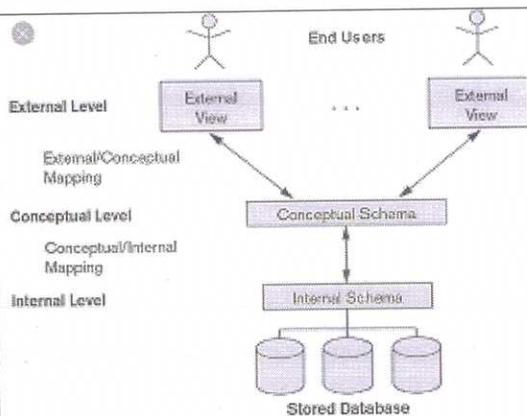


SCHEME OF VALUATION

	REVISION: 2015		COURSE CODE: 3132		
COURSE TITLE: DATABASE MANAGEMENT SYSTEMS					
QUESTION NO.	SCORING INDICATOR	SPLIT UP SCORE	SUB TOTAL	TOTAL	
PART A					
D)	1.	Casual users Naïve or parametric users Application programmers Sophisticated endusers (Database designers, Database administrators, standalone users).	0.5 0.5 0.5 0.5	2	
	2.	AVG MIN MAX COUNT SUM	0.5 0.5 0.5 0.5	2	
	3.	Every attribute has possible range of values.ie predefined value,scope.Adatatype or format is also specified for each domain.eg:phone_no - set of 10 difit phone number valid in india.	2	2	
	4.	NUMERIC DATE TIME VARCHAR TIMESTAMP INT	0.5 0.5 0.5 0.5	2	5×2=10
	5.	Mining or discovery of new information in terms of patterns or rules from vast amount of data.	2	2	
PART B					
II)	1.	1.Controlling redundancy . 2.Restricting unauthorized access. 3.Providing persistent storage for program objects. 4.Providing storage structures for efficient query processing. 5.Providingback up and recovery. 6.Providing multiple user interfaces. 7.Enforce integrity constraints. 8.Representing complex relationship among data	1 1 1 1 1 1	6	
	2.	Three schema architecture The goal of three schema architecture to separate the user application and the physical database.Inthis architechure ,schemas can be defined at the following three levels. Internal level,conceptual level and external level.			



Internal level(physical level)- It has an internal schema which describes physical storage structure of db.It describes the complete details of data storage and access path of the db.

Conceptual level(logical level)-It has a conceptual schema which describes structure of whole db for a community of users.It hides the details of physical storage structures. Concentrate on describing entities, datatypes, relationship, user operations and constraints.

External level(view level)- It includes no of external schemas or user views. Each external schema describes only the part of the db that a particular user is interested in and it hides the rest of db from users.

3. **Key** –An entity type usually has an attribute whose values are distinct for an individual entity in the entity set. Such an attribute is known as key or key attribute.

Superkey : A set of one or more attribute taken together helps in uniquely identifying each entity is called super key.

Eg: Publisher(pid ,adress,phone no,emailid) here pid is a super key and the combination of pid and name is also superkey.

Candidate key: Minimal super key that does not contain any extra attributes and it is candidate key. In the above example pid and name are candidate key.

Primary key: The candidate key which is chosen by the database designer to uniquely identifies entities is known as primary key. The attributes whose values are never or rarely changed are chosen as primary key. eg. pid in above example.

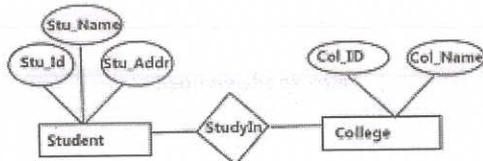
Composite key: If a primary key is formed by combination of two or more attribute is known as composite key

4. An **Entity-relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to

understand this concept.

In the following diagram we have two entities Student and College and their relationship. The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time. Student entity has attributes such as Stu_Id, Stu_Name&Stu_Addr and College entity has attributes such as Col_ID&Col_Name.



3

6

5.

Characteristics	Parallel Database	Distributed database
Definition	It is a software system where multiple processors or machines are used to execute and run queries in parallel.	It is a software system that manages multiple logically interrelated databases distributed over a computer network.
Geographical Location	The nodes are located at geographically same location.	The nodes are usually located at geographically different locations.
Execution Speed	Quicker	Slower
Overhead	Less	More

1.5

1.5

1.5

1.5

6

6.

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute.

2

If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X. Here X is a determinant set and Y is a dependent attribute. Each value of X is associated with precisely one Y value.

2

Functional dependency in a database serves as a constraint between two sets of attributes. Defining functional dependency is an important part of relational database design and contributes to aspect normalization.

2

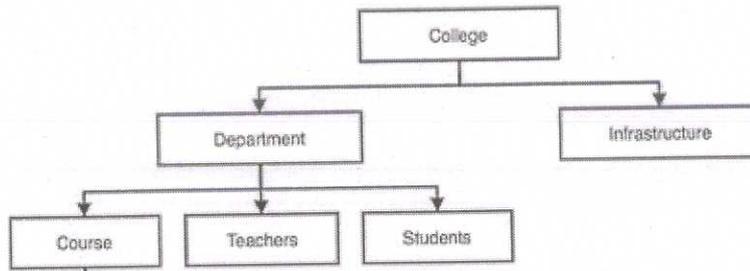
6

<p>7.</p>	<p>A stored function is a special kind stored program that returns a single value. Typically, you use stored functions to encapsulate common formulas or business rules that are reusable among SQL statements or stored programs. Different from a <u>stored procedure</u>, you can use a stored function in SQL statements wherever an expression is used. This helps improve the readability and maintainability of the procedural code. To create a stored function, you use the <code>CREATE FUNCTION</code> statement. It is used for creating a stored function and user-defined functions. A stored function is a set of SQL statements that perform some operation and return a single value.</p> <p>Syntax of stored function in mysql</p> <pre> DELIMITER \$\$ CREATE FUNCTION function_name(param1, param2,...) RETURNS datatype BEGIN -- statements END \$\$ DELIMITER ; </pre> <p>First, specify the name of the stored function that you want to create after <code>CREATE FUNCTION</code> keywords.</p> <p>Second, list all <u>parameters</u> of the stored function inside the parentheses followed by the function name. By default, all parameters are the <code>IN</code> parameters. You cannot specify <code>IN</code>, <code>OUT</code> or <code>INOUT</code> modifiers to parameters</p> <p>Third, specify the data type of the return value in the <code>RETURNS</code> statement, which can be any valid <u>MySQL data types</u>.</p> <p>Fourth, write the code in the body of the stored function in the <code>BEGIN</code> <code>END</code> block. Inside the body section, you need to specify at least one <code>RETURN</code> statement. The <code>RETURN</code> statement returns a value to the calling programs. Whenever the <code>RETURN</code> statement is reached, the execution of the stored function is terminated immediately.</p> <p>Eg; stored function for finding square of a number</p> <pre> Create function square(a int(3)) Returns int(5) begin declare s int(3); set s=a*a; return s; end// </pre> <p>to run this function – <code>select square(5)</code></p>	<p>2</p> <p>1</p> <p>1</p> <p>2</p>	<p>6</p>	<p>5×6=30</p>
-----------	---	-------------------------------------	----------	---------------

	PART C			
	UNIT I			
III)	<p>a. Railway Reservation System Database is required to keep record of ticket booking, train's departure and arrival status. Also if trains get late then people get to know it through database update.</p> <p>Library Management System There are thousands of books in the library so it is very difficult to keep record of all the books in a copy or register. So DBMS used to maintain all the information relate to book issue dates, name of the book, author and availability of the book.</p> <p>Banking We make thousands of transactions through banks daily and we can do this without going to the bank. So how banking has become so easy that by sitting at home we can send or get money through banks. That is all possible just because of DBMS that manages all the bank transactions.</p> <p>Universities and colleges Examinations are done online today and universities and colleges maintain all these records through DBMS. Student's registrations details, results, courses and grades all the information are stored in database.</p> <p>Credit card transactions For purchase of credit cards and all the other transactions are made possible only by DBMS. A credit card holder knows the importance of their information that all are secured through DBMS.</p> <p>Social Media Sites We all are on social media websites to share our views and connect with our friends. Daily millions of users signed up for these social media accounts like facebook, twitter, pinterest and Google plus. But how all the information of users are stored and how we become able to connect to other people, yes this all because DBMS.</p> <p>(Telecommunications, Finance, Online Shopping, Airline reservation, HR management, Military)</p>	1		
		1		
		1		
		1		
		1		
		1		
		1		
b.	<p>A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system.</p> <p>1. Hierarchical Model 2. Network Model 3. Entity-relationship Model 4. Relational Model</p> <p>1. Hierarchical Model</p> <p>This database model organises data into a tree-like-structure, with a single root, to which all the other data is linked. The heirarchy starts from the Root data, and expands like a tree, adding child nodes to the parent nodes.</p>		6	
		3		

In this model, a child node will only have a single parent node.

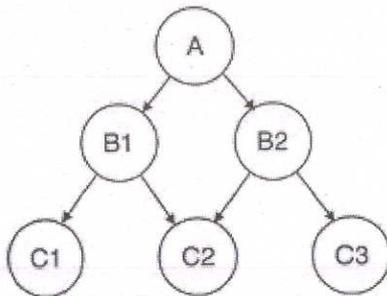
This model efficiently describes many real-world relationships like index of a book, recipes etc. for example, one department can have many courses, many professors and of-course many students.



Network Model

This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships. This was the most widely used database model, before Relational Model was introduced.



Relational Model

In this model, data is organised in two-dimensional **tables** and the relationship is maintained by storing a common field. The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table. Hence, tables are also known as **relations** in relational model.

Eg. STUDENT relation with three fields student id,name and age.

STUD_ID	NAME	AGE
1	ABC	17
2	DEF	18
3	MNO	19

3

3

IV) a.

BASIS FOR COMPARISON	DDL	DML
Basic	DDL is used to create the database schema.	DML is used to populate and manipulate database
Full Form	Data Definition Language	Data Manipulation Language
Classification	DDL is not classified further.	DML is further classified as Procedural and Non-Procedural DMLs.
Commands	CREATE, ALTER, DROP, TRUNCATE AND COMMENT and RENAME, etc.	SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, etc

9

2

1

2

b.

Centralized Database System

The centralized database system consists of a single processor together with its associated data storage devices and other peripherals. It is physically confined to a single location. Data can be accessed from the multiple sites with the use of a computer network while the database is maintained at the central site.

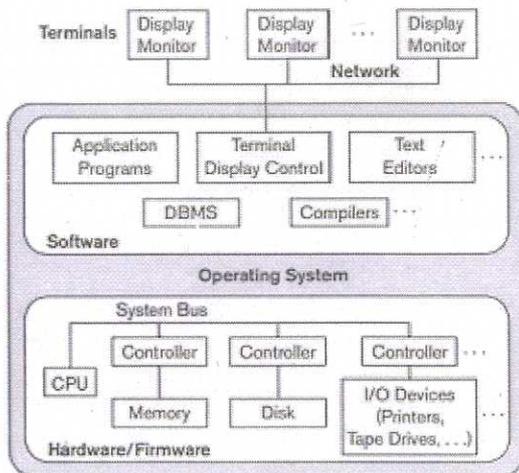
Disadvantages of Centralized Database System

- When the central site computer or database system goes down, then every one (users) is blocked from using the system until the system comes back.
- Communication costs from the terminals to the central site can be expensive.

2

7

2



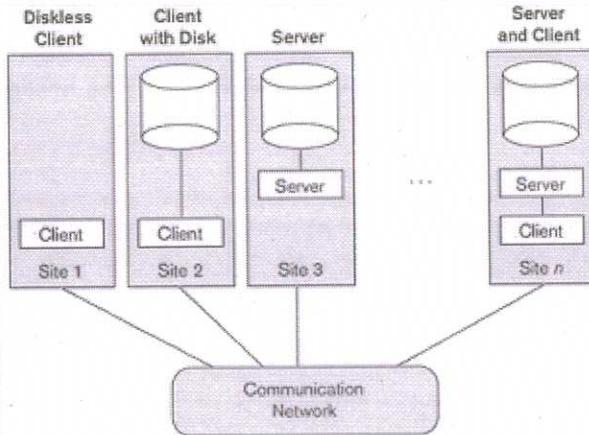
2

Architecture of centralized database(above fig)

DBMS itself was still a **centralized** DBMS in which all the DBMS functionality, application program execution, and user inter-face processing were carried out on one machine.

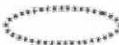
Client-Server DBMS

Client/Server architecture of database system has two logical components namely client, and server. Clients are generally personal computers or workstations whereas server is large workstations, mini range computer system or a mainframe computer system. The applications and tools of DBMS run on one or more client platforms, while the DBMS soft wares reside on the server. The server computer is caned backend and the client's computer is called front end. These server and client computers are connected into a network



UNIT II

E-R diagram is the short form of "Entity-Relationship" diagram. An e-r diagram efficiently shows the relationships between various entities stored in a database.

-  Represents Entity
-  Represents Attribute
-  Represents Relationship
-  Links Attribute(s) to entity set(s) or Entity set(s) to Relationship set(s)
-  Represents Multivalued Attributes
-  Represents Derived Attributes
-  Represents Total Participation of Entity

v) a.

2

2

8

1

1

1

1

1

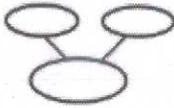
1



Represents Weak Entity



Represents Weak Relationships



Represents Composite Attributes



Represents Key Attributes / Single Value Attributes

These are the different notations and its meaning in ER diagram.

6

b.

Relational algebra is a widely used procedural query language. It collects instances of relations as input and gives occurrences of relations as output. It uses various operation to perform this action. Relational algebra operations are performed recursively on a relation. The output of these operations is a new relation, which might be formed from one or more input relations.

Unary operations: The operations operating on single relation is known as unary operations.

1) Select operations:

SELECT (σ)- The SELECT operation is used for selecting a subset of the tuples according to a given selection condition. Sigma(σ) Symbol denotes it. It is used as an expression to choose tuples which meet the selection condition. Select operation selects tuples that satisfy a given predicate.

$$\sigma_p(r)$$

σ is the predicate, r stands for relation which is the name of the table

p is propositional logic

Example 1 : $\sigma_{\text{topic} = \text{"Database"}}(\text{Tutorials})$

Output - Selects tuples from Tutorials where topic = 'Database'. $\sigma_{\text{topic} = \text{"Database"} \text{ and } \text{author} = \text{"abc"}}(\text{Tutorials})$

3

Output - Selects tuples from Tutorials where the topic is 'Database' and 'author' is abc. **2) Projection(π)** : The projection eliminates all attributes of the input relation but those mentioned in the projection list. The projection method defines a relation that contains a vertical subset of Relation. This helps to extract the values of specified attributes to eliminates duplicate values. (π)

	<p>The symbol used to choose attributes from a relation. This operation helps you to keep specific columns from a relation and discards the other columns. $\Pi_{CustomerName, Status}(Customers)$ Here project means selecting the two columns of customers relation. i.e, customername, status and its corresponding values.</p> <p>3. Rename Operation (ρ)</p> <p>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.</p> <p>Notation – $\rho_x(E)$</p> <p>Where the result of expression E is saved with name of x.</p> <p>EER Model - EER is a high-level data model that incorporates the extensions to the original ER model. Enhanced ER Model.</p> <p>It is a diagrammatic technique for displaying the following concepts Sub Class and Super Class Specialization and Generalization Union or Category Aggregation</p> <p>These concepts are used when they come in EER schema and the resulting schema diagrams called as EER Diagrams. Features of EER Model</p> <p>EER creates a design more accurate to database schemas. It reflects the data properties and constraints more precisely. It includes the concept of specialization and generalization.</p> <p>A. Sub Class and Super Class</p> <p>Sub class and Super class relationship leads the concept of Inheritance. The relationship between sub class and super class is denoted with  symbol.</p> <p>1. Super Class</p> <p>Super class is an entity type that has a relationship with one or more subtypes. An entity cannot exist in database merely by being member of any super class.</p> <p>For example: Shape super class is having sub groups as Square, Circle, Triangle.</p> <p>2. Sub Class</p> <p>Sub class is a group of entities with unique attributes. Sub class inherits properties and attributes from its super class.</p>	<p>3</p> <p>3</p> <p>1</p> <p>1</p>	<p>9</p>	
--	--	-------------------------------------	----------	--

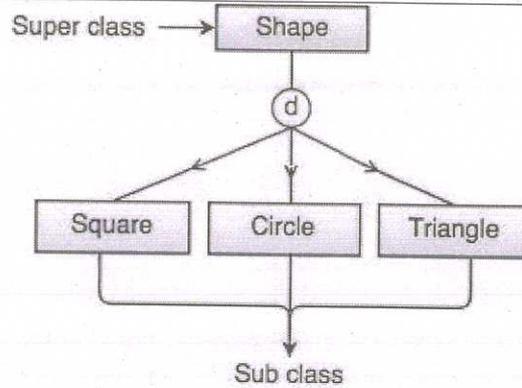


Fig. Super class/Sub class Relationship

For example: Squar

Circle, Triangle are the sub class of Shape super class.

Specialization and Generalization

1. Generalization :Generalization is the process of generalizing the entities which contain the properties of all the generalized entities.

It is a bottom approach, in which two lower level entities combine to form a higher level entity.

Generalization is the reverse process of Specialization.It defines a general entity type from a set of specialized entity type.It minimizes the difference between the entities by identifying the common features.

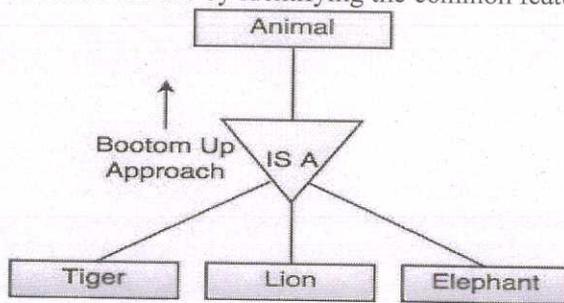


Fig. Generalization

Eg:

In the above example, Tiger, Lion, Elephant can all be generalized as Animals.

Specialization:Specialization is a process that defines a group entities which is divided into sub groups based on their characteristic.

It is a top down approach, in which one higher entity can be broken down into two lower level entity.

It maximizes the difference between the members of an entity by identifying

1

1

3

the unique characteristic or attributes of each member.
 It defines one or more sub class for the super class and also forms the superclass/subclass relationship.

For example

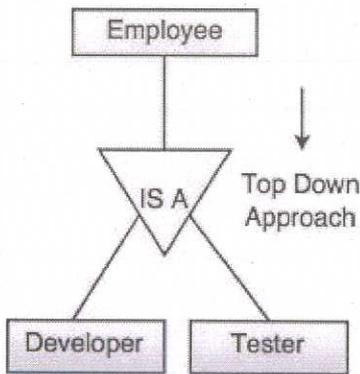


Fig. Specialization

In the above example, Employee can be specialized as Developer or Tester, based on what role they play in an Organization.

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. **Types of JOIN:** Inner Joins:
 a) Theta join
 b) EQUI join
 c) Natural join
 Outer join:

- a) Left Outer Join
- b) Right Outer Join
- c) Full Outer Join

b.

CREATE TABLE COMMANDS WITH CONSTRAINTS:

CREATE TABLE command is used to specify a new relation by giving it a name and specifying its attributes and initial constraints. The attribute are specified first and each attribute given a name, a data type to specify its domain of values and attribute constraints such NOT NULL, PRIMARY KEY, etc. The key, entity integrity and referential integrity constraints can be specified within the create table statement after attributes are declared, or the can be added later using alter command.

Eg; Create table student(roll_no int primary key, name varchar(20));

In this example student table is created with fields roll no and name. Here Rollno is unique. so primarykey constraint is added. Eg: Create table

student(roll_no int primary key, name varchar(20) not null); Here the name is added with not null constraint. It means NULL value is not permitted for that attribute.
 Eg: Create table

3

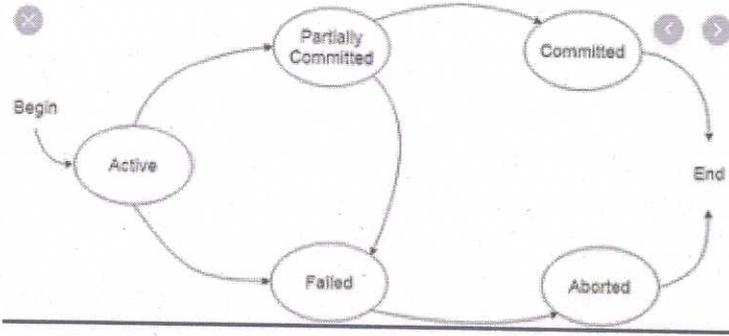
10

2

1.5

1.5

5

VII)	a. student(roll_no int primary key, name varchar(20), mark int, check mark > 50); Here the condition in the check constraint is checked before inserting or modifying. Here it checks the mark is greater than 50. Eg: create table class(class_id int primary key, class_name varchar(20), stud_id int, foreign key(stud_id) references student(roll_no)); Here the class table is linked to student table. That means it refers student table for student details.	1		
b.	States Of transaction 	2 2 2 2	9	
	Active state <ul style="list-style-type: none"> ○ The active state is the first state of every transaction. In this state, the transaction is being executed. ○ For example: Insertion or deletion or updating a record is done here. But all the records are still not saved to the database. Partially committed <ul style="list-style-type: none"> ○ In the partially committed state, a transaction executes its final operation, but the data is still not saved to the database. ○ In the total mark calculation example, a final display of the total marks step is executed in this state. Committed A transaction is said to be in a committed state if it executes all its operations successfully. In this state, all the effects are now permanently saved on the database system.	2 1		
	Failed state <ul style="list-style-type: none"> ○ If any of the checks made by the database recovery system fails, then the transaction is said to be in the failed state. 	1		

	<ul style="list-style-type: none"> o In the example of total mark calculation, if the database is not able to fire a query to fetch the marks, then the transaction will fail to execute. 	1	
	<p>Aborted</p> <ul style="list-style-type: none"> o If any of the checks fail and the transaction has reached a failed state then the database recovery system will make sure that the database is in its previous consistent state. If not then it will abort or roll back the transaction to bring the database into a consistent state. <p>Triggers :A trigger is a procedure (code segment) that is executed automatically when some specific events occur in a table/view of a database. Among its other uses, triggers are mainly used for maintaining integrity in a database. Triggers are also used for enforcing business rules, auditing changes in the database and replicating data. Most common triggers are Data Manipulation <u>Language</u> (DML) triggers that are triggered when data is manipulated. Some database systems support non-data triggers, which are triggered when Data Definition Language (DDL) events occur. Some examples are triggers that are fired when tables are created, during commit or rollback operations occur, etc. These triggers can be especially used for auditing.</p>	1	6
VIII)	<p>a.</p> <p>Cursors : A cursor is a control structure used in databases to go through the database records. It is very similar to the iterator provided by many programming languages. In addition to traversing through records in a database, cursors also facilitate data retrieval, adding and deleting records. By defining the correct way, cursors can also be used to traverse backwards. When a SQL query returns a set of rows, those are actually processed using cursors. A cursor needs to be declared and assigned a name, before it could be used. Then the cursor needs to be opened using the OPEN command. This operation would place the cursor just before the first row of the result set of records. Then the cursor has to perform the FETCH operation to actually get a row of data in to the application. Finally, the cursor has to be closed using the CLOSE operation. Closed cursors can be opened again.</p> <p>What is the difference between Triggers and Cursors?</p> <p>A trigger is a procedure (code segment) that is executed automatically when some specific events occur in a table/view of a database, while a cursor is a control structure used in databases to go through the database records. A cursor can be declared and used within a trigger. In such a situation, the declare statement would be inside the trigger. Then the scope of the cursor would be limited to that trigger. Within a trigger, if a cursor is declared on an inserted or a deleted table, such a cursor would not be accessible from a nested trigger. Once a trigger is completed, all the cursors created within the trigger will be de-allocated.Authorization includes primarily two processes:</p> <ol style="list-style-type: none"> 1. Permitting only certain users to access, process, or alter data 2. Applying varying limitations on user access or actions. The limitations placed on 	3	

<p>b.</p>	<p>4. Biological data analysis 5. Scientific Application 6. Intrusion detection Mobile database : Mobile Database is a database that is transportable, portable and physically separate or detached from the corporate database server but has the capability to communicate with those servers from remote sites allowing the sharing of various kinds of data. With mobile databases, users have access to corporate data on their laptop, PDA, or other Internet access device that is required for applications at remote sites. <u>The components of a mobile database environment include:</u></p> <ol style="list-style-type: none"> 1. Corporate database server and DBMS that deals with and stores the corporate data and provides corporate applications 2. Remote database and DBMS usually manages and stores the mobile data and provides mobile applications 3. mobile database platform that includes a laptop, PDA, or other Internet access devices 4. Two-way communication links between the corporate and mobile DBMS. <p><u>The additional functionality required for mobile DBMSs includes the capability to:</u></p> <ol style="list-style-type: none"> 1. communicate with the centralized or primary database server through modes 2. repeat those data on the centralized database server and mobile device 3. coordinate data on the centralized database server and mobile device 4. capture data from a range of sources such as the Internet 5. deal with those data on the mobile device 6. analyze those data on a mobile device 7. create customized and personalized mobile applications <p>Normalization: It is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.</p> <p>1. First Normal Form – If a relation contain composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is singled valued attribute.</p> <p>ID Name Courses -----</p>	<p>1 1 1 3 2 2 2</p>	<p>9</p>
-----------	---	--	----------

X)

a.

1	A	c1, c2
2	E	c3
3	M	C2, c3

In the above table Course is a multi valued attribute so-it is not in 1NF.

Below Table is in 1NF as there is no multi valued attribute.

ID	Name	Course
1	A	c1
1	A	c2
2	E	c3
3	M	c1
3	M	c2

2. Second Normal Form –To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has **No Partial Dependency**, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table. Partial Dependency – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

STUD_NO	COURSE_NO	COURSE_FEE
1	C1	1000
2	C2	1500
1	C4	2000
4	C3	1000
4	C1	1000
2	C5	2000

To convert the above relation to 2NF, we need to split the table into two tables such as :
 Table 1: STUD_NO, COURSE_NO
 Table 2: COURSE_NO, COURSE_FEE
 3. Third Normal Form –A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form. A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency $X \rightarrow Y$

1. X is a super key.
2. Y is a prime attribute (each element of Y is part of some candidate key).

4. Boyce-Codd Normal Form (BCNF) –A relation R is in BCNF if R is in Third Normal Form and for every FD, LHS is super key. A relation is in

6

1

3

2

<p>b.</p>	<p>BCNF if in every non-trivial functional dependency $X \rightarrow Y$, X is a super key. Datawarehouse</p> <p>1. data warehouse is a database, which is kept separate from the organization's operational database.. 2. There is no frequent updating done in a data warehouse.. 3. It possesses consolidated historical data, which helps the organization to analyze its business</p> <p>3. It possesses consolidated historical data, which helps the organization to analyze its business.</p> <p>4. A data warehouse helps executives to organize, understand, and use their data to take strategic decisions.</p> <p>5. Data warehouse systems help in the integration of diversity of application systems.</p> <p>data warehouse system helps in consolidated historical data analysis. A data warehouses provides us generalized and consolidated data in multidimensional view. Along with generalized and consolidated view of data, a data warehouses also provides us Online Analytical Processing (OLAP) tools. These tools help us in interactive and effective analysis of data in a multidimensional space. This analysis results in data generalization and data mining. Data mining functions such as association, clustering, classification, prediction can be integrated with OLAP operations to enhance the interactive mining of knowledge at multiple level of abstraction.</p> <p>Data Warehouse Features</p> <p>1. Subject Oriented , 2. Integrated , 3. Time Variant , 4. Non-volatile</p>	<p>2</p> <p>1</p>	<p>9</p>	<p>15×4=60</p>
-----------	--	-------------------	----------	----------------