

Scoring Indicators

Database Management System

Qn: No		Split score	Sub Total	Total
I. 1	A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database	2	2	2
I. 2	Entities are represented by means of their properties, called attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.	1+1	2	2
I. 3	SELECT */attributes FROM tablename WHERE condition;	2 X 1	2	2
I. 4	Views are virtual relations whose values are derived from existing relations.	2	2	2
I. 5	Functional dependencies (FDs) $X \rightarrow Y$, Y Functionally depends on X or the value of Y can be determined by X) FD, $X \rightarrow Y$ holds if whenever two tuples have the same value for X, they must have the same value for Y.	1+1	2	2
II.1	<p>1. Advantages of DBMS</p> <ul style="list-style-type: none"> • Controlling Redundancy • Restricting Unauthorized Access • Providing Persistent Storage for Program Objects • Providing Storage Structures and Search Techniques for Efficient Query Processing • Providing Backup and Recovery • Providing Multiple User Interfaces • Representing Complex Relationships among Data • Enforcing Integrity Constraints • Permitting Inferencing and Actions Using Rules 	Any 4X1.5	6	6
II.2	<p>The diagram illustrates the three-level architecture of a DBMS. It consists of three levels: External Level, Conceptual Level, and Internal Level. The External Level is connected to the Conceptual Level by two arrows labeled 'External / Conceptual Mapping'. The Conceptual Level is connected to the Internal Level by a double-headed arrow labeled 'Conceptual / Internal Mapping'. The Internal Level is connected to a cylinder representing the 'Database'.</p>	1+1 3+3	2 6	2 6

Scoring Indicators

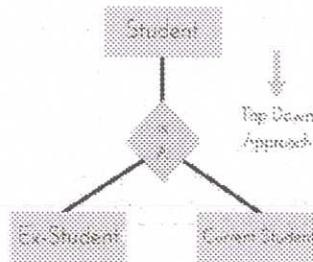
Database Management System

	<p>The three schema architecture contains three-levels. It breaks the database down into three different categories</p> <p>Internal Level- The internal level has an internal schema which describes the physical storage structure of the database-The internal schema is also known as a physical schema- It is used to define that how the data will be stored in a block-The physical level is used to describe complex low-level data structures in detail.</p> <p>Conceptual Level- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level- The conceptual schema describes the structure of the whole database-</p> <p>External Level- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database- Also known as view schema- The view schema describes the end user interaction with database systems.</p>			
<p>II. 3</p>	<p>2. Generalization- Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity- It's more like Superclass and Subclass system, but the only difference is the approach, which is bottom-up. Hence, entities are combined to form a more generalised entity, in other words, sub-classes are combined to form a super-class.</p> <div style="text-align: center;"> <pre> graph BT Account[Account] Saving[Saving] Current[Current] Saving --- Plus{+} Current --- Plus Plus --- Account style Plus fill:none,stroke:none </pre> </div>	<p style="text-align: center;">3+3</p>	<p style="text-align: center;">6</p>	<p style="text-align: center;">6</p>
<p>II. 3</p>				

Scoring Indicators

Database Management System

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, a higher level entity may not have any lower-level entity sets, it's possible.



II. 4

● **SELECT Operation**

SELECT operation is used to select a *subset* of the tuples from a relation that satisfy a **selection condition**.

- Tuples those satisfying the condition are selected while others are discarded.
- In general, the select operation is denoted by $\sigma_{\langle \text{selection condition} \rangle}(\mathbf{R})$
Horizontal partitioning.

Example:

1. To select the EMPLOYEE tuples whose department number is four

$$\sigma_{DNO = 4}(\mathbf{EMPLOYEE})$$

3+3

6

6

● **PROJECT Operation**

- This operation selects certain *columns* from the table and discards the other columns.
- The PROJECT creates a vertical partitioning – one with the needed columns (attributes)

The general form of the project operation is

$$\pi_{\langle \text{attribute list} \rangle}(\mathbf{R})$$

The project operation *removes any duplicate tuples*.

Example: To list each employee's first and last name and salary, the following is used:

$$\pi_{LNAME, FNAME, SALARY}(\mathbf{EMPLOYEE})$$

Scoring Indicators

Database Management System

II. 5	<p>Aggregate functions are Count()—to get number of tuples or number of not null attribute values</p> <p>Sum() Max() Min() Avg()</p>	2 + 4 x 1	6 6 6	6 6 6
II. 6	<p>Normalization</p> <ul style="list-style-type: none"> • The process of modifying the relational schema to satisfy any of the normal forms is called normalization. • Normalization is used for removing the redundancies from the relations. Removal of redundancy improves storage efficiency and data integrity. • Normalization is used to make database structure flexible . • Normalization can be used to avoid anomalies. <ul style="list-style-type: none"> – Anomalies problems arised from relations due to insertion, deletion and updation) 	3 x 2=6	6	6
II. 7	<ul style="list-style-type: none"> ▪ A Dataware house is a repository(collection) of data gathered from multiple sources stored under a unified schemea at a single site. The data once collected can be stored for a long period allowing access to historical data. ▪ Dataware house is a <ul style="list-style-type: none"> ▪ Subject oriented → organized around major subject ▪ Integrated → collected from multiple sources ▪ Time-variant → stored for a long period(5 to 10 years) ▪ Non volatile collection of data. --> storage is permanent. 	2 + 4	6 6 6	6 6 6
III a	<p>By data, we mean known facts that can be recorded and that have implicit meaning. Eg: name of a student</p> <p>A database is a collection of related data. Eg: college database: (details of students in a college, details of departents, staff details etc)</p> <p>The data in the database at a particular moment in time is called a database state or snapshot. It is also called the current set of occurrences or instances in the database.</p> <p>The description of a database is called the database schema, which is specified during database design and is not expected to change frequently. Eg: schema Student(rollno, name ,branch) Dept(deptid, dname) Staff(staffid, name dob, designation)</p>	2+2+2+3	9	15

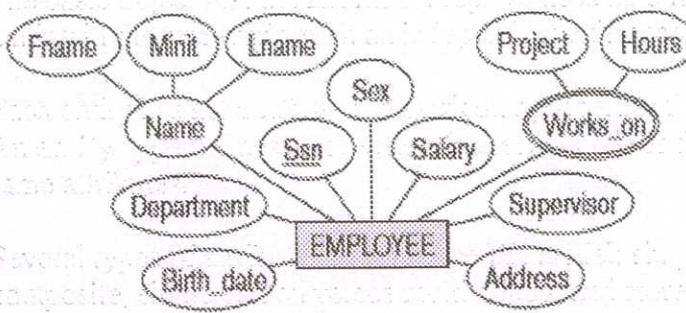
Scoring Indicators

Database Management System

<p>III b</p>	<ul style="list-style-type: none"> • Data Definition Language- CREATE,ALTER, DROP ,TRUNCATE AND RENAME • Data Manipulation Language- SELECT,UPDATE,INSERT AND DELETE • Data Control Language - GRANT AND REVOKE • Transaction Control Language- COMMIT, ROLLBACK AND SAVEPOINT 	<p>Any 3 x 2</p>	<p>6</p>	
<p>IVa</p>	<p style="text-align: center;">Explanation of each block</p>	<p>5+4</p> <p>Any 3 x 2</p>	<p>9</p> <p>6</p>	<p>15</p>
<p>IV b</p>	<p>Hierarchical Network</p> <p>The hierarchical model represents data as hierarchical tree structures. Each hierarchy represents a number of related records. There is no standard language for the hierarchical model.</p> <p>The network model represents data as record types and also represents a limited type of 1:N relationship, called a set type. A 1:N, or one-to-many, relationship relates one instance of a record to many record instances using some pointer linking mechanism in these models</p>	<p>Any one 3+3</p>	<p>6</p>	<p>15</p>

Scoring Indicators

Database Management System

<p>V a</p>	<p>ER model The ER model describes data as entities, relationships, and attributes. The basic object that the ER model represents is an entity, which is a thing in the real world with an independent existence.</p> <p>Each entity has attributes—the particular properties that describe it. An entity type defines a collection (or set) of entities that have the same attributes.</p> <p>Several types of attributes occur in the ER model: simple versus composite, single valued versus multivalued, and stored versus derived.</p> <p>Diagrammatic notation associated with the ER model, known as ER diagrams.</p> <p>Cardinality Ratios for Binary Relationships. The cardinality ratio for a binary relationship specifies the maximum number of relationship instances that an entity can participate in. ER diagram for an entity EMPLOYEE is shown below.</p> 	<p>6+3</p>		
<p>V b</p>	<p>A superkey is a group of single or multiple keys which identifies rows in a table. A Super key may have additional attributes that are not needed for unique identification.</p> <p>A column or group of columns in a table which helps us to uniquely identifies every row in that table is called a primary key. This DBMS can't be a duplicate. The same value can't appear more than once in the table.</p> <p>A foreign key is a column which is added to create a relationship with another table. Foreign keys help us to maintain data integrity and also allow navigation between two different instances of an entity. Every relationship in the model needs to be supported by a foreign key.</p>	<p>3 x 2</p>	<p>6</p>	<p>15</p>

Scoring Indicators

Database Management System

VI a	<p>Natural Join (\bowtie)- Natural join does not use any comparison operator. It does not concatenate the way a Cartesian product does. We can perform a Natural Join only if there is at least one common attribute that exists between two relations. In addition, the attributes must have the same name and domain.</p> <p>Outer Joins- Theta Join, Equijoin, and Natural Join are called inner joins</p> <p>Left Outer Join($R \ltimes S$)- All the tuples from the Left relation, R, are included in the resulting relation. If there are tuples in R without any matching tuple in the Right relation S, then the S-attributes of the resulting relation are made NULL.</p> <p>Right Outer Join: ($R \ltimes S$)- All the tuples from the Right relation, S, are included in the resulting relation. If there are tuples in S without any matching tuple in R, then the R-attributes of resulting relation are made NULL.</p> <p>Full Outer Join: ($R \ltimes S$)- All the tuples from both participating relations are included in the resulting relation. If there are no matching tuples for both relations, their respective unmatched attributes are made NULL.</p>	3 +	3 x 2	9	
VI b	<p>Explain the mapping of E-R model to relational model.</p> <p>Step 1: Mapping of Regular Entity Types. For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.</p> <p>Step 2: Mapping of Weak Entity Types. For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of W as attributes of R.</p> <p>Step 3: Mapping of Binary 1:1 Relationship Types. For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. There are three possible approaches: (1) the foreign key approach, (2) the merged relationship approach, and (3) the cross reference or relationship relation approach.</p>	+	6	6	15

Scoring Indicators

Database Management System

VII a	<p>CREATE The CREATE TABLE command is used to specify a new relation by giving it a name and specifying its attributes and initial constraints. Syntax: CREATE TABLE TABLE NAME(ATTRIBUTE1 DATATYPE,.....); Eg: create table employee(empid int primary key, ename varchar(10), dob date, phoneno bigint);</p> <p>INSERT INSERT command is used to insert rows to the created table. Syntax: INSERT INTO TABLENAME VALUES(VAL1, VAL2,...); Eg: insert into employee values(1, 'ram', '1990/02/10', 9876262711);</p> <p>UPDATE U PDATE command is used to change the vlues stored in a table. Syntax: update tablename set attribute name = value, Where condition. Eg: update employee set ename='abc' , phoneno= 985643872 where eid=1;</p> <p>DELETE DELETE command is used to delete the specified record Syntax: DELETE FROM TABLENAME WHERE condition; Eg: delete from employee where eid=1;</p>	3+2+2+2	9	
VII b	<ul style="list-style-type: none"> ○ The select query which is specified in the where clause of another query is called nested query or inner query. The query in which nested query is enclosed is called outer query. ○ Outer query can be update or delete. ○ In inner query if more than one attribute value returns then 'in' operator should be used instead of = operator. <p>Syntax: select * from table name where attributename=(select attribute name from tablename);</p> <p>Eg: select *from employee where dept=(select deptno from department where dname ='ct');</p>	3+3	9 6	15

Scoring Indicators

Database Management System

VIII a	<p><u>BASIC STEPS TO CONNECT A DATABASE USING JAVA / JDBC</u></p> <p>1. Importing required packages</p> <p>Before you can create a java jdbc connection to the database, you must first import the java.sql package.</p> <p>Import java.sql.*;</p> <p>2. Loading the JDBC drivers- Dynamically loads a driver class using Class.forName.</p> <p>3. Establishing the connection to a database.</p> <p>The getConnection() method defined in JDBC DriverManager class is used to establish a connection to a database. It uses a jdbc URL (Path to db name), username and password.</p> <p>4. Creating a Statement</p> <p>To execute SQL statements, create a Statement object from the connection object by using the createStatement() method. Statement st = con.createStatement();</p> <p>5. Executing a SQL query and returning a ResultSet object if any.</p> <p>executeQuery() for QUERY(select) statements. executeUpdate() for INSERT, UPDATE, DELETE</p> <p>6. Processing the ResultSet object</p> <p>7. Closing the ResultSet and Statement objects</p> <p>8. Closing the connection</p>	9	9	15
-----------	---	---	---	----

Scoring Indicators

Database Management System

VIII b	<p>✓ A trigger is a type of stored procedure that is executed automatically when some database related event like insert, update and delete occur.</p> <p>Syntax of creating trigger Create trigger triggername BEFORE /AFTER INSERT/UPDATE/DELETE ON tablename</p> <p>For each row Begin Stmt1 Stmt2 End</p> <p>Use of trigger</p> <ul style="list-style-type: none"> ➤ Trigger is used to automatically execute a procedure when some database event like insert or update occurs, (Eg: prevention of withdrawal from a bank account that doesn't have a minimum balance) ➤ No need to invoke a trigger, it will be automatically executed. 	<p>1</p> <p>+</p> <p>3</p> <p>+</p> <p>2</p>	6	
IX a	<p>DECOMPOSITION</p> <ul style="list-style-type: none"> • One relation can be decomposed to several sub relations • The process of dividing one relation into several sub relations is called decomposition. • DECOMPOSITION category <ol style="list-style-type: none"> 1. LOSSLESS → Original relation can be recovered by combining the sub relations 2. LOSSY(LOSSY-JOIN) → decomposition that is not lossless • Let R be a relation with a given set of FDs. The decomposition $D = \{R_1, R_2, \dots, R_n\}$ in R is lossless if $\pi_{R_1}(R) \bowtie \pi_{R_2}(R) \bowtie \dots \bowtie \pi_{R_n}(R) = R$ <ul style="list-style-type: none"> • Projection of attributes of relation R1 natural join Projection of attributes of relation R2 upto Rn = R. 	<p>2</p> <p>+</p> <p>+</p> <p>Category (4)</p> <p>+</p> <p>3</p> <p>9</p>	15	

Scoring Indicators

Database Management System

Eg: BOOK

Bookno	Price	pagecount
B1	25	800
B2	22	200
B3	25	810

Decomposition 1 (BOOK table into two B1 and B2)

B1

Book no	Price
B1	25
B2	22
B3	25

B2

Book no	Page count
B1	800
B2	200
B3	810

Decomposition 1 is LOSSLESS
(B1 NATURAL JOIN B2 GIVES BOOK)

Decomposition 2 (BOOK into B3 and B4)

B3

price	Page count
25	800
22	200
25	810

B4

Book no	Price
B1	25
B2	22
B3	25

B3 natural join B4 is not the original relation BOOK. Hence LOSSY,

Bookno	Price	pagecount
B1	25	800
B1	25	810
B2	22	200
B3	25	800
B3	25	810

Spurious tuple

Scoring Indicators

Database Management System

IX b	<p style="text-align: center;">OODB CONCEPTS / FEATURES / CHARACTERISTICS</p> <p>1. Object and class:</p> <p>OBJECT → real world entity. An object contains both the data structure and the set of operations that can be used to manipulate it.</p> <p>CLASS: group of all objects which share the same set of attributes and methods.</p> <p>2. Abstraction: the process of identifying the essential aspects of an entity and ignoring the unimportant properties.</p> <p style="margin-left: 20px;">A). Encapsulation: Combine the object's value and operations into a single unit.</p> <p style="margin-left: 20px;">B. Information hiding:</p> <ul style="list-style-type: none"> • Separates the external aspects of an object from its internal details, which are hidden from the outside world. <p>3. Operator polymorphism:</p> <p style="margin-left: 20px;">a. Same method with different types of objects.</p> <p style="margin-left: 20px;">b. Also called operator overloading</p> <p>4. Inheritance</p> <ul style="list-style-type: none"> ➤ Inheritance allows one class (subclass) to be defined as a special case of a more general class (superclass). ➤ The process of forming a superclass is referred to as generalization. The process of forming a subclass is referred to as specialization. ➤ <u>By default</u>, a subclass inherits all the properties of its superclass(es) and, additionally, defines its own unique properties. ➤ Inheritance: egs: <ol style="list-style-type: none"> 1. Single inheritance: 2. Multiple inheritance: etc 			
		4 x 1.5	6	

Database Management System

X a

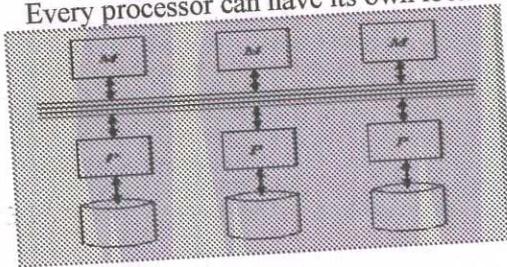
PARALLEL DBMS ARCHITECTURE

Three possible architectures for passing information-

1. Shared-memory
2. Shared-disk
3. Shared-nothing

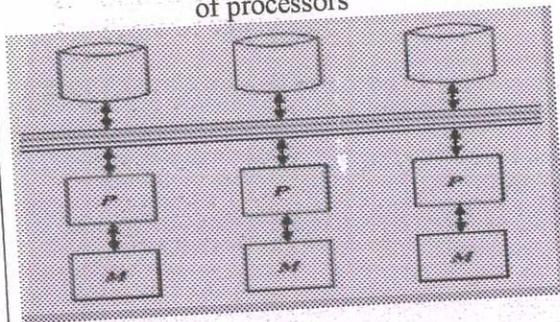
SHARED-MEMORY ARCHITECTURE

- Every processor has its own disk
- Single memory address-space for all processors
- Reading or writing to far memory can be slightly more expensive
- Every processor can have its own local memory and cache as well



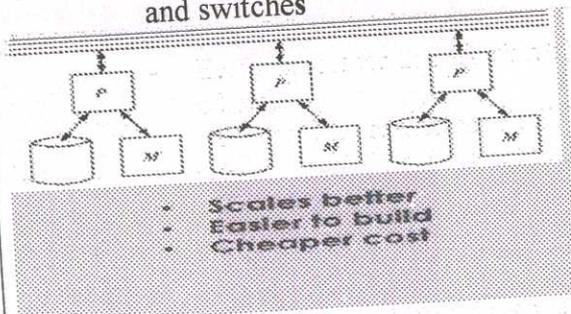
SHARED-DISK ARCHITECTURE

- Every processor has its own memory (not accessible by others)
- All machines can access all disks in the system
- Number of disks does not necessarily match the number of processors



SHARED-NOTHING ARCHITECTURE

- Most common architecture nowadays
- Every machine has its own memory and disk
- Communication is done through high speed network and switches



- Scales better
- Easier to build
- Cheaper cost

3+3+3

9

15

S

