

# MCQ Answering and Explanation

Sure! Paste the MCQs here, and I'll analyze them carefully, generate the answers, and explain the solving process where needed in a structured Q&A format.

Here are the answers to the MCQs with explanations where needed:

---

## 1.1 What is metadata in a database?

**Answer:** Metadata is **data about data**. It describes the structure, properties, and constraints of data stored in a database, including information like table names, column types, constraints, indexes, and relationships.

---

## 1.2 Specify the ER notations for the different categories of attributes.

**Answer:**

- **Simple Attribute** → Represented as an oval connected to an entity.
  - **Composite Attribute** → An oval with sub-ovals representing its components.
  - **Derived Attribute** → An oval with a dashed border.
  - **Multivalued Attribute** → An oval with a double border.
  - **Key Attribute** → An underlined oval.
- 

## 1.3 If clauses HAVING, GROUP BY, WHERE, ORDER BY are used in a SELECT statement, then the correct order is \_\_\_\_\_.

**Answer:**

The correct order of execution in SQL is:

1. **WHERE** (Filters individual rows)

2. **GROUP BY** (Groups rows based on common values)
3. **HAVING** (Filters groups)
4. **ORDER BY** (Sorts the result)

Thus, the correct order is:

**WHERE → GROUP BY → HAVING → ORDER BY**

---

## 1.4 List out the different types of join operations in SQL.

Answer:

- **INNER JOIN** – Returns only matching records between tables.
  - **LEFT JOIN (LEFT OUTER JOIN)** – Returns all records from the left table and matching records from the right.
  - **RIGHT JOIN (RIGHT OUTER JOIN)** – Returns all records from the right table and matching records from the left.
  - **FULL JOIN (FULL OUTER JOIN)** – Returns all records from both tables, filling missing values with NULL.
  - **CROSS JOIN** – Produces a Cartesian product of both tables.
  - **SELF JOIN** – A table is joined with itself.
- 

## 1.5 What is the use of Normalization?

Answer:

Normalization **reduces redundancy and improves data integrity** by organizing data into multiple related tables. It helps in **minimizing anomalies** (insertion, update, deletion anomalies).

---

**1.6 Given relation  $R(A, B, C, D)$  with FD sets  $FD1 = \{A \rightarrow B, B \rightarrow C, AB \rightarrow D\}$  and  $FD2 = \{A \rightarrow B, B \rightarrow C, A \rightarrow C, A \rightarrow D\}$ . Check whether  $FD1$  is equivalent to  $FD2$ .**

**Answer:**

To check equivalence, we need to see if  $FD1 \subseteq FD2$  and  $FD2 \subseteq FD1$  by computing the closure of attributes.

- **FD1 Closure Calculation:**

- $A^+ = \{A, B\}$  ( $A \rightarrow B$ )
- $B^+ = \{B, C\}$  ( $B \rightarrow C$ )  $\rightarrow A^+ = \{A, B, C\}$
- $AB^+ = \{A, B, C, D\}$  ( $AB \rightarrow D$ )

So,  $A^+ = \{A, B, C, D\}$

- **FD2 Closure Calculation:**

- $A^+ = \{A, B, C, D\}$  ( $A \rightarrow B, A \rightarrow C, A \rightarrow D$ )

Since both produce the **same closure**,  $FD1$  and  $FD2$  are **equivalent**.

---

## **1.7 List out the desirable properties of transactions.**

**Answer:**

The desirable properties of transactions are **ACID**:

- **Atomicity** – A transaction is either fully completed or not at all.
  - **Consistency** – The database remains in a consistent state before and after the transaction.
  - **Isolation** – Transactions do not interfere with each other.
  - **Durability** – Once committed, the changes remain even after system failure.
- 

**1.8 Let  $F = \{A \rightarrow B, AB \rightarrow E, BG \rightarrow E, CD \rightarrow I, E \rightarrow C\}$ . Find the closures of  $A^+$  and  $(AE)^+$ .**

Answer:

- **A+ Calculation:**

- $A \rightarrow B$  ( $A^+ = \{A, B\}$ )
- $AB \rightarrow E$  ( $A^+ = \{A, B, E\}$ )
- $E \rightarrow C$  ( $A^+ = \{A, B, E, C\}$ )
- $CD \rightarrow I$  does not apply since D is not in  $A^+$ .

So,  $A^+ = \{A, B, E, C\}$

- **(AE)+ Calculation:**

- $A \rightarrow B$  ( $AE^+ = \{A, E, B\}$ )
- $AB \rightarrow E$  ( $AE^+$  already has E)
- $E \rightarrow C$  ( $AE^+ = \{A, E, B, C\}$ )
- $CD \rightarrow I$  does not apply since D is not in  $AE^+$ .

So,  $(AE)^+ = \{A, B, E, C\}$

---

## 1.9 Does Elasticsearch have a schema? Justify your answer.

Answer:

ElasticSearch is **schema-less**, meaning it does not require a predefined schema like SQL databases. However, it uses **dynamic mapping** to infer data types but allows for explicit schema definitions if needed.

---

## 1.10 Differentiate between SQL and NoSQL databases.

Answer:

Feature	SQL Database	NoSQL Database
Structure	Structured, Table-based	Unstructured, Key-Value, Document, Column, Graph-based
Schema	Fixed Schema	Flexible Schema
Scalability	Vertical Scaling	Horizontal Scaling

Feature	SQL Database	NoSQL Database
ACID	Strong ACID Compliance	Eventual Consistency
Examples	MySQL, PostgreSQL	MongoDB, Cassandra

---

## 1.11 List out the different concurrency problems in DBMS transactions.

Answer:

- **Dirty Read** – A transaction reads uncommitted data from another transaction.
- **Lost Update** – Two transactions update the same data, causing one update to be lost.
- **Non-Repeatable Read** – A transaction reads data twice, but another transaction modifies it in between.
- **Phantom Read** – A transaction reads a set of rows, but another transaction inserts/deletes rows, changing the results.

---

This format should be easy to paste elsewhere! 🚀 Let me know if you need modifications! 😊

Here are the answers to the MCQs in a structured Q&A format:

---

## 1.1 A relational database consists of a collection of \_\_\_\_.

Answer: Tables (Relations)

A relational database consists of multiple **tables** that store data in rows and columns, with relationships established between them.

---

## 1.2 The term \_\_\_\_ is used to refer to a row.

**Answer: Tuple**

A **tuple** is a single record (row) in a table, containing values for different attributes (columns).

---

**1.3 A domain is atomic if elements of the domain are considered to be \_\_\_\_\_ unit.**

**Answer: Indivisible**

An atomic domain means that values in the domain cannot be further divided, ensuring **1NF (First Normal Form)** compliance.

---

**1.4 In SQL, the spaces at the end of the strings are removed by \_\_\_\_\_ function.**

**Answer: RTRIM()**

The **RTRIM()** function removes trailing spaces from a string.

---

**1.5 If we want to retain all duplicates, we must write \_\_\_\_\_ in place of UNION.**

**Answer: UNION ALL**

The **UNION** operator removes duplicates, whereas **UNION ALL** retains all occurrences, including duplicates.

---

**1.6 What is an Entity?**

**Answer:**

An **entity** is an **object** or **concept** that can be uniquely identified and stored in a database.

Example: Student, Employee, Car.

---

## 1.7 What are ACID properties?

Answer:

ACID properties ensure reliability in transactions:

- **Atomicity** – A transaction is fully completed or not executed at all.
  - **Consistency** – Ensures that database constraints are maintained.
  - **Isolation** – Ensures transactions do not interfere with each other.
  - **Durability** – Ensures committed transactions persist even after system failure.
- 

## 1.8 List different types of notations used for attribute representation in an E-R diagram.

Answer:

- Simple Attribute → Oval
  - Composite Attribute → Oval with sub-ovals
  - Derived Attribute → Dashed oval
  - Multivalued Attribute → Double-bordered oval
  - Key Attribute → Underlined oval
- 

## 1.9 If a relation is in BCNF, then the relation should be \_\_\_\_.

Answer: In 3NF and have no partial dependencies

BCNF (Boyce-Codd Normal Form) is a stricter version of 3NF, ensuring that every determinant is a **super key**.

---

## 1.10 List all cardinality ratio representations in an E-R diagram.

Answer:

- One-to-One (1:1)
  - One-to-Many (1:M)
  - Many-to-One (M:1)
  - Many-to-Many (M:N)
- 

## 1.11 Define Sharding.

Answer:

**Sharding** is a database partitioning technique where large datasets are split into smaller, more manageable **horizontal partitions** across multiple servers to improve **scalability and performance**.

---

## 1.12 List three layers in the three-schema architecture.

Answer:

1. **Internal Schema** – Defines physical storage.
  2. **Conceptual Schema** – Defines logical structure.
  3. **External Schema** – Defines user views.
- 

## 1.13 Define Serialization.

Answer:

Serialization ensures that transactions execute in a **sequential** order, preserving **consistency** and avoiding conflicts in a multi-user database.

---



## 1.14 List types of locks used in concurrency control.

Answer:

- Shared Lock (S-Lock) – Allows read access.
  - Exclusive Lock (X-Lock) – Allows read and write access.
  - Binary Lock – Can be locked or unlocked.
  - Two-Phase Locking (2PL) – Ensures serializability by acquiring locks in **growing** and **shrinking** phases.
- 

## 1.15 Define Namespace.

Answer:

A **namespace** is a container for uniquely identifying database objects, such as tables or collections, preventing name conflicts within a system.

---

This format makes it easy for you to paste elsewhere. Let me know if you need further clarification! 😊

Here are the answers in a structured format:

---

## 1.1 \_\_\_\_\_ is a collection of programs that enables users to create and maintain a database.

Answer: Database Management System (DBMS)

A DBMS allows users to **store, retrieve, manage, and modify** data efficiently.

---

## 1.2 Give an example for Entity type.

Answer: Student, Employee, Product, Car

An entity type represents a category of real-world objects with common attributes.

---

## 1.3 Differentiate between operation and operation.

Answer:

*(Need more details to specify the operations being compared.)*

---

## 1.4 Mention the different languages present in DBMS.

Answer:

1. DDL (Data Definition Language) – Defines schema (e.g., CREATE, ALTER).
  2. DML (Data Manipulation Language) – Manages data (e.g., INSERT, UPDATE).
  3. DCL (Data Control Language) – Controls access (e.g., GRANT, REVOKE).
  4. TCL (Transaction Control Language) – Manages transactions (e.g., COMMIT, ROLLBACK).
- 

## 1.5 'AS' clause is used in SQL for \_\_\_\_\_ operation.

Answer: Alias (Renaming Columns or Tables)

Example:

```
sql
```

```
SELECT name AS student_name FROM Students;
```

This renames the **name** column as **student\_name**.

---

**1.6 The statement in SQL which allows changing the definition of a table is \_\_\_\_\_.**

Answer: ALTER TABLE

The ALTER TABLE command modifies table structure, such as adding or deleting columns.

---

**1.7 Mention the function of X in Y.**

Answer: (Need more details to specify the function and system being referenced.)

---

**1.8 \_\_\_\_\_ is a constraint between sets of attributes from the database relation.**

Answer: Functional Dependency (FD)

A functional dependency defines relationships between attributes in a relation.

---

**1.9 State Boyce-Codd Normal Form (BCNF).**

Answer:

A relation is in BCNF if for every functional dependency  $X \rightarrow Y$ , X is a superkey.

---

**1.10 When do you say a schedule is cascadeless?**

Answer:

A schedule is cascadeless if it does not lead to cascading rollbacks, meaning no transaction depends on uncommitted changes from another transaction.

---

### 1.11 Define Fourth Normal Form (4NF).

Answer:

A relation is in 4NF if it is in BCNF and has no multi-valued dependencies.

---

### 1.12 A binary lock can have \_\_\_\_\_ and \_\_\_\_\_ states.

Answer: Locked and Unlocked

A binary lock ensures that a data item can be either accessed or restricted.

---

### 1.13 List the two conditions that decide whether a given dependency is trivial.

Answer:

1. If  $Y \subseteq X$  in  $X \rightarrow Y$  (i.e.,  $Y$  is a subset of  $X$ ).
2. If  $Y$  is already functionally dependent on  $X$  by definition.

Example:  $A \rightarrow A$  is trivial.

---

### 1.14 Elasticsearch uses \_\_\_\_\_ as the serialization format for documents.

Answer: JSON (JavaScript Object Notation)

Elasticsearch stores and retrieves data in **JSON format** for easy processing.

---

### 1.15 \_\_\_\_\_ is specified using a request body.

Answer: Query (or Index Mapping in Elasticsearch)

Elasticsearch queries and mappings are sent via an HTTP request body.

---

This format ensures clarity and easy pasting. Let me know if you need refinements! 😊

Here are the answers in a structured format:

---

**1.1 \_\_\_\_\_ is concerned with rearrangement and possible reordering of operations, use of correct algorithms and indexes during execution of queries in component modules of DBMS.**

**Answer: Query Optimization**

Query optimization improves query execution efficiency by choosing the best execution plan.

---

**1.2 Capacity to change internal schema without having to change conceptual schema is called \_\_\_\_\_.**

**Answer: Physical Data Independence**

It ensures that modifications in storage structure do not affect the logical schema.

---

**1.3 Justify with an example that weak entity type always has a total participation constraint with respect to its identifying relationship.**

**Answer:**

A weak entity cannot exist without a strong entity and has a total participation constraint.

Example:

- "Dependent" (weak entity) depends on "Employee" (strong entity).
  - A dependent must be linked to an employee, meaning total participation in the relationship.
- 

## 1.4 Define complex attribute with example.

Answer:

A complex attribute is a combination of multi-valued and composite attributes.

Example:

- Address = { (Street, City), {Phone Numbers} }
- 

## 1.5 Name any two unary and binary relational operations.

Answer:

- Unary Operations: SELECT ( $\sigma$ ), PROJECT ( $\pi$ )
  - Binary Operations: JOIN, UNION
- 

## 1.6 What are the reasons for any attribute to be NULL? Give an example.

Answer:

Reasons for NULL values:

1. Missing data (e.g., phone number not provided).
2. Not applicable (e.g., spouse name for an unmarried person).

Example:

```
sql
```

```
INSERT INTO Students (ID, Name, Age, Email) VALUES (101, 'John', 20, NULL);
```

Here, Email is NULL since the student has not provided it.

---

## 1.7 Prove that $A \rightarrow E$ using Armstrong's rule.

Answer:

Using Armstrong's Axioms:

1. Given Functional Dependencies:  $A \rightarrow B$ ,  $B \rightarrow C$ ,  $A \rightarrow D$ ,  $D \rightarrow E$
2. Applying Transitivity:
  - $A \rightarrow B$ ,  $B \rightarrow C \Rightarrow A \rightarrow C$
  - $A \rightarrow D$ ,  $D \rightarrow E \Rightarrow A \rightarrow E$

Thus,  $A \rightarrow E$  is derived.

---

## 1.8 "DROP TABLE" deletes the \_\_\_\_\_ along with all of its indexes.

Answer: Table

**DROP TABLE** removes the table structure, data, and indexes permanently.

---

## 1.9 Collections are allocated space inside each data file in chunks called \_\_\_\_\_.

Answer: Extents

In MongoDB, collections use **extents** to store data efficiently.

---

**1.10 A node is a \_\_\_\_\_ of Elasticsearch, while \_\_\_\_\_ consists of one or more nodes.**

Answer:

- Node is a single server in Elasticsearch.
  - Cluster consists of one or more nodes working together.
- 

**1.11 Compare “failed state” and “terminated state” of a transaction.**

Answer:

- **Failed State:** A transaction cannot proceed due to errors or conflicts.
  - **Terminated State:** A transaction has completed, either committed or aborted.
- 

**1.12 \_\_\_\_\_ algorithm was proposed to try to reduce the number of needless aborts/restarts in transaction processing.**

Answer: Thomas Write Rule

It avoids unnecessary rollbacks by discarding outdated writes in concurrency control.

---

**1.13 Two tables needed for efficient transaction recovery are \_\_\_\_\_ and \_\_\_\_\_.**

Answer:

1. Transaction Table (stores active transactions).
  2. Log File (Redo/Undo Log) (records all changes for recovery).
-



This format ensures clarity and easy reference. Let me know if you need more details! 😊

Here are the answers to your questions in a structured format:

---

**1.1 \_\_\_\_\_ is a collection of programs that enables users to create and maintain a database.**

**Answer:** Database Management System (DBMS)

DBMS allows users to store, retrieve, and manage data efficiently.

---

**1.2 Define data abstraction.**

**Answer:**

Data abstraction is the process of hiding complex implementation details and providing a simplified user view of the data.

It has three levels:

1. **Physical Level** (low-level storage details).
  2. **Logical Level** (structure of data, relationships).
  3. **View Level** (user interaction with data).
- 

**1.3 The processes for transforming requests and results between levels are called \_\_\_\_\_.**

**Answer:** Mapping

Mappings convert data requests between the internal, conceptual, and external levels of the database.

---

## 1.4 What is participation constraint?

Answer:

A participation constraint defines whether an entity is required to participate in a relationship.

- **Total Participation:** Every entity in the set **must** participate.
- **Partial Participation:** Some entities **may** participate.

Example:

- In a "Works\_For" relationship, **every Employee must be assigned to a Department** (total participation).

---

## 1.5 Differentiate between UNION operation and INTERSECT operation.

Answer:

UNION	INTERSECT
Combines all tuples from two relations, removing duplicates.	Returns only common tuples present in both relations.
Syntax: <code>SELECT * FROM A UNION SELECT * FROM B;</code>	Syntax: <code>SELECT * FROM A INTERSECT SELECT * FROM B;</code>
Example: Combining students from two classes.	Example: Finding students enrolled in both classes.

---

## 1.6 Give the general syntax of ALTER command.

Answer:

The **ALTER** command is used to modify a table structure.

```
sql
```

```
ALTER TABLE table_name
ADD column_name datatype; -- Add a new column
```

sql

```
ALTER TABLE table_name  
DROP COLUMN column_name; -- Remove a column
```

sql

```
ALTER TABLE table_name  
MODIFY column_name new_datatype; -- Change datatype
```

---

## 1.7 Mention the function of INDEX in SQL.

**Answer:**

An INDEX improves query performance by speeding up data retrieval.

**Example:**

sql

```
CREATE INDEX idx_name ON Students(Name);
```

This creates an index on the "Name" column for faster searches.

---

## 1.8 \_\_\_\_\_ is a constraint between two sets of attributes from the database.

**Answer:** Functional Dependency

**Example:**

- Roll\_No → Student\_Name (Roll number uniquely determines student name).

---

## 1.9 When do you say that any two sets of functional dependencies are equivalent?

**Answer:**

Two sets of **functional dependencies (FDs)** are **equivalent** if they **imply** each other.

That is, if  $F1 \subseteq F2$  and  $F2 \subseteq F1$ , then **F1 and F2 are equivalent**.

---

### **1.10 Give any two conditions for a set of functional dependencies to be minimal.**

**Answer:**

A set of functional dependencies is **minimal** if:

1. **No FD can be removed** without changing the closure.
2. **No attribute in the left side** of any FD can be removed without affecting closure.

Example:

- $A \rightarrow BC$  is minimal, but  $A \rightarrow B, A \rightarrow C$  is not minimal.
- 

### **1.11 State Boyce–Codd Normal Form (BCNF).**

**Answer:**

A relation is in **BCNF** if:

1. It is in **3NF**.
2. For every FD ( $X \rightarrow Y$ ), **X must be a superkey**.

BCNF removes **redundancy** by ensuring **no partial dependencies** exist.

---

### **1.12 When do you say a schedule is cascadeless?**

**Answer:**

A schedule is **cascadeless** if **no transaction reads uncommitted data** from another transaction.

It prevents **cascading rollbacks**, improving transaction reliability.

---

**1.13 A binary lock can have \_\_\_\_\_ and \_\_\_\_\_ states.**

Answer: Locked, Unlocked

A **binary lock** restricts access to a data item by allowing only **one transaction** at a time.

---

**1.14 \_\_\_\_\_ is a name of a field or property.**

Answer: Attribute

In a **relation (table)**, attributes are **columns** storing **specific data**.

Example: "Name", "Age", "Salary" are attributes.

---

**1.15 Elasticsearch uses \_\_\_\_\_ as the serialization format for documents.**

Answer: JSON (JavaScript Object Notation)

Elasticsearch stores and retrieves data in **JSON format**, making it flexible and lightweight.

---

**1.16 \_\_\_\_\_ is specified using a request body in Elasticsearch.**

Answer: Search Query

In Elasticsearch, queries are specified in the **request body** in JSON format.

Example:

```
json
```

```
{
  "query": {
    "match": {
      "name": "Harshit"
    }
  }
}
```

---

This format ensures **clarity, simplicity, and easy reference**. Let me know if you need further explanations! 😊

I can extract and answer the questions from the image. Here are the answers:

---

## 1.1 Find the minimal cover for the given functional dependencies:

Given FDs:

$P \rightarrow L$ ,  $P \rightarrow CA$ ,  $LC \rightarrow AP$ ,  $A \rightarrow LC$

Steps:

1. **Remove extraneous attributes:**
  - No redundant attributes in individual dependencies.
2. **Decompose to minimal form:**
  - Already minimal.
3. **Final minimal cover:**  
 $P \rightarrow L$ ,  $P \rightarrow CA$ ,  $LC \rightarrow AP$ ,  $A \rightarrow LC$

---

## 1.2 SQL Query for Employee Names Containing “TH” or “LL”

```
sql
```

```
SELECT Ename
FROM EMP
WHERE Ename LIKE '%TH%' OR Ename LIKE '%LL%';
```

### 1.3 Difference Between COUNT(\*) and COUNT(Column\_Name)

- **COUNT(\*)** : Counts all rows, including NULL values.
- **COUNT(Column\_Name)** : Counts only non-NULL values in the given column.

Example:

ID	Name
1	Alice
2	NULL
3	Bob

- **COUNT(\*) = 3**
- **COUNT(Name) = 2**

### 1.4 Find the Key for Relation R with Given FDs

$R = \{A, B, C, D, E, F, G, H, I\}$

FDs:

$A \rightarrow BH, BC \rightarrow E, ED \rightarrow AF, G \rightarrow IH, FH \rightarrow CG$

Steps to Find Candidate Key:

- Closure of A:  **$\{A, B, H, C, E, D, F, G, I\}$**
- Since  **$\{A\}$**  determines all attributes, **A is the key.**

## 1.5 Normal Form for Composite Key Dependency

- If an attribute of a **composite key** is dependent on another attribute of the **same composite key**, the relation is in 2NF but not in 3NF.
- 

## 1.6 Conditions for a Trivial MVD ( $X \twoheadrightarrow Y$ is trivial)

1. Y is a subset of X ( $Y \subseteq X$ ).
  2.  $X \rightarrow Y$  (Functional Dependency exists).
- 

## 1.7 Types of Attributes in ER Diagram & Notations

1. **Simple Attribute** - Single-valued (e.g., Age).
  2. **Composite Attribute** - Divisible (e.g., Name  $\rightarrow$  First, Last).
  3. **Derived Attribute** - Computed (e.g., Age from DOB).
  4. **Multivalued Attribute** - Multiple values (e.g., Phone Numbers).
  5. **Key Attribute** - Unique identifier (e.g., Roll No).
- 

## 1.8 Referential Integrity & Cascade Deletion

When deleting tuple (3,4):

- Any tuple with C = 3 must also be deleted.

Affected tuples: (5,3), (7,3)

---

## 1.9 Aggregation Function & GROUP BY in Relational Algebra

- **Aggregation Functions:** COUNT, SUM, AVG, MIN, MAX



- **GROUP BY Clause in Relational Algebra:**

```
sql
```

```
y DeptNo, COUNT(E_ID) (EMP)
```

Groups employees by **DeptNo** and counts them.

---

**1.10 Attribute of R is \_\_\_\_\_ if it is a member of some candidate key.**

Answer: Prime Attribute

---

**1.11 Main Problem with Recoverable Schedules**

Answer: Cascading Rollbacks

---

**1.12 \_\_\_\_\_ is a transaction property ensured by concurrency control.**

Answer: Serializability

---

Let me know if you need further explanations! 😊

Here's the formatted Q&A for you to paste into your document:

---

**1.1 Define Database Schema.**

A **database schema** defines the structure of a database, including tables, attributes, data types, constraints, and relationships.

**1.2 \_\_\_\_\_ is the characteristic that allows program data independence & program operational independence.**

Answer: Data Abstraction

**1.3 \_\_\_\_\_ entity types do not have key attributes of their own.**

Answer: Weak Entity

**1.4 \_\_\_\_\_ is the relationship type with two-degree classification.**

Answer: Binary Relationship

**1.5 \_\_\_\_\_ indicates the maximum number of entities that can be involved in a relationship.**

Answer: Cardinality

**1.6 \_\_\_\_\_ symbol is used to represent entity relationship set in an E-R Diagram.**

Answer: Diamond

**1.7 In an E-R Diagram, double ovals are used to denote \_\_\_\_\_ attributes.**

Answer: Multivalued Attributes

**1.8 How are derived attributes denoted in an E-R diagram?**

Derived attributes are represented using dashed ovals in an E-R diagram.

**1.9 What is Normalization?**

Normalization is the process of organizing a database to reduce redundancy and dependency, improving data integrity by dividing tables into smaller, structured forms.

**1.10 \_\_\_\_\_ is a query used to retrieve the data from the database.**

Answer: SELECT Query

**1.11 Consider the following table of loan\_record & predict the output for the given SQL query.**

(Provide the table and SQL query output based on the data.)

**1.12 \_\_\_\_\_ storage engine of MongoDB uses memory-mapped files to store data.**

Answer: WiredTiger

**1.13 \_\_\_\_\_ is used to inspect & check the amount of mapped memory in MongoDB.**

Answer: `db.serverStatus().mem`

**1.14 MongoDB reports its triggered page faults as total number of page faults in \_\_\_\_\_ no. of seconds.**

Answer: 1 Second

**1.15 What is the need for a concurrency control mechanism?**

Concurrency control ensures data consistency, prevents conflicts, and maintains integrity when multiple transactions access the database simultaneously.

**1.16 What is a deadlock & list the conditions that cause a deadlock.**

A **deadlock** occurs when two or more transactions wait indefinitely for each other to release locks.

**Conditions for Deadlock:**

1. **Mutual Exclusion** – Only one process can hold a resource.
2. **Hold and Wait** – A process holding a resource waits for another.
3. **No Preemption** – Resources cannot be forcibly taken.
4. **Circular Wait** – A closed loop of waiting processes exists.

**1.17 Suppose we log only "after images" of an updated page. Describe what should happen at commit time & what should happen during transaction execution time.**

- **During execution** – The system records the **after-image** (updated version) of the page before committing.
- **At commit time** – The after-image is written to permanent storage, ensuring durability. If a failure occurs before commit, the transaction must be rolled back.

Let me know if you need any modifications! 😊

## Questions and Answers

1. For each attribute of a relation, there is a set of permuted values, called the \_\_\_\_\_ of the attribute.

Answer: Domain

2. Characteristic of DBMS which allows program operation independence and program data independence is called \_\_\_\_\_.

Answer: Data Abstraction

3. Following table has two attributes A and C where A is the primary key and C is the foreign key referencing A with ON DELETE CASCADE:

A	C
2	4
3	4
4	3
5	2
7	2
9	5
6	4

The set of tuples that must be additionally deleted to preserve referential integrity when the tuple (2,4) is deleted.

Answer: (5,2) and (7,2)

4. If E and E be two entities in an ER diagram with single-valued attributes, R and R are two relationships between E and E, where R is one-to-many and R is many-to-many. R and R do not have any attributes on their own. What is the minimum number of tables required to represent this selection in the relational model?

Answer: 3

5. Consider the relations  $r(P,Q,R)$ ,  $r(R,S,T)$  with primary keys P and R respectively. The relation r contains 2000 tuples and r contains 2500 tuples. The max size of the form  $r \bowtie r$  is \_\_\_\_\_.

Answer: 5,000,000 ( $2000 \times 2500$ )

6. Consider the following relational schemas for the library database:

Collection(Title, Author, Catalog\_no.)

Book(Title, Author, Catalog\_no, Publisher, Year, Price)

With the following Functional Dependencies (FDs):

a) Title, Author  $\rightarrow$  Catalog\_no

b) Catalog\_no  $\rightarrow$  Title, Author, Publisher, Year

c) Publisher, Title, Year  $\rightarrow$  Price

Assume {Author, Title} is the key for both schemes. Which normalized form is the relation Book and Collection in?

Answer: 3NF

7. Differentiate between 3NF and BCNF.

Answer:

- **3NF (Third Normal Form):** A relation is in 3NF if it is in 2NF and all non-key attributes are dependent only on the primary key.
- **BCNF (Boyce-Codd Normal Form):** A relation is in BCNF if for every functional dependency ( $X \rightarrow Y$ ), X is a superkey. BCNF is stricter than 3NF.

8. MongoDB is a \_\_\_\_\_ database. The concatenation of the collection name and database name is called \_\_\_\_\_.

Answer: NoSQL, Namespace

9. The database system must take special actions to ensure that transactions operate properly without interference from concurrently executing database statements. This property is referred to as \_\_\_\_\_.

Answer: Concurrency Control

10. List any two ways of searching in Elasticsearch.

Answer:

- Query DSL (Domain-Specific Language)
- Full-text search

11. What is a checkpoint and when does it occur?

Answer: A checkpoint is a mechanism in database management systems (DBMS) that saves the current state of the database to reduce recovery time in case of failure. It occurs periodically or before significant transactions.

Here are the answers to the questions from the provided image:

## PART - B

1. Define canned transactions. Which database user is responsible for developing specifications for canned transactions?

Answer:

- Canned Transactions are predefined queries that are executed by naive users (end users) without the need for writing SQL commands.
- Database administrators (DBAs) and application developers are responsible for developing specifications for canned transactions.

2. List out all the characteristics of the DBMS approach.

Answer:

- Data Abstraction
- Data Independence
- Efficient Data Access
- Data Integrity and Security
- Multi-User Access
- Transaction Management
- Backup and Recovery Mechanisms

3. Perform the set operations (Union, Intersection, and Minus) on the given relations and display the tuples.

- Union ( $X1 \cup X2, Y1 \cup Y2$ ): Combines all unique tuples from both sets.
- Intersection ( $X1 \cap X2, Y1 \cap Y2$ ): Finds common tuples in both sets.
- Minus ( $X1 - X2, Y1 - Y2$ ): Finds tuples present in  $X1$  but not in  $X2$ .

*(Detailed set operation results will be based on performing these operations on the provided table data.)*

4. Specify the ER notations for the different categories of attributes.

Answer:

- Simple Attribute – Single oval
- Composite Attribute – Oval with multiple sub-ovals
- Derived Attribute – Dashed oval

- **Multivalued Attribute** – Double oval
- **Key Attribute** – Underlined oval

5. For the relation schema  $R(A,B,C,D,E,F,G,H,I)$  with FDs  $F = \{ A \rightarrow BC, B \rightarrow CD, HB \rightarrow I, F \rightarrow H, D \rightarrow F \}$  find  $(AB)^+$ .

**Answer:**

- $(AB)^+ = \{A, B, C, D, E, F, G, H, I\}$

6. Mention the type of constraint specified by join dependency.

**Answer:** **Join Dependency** ensures that a relation is decomposed into multiple smaller relations that can be losslessly joined back. It is a **generalization of Multivalued Dependency (MVD)**.

7. Does Elasticsearch have a schema? Justify your answer.

**Answer:**

- Elasticsearch is **schema-less**, meaning it does not enforce a strict schema like relational databases.
- However, it **uses dynamic mapping** to infer types from the data, which allows flexible indexing.

8. What is a Document-Oriented Database?

**Answer:**

- A **Document-Oriented Database** stores data in semi-structured formats like JSON or BSON.
- It is **schema-less**, meaning different documents in the same collection can have different structures.
- Examples: MongoDB, CouchDB.

9. List out the desirable properties of transactions.

**Answer:** The **ACID** properties:

- **Atomicity** (All or nothing)
- **Consistency** (Valid state before and after transaction)
- **Isolation** (No interference between transactions)
- **Durability** (Changes persist even after failures)

10. Differentiate between serial and non-serial schedules.

**Answer:**

- **Serial Schedule:** Transactions are executed one after another without interleaving.
- **Non-Serial Schedule:** Transactions are interleaved but must follow consistency rules.

11. Why is it desirable to have concurrent execution of multiple transactions?

Answer:

- Increases **system throughput**
- Reduces **waiting time** for transactions
- Improves **CPU and resource utilization**
- Allows **faster response time** for users

12. How does a DBMS detect and resolve deadlocks?

Answer:

- **Detection:** Using **Wait-for Graph (WFG)** or timeout mechanisms.
- **Resolution:**
  - **Deadlock Prevention:** By ordering resource requests or using timeouts.
  - **Deadlock Recovery:** By rolling back one or more transactions to break the cycle.

Let me know if you need any explanations or clarifications! 😊