Model Answer for ISE-I DBMS

- **2 Marks Questions**

1. **Explain the role of a Database Management System (DBMS). What are the primary functions and role of using a DBMS?**

➢ The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. Database management system is a software which is used to manage the database. DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more. It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.
Primary functions of DBMS are :-

i. Data Definition: It is used for creation, modification, and removal of definition that defines the organization of data in the database.
ii. Data Updation: It is used for the insertion, modification, and deletion of the actual data in the database.
iii. Data Retrieval: It is used to retrieve the data from the database which can be used by applications for various purposes.
iv. User Administration: It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

2. **Describe the Three-Level Architecture of a database system. What are the main levels and their funtions?**

➢ The Three-Level Architecture of a database system is a framework that separates the database system into three distinct levels: Physical, Conceptual and View. This separation helps achieve data independence and simplifies database management. The functions of all these levels are :-

i. Physical Level (Internal Schema) – As the name suggests, the Physical level tells us that where the data is actually stored i.e. it tells the actual location of the data that is being stored by the user. This level defines how the data is physically stored on the hardware (disk, memory, etc.). It deals with the organization of data, storage structures, and access paths such as indexing and file formats.

ii. Conceptual Level (Logical Level) - This level provides an abstract, logical view of the entire database, without concerning itself with how the data is physically stored. It defines the structure of the data, the relationships between different data elements, and constraints (e.g., integrity rules). In short, this level tells how the data is actually stored and structured. We have different data models by which we can store the data.

iii. View Level (External Schema) - This level defines how individual users or user groups interact with the data. It provides specific views or subsets of the data, customized for different users' needs. For example, a sales department might have a different view than the HR department. This level tells the application about how the data should be shown to the user.

3. Differentiate between data and information within the context of databases. How do these concepts relate to database management and analysis?

Parametrs	Data	Information
Definition	Raw, unprocessed facts and figures.	Processed, organized, and meaningful data.
Example	Customer name, age, email, product ID, transaction amount. Such as numbers like "100", text like "John", date like "2023-10-15".	Average customer age, total sales for a product, customer demographics. Such as "John is 30 years old", "Revenue for Q3 2023 is $5 million".
Meaning	Has no inherent meaning by itself.	Has context, relevance, and purpose.
Role in Databases	Stored as basic units in tables (e.g., rows/columns).	Generated by querying and analyzing data to produce insights.
Relevance	Serves as the input for processing.	Serves as the output after processing, useful for decision-making.

In database management, the system's role is to store and manage data efficiently. Through data analysis (queries, reports), this data is transformed into information, which is actionable and useful for users and organizations.

4. **Compare and contrast Single-Values and Multi-Valued Attributes with examples. How do these attribute types affect data representation?**

Parameters	SINGLE-VALUED ATTRIBUTES	MULTI-VALUED ATTRIBUTES
Definition	Attributes that hold a single value for each entity.	Attributes that can hold multiple values for a single entity.
Example	EmployeeID (a single unique ID for each employee).	PhoneNumbers (an employee can have multiple phone numbers).
Representation	Represented as a single field/column in a database table.	Represented as multiple fields or a separate table (in normalized form).
Data Modeling	Simpler and directly represented in a table.	More complex, requiring normalization or a related table to store the multiple values.
Impact on data Representation	Easily represented in a relational table with a single column for each attribute.	Requires additional tables or columns to represent multiple values, often leading to many-to-one or one-to-many relationships.

These attribute types affect data representation in the following ways:

i. Single-valued attributes are straightforward to represent, as each value is stored in a single column of a table.

ii. Multi-valued attributes require more complex handling, often leading to the creation of additional tables to represent the multiple values. For instance, a separate table may be used to store each phone number for an employee, with a foreign key linking back to the employee's record.

iii.

5. **Describe the purpose of composite attribute and provide an example. How do composite attribute enhance the representation of data in the ER Model?**

➢ A composite attribute is an attribute in an Entity-Relationship (ER) model that can be broken down into smaller sub-parts, which represent more specific attributes. The sub-parts or components collectively form the complete value of the composite attribute.

The purpose of a composite attribute is to group related pieces of information together into a single logical unit, improving the clarity and organization of data. For example, consider an entity ADDRESS. The composite attribute Address can be broken down

into smaller sub-attributes like STREET, CITY, STATE and POSTAL CODE. Together, these sub-attributes form the complete ADDRESS. Instead of storing each part of the address as separate attributes, the ADDRESS is treated as a single composite attribute.

Composite attributes enhance data representation in the Entity-Relationship (ER) model by grouping related sub-attributes into a single, logical unit. This simplifies the data structure and improves clarity. For example, an ADDRESS can be represented as a composite attribute consisting of STREET, CITY, STATE and POSTAL CODE. Instead of modeling each part separately, the composite attribute organizes these related components together, making the data model more concise and easier to understand. This reduces redundancy and reflects the real-world relationships more naturally, enhancing the overall representation of complex data.

6. **List the Data Control Language (DCL) commands. What are the roles of these commands in managing database permissions and access?**

- The Data Control Language (DCL) commands are used to control access to data and manage permissions in a database. The main DCL commands and their roles in managing database permissions and access are:

i. GRANT: Gives specific privileges (like SELECT, INSERT, UPDATE, DELETE) to users or roles on database objects (such as tables, views, or procedures). GRANT allows database administrators (DBAs) to assign access rights and privileges to users or roles, controlling who can perform specific actions on the database.

ii. REVOKE: Removes previously granted privileges from users or roles, restricting their access to database objects. REVOKE ensures that access can be withdrawn if necessary, enhancing security by restricting unauthorized access or activities.

These commands help in managing data security, ensuring that users only have the necessary permissions to perform their tasks while maintaining control over database access.

7. **Differentiate between the DELETE and DROP commands in SQL. How do these commands affect database tables and records?**

Parameters	DELETE	DROP
Purpose	Removes specific rows or records from a table.	Removes an entire table (or database object) from the database.
Effect on Data	Deletes data (records) but keeps the table structure intact.	Removes the table and all of its data, constraints, and indexes permanently.
Usage	Can be used with a WHERE clause to delete specific rows.	Used to delete the whole table or object, without conditions.
Rollback Capability	Can be rolled back (if in a transaction).	Cannot be rolled back once executed (unless using a transaction).
Impact on Structure	Does not affect the structure or schema of the table.	Completely removes the table or object from the database.
Performance	Slower for large tables, as it removes rows one by one.	Faster, as it directly removes the entire table and all associated data.

- **5 Marks Questions**

1. **Explain the concept of Specialization in the EER Model and provide an example of its application. How does specialization refine entity definitions?**

➢ Specialization in the Enhanced Entity-Relationship (EER) model is the process of dividing a general entity (called a supertype) into more specific subtypes based on shared characteristics or behaviors. This technique helps to refine entity definitions and capture unique properties of more specific categories, improving data organization and representation.

i. Supertype: A general entity containing attributes common to all its subtypes.
ii. Subtypes: More specific entities derived from the supertype, with additional attributes or relationships unique to each.
iii. Disjoint vs. Overlapping: In disjoint specialization, an instance can belong to only one subtype. In overlapping specialization, an instance can belong to multiple subtypes.

iv. Total vs. Partial: In total specialization, every instance of the supertype must belong to at least one subtype. In partial specialization, some instances of the supertype may not belong to any subtype.
For Example:-
Consider a Vehicle entity:

- **Vehicle (Supertype)**: Common attributes like vehicle ID, manufacturer, model year.
- **Subtypes**:

i. **Car**: Additional attribute like "number of doors."
ii. **Truck**: Additional attribute like "payload capacity."
iii. **Motorcycle**: Additional attribute like "engine type."

One more example we can see is of a UNIVERSITY DATABASE.

- **General Entity (Superclass): Person**
  - Attributes: Person_ID, Name, Date_of_Birth
- **Specialized Entities (Subtypes):**

i. **Student**
   - Additional Attributes: Student_ID, Program, Year_of_Study

ii. **Faculty**
   - Additional Attributes: Faculty_ID, Department, Salary

iii. **Staff**
   - Additional Attributes: Staff_ID, Role, Office_Location

In above example:

- The **Person** entity is the general (superclass) entity that holds common attributes shared by all types of persons (e.g., name, ID, date of birth).
- The **Student**, **Faculty**, and **Staff** entities are specialized subtypes, each with its own additional attributes that are specific to that category of person. For example, only **Students** will have information about their Program and Year_of_Study, while **Faculty** will have attributes like Department and Salary.

In following ways specialization refine entity definitions:-

i. Specialization refines the **Vehicle** entity by ensuring that each subtype captures unique attributes (e.g., "payload capacity" for Trucks).
ii. It reduces redundancy and ensures business rules are enforced (e.g., only trucks can have "cargo capacity").
iii. It provides a clearer, more organized model by differentiating between vehicle types while preserving common attributes in the supertype.
iv. This technique makes the model more efficient, flexible, and easier to maintain.

2. Explain the importance of transaction management in a database system. Discuss the roles of COMMIT, ROLLBACK and SAVEPOINT commands in ensuring data consistency and integrity.

- Transaction management is critical in a database system to ensure that the database remains consistent, accurate, and reliable during concurrent access by multiple users or applications. A transaction is a logical unit of work that must be completed entirely or not at all (atomicity). Transaction management helps to ensure that the database maintains its integrity even in the event of system failures, errors, or crashes. The key goals of transaction management are to maintain ACID properties (Atomicity, Consistency, Isolation, Durability).

  The roles of COMMIT, ROLLBACK and SAVEPOINT commands in ensuring data consistency and integrity are:-

  i. **COMMIT**

  The COMMIT command is used to make all the changes made during a transaction permanent. Once a COMMIT is issued, the database records all modifications (inserts, updates, deletes) made during the transaction. This ensures that the transaction has been completed successfully, and the changes are reflected in the database. The COMMIT command is crucial for maintaining the Durability property of ACID (Atomicity, Consistency, Isolation, Durability), ensuring that once a transaction is committed, its results are not lost, even in the case of a system crash.

  ii. **ROLLBACK**

  The ROLLBACK command is used to undo all changes made during the current transaction, reverting the database to its state before the transaction began. This command is essential for ensuring Atomicity—the principle that a transaction is an all-or-nothing operation. If an error occurs during the transaction (e.g., constraint violation, invalid data input), a ROLLBACK ensures that no partial changes are made to the database, and it restores the database to a consistent state.

  iii. **SAVEPOINT**

  The SAVEPOINT command allows the transaction to be divided into smaller, more manageable sections by marking specific points during the transaction. If an error occurs after a savepoint, the transaction can be rolled back to that point rather than completely undoing all changes. This provides greater flexibility and granularity in transaction management. For example, in complex transactions involving multiple steps, a SAVEPOINT enables partial rollbacks, ensuring that earlier, successful operations can be preserved while only reversing the problematic part of the transaction. This enhances control over long-running or complex transactions without having to discard all changes.

3. **Describe constraints in a database and elaborate on their significance in maintaining data quality. Differentiate between UNIQUE, PRIMARY KEY, FOREIGN KEY and CHECK constraints, providing examples for each.**

- In a database, constraints are rules that define the conditions under which data is stored, ensuring data integrity, consistency, and quality. Constraints restrict the type, format, or value of data that can be inserted into a table. They are essential for maintaining accurate, reliable, and valid data throughout the system. Constraints help in enforcing the business rules and ensure that the database remains logically correct. There are many types of Constraints such as UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, etc.

  Constraints are essential for maintaining data quality in a database by ensuring that data remains accurate, consistent, and reliable. They enforce rules that prevent invalid, redundant, or inconsistent data from being entered into the database. For example, unique constraints ensure that duplicate records are avoided, primary keys guarantee each row has a unique identifier, foreign keys maintain relationships between tables and prevent orphaned records, and check constraints enforce specific data conditions, such as valid ranges or formats. By defining these rules, constraints help preserve data integrity, consistency, and reliability, ensuring that the database accurately reflects the intended business logic and prevents data anomalies.

Constraint	Purpose	Uniqueness	Nullability	Example
UNIQUE	Ensures all values in a column are unique.	Yes	Can be NULL	Email VARCHAR(255) UNIQUE
PRIMARY KEY	Uniquely identifies each row in the table.	Yes	Cannot be NULL	StudentID INT PRIMARY KEY
FOREIGN KEY	Ensures referential integrity between tables.	Yes	Can be NULL	FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
CHECK	Ensures a condition or rule is satisfied for data.	Depends on the condition	Can be NULL	CHECK (Age >= 18)