ENSF 608: Normalization and Dependencies

Dr. Emily Marasco
Fall 2021
Textbook: Fundamentals of Database Systems, 7th Ed., Elmasri & Navathe

14.2 Functional Dependencies

- Functional dependencies (FDs)
  - Are used to specify **formal measures** of the "goodness" of relational designs

- A set of attributes *X* **functionally determines** a set of attributes *Y* if the value of *X* determines a unique value for *Y*

14.2.1 Defining Functional Dependencies

- $X \rightarrow Y$ holds if whenever two tuples have the same value for X, they **must have** the same value for Y
  - For any two tuples t_1 and t_2 in any relation instance

$$r(R): \text{If } t_1[X] = t_2[X], \textbf{ then } t_1[Y] = t_2[Y]$$

- $X \rightarrow Y$ in R specifies a **constraint** on all relation instances $r(R)$
- Written as $X \rightarrow Y$ displayed graphically on a schema
- Derived from the real-world constraints on the attributes
- If K is a key of R, then K functionally determines all attributes in R
  - (since we never have two distinct tuples with $t_1[K]=t_2[K]$)

Examples of FD Constraints

- Social security number determines employee name
  - SSN → ENAME
- Project number determines project name and location
  - PNUMBER → {PNAME, PLOCATION}
- Employee ssn and project number determines the hours per week that the employee works on the project
  - {SSN, PNUMBER} → HOURS

Defining FDs From Instances

- Note that in order to define the FDs, we need to understand the meaning of the attributes involved and the relationship between them.
- An FD is a property of the attributes in the schema *R*
- Given the instance (population) of a relation, all we can conclude is that an FD **may exist** between certain attributes.
- What we can definitely conclude is – that certain FDs **do not exist** because there are tuples that show a violation of those dependencies.

Figure 14.7 Ruling Out FDs

Note that given the state of the TEACH relation, we can say that the FD: Text → Course may exist. However, the F Ds Teacher → Course, Teacher →Text and Couse → Text are ruled out.

TEACH

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

Figure 14.8 What FDs May Exist?

- A relation A relation *R*(A, B, C, D) with its extension.
- Which FDs may exist in this relation?

A	B	C	D
a1	b1	c1	d1
a1	b2	c2	d2
a2	b2	c2	d3
a3	b3	c4	d3

14.3.1 Normalization of Relations

- **Normalization:**
  - The process of decomposing unsatisfactory “bad” relations by breaking up their attributes into smaller relations
- **Normal form:**
  - Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form

14.3.3 Definitions of Keys and Attributes Participating in Keys (1 of 2)

- A **superkey** of a relation schema is $R = \{A_1, A_2, ..., A_n\}$

  a set of attributes S **subset-of** *R* with the property that no two tuples t_1 and t_2 in any legal relation state *r* of *R* will have $t_1[S] = t_2[S]$

- A **key** *K* is a **superkey** with the **additional property** that removal of any attribute from *K* will cause *K* not to be a superkey any more.

14.3.3 Definitions of Keys and Attributes Participating in Keys (2 of 2)

- 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**.
- A **Prime attribute** must be a member of **some** candidate key
- A **Nonprime attribute** is not a prime attribute—that is, it is not a member of any candidate key.

14.3.4 First Normal Form

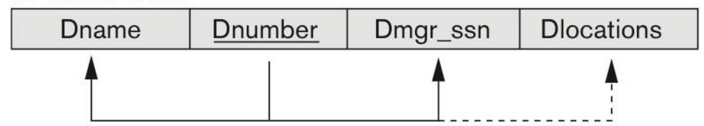
- Disallows
  - composite attributes
  - multivalued attributes
  - **nested relations**; attributes whose values for an **individual tuple** are non-atomic
- Considered to be part of the definition of a relation
- Most RDBMSs allow only those relations to be defined that are in First Normal Form

Figure 14.9 Normalization into 1NF

(a) A relation schema that is not in 1NF. (b) Sample state of relation DEPARTMENT. (c) 1NF version of the same relation with redundancy.

(a)

DEPARTMENT



(b)

DEPARTMENT

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

(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Figure 14.10 Normalizing Nested Relations into 1NF

(a) Schema of the EMP_PROJ relation with a nested relation attribute PROJS. (b) Sample extension of the EMP_PROJ relation showing nested relations within each tuple. (c) Decomposition of EMP_PROJ into relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key.

(a)

EMP_PROJ

		Projs	
Ssn	Ename	Pnumber	Hours

(b)

EMP_PROJ

Ssn	Ename	Pnumber	Hours
123456789	Smith, John B.	1	32.5
		2	7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1	20.0
		2	20.0
333445555	Wong, Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya, Alicia J.	30	30.0
		10	10.0
987987987	Jabbar, Ahmad V.	10	35.0
		30	5.0
987654321	Wallace, Jennifer S.	30	20.0
		20	15.0
888665555	Borg, James E.	20	NULL

(c)

EMP_PROJ1

Ssn	Ename

EMP_PROJ2

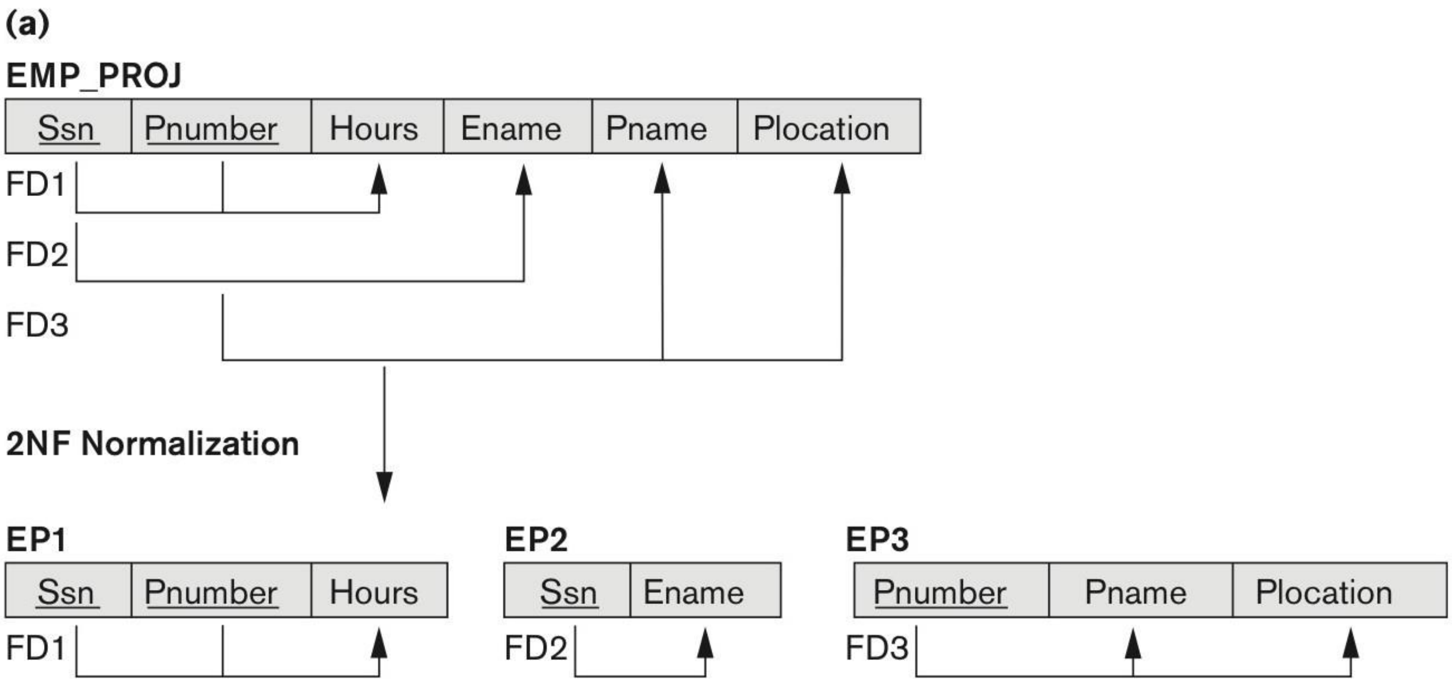
Ssn	Pnumber	Hours

14.3.5 Second Normal Form

- A relation schema R is in second normal form (2N F) if every non-prime attribute A in R is fully functionally dependent on the primary key
- **Full functional dependency**: a FD Y → Z where removal of any attribute from Y means the FD does not hold any more
- Examples:
  - {SSN, PNUMBER} → HOURS is a full FD since neither SSN → HOURS nor PNUMBER → HOURS hold
  - {SSN, PNUMBER} → ENAME is not a full FD (it is called a **partial dependency**) since SSN → ENAME also holds

Figure 14.11 Normalizing into 2NF

(a) Normalizing EMP_PROJ into 2NF relations.



14.3.6 Third Normal Form (1 of 2)

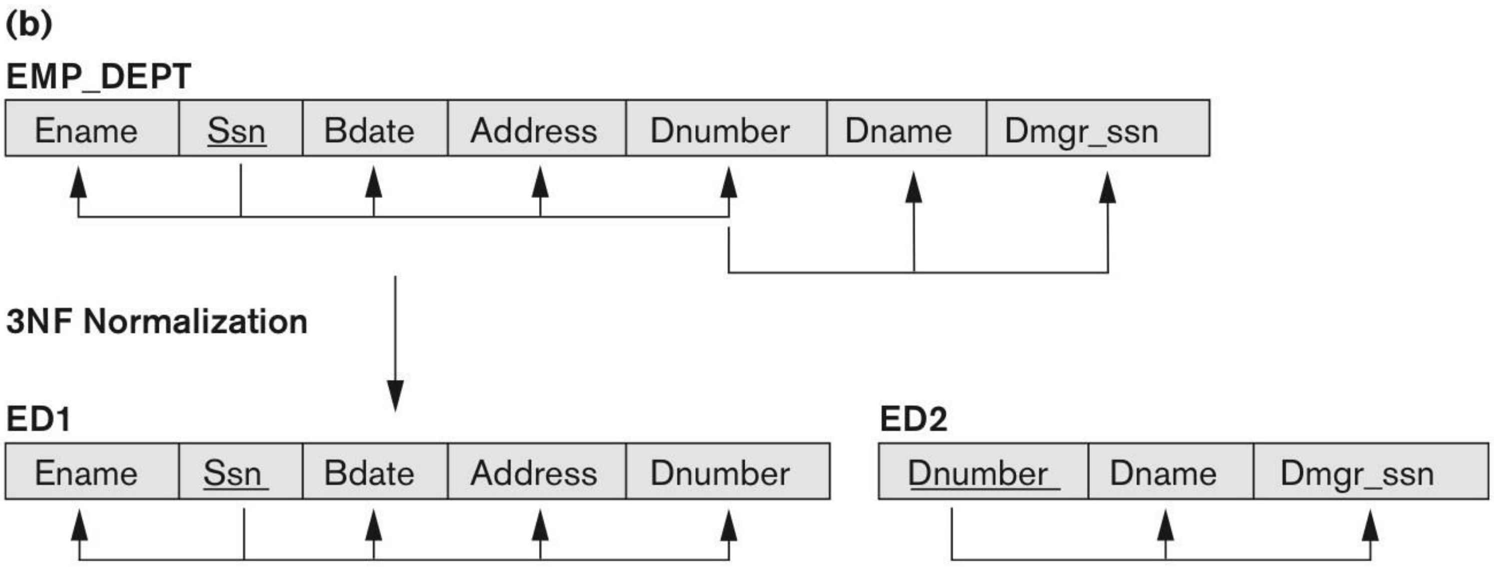
- Definition:
  - **Transitive functional dependency:** a FD $X \rightarrow Z$ that can be derived from two FDs $X \rightarrow Y$ and $Y \rightarrow Z$
- Examples:
  - SSN → DMGRSSN is a **transitive** FD
    - Since SSN → DNUMBER and DNUMBER → DMGRSSN hold
  - SSN → ENAME is **non-transitive**
    - Since there is no set of attributes X where SSN → X and X → ENAME

14.3.6 Third Normal Form (2 of 2)

- A relation schema *R* is in **third normal form (3NF)** if it is in 2NF **and** no non-prime attribute *A* in *R* is transitively dependent on the primary key
- *R* can be decomposed into 3NF relations via the process of 3NF normalization
- NOTE:
  - In $X \rightarrow Y$ and $Y \rightarrow Z$, with *X* as the primary key, we consider this a problem only if *Y* is not a candidate key.
  - When *Y* is a candidate key, there is no problem with the transitive dependency .
  - E.g., Consider EMP (SSN, Emp#, Salary).
    - Here, SSN → Emp# → Salary and Emp# is a candidate key.

Figure 14.11 Normalizing into 3NF

(b) Normalizing EMP_DEPT into 3NF relations.



Normal Forms Defined Informally

- 1st normal form
  - All attributes depend on **the key**
- 2nd normal form
  - All attributes depend on **the whole key**
- 3rd normal form
  - All attributes depend on **nothing but the key**

14.5 BCNF (Boyce-Codd Normal Form)

- A relation schema *R* is in **Boyce-Codd Normal Form (BCNF)** if whenever an **FD *X* → *A*** holds in *R*, then ***X* is a superkey** of *R*
- Each normal form is strictly stronger than the previous one
  - Every 2NF relation is in 1NF
  - Every 3NF relation is in 2NF
  - Every BCNF relation is in 3NF
- There exist relations that are in 3NF but not in BCNF
- Hence BCNF is considered a **stronger form of 3NF**
- The goal is to have each relation in BCNF (or 3NF)

Topic Summary

- Informal design guidelines for relational databases
- Functional dependencies (FDs)
- Normal forms
  - 1NF, 2NF, 3NF, BCNF

Copyright

