Data Bases
Functional dependencies

Laurent D’ORAZIO

Université de Rennes

Introduction

- Context
  - Constraints that apply to a relation
  - “Functional dependency” most common constraint, generalizing the idea of a key for a relation

- Notes
  - FD are based on knowledge of real word
  - All instances of relation must adhere

- Use cases
  - Data storage and compression
  - Query optimization

Introduction
Running example: college application information

- Relations

```
Student(SSN, SName, address, HScode, HSname, HScity, GPA, priority)
Apply(SSN, Cname, state, date, major)
```

- Note
  - `priority` is determined by `GPA`

```
GPA >= 18 priority 1
16 <= GPA < 18 priority 2
GPA < 16 priority 3
```

  - Two tuples with same `GPA` have the same `priority`

- Formalism
  - $\forall\, t, u \ \in Student: t.gpa = u.gpa \ \Rightarrow t.priority = u.priority$
  - GPA → priority

Introduction
Functional dependency

- Generalization
  - $\forall\, t, u \ \in R: t.A = u.A \ \Rightarrow t.B = u.B$
  - A → B

  - $\forall\, t, u \ \in R: t[A1, \ldots, An] = u[A1, \ldots, An] \Rightarrow t[B1, \ldots, Bm] = u[B1, \ldots, Bm]$
  - A1, A2, …An → B1, B2, …, Bm

$\bar{A}$ $\bar{B}$

$\bar{A}$	$\bar{\mathrm{B}}$	$\bar{\mathrm{C}}$
$\bar{\mathrm{a}}$	$\bar{\mathrm{b}}$	$\overline{\mathrm{c1}}$
$\bar{\mathrm{a}}$	$\bar{\mathrm{b}}$	$\overline{\mathrm{c2}}$
…	…	…

Running example (1)

```
Student(SSN, SName, address, HScode, HSname, HScity, GPA, priority)
```

SSN → Sname

SSN → address

HScode → HSname, HScity

HSname, HScity → HScode

SSN → GPA

GPA → priority

SSN → priority

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	14	3

Running example (2)

```
Apply(SSN, Cname, state, date, major)
```

cName → date

SSN, cName -> major

SSN → state

FD and keys

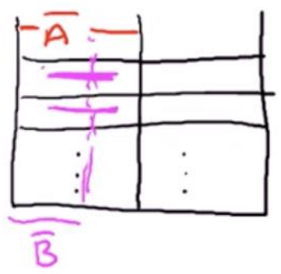
- Definition
  - $\bar{A}$ is a key
  - $\Leftrightarrow$
  - Relations with no duplication
  - $\bar{A} \longrightarrow all\ attributes$

$\bar{A}$	$\bar{\mathrm{B}}$
$\bar{\mathrm{a}}$	$\bar{\mathrm{b}}$
$\bar{\mathrm{a}}$	$\bar{\mathrm{b}}$
…	…

=

FD
Definitions (1)

- Trivial FD
  - $\bar{A} \longrightarrow \bar{B}$
  - $\bar{B} \subseteq \bar{A}$



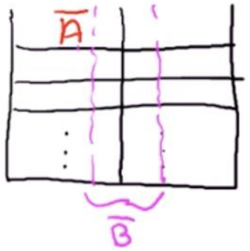
Source: Jennifer Widom, Stanford

- Example
  - HSname, HScity → HSname

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

FD
Definitions (2)

- Nontrivial FD
  - $\bar{A} \longrightarrow \bar{B}$
  - $\bar{B} \nsubseteq \bar{A}$



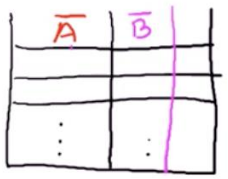
Source: Jennifer Widom, Stanford

- Example
  - HSname, HScity → Hsname, HScode

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

FD
Definitions (3)

- Completely nontrivial FD
  - $\bar{A} \longrightarrow \bar{B}$
  - $\overline{\mathrm{A}} \cap \overline{\mathrm{B}} = \emptyset$



Source: Jennifer Widom, Stanford

- Example
  - HScode → HSname, HScity

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

Rules on FD
Splitting rule (1)

- Splitting rule
  - $\bar{A} \rightarrow B_1B_2...B_m$
  - $\Longrightarrow$
  - $\bar{A} \rightarrow B_i$ for i=1, 2, …, m.

- Example
  - SSN → Sname, address
  - $\Longrightarrow$
  - SSN → Sname
  - SSN → address

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

Rules on FD
Splitting rule (2)

- ## Question
  - Is it possible to split on the left-hand side?
  - No!

- ## Example
  - HSname, HScity → HScode
  - $\not\Rightarrow$
  - HSname → HScode

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
…							
1006	Florian	Talence	HS05	Montaigne	Bordeaux	12	4
…							

Rules on FD
Combining rule

- Combining rule
  - $\bar{A} \rightarrow B_i$ for i=1, 2, ..., m
  - $\Longrightarrow$
  - $\bar{A} \rightarrow B_1B_2...B_m$

- Example
  - SSN → Sname
  - SSN → address
  - $\Longrightarrow$
  - SSN → Sname, address

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

Rules on FD
Trivial-dependency rules (1)

- Reminder
  - $\bar{A} \longrightarrow \bar{B}$ when $\bar{B} \subseteq \bar{A}$

- Rule 1
  - $\bar{A} \longrightarrow \bar{B}$
  - $\Longrightarrow$
  - $\bar{A} \longrightarrow \bar{A} \cup \bar{B}$

- Example
  - GPA → priority
  - $\Longrightarrow$
  - GPA → priority, GPA

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

Rules on FD
Trivial-dependency rules (2)

- Reminder
  - $\bar{A} \longrightarrow \bar{B}$ when $\bar{B} \subseteq \bar{A}$

- Rule 2
  - $\bar{A} \longrightarrow \bar{B}$
  - $\Longrightarrow$
  - $\bar{A} \longrightarrow \bar{A} \cap \bar{B}$

- Example
  - HSname, HScity → HScode, Hscity
  - $\Longrightarrow$
  - HSname, HScity → HScity

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

Rules on FD
Trivial-dependency rules (3)

- Reminder
  - $\bar{A} \longrightarrow \bar{B}$ when $\bar{B} \subseteq \bar{A}$

- Rule 2 (also implied by splitting rule)
  - $\bar{A} \longrightarrow \bar{B}$
  - $\Longrightarrow$
  - $\bar{A} \longrightarrow \bar{A} \cap \bar{B}$

- Example
  - HSname, HScity → HScode, Hscity
  - $\Longrightarrow$
  - HSname, HScity → HScity
  - (HSname, HScity → HScode)

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

Rules on FD
Transitive rule (1)

- Transitive rule
  - $\bar{A} \longrightarrow \bar{B}$
  - $\bar{\mathrm{B}} \longrightarrow \bar{\mathrm{C}}$
  - $\Longrightarrow$
  - $\bar{\mathrm{A}} \longrightarrow \bar{\mathrm{C}}$

- Demonstration

$\bar{A}$	$\bar{\mathrm{B}}$	$\bar{\mathrm{C}}$	$\bar{\mathrm{D}}$
$\bar{\mathrm{a}}$	$\bar{\mathrm{b}}$	$\bar{\mathrm{c}}$	
$\bar{\mathrm{a}}$	$\bar{\mathrm{b}}$	$\bar{\mathrm{c}}$	
…	…	…	…

Rules on FD
Transitive rule (2)

- Transitive rule
  - $\bar{A} \longrightarrow \bar{B}$
  - $\bar{B} \longrightarrow \bar{C}$
  - $\Longrightarrow$
  - $\bar{A} \longrightarrow \bar{C}$

- Example
  - SSN → GPA
  - GPA → priority
  - $\Longrightarrow$
  - SSN → priority

SSN	SName	address	HScode	HSname	HScity	GPA	priority
1001	Alice	Lannion	HS01	Le Dantec	Lannion	18	1
1002	Bob	Paris	HS02	Montaigne	Paris	15	3
1003	Claire	Venissieu	HS03	Hugo	Lyon	17	2
1004	David	Trégastel	HS01	Le Dantec	Lannion	16	2
1005	Emma	Marseilles	HS04	Descartes	Marseilles	15	3

Closure of attributes
Definition and notation

- Definition
  - Given a relation, FDs and a set of attributes $\bar{A}$
  - Set of all attributes B such that $\bar{A} \longrightarrow B$

- Notation
  - $\bar{A}^{+}$

Closure of attributes
Example 1

- Example
  - A ⟶ C, D
  - C ⟶ E

```
Start with {A1, ..., An}
Repeat until no change
      if A ⟶ B and A in set then add B in set
```

- {A1, ..., AN, C, D}
- {A1, ..., AN, C, D, E}

Closure of attributes
Example 2

- FDs
  - SSN → *s*Name, address, GPA
  - GPA → priority
  - HScode → HSname, HScity

- Closure of {SSN, HScode}, $\{SSN, HScode\}^+$
  - {SSN, HScode, sName, address, GPA} with SSN → *s*Name, address, GPA
  - {SSN, HScode, sName, address, GPA, priority} with GPA → priority
  - {SSN, HScode, sName, address, GPA, priority, Hsname, HScity} with HScode → HSname, HScity
  - Thus $\{SSN, HScode\}^+$ = all attributes
  - Thus {SSN, HScode} is a key

Closure and keys

- Is $\bar{A}$ a key for R?
  - Compute $\bar{A}^+$
  - If $\bar{A}^+$= all attributes then $\bar{A}$ is a key

- How can we find all keys given a set of FDs?
  - Consider every subsets of attributes and check if they determine all attributes
  - Start with single attribute

Specifying FDs for a relation (1)

- Definition
  - S_1 and S_2 sets of FDs
  - S_2 follows from S_1 if every relation instance satisfying S_1 also satisfies S_2

- Example
  - S_2: {SSN → priority}
  - S_1: {SSN → GPA, GPA → priority}
  - S_2 follows from S_1

Specifying FDs for a relation (2)

- Questions
  - How to test (that S_2 follows from S_1)?
  - Does $\bar{A} \rightarrow \overline{\mathrm{B}}$ follow from S?

- Method 1
  - Compute $\bar{A}^+$ based on S check if B in set

- Method 2
  - Armstrong's axioms

Specifying FDs for a relation (3)

- Questions
  - How to test (that S_2 follows from S_1)?
  - Does $\bar{A} \rightarrow \overline{B}$ follow from S?

- Method 1
  - Compute $\bar{A}^+$ based on S check if B in set

- Example
  - S_1: {SSN → GPA, GPA → priority}
  - S_2: {SSN → priority}
  - SSN^+ based on S_1 = {SSN, GPA, priority}
  - Priority is in SSN^+ based on S_1
  - S_2 follows from S_1

Specifying FDs for a relation (4)

- Questions
  - How to test (that S_2 follows from S_1)?
  - Does $\bar{A} \rightarrow \overline{\mathrm{B}}$ follow from S?

- Method 2
  - Armstrong's axioms

- Example
  - S_1: {SSN → GPA, GPA → priority}
  - S_2: {SSN → priority}
  - Transitivity rule: SSN → GPA and GPA → priority thus SSN → priority
  - S_2 follows from S_1

Specifying FDs for a relation (5)

- Objective
  - Minimal set of completely nontrivial FDs such that all FDs that hold on the relation follow from the dependencies in this set

BCNF decomposition algorithm (1)

- Input
  - Relation R
  - FSs for R

- Output
  - Decomposition of R intro BCNF relations

- Algorithm

```
Compute keys for R
Repeat until all relations are in BCNF:
      Pick any R' with A -> B that violates BCNF
      Decompose R' into R1(A,B) and R2(A, rest)
      Compute FDs for R1 and R2
      Compute keys for R1 and R2
```

BCNF decomposition algorithm (2)

- Example

`Student(SSN, HScode, SName, address, HSname, HScity, GPA, priority)`

SSN → SName, address, GPA, priority

HScode → HSname, HScity

GPA → priority

1NF?

Yes

2NF?

No

SSN → SName, address, GPA, priority

and

HScode → HSname, HScity

BCNF decomposition algorithm (3)

- Example

Step 1

`Student(SSN, HScode, SName, address, HSname, HScity, GPA, priority)`

with SSN → SName, address, GPA, priority

`StudentInfo(SSN, Sname, address, GPA, priority)`

GPA → priority

2NF? No

`StudentRest(SSN, HScode, HSname, HScity)`

HScode → HSname, Hscity

2NF? No

BCNF decomposition algorithm (4)

- Example

Step 2

`StudentRest(SSN, HScode, HSname, HScity)`

With HScode → HSname, HScity

`HighSchool(HScode, HSname, HScity)`

HScode → HSname, Hscity

2NF? Yes 3NF? Yes BCNF? Yes

`HSRest(SSN, HScode)`

2NF? Yes 3NF? Yes BCNF? Yes

BCNF decomposition algorithm (5)

- Example

Step 3

`StudentInfo(SSN, Sname, address, GPA, priority)`

with GPA → priority

`GPAandPriority(GPA, priority)`

2NF? Yes3NF? Yes BCNF? Yes

`StudentInfo2(SSN, Sname, address, GPA)`

2NF? Yes 3NF? Yes BCNF? Yes

BCNF decomposition algorithm (6)

- Example

Synthesis

```
HighSchool(HScode, HSname, HScity)
Application(SSN, HScode)
GPAandPriority(GPA, priority)
Student(SSN, Sname, address, GPA)
```

HScode	HSname	HScity
HS01	Le Dantec	Lannion
HS02	Montaigne	Paris
HS03	Hugo	Lyon
HS04	Descartes	Marseilles

SSN	HScode
1001	HS01
1002	HS02
1003	HS03
1004	HS01
1005	HS04

GPA	priority
18	1
15	3
17	2
16	2
14	3

SSN	SName	address	GPA
1001	Alice	Lannion	18
1002	Bob	Paris	15
1003	Claire	Venissieu	17
1004	David	Trégastel	16
1005	Emma	Marseilles	14

Bibliography

- Database Systems: The Complete Book: International Edition : Garcia-Molina, Hector, Ullman, Jeffrey D., Widom, Jennifer

Questions?