DATABASE SYSTEMS

Classroom Log – 8

Harsha Raj
IMT2014023

FUNCTIONAL DEPENDENCIES

Background

- Functional dependencies: a formal mechanism to evaluate the “goodness” of a relation schema
- Goodness of a physical schema is a measure of search and update complexities

Measures of Goodness

- Semantics of attributes
- Redundant values in tuples
- Null values in tuples
- Possibility of generating spurious tuples

Semantics of a Relation

Emp ID	Name	Dep ID	Dep Name	No of hours worked

Work done by an employee in a department

Emp ID	Name

Dep ID	Dep Name

Emp ID	**Dep ID**	No of hours worked

Here the above schema has simpler semantics

Redundant information

Emp ID	Name	Department
2	A	L
3	B	L
4	C	M
3	B	M
1	C	N

- Wastes space
- Creates *update* and *insertion* and *deletion anomalies*.

If B changes their name, then 2 tuples have to be modified.

If the employee with employee id 1 decides to work on another department, their name has to be entered into the database again.

If employee id 1 is deleted, then the department name "N" is lost.

Null Values

RollNo	Core I Marks	Core II Marks	Elective I Marks	Elective II Marks	Elective III Marks

If a student has taken only two electives, then the third elective will contain a *null* value.

Null values are different from missing values and 0.

Spurious Tuples

Project

Proj_id	Proj_Name	Faculty	RA
21	Distributed File System	Prof. Arindam Roy	Sameer
22	Intelligent WiFi	Prof. Anupam Joshi	Swati Reddy
25	Autonomic databses	Prof. Dwarakish	Vivekanad Pai

Course

Course_id	Course_Name	Faculty	TA
IT800	Wireless Ad Hoc Systems	Prof. Anupam Joshi	Swati Reddy
IT803	File System Design	Prof. Arindam Roy	Sushma Sharma
IT805	Adaptive Systems	Prof. Dwarakish	Vivekanad Pai

$\pi_{\text{Course\_Name,RA}}$ (Project $\bowtie$ Course)

Course_Name	RA
Wireless Ad Hoc System	Swati Reddy
File System Design	**Sameer**
Adaptive Systems	Vivekanad Pai

Second tuple in the result relation is a spurious tuple. Sameer does not work as TA for File System Design course.

The source for Spurious tuple is the common *non-key* field called `Faculty` between the two relations.

Informal Design Principles

- Should not have confounding semantics
- Minimize insertion, update and delete anomalies
- Avoid table design that may result in null values
- Bad design whenever two or more tables which can be joined have a non-key attribute in common

Functional Dependency

Let $R(A_1,A_2, \ldots,A_n)$ be a relation and let X and Y be any subset of attributes of R.
The set of attributes X is said to *functionally determine* the set of attributes Y , (denoted as $X \rightarrow Y$) if for any tuples t_1, t_2 Є R

$$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y].$$

- X and Y need not be disjoint
- If Y is the set of all attributes of R, then X is a *candidate key* of R

Example:

Let Student (RollNo, Name, Course id, CourseName, Grade) denote a student record. Some of the FDs in this relation are:

{RollNo} ⟶ {Name}
{Course id} ⟶ {CourseName}
{RollNo,Course id} ⟶ {Grade}
{RollNo,Course id} ⟶ {Name}
{RollNo,Course id} ⟶ {CourseName}
{RollNo, Name, Course id} ⟶ {Grade}

'good minimal superkey' ⟶ {RollNo,Course id}

- FD is a property of the semantics of the relation schema (intention)
- Not a particular legal relation state (extension)
- FD cannot be *inferred* by an algorithm that looks at a relation schema
- Has to be *specified* by someone who knows the semantics of the relation.

Properties of FDs

While FDs are intentional, they are also *deductive*.

Given a relation R and an initial set of FDs F, it is possible to *infer* more FDs from the elements of F. The set F along with the set of all inferred FDs is called the *closure* of F denoted by F^+.

Two sets of FDs F and G are given.
How to determine if F and G are the same set of FDs?
If F^+ and G^+ are equal then F and G are the same set of FDs.

Rules of FD inference:

1. Reflexive rule: $Y \subseteq X \Rightarrow X \rightarrow Y$, $X \rightarrow X$
2. Augmentation rule: $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$;
   $X \rightarrow Y \Rightarrow XZ \rightarrow Y$
3. Transitivity rule: $X \rightarrow Y; Y \rightarrow Z \Rightarrow X \rightarrow Z$
4. Decomposition or projective rule: $X \rightarrow YZ \Rightarrow X \rightarrow Y$
5. Union or additive rule: $X \rightarrow Y; X \rightarrow Z \Rightarrow X \rightarrow YZ$
6. Pseudotransitive rule: $X \rightarrow Y; WY \rightarrow Z \Rightarrow WX \rightarrow Z$

THANK YOU