DBMS Assignment — Unit 5 (Solved)

Prepared by: ChatGPT (answers based on the uploaded assignment).

Fill in the blanks

Q: 1. A ___________ dependency is a constraint between two attributes in a database.

A: functional dependency

Q: 2. A functional dependency of the form A -> A is called a ___________ dependency.

A: trivial dependency

Q: 3. Armstrong's axioms include Reflexivity, Augmentation, and ___________.

A: Transitivity

Q: 4. The closure of a set of functional dependencies is the set of all functional dependencies that can be ___________ from the given set.

A: inferred

Short Questions

1. What is a trivial functional dependency?

A functional dependency X → Y is trivial if Y is a subset of X. For example, RollNo, Name → RollNo is a trivial dependency because RollNo $\subseteq$ {RollNo, Name}.

2. What is a candidate key?

A candidate key is a minimal set of attributes that uniquely identifies a tuple in a relation. There may be multiple candidate keys, one of which becomes the primary key.

3. What is a lossless decomposition?

A decomposition of a relation R into R1 and R2 is lossless if no information is lost during decomposition and join. It ensures that R can be perfectly reconstructed from R1 and R2 using a natural join.

4. How does BCNF differ from 3NF?

3NF allows some redundancy if the non-prime attribute is transitively dependent on a key through a superkey. BCNF is stricter — every determinant must be a candidate key, ensuring minimal redundancy.

1. What is functional dependency? Explain all its types with example.

A **functional dependency (FD)** is a constraint between two sets of attributes in a relation. It states that if two tuples have the same values for attribute(s) X, then they must also have the same values for attribute(s) Y, written as X → Y. **Types of Functional Dependencies:**
1. **Trivial FD:** Y $\subseteq$ X (e.g., RollNo, Name → RollNo).
2. **Non-trivial FD:** Y $\not\subset$ X (e.g., RollNo → Name).
3. **Transitive FD:** If X → Y and Y → Z, then X → Z.
4. **Multivalued Dependency:** A dependency where one attribute determines multiple independent attributes.
Example: In Student(RollNo, Name, Dept), RollNo → Name means Name depends on RollNo.

2. What is normalization in DBMS? Also discuss the need of normalization. Explain 2NF, 3NF, BCNF with suitable example.

Normalization is the process of organizing data in a database to minimize redundancy and improve data integrity. It divides large tables into smaller, related tables and defines relationships among them. **Need for Normalization:**
• To eliminate redundancy
• To prevent anomalies (update, insertion, deletion)
• To improve consistency and efficiency. **2NF (Second Normal Form):** A relation is in 2NF if it is in 1NF and no non-prime attribute is partially dependent on a candidate key.
Example: Student(RollNo, Subject, Marks). Here, RollNo → Name and (RollNo, Subject) → Marks. To remove partial dependency, split into Student(RollNo, Name) and Marks(RollNo, Subject, Marks). **3NF (Third Normal Form):** A relation is in 3NF if it is in 2NF and there is no transitive dependency of non-prime attributes on a key.
Example: Emp(EmpID, DeptID, DeptName). DeptName depends on DeptID, not directly on EmpID, so it's transitive. Decompose into Emp(EmpID, DeptID) and Dept(DeptID, DeptName).
BCNF (Boyce-Codd Normal Form): A relation is in BCNF if every determinant is a candidate key. It removes all redundancy allowed in 3NF.
Example: In R(A,B,C) with FDs A → B and B → A, only BCNF decomposition ensures both A and B are keys.

3. What is the need for Normalization? Explain 1NF, 2NF and BCNF.

Normalization is needed to remove redundancy, prevent anomalies, and maintain data integrity. **1NF:** A table is in 1NF if it contains only atomic (indivisible) values and each column contains values of a single type. Example: Repeating groups are removed. **2NF:** A table is in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on the primary key.
BCNF: A table is in BCNF if for every functional dependency X → Y, X is a superkey. It removes all anomalies possible in 3NF.

4. What is closure of a set of attributes? Explain how to find closure of a set of attributes.

The **closure** of an attribute set X, denoted X■, is the set of all attributes that can be functionally determined from X using a given set of functional dependencies. **Steps to find closure:**
1. Start with X■ = X.
2. For each functional dependency Y → Z, if Y ⊆ X■, then add Z to X■.
3. Repeat until no new attributes can be added.
Example: R(A, B, C, D), FDs = {A → B, B → C}. Then A■ = {A, B, C}.

5. What is meant by normalization? Write its need. List and discuss database anomaly during database design.

Normalization is the process of structuring a database to reduce redundancy and improve integrity. **Need:** To remove duplication and anomalies. **Database anomalies:**
• **Insertion anomaly:** Cannot insert data due to missing related data.
• **Deletion anomaly:** Deleting a record unintentionally removes useful data.
• **Update anomaly:** Inconsistency due to data redundancy when updating records.

6. List and explain Armstrong's axioms.

Armstrong’s axioms are a set of inference rules used to derive all possible functional dependencies from a given set: 1. **Reflexivity:** If $Y \subseteq X$, then $X \rightarrow Y$. 2. **Augmentation:** If $X \rightarrow Y$, then $XZ \rightarrow YZ$. 3. **Transitivity:** If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$. **Additional derived rules:**

- **Union:** If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$.
- **Decomposition:** If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$.
- **Pseudotransitivity:** If $X \rightarrow Y$ and $YZ \rightarrow W$, then $XZ \rightarrow W$.

7. What is an anomaly in database design? How it can be solved.

An **anomaly** is an inconsistency or problem in database operations caused by data redundancy. • **Update anomaly:** Multiple updates required to maintain consistency. • **Insertion anomaly:** Inability to add new data due to dependency on other data. • **Deletion anomaly:** Loss of data when deleting unrelated records. **Solution:** Anomalies can be resolved through normalization (1NF, 2NF, 3NF, BCNF), which ensures that data is stored in well-structured, non-redundant tables.

References:

Source: Assignment Unit 5 uploaded by user.