NORMALIZATION

* Normalization of data is the process of analyzing the given relation schemas based on their FDs and primary keys to achieve the properties of normalization:
  1) minimizing redundancy of data
  2) Minimizing the insertion, deletion and update anomalies (Modification Anomalies)

* The normalization process takes a relation schema and applies a series of tests to "certify" whether it satisfies a certain normal form.

* The relation schemas that do not satisfy the normal form tests are decomposed into smaller relation schemas that meet the tests and possess the properties of normalization.

* The normal form of a relation refers to the highest normal form condition that it meets, which indicates the degree to which the relation is normalized.

* The process of normalization through decomposition must also confirm the existence of additional properties that the relation schemas should possess after decomposition:

Properties of Decomposition

1) Lossless Join / Nonadditive Join Property
   Decomposition

- Let R be a relation schema and let F be a set of FDs on R. Suppose, R is decomposed into relations R1 and R2 and r(R) be a relation instance with schema R

- The decomposition is a lossless decomposition if the following condition holds:

$$\boxed{\Pi_{R1}(r) \bowtie \Pi_{R2}(r) = r}$$

i.e. If we project r onto R1 and R2 and compute the natural join of the projection results, we get back exactly r.

* The decomposition that is not lossless is called a lossy decomposition.

Test for Lossless Join Decomposition

Let R be decomposed into R1 and R2. The decomposition is lossless if:

$$\boxed{\begin{array}{l}\text{(i)} \quad R1 \cap R2 \rightarrow R1 - R2 \text{ is in } F^+ \text{ or} \\ \text{(ii)} \quad R1 \cap R2 \rightarrow R2 - R1 \text{ is in } F^+\end{array}}$$

Example

Consider the relation schema $R(A, B, C, D)$.
The set of FDs $F = \{AB \rightarrow C, C \rightarrow D\}$.

Sol^n: suppose R is decomposed into relation schemas ~~R1(A, BC~~ : $R1(A, B, C)$ and $R2(C, D)$ based on the FDS $AB \rightarrow C$ and $C \rightarrow D$.

closure of F: $F^+ = \{AB \rightarrow C, C \rightarrow D, AB \rightarrow D\}$

$R1 \cap R2 = C$

$R1 - R2 = AB$, $R2 - R1 = D$

$C \rightarrow AB$ is not present in F^+ but $C \rightarrow D$ is present in F^+. $(R1 \cap R2 \rightarrow R2 - R1)$

$\therefore$ The decomposition is lossless.

2) Dependency - Preserving Decomposition

* We consider a relation schema R with the set of FDs represented as F.

* It would be useful if each FD $X \rightarrow Y$ specified in F either appears directly in one of the decomposed relation R_i of R or could be inferred from the dependencies that appear in some R_i. This is known as the dependency preservation condition.

* The dependencies needs to be preserved since each dependency in F represents a constraint on the database

* If one of the dependencies is not present in any of the decomposed relations R_i, this constraint cannot be enforced.

Formal Definition

R: original database schema

* Given a set of dependencies F on R, the projection of F on R_i, denoted by $\pi_{R_i}(F)$ where R_i is a subset schema of R, is the set of functional dependencies $X \rightarrow Y$ in F^+ such that X and Y are attributes in the decomposed relation schema R_i.

$$\boxed{F_i = \pi_{R_i}(F^+) = \{X \rightarrow Y \mid \{X, Y\} \subseteq R_i \text{ and } X \rightarrow Y \in F^+\}}$$

where F_i : Projection of the FD set F onto R_i

* A decomposition $D = \{R1, R2, \ldots, Rm\}$ of R is dependency-preserving with respect to F if the union of projections of F on each Ri in D is equivalent to F.

$$\boxed{((\pi_{R_1}(F)) \cup \ldots \cup (\pi_{Rm}(F)))^+ = F^+ \quad \text{or} \quad (F1 \cup \ldots Fm)^+ = F^+}$$

* If a decomposition is not dependency-preserving, some dependency is lost in the decomposition.

Example 1

Consider a relation schema R(City, Street, Zip-Code) = R(C, S, Z)

and the set of FDs $F = \{CS \to Z, Z \to C\}$

Test for lossless join decomposition and dependency-preserving. for the decomposition R1(S, Z) and R2(C, Z)

Proof:

Test for lossless join decomposition

$$F^+ = \{\underbrace{\cancel{C \to C}, \cancel{S \to S}, \cancel{Z \to Z}}_{\text{Trivial FDs}}, CS \to Z, Z \to C, CS \to C\}$$

$R1 \cap R2 = Z$

$R1 - R2 = S$

$Z \to S$ is not in F^+.

$R2 - R1 = C$

$Z \to C$ is is F^+.

So, the decomposition is ~~in~~ lossless.

Test

Test for Dependency Preservation

We first find the projections of F on the decomposed relation schemas R1 and R2 denoted as F1 and F2. $R1(S, Z)$, $R2(C, Z)$

$$F1 = \Pi_{R1}(F^+) = \{S \to S,\ Z \to Z\}$$

$$F2 = \Pi_{R2}(F^+) = \{C \to C,\ Z \to Z,\ Z \to C\}$$

$$F1 \cup F2 = \{S \to S,\ Z \to Z,\ C \to C,\ Z \to C\}$$

$$(F1 \cup F2)^+ = \{S \to S,\ Z \to Z,\ C \to C,\ Z \to C\} \neq F^+$$

$$F^+ = \{C \to C,\ S \to S,\ Z \to Z,\ CS \to Z,\ Z \to C,\ CS \to C\}$$

$CS \to Z$ cannot be derived in this decomposition.
So, it is not dependency preserving.

Example 2 Consider a relation schema $R(A, B, C, D)$ with FD $F = \{AB \to C,\ C \to D\}$. R is decomposed as $R1(A, B, C)$ and $R2(C, D)$. Test whether the decomposition is dependency preserving?

trivial FDs

Proof:

$$F^+ = \{AB \to C,\ C \to D,\ A \to A,\ B \to B,\ C \to C,\ D \to D,\ AB \to AB,\ AB \to D\}$$

$$F1 = \Pi_{R1}(F) = \cancel{A \to A,\ B \to B},\ \{AB \to C,\ A \to A,\ B \to B,\ C \to C,\ AB \to AB\}$$

$$F2 = \Pi_{R2}(F) = \{C \to C,\ D \to D,\ C \to D\}$$

$$F1 \cup F2 = \{A \to A,\ B \to B,\ C \to C,\ D \to D,\ AB \to AB,\ AB \to C,\ C \to D\}$$

Trivial FDs

$$(F1 \cup F2)^+ = \{A \to A,\ B \to B,\ C \to C,\ AB \to AB,$$
$$D \to D,\ AB \to C,\ C \to D,\ AB \to D\}$$
$$= F^+$$

So, the decomposition is dependency preserving.