CS166 Database Management

Normalization Worksheet

Let us review some concepts

- We discussed several normal forms, which includes 1NF, 2NF, 3NF, and Boyce-Codd Normal Form (BCNF).

- These normal forms have increasingly restrictive requirements, i.e. every relation in BCNF is also in 3NF, every relation in 3NF is also in 2NF, and every relation in 2NF is in INF.

- *Functional dependencies* (FDs) are **constraints** that are derived from the *meaning* and *relationships* of the attributes. *i.e.* A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y.

- Prime attribute are those **attributes** which are present in the candidate key of that table. While **Non** -primary **attributes** are those that are not there in the candidate key.

- Candidate key is a set of attributes (or attribute) which uniquely identify the tuples in relation or table.

- **Super Key** is an attribute (or set of attributes) that is used to uniquely identifies all attributes in a relation. **Candidate Key** is a proper subset of a **super key.**

  ssn, name - candidate key
  ssn - super key

1NF: A relation is in 1NF if each attribute is atomic (simple structure, i.e., a string, integer, double, boolean, etc.). It disallows multi-valued and composite attributes.

2NF: A relation is in 2NF if it is in 1NF and has no partial dependencies.

A dependency **A** → **B** is a *partial dependency* if

- B is a not subset of A
- A is a proper subset of some key for the relation
- B is not part of any candidate key of the table (relation) .

Essentially, a table is not in 2NF if an attribute is partially dependent on a subset of a candidate key.

3NF: A relation schema <R, F>, where R is the relational schema and F is the set of functional dependencies for this schema, is in 3NF if it is in 2NF and has no transitive dependencies.

- A *transitive dependency* is a functional dependency on a non-primary key attribute.
- Essentially, for a relation to be in 3NF
  - For any dependency X → A at least one of the following holds:
    - ***X*** is **super key**, ***or***
    - ***A*** is **key attribute, i.e. A-X is contained in a key**

Boyce-Codd Normal Form (BCNF) – R is in BCNF if for every FD X → A in F, either

- A is a subset of X(trivial dependency) or
- X is a superkey

Consider the relation schema R = ABCDE and the following functional dependencies on R:
A → D
BC → E
D → AB

Provide the two candidate keys of the relation R :

_____CD_____ &. _____CA_____

Consider the relation schema S = ABCD and the following functional dependencies on S:
A → BCD
B → C
CD → A

Consider the decomposition of S into S1 = ABC and S2 = BCD.

- Is this a valid decomposition into BCNF? Explain. No, because B C and B is not a candidate key (for either S1 and S2)

- Is the decomposition of S into S1 and S2 lossless? Why or why not? No, because BC is the intersection and BC is not a key for either S1 and S2

- Is the decomposition of S into S1 and S2 dependency preserving? Why or why not? No, because CDA (or AC) are lost

Consider the decomposition of S into S3 = ABD and S4 = BC.

- Is this a valid decomposition into BCNF? Explain. Yes, A is a key for ABD and B is a key for BC (also all 2 attr relns are in BCNF)

- Is the decomposition of S into S3 and S4 lossless? Why or why not? Yes, Intersection is B and it is a key for BC

- Is the decomposition of S into S3 and S4 dependency preserving? Why or why not? No, because CDA (or AC) are lost ;

Consider relation R (ABCDE) with functional dependencies (FDs): S = {A → B ; A → D ; B → C ; BC → D ; BE → A}

What are the candidate keys of R? The sets attributes that have all attributes of R in their closure are AE and BE. These are the candidate keys of R under S.

Compute T, the minimum basis of the set of FDs in S.

Answer T = {A → B ; B → C ; B → D ; BE → A}

For each FD in {A → B ; B → C ; B → D ; BE → A}, state whether or not it violates BCNF and 3NF. Clearly explain your answer for each Functional Dependency

A → B : violates BCNF since A is not a candidate key or a superkey and the FD is not trivial; this FD does not violate 3NF, since B is part of the candidate key BE

B → C: violates both BCNF and 3NF, since B is not a candidate key or a superkey, the FD is not trivial, and D is not part of any candidate key

B → D: violates both BCNF and 3NF, since B is not a candidate key or a superkey, the FD is not trivial, and D is not part of any candidate key

BE → A: does not violate BCNF, and therefore also does not violate 3NF since BE is a candidate key of R