Week 5 Workshop

Alice: Your model reduces the most interesting information to something flat and boring.

ing a database.

Riccardo: Also, the system knows so l

s hard to obtain good perfo

Alice: Are you telling me that the m

Vittorio: No, wait, we are going to fix it!

(Foundations of Databases, S. Abiteboul, R. Hull, V. Vianu, Addison-Wesley, 1995)

Australian National University

Housekeeping

1. **Assignment 1 (SQL)** (due 11.59pm 3 Sep 2021)

Housekeeping

1. **Assignment 1 (SQL)** (due 11:59pm, 3 Sep 2021)
    - Which directors have collaborated with at least two different writers?

)

Australian National University

Housekeeping

- **Assignment 1 (SQL)** (due 11:59pm 3 Sep 2021)
  - Which directors have collaborated with at least two different writers?

)

Housekeeping

1. **Assignment 1 (SQL)** (due 11:59pm 3 Sep 2021)
   - Which directors have collaborated with at least two different writers?
)
   -

   -
   order the tuples, syntax issues,etc. (Partia
   - **Do not wait until the last minute to check/s**
   (Refer to the instructions in the assignment s

Housekeeping

1. **Assignment 1 (SQL)** (due 11:59pm 3 Sep 2021)
   - Which directors have collaborated with at least two different writers?
)
   -

   -
     order the tuples, syntax issues,etc. (Partia
   - **Do not wait until the last minute to check/s**
     (Refer to the instructions in the assignment s

2. **More consultation hours in Week 6**
   - Aug 30 (Mon) 2-3 pm
   - Aug 31 (Tue) 2-3 pm
   - Sep 1 (Wed) 8-9 pm

Australian National University

Housekeeping

1. **Assignment 1 (SQL)** (due 11:59pm 3 Sep 2021)
   - Which directors have collaborated with at least two different writers?
)
   -
   -
     order the tuples, syntax issues,etc. (Partia
   - **Do not wait until the last minute to check/s**
     (Refer to the instructions in the assignment s

2. **More consultation hours in Week 6**
   - Aug 30 (Mon) 2-3 pm
   - Aug 31 (Tue) 2-3 pm
   - Sep 1 (Wed) 8-9 pm

Update Anomalies

- What could happen to insert, delete and update operations?

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

Update Anomalies

- What could happen to insert, delete and update operations?

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- **Insertion anomalies:** If inserting a new c

Update Anomalies

- **What could happen to insert, delete and update operations?**

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- **Insertion anomalies:** If inserting a new c (i.e., cannot insert NULL values into integrity constraint).

Update Anomalies

- **What could happen to insert, delete and update operations?**

ENROLMENT

Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- **Insertion anomalies**: If inserting a new c (i.e., cannot insert NULL values into C integrity constraint).
- **Deletion anomalies**: If deleting the enrolled course COMP2400 of Fran, then ...

Australian National University

Update Anomalies

- **What could happen to insert, delete and update operations?**

ENROLMENT

Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- **Insertion anomalies**: If inserting a new c (i.e., cannot insert NULL values into C integrity constraint).
- **Deletion anomalies**: If deleting the enrolled course COMP2400 of Fran, then ... the personal information of Fran, such as DoB, will be lost as well.

Update Anomalies

- **What could happen to insert, delete and update operations?**

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- **Insertion anomalies**: If inserting a new c (i.e., cannot insert NULL values into C integrity constraint).
- **Deletion anomalies**: If deleting the enrolled course COMP2400 of Fran, then ... the personal information of Fran, such as DoB, will be lost as well.
- **Modification anomalies**: If changing the DoB of Michael, then ...

Update Anomalies

- **What could happen to insert, delete and update operations?**

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- **Insertion anomalies**: If inserting a new c (i.e., cannot insert NULL values into C integrity constraint).
- **Deletion anomalies**: If deleting the enrolled course COMP2400 of Fran, then ... the personal information of Fran, such as DoB, will be lost as well.
- **Modification anomalies**: If changing the DoB of Michael, then ... update every tuple that records the DoB of this student.

Australian National University

Update Anomalies?

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
Tom	123456	25/01/1988	COMP2400	2010 S2	6
Tom	123456	25/01/1988	COMP8740	2011 S2	12
					6
					12
					6

STUDENT		
Name	StudentID	DoB
Tom	123456	25/01/1988
Michael	123458	21/04/1985
Fran	123457	11/09/1987

St[illegible]	[illegible]	[illegible]ster
1[illegible]	[illegible]	[illegible] S2
123456	COMP8740	2011 S2
123458	COMP2400	2009 S2
123458	COMP8740	2011 S2
123457	COMP2400	2009 S2

COURSE	
CourseNo	Unit
COMP2400	6
COMP8740	12

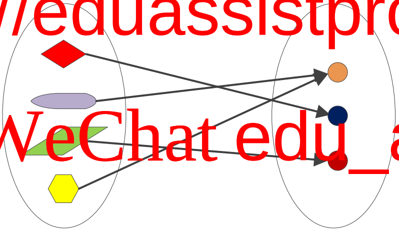
Why Functional Dependencies?

FD

- F ... ional, da
  - **Top down**: start with a relation schema and ... relation schemas in certain normal form (ca
  - **Bottom up**: start with attributes and FDs, a ... schemas (*not popular in practice*).

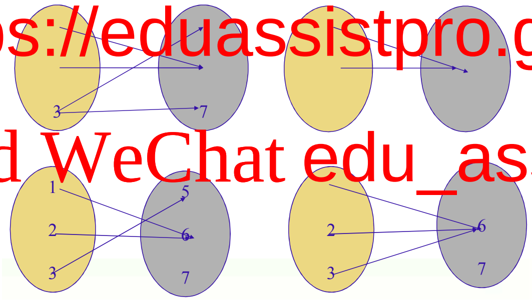
What is "Functional" about Functional Dependencies?

- The notion of functional dependency is very close to the notion of function.
- A (t : X an Y.



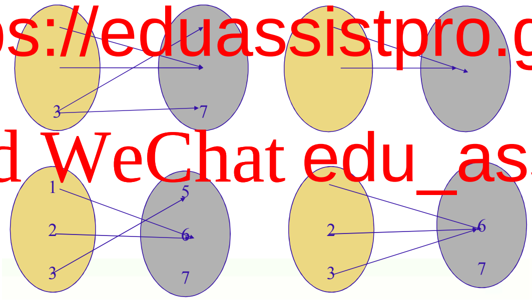
What is "Functional" about Functional Dependencies?

- A (total) **function** $f : X \rightarrow Y$ describes a relationship between two sets X and Y such that each element of X is mapped to a unique element of Y.
- **E**



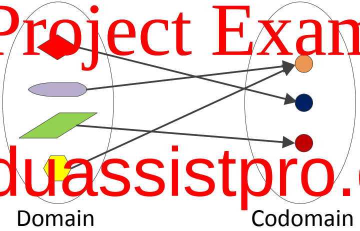
What is “Functional” about Functional Dependencies?

- A (total) **function** $f : X \to Y$ describes a relationship between two sets X and Y such that each element of X is mapped to a unique element of Y.
- **E**

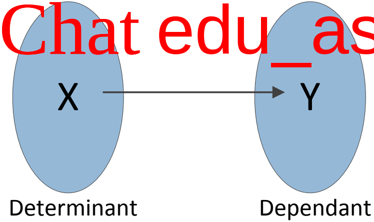


Answer: The ones at the bottom.

Functions vs Functional Dependencies



Functional dependency



Functions vs Functional Dependencies

$$f(x) = x^2$$

Functions vs Functional Dependencies

$$f(x) = x^2$$

3	9
4	16
5	25
6	36
...	...

Functions vs Functional Dependencies

$$f(x) = x^2$$

3	9
4	16
5	25
6	36
...	...

$$X \rightarrow f(x)$$

Australian National University

Formal Definition

- Let R be a relation schema.
  - A **FD** on R is an expression $X \rightarrow Y$ with attribute sets $X, Y \subseteq R$.
  - A relation $r(R)$ **satisfies $X \rightarrow Y$ on R** if, for any two tuples

    f X,

    $t_1[Y] = t_2[$

- A FD is **trivial** if it can *always* be satisfied e
  - $\{A, B\} \rightarrow \{A\}$
  - $\{A, B, C\} \rightarrow \{A, B, C\}$
- **Syntactical convention**: (1) Instead of $\{A, B, C\}$, we may use ABC. (2) $A, B, \ldots$ for individual attributes and $X, Y, \ldots$ for sets of attributes.

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C atisfy th

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C atisfy th

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

1. $ABC \rightarrow AB$

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C atisfy th

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

1. $ABC \rightarrow AB$ Yes.

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C atisfy th

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

1. $ABC \rightarrow AB$ Yes.
2. $ABC \rightarrow D$

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C atisfy th

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

1. $ABC \rightarrow AB$ Yes.
2. $ABC \rightarrow D$ No.

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C ... atisfy th ...

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

1. $ABC \rightarrow AB$ Yes.
2. $ABC \rightarrow D$ No.
3. $E \rightarrow ABCD$

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C [illegible] atisfy th [illegible]

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

1. $ABC \rightarrow AB$ Yes.
2. $ABC \rightarrow D$ No.
3. $E \rightarrow ABCD$ Yes.

Exercise - Functional Dependencies

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- C atisfy th

1	2	3	4	
1	2	2	2	
1	2	3	2	
2	2	2	4	

1. $ABC \rightarrow AB$ Yes.
2. $ABC \rightarrow D$ No.
3. $E \rightarrow ABCD$ Yes.

How to Identify FDs in General?

- A functional dependency specifies a constraint on the relation schema that must hold .

How to Identify FDs in General?

- A functional dependency specifies a constraint on the relation schema that must hold .

- In

How to Identify FDs in General?

- A functional dependency specifies a constraint on the relation schema that must hold .

- In

  (1) **Analyse data requirements**
  Can be provided in the form of discussion wit and/or data requirement specifications

How to Identify FDs in General?

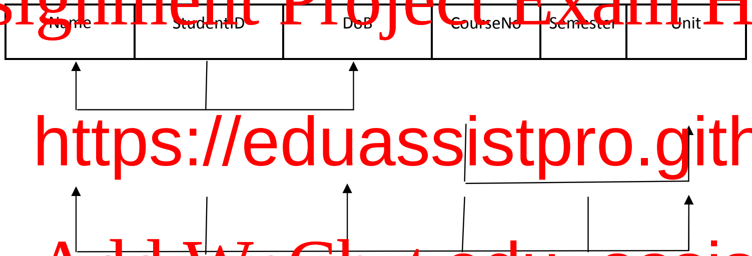
- A functional dependency specifies a constraint on the relation schema that must hold .

- In

(1) **Analyse data requirements**
Can be provided in the form of discussion wit and/or data requirement specifications.

(2) **Analyse sample data**
Useful when application users are unavailable for consultation and/or the document is incomplete.

(1) Analyse Data Requirements and FD Diagram



- StudentID $\rightarrow$ Name, DoB;
- CourseNo $\rightarrow$ Unit;
- StudentID, CourseNo, Semester $\rightarrow$ Name, DoB, Unit.

(2) Analyse Sample Data

- Can you find some FDs on ENROLMENT based on the sample data?

(2) Analyse Sample Data

- Can you find some FDs on ENROLMENT based on the sample data?

ENROLMENT

Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

(2) Analyse Sample Data

- Can you find some FDs on ENROLMENT based on the sample data?

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- We may have:

(2) Analyse Sample Data

- Can you find some FDs on ENROLMENT based on the sample data?

ENROLMENT

Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- We may have:
  - $\{\text{StudentID}\} \rightarrow \{\text{Name, DoB}\}$;

(2) Analyse Sample Data

- Can you find some FDs on ENROLMENT based on the sample data?

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- We may have:
  - {StudentID} → {Name, DoB};
  - {StudentID, Name} → {DoB};

(2) Analyse Sample Data

- Can you find some FDs on ENROLMENT based on the sample data?

ENROLMENT					
Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- We may have:
  - {StudentID} → {Name, DoB};
  - {StudentID, Name} → {DoB};
  - {Name} → {StudentID} ×;
  - ......

(2) Analyse Sample Data

- Can you find some FDs on ENROLMENT based on the sample data?

ENROLMENT

Name	StudentID	DoB	CourseNo	Semester	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- We may have:
  - {StudentID} → {Name, DoB};
  - {StudentID, Name} → {DoB};
  - {Name} → {StudentID} ×;
  - ......

Limitations:
(1) Sample data needs to be a true representation of **all possible values** in the database.
(2) Do we need all FDs?

Australian National University

Inference?

Assignment Project Exam Help

- To design a good database, we need to consider **all possible FDs**.

E

If https://eduassistpro.github.i }, we

ca { } → { }

Add WeChat edu_assist_pr

Inference?

- To design a good database, we need to consider **all possible FDs**.

E

If $\}$, we

ca $\{ \quad \} \rightarrow \{ \quad \}$

If each student works on one project and each proje

then each student must have one project supervis

Inference?

- To design a good database, we need to consider **all possible FDs**.

  E

  If }, we

  ca { } $\rightarrow$ { }

  If each student works on one project and each proje
  then each student must have one project supervis

- **Can we systematically infer all possible FDs?**

Armstrong's Inference Rules (Slides 16-25 will not to be assessed)

- Th

  - **Augmentation rule**: $\{X \rightarrow$

  - **Transitive rule**: $\{X \rightarrow$

- We use the notation $\Sigma \models X \rightarrow Y$ to denote that $X \rightarrow Y$ is **inferred** from the set Σ of functional dependencies.

Rule 1 – Reflexive Rule

- $XY \rightarrow Y$.

ENROLMENT					
	______		______	______	Unit
					6
					12
					6
					12
Fran	123457	11/09/1987			

- **Example**: {StudentID, CourseNo, Semester} where
  - X={StudentID};
  - Y={CourseNo, Semester}.

Rule 2 – Augmentation Rule

- $\{X \to Y\} \models XZ \to YZ$

ENROLMENT					
					Unit
					6
					12
					6
					12
Fran	123457	11/09/1987			

- **Example**:
  $\{\{CourseNo\} \to \{Unit\}\} \models \{CourseNo,$ ... er$\}$,
  where
  - X={CourseNo};
  - Y={Unit};
  - Z={Semester}.

Rule 3 – Transitive Rule

- $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$.

ENROLMENT					
					Unit
					6
					12
					6
					12
Fran	123457	11/09/1987			

- **Example**: $\{\text{StudentID, CourseNo}\} \rightarrow$ … $\rightarrow$ $\{\text{Unit}\} \models \{\text{StudentID, CourseNo}\} \rightarrow \{$
  - $X=\{\text{StudentID, CourseNo}\}$;
  - $Y=\{\text{CourseNo}\}$;
  - $Z=\{\text{Unit}\}$.

Other Derived Rules

- From Armstrong's axioms (i.e. reflexive, augmentation, transitive rules), we can derive the following rules:

Other Derived Rules

- From Armstrong's axioms (i.e., reflexive, augmentation, transitive rules), we can derive the following rules:
  -
  - , then we

    -
    - Y=Name;
    - Z=DoB.

Other Derived Rules

- From Armstrong's axioms (i.e. reflexive, augmentation, transitive rules), we can derive the following rules:
  -
  - , then we

    -
    - Y=Name;
    - Z=DoB.

  - **Decomposition rule:** If $X \rightarrow YZ$, th
  - **Example:** If StudentID $\rightarrow$ Name, Do $\rightarrow$ Name and StudentID $\rightarrow$ DoB, where
    - X=StudentID;
    - Y=Name;
    - Z=DoB.

Australian National University

Example on Armstrong's Inference Rules

- If e
  do

{ProjectNo}→{Supervisor}} ⊨ StudentID Supervisor}

Example on Armstrong's Inference Rules

- If e
  do

$\{$ProjectNo$\}\rightarrow\{$Supervisor$\}\}$ $\models$ StudentID Supervisor$\}$

- This can be proven by using the Transitive rule:

$$\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$$

Example on Armstrong's Inference Rules

- Can we use the following rules to infer FDs, i.e., are they correct?
  (1) $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

Example on Armstrong's Inference Rules

- Can we use the following rules to infer FDs, i.e., are they correct?
  (1) $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

Example on Armstrong's Inference Rules

- Can we use the following rules to infer FDs, i.e., are they correct?

(1) $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

(

Australian National University

Example on Armstrong's Inference Rules

- Can we use the following rules to infer FDs, i.e., are they correct?

  (1) $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

  ([illegible]

X	Y	Z
a	b	c
a	c	d

Example on Armstrong's Inference Rules

- Can we use the following rules to infer FDs, i.e., are they correct?

(1) $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

(

X	Y	Z
a	b	c
a	c	d

(3) $\{X \rightarrow Y\} \models Y \rightarrow X$

Example on Armstrong's Inference Rules

- Can we use the following rules to infer FDs, i.e., are they correct?

(1) $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

(

X	Y	Z
a	b	c
a	c	d

(3) $\{X \rightarrow Y\} \models Y \rightarrow X$

No. See the counter-example below:

X	Y
0	2
1	2

Armstrong's Inference Rules

Assignment Project Exam Help

- **Two questions:**

https://eduassistpro.github.i

Add WeChat edu_assist_pr

[1] William Ward Armstrong: Dependency Structures of Data Base Relationships, page 580-583. IFIP Congress, 1974.

Armstrong's Inference Rules

- **Two questions:**
  -

[1] William Ward Armstrong: Dependency Structures of Data Base Relationships, page 580-583. IFIP Congress, 1974.

Australian National University

Armstrong's Inference Rules

Assignment Project Exam Help

- **Two questions:**
  - https://eduassistpro.github.i
  -
    - ⇝ **completeness** (you can prove anyt

Add WeChat edu_assist_pr

[1] William Ward Armstrong: Dependency Structures of Data Base Relationships, page 580-583. IFIP Congress, 1974.

Armstrong's Inference Rules

- **Two questions:**
  -
  -
    ⇝ **completeness** (you can prove anyt

- **Theorem** (W. W. Armstrong, 1974[1])
  - The Armstrong's inference rules are both **sound** and **complete**.

[1] William Ward Armstrong: Dependency Structures of Data Base Relationships, page 580-583. IFIP Congress, 1974.

Implied Functional Dependencies

- We write Σ^* for all possible FDs **implied** by Σ.

Implied Functional Dependencies

- We write Σ^* for all possible FDs **implied** by Σ.

- Σ^* can be computed using the Armstrong's inference rules.

Implied Functional Dependencies

- We write Σ^* for all possible FDs **implied** by Σ.

- Σ^* can be computed using the Armstrong's inference rules.

- **Why can we compute Σ^* using the Arms**

Implied Functional Dependencies

- We write Σ^* for all possible FDs **implied** by Σ.

- Σ^* can be computed using the Armstrong's inference rules.

- **Why can we compute Σ^* using the Arms**

  Because the Armstrong's inference rules are both **sound** and **complete**.

Implied Functional Dependencies

- We write Σ^* for all possible FDs **implied** by Σ.
- Σ^* can be computed using the Armstrong's inference rules.

- **Why can we compute Σ^* using the Arms**

  Because the Armstrong's inference rules are both **sound** and **complete**.
- Nonetheless, computing Σ^* using the Armstrong's inference rules is **not efficient**.

Implied Functional Dependencies

- Computing Σ^* using the Armstrong's inference rules is **not efficient**.

  Example: Consider a relation schema $R = \{A, B, C, D, E\}$ and a set of FDs
  Σ o
  sh

Implied Functional Dependencies

- Computing Σ^* using the Armstrong's inference rules is **not efficient**.

Example: Consider a relation schema $R = \{A, B, C, D, E\}$ and a set of FDs
Σ o
sh

Implied Functional Dependencies

- Computing Σ^* using the Armstrong's inference rules is **not efficient**.

  Example: Consider a relation schema $R = \{A, B, C, D, E\}$ and a set of FDs Σ … o

  sh…

- **How can we derive the proof more efficiently?**

Implied Functional Dependencies

- Let Σ be a set of FDs. Check whether or not $\Sigma \models X \rightarrow W$ holds?

[2] See Algorithm 15.1 on Page 538 in [Elmasri & Navathe, 7th edition] or Algorithm 1 on Page 555 in [Elmasri & Navathe, 6th edition]

Implied Functional Dependencies

- Let Σ be a set of FDs. Check whether or not $\Sigma \models X \rightarrow W$ holds? We need to

2 See Algorithm 15.1 on Page 538 in [Elmasri & Navathe, 7th edition] or Algorithm 1 on Page 555 in [Elmasri & Navathe, 6th edition]

Implied Functional Dependencies

- Let Σ be a set of FDs. Check whether or not $\Sigma \models X \rightarrow W$ holds? We need to
  1. Compute **the set of all attributes** that are dependent on X, which is

[2] See Algorithm 15.1 on Page 538 in [Elmasri & Navathe, 7th edition] or Algorithm 1 on Page 555 in [Elmasri & Navathe, 6th edition]

Implied Functional Dependencies

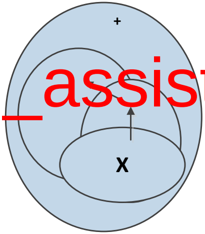
- Let Σ be a set of FDs. Check whether or not $\Sigma \models X \rightarrow W$ holds? We need to
  1. Compute **the set of all attributes** that are dependent on X, which is

[2] See Algorithm 15.1 on Page 538 in [Elmasri & Navathe, 7th edition] or Algorithm 1 on Page 555 in [Elmasri & Navathe, 6th edition]

Implied Functional Dependencies

- Let Σ be a set of FDs. Check whether or not $\Sigma \models X \rightarrow W$ holds? We need to
  1. Compute **the set of all attributes** that are dependent on X, which is
  2.

- Al



  - $X^+ := X$;
  - repeat until no more change on X^+

    - for each $Y \rightarrow Z \in \Sigma$ with $Y \subseteq$ add all the attributes in Z to X^+, i.e., replace X^+ by $X^+ \cup Z$.

[2] See Algorithm 15.1 on Page 538 in [Elmasri & Navathe, 7th edition] or Algorithm 1 on Page 555 in [Elmasri & Navathe, 6th edition]

Implied Functional Dependencies – Example

- Consider a relation scheme $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \rightarrow B, B \rightarrow CD, C \rightarrow E, AF \rightarrow B\}$ on R.
- D

Australian National University

Implied Functional Dependencies – Example

- Consider a relation schema $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \to B, B \to CD, C \to E, AF \to B\}$ on R.

- D

-

$\supseteq$

Implied Functional Dependencies – Example

- Consider a relation schema $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \rightarrow B, B \rightarrow CD, C \rightarrow E, AF \rightarrow B\}$ on R.
- D

$\supseteq$

$\supseteq ACB$ u

Implied Functional Dependencies – Example

- Consider a relation scheme $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \rightarrow B, B \rightarrow CD, C \rightarrow E, AF \rightarrow B\}$ on R.
- D

$\supseteq$
$\supseteq ACB$ u
$\supseteq ACBD$ u

Implied Functional Dependencies – Example

- Consider a relation schema $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \to B, B \to CD, C \to E, AF \to B\}$ on R.
- D

$\supseteq$
$\supseteq ACB$ u
$\supseteq ACBD$ u
$\supseteq ACBDE$ u

Implied Functional Dependencies – Example

- Consider a relation scheme $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \to B, B \to CD, C \to E, AF \to B\}$ on R.
- D

$\supseteq$
$\supseteq ACB$ u
$\supseteq ACBD$ u
$\supseteq ACBDE$ u
$= ACBDE$

Implied Functional Dependencies – Example

- Consider a relation schema $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \to B, B \to CD, C \to E, AF \to B\}$ on R.

- D

$$\begin{aligned} & \supseteq \\ & \supseteq ACB & \text{u} \\ & \supseteq ACBD & \text{u} \\ & \supseteq ACBDE & \text{u} \\ & = ACBDE \end{aligned}$$

2. Then we check that $DE \subseteq (AC)^+$. Hence $\Sigma \models AC \to DE$.

Implied Functional Dependencies – Example

- Consider a relation schema $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \rightarrow B, B \rightarrow CD, C \rightarrow E, AF \rightarrow B\}$ on R.
- D

$$\begin{aligned} & \supseteq \\ & \supseteq ACB && \text{u} \\ & \supseteq ACBD && \text{u} \\ & \supseteq ACBDE && \text{u} \\ & = ACBDE \end{aligned}$$

2. Then we check that $DE \subseteq (AC)^+$. Hence $\Sigma \models AC \rightarrow DE$.

- **Can you quickly tell whether or not $\Sigma \models AC \rightarrow EF$ holds?**

Implied Functional Dependencies – Example

- Consider a relation schema $R = \{A, B, C, D, E, F\}$, a set of FDs $\Sigma = \{AC \rightarrow B, B \rightarrow CD, C \rightarrow E, AE \rightarrow B\}$ on R.
- D

$$
\begin{aligned}
& \supseteq \\
& \supseteq ACB && \text{u} \\
& \supseteq ACBD && \text{u} \\
& \supseteq ACBDE && \text{u} \\
& = ACBDE
\end{aligned}
$$

2. Then we check that $DE \subseteq (AC)^+$. Hence $\Sigma \models AC \rightarrow DE$.

- **Can you quickly tell whether or not $\Sigma \models AC \rightarrow EF$ holds?**

  $\Sigma \models AC \rightarrow EF$ does not hold because $EF \not\subseteq (AC)^+$

Exercise – Implied Functional Dependencies

- Consider a relation schema $R = \{A, B, C, D, E\}$ and a set of functional dependencies $\Sigma = \{A \rightarrow C, B \rightarrow C, CD \rightarrow E\}$ on R.

- D

  2. $\Sigma \models BD \rightarrow AC$ holds

Exercise – Implied Functional Dependencies

- Consider a relation schema $R = \{A, B, C, D, E\}$ and a set of functional dependencies $\Sigma = \{A \rightarrow C, B \rightarrow C, CD \rightarrow E\}$ on R.
- D

  1.
  2. $\Sigma \models BD \rightarrow AC$ holds

- We build the closure for the set of attributes and chec

  1. $(AD)^+ = (ACD)^+ = (ACDE)^+ = A$
     $\Sigma \models AD \rightarrow CE$.
  2. $(BD)^+ = (BCD)^+ = (BCDE)^+ = BCDE$ and $AC \not\subseteq (BD)^+$, hence
     $\Sigma \not\models BD \rightarrow AC$.

Equivalence of Functional Dependencies

- Σ_1 and Σ_2 are **equivalent** if $\Sigma_1^* = \Sigma_2^*$.

Equivalence of Functional Dependencies

- Σ_1 and Σ_2 are **equivalent** if $\Sigma_1^* = \Sigma_2^*$.

- Let $\Sigma_1 = \{X \to Y, Y \to Z\}$ and $\Sigma_2 = \{X$
  $\Sigma_1 \neq \Sigma_2$ but $\Sigma_1^* = \Sigma_2^* = \{X \to Y, Y \to Z,$ alent)

Equivalence of Functional Dependencies

- Σ_1 and Σ_2 are **equivalent** if $\Sigma_1^* = \Sigma_2^*$.
- Let $\Sigma_1 = \{X \to Y, Y \to Z\}$ and $\Sigma_2 = \{X$ ... $\Sigma_1 \neq \Sigma_2$ but $\Sigma_1^* = \Sigma_2^* = \{X \to Y, Y \to Z,$ ... alent)
- If $\Sigma_1 \models \Sigma_2$ and $\Sigma_2 \models \Sigma_1$, are Σ_1 and Σ_2 equivalent?

Equivalence of Functional Dependencies

- Σ_1 and Σ_2 are **equivalent** if $\Sigma_1^* = \Sigma_2^*$.

- Let $\Sigma_1 = \{X \rightarrow Y, Y \rightarrow Z\}$ and $\Sigma_2 = \{X$ $\Sigma_1 \neq \Sigma_2$ but $\Sigma_1^* = \Sigma_2^* = \{X \rightarrow Y, Y \rightarrow Z,$ alent)
- If $\Sigma_1 \models \Sigma_2$ and $\Sigma_2 \models \Sigma_1$, are Σ_1 and Σ_2 equivalent? Yes.

Equivalence of Functional Dependencies

- Σ_1 and Σ_2 are **equivalent** if $\Sigma_1^* = \Sigma_2^*$.

- Let $\Sigma_1 = \{X \to Y, Y \to Z\}$ and $\Sigma_2 = \{X$
  $\Sigma_1 \neq \Sigma_2$ but $\Sigma_1^* = \Sigma_2^* = \{X \to Y, Y \to Z,$ alent)
- If $\Sigma_1 \models \Sigma_2$ and $\Sigma_2 \models \Sigma_1$, are Σ_1 and Σ_2 equivalent? Yes.
- **Questions:** Can we find the **minimal** one among equivalent sets of FDs?

Minimal Cover – The Hard Part!

Australian National University

Minimal Cover – The Hard Part!

- Let Σ be a set of FDs. A **minimal cover** Σ_m of Σ is a set of FDs such that
    -

Minimal Cover – The Hard Part!

- Let Σ be a set of FDs. A **minimal cover** Σ_m of Σ is a set of FDs such that

$_1$ $_k$ $_m$ ith

$X \rightarrow A_1, \ldots, X \rightarrow A_k$;

Minimal Cover – The Hard Part!

- Let Σ be a set of FDs. A **minimal cover** Σ_m of Σ is a set of FDs such that

$_1$ $_k$ $_m$ ith $X \rightarrow A_1, \ldots, X \rightarrow A_k$;

3. **Determinant:** each FD has as few attribu possible, i.e., for each FD $X \rightarrow A$ in f X to see if we can replace $X \rightarrow A$ with $(X - B) \rightarrow A$ in Σ_m;

Australian National University

Minimal Cover – The Hard Part!

- Let Σ be a set of FDs. A **minimal cover** Σ_m of Σ is a set of FDs such that

1.
2. $_1$ $_k$ $_m$ ith
   $X \to A_1, \ldots, X \to A_k$;

3. **Determinant:** each FD has as few attribu
   possible, i.e., for each FD $X \to A$ in f X to
   see if we can replace $X \to A$ with $(X - B) \to A$ in Σ_m;
4. Remove a FD from Σ_m if it is redundant.

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  -
  -

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \rightarrow D$ can be replaced by $A \quad D$?

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

3. check if $AB \rightarrow D$ can be replaced by $A \quad D$?
   - $\Sigma = \{B \rightarrow A, D \rightarrow A, \mathbf{AB \rightarrow D} \quad \mathbf{D}\}$

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \rightarrow D$ can be replaced by A D?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A,$ **AB → D** **D**$\}$
     - check whether $\Sigma^* = \Sigma_1^*$?

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \rightarrow D$ can be replaced by A D?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A,$ **$AB \rightarrow D$** **D**$\}$
     - check whether $\Sigma^* = \Sigma_1^*$? (we h

Australian National University

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \rightarrow D$ can be replaced by $A \quad D$?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A,$ **AB → D** **D**$\}$
     - check whether $\Sigma^* = \Sigma_1^*$? (we h
     - check $\Sigma \models$ **A → D**?

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

3. check if $AB \rightarrow D$ can be replaced by $A \quad D$?
   - $\Sigma = \{B \rightarrow A, D \rightarrow A, \mathbf{AB} \rightarrow \mathbf{D} \quad \mathbf{D}\}$
   - check whether $\Sigma^* = \Sigma_1^*$? (we h
   - check $\Sigma \models \mathbf{A} \rightarrow \mathbf{D}$?

     If $\Sigma \models \mathbf{A} \rightarrow \mathbf{D}$, then $\Sigma \models \Sigma_1 \quad \Sigma_1 \models \Sigma \quad \Sigma \quad = \Sigma_1^*$.

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \to A, D \to A, AB \to D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.
  3. check if $AB \to D$ can be replaced by $A \quad D$?
     - $\Sigma = \{B \to A, D \to A, \mathbf{AB \to D} \quad \mathbf{D}\}$
     - check whether $\Sigma^* = \Sigma_1^*$? (we h
     - check $\Sigma \models \mathbf{A \to D}$?

       If $\Sigma \models \mathbf{A \to D}$, then $\Sigma \models \Sigma_1 \quad \Sigma_1 \models \Sigma \quad \Sigma \quad = \Sigma_1^*$.

       If $\Sigma \not\models \mathbf{A \to D}$, then $\Sigma^* \neq \Sigma_1^*$.

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \rightarrow D$ can be replaced by $A \rightarrow D$?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A, \mathbf{AB \rightarrow D}$ $\mathbf{D}\}$
     - check whether $\Sigma^* = \Sigma_1^*$? (we h
     - check $\Sigma \models \mathbf{A \rightarrow D}$?

       If $\Sigma \models \mathbf{A \rightarrow D}$, then $\Sigma \models \Sigma_1$ $\Sigma_1 \models \Sigma$ $\Sigma = \Sigma_1^*$.

       If $\Sigma \not\models \mathbf{A \rightarrow D}$, then $\Sigma^* \neq \Sigma_1^*$.
     - $\Sigma \not\models \mathbf{A \rightarrow D}$ because $D \not\subseteq (A)^+$.

     No. $AB \rightarrow D$ cannot be replaced by $A \rightarrow D$.

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \to A, D \to A, AB \to D\}$, we can compute the minimal cover of Σ as follows:

3. check if $AB \to D$ can be replaced by $B \quad D$?

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

3. check if $AB \rightarrow D$ can be replaced by $B \quad D$?
   - $\Sigma = \{B \rightarrow A, D \rightarrow A, \mathbf{AB \rightarrow D} \quad \mathbf{D}\}$

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.
  3. check if $AB \rightarrow D$ can be replaced by $B \quad D$?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A,$ **$AB \rightarrow D$** **D**$\}$
     - check whether $\Sigma^* = \Sigma_2^*$?

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \to A, D \to A, AB \to D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \to D$ can be replaced by $B \quad D$?
     - $\Sigma = \{B \to A, D \to A,$ **$AB \to D$** **D**$\}$
     - check whether $\Sigma^* = \Sigma_2^*$?

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \to A, D \to A, AB \to D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.
  3. check if $AB \to D$ can be replaced by B D?
     - $\Sigma = \{B \to A, D \to A,$ $\mathbf{AB \to D}$ $\mathbf{D}\}$
     - check whether $\Sigma^* = \Sigma_2^*$? (we h

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

  3. check if $AB \rightarrow D$ can be replaced by B D?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A,$ **AB → D** **D**$\}$
     - check whether $\Sigma^* = \Sigma_2^*$? (we h
     - check $\Sigma \models$ **B → D**?

Australian National University

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \rightarrow D$ can be replaced by $B \quad D$?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A, \mathbf{AB \rightarrow D} \quad \mathbf{D}\}$
     - check whether $\Sigma^* = \Sigma_2^*$? (we h

     - check $\Sigma \models \mathbf{B \rightarrow D}$?

       If $\Sigma \models \mathbf{B \rightarrow D}$, then $\Sigma \models \Sigma_2 \quad \Sigma_2 \models \Sigma \quad \Sigma \; = \Sigma_2^*$.

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \rightarrow D$ can be replaced by $B \quad D$?
     - $\Sigma = \{B \rightarrow A, D \rightarrow A, \mathbf{AB \rightarrow D} \quad \mathbf{D}\}$
     - check whether $\Sigma^* = \Sigma_2^*$? (we h
     - check $\Sigma \models \mathbf{B \rightarrow D}$?

       If $\Sigma \models \mathbf{B \rightarrow D}$, then $\Sigma \models \Sigma_2 \quad \Sigma_2 \models \Sigma \quad \Sigma \quad = \Sigma_2^*$.

       If $\Sigma \not\models \mathbf{B \rightarrow D}$, then $\Sigma^* \neq \Sigma_2^*$

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \to A, D \to A, AB \to D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. check if $AB \to D$ can be replaced by B D?
     - $\Sigma = \{B \to A, D \to A, \mathbf{AB \to D}$ $\mathbf{D}\}$
     - check whether $\Sigma^* = \Sigma_2^*$? (we h
     - check $\Sigma \models \mathbf{B \to D}$?

       If $\Sigma \models \mathbf{B \to D}$, then $\Sigma \models \Sigma_2$ $\Sigma_2 \models \Sigma$ Σ $= \Sigma_2^*$.

       If $\Sigma \not\models \mathbf{B \to D}$, then $\Sigma^* \neq \Sigma_2^*$
     - $\Sigma \models \mathbf{B \to D}$ because $D \subseteq (B)^+$.

     Yes. $AB \to D$ can be replaced by $B \to D$.

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. $AB \rightarrow D$ can be replaced by $B \rightarrow D$

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

3. $AB \rightarrow D$ can be replaced by $B \rightarrow D$
4. Look for a redundant FD in $\{B \rightarrow A,$

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. $AB \rightarrow D$ can be replaced by $B \rightarrow D$
  4. Look for a redundant FD in $\{B \rightarrow A,$
     - check whether $B \rightarrow A$ is redun

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:

3. $AB \rightarrow D$ can be replaced by $B \rightarrow D$

4. Look for a redundant FD in $\{B \rightarrow A,$
   - check whether $B \rightarrow A$ is redun
   - $B \rightarrow A$ is redundant because $\{D \rightarrow A, B \rightarrow D\} \models B \rightarrow A$;

Minimal Cover - Examples

- Given the set of FDs $\Sigma = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$, we can compute the minimal cover of Σ as follows:
  1.
  2.

  3. $AB \rightarrow D$ can be replaced by $B \rightarrow D$
  4. Look for a redundant FD in $\{B \rightarrow A,$

     - check whether $B \rightarrow A$ is redun
     - $B \rightarrow A$ is redundant because $\{D \rightarrow A, B \rightarrow D\} \models B \rightarrow A$;

Therefore, the minimal cover of Σ is $\{D \rightarrow A, B \rightarrow D\}$.

Minimal Cover

- **Theorem:**

The minimal cover of a set of functional dependencies Σ always

Minimal Cover

- **Theorem:**

The minimal cover of a set of functional dependencies Σ always

- **Examples:** Consider the following set of functi

$\Sigma = \{A \rightarrow BC, B \rightarrow C,$

Australian National University

Minimal Cover

- **Theorem:**

> The minimal cover of a set of functional dependencies Σ always

- **Examples:** Consider the following set of functi

$$\Sigma = \{A \to BC, B \to C,$$

Σ has two different minimal covers:

- $\Sigma_1 = \{A \to B, B \to C, C \to A\}$
- $\Sigma_2 = \{A \to C, C \to B, B \to A\}$

Minimal Cover

- **Theorem:**

The minimal cover of a set of functional dependencies Σ always

- **Examples:** Consider the following set of functi

$$\Sigma = \{A \rightarrow BC, B \rightarrow C,$$

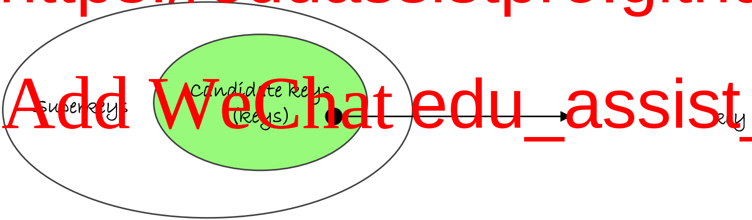
Σ has two different minimal covers:

  - $\Sigma_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$
  - $\Sigma_2 = \{A \rightarrow C, C \rightarrow B, B \rightarrow A\}$

- The algorithm in the previous slide can find one, but not all minimal covers of a set of functional dependencies Σ.

Finding Keys

- Gi



Finding Keys

- **Fact**: A key K of R always defines a FD $K \to R$.

[3] It extends Algorithm 15.2(a) in [Elmasri & Navathe, 7th edition, pp. 542], or Algorithm 2(a) or in Algorithm 2(a) in [Elmasri & Navathe, 6th edition pp. 558] to finding all keys of R

Finding Keys

- **Fact**: A key K of R always defines a FD $K \rightarrow R$.
- **Algorithm**[3]:

In

o

[3] It extends Algorithm 15.2(a) in [Elmasri & Navathe, 7th edition, pp. 542], or Algorithm 2(a) or in Algorithm 2(a) in [Elmasri & Navathe, 6th edition pp. 558] to finding all keys of R

Finding Keys

- **Fact**: A key K of R always defines a FD $K \rightarrow R$.
- **Algorithm**[3]:

In

o

[3] It extends Algorithm 15.2(a) in [Elmasri & Navathe, 7th edition, pp. 542], or Algorithm 2(a) or in Algorithm 2(a) in [Elmasri & Navathe, 6th edition pp. 558] to finding all keys of R

Finding Keys

- **Fact**: A key K of R always defines a FD $K \to R$.
- **Algorithm**[3]:

  In

  o

  - for every subset X of the relation R, compute its closure X^+

[3] It extends Algorithm 15.2(a) in [Elmasri & Navathe, 7th edition, pp. 542], or Algorithm 2(a) or in Algorithm 2(a) in [Elmasri & Navathe, 6th edition pp. 558] to finding all keys of R

Finding Keys

- **Fact**: A key K of R always defines a FD $K \rightarrow R$.
- **Algorithm**[3]:

In

O

- for every subset X of the relation R, compute its closure X^+
- if $X^+ = R$, then X is a superkey.

[3] It extends Algorithm 15.2(a) in [Elmasri & Navathe, 7th edition, pp. 542], or Algorithm 2(a) or in Algorithm 2(a) in [Elmasri & Navathe, 6th edition pp. 558] to finding all keys of R

Finding Keys

- **Fact**: A key K of R always defines a FD $K \to R$.
- **Algorithm**[3]:

  In

  O

  - for every subset X of the relation R, compute its closure X^+
  - if $X^+ = R$, then X is a superkey.
  - if no proper subset Y of X with Y^+

[3] It extends Algorithm 15.2(a) in [Elmasri & Navathe, 7th edition, pp. 542], or Algorithm 2(a) or in Algorithm 2(a) in [Elmasri & Navathe, 6th edition pp. 558] to finding all keys of R

Finding Keys

- **Fact**: A key K of R always defines a FD $K \rightarrow R$.
- **Algorithm**[3]:

  In

  O

  - for every subset X of the relation R, compute its closure X^+
  - if $X^+ = R$, then X is a superkey.
  - if no proper subset Y of X with Y^+

- A **prime attribute** is an attribute occurring in a key, and a **non-prime attribute** is an attribute that is not a prime attribute.

[3] It extends Algorithm 15.2(a) in [Elmasri & Navathe, 7th edition, pp. 542], or Algorithm 2(a) or in Algorithm 2(a) in [Elmasri & Navathe, 6th edition pp. 558] to finding all keys of R

Exercises - Keys and Minimal Cover

- Consider RENTAL = {CustID, CustName, PropertyNo, DateStart, Owner} and the following set Σ of FDs:
  -
  -
  -
  - $\{$CustID, StartDate$\} \rightarrow \{$PropertyNo
  - $\{$Owner$\} \rightarrow \{$PropertyNo$\}$

- **Questions:**
  1. What are the keys of RENTAL?
  2. What is a minimal cover of Σ?

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  -
  -

- W ENTAL

Australian National University

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  -
  -

- W ENTAL

- **Solution**: Check $(X)^+$ for every subset of
  - O never appears in the dependent of any F key.

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  -
  -

- W ENTAL
- **Solution**: Check $(X)^+$ for every subset of

  - O never appears in the dependent of any F key.
  - $(O)^+ = OP$

Exercises - Keys and Minimal Cover

- Consider RENTAL = {CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  -
  -

- W ENTAL
- **Solution**: Check $(X)^+$ for every subset of

  - O never appears in the dependent of any F key.
  - $(O)^+ = OP$
  - $(CO)^+ = CPNDO$, $(DO)^+ = CPNDO$ ...
  - Thus, {CustID, Owner} and {Owner, DateStart} are the keys.

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  - R= C, N, P, D, O , and
  -
- W

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  - R= C, N, P, D, O , and
  -
- W

- **Solution**:
  1. start from Σ

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  - R= C, N, P, D, O , and
  -
- W

- **Solution**:
  1. start from Σ;
  2. check whether all the FDs in Σ hav hand side (look good);

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  - R= C, N, P, D, O , and
  -
- W

- **Solution**:
  1. start from Σ;
  2. check whether all the FDs in Σ hav
     hand side (look good);
  3. determine if PD → C, CP → D and CD → P have any redundant attribute on the left hand side (look good);

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  - R= C, N, P, D, O , and
  -
- W

- **Solution**:
  1. start from Σ;
  2. check whether all the FDs in Σ hav
     hand side (look good);
  3. determine if PD → C, CP → D and CD → P have any redundant attribute on the left hand side (look good);
  4. look for a redundant FD in Σ (none of FDs in Σ are redundant);

Exercises - Keys and Minimal Cover

- Consider RENTAL={CustID, CustName, PropertyNo, DateStart, Owner} and its FDs in the abbreviated form as
  - R= C, N, P, D, O , and
  -
- W

- **Solution**:
  1. start from Σ;
  2. check whether all the FDs in Σ hav
     hand side (look good);
  3. determine if PD → C, CP → D and CD → P have any redundant attribute on the left hand side (look good);
  4. look for a redundant FD in Σ (none of FDs in Σ are redundant);

Therefore, Σ is a minimal cover itself.

Accommodation Database

- Consider the following:
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - ROOM(roomNo, hotelNo, type, price) with PK {roomNo, hotelNo}
  - stNo}
  -

Accommodation Database

- Consider the following:
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - ROOM(roomNo, hotelNo, type, price) with PK roomNo, hotelNo}
  - stNo}
  -

- We have some requirements on BOOKING

R1 A booking can be made for one day only.

R2 A guest can make several bookings in a hotel f

R3 A guest cannot make two or more bookings in t same day.

R4 A guest can make two or more bookings in different hotels for the same day.

R5 A room in any hotel can only be booked by one guest on the same date, i.e., no *double-booking*.

How to Identify FDs?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}
  -

- Which functional dependency does the followin

R1 A booking can be made for one day only.

How to Identify FDs?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}

- Which functional dependency does the followin

  R1 A booking can be made for one day only.

  $\hookrightarrow$ {guestNo, hotelNo, roomNo} $\rightarrow$ {

How to Identify FDs?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}
  -

- Which functional dependency does the followin

  R1 A booking can be made for one day only.

  ↪ {guestNo, hotelNo, roomNo} → {

Australian National University

How to Identify FDs?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}
  -

- Which functional dependency does the followin

R1 A booking can be made for one day only.

$\hookrightarrow$ {guestNo, hotelNo, roomNo} $\rightarrow$ {

guestNo	hotelNo	roomNo	Date
001	H1	R101	28/08/2020
001	H1	R101	29/08/2020

How to Identify FDs?

- Consider the following:
  -
  - lNo}
  - stNo}
  - BOOKING(guestNo, hotelNo, date, roomNo) with PK ?

- Which functional dependency does the followin

R2 A guest can make several bookings in a hotel f

Australian National University

How to Identify FDs?

- Consider the following:
  -
  - lNo}
  - stNo}
  - BOOKING(guestNo, hotelNo, date, roomNo) with PK ?

- Which functional dependency does the followin

R2 A guest can make several bookings in a hotel f

None

Australian National University

How to Identify FDs?

- Consider the following:
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - ROOM(roomNo, hotelNo, type, price) with PK roomNo, hotelNo}
  - stNo}
  -

- Which functional dependency does the followin

R3 A guest cannot make two or more bookings in t same day.

How to Identify FDs?

- Consider the following:
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - ROOM(roomNo, hotelNo, type, price) with PK roomNo, hotelNo}
  - stNo}
  -

- Which functional dependency does the followin

  R3 A guest cannot make two or more bookings in t same day.

  $\hookrightarrow$ {guestNo, hotelNo, date} $\rightarrow$ {ro }

How to Identify FDs?

- Consider the following:
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - ROOM(roomNo, hotelNo, type, price) with PK roomNo, hotelNo}
  - stNo}
  -

- Which functional dependency does the followin

R3 A guest cannot make two or more bookings in t same day.

$\hookrightarrow$ {guestNo, hotelNo, date} $\rightarrow$ {ro }

guestNo	hotelNo	roomNo	Date
001	H1	R101	29/08/2020
001	H1	R102 ×	29/08/2020

How to Identify FDs?

- Consider the following:
  -
  - lNo}
  - stNo}
  - OOKING

- Which functional dependency does the followin

R4 A guest can make two or more bookings in diff same day.

Australian National University

How to Identify FDs?

- Consider the following:
  -
  - lNo}
  - stNo}
  - OOKING

- Which functional dependency does the followin

R4 A guest can make two or more bookings in diff same day.

None

Australian National University

How to Identify FDs?

- Consider the following:
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - ROOM(roomNo, hotelNo, type, price) with PK {roomNo, hotelNo}
  - stNo}
  -

- Which functional dependency does the followin

R5 A room in any hotel can only be booked by one g
date, i.e., no *double-booking*.

$\hookrightarrow$ {hotelNo, date, roomNo} $\rightarrow$ {gu }

Australian National University

How to Identify FDs?

- Consider the following:
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - ROOM(roomNo, hotelNo, type, price) with PK roomNo, hotelNo}
  - stNo}
  -

- Which functional dependency does the followin

  R5 A room in any hotel can only be booked by one g
  date, i.e., no *double-booking*.

  $\hookrightarrow$ {hotelNo, date, roomNo} $\rightarrow$ {gu }

guestNo	hotelNo	roomNo	Date
001	H1	R101	29/08/2020
002 ×	H1	R101	29/08/2020

How to Find Candidate Keys?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}
  -

- FDs on BOOKING
  - {guestNo, hotelNo, date} → {room
  - {hotelNo, date, roomNo} → {guest

How to Find Candidate Keys?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}
  -

- FDs on BOOKING
  - {guestNo, hotelNo, date} → {room
  - {hotelNo, date, roomNo} → {guest

- Candidate keys on BOOKING
  - {guestNo, hotelNo, date}
  - {hotelNo, date, roomNo}

How to Identify FDs?

- Consider BOOKING(guestNo, hotelNo, date, roomNo) and the following changes:

~~... tel for the same day.~~

~~R4~~ ~~A guest can make two or more boo ... e same day.~~

R5 ~~A room in any hotel can only be booked by one g~~ ... date, i.e., no *double-booking*.

R6 A guest is not allowed to make more than one booking for the same day even in the different hotels.

How to Identify FDs?

- Consider the following:
  -
  - lNo}
  - stNo}
  -

- Which functional dependency does the followin

R6 A guest is not allowed to make more than one b day even in the different hotels.

How to Identify FDs?

- Consider the following:
  -
  - lNo}
  - stNo}
  -

- Which functional dependency does the followin

R6 A guest is not allowed to make more than one b day even in the different hotels.

$\hookrightarrow$ {guestNo, date} $\rightarrow$ {hotelNo, roomNo}

How to Find Candidate Keys?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}
  -

- FDs on BOOKING
  - {hotelNo, date, roomNo} → {guest
  - {guestNo, date} → {hotelNo, roomN

How to Find Candidate Keys?

- Consider the following
  - HOTEL(hotelNo, hotelName, city) with PK {hotelNo}
  - lNo}
  - stNo}
  -

- FDs on BOOKING
  - {hotelNo, date, roomNo} → {guest
  - {guestNo, date} → {hotelNo, roomN

- Candidate keys on BOOKING
  - {hotelNo, date, roomNo}
  - {guestNo, date}

(credit cookie) Kurt Gödel and Incompleteness Theorem

Kurt Gödel (1906-1978)

Armstrong's Inference Rules

- **Two questions:**
  -
  -
    ⇝ **completeness (you can prove anyt**

- **Theorem** (W. W. Armstrong)
  - The Armstrong's inference rules are both **sound** and **complete**.

Hilbert's program (1920s)

- **Formulation of mathematics**: formalize all true mathematical statements
- **Completeness**: all true mathematical statements can be proved
- **Consistency**: no contradiction can be obtained in the formalism
- **D**

Hilbert's program (1920s)

- **Formulation of mathematics**: formalize all true mathematical statements
- **Completeness**: all true mathematical statements can be proved
- **Consistency**: no contradiction can be obtained in the formalism
- **D**

David Hilbert (1862-1943)

We must know. We will know.

Kurt Gödel and Incompleteness Theorem

Kurt Göde
(1906-197

- **Theorem** (Kurt Gödel, 1931)
  For any computable axiomatic system that is powerful enough to describe the arithmetic of the natural numbers, **there will always be at least one true but unprovable statement**.

Kurt Gödel and Gödel Prize

Kurt Gödel
(1906-1978)

Jo

- Kurt Gödel's achievement in modern logic is sing
indeed it is more than a monument, it is a landmark which will remain visible far in space and time. — **John von Neumann**

Kurt Gödel and Gödel Prize

Kurt Gödel
(1906-1978)

Jo

- Kurt Gödel's achievement in modern logic is sing indeed it is more than a monument, it is a landmark which will remain visible far in space and time. — **John von Neumann**
- The **Gödel prize** became an annual prize for outstanding papers in the area of theoretical computer science since 1993.