

Chapter Review - 14

Name: Pinnimti, Sri Harish

Student ID: 1001865949

Net ID: sxp5949

Q1:

Sol.

NULL should be avoided because:

- A column with NULL values can't have PRIMARY KEY constraints.
- The storage level Memory space will be wasted.
- When aggregate operations such as SUM, COUNT are performed on the null value attribute, the result will be incorrect
- The result of JOIN operations with null value attribute may be unpredictable

Spurious Tuples:

An improper decomposition of the base table leads to Spurious tuples. When 2 relations are combined using JOIN operation with poor conditions the resultant relation will have more tuples than the original set of tuples these extra tuples are Spurious tuples.

Avoiding Spurious Tuples:

Spurious tuples can be avoided by taking care while designing relation schemas. Always implement JOIN operations on the relation that has common attributes such that the selected attribute is a primary key in one relation and a foreign key in another relation.

|

Q2:

Sol.

First Normal Form: A relation is said to obey First Normal Form if every column in the table is uniquely defined. For each set of related data, separate tables must be created and each table must be identified with a unique column called the primary key.

Second Normal Form: A relation is said to be in Second Normal Form if the relation obeys 1st Normal Form and every non-prime attribute is fully functionally dependent on Primary Key.

3NF: A relation is said to be in Third Normal Form if the relation obeys 2nd Normal Form and no non-prime attribute in relation is transitively dependent on the primary key.

Q3:

Sol. The Second Normal Form removes all partial dependencies of nonprime attributes A in Relation R on the key. It also ensures that all non-prime attributes are fully functionally dependent on the key of R.

Q4:

Sol. The Third Normal Form eliminates all transitive dependencies on a key of Relation R and guarantees that no non-prime attribute is transitively reliant on it.

Q5:

Sol.

A	B	C	Tuple#
10	b1	c1	1
10	b2	c2	2
11	b4	c1	3
12	b3	c4	4
13	b1	c1	5
14	b3	c4	6

The only possible function dependency is B on C because for every repeating value of tuple in B the values of C are the same. Ex: b1 always corresponds to c1, b3 always corresponds to c4. However, this does not hold true for other tuple relations.

Q6:

Sol:

Checking for 1 NF: According to the definition of 1NF, the relational R should contain only atomic values. Since there will be no repeating groups in the relation R, it is safe to say that the given relation is in 1NF.

Checking for 2 NF: According to the definition of 2NF, the relation R should be in 1NF and each non-key must depend on the primary key and should not have any partial dependency. Although relation R is in 1NF, its non-key is partially dependent on the key attribute **commission%** **partially dependent on Salesperson#** . So, the relation will not be in 2NF.

Checking for 3 NF: According to the definition of 3NF, the relation R should be in 2NF and the non-key attribute should not be functionally dependent on other non-key attributes. However, the given relation is not in 2NF and the non-key attribute **Date_sold** is functionally dependent on **Discount_amt** which is again a non-key attribute given relation is not in 3NF.

Updating relation R:

For 2NF:

CAR_SALES (Car_id, Date_sold, Salesperson_id, Discount_amt)

SALES_COMM (Salesperson_id, Commission_percent)

For 3NF:

DATE_DIS(Date_sold, discount_amt)

CAR_SALE_DATE (car_id, salesperson_id, Date_sold)

SALES_COMM (Salesperson_id, Commission_percent)