Functional Dependencies

Dr. Seema Gupta Bhol

Database Design

- Why are some designs bad?
- What's a functional dependency?
- What's the theory of functional dependencies?

Not all designs are equally good

- Why is this design bad?

```
Data(sid,sname,address,cid,cname,grade)
```

- Why is this one preferable?

```
Student(sid,sname,address)
Course(cid,cname)
Enrolled(sid,cid,grade)
```

An instance of bad design

sid	sname	address	cid	cname	grade
124	Britney	USA	206	Database	A++
204	Victoria	Essex	202	Semantics	C
124	Britney	USA	201	Eng I	A+
206	Emma	London	206	Database	B-
124	Britney	USA	202	Semantics	B+

Evils of redundancy

- **Redundancy** is the root of many problems associated with relational schemas
  - Redundant storage
  - Update anomalies: if address of student Britny is updated from USA to USSR then, we have to made three updations. Updations not properly done will cause inconsistency.
  - Insertion anomalies: we cant add info that code for course AI is 205, till we have some student enrolled in course.
  - Deletion anomalies: If only student learning Eng 1 will leave then we will also lost information that code for Eng1 is 201.

Decomposition

We remove anomalies by replacing the schema

```
Data(sid,sname,address,cid,cname,grade)
```

with

```
Student(sid,sname,address)
Course(cid,cname)
Enrolled(sid,cid,grade)
```

Thus,Design the base relation schemas so that no insertion, deletion, or modification anomalies occur.

Functional dependencies

- We can say that `sid` determines `address`
  - We'll write this

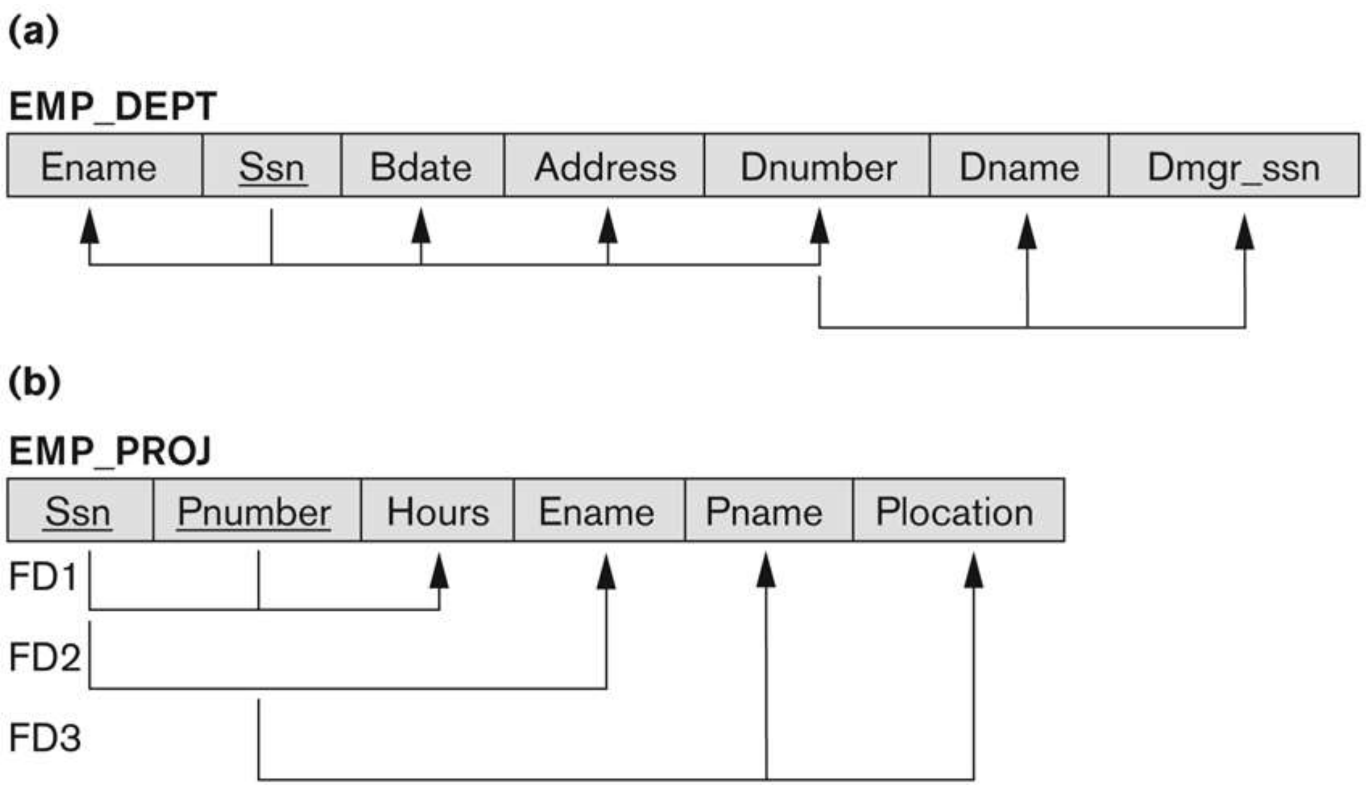
$$\texttt{sid} \rightarrow \texttt{address}$$

- This is called a **functional dependency (FD).**
- FDs are derived from the real-world constraints on the attributes

Functional dependencies

- We'd expect the following functional dependencies to hold in our Student database
  - `sid → sname,address`
  - `cid → cname`
  - `sid,cid → grade`
- A functional dependency $X \rightarrow Y$ is simply a pair of sets (of field names)

Example2


(a)
EMP_DEPT
Ename
Ssn
Bdate
Address
Dnumber
Dname
Dmgr_ssn
(b)
EMP_PROJ
Ssn
Pnumber
Hours
Ename
Pname
Plocation
FD1
FD2
FD3

Examples of FD

- 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

Redundancy

EMP_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn
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 Redundancy

EMP_PROJ

Ssn	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

EXAMPLE OF AN UPDATE ANOMALY

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)
- Update Anomaly:
  - Changing the name of project number P1 from "Product X" to "Customer-Accounting" may cause this update to be made for all 100 employees working on project P1.

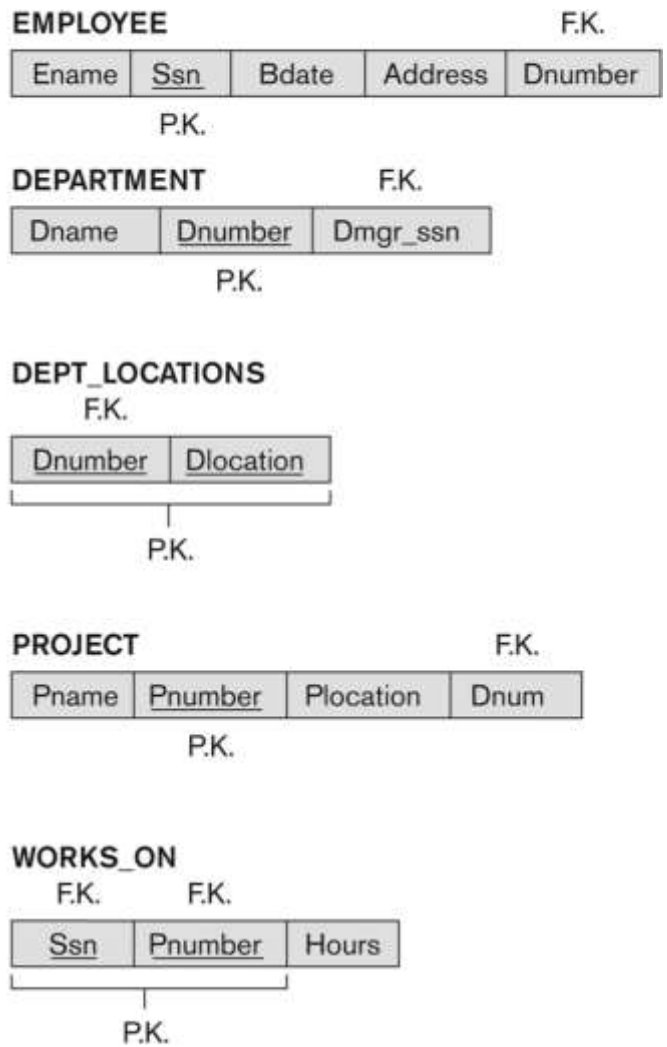
EXAMPLE OF AN INSERT ANOMALY

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)
- Insert Anomaly:
  - Cannot insert a project unless an employee is assigned to it.
- Conversely
  - Cannot insert an employee unless an he/she is assigned to a project.

EXAMPLE OF AN DELETE ANOMALY

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)
- Delete Anomaly:
  - When a project is deleted, it will result in deleting all the employees who work on that project.
  - Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

Decoposed COMPANY relational database schema



Functional dependencies Formal definition

- Given a relation $R=R(A_1:\tau_1, \ldots, A_n:\tau_n)$, and X, Y ($\subseteq\{A_1, \ldots, A_n\}$), an instance r of R satisfies **$X \rightarrow Y$**, if

  For any two tuples t_1, t_2 in R, if $t_1.X=t_2.X$ then $t_1.Y=t_2.Y$

  or

  If t1[X]=t2[X], *then* t1[Y]=t2[Y]

- If $X \rightarrow Y$, we say X functionally determines Y or Y is functionally dependent on X.

- Functional dependency is abbreviate as FD. X is called the left-hand side of the FD. Y is called the right-hand side of the FD.

Types of Functional Dependencies

- Trivial functional dependency
- Non-Trivial functional dependency
- Transitive functional dependency
- Multivalued functional dependency

Trivial Functional Dependency

- In **Trivial Functional Dependency**, a dependent is always a subset of the determinant. i.e.

  If **X** → **Y** and **Y is the subset of X**, then it is called trivial functional dependency

e.g

{ssid, sname} → sname

Non-trivial functional dependency

- In **Non-trivial functional dependency**, the dependent is strictly not a subset of the determinant. i.e.

  If **X** → **Y** and **Y is not a subset of X**, then it is called Non-trivial functional dependency.

e.g.

`sid,cid → grade`

Multivalued functional dependency

- In **Multivalued functional dependency**, entities of the dependent set are **not dependent on each other.** i.e.

  If **a** → **{b, c}** and there exists **no functional dependency** between **b and c**, then it is called a **multivalued functional dependency.**

- Car(carmodel,mafyear,colour)

  - maf_year and color are independent of each other but dependent on car_model.
  - In this example, these two columns are said to be multivalue dependent on car_model.
  - This dependence can be represented like this:
    - car_model -> maf_year
    - car_model-> colour

Transitive functional dependency

- In transitive functional dependency, dependent is indirectly dependent on determinant. i.e.

  If **a** → **b** & **b** → **c**, then according to axiom of transitivity, **a** → **c**.

  This is a **transitive functional dependency.**

Fully Functional Dependency

If X and Y are attributes of a relation R, Y is fully functional dependent on X, if Y is functionally dependent on X but not on any proper subset of X.

Example –

In the relation ABC->D, attribute D is fully functionally dependent on ABC and not on any proper subset of ABC. That means that subsets of ABC like AB, BC, A, B, etc cannot determine D.

Partial Functional Dependency

A functional dependency X->Y is a partial dependency if Y is functionally dependent on X and Y can be determined by any proper subset of X.

For example, we have a relationship

{ AC->B, A->D, and D->B}.

A->D and D->B , thus A->B

Here A is alone capable of determining B, which means B is partially dependent on AC.

Inference Rules for FDs

- Given a set of FDs F, we can *infer* additional FDs that hold whenever the FDs in F hold.

- An FD set F **logically implies** $X \rightarrow Y$, and write **F** $\models$ $\mathbf{X \rightarrow Y}$

Armstrong's inference rules

A1. (Reflexive) If Y <u>subset-of</u> X, then X → Y

A2. (Augmentation) If X → Y, then XZ → YZ

A3. (Transitive) If X → Y and Y → Z, then X → Z

- A1, A2, A3 form a *sound* and *complete* set of inference rules

Soundness &Completeness

- **Soundness**
  - If F |= $X \rightarrow Y$ is deduced using the rules, then $X \rightarrow Y$ is true in any relation in which the dependencies of F are true.
- **Completeness**
  - If $X \rightarrow Y$ is true in any relation in which the dependencies of F are true, then F |= $X \rightarrow Y$ can be deduced using the rules.

Additional Inference Rules

- Decomposition
  - If X $\rightarrow$ YZ, then X $\rightarrow$ Y and X $\rightarrow$ Z
- Union
  - If X $\rightarrow$ Y and X $\rightarrow$ Z, then X $\rightarrow$ YZ
- Psuedotransitivity
  - If X $\rightarrow$ Y and WY $\rightarrow$ Z, then WX $\rightarrow$ Z

Closure of F

- **Closure** of a set F of FDs is the set F+ of all FDs that can be inferred from F
- The *closure* of *F*, denoted by F^+, is the set of all FDs that can be inferred from F., i.e.

$$\mathbf{F^+} = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$$

- The set F^+ can be big, even if F is small

Closure of a set of FDs

- *Which of the following are in the closure of our Student FDs?*
  - `sid→address`
  - `cid→cname`
  - `cid→cname,sname`
  - `cid,sid→cname,sname`

Equivalence

- Two sets of FDs, F and G, are said to be **equivalent** if $F^+=G^+$
- For example:

  $F1=\{(A,B \rightarrow C), (A \rightarrow B)\}$ and

  $F2= \{(A \rightarrow C), (A \rightarrow B)\}$

  are equivalent

Closure of an Attribute

- **Closure of an Attribute:** Closure of an Attribute can be defined as a set of attributes that can be functionally determined from it.
- Closure of an attribute X is X^+

Find the closure of A,B,C,D in R(A,B,C,D) where:

FD : {A->B,B->D,C->B}

A+

A->B,

B->D =>A->D

Thus, A+=ABD

B+

B->D

Thus B+=BD

C+

C->B

B->D =>C->D

Thus, C+=CBD

D+=D

Closure of attribute A

- Consider a relation R (A , B , C , D , E , F , G) with the functional dependencies-
- $A \rightarrow BC$
- $BC \rightarrow DE$
- $D \rightarrow F$
- $CF \rightarrow G$

- $A^+ = \{ A \}$
- $= \{ A , B , C \}$ (Using $A \rightarrow BC$)
- $= \{ A , B , C , D , E \}$ (Using $BC \rightarrow DE$)
- $= \{ A , B , C , D , E , F \}$ (Using $D \rightarrow F$)
- $= \{ A , B , C , D , E , F , G \}$ (Using $CF \rightarrow G$)
- Thus,
- $A^+ = \{ A , B , C , D , E , F , G \}$

Q. Find closure of D

Closure of D

- $D^+ = \{ D \}$
- $= \{ D , F \}$ (Using $D \rightarrow F$)

Find **Closure of { B , C }, i.e { B , C }$^+$**

- $\{ B , C \}^+ = \{ B , C \}$
- $= \{ B , C , D , E \}$ (Using $BC \rightarrow DE$)
- $= \{ B , C , D , E , F \}$ (Using $D \rightarrow F$)
- $= \{ B , C , D , E , F , G \}$ (Using $CF \rightarrow G$)

Thus,

- **$\{ B , C \}^+ = \{ B , C , D , E , F , G \}$**

EXAMPLE:

Given relational schema R(P Q R S T U V) having following attribute P Q R S T U and V, also there is a set of functional dependency denoted by

FD = { P->Q, QR->ST, PTV->V }.

Determine Closure of $(QR)^+$ and $(PR)^+$

- QR^+ = QR

  =QRST (given QR→ST)
- PR $^+$ = PR

given **P** → **Q**

- PR $^+$ **=PRQ**
- QR→ST
- =PRQST

Q1. Let R(ABCDEFGH) satisfy the following functional dependencies:

F= {A->B, CH->A, B->E, BD->C, EG->H, DE->F}

Which of the following FD is also guaranteed to be satisfied by R?

1. BFG ->AE
2. ACG -> DH
3. CEG ->AB

HINT: Compute the closure of the LHS of each FD that you get as a choice. If the RHS of the candidate FD is contained in the closure, then the candidate follows from the given FDs, otherwise not.

SOLUTION

1. BFG AE ???

Incorrect: BFG+ = BFGEH, which includes E, but not A

2. ACG DH ???

Incorrect: ACG+ = ACGBE, which includes neither D nor H.

3. CEG AB ???

Correct: CEG+ = CEGHAB, which contains AB

Candidate keys and FDs

- If $R=R(A_1:\tau_1, \ldots, A_n:\tau_n)$ with FDs F and $X \subseteq \{A_1, \ldots, A_n\}$, then X is a **candidate key** for R if
  - $X \rightarrow A_1, \ldots, A_n \in F^+$
  - For no proper subset $Y \subseteq X$ is
    $Y \rightarrow A_1, \ldots, A_n \in F^+$

Example

- Consider the Universal relation R={ABCDEFGHIJ} and the set of FDs
- F= {AB $\rightarrow$ C, A $\rightarrow$ DE, B $\rightarrow$ F, F $\rightarrow$ GH, D $\rightarrow$ IJ}.
- What is the key of R?

AB IS THE CANDIDATE KEY

1. AB->C
2. A->DE => A-> D AND A->E
3. B->F => AB=>F
4. F->GH AND B->F => B->GH => AB-> GH
5. D->IJ AND A->D => A->IJ

Finding Keys using FDs

Tricks for finding the key:

- If an attribute never appears on the RHS of any FD, it must be part of the key.
- If an attribute never appears on the LHS of any FD, but appears on the RHS of any FD, it must not be part of any key.

QUESTION

- Which of the following could be a key for R(A,B,C,D,E,F,G) with functional dependencies

F={ABC, CDE, EFG, FGE, DEC, and BCA}

- 1. BDF
- 2. ACDF
- 3. ABDFG
- 4. BDFG

SOLUTION

- 1. BDF ???

No. BDF+ = BDF

- 2. ACDF ???

No. ACDF+ = ACDFEG (The closure does not include B)

- 3. ABDFG ???
- No. This choice is a superkey, but it has proper subsets that are also keys (e.g. BDFG+ = BDFGECA).
- 4. BDFG ???
- BDFG+ = ABCDEFG

Check if any subset of BDFG is a key:

Since B, D, F never appear on the RHS of the FDs, they must form part of the key.

BDF+ = BDF

So, BDFG is the minimal key, hence the candidate key