# ST.XAVIER’S COLLEGE

# MAITIGHAR, KATHMANDU

****

**ASSIGNMENT #10**

**Database Management System**

**Submitted By:**

Ronak Agrawal

013BSCCSIT033

4th Sem, 2nd Year

**Submitted To:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Er. Sanjay Kumar Yadav

Department of Computer Science

Lecturer

**Functional Dependencies:**

Functional dependency (FD) is a set of constraints between two attributes in a relation. Functional dependency says that if two tuples have same values for attributes A1, A2,..., An, then those two tuples must have to have same values for attributes B1, B2, ..., Bn [1].

Functional dependency is represented by an arrow sign (→) that is, X→Y, where X functionally determines Y. The left-hand side attributes determine thne values of attributes on the right-hand side [1].

**E.g.:** In table listing employee characteristics including Social Security Number (SSN) and name, it can be said that name is functionally dependent upon SSN (or SSN -> name) because an employee's name can be uniquely determined from their SSN.

However, the reverse statement (name -> SSN) is not true because more than one employee can have the same name but different SSNs [2].

**Closure of a set of Functional Dependencies:**

* Given a set F set of functional dependencies, there are certain other functional dependencies that are logically implied by F [3].
* The set of all functional dependencies logically implied by F is the closure of F.
* We denote the closure of F by F+.
* We can find all of F+ by applying Armstrong’s Axioms [3]:
* if β ⊆ α, then α → β (reflexivity)
* if α → β, then γα → γβ (augmentation)
* if α → β and β → γ, then α → γ (transitivity)
* These rules are sound and complete.
* We can further simplify computation of F+ by using the following additional rules.
* If α → β holds and α → γ holds, then α → βγ holds (union)
* If α → βγ holds, then α → β holds and α → γ holds (decomposition)
* If α → β holds and γβ → δ holds, then αγ → δ holds (pseudotransitivity)
* The above rules can be inferred from Armstrong’s axioms.

**Example:**

* R = (A, B, C, G, H, I)
* F = {A → B, A → C, CG → H, CG → I, B → H}
* some members of F+ are A → H, AG → I, CG → HI

**Closure of Attribute sets:**

Given a set α of attributes of R and a set of functional dependencies F, we need a way to find all of the attributes of R that are functionally determined by α. This set of attributes is called the **closure of** α **under F** and is denoted α+. Finding α+ is useful because:

* if α+ = R, then α is a super key for R
* if we find α+ for all α⊆ R, we've computed F+ (except that we'd need to use decomposition to get all of it).
* An algorithm for computing α+:

result := α

repeat

temp := result

for each functional dependency β → γ in F do

if β ⊆ result then

result := result ∪ γ

until temp = result

**Decomposition:**

* Decomposition in database means breaking tables down into multiple tables
* From Database perspective means going to a higher normal form

**Two Characteristics of Good Decompositions**

1. **Lossless- Join Decomposition:**

* Let *R* be a relation schema.
* Let *F* be a set of functional dependencies on *R*.
* Let and form a decomposition of *R*.
* The decomposition is a lossless-join decomposition of *R* if at least one of the following functional dependencies are in *F+*

R1 ∩ R2 🡪 R1

R1 ∩ R2 🡪 R2

If R is split into R1 and R2, for the decomposition to be lossless then **at least** **one** of the two should hold true.

1. **Dependency Preservation:**

A decomposition D = {R1, …, Rm} of R is dependency-preserving with respect to a set F of FDs if (F1 ∪ … ∪ Fm)+ = F+, Where Fi means the projection of the dependency set F onto Ri.

Fi =Π Ri(F+) denotes a set of FDs X → Y in F+ such that all attributes in X ∪ Y are contained in Ri: Fi=Π Ri(F+) ={ X→Y| {X,Y}⊆ Ri and X→Y ∈ F+ }

We do not want FDs to be lost in the decomposition.

Always possible to have a dependency-preserving decomposition D such that each Ri in D is in 3NF. Not always possible to find decomposition that preserves dependencies into BCNF [5].

**References:**

1. DBMS Normalization, www.tutorialspoint.com

Url: <http://www.tutorialspoint.com/dbms/database_normalization.htm> [8/10/2015]

1. Wondering What "Functional Dependency" Is in Programming? Here's the Answer, About.com Tech

Url: <http://databases.about.com/cs/specificproducts/g/functdep.htm> [8/10/2015]

1. Url:[http://www.iai.unibonn.de/III/lehre/vorlesungen/Informationssysteme/SS01/9.1\_ vorlesung.pdf](http://www.iai.unibonn.de/III/lehre/vorlesungen/Informationssysteme/SS01/9.1_%20vorlesung.pdf) [8/10/2015]
2. <Url:http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter6/node13.html> [8/10/2015]
3. Url: <http://www.sztaki.hu/~fodroczi/dbs/dep-pres-own.pdf> [8/10/2015]