Database Systems
ICT1407

Revision & Practice Question

Assignment 01

ICT 17/18/017

G.G.P.S.Gunathilaka

1. Discuss insertion, deletion, and modification anomalies. Why are they considered bad? Illustrate with examples.

EMPLOYEE

Ename	Ssn	Bdate	Address	Dnumber
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5
Zelaya, Alicia J.	453453453	1972-07-31	5631 Rice, Houston, TX	4
Wallace, Jennifer S.	987654321	1969-03-29	980 Dallas, Houston, TX	4
Narayan, Ramesh K.	666884444	1965-01-09	450 Stone, Houston, TX	5
English, Joyce A.	987987987	1955-12-08	731 Fondren, Houston, TX	5
Jabbar, Ahmad V.	888665555	1972-07-31	638 Voss, Houston, TX	4
Borg, James E.	123456789	1969-03-29	5631 Rice, Houston, TX	1

DEPARTMENT

Dname	Dnumber	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATION

Dnumber	Dlocation
1	Houston
4	Bellaire
5	Stafford
5	Houston
5	Sugarland

- If we cannot enter data for a certain attributes without entering data for another attribute of a database, called as insertion anomalies.
- In above example, if we want to add new tuple for employee who work for a certain department we must enter all the attribute values of that department.
- If we delete some data of an attribute it may case to delete data of another attribute. This is called as deletion anomalies.
- In above example, if we want to delete an employee tuple, we could not do that because of the department details that employee works for will lost.
- If we change value of a particular attribute of an entity, we should update the all attributes which related with that attribute.
- In department entity, if we change the value of some attributes of a department, we must update the all tuple of employee who works for that department. Else, details of database will incorrect.

So these three anomalies are considered bad because,

- These anomalies are undesirable.
- It is difficult to maintain consistency of data in the database.
- It causes unnecessary updates of data.
- Waste the memory space.

2. What is a functional dependency? What are the possible sources of the information that defines the functional dependencies that hold among the attributes of a relation schema?

    - A functional dependency is a constraint between two set of attributes of a database.
    - The functional dependency of X and Y is represented by $X \longrightarrow Y$.
    - It is helps us to maintain the quality of data in the database.

3. What does the term un*normalized relation* refer to? How did the normal forms develop historically from first normal form up to Boyce-Codd normal form?
    - To develop normal forms historically from first normal form up to Codd normal form,
        - First normal form will remove repeating columns, it should have single valued attributes.
        - Second normal form will remove sub set of data. It should be in the first normal form.
        - In Third normal form all columns should depend directly upon primary key and it doesn't have transitive dependency.
        - In fourth normal form multi valued dependency should be removed.
        - In Boyce and Codd normal form every determinant should be a candidate key.

4. Define first, second, and third normal forms when only primary keys are considered. How do the general definitions of 2NF and 3NF, which consider all keys of a relation, differ from those that consider only primary keys?
        - Fist Normal form – Relation should have no multivalued attributes or nested relations.
        - Second Normal Form - A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.
        - Third Normal Form - According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.
        - General definition of 2NF and 3NF that take all candidate keys of a relation into account.
        - An attribute that is part of any candidate key will be considered as prime. Partial and full functional dependencies and transitive dependencies will now be considered with respect to all candidate keys of a relation.

5. What undesirable dependencies are avoided when a relation is in 2NF?
    - Partial dependencies

6. What undesirable dependencies are avoided when a relation is in 3NF?
    - Transitive dependencies

7. In what way do the generalized definition of 2NF and 3NF extend the definitions beyond primary key?
    - Second normal form definition - A relation schema *R* is in second normal form (2NF) if every Non prime attribute *A* in *R* is not partially dependent on any key of R
    - Third normal form – A relation schema R is in third normal form (3NF) if, whenever a nontrivial functional dependency X → A holds in R, either (a) X is a super key of R, or (b) A is a prime attribute of R.
    - The generalized definitions of second normal form and third normal form extend beyond primary key by taking in to consideration all the candidate keys of a relation.
    - These definitions take into consideration all the attributes that can be a possible key of a relation and also consider the partial and transitive dependencies on the candidate keys.

8. Define Boyce-Codd normal form. How does it differ from 3NF? Why is it considered a stronger form of 3NF?
    - A relation is said to be in BCNF if and only if every determinate is a candidate key.
    - A relation schema R is in BCNF if whenever a nontrivial functional dependency X → A holds in R, then X is a super key of R.
    - Boyce-Codd normal form stronger than third normal form.

      Because,

    - Third normal form allows functional dependency xy , y (Right hand side) must be prime attribute.
    - Third Normal Form allows non key attributes as determinates but Boyce-Codd normal form doesn't allows.