IE301

Analysis and Design of Data Systems

Lecture 15

Functional Dependencies and Normalization Theory

Aram Keryan

November 2, 2015

Functional Dependency

Let's think of the whole database as being described by a single **universal** relation schema $R = \{A_1, A_2, \ldots, A_n\}$.

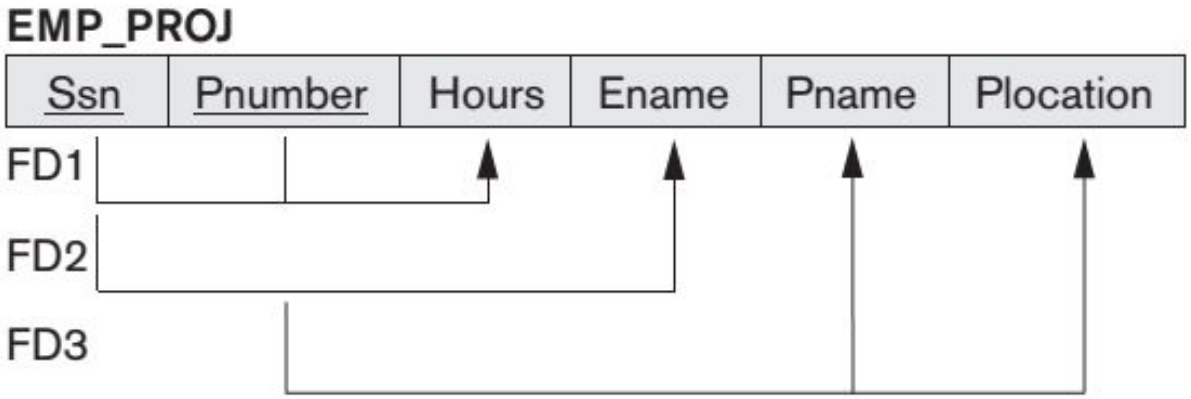
Let's X and Y are subsets of R.

Definition: A functional dependency, denoted $X \rightarrow Y$, on a relation R, specifies a constraint of the form "If two tuples of R agree on all of the attributes of X, then they also agree on all of the attributes of Y".

If $X \rightarrow Y$, then we say that Y is functionally dependent on X, or X functionally determine Y; We also may say that there is a functional dependency from X to Y.

✔ The abbreviation for functional dependency is **FD** or **f.d.**

Example



a. Ssn → Ename

b. Pnumber →{Pname, Plocation}

c. {Ssn, Pnumber} → Hours

Functional Dependency (meaning)

- A functional dependency is a property of the **semantics** or **meaning of the attributes**.

- The database designers will use their understanding of the semantics (meaning) of the attributes of *R*—that is, how they relate to one another—to specify the functional dependencies that should hold on *all* relation states of *R*.

The main use of functional dependencies is to describe further a relation schema *R* by specifying constraints on its attributes that must hold *at all times.*

FD is a property of a Relation

- A functional dependency is a *property of the relation schema R*, not of a particular instance of *R*. Therefore, an FD must be defined explicitly by someone who knows the semantics of the attributes of *R*

- One cannot determine which FDs hold and which do not unless the meaning of and the relationships among the attributes are clearly known and understood

TEACH

Teacher	Course	Text
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

TEACH

Teacher	Course	Text
Smith	Data Structures	Bartram
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

FD is a property of a Relation

- A functional dependency is a *property of the relation schema R*, not of a particular instance of *R*. Therefore, an FD must be defined explicitly by someone who knows the semantics of the attributes of *R*

- One cannot determine which FDs hold and which do not unless the meaning of and the relationships among the attributes are clearly known and understood

TEACH

Teacher	Course	Text
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

Other Properties of FDs

Let's

- $R = \{A_1, A_2, \dots, A_n\}$ is a relation schema
- Let's X and Y are subsets of R
- Let's $X \rightarrow Y$

We can write $X \rightarrow Y$ as $\{X_1, X_2, \dots, X_n\} \rightarrow \{Y_1, Y_2, \dots, Y_m\}$ that is equivalent to a set of FD's:

$$\begin{cases} \{X_1, X_2, \dots, X_n\} \rightarrow Y_1 \\ \{X_1, X_2, \dots, X_n\} \rightarrow Y_2 \\ \cdots \\ \{X_1, X_2, \dots, X_n\} \rightarrow Y_m \end{cases}$$

Normal Forms based of PK's

Motivation

Normalization can be considered a process of analyzing the given relation schemas based on their FDs and primary keys to achieve the desirable properties of :

(1) minimizing redundancy and
(2) minimizing the insertion, deletion, and update anomalies

In other words **Normalization** is a process to make the design have successively better quality.

If relations doesn't meet certain conditions (normal form tests) they are decomposed into 'smaller' relations schemas that meet the tests hence possess desirable properties.

FD is a generalization of a Key

Let's
- $R = \{A_1, A_2, \ldots, A_n\}$ is a relation schema
- Let's X is a subset of R.

We say a set of one or more attributes X is a **superkey** for R if attributes of X functionally determine all other attributes of R;

We say a superkey X is a **key** for R if no proper subset of X functionally determines all other attributes of R; i.e. a key must be minimal.

If a relation schema has more than one key, each is called a **candidate key**

One of the candidate keys is *arbitrarily* designated to be the **primary key**, and the others are called **secondary keys**

An attribute of *R* is called a **prime attribute** if it is a member of *some candidate key* of *R*. An attribute is called **nonprime** if it is not prime.

First Normal Form (1NF)

The domain of an attribute must include only *atomic* (simple, indivisible) *values* and that the value of any attribute in a tuple must be a *single value* from the domain of that attribute.

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

DEPARTMENT

Dname	Dnumber	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

Dnumber	Dlocations
5	Bellaire
5	Sugarland
5	Houston
4	Stafford
1	Houston

Second Normal Form (2NF)

Definitions:

- A functional dependency $X \rightarrow Y$ is a **full functional dependency** if removal of any attribute A from X means that the dependency does not hold any more;
- A functional dependency $X \rightarrow Y$ is a **partial dependency** if some attribute A can be removed from X and the dependency still holds;

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation

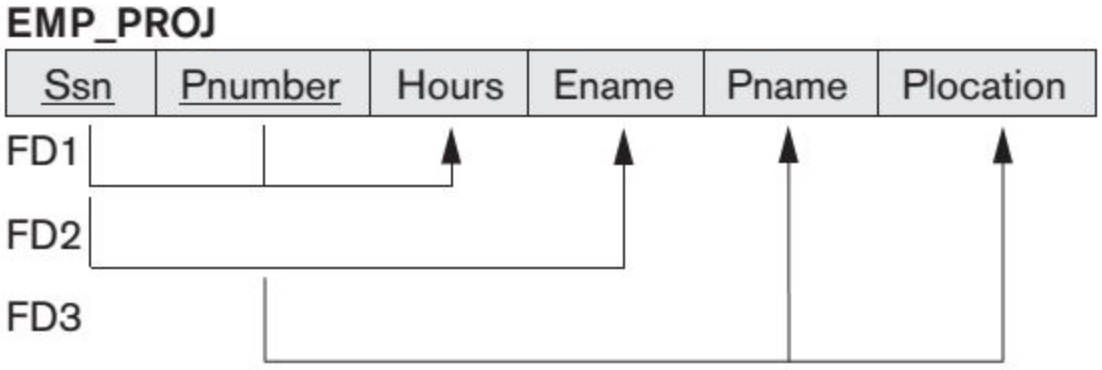
$\{Ssn, Pnumber\} \rightarrow Hours$ - Full dependency

$\{Ssn, Pnumber\} \rightarrow Ename$ - Partial dependency

Second Normal Form (2NF)

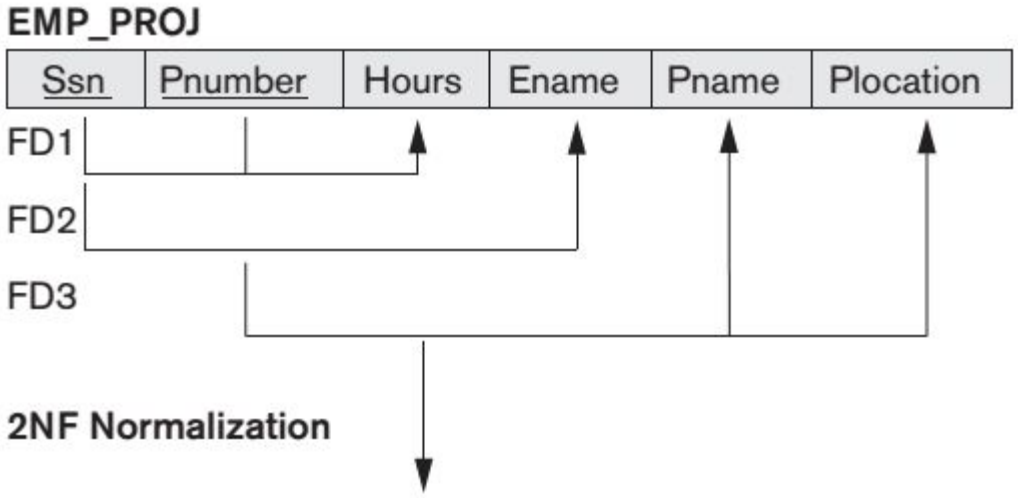
A relation schema R is in 2NF if every nonprime attribute A in R is *fully functionally dependent* on the primary key of R.

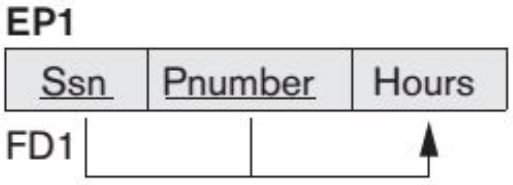
- ✔ The test for 2NF involves testing for functional dependencies whose left-hand side attributes are part of the primary key.
- ✔ If the primary key contains a single attribute, the test need not be applied at all.

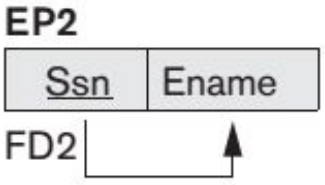


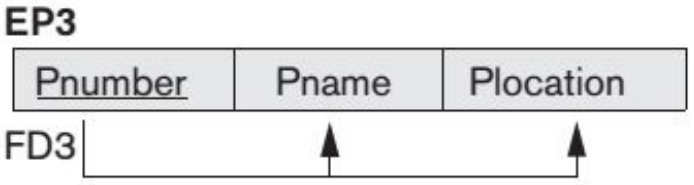
Nonprime attributes Ename, Pname, and Plocation violate 2NF.

Second Normal Form (2NF)



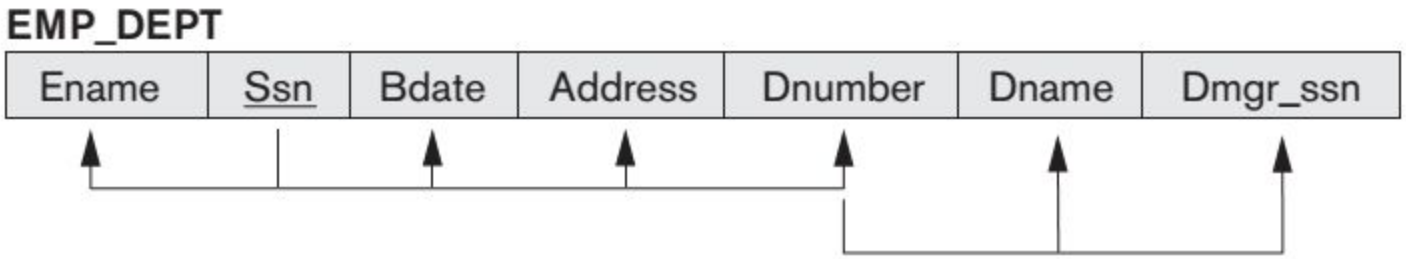






Third Normal Form (3NF)

Definition:
A functional dependency $X \rightarrow Y$ in a relation schema R is a **transitive dependency** if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R, and both $X \rightarrow Z$ and $Z \rightarrow Y$ hold.



The dependency $Ssn \rightarrow Dmgr\_ssn$ is transitive through $Dnumber$

A relation schema *R* is in **3NF** if it satisfies 2NF *and* no nonprime attribute of *R* is transitively dependent on the primary key.

Third Normal Form (3NF)