NAME: ROHITH Y V GWID: G26286080

A) Relational Design Theory: Motivation & Overview

Relational Design Theory aims to improve database efficiency by addressing common data issues such as redundancy (repetitive data), update anomalies (inconsistencies during data modifications), and deletion anomalies (loss of critical data when deleting records). The approach involves breaking down large, complex tables (called mega relations) into smaller, well-organized tables, ensuring no data is lost in the process. Normalization ensures the resulting tables adhere to specific normal forms to avoid these issues. The primary goal is to maintain data integrity, consistency, and efficiency by eliminating anomalies caused by poor database structure.

B) Relational Design Theory: Functional Dependencies

Functional Dependency in database design describes how one attribute (or a set of attributes) uniquely determines another. It helps organize data efficiently and reduce redundancy. A Trivial Functional Dependency occurs when an attribute or set of attributes depends on itself or a part of itself. For instance, if {A, B} is a set, then $AB \rightarrow A$ is trivial because A is already in the left side of the equation. The Closure of Attributes is used to determine all attributes that can be derived from a set of attributes, helping in efficient database design and normalization.

C) Relational Design Theory: Boyce-Codd Normal Form

Boyce-Codd Normal Form (BCNF) is an advanced normalization level designed to eliminate redundancy and anomalies by strictly enforcing functional dependencies. A relation is in BCNF if, for every functional dependency ($X \rightarrow Y$), X is a superkey. BCNF is achieved by decomposing large tables into smaller, well-structured ones, ensuring no loss of data while removing redundancy and inconsistencies. This decomposition ensures data integrity, eliminates anomalies, and results in a more efficient, consistent, and well-organized database.

D) Relational Design Theory: Multivalued Dependencies & 4th Normal Form

Multivalued Dependencies (MVDs) arise when one attribute determines multiple independent values of another attribute. Unlike functional dependencies, MVDs indicate that attributes are independent but still linked to the same key. A Trivial MVD occurs when the dependent attributes are subsets of the original ones. To eliminate redundancy caused by MVDs, Fourth Normal Form (4NF) is applied, ensuring that if an MVD exists, the determining attribute is a superkey. The 4NF decomposition process ensures minimal redundancy and creates a more efficient, consistent, and anomaly-free database structure.

NAME: ROHITH Y V GWID: G26286080

E) Relational Design Theory: Shortcomings of BCNF/4NF

While BCNF and 4NF are effective at eliminating redundancy and anomalies, they can introduce complexities. These challenges include handling dependencies across multiple tables, making queries more complex, and reducing performance due to frequent joins. The normalization process can lead to query performance issues, particularly in large databases or high-traffic environments. Over-decomposition may also occur, where excessive breaking down of tables complicates database management and consistency.

F) Indexes Video:

Indexes in databases enhance data retrieval speed by allowing quick location of specific rows, much like an index in a book helps locate topics fast. They optimize query performance, especially for large datasets or frequently queried columns. However, indexes come with trade-offs, such as extra space usage, the need for regular maintenance, and potential performance overhead during insertions, updates, or deletions. While the space cost is relatively small compared to the benefits, the maintenance effort can offset some of the performance gains.

G) Relational Algebra 1:

Relational Algebra uses operators to manipulate and query relational database data. The Select operator (σ) filters rows based on a condition, retrieving only the records that match specified criteria. The Project operator (π) selects specific columns, reducing the result set. The Cross Product (×), or Cartesian Product, pairs each row from one table with every row from another, producing all possible combinations of rows. The Natural Join (⋈) combines two tables based on matching column values, eliminating duplicate columns from the result.

H) Relational Algebra 2:

Relational Algebra 2 introduces more operators for relational data manipulation:

1. The Union operator (∪) combines the results of two tables, returning all rows from both while removing duplicates.
2. The Difference operator (-) finds rows in one table that do not appear in another.
3. The Intersection operator (∩) returns only the rows that exist in both tables.
4. The Rename operator (ρ) allows the renaming of columns or tables, improving readability, especially when using the same table multiple times in queries.

NAME: ROHITH Y V GWID: G26286080