IDSc 6201:
Information Systems Development

Normalization

Jungpil Hahn
Information and Decision Sciences
Carlson School of Management
University of Minnesota
jhahn@csom.umn.edu
http://www.jungpil.com

Pitfalls in Relational Database Design

- **Relational database design requires that we find a “good” collection of relation schemas.**
- **A bad design may lead to**
  - **Redundant information**
  - **Update anomalies**
  - **Inability to represent certain information**

Design Guidelines for RDB

1. **Design a relation schema so that it is easy to explain its meaning.**
   - **Do not combine attributes from multiple entity types and relationship types into a single relation.**
     - ***Otherwise, it tends to be a mixture of multiple entities and relationships and hence semantically unclear.***
2. **Reduce the redundant values in tuples.**
   - **Design the base relation schemas so that no insertion, deletion, or modification anomalies occur in the relations.**
3. **Reduce the null values in tuples.**
   - **As far as possible, avoid placing attributes in a base relation whose values may be null. If nulls are unavoidable, make sure that they apply in exceptional cases only and do not apply to a majority of tuples in the relation.**

Example of a Bad Relational Schema

- EmpDept (SSN, Ename, Address, Dnumber, Dname, Mgr-SSN)

- Problems
  - Semantically unclear (Guideline 1)
  - Redundant data (Guideline 2)
  - Inconsistent information (Guideline 2)
  - Insertion, deletion, modification anomalies (Guideline 2)
  - Null values (Guideline 3)

Normalization Theory

- **Normalization process is a process during which unsatisfactory relation schemas are decomposed by breaking up their attributes into smaller relation schemas that posses desirable properties**
- **The overall objectives of the normalization process are:**
  - **To eliminate certain kinds of data redundancy**
  - **To avoid certain update (insertion, deletion, and modification) anomalies**

First Normal Form (1NF)

- **Definition**
  - **A relation R is in first normal form (1NF), if and only if every attributes is *single-valued* (atomic) for each tuple**
- **A relation in 1NF mean that each attributes in each tuple contains only one value**
- **1NF is considered to be part of the formal definition of a relation (i.e., a relation is already in (at least) 1NF)**

Example (1NF)

Student

StudentID	Sname	GPA	CourseID	Cname	InstructorID	Iname
1234567	David Jones	3.5	IDS6201 IDS6441	Info Sys Dev Intro EC	123456789 234567891	Hahn Kauffman
1346795	Ben Hacker	3.7	IDS6451 IDS6201 IDS6202	Telecomm Info Sys Dev Mgmt of IS Dev	345678912 123456789 456789123	Naumann Hahn Park
…	…	…	…	…	…	…

Student

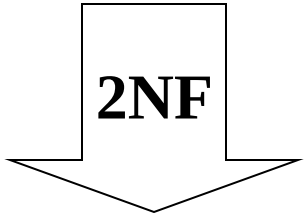
StudentID	Sname	GPA	CourseID	Cname	InstructorID	Iname
111223333	David Jones	3.5	IDS6201	Info Sys Dev	123456789	Hahn
111223333	David Jones	3.5	IDS6441	Intro EC	234567891	Kauffman
222334444	Mary Jones	3.7	IDS6451	Telecomm	345678912	Naumann
222334444	Mary Jones	3.7	IDS6201	Info Sys Dev	123456789	Hahn
222334444	Mary Jones	3.7	IDS6202	Mgmt of IS Dev	456789123	Park
…	…	…	…	…	…	…

Second Normal Form (2NF)

- **Definition**
  - **Relation *R* is in second normal form (2NF), if and only if it is in 1NF and every non-key attributes in R is not partially dependent on the primary key of *R***
- **Functional Dependency**
  - ***a* -> *b* is a full functional dependency, if *b* is functionally dependent on *a*, but not functionally dependent on any proper subset of a (i.e., the removal of any attribute from *a* means that the dependency does not hold any more)**
  - ***a* -> *b* is a partial dependency, if *b* is functionally dependent on *a* and also functionally dependent on any proper subset of *a***

Example (2NF)

STUDENT (StudentID, Sname, GPA, CourseID, Cname, InstructorID, Iname)



STUDENT(StudentID, Sname, GPA)

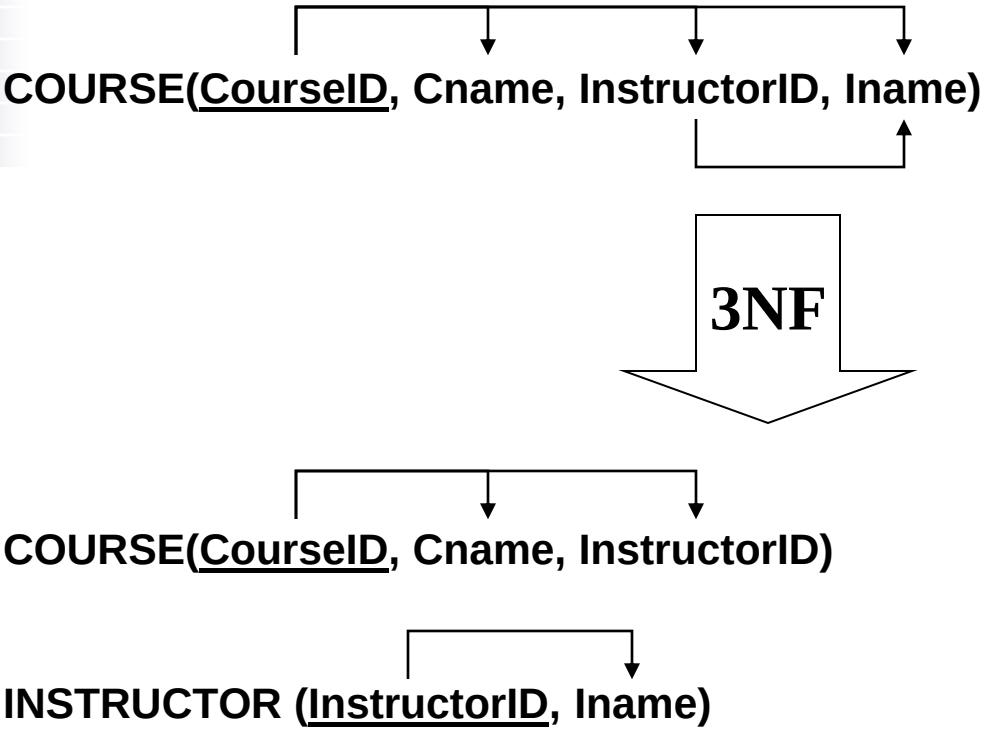
COURSE(CourseID, Cname, InstructorID, Iname)

TAKES(StudentID, CourseID)

Third Normal Form (3NF)

- **Definition**
  - **Relation *R* is in 3NF, if and only if it is in 2NF and no non-key attribute of *R* is transitively dependent on the primary key**
- **Transitive Dependency**
  - ***a* -> *b* in a relation R is a transitive dependency, if there is a set of attributes *g* that is not a subset of the primary key of R, and both *a* -> *g* and *g* -> *b* hold. In other words, if *a* -> *g* and *g* -> *b*, then *a* -> *b* is a transitive functional dependency**

Example (3NF)


COURSE(CourseID, Cname, InstructorID, Iname)
3NF
COURSE(CourseID, Cname, InstructorID)
INSTRUCTOR (InstructorID, Iname)

Boyce-Codd Normal Form (BCNF)

- **Definition**
  - **Relation *R* is in BCNF, if and only if it is in 3NF and every determinant (left-hand side of FD) is a candidate key**

- **In practice, most relation schemas that are in 3NF are also in BCNF.**