COSC344
Database Theory and Applications

Lecture 8
Functional dependencies

Learning Objectives of this Lecture

- You should
  - understand the four informal guidelines for measuring the quality of a relational schema.
  - understand what is a functional dependency.
  - understand what is decomposition and the three properties of a successful decomposition.
  - understand the inference rules and Armstrong's Axioms.
  - be able to compute the closure for a set of attributes under a set of functional dependencies.
  - be able to compute the minimal cover for a set of functional dependencies.

- Source
  - Textbook: Chapter 14.1-14.2, Chapter 15.1

Questions to Ponder

- What constitutes a good database design?
- When you turn your ER diagram into tables, is the resultant set of tables the most desirable?
- How can we decide whether a given set of tables is “good”?
- Is it possible for a set of tables to cause us grief later on?

Quality Measures for Relation Schema

- Four informal guidelines
  - Make sure that the semantic of the attributes is clear in the schema
  - Reducing redundant values in tuples
  - Reducing NULLs in tuples
  - Disallowing the generation of spurious tuples when tables are joined

Semantics of the Attributes

Guideline 1:

- Design a schema in a way that it is easy to explain its meaning.
- Do not combine attributes from multiple entity types and relationship types into a single relation.

- Consider these relations
  - Are they good designs?

EMP_DEPT

ENAME	IRD	BDATE	ADDRESS	DNUMBER	DNAME	DMGRIRD

EMP_PROJ

IRD	PNUMBER	HOURS	ENAME	PNAME	LOCATION

EMP_DEPT

ENAME	IRD	BDATE	ADDRESS	DNUMBER	DNAME	DMGRIRD
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5	Research	333445555
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4	Administration	987654321
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4	Administration	987654321
Narayan,Ramesh K.	666884444	1962-09-15	975 FireOak,Humble,TX	5	Research	333445555
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5	Research	333445555
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4	Administration	987654321
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1	Headquarters	888665555

Redundancy

EMP_PROJ

IRD	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
123456789	1	32.5	Smith,John B.	ProductX	Bellaire
123456789	2	7.5	Smith,John B.	ProductY	Sugarland
666884444	3	40.0	Narayan,Ramesh K.	ProductZ	Houston
453453453	1	20.0	English,Joyce A.	ProductX	Bellaire
453453453	2	20.0	English,Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong,Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong,Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong,Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

Figure 14.4 Example relations for the schemas in Figure 14.3 that result from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

Redundant Information in Tuples

- Insertion anomalies
  - What happens if we insert a new employee into EMP_DEPT?
  - What happens if we want to insert a new department that does not yet have any employees?
- Deletion anomalies
  - What happens if we delete the last employee who works for a department?
- Modification anomalies
  - What happens if the manager of a department is changed?

Guideline 2:

Design the schemas so that there are no insertion, deletion or modification anomalies.

Reducing Null Markers in Tuples

- If many attributes in a schema do not apply to all tuples, many NULLs exist.
  - Takes up storage space
  - Problems with understanding the meaning of the attributes
  - Problem with join operations
  - Hard to account for them in aggregate functions
- Multiple interpretations of NULL
  - The attribute does not apply to this tuple.
  - The attribute value of this tuple is unknown.
  - The attribute value of this tuple is known, but is absent.

Guideline 3:

- Avoid placing attributes in a relation whose value frequently may be null.
- If nulls are unavoidable, make sure they apply in exceptional cases only, and do not apply to a majority of tuples

Generation of Spurious Tuples

(a)

EMP_LOCS

Ename	Plocation

P.K. (Ename, Plocation)

EMP_PROJ1

IRD	Pnumber	Hours	Pname	Plocation

P.K. (IRD, Pnumber)

(b)

EMP_LOCS

Ename	Plocation
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford
Zelaya, Alicia J.	Stafford
Jabbar, Ahmad V.	Stafford
Wallace, Jennifer S.	Stafford
Wallace, Jennifer S.	Houston
Borg, James E.	Houston

EMP_PROJ1

IRD	Pnumber	Hours	Pname	Plocation
123456789	1	32.5	ProductX	Bellaire
123456789	2	7.5	ProductY	Sugarland
666884444	3	40.0	ProductZ	Houston
453453453	1	20.0	ProductX	Bellaire
453453453	2	20.0	ProductY	Sugarland
333445555	2	10.0	ProductY	Sugarland
333445555	3	10.0	ProductZ	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford
987654321	20	15.0	Reorganization	Houston
888665555	20	NULL	Reorganization	Houston

Figure 15.5
Particularly poor design for the EMP_PROJ relation in Figure 15.3(b). (a) The two relation schemas EMP_LOCS and EMP_PROJ1. (b) The result of projecting the extension of EMP_PROJ from Figure 15.4 onto the relations EMP_LOCS and EMP_PROJ1.

Generation of Spurious Tuples (cont'd)

	IRD	Pnumber	Hours	Pname	Plocation	Ename
	123456789	1	32.5	ProductX	Bellaire	Smith, John B.
*	123456789	1	32.5	ProductX	Bellaire	English, Joyce A.
	123456789	2	7.5	ProductY	Sugarland	Smith, John B.
*	123456789	2	7.5	ProductY	Sugarland	English, Joyce A.
*	123456789	2	7.5	ProductY	Sugarland	Wong, Franklin T.
	666884444	3	40.0	ProductZ	Houston	Narayan, Ramesh K.
*	666884444	3	40.0	ProductZ	Houston	Wong, Franklin T.
*	453453453	1	20.0	ProductX	Bellaire	Smith, John B.
	453453453	1	20.0	ProductX	Bellaire	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Smith, John B.
	453453453	2	20.0	ProductY	Sugarland	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	2	10.0	ProductY	Sugarland	Smith, John B.
*	333445555	2	10.0	ProductY	Sugarland	English, Joyce A.
	333445555	2	10.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	3	10.0	ProductZ	Houston	Narayan, Ramesh K.
	333445555	3	10.0	ProductZ	Houston	Wong, Franklin T.
	333445555	10	10.0	Computerization	Stafford	Wong, Franklin T.
*	333445555	20	10.0	Reorganization	Houston	Narayan, Ramesh K.
	333445555	20	10.0	Reorganization	Houston	Wong, Franklin T.

* * *

Figure 15.6
Result of applying NATURAL JOIN to the tuples above the dashed lines in EMP_PROJ1 and EMP_LOCS of Figure 15.5. Generated spurious tuples are marked by asterisks.

Generation of Spurious Tuples (cont'd)

- The table in the previous slide contains tuples not in the original table. These tuples are called spurious tuples which represent information that is not valid.

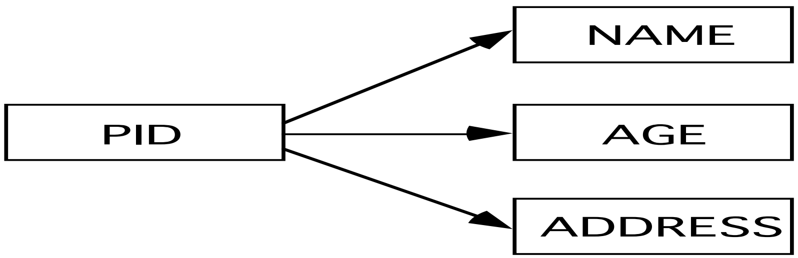
Guideline 4:

- Design schemas so that they can be joined with equality conditions on attributes that are either primary keys or foreign keys in a way that guarantees no spurious tuples are created.
- Do not have relations that contain matching attributes other than primary key-foreign key combinations because joining on such attributes may produce spurious tuples.

Is there a formal tool for analysis of relational schemas that enables us to detect and describe the mentioned problems in precise terms?

Informal Introduction to Functional Dependency

- Person (PID, NAME, AGE, ADDRESS)

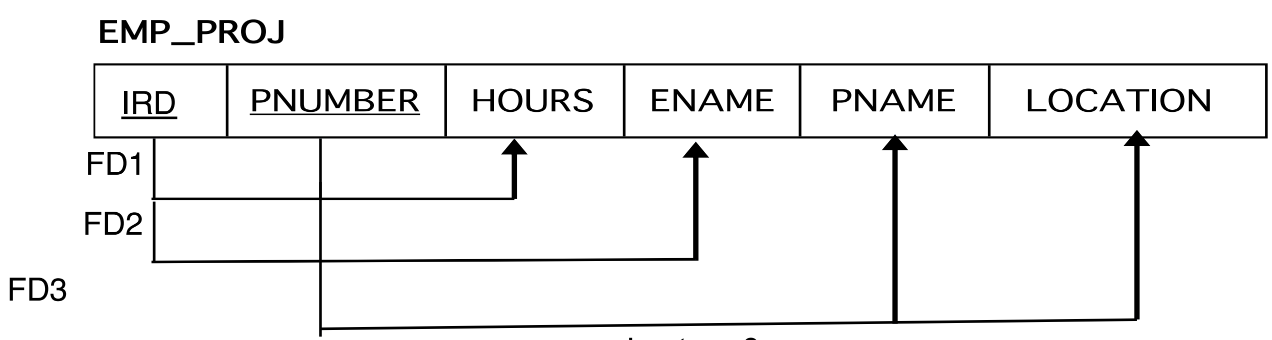


- F = {PID → NAME, PID → AGE, PID → ADDRESS}
- The notation PID → AGE means that PID *determines* AGE or AGE is *functionally dependent* on PID.
- Many real world situations which lead to data restrictions or constraints in the modelled world.
  - *You can't be in two places at the same time*

{Person, date, time} → Place

Formal Definition of Functional Dependency

- A *functional dependency* (FD), denoted by $X \rightarrow Y$ where X and Y are two subsets of attributes of a relation, *specifies a constraint* on the possible tuples that can form a state of the relation. The constraint is that, for any two tuples t_1 and t_2 that have $t_1[X] = t_2[X]$, we must also have $t_1[Y] = t_2[Y]$.
- The Y values of a tuple depend on or are determined by the X values; or the X value determines the Y value.
- The set of attributes X is called the left-hand side (LHS) of the FD, and Y is called the right-hand side (RHS).
- Diagrammatic notation



Observations About FD's

- If $X \rightarrow Y$, whenever two tuples have the same X value, they must necessarily have the same Y value.
- If X is a candidate key of a relation, then $X \rightarrow Y$ for any subset of attributes Y of the relation.
- If $X \rightarrow Y$, it does not imply $Y \rightarrow X$.
- A FD is a property of the semantics or meaning of the attributes.
- A functional dependency is a property of the schema not a property of a particular instance.
- The main use of functional dependencies is to describe further a relation schema by specifying constraints on its attributes that must hold at all times.

Decompositions

- Decomposition is the process of breaking down a relation schema R into a set of relation schemas $D=\{R_1,R_2,\ldots, R_m\}$ to get rid of unwanted dependencies.
  - A decomposition is just a relational algebra PROJECT
- Properties of a successful decomposition
  - **Attribute preservation:** each attribute in R will appear in at least one relation schema R_i so that no attributes are lost.
  - **Dependency preservation:** each functional dependency $X \rightarrow Y$ specified on R either appeared directly in some R_i in D or could be inferred from the dependencies that appear in some R_i.
  - **Lossless join:** no spurious tuples are generated when a NATURAL JOIN operation is applied to the relations resulting from the decomposition.
- Heath's Theorem (1971)

A relation R(X,Y,Z) that satisfies a functional dependency $X \rightarrow Y$ can always be non-loss decomposed into: R1 [X, Y] and R2 [X, Z]

Inference Rules for Functional Dependencies

- Reflexive rule
  - If Y is a subset of X, then $X \rightarrow Y$
  - Sometimes stated $X \rightarrow X$
- Augmentation rule
  - If $X \rightarrow Y$, then $XZ \rightarrow YZ$
  - Also $X \rightarrow Y$ implies $XZ \rightarrow Y$
- Transitive rule
  - If $\{X \rightarrow Y, Y \rightarrow Z\}$, then $X \rightarrow Z$
- Decomposition or projective rule
  - If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$.
- Union or additive rule
  - If $\{X \rightarrow Y, X \rightarrow Z\}$, then $X \rightarrow YZ$
- Pseudotransitive rule
  - If $\{X \rightarrow Y, WY \rightarrow Z\}$, then $WX \rightarrow Z$

Closure

- Use F to denote the set of functional dependencies that are specified on a relation R
- Closure of F, denoted by F^+, is defined as the set of all dependencies, that is,

  F and all dependencies that can be inferred from F.
- Given R(A, B, C) and F = {A → B, B → C}, what is F^+?

F⁺ = { A → A, B → B, C → C,

AB → AB, AC → AC, BC → BC, ABC → ABC,

AB → A,..................

A → B, AB → B, AC → B, ABC → B,

B → C, AB → C, BC → C, ABC → C,

B → BC, A → ABC,...............................}

Equivalence of Sets of Functional Dependencies

- A set of functional dependencies F is said to cover another set of functional dependencies E if every FD in E is also in F^+
  - That is, if every dependency in E can be inferred from F;
  - Alternatively, we can say the E is covered by F.
- Two sets of functional dependencies F and G are equivalent if $F^+ = G^+$.
  - Equivalence means that every FD in F can be inferred from G, and every FD in G can be inferred from F.

The closure of a set of attributes

- Given a set of functional dependencies F and a set of attributes X, the closure of X under F, denoted by $\mathbf{X^+}$, is the **maximal set** of attributes determined by X.
- Algorithm to compute $\mathbf{X^+}$

```
X+:= X;
repeat
    oldX+:= X+
   for (each FD Y → Z in F) {
     if ( Y ⊆ X+ ) then
          X+ = X+ ∪  Z;
   }
until (X+  = oldX+ );
```

- Example:

  R(A, B, C) and F = {A → B, B → C}

  $\{A\}^+ = \{A, B, C\}$ $\{A, B\}^+ = \{A, B, C\}$

  $\{B\}^+ = \{B, C\}$ $\{B, C\}^+ = \{B, C\}$ $\{A, B, C\}^+ = \{A, B, C\}$

  $\{C\}^+ = \{C\}$ $\{A, C\}^+ = \{A, B, C\}$

- Is {B,C} a candidate key? Why?
- Does F imply BC→ A?

Minimal Sets of Functional Dependencies

- A set of functional dependencies F is minimal (irreducible) if it satisfies:
  - Every RHS is a single attribute
  - Every LHS is irreducible - no attribute is redundant
  - No FD in F is redundant
- Algorithm: Given a set of function dependencies E, compute minimal cover F

1. Set F: = E;
2. Replace each functional dependency $X \rightarrow \{A_1, A_2, \ldots, A_n\}$ in F with n functional dependencies $X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n$
3. For each functional dependency $X \rightarrow A$ in F
   For each attribute B that is an element of X
   if $\{\{F - \{X \rightarrow A\}\} \cup \{(X - \{B\}) \rightarrow A\}\}$ is equivalent to F
   then replace $X \rightarrow A$ with $(X - \{B\}) \rightarrow A$ in F
4. For each remaining functional dependency $X \rightarrow A$ in F
   if $\{F - \{X \rightarrow A\}\}$ is equivalent to F
   then remove $X \rightarrow A$ from F.

Minimal Cover - example

$E = \{ A \rightarrow B,\ AB \rightarrow D,\ C \rightarrow AD,\ C \rightarrow J \}$

Step 1: $F = E = \{ A \rightarrow B,\ AB \rightarrow D,\ C \rightarrow AD,\ C \rightarrow J \}$

Step 2: $F = \{ A \rightarrow B,\ AB \rightarrow D,\ C \rightarrow AD,\ C \rightarrow J \}$

$F = \{ A \rightarrow B,\ AB \rightarrow D,\ C \rightarrow A,\ C \rightarrow D,\ C \rightarrow J \}$

Step 3:

$F = \{ A \rightarrow B,\ \sout{AB \rightarrow} D,\ C \rightarrow A,\ C \rightarrow D,\ C \rightarrow J \}$

$F = \{ A \rightarrow B,\ A \rightarrow D,\ C \rightarrow A,\ C \rightarrow D,\ C \rightarrow J \}$

Step 4:

$F = \{ A \rightarrow B,\ A \rightarrow D,\ C \rightarrow A,\ C \rightarrow D,\ C \rightarrow J \}$

$F = \{ A \rightarrow B,\ A \rightarrow D,\ C \rightarrow A, C \rightarrow J \}$

Summary

- Informal design
  - 4 design guidelines
- Formal definition of Functional Dependency
  - Observations about FD
  - Inference rules
  - Armstrong's Axioms
  - Closure
  - Equivalence
  - Minimal sets of FD