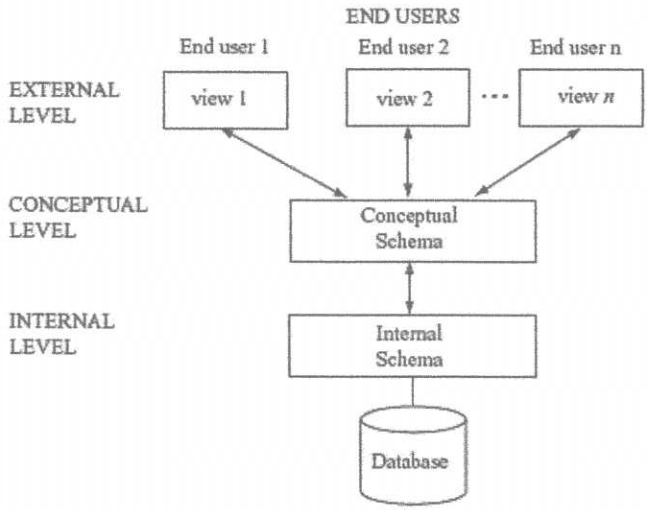
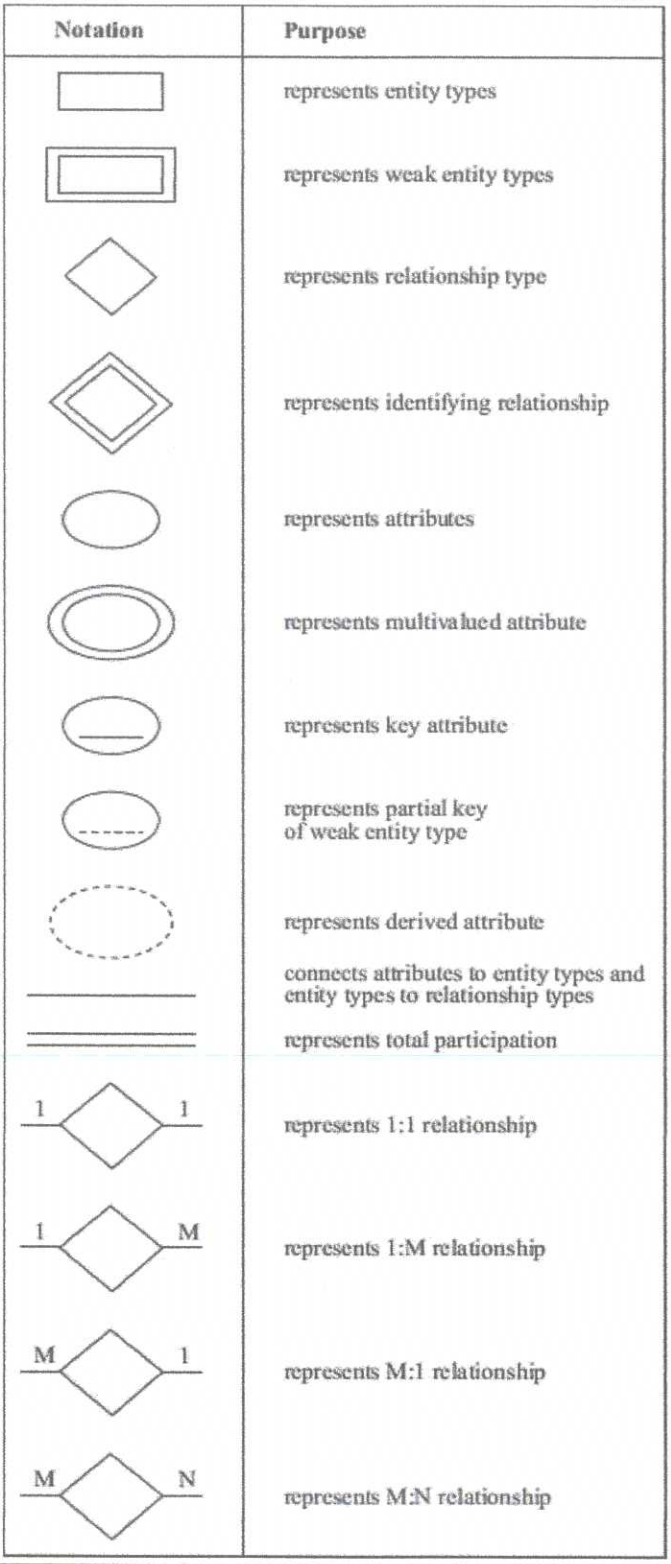
SCHEME OF VALUATION
(Scoring Indicators)

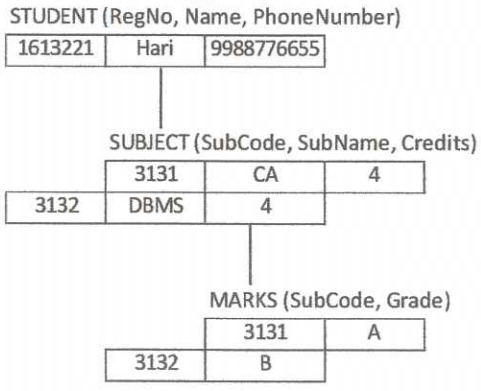
Revision: 2015 Course Title: Database Management System	Course Code: 3132

Qst No	Scoring Indicators	Split up Score	Sub Total	Total
I (1)	A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.	2	2	2
I (2)	Naive users, sophisticated users, specialized users, system analyst, application programmers, database administrator (any 4)	4*0.25	2	2
I (3)	The attributes that are indivisible are known as simple attributes Composite attributes can be divided into smaller subparts	1 1	2	2
I (4)	A trigger is a type of stored procedure that is executed automatically when some database related events like, insert, update, delete, etc., occur. • Implementing and maintaining complex integrity constraints. • Recording the changes for auditing purposes. • Automatically passing signal to other programs that action needs to be taken whenever specific changes are made to a relation.	2*1	2	2
I (5)	A functional dependency(FD), denoted by X → Y, between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuples that can form a relation state r of R.	2	2	2
II (1)	1. Controlled data redundancy: Eliminates replicating the data item in different files, and ensures consistency and saves the storage space. 2. Enforcing data integrity: integrity constraints are identified by database designer during database design. enforced by the DBMS 3. Data sharing: The data stored in the database can be shared among multiple users or application programs. 4. Ease of application development: The application programmer needs to develop the application programs according to the users' needs. The other issues like concurrent access, security, data integrity, etc., are handled by the DBMS itself. 5. Data security: The DBMS ensures that the only means of access to the database is through an authorized channel. Data security checks can be carried out whenever access is attempted to sensitive data. Different checks can be established for each type of access (addition, modification, deletion, etc.) to each piece of information in the database. 6. Multiple user interfaces: DBMS provides different types of interfaces such as query languages, application program interfaces, and graphical user interfaces (GUI) that include forms-style and menu-driven interfaces. 7. Backup and recovery: DBMS provides backup and recovery subsystem that is responsible for recovery from hardware and software failures. e.g., if the failure occurs in between the transaction, the DBMS recovery subsystem either reverts back the database to the state which existed prior to the start of the transaction	Any 4 List: 2 Expln: 4 * 1	6	6

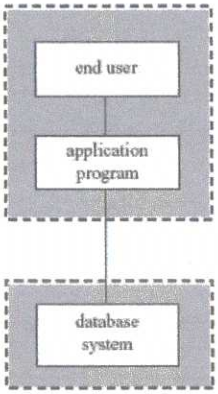
	or resumes the transaction from the point it was interrupted so that its complete effect can be recorded in the database. 8. Program-data independence: The independence between the programs and the data is known as program-data independence (or simply data independence). It is an important characteristic of DBMS as it allows changing the structure of the database without making any changes in the application programs that are using the database. 9. Data abstraction: The property of DBMS that allows program-data independence is known as data abstraction. Data abstraction allows the database system to provide an abstract view of the data to its users without giving the physical storage and implementation details. 10. Supports multiple views of the data: A database can be accessed by many users and each of them may have a different perspective or view of the data. A database system provides a facility to define different views of the data for different users. A view is a subset of the database that contains virtual data derived from the database files but it does not exist in physical form.			
II (2)	 1. Internal level: It is the lowest level of data abstraction that deals with the physical representation of the database on the computer and also known as physical level. It describes how the data is physically stored and organized on the storage medium. At this level, various aspects are considered to achieve optimal runtime performance and storage space utilization. These aspects include storage space allocation techniques for data and indexes, access paths such as indexes, data compression and encryption techniques, and record placement. 2. Conceptual level: This level of abstraction deals with the logical structure of the entire database and also known as logical level. It describes what data is stored in the database, the relationships among the data and complete view of the user's requirements without any concern for the physical implementation. It hides the complexity of physical storage structures. The conceptual view is the overall view of the database and it includes all the information that is going to be represented in the database. 3. External level: It is the highest level of abstraction that deals with the user's view of the database and thus, is also known as view level. Most of the users and application programs do not require the entire data stored in the database.	Diagram: 2 List: 1 Expln: 3 * 1	6	6

	The external level describes a part of the database for a particular group of users. It permits users to access data in a way that is customized according to their needs, so that the same data can be seen by different users in different ways, at the same time.			
II (3)		any 6 6 * 1	6	6
II (4)	Specialization is the process of defining a set of subclasses of an entity type; this entity type is called the superclass of the specialization. The set of subclasses			

	that forms a specialization is defined on the basis of some distinguishing characteristic of the entities in the superclass. For example, the set of subclasses {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of the superclass EMPLOYEE that distinguishes among employee entities based on the job type of each employee entity. Another specialization of the EMPLOYEE entity type may yield the set of subclasses {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE} based on the method of pay.	Expln: 4 Eg: 2	6	6
II (5)	Data Types: Data type identifies the type of data to be stored in an attribute of a relation and also specifies associated operations for handling the data. Data type defined for an attribute associates a set of properties with that attribute. These properties cause attributes of different data types to have different set of operations that can be performed on these attributes. • NUMERIC(p, s): used to represent data as floating-point number like 17.312, 27.1204, etc. The number can have p significant digits (including sign) and s number of the p digits can be present on the right of decimal point. For example, data type specified as NUMERIC(5, 2) indicates that value of an attribute can be of form 332.32, where as number of the forms 32.332 or 0.332 are not allowed. • INT or INTEGER: used to represent data as a number without a decimal point. • SMALLINT: used to represent data as a number without a decimal point. It is a subset of the INTEGER so the default size is usually smaller than INT. • CHAR(n) or CHARACTER(n): used to represent data as fixed-length string of characters of size n. In case of fixed-length strings, a shorter string is padded with blank characters to the right. For example, if the value ABC is to be stored for an attribute with data type CHAR(8), the string is padded with five blanks to the right. • VARCHAR(n) or CHARACTER VARYING: used to represent data as variable length string of characters of maximum size n. In case of variable length string, a shorter string is not padded with blank characters. • DATE and TIME: used to represent data as date or time. The DATE data type has three components, namely, year, month, and day in the form YYYY-MM-DD. The TIME data type also have three components, namely, hours, minutes, and seconds in the form HH:MM:SS. • BOOLEAN: used to represent the third value unknown, in addition to true and false values, because of the presence of null values in SQL. • TIMESTAMP: used to represent data consisting of both date and time. The TIMESTAMP data type has six components, year, month, day, hour, minute, and second in the form YYYY-MM-DDHH:MM:SS[.sF], where F is the fractional part of the second value.	6*1	6	6
II (6)	Data warehouse is defined as a subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions. Subject-Oriented: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject. Data is organized according to a subject instead of application. Integrated: A data warehouse integrates data from multiple, heterogeneous data sources such as databases, files, documents, web etc. For example, source A and source B may have different ways of identifying a product, but in a data	3*2	6	6

	warehouse, there will be only a single way of identifying a product. Time-Variant: Historical data is kept in a data warehouse to provide a historical perspective. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer. Non-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered. The data is not overwritten or deleted once it is entered but is only loaded, refreshed and accessed for queries.			
II (7)	A mobile database is a database that can be connected to by a mobile computing device over a wireless mobile network. It is transportable, portable and physically separate or detached from the central database server but has the capability to communicate with those servers from remote sites allowing the sharing of various kinds of data. Mobile databases: • Physically separate from the central database server • Resided on mobile devices • Capable of communicating with a central database server or other mobile clients from remote sites • Handle local queries without connectivity Client-server model is the dominant model for existing mobile databases. Mobile DBMSs should satisfy the following requirements: Small memory footprint, Flash-optimized storage system, Data synchronization, Security, Low power consumption, Self-management, Embeddable in applications.	6	6	6
III (a)	Hierarchical Data Model - This data model organizes the data in a tree-like structure, in which each child node (also known as dependents) can have only one parent node. The database based on the hierarchical data model comprises a set of records connected to one another through links. The link is an association between two or more records. The top of the tree structure consists of a single node that does not have any parent and is called the root node. The root may have any number of dependents; each of these dependents may have any number of lower level dependents. Each child node can have only one parent node and a parent node can have any number of (many) child nodes. The collection of same type of records is known as a record type. One complete record of each record type represents a node.  	Expln: 3*2 Eg: 1*2	8	8

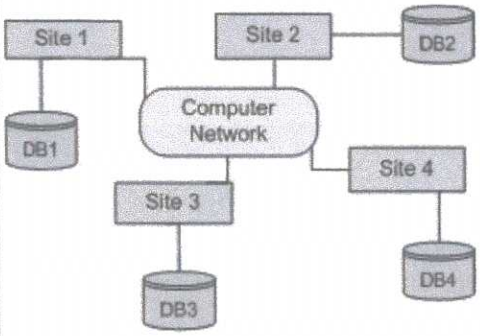
<table>
<tr>
<td></td>
<td>Relational Data Model - In the relational data model, all data is maintained in the form of tables, known as relations consisting of rows and columns. Each row (record) represents an entity and a column (field) represents an attribute of the entity. The relationship between the two tables is implemented through a common attribute in the tables and not by physical links or pointers.
STUDENT
<table><tr><th>RegNo</th><th>Name</th><th>PhoneNo</th></tr><tr><td>1613221</td><td>Hari</td><td>9988776655</td></tr></table>
SUBJECT
<table><tr><th>RegNo</th><th>SubCode</th><th>SubName</th><th>Credits</th></tr><tr><td>1613221</td><td>3131</td><td>CA</td><td>4</td></tr><tr><td>1613221</td><td>3132</td><td>DBMS</td><td>4</td></tr></table>
MARKS
<table><tr><th>RegNo</th><th>SubCode</th><th>Grade</th></tr><tr><td>1613221</td><td>3131</td><td>A</td></tr><tr><td>1613221</td><td>3132</td><td>B</td></tr></table></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III (b)</td>
<td>Data Definition Language (DDL) is used to define the conceptual and internal schemas for the database. The DBMS comprises DDL compiler that identifies and stores the schema description in the DBMS catalog. The DDL statements are also used to specify the integrity rules (constraints) in order to maintain the integrity of the database. Example: Create, Alter, Truncate, Drop
Data Manipulation Language (DML) is used to manipulate the database. The DBMS provides data manipulation language (DML) that enables users to retrieve and manipulate the data. The statement which is used to retrieve the information is called a query. The part of the DML used to retrieve the information is called a query language. Example: Select, Insert, Update, Delete
Data Control Language (DCL) is used to control access to data stored in a database. DCL commands are used to enforce database security in a multiple user database environment. Two types of DCL commands are GRANT and REVOKE.</td>
<td>List: 1
Expln: 3*2</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>IV (a)</td>
<td>In centralized database systems, the database system, application programs, and user-interface all are executed on a single system and dummy terminals are connected to it. The processing power of single system is utilized and dummy terminals are used only to display the information.
In client/server architecture, the processing power of the computer system at the user's end is utilized by processing the user-interface on that system. A client is a computer system that sends request to the server connected to the network, and a server is a computer system that receives the request, processes it, and returns the requested information back to the client. The end users work on client computer system and database system runs on the server.</td>
<td>Expln: 3*2
Diagram: 2</td>
<td>8</td>
<td>8</td>
</tr>
</table>

	 CLIENT SERVER			
IV (b)	1. Airlines and railways: Airlines and railways use online databases for reservation, and for displaying the schedule information. 2. Banking: Banks use databases for customer inquiry, accounts, loans, and other transactions. 3. Education: Schools and colleges use databases for course registration, result, and other information. 4. Telecommunications: Telecommunication departments use databases to store information about the communication network, telephone numbers, record of calls, for generating monthly bills, etc. 5. Credit card transactions: Databases are used for keeping track of purchases on credit cards in order to generate monthly statements. 6. E-commerce: Integration of heterogeneous information sources (for example, catalogs) for business activity such as online shopping, booking of holiday package, consulting a doctor, etc. 7. Health care information systems and electronic patient record: Databases are used for maintaining the patient health care details. 8. Digital libraries and digital publishing: Databases are used for management and delivery of large bodies of textual and multimedia data. 9. Finance: Databases are used for storing information such as sales, purchases of stocks and bonds or data useful for online trading. 10. Sales: Databases are used to store product, customer and transaction details. 11. Human resources: Organizations use databases for storing information about their employees, salaries, benefits, taxes, and for generating salary checks.	any 7 7*1	7	7
V(a)	Unary Operations: The operations operating on a single relation are known as unary operations. SELECT Operation: The SELECT operation is used to choose a subset of the tuples from a relation that satisfies a selection condition. The SELECT operation is denoted by $\sigma_{<\text{selection condition}>}(R)$ where the symbol σ (sigma) is used to denote the SELECT operator and the selection condition is a Boolean expression (condition) specified on the attributes of relation R. The comparison operators are normally used to specify the conditions $\{=, <, \leq, >, \geq, \neq\}$, and the clauses can be connected by the logical operators AND, OR and NOT to form a general selection condition. For example, to select the EMPLOYEE tuples whose department is 4, $\sigma_{\text{DEPTNO=4}}(\text{EMPLOYEE})$ PROJECT Operation: The project operation is used to select some required attributes from a relation while discarding the other attributes. The general form of the PROJECT operation is	Expln: 3*2 Eg: 2*1	8	8

	$\pi_{<attribute list>}(R)$ where π (pi) is the symbol used to represent the PROJECT operation, and <attribute list> is the desired sublist of attributes from the attributes of relation R. The result of the PROJECT operation has only the attributes specified in <attribute list> in the same order as they appear in the list. For example, to list each employee's number, name and salary, the PROJECT operation is as follows: $\pi_{Lempno, empname, salary}$(EMPLOYEE)			
V(b)	Natural Join: The joins having equality condition as their join condition result in two attributes in the resulting relation having exactly the same value. If one of the two identical attributes is removed from the result of equijoin, it is known as natural join. The natural join of two relations R1 and R2 is obtained by applying a project operation to the equijoin of these two relations in a sequence given here. 1. Find the equijoin of two relations R1 and R2. 2. For each attribute A that is common to both relations R1 and R2, project operation is applied to remove the column R1.A or R2.A. Therefore, if there are m attributes common in both the relations, then m duplicate columns are removed from the resultant relation of equijoin. For example, the natural join operation is, $\pi_{EMPNO, EMPNAME, PHONENO, DEPTNO, DEPTNAME}$(EMPLOYEE $\bowtie_{EMPLOYEE.DEPTNO = DEPARTMENT.DEPTNO}$ DEPARTMENT)	Expln: 5 Eg: 2	7	7
VI(a)	A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it references the primary key of another table, thereby establishing a link between them. Example, Dept No is primary key in Dept table and as foreign key in Employee table. A minimal superkey that does not contain any extra attributes in it is called a candidate key. A candidate key contains a minimized set of attributes that can be used to uniquely identify a single entity instance. Example, Employee Number and Name are the candidate keys. The candidate key, which is chosen by the database designer to uniquely identify entities, is known as the primary key. For example, the attribute Employee Number can be chosen as the primary key.	3 3 2	8	8
VI(b)	1. Mapping Entity • Create table for each entity • Entity's attributes should become fields of tables with their data types • Declare primary key 2. Mapping Relationship • Create table for a relationship. • Add the primary keys of all participating Entities as fields of table • 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. 3. Mapping Weak Entity Sets	2 3	7	7

	• 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.	2		
VII(a)	The Structured Query Language (SQL) is a language which can be used for retrieval and management of data stored in relational database. It can be used for defining the structure of data, modifying data in the database and specifying the security constraints. High Speed, Well Defined Standards Exist, No Coding Required, High Scalability and Flexibility, Robust Transactional Support, Comprehensive Application Development support.	2 3*2	8	8
VII(b)	Stored procedures can be compiled and executed with different parameters and results. They can accept input parameters and pass values to output parameters. A stored procedure can be created as shown here. CREATE PROCEDURE <name> (<parameters1, …, parametersn >) <local_declarations> <body of procedure>; Parameters and local declarations are optional and if declared must be of valid SQL data types. Parameters declared must have one of the three modes, namely, IN, OUT, or INOUT specified for it. Parameters specified with IN mode are arguments passed to the stored procedure and act like a constant, whereas, OUT parameters act like an un-initialized variables and they cannot appear on the right hand side of = symbol. Parameters with INOUT mode have combined properties of both IN and OUT mode. They contain values to be passed to the stored procedure and also can be assigned values to be returned from the stored procedure. Functions are required when value is required to be returned to the calling program since procedures cannot return a value. The function can be created as CREATE FUNCTION <name> (<parameters1, …, parametersn >) RETURNS <return_type> <local_declarations> <body of function> ;	4+3	7	7
VIII(a)	The constraints are required to maintain the integrity of the data, which ensures that the data in database is consistent, correct, and valid. These constraints can be specified within the definition of a relation. These include key, integrity, and referential constraints along with the restrictions on attribute domains and null values. 1. PRIMARY KEY Constraint - This constraint ensures that attribute declared as primary key cannot have null value and no two tuples can have same value for primary key attribute. 2. UNIQUE Constraint - The UNIQUE constraint ensures that the set of attributes have unique values, that is, no two tuples can have same value in the specified attributes. 3. CHECK Constraint - This constraint ascertains that the value inserted in an attribute must satisfy a given expression. 4. NOT NULL Constraint - The NOT NULL constraint is used to specify that an attribute will not accept null values. 5. FOREIGN KEY Constraint - This constraint ensures that the foreign key	list: 2 expln: 3*2 (any 3)	8	8

	value in the referencing relation must exist in the primary key attribute of the referenced relation, that is, foreign key references the primary key attribute of referenced relation.			
VIII(b)	It is used to retrieve the subset of tuples or attributes from one or more relations. SELECT <attribute1>, <attribute2>;, …, <attributen > FROM <table_name> WHERE <condition> ORDER BY attribute; Required: FROM Optional: WHERE, ORDER BY, GROUP BY	Expln: 2 Eg: 3 1 1	7	7
IX(a)	Normalization theory provides the guidelines to assess the quality of a database in designing process. The normalization is the application of set of simple rules called normal forms to the relation schema. To qualify as a good relational database design, it should have following features. 1. Minimum redundancy: A relational database should be designed in such a way that it should have minimum redundancy. 2. Fewer Null Values in Tuples: the values of all the attributes of a tuple are not know as the value is not known while inserting the data. Null values do not provide complete solution as they cannot be inserted in the primary key attributes and attributes for which not null constraint is specified. 3. Generation of Invalid data: Design relation schemas so that they can be joined with equality conditions on attributes that are appropriately related (primary key, foreign key) pairs in a way that guarantees that no spurious tuples are generated.	2 3*2	8	8
IX(b)	An object database (also object-oriented database management system, OODBMS) is a database management system in which information is represented in the form of objects as used in object-oriented programming. Object-oriented database management systems combines database capabilities with object-oriented programming language capabilities. OODBMSs allow object-oriented programmers to develop the product, store them as objects, and replicate or modify existing objects to make new objects within the OODBMS. Features: Complexity - OODBMS has the ability to represent the complex internal structure (of object) with multilevel complexity. Inheritance - Creating a new object from an existing object in such a way that new object inherits all characteristics of an existing object. Encapsulation - It is a data hiding concept in OOPL which binds the data and functions together which can manipulate data and not visible to outside world. Persistence - OODBMS allows to create persistent object (Object remains in memory even after execution). This feature can automatically solve the problem of recovery and concurrency. The data stored in OO database is accessed directly from the OOPL using the native type system of the language. Whenever a persistent object is created, the system returns a persistent object identifier which points to an object in the database and remains valid even after the termination of program. Versioning – versioning allows maintaining multiple versions of an object and OODBMS provide capabilities for dealing with all versions of the object.	List: 1 Expln: 3*2	7	7
X(a)	In decomposition, a relation is placed with a collection of smaller relations with			

	specific relationship between them. The decomposition of a relation schema R is defined as its replacement by a set if relation schemas such that each relation schema contains a subset of the attributes of R. Decomposition reduce the redundancy and avoids data inconsistency. During decomposition, it must be ensured that each attribute in R must appear in at least on relation schema Ri, so that no attributes are lost. Example:	4 4	8	8
X(b)	A distributed database (DDB) is a collection of multiple, logically interrelated databases distributed over a computer network. The computers are geographically dispersed and are connected with one another through various communication media. A distributed database management system (D–DBMS) is the software that manages the DDB and provides an access mechanism that makes this distribution transparent to the users.  Distributed Database System Features: It is used to create, retrieve, update and delete distributed databases. It synchronizes the database periodically and provides access mechanisms by the virtue of which the distribution becomes transparent to the users. It ensures that the data modified at any site is universally updated. It is used in application areas where large volumes of data are processed and accessed by numerous users simultaneously. It is designed for heterogeneous database platforms. It maintains confidentiality and data integrity of the databases. Homogenous DDBMS Heterogeneous DDBMS or multi-database system	Expln: 3 Diagram: 1 Features: 3	7	7