

SCHEME OF VALUATION

(Scoring Indicators)

Revision: 2015

Course Title: Data Base Management Systems.

Course Code: 3132

| Qn:No: | Scoring Indicator | Split up score | Sub total | Total |
|---------------|---|----------------|-----------|-------|
| PART A | | | | |
| I(1) | The overall design or description of the database is called database schema. | 2 | 2 | 2 |
| I(2) | A set of one or more attributes, when taken together ,helps in uniquely identifying each entity is called a super key. A minimal super key that does not contain any extra attributes in it is called a candidate key. | 1+1 | 2 | 2 |
| I(3) | Triggers are used to maintain data integrity. Triggers are designed for recording information, which can be used for auditing purposes. | 2 | 2 | 2 |
| I(4) | Int, smallint, char(n), varchar(n), date, boolean | Any 4-2 | 2 | 2 |
| I(5) | There are two goals of the normalization process: <ul style="list-style-type: none"> eliminating redundant data (for example, storing the same data in more than one table) Ensuring data dependencies make sense (only storing related data in a table). | 2 | 2 | 2 |

PART B

Data independence is the ability to change the schema at one level of the database system without having to change the schema at other levels. It is of two types.

1) Logical data independence-It is the ability to change the conceptual schema without affecting the external schema or application programs.

2) Physical data independence- It is the ability to change the internal schema without affecting the conceptual or external schema .

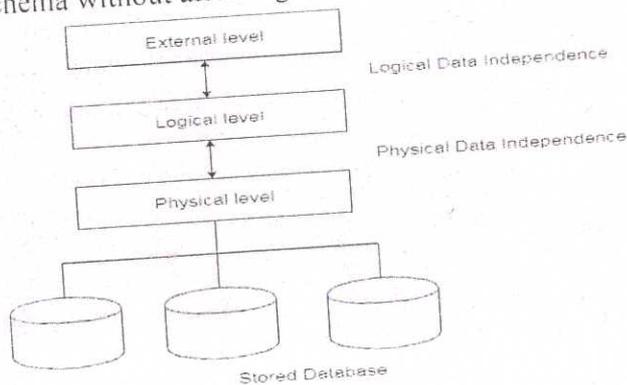


Fig-3
Expln-3

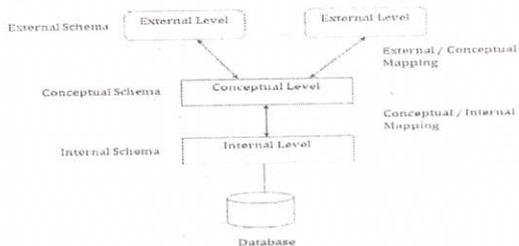
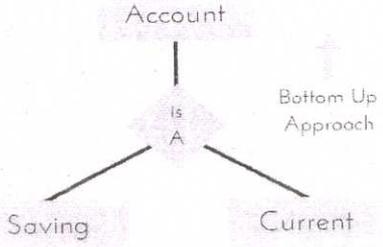
3+3

6

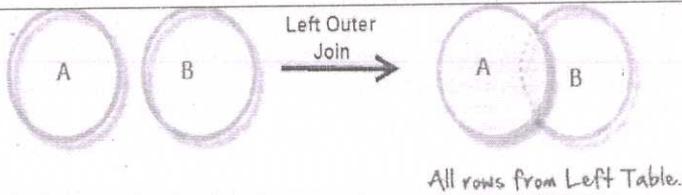
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|--------------------------|------------------------|--------------------------------------|------------------------------------|--|--|-----------------------------------|-----------------------------------|-------------------|-------|-----------|--|------------------|--|-----------------------|--|-------------------|--|-------------------|--|---------------|--|--|--|-----------------------|------------------|---|---|
| II(2) | <ul style="list-style-type: none"> • Airlines and railways • Banking • Education • Telecommunications • E-commerce • Finance • Human resources | Any 6 | 6*1=6 | 6 | | | | | | | | | | | | | | | | | | | | | | | | | |
| II(3) | <table border="1"> <tr> <td>Strong entity set</td> <td>Weak entity set</td> </tr> <tr> <td>The strong entity has a primary key.</td> <td>The weak entity has a partial key.</td> </tr> <tr> <td>The strong entity is independent of any other entity in a schema</td> <td>Weak entity depends on strong entity for its existence</td> </tr> <tr> <td>Represented by a single rectangle</td> <td>Represented by a double rectangle</td> </tr> </table> | Strong entity set | Weak entity set | The strong entity has a primary key. | The weak entity has a partial key. | The strong entity is independent of any other entity in a schema | Weak entity depends on strong entity for its existence | Represented by a single rectangle | Represented by a double rectangle | Explanatio-4 eg-2 | 4+2=6 | 6 | | | | | | | | | | | | | | | | | |
| Strong entity set | Weak entity set | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| The strong entity has a primary key. | The weak entity has a partial key. | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| The strong entity is independent of any other entity in a schema | Weak entity depends on strong entity for its existence | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Represented by a single rectangle | Represented by a double rectangle | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| II(4) | <h3>E-R Diagram Symbols</h3> <table border="0"> <tr> <td></td> <td>Entity Set</td> <td></td> <td>Relationship</td> </tr> <tr> <td></td> <td>Weak Entity Set</td> <td></td> <td>One-to-one link</td> </tr> <tr> <td></td> <td>Attribute</td> <td></td> <td>Many-to-one link</td> </tr> <tr> <td></td> <td>Multivalued Attribute</td> <td></td> <td>Many-to-many link</td> </tr> <tr> <td></td> <td>Derived Attribute</td> <td></td> <td>Key Attribute</td> </tr> <tr> <td></td> <td></td> <td></td> <td>Attribute can be NULL</td> </tr> </table> | | | Entity Set | | Relationship | | Weak Entity Set | | One-to-one link | | Attribute | | Many-to-one link | | Multivalued Attribute | | Many-to-many link | | Derived Attribute | | Key Attribute | | | | Attribute can be NULL | Any 10 symbols-6 | 6 | 6 |
| | Entity Set | | Relationship | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Weak Entity Set | | One-to-one link | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute | | Many-to-one link | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Multivalued Attribute | | Many-to-many link | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Derived Attribute | | Key Attribute | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | Attribute can be NULL | | | | | | | | | | | | | | | | | | | | | | | | | | |
| II(5) | <p>a) Atomicity – This property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none</p> <p>b) Consistency – The database must remain in a consistent state after any transaction. No transaction should have any adverse effect on the data residing in the database.</p> <p>c) Durability – The database should be durable enough to hold all its latest updates even if the system fails or restarts.</p> <p>d) Isolation – In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system.</p> | | 6 | 6 | 6 | | | | | | | | | | | | | | | | | | | | | | | | |
| II(6) | <ul style="list-style-type: none"> • It is a database that can be connected to a mobile computing device over a mobile over a mobile network. • It is a database that is portable and is physically separate from the | | 3+3=6 | 6 | 6 | | | | | | | | | | | | | | | | | | | | | | | | |

- cooperate database server.
- It 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

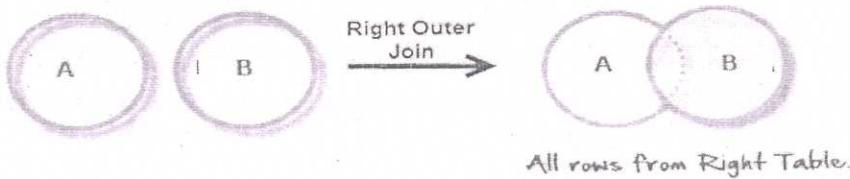
| II(7) | <table border="1"> <thead> <tr> <th>Homogeneous distributed dbms</th> <th>Heterogeneous distributed dbms</th> </tr> </thead> <tbody> <tr> <td>The sites use very similar software.</td> <td>Different sites use dissimilar schemas and software.</td> </tr> <tr> <td>Each site is aware of all other sites and cooperates with other sites to process user requests</td> <td>A site may not be aware of other sites and so there is limited co-operation in processing user requests.</td> </tr> <tr> <td>The database is accessed through a single interface as if it is a single database.</td> <td>Query processing is complex due to dissimilar schemas. Transaction processing is complex due to dissimilar software.</td> </tr> </tbody> </table> | Homogeneous distributed dbms | Heterogeneous distributed dbms | The sites use very similar software. | Different sites use dissimilar schemas and software. | Each site is aware of all other sites and cooperates with other sites to process user requests | A site may not be aware of other sites and so there is limited co-operation in processing user requests. | The database is accessed through a single interface as if it is a single database. | Query processing is complex due to dissimilar schemas. Transaction processing is complex due to dissimilar software. | 3*2=6 | 6 | 6 |
|--|--|------------------------------|--------------------------------|--------------------------------------|--|--|--|--|---|-------|---|---|
| Homogeneous distributed dbms | Heterogeneous distributed dbms | | | | | | | | | | | |
| The sites use very similar software. | Different sites use dissimilar schemas and software. | | | | | | | | | | | |
| Each site is aware of all other sites and cooperates with other sites to process user requests | A site may not be aware of other sites and so there is limited co-operation in processing user requests. | | | | | | | | | | | |
| The database is accessed through a single interface as if it is a single database. | Query processing is complex due to dissimilar schemas. Transaction processing is complex due to dissimilar software. | | | | | | | | | | | |
| III(a) | <p><u>Hierarchical data model</u> → It is the oldest type model, developed by IBM in 1968. This model organizes the data in a tree like structure, in which each child can have only one parent node. The top of the tree structure consists of a single node that does not have any parent and is called root node.</p> <p><u>Network data model</u> → The data is organized in the form of graphs. All the nodes are linked to each other without any hierarchy. A parent node can have many child nodes and a child can also have many parent nodes. Thus the network model permits the modelling of many to many relationships in data.</p> <p><u>Relational data model</u> → There are no physical links. 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.</p> | 3*3=9 | 9 | 9 | | | | | | | | |
| III(b) | <ol style="list-style-type: none"> 1. Data Definition languages (DDL) - It is used for defining the database schema. The DDL statements are also used to specify the constraints. eg: <i>create, alter</i> 2. Data Manipulation languages (DML) - It is used to retrieve and manipulate the data. eg: <i>select, update</i> 3. Data Control languages (DCL) - The DCL is used to create user roles, grant permissions and control access to database by securing it. eg: <i>grant, revoke</i> | 3*2=6 | 6 | 6 | | | | | | | | |

| | | | | |
|----------------|--|--|-------------------|----------|
| <p>IV(a)</p> |  <p>The three levels are</p> <ol style="list-style-type: none"> 1. Internal level→It is the lowest level of data abstraction that deals with the physical representation of the database on the computer and thus also known as physical level. It describes how the data is physically stored and organized on the storage medium. 2. Conceptual level→this level of abstraction deals with the logical structure of the entire database and thus is also known as logical level. It describes what data is stored in the database, the relationship among the data. It hides the complexity of the physical storage structures. 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. The external level describes a part of the database for a particular group of users. | <p>Fig-3 Expln - 6(3*2 =6)</p> | <p>3+6= 9</p> | <p>9</p> |
| <p>IV ((b)</p> | <p>Duties of Database Administrator:-</p> <ol style="list-style-type: none"> Schema definition and modifications. New software installation. Security enforcement and administration. Data analysis. Preliminary database design. Physical organization modification. | <p>Any 6 duties- 6</p> | <p>6</p> | <p>6</p> |
| <p>V(a)</p> | <p>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. Entities are combined to form a more generalized entity, in other words, sub-classes are combined to form a super-class.</p>  <p>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.</p> | <p>4+4=8</p> | <p>4+4= 8</p> | <p>8</p> |

| | | | | |
|-------|---|-------------|-------------|---|
| | | | | |
| V(b) | <p>Attribute → Each column in a Table. Attributes are the properties which define a relation</p> <p>Domain → A domain is defined as the set of all unique values permitted for an attribute.</p> <p>Tuples → A single row of a table, which contains a single record for that relation is called a tuple.</p> <p>Degree → The total number of attributes which in the relation is called the degree of the relation.</p> | $2+2+2+1=7$ | $2+2+2+1=7$ | 7 |
| VI(a) | <p>In relational algebra <i>select</i>, <i>project</i> and <i>rename</i> are unary operations.</p> <ol style="list-style-type: none"> Select operation → The select operation retrieves all those tuples from a relation that satisfy a specific condition. It is denoted by sigma (σ). It is used as an expression to choose tuples which meet the selection condition. Select operation selects tuples that satisfy a given predicate. Eg: $\sigma_{\text{category} = \text{"Novel"}}(\text{BOOK})$ Project operation → It is used to select some required attributes from a relation while discarding the other attributes. It can be viewed as the vertical subset of a relation. Eg: $\Pi_{\text{ISBN, title}}(\text{BOOK})$ Rename operation → It is used to provide name to the relation obtained after applying any relational algebra operation. 'rename' operation is denoted with small Greek letter rho ρ. Eg: $\rho_{(R1, \sigma_{\text{category} = \text{"novel"}}(\text{BOOK}))}$ | $3*3=9$ | 9 | 9 |
| VI(b) | <p>Outer join → In an outer join, along with tuples that satisfy the matching criteria, we also include some or all tuples that do not match the criteria.</p> <p>The three types of outer joins are</p> <ul style="list-style-type: none"> Left Outer Join ($A \bowtie B$) → In the left outer join, operation allows keeping all tuple in the left relation. However, if there is no matching tuple is found in right relation, then the attributes of right relation in the join result are filled with null values. | $3*2=6$ | 6 | 6 |



- **Right Outer Join:** $(A \bowtie B) \rightarrow$ In the right outer join, operation allows keeping all tuple in the right relation. However, if there is no matching tuple is found in the left relation, then the attributes of the left relation in the join result are filled with null values.



- **Full Outer Join:** $(A \bowtie B) \rightarrow$ In a full outer join, all tuples from both relations are included in the result, irrespective of the matching condition.

VII(a)

- **INSERT :**The insert statement is used to add new row to a table.
`INSERT INTO table_name VALUES (value1, value2, value3,...);`
- **UPDATE :**The update statement is used to change values that are already in a table.
`UPDATE table name SET column1=value1,column2=value2,... WHERE condition;`
- **DELETE :**The delete statement deletes row(s) from a table.
`DELETE FROM table name WHERE condition;`
- **SELECT :**The **SELECT** statement is used to form queries for extracting information out of the database.
`SELECT * FROM table_name;`

4*2=8

8

8

VII(b)

Views in SQL

- Views in SQL are considered as a virtual table. A view also contains

Create-
3
Use 4

3+4=
7

7

| | | | | |
|---------|--|-------|---|---|
| | <p>rows and columns.</p> <ul style="list-style-type: none"> ○ To create the view, we can select the fields from one or more tables present in the database. ○ A view can either have specific rows based on certain condition or all the rows of a table. <p>CREATE VIEW Syntax</p> <p>CREATE VIEW <i>view_name</i> AS SELECT <i>column1, column2, ...</i> FROM <i>table_name</i> WHERE <i>condition</i>;</p> <p>To see the created view:</p> <p>SELECT * FROM <i>view_name</i></p> <p>Deleting View</p> <p>A view can be deleted using the Drop View statement</p> <p>DROP VIEW <i>view_name</i>;</p> | | | |
| VIII(a) | <p>AVG – calculates the average of a set of values. Eg: SELECT AVG(Price) FROM BOOK;</p> <p>COUNT – counts rows in a specified table or view. Eg: SELECT COUNT(*) FROM BOOK;</p> <p>MIN – gets the minimum value in a set of values. Eg: SELECT MIN (Price) FROM BOOK WHERE category='novel';</p> <p>MAX – gets the maximum value in a set of values. Eg: SELECT MAX (Price) FROM BOOK WHERE category='novel';</p> <p>SUM – calculates the sum of values Eg: SELECT SUM (Price) FROM BOOK WHERE category='novel';</p> | 8 | 8 | 8 |
| VIII(b) | <p>SQL Grant Command</p> <ul style="list-style-type: none"> • SQL Grant command is used to provide access or privileges on the database objects to the users. • The syntax for the GRANT command is: • <code>“GRANT privilege_name ON object_name “”TO {user_name PUBLIC role_name} [with GRANT option];</code> | 4+3=7 | 7 | 7 |

| | <ul style="list-style-type: none"> • privilege_name: is the access right or privilege granted to the user. • object_name: is the name of the database object like table, view etc., • user_name: is the name of the user to whom an access right is being granted. Public is used to grant rights to all the users. With Grant option: allows users to grant access rights to other users. <p>SQL Revoke Command</p> <ul style="list-style-type: none"> • The revoke command removes user access rights or privileges to the database objects. • The syntax for the REVOKE command is: • REVOKE privilege_name ON object_name FROM {User_name PUBLIC Role_name} • For Example: • (a) GRANT SELECT ON employee TO user1 • (b) REVOKE SELECT ON employee FROM user1 | | | | | | | | | | | | | |
|--|---|-------------|------------------|--|---|---|--|---|--|--|--|-------|---|---|
| IX(a) | <table border="1"> <thead> <tr> <th data-bbox="183 705 630 750">Data mining</th> <th data-bbox="630 705 1077 750">Data warehousing</th> </tr> </thead> <tbody> <tr> <td data-bbox="183 750 630 840">Data mining is the process of analyzing unknown patterns of data</td> <td data-bbox="630 750 1077 840">A data warehouse is database system which is designed for analytical instead of transactional work.</td> </tr> <tr> <td data-bbox="183 840 630 952">Data mining is a method of comparing large amounts of data to finding right patterns.</td> <td data-bbox="630 840 1077 952">Data warehousing is a method of centralizing data from different sources into one common repository.</td> </tr> <tr> <td data-bbox="183 952 630 1064">Data mining is the considered as a process of extracting data from large data sets.</td> <td data-bbox="630 952 1077 1064">Data warehousing is the process of pooling all relevant data together.</td> </tr> <tr> <td data-bbox="183 1064 630 1220">One of the most important benefits of data mining techniques is the detection and identification of errors in the system</td> <td data-bbox="630 1064 1077 1220">One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features.</td> </tr> </tbody> </table> | Data mining | Data warehousing | Data mining is the process of analyzing unknown patterns of data | A data warehouse is database system which is designed for analytical instead of transactional work. | Data mining is a method of comparing large amounts of data to finding right patterns. | Data warehousing is a method of centralizing data from different sources into one common repository. | Data mining is the considered as a process of extracting data from large data sets. | Data warehousing is the process of pooling all relevant data together. | One of the most important benefits of data mining techniques is the detection and identification of errors in the system | One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features. | 4*2=8 | 8 | 8 |
| Data mining | Data warehousing | | | | | | | | | | | | | |
| Data mining is the process of analyzing unknown patterns of data | A data warehouse is database system which is designed for analytical instead of transactional work. | | | | | | | | | | | | | |
| Data mining is a method of comparing large amounts of data to finding right patterns. | Data warehousing is a method of centralizing data from different sources into one common repository. | | | | | | | | | | | | | |
| Data mining is the considered as a process of extracting data from large data sets. | Data warehousing is the process of pooling all relevant data together. | | | | | | | | | | | | | |
| One of the most important benefits of data mining techniques is the detection and identification of errors in the system | One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features. | | | | | | | | | | | | | |
| IX(b) | <p>ObjectStructure: The structure of an object refers to the properties that an object is made up of. These properties of an object are referred to as an attribute.. The object structure is further composed of three types of components: Messages, Methods, and Variables. These are explained as following below.</p> <ol style="list-style-type: none"> 1. Messages – A message provides an interface or acts as a communication medium between an object and the outside world. A message can be of two types: <ul style="list-style-type: none"> • Read-only message: If the invoked method does not change the value of a variable, then the invoking message is said to be a read-only message. • Update message: If the invoked method changes the value of a variable, then the invoking message is said to be an update message. 2. Methods – When a message is passed then the body of code that is executed is known as a method. Every time when a method is executed, it returns a value as output. A method can be of two types: | 7 | 7 | 7 | | | | | | | | | | |

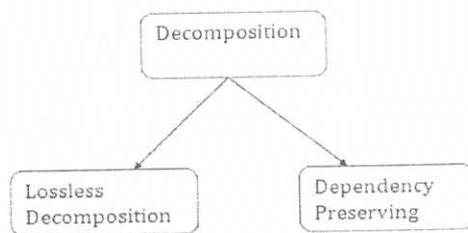
- **Read-only method:** When the value of a variable is not affected by a method, then it is known as read-only method.
 - **Update-method:** When the value of a variable changes by a method, then it is known as an update method.
3. **Variables –**
It stores the data of an object. The data stored in the variables makes the object distinguishable from one another

X(a)

Relational Decomposition

- When a relation in the relational model is not in appropriate normal form then the decomposition of a relation is required.
- In a database, it breaks the table into multiple tables.
- If the relation has no proper decomposition, then it may lead to problems like loss of information.
- Decomposition is used to eliminate some of the problems of bad design like anomalies, inconsistencies, and redundancy.

Types of Decomposition



Lossless Decomposition

- If the information is not lost from the relation that is decomposed, then the decomposition will be lossless.
- The lossless decomposition guarantees that the join of relations will result in the same relation as it was decomposed.
- The relation is said to be lossless decomposition if natural joins of all the decomposition give the original relation.

Dependency Preserving

- It is an important constraint of the database.
- In the dependency preservation, at least one decomposed table must satisfy every dependency.
- If a relation R is decomposed into relation R1 and R2, then the dependencies of R either must be a part of R1 or R2 or must be derivable from the combination of functional dependencies of R1

Decomposition
-3
Types-
3+3

3+3+
3=9

9

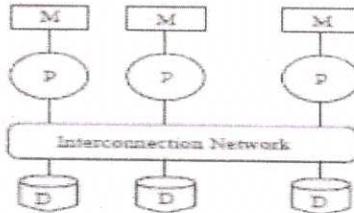
and R2.

X(b)

The three main architectures are

- Shared-memory: all the processors have access to a common memory and storage disks via an interconnection network.

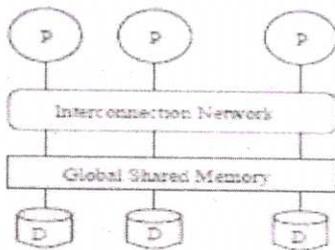
Shared Disk



SHARED DISK

- Shared-disk: each processor has a private memory and shares only storage disks via interconnection network

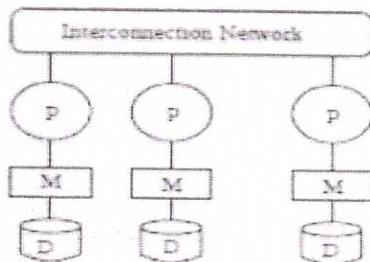
Shared Memory



SHARED MEMORY

- Shared-nothing: each processor has private memory and one or more private disk storage. No two processors can access the same storage.

Shared Nothing



SHARED NOTHING

3*2=6

6

6