Functional Dependencies
- Definition
- Functional Dependencies and Keys
- Inference Rules for Functional Dependencies
- Closure of Attribute Sets
- Canonical Cover

Normalization
- Goals
- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)
Relational database design requires that we find a “good” collection of relation schemas.

☞ A bad design may lead to several problems.

Functional dependencies can be used to refine a relation schema reduced from an E-R schema by iteratively decomposing it (called normalization) to place it in a certain normal form.

– The first four normal forms ⇒ use only functional dependencies.
– Additional normal forms ⇒ use other types of dependencies

☞ Normal forms do not guarantee a good design!
FUNCTIONAL DEPENDENCY (FD): DEFINITION

Let $R$ be a relation schema, $X, Y$ be sets of attributes in $R$ and $f$ be a time-varying function from $X$ to $Y$. Then

$$f: X \rightarrow Y$$

is a functional dependency (FD) if at every point in time, for a given value of $x$ in $X$ there is at most one value of $y$ in $Y$.

Example: $\text{PGStudent}(\text{studentId}, \text{name}, \text{supervisor}, \text{specialization})$

$$f: \text{supervisor} \rightarrow \text{specialization}$$

Consequences of the FD:

- If two student records have the same supervisor (e.g., Papadias), then they must have the same specialization (e.g., Database).
- On the other hand, if two student records have different supervisors, then they may have the same or different specializations.

<table>
<thead>
<tr>
<th>studentId</th>
<th>name</th>
<th>supervisor</th>
<th>specialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>23455789</td>
<td>Bruno Ho</td>
<td>Yang</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>23556789</td>
<td>Jenny Jones</td>
<td>Papadias</td>
<td>Database</td>
</tr>
<tr>
<td>25678989</td>
<td>Kathy Ko</td>
<td>Kim</td>
<td>Software Technology</td>
</tr>
<tr>
<td>26789012</td>
<td>Susan Sze</td>
<td>Papadias</td>
<td>Database</td>
</tr>
<tr>
<td>26184624</td>
<td>Terry Tam</td>
<td>Song</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>26186666</td>
<td>Carol Chen</td>
<td>Tai</td>
<td>Graphics</td>
</tr>
</tbody>
</table>
We normally omit the “f:” and simply write the FD as $X \rightarrow Y$.
- $X$ is called the determinant set or left-hand side (LHS) of the FD.
- $Y$ is called the dependent set or right-hand side (RHS) of the FD.

We say that $X$ determines $Y$ or $Y$ depends on $X$.

A functional dependency $X \rightarrow Y$ is trivial if $Y \subseteq X$.
- $Y$ appears on both the LHS and the RHS of the FD.
- Trivial FDs hold for all relation instances.

A functional dependency $X \rightarrow Y$ is non-trivial if $Y \cap X = \emptyset$.
- $Y$ does not appear on both the LHS and the RHS of the FD.
- Non-trivial FDs are given as constraints when designing a database.
- Non-trivial FDs constrain the set of legal relation instances.

In general, $X$ (and $Y$) can be sets of attributes.
E.g., if $X = X_1 \cup X_2 \cup \ldots \cup X_n$, then we can write $X_1X_2\ldots X_n \rightarrow Y$.
RELATIONAL DATABASE DESIGN: FUNCTIONAL DEPENDENCIES

EXERCISES 1, 2
EXERCISE 1

Assume that this table contains the only set of tuples that may appear in a relation R(X, Y, V, W). Which of the following FDs hold in R?

<table>
<thead>
<tr>
<th>tuple</th>
<th>X</th>
<th>Y</th>
<th>V</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x₁</td>
<td>y₁</td>
<td>v₁</td>
<td>w₁</td>
</tr>
<tr>
<td>2</td>
<td>x₁</td>
<td>y₁</td>
<td>v₂</td>
<td>w₂</td>
</tr>
<tr>
<td>3</td>
<td>x₂</td>
<td>y₁</td>
<td>v₁</td>
<td>w₃</td>
</tr>
<tr>
<td>4</td>
<td>x₂</td>
<td>y₁</td>
<td>v₃</td>
<td>w₄</td>
</tr>
</tbody>
</table>

X→X   Yes – trivial (holds in any relation)

X→Y   Yes – for a given X value all Y values are identical

X→V   No  – V values differ for same X value (e.g., tuples 1 & 2)

X→W   No  – W values differ for same X value (e.g., tuples 1 & 2)

Y→X   No  – X values differ for same Y value (e.g., tuples 2 & 3)

W→X   Yes – all W values are different

XV→Y  Yes – X alone determines Y

YV→X  No  – X values differ for same YV value in tuples 1 and 3
In Exercise 1, we assumed that we know all possible records in the table, which is not usually true. In general, by looking at an instance of a relation, we can only tell FDs that are not satisfied.

List 5 FDs that are not satisfied in the table.

- A → C
- B → A
- B → C
- C → A
- AB → C
FUNCTIONAL DEPENDENCIES AND KEYS

- An FD is a generalization of the concept of a *key*. For the relation
  
  \[ \text{PGStudent} \ (\text{studentId}, \text{name}, \text{supervisor}, \text{specialization}) \]

  we can write:
  \[ \text{studentId} \rightarrow \text{name}, \text{supervisor}, \text{specialization} \]

  because the key, \text{studentId}, determines the (single) value of all the attributes (i.e., the entire tuple).

- If two tuples in the PGStudent relation have the same studentId, then they must have the same values on all attributes.

  They must be the same tuple! (Since the relational model does not allow duplicate tuples.)
INFEREN CE RULES FOR FDS

- Given a set of functional dependencies $F$, there are certain other functional dependencies that are *logically implied* by $F$.

- We can find all functional dependencies implied by $F$ by applying the following *inference rules for FDs*.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR1: Reflexivity</td>
<td>If $Y \subseteq X$, then $X \rightarrow Y$ (<em>trivial FD</em>)</td>
</tr>
<tr>
<td>IR2: Augmentation</td>
<td>$X \rightarrow Y \vdash XZ \rightarrow YZ$</td>
</tr>
<tr>
<td>IR3: Transitivity</td>
<td>$X \rightarrow Y$, $Y \rightarrow Z \vdash X \rightarrow Z$</td>
</tr>
<tr>
<td>IR4: Union</td>
<td>$X \rightarrow Y$, $X \rightarrow Z \vdash X \rightarrow YZ$</td>
</tr>
<tr>
<td>IR5: Decomposition</td>
<td>$X \rightarrow YZ \vdash X \rightarrow Y$ and $X \rightarrow Z$</td>
</tr>
<tr>
<td>IR6: Psuedotransitivity</td>
<td>$X \rightarrow Y$, $WY \rightarrow Z \vdash WX \rightarrow Z$</td>
</tr>
</tbody>
</table>

Armstrong’s Axioms (basic rules)

Additional rules (derivable from IR1, IR2 and IR3)
EXAMPLES OF ARMSTRONG’S AXIOMS

**IR1: Reflexivity**  
If $Y \subseteq X$, then $X \rightarrow Y$ *(trivial FD)*

- $\text{name} \rightarrow \text{name}$ since $\text{name} \subseteq \text{name}$
- $\text{name, supervisor} \rightarrow \text{name}$ since $\text{name} \subseteq \{\text{name, supervisor}\}$
- $\text{name, supervisor} \rightarrow \text{supervisor}$ since $\text{supervisor} \subseteq \{\text{name, supervisor}\}$

**IR2: Augmentation**  
$X \rightarrow Y \models XZ \rightarrow YZ$

- $\text{studentId} \rightarrow \text{name}$ *(given)*
- $\text{studentId, supervisor} \rightarrow \text{name, supervisor}$ *(inferred)*

**IR3: Transitivity**  
$X \rightarrow Y$, $Y \rightarrow Z \models X \rightarrow Z$

- $\text{studentId} \rightarrow \text{supervisor}$ *(given)*
- $\text{supervisor} \rightarrow \text{specialization}$ *(given)*
- $\text{studentId} \rightarrow \text{specialization}$ *(inferred)*
Inference rules IR1, IR2 and IR3 are sound and complete.

**Sound:** Given a set of FDs, \( F \), specified on a relation schema \( R \), any FD that we can infer from \( F \) by using IR1 to IR3 will hold in every relation instance \( r \) of \( R \) that satisfies \( F \) (i.e., it is a true FD).

**Complete:** Using IR1, IR2 and IR3 repeatedly to infer FDs will infer all the FDs that can be inferred from \( F \) (i.e., there are no other FDs that are true).

The set of all functional dependencies logically implied by \( F \) is called the closure of \( F \) denoted as \( F^+ \).
EXAMPLE OF FDS IN THE CLOSURE $F^+$


$$F = \{ A \rightarrow B, \quad A \rightarrow C, \quad \text{CG} \rightarrow H, \quad \text{CG} \rightarrow I, \quad B \rightarrow H \}$$

Some members of $F^+$

- $A \rightarrow H$
- $AG \rightarrow I$
- $\text{CG} \rightarrow HI$
The closure of $X$ under $F$ (denoted by $X^+$) is the set of attributes that are functionally determined by $X$ under $F$.

$X \rightarrow Y$ is in $F^+$ $\iff$ $Y \subseteq X^+$

Given `studentId` (e.g., `studentId` is $X$).

If `studentId $\rightarrow$ name` is in $F^+$
then `name` is part of `$studentId^+` (i.e., `$studentId^+ = \{studentId, name, \ldots\}$`)

If `studentId $\rightarrow$ supervisor` is in $F^+$
then `supervisor` is part of `$studentId^+` (i.e., `$studentId^+ = \{studentId, name, supervisor, \ldots\}$`)

Why is `studentId` in $X^+$?
ATTRIBUTE CLOSURE: ALGORITHM

Input:
R a relation schema
F a set of functional dependencies
X ⊂ R (the set of attributes for which we want to compute the closure)

Output:
X⁺ the closure of X w.r.t. F

\[ X^{(0)} := X \]

Repeat
\[ X^{(i+1)} := X^{(i)} \cup Z, \text{ where } Z \text{ is the set of attributes such that there exists } Y \rightarrow Z \text{ in } F, \text{ and } Y \subset X^{(i)} \]
Until \[ X^{(i+1)} := X^{(i)} \]
Return \[ X^{(i+1)} \]

☞ For every attribute Y in Xᵢ, if Y is the LHS of an FD, then add the RHS attributes Z to the closure; repeat until there are no more attributes to add.
Testing for Superkey

To test if $X$ is a superkey, compute $X^+$, and check if $X^+$ contains all attributes of $R$. If $X$ is minimal, then it is a candidate key.

☞ An attribute that is part of any candidate key is called a prime attribute; otherwise it is a non-prime attribute.

Testing Functional Dependencies

To determine whether a functional dependency $X \rightarrow Y$ holds (i.e., if $X \rightarrow Y$ is in $F^+$), compute $X^+$ and check if $Y \subseteq X^+$.

Computing the Closure of $F$

For each subset $X \subseteq R$, compute $X^+$ and, for each $Y \subseteq X^+$, output a functional dependency $X \rightarrow Y$. 
ATTRIBUTE CLOSURE: EXAMPLE

R(A, B, C, D, E, G)

\( F = \{ \text{AB} \rightarrow \text{C}, \text{C} \rightarrow \text{A}, \text{BC} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{D} \rightarrow \text{EG}, \text{BE} \rightarrow \text{C}, \text{CG} \rightarrow \text{BD}, \text{CE} \rightarrow \text{AG} \} \)

Compute the closure of \( X = \text{BD} \) w.r.t \( F \)

\[ X^{(0)} = \{ \text{B, D} \} \]

\[ X^{(1)} = \{ \text{B, D, E, G} \} \quad \text{apply D} \rightarrow \text{EG add E, G to X} \]

\[ X^{(2)} = \{ \text{B, C, D, E, G} \} \quad \text{apply BE} \rightarrow \text{C add C to X} \]

\[ X^{(3)} = \{ \text{A, B, C, D, E, G} \} \quad \text{apply C} \rightarrow \text{A add A to X} \]

\[ X^{(4)} = X^{(3)} \quad \text{no more FDs can be applied} \]

\( \{ \text{B, D} \}^+ = \{ \text{A, B, C, D, E, G} \} \) Is BD a candidate key?
RELATIONAL DATABASE DESIGN: FUNCTIONAL DEPENDENCIES

EXERCISES 3, 4
EXERCISE 3

Given relation schema \( R(X, Y, U, V, W) \) and \( F = \{X \rightarrow Y, UV \rightarrow W, V \rightarrow X\} \)

a) Determine the closure of each attribute.

\[
\begin{align*}
X^+ &= \{X, Y\} \quad \text{(Look for X on LHS of FDs)} \\
Y^+ &= \{Y\} \\
U^+ &= \{U\} \\
V^+ &= \{V, X, Y\} \\
W^+ &= \{W\}
\end{align*}
\]

b) What are the candidate keys of \( R \)?

The candidate key is \( UV \) since \( UV^+ = \{X, Y, U, V, W\} \).
Given relation schema \( R(A, B, C, G, H, I) \) and
\[ F = \{A \rightarrow B, \quad A \rightarrow C, \quad CG \rightarrow H, \quad CG \rightarrow I, \quad B \rightarrow H\} \]

a) Is \( AG \) a (super)key of \( R \) given \( F \)? Yes Why? Since \( AG \rightarrow R \)

Compute \( AG^+ \)
\[
\begin{align*}
AG^{(0)} &= \{A, G\} \\
AG^{(1)} &= \{A, G, B\} \quad (A \rightarrow B \text{ and } A \subseteq \{A, G\}) \\
AG^{(2)} &= \{A, G, B, C\} \quad (A \rightarrow C \text{ and } A \subseteq \{A, G\}) \\
AG^{(3)} &= \{A, G, B, C, H\} \quad (CG \rightarrow H \text{ and } CG \subseteq \{A, G, B, C\}) \\
AG^{(4)} &= \{A, G, B, C, H, I\} \quad (CG \rightarrow I \text{ and } CG \subseteq \{A, G, B, C, H\})
\end{align*}
\]

b) Is \( AG \) a candidate key? Yes Why? Cannot remove \( A \) or \( G \)

c) Does \( A^+ \rightarrow R \) hold? No since \( A^+ = \{A, B, C, H\} \)

d) Does \( G^+ \rightarrow R \) hold? No since \( G^+ = \{G\} \)
Sets of functional dependencies may have redundant functional dependencies that can be inferred from the others. A→C is redundant in: \{A→B, B→C, A→C\} WHY?

Parts of a functional dependency may be redundant. Example of extraneous/redundant attribute in RHS.

\{ A→B, B→C, A→CD \} can be simplified to \{ A→B, B→C, A→D \} WHY?

Example of extraneous/redundant attribute in LHS (partial dependency).

\{ A→B, B→C, AC→D \} can be simplified to \{ A→B, B→C, A→D \} WHY?
A **canonical cover** for $F$ is a set of functional dependencies $F_c$ such that

- $F$ and $F_c$ are **equivalent** (i.e., they have the same closure $F^+$).
- $F_c$ contains **no redundancy** (i.e., no extraneous attributes).
- The LHS of each functional dependency in $F_c$ is **unique**.

☞ Every set of FDs has a canonical cover.

☞ The canonical cover for a set of FDs is **not unique**.

☞ Testing for FD violation using $F_c$ is **usually simpler**.
Recall that two FDs $X \rightarrow Y$, $X \rightarrow Z$, can be converted to $X \rightarrow YZ$ (IR4: Union).

Algorithm to compute the canonical cover of $F$.

1. $F_c = F$
2. Repeat:
   a) Use the union rule (IR4) to replace any FDs in $F_c$ of the form $X \rightarrow Y$ and $X \rightarrow Z$ with $X \rightarrow YZ$.
   b) Find an FD $X \rightarrow Y$ in $F_c$ with an extraneous attribute either in $X$ or in $Y$.
      If an extraneous attribute is found, delete it from $X \rightarrow Y$ in $F_c$.
3. until $F_c$ does not change.

The union rule may become applicable after some extraneous attributes have been deleted, so it must be reapplied.
CANONICAL COVER: EXAMPLE COMPUTATION

R(A, B, C)

\[ F = \{ A \rightarrow BC, \quad B \rightarrow C, \quad A \rightarrow B, \quad AB \rightarrow C \} \]

- **Use union rule (IR4)** to combine \( A \rightarrow BC \) and \( A \rightarrow B \) into \( A \rightarrow BC \)
  - \( F_c \) is now \{A→BC, B→C, AB→C\}

- **Delete extraneous attribute** \( A \) in \( AB \rightarrow C \) because of \( B \rightarrow C \).
  - \( F_c \) is now \{A→BC, B→C\}

- **Delete extraneous attribute** \( C \) in \( A \rightarrow BC \) because of \( A \rightarrow B \) and \( B \rightarrow C \).
  - \( F_c \) is now \{A→B, B→C\}

☞ The canonical cover \( F_c = \{A \rightarrow B, \ B \rightarrow C\} \)
RELATIONAL DATABASE DESIGN: FUNCTIONAL DEPENDENCIES

EXERCISES 5, 6, 7

Upload your completed exercise worksheet to Canvas by 11 p.m. on March 3rd.