ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**Database Management System**

**Theory Assignment**

**SUBMITTED BY:**

Sneha Prasai

013BSCCSIT040

**SUBMITTED TO**

|  |  |
| --- | --- |
| **Er. Sanjay Kr. Yadav**  **( Lecturer )** |  |
| **Department of Computer Science** | |

1. GRANT and REVOKE authorization.
2. Data encryption
3. **Transivity, Reflexivity, and Augmentation properties of FDs**

Given that *X*, *Y*, and *Z* are sets of attributes in a relation *R*, one can derive several properties of functional dependencies. Among the most important are the following, usually called [Armstrong's axioms](https://en.wikipedia.org/wiki/Armstrong%27s_axioms):

* **Reflexivity**: If *Y* is a subset of *X*, then *X* → *Y*
* **Augmentation**: If *X* → *Y*, then *XZ* → *YZ*
* **Transitivity**: If *X* → *Y* and *Y* → *Z*, then *X* → *Z*

"Reflexivity" can be weakened to just X \rightarrow \varnothing, i.e. it is an actual [axiom](https://en.wikipedia.org/wiki/Axiom), where the other two are proper [inference rules](https://en.wikipedia.org/wiki/Inference_rules), more precisely giving rise to the following rules of syntactic consequence:

\vdash X \rightarrow \varnothing  
X \rightarrow Y \vdash XZ \rightarrow YZ  
X \rightarrow Y, Y \rightarrow Z \vdash X \rightarrow Z.

These three rules are a [sound](https://en.wikipedia.org/wiki/Soundness) and [complete](https://en.wikipedia.org/wiki/Completeness_(logic)) axiomatization of functional dependencies. This axiomatization is sometimes described as finite because the number of inference rules is finite, with the caveat that the axiom and rules of inference are all [schemata](https://en.wikipedia.org/wiki/Schema_(logic)), meaning that the *X*, *Y* and *Z* range over all ground terms (attribute sets).

1. **BCNF and decomposition into BCNF**

BCNF:

We say a relation R is in BCNF if whenever X → Y is a nontrivial FD that holds in R, X is a superkey. Remember: nontrivial means Y is not contained in X. Remember, a superkey is any superset of a key (not necessarily a proper superset)

Example:

Drinkers(name, addr, beersLiked, manf, favBeer)

FD’s: name → addr favBeer, beersLiked → manf

Only key is {name, beersLiked}

In each FD, the left side is not a superkey

Any one of these FD’s shows Drinkers is not in BCNF

Decomposition into BCNF:

Given: relation R with FD’s F § Look among the given FD’s for a BCNF violation X → Y

If any FD following from F violates BCNF, then there will surely be an FD in F itself that violates BCNF

Compute X+

Not all attributes, or else X is a superkey

1. **Characteristics schedules based on Recoverability**
2. Characteristics schedules based on Serializabiliy
3. Transactions supports in SQL