1. What are Functional Dependencies (FDs)?

A **functional dependency (FD)** is a relationship between two sets of attributes (columns) in a table of a relational database

It means if two rows (tuples) in the table have the same values for the attributes on the left side of the functional dependency, they must also have the same values for the attributes on the right side.

--**X → Y** means that if two tuples have the same values in the **X** columns, they will also have the same values in the **Y** columns.

**2. What is a minimal cover (minimal set) of functional dependencies?**

A minimal cover of a set of functional dependencies is an equivalent set of dependencies that has the following properties:

* Every dependency has a single attribute on the right side.
* No dependency can be removed without changing the closure of the set.
* No attribute can be removed from the left side of any dependency without changing the closure.

Determined set = closure

### 3. ****What are the normal forms and why do we normalize relations?****

Normalization is the process of organizing the attributes and tables of a relational database to minimize redundancy and dependency anomalies.

* **1NF (First Normal Form):** Each attribute must have atomic (indivisible) values. No repeating groups or arrays.
* **2NF (Second Normal Form):** Relation is in 1NF and every non-prime attribute is fully functionally dependent on the primary key. No partial dependencies.
* **3NF (Third Normal Form):** Relation is in 2NF and no transitive dependency exists (non-key attributes do not depend on other non-key attributes).
* **BCNF (Boyce-Codd Normal Form):** Stronger than 3NF; every determinant is a candidate key.

Normalization helps avoid anomalies (insertion, update, deletion) and improves data integrity.

### ****4. What are common database anomalies?****

**Answer:**

* **Insertion Anomaly:** You cannot insert data without the presence of other data.
* **Deletion Anomaly:** Deleting some data causes unintended loss of other data.
* **Update Anomaly:** Changing data in one place requires multiple updates to maintain consistency.

Normalization helps prevent these anomalies.

### ****5. Types of Functional Dependencies:****

**Trivial Functional Dependency:**

-A **trivial FD** is when the **right side** of the dependency is a **subset** of the **left side**.

**Non-trivial Functional Dependency:**

-A **non-trivial FD** is one where the **right side** is **not** a subset of the **left side**.

**Full Functional Dependency:**

-A **full FD** means that **all** the attributes in the left side of the dependency are **required** to determine the right side.

**Partial Functional Dependency:**

-A **partial FD** happens when **only part of the primary key** determines another attribute, instead of the entire primary key.

**Transitive Functional Dependency:**

-A **transitive FD** occurs when one attribute determines another, and the second attribute determines a third.

( **Trivial FD:** X→YX \to YX→Y where Y⊆XY \subseteq XY⊆X. Always holds.

 **Non-trivial FD:** X→YX \to YX→Y where Y⊈XY \not\subseteq XY⊆X.

 **Full FD:** X→YX \to YX→Y and no proper subset of XXX determines YYY (no partial dependency).

 **Partial FD:** An attribute depends on part of a composite key, not the whole.

 **Transitive FD:** X→YX \to YX→Y and Y→ZY \to ZY→Z imply X→ZX \to ZX→Z, where ZZZ is indirectly dependent on XXX.)

### 6. ****What are Armstrong's axioms?****

**Answer:**  
Armstrong's axioms are a set of inference rules to derive all FDs from a given set:

* **Reflexivity:** If Y⊆XY \subseteq XY⊆X, then X→YX \to YX→Y.
* **Augmentation:** If X→YX \to YX→Y, then XZ→YZXZ \to YZXZ→YZ for any ZZZ.
* **Transitivity:** If X→YX \to YX→Y and Y→ZY \to ZY→Z, then X→ZX \to ZX→Z.

These axioms are sound and complete for reasoning about FDs.

### 7. ****What types of triggers exist in databases?****

**Answer:**

* **Row-level triggers:** Fire once for each row affected by an event.
* **Statement-level triggers:** Fire once per SQL statement regardless of how many rows affected.
* **Before triggers:** Execute before the triggering event.
* **After triggers:** Execute after the triggering event.
* **Instead of triggers:** Used on views to execute instead of the triggering event.
* **Event triggers:** Fire on DDL events or database system events, not just DML.

### 8. ****What are the main operations of relational algebra?****

**Answer:**

* **Selection (σ):** Selects rows based on a predicate.
* **Projection (π):** Selects columns.
* **Union (∪):** Combines tuples from two relations.
* **Set Difference (-):** Tuples in one relation but not in another.
* **Cartesian Product (×):** Combines tuples from two relations.
* **Join (⨝):** Combines tuples from two relations based on a condition.

### 9. ****What is the difference between a function and a procedure?****

**Answer:**

* A **function** returns a single value and can be used in SQL expressions.
* A **procedure** performs actions (like insert, update) and does not necessarily return a value. Procedures can return multiple results via OUT parameters.