CSCI235 – Database Systems

Functional Dependencies

sjapit@uow.edu.au

27 March 2023

Acknowledgements

The following presentation were adapted from the lecture slides of:
CSCI235 – Database Systems, 02 Functional Dependencies
By Dr Janusz R. Getta, University of Wollongong, Australia
Spring, 2018

Outline

- Functional dependency? What is it?
- Functional dependencies versus classes of objects
- Functional dependencies versus associations
- Derivations of functional dependencies
- Armstrong axioms
- Other inference rules
- Using inference rules

Functional dependency? What is it?

Let $\boldsymbol{R} = (\boldsymbol{A_1}, \ldots, \boldsymbol{A_n})$ be a relational schema (a header of relational table) and let $\boldsymbol{X}, \boldsymbol{Y}$ be the nonempty subsets of $\boldsymbol{R}$

We say that a functional dependency $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ is valid in a relational schema $\boldsymbol{R}$ if for any contents of a relational table $\boldsymbol{R}$, it is not possible that $\boldsymbol{R}$ has two rows that agree in the components for all attributes in a set $\boldsymbol{X}$ yet disagree on one or more component for the attributes in a set $\boldsymbol{Y}$

Functional dependency? What is it?

Examples

- A warehouse is located at exactly one address:

  $$\boldsymbol{warehouse \rightarrow address}$$

- An address is related to exactly one warehouse:

  $$\boldsymbol{address \rightarrow warehouse}$$

- At a warehouse, the parts of the same sort have only one total quantity:

  $$\boldsymbol{warehouse, part \rightarrow quantity}$$

- A car has one owner:

  $$\boldsymbol{registration \rightarrow drivingLicense}$$

Functional dependency? What is it?

More examples

- A student has one first name and one last name and one date of birth:
  $$studentNumber \rightarrow firstName, lastName, dateOfBirth$$
- An employee belongs to one department:
  $$employeeNumber \rightarrow departmentName$$
- A department has one manager:
  $$departmentName \rightarrow managerNumber$$
- An employee has one manager:
  $$employeeNumber \rightarrow managerNumber$$

Functional dependency? What is it?

More examples

- A student enrols a subject one time:
  $\boldsymbol{studentNumber, subjectCode \rightarrow enrolmentDate}$
- An employee is located in one building in one office:
  $\boldsymbol{employeeNumber \rightarrow buildingNumber, officeNumber}$
- An office in a building hosts one employee:
  $\boldsymbol{buildingNumber, officeNumber \rightarrow employeeNumber}$
- An office in a building at a campus hosts one employee:
  $\boldsymbol{campusName, buildingNumber, officeNumber \rightarrow employeeNumber}$

Functional dependency? What is it?

More examples

- A department has one manager:
  $\boldsymbol{departmentName \rightarrow managerNumber}$
- A department is located in one building:
  $\boldsymbol{departmentName \rightarrow buildingNumber}$
- A department has one manager and it is located in one building:
  $\boldsymbol{departmentName \rightarrow managerNumber, buildingNumber}$

Functional dependency? What is it?

How to discover the **functional dependencies** in a relational table?

- Is it possible to discover the **functional dependencies** in a **relational schema** (a header of relational table) $\boldsymbol{R(A, B, C, D, E)}$?
- Of course it is impossible to do it because we do not know the **semantics (the meanings)** of the names: $\boldsymbol{R, A, B, C, D, E}$
- To discover the **functional dependencies** in a relational table we MUST use the **semantics** of a **relational table name** and the **names of attributes**

Functional dependency? What is it?

- For example consider a relational schema (a header of relational table) $TRIP(rego\#, licence\#, tdate)$ of a relational table that contains information about the **trips** made by the **drivers** (**licence#**) who used the **trucks** (**rego#**) on a given **day** (**tdate**)
- Can a truck be used only one time ?

  If yes then $rego\# \rightarrow tdate$
- Can a driver make only one trip ?

  If yes the $licence\# \rightarrow tdate$

Functional dependency? What is it?

- Can a driver use more than one truck ?

  If yes then $\boldsymbol{licence\#} \nrightarrow \boldsymbol{rego\#}$
- Can a truck be used by more than one driver ?

  If yes then $\boldsymbol{rego\#} \nrightarrow \boldsymbol{licence\#}$
- And so on …

Functional dependencies versus classes of objects

A class of object STUDENT.

STUDENT	
stdNum	ID1
stdFName	ID2
stdLName	ID2
stdDOB	ID2
average	
language[1..*]	

Validates (satisfies) the following functional dependencies:

$stdNum \rightarrow stdFName$

$stdNum \rightarrow stdLName$

$stdNum \rightarrow stdDOB$

$stdNum \rightarrow average$

$stdFName, stdLName, stdDOB \rightarrow stdNum$

$stdFName, stdLName, stdDOB \rightarrow average$

Functional dependencies versus classes of objects

$stdNum \rightarrow stdFName$
$stdNum \rightarrow stdLName$
$stdNum \rightarrow stdDOB$
$stdNum \rightarrow average$

The four functional dependencies shown above are equivalent to:

$stdNum \rightarrow stdFName, stdLName, stdDOB$

Functional dependencies versus classes of objects

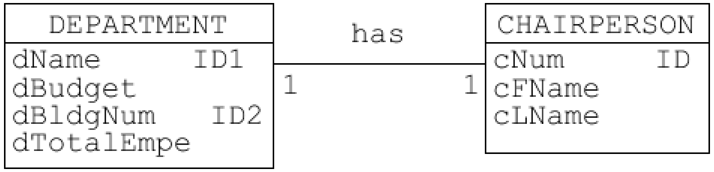
$stdFName, stdLName, stdDOB \longrightarrow stdNum$
$stdFName, stdLName, stdDOB \longrightarrow average$

The two functional dependencies shown above are equivalent to:

$stdFName, stdLName, stdDOB \longrightarrow stdNum, average$

Functional dependencies versus associations

The classes of objects DEPARTMENT and CHAIRPERSON and association has

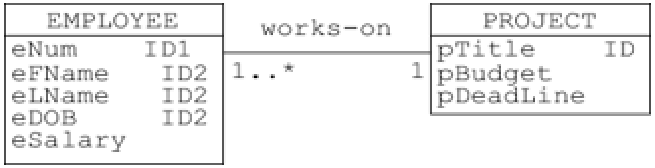


validate (satisfy) the following functional dependencies:

$$dName \rightarrow dBudget, dBldgNum, dTotalEmpe$$
$$dBldgNum \rightarrow dName, dBudget, dTotalEmpe$$
$$cNum \rightarrow cFName, cLName$$
$$dName \rightarrow cNum, cFName, cLName$$
$$dBldhNum \rightarrow cNum, cFLname, cLName$$
$$cNum \rightarrow dName, dBudget, dBldgNum, dTotalEmpe$$

Functional dependencies versus associations

The classes of objects EMPLOYEE and PROJECT and association works-on



validate (satisfy) the following functional dependencies:

$eNum \rightarrow eFName, eLName, eDOB, eSalary$

$eFName, eLName, eDOB \rightarrow eNum, eSalary$

$pTitle \rightarrow pBudget, pDeadLine$

$eNum \rightarrow pTitle, pBudget, pDeadLine$

$eFName, eLName, eDOB \rightarrow pTitle, pBudget, pDeadLine$

Functional dependencies versus associations

The classes of objects STUDENT and COURSE and association enroll

validate (satisfy) the following functional dependencies:

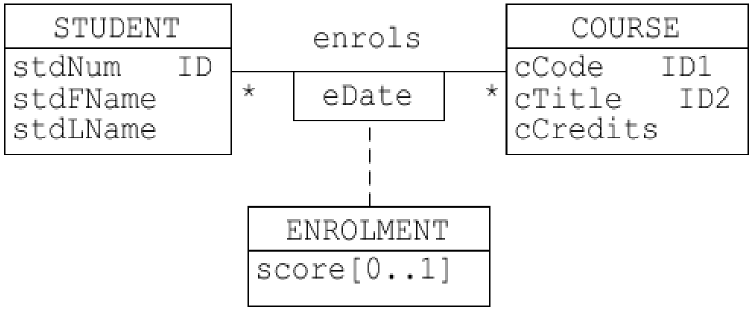
$stdNum \rightarrow stdFName, stdLName$

$cCode \rightarrow cTitle,\ cCredits$

$cTitle \rightarrow cCode, cCredits$

$stdNum,\ cCode,\ eDate \rightarrow score$

$stdNum,\ cTitle,\ eDate \rightarrow score$



Derivations of functional dependencies

Consider a relational schema (a header of relational table)

EMPLOYEE(e#, ename, department, address, chairperson)

If $\boldsymbol{e\# \rightarrow ename}$ and $\boldsymbol{e\# \rightarrow department}$

Then $\boldsymbol{e\# \rightarrow ename,\ department}$

If $\boldsymbol{e\# \rightarrow department}$ and $\boldsymbol{department \rightarrow address}$

Then $\boldsymbol{e\# \rightarrow address}$

If $\boldsymbol{e\# \rightarrow department}$ and $\boldsymbol{department \rightarrow chairperson}$

Then $\boldsymbol{e\# \rightarrow chairperson}$

Derivations of functional dependencies

If $\boldsymbol{e\# \rightarrow department}$

then $\boldsymbol{e\#, ename \rightarrow department}$

If $\boldsymbol{e\#, ename \rightarrow department}$ then $\boldsymbol{e\#, ename, address \rightarrow department}$

Derivations of functional dependencies

It is always true that $e\# \rightarrow e\#$

Functional dependency $e\# \rightarrow e\#$ is called as a **trivial functional dependency**

It is always true that $e\#, ename \rightarrow e\#$

A functional dependency $e\#, ename \rightarrow e\#$ is also called as a **trivial functional dependency**

A **trivial functional dependency** is a functional dependency that is always true no matter what its left and right hand sides are.

Derivations of functional dependencies

Consider a relational schema $R(A, B, C)$

It is always true that $A \rightarrow A$

It is always true that $A, B \rightarrow A$

It is always true that $A, B, C \rightarrow A$

If $A \rightarrow B$ then $A, C \rightarrow B$

If $A \rightarrow B, C$ then $A \rightarrow B$ and $A \rightarrow C$

If $A \rightarrow B$ and $B \rightarrow C$ then $A \rightarrow C$

Armstrong axioms

Let $\boldsymbol{R} = (A_1, \dots, A_n)$ be a relational schema (a header of relational table) and

let $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ be the nonempty subsets of $\{\boldsymbol{A_1}, \dots, \boldsymbol{An}\}$

1. If $\boldsymbol{Y} \subseteq \boldsymbol{X}$ then $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ (**reflexivity axiom**)
2. If $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ then $\boldsymbol{X}, \boldsymbol{Z} \rightarrow \boldsymbol{Y}, \boldsymbol{Z}$ (**augmentation axiom**)
3. If $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ and $\boldsymbol{Y} \rightarrow \boldsymbol{Z}$ then $\boldsymbol{X} \rightarrow \boldsymbol{Z}$ (**transitivity axiom**)

The axioms 1, 2, and 3 form a **minimal and complete set of axioms**

Other inference rules

Let $\boldsymbol{R} = (\boldsymbol{A_1}, \dots, \boldsymbol{A_n})$ be a relational schema (a header of relational table) and let $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ be the nonempty subsets of $\{A_1, \dots, A_n\}$

1. If $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ and $\boldsymbol{X} \rightarrow \boldsymbol{Z}$ then $\boldsymbol{X} \rightarrow \boldsymbol{Y}, \boldsymbol{Z}$ (**union rule**)
2. If $\boldsymbol{A} \rightarrow \boldsymbol{B}$ and $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ then $\boldsymbol{AX} \rightarrow \boldsymbol{BY}$ **(composition rule)**
3. If $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ and $\boldsymbol{Z} \subseteq \boldsymbol{Y}$ then $\boldsymbol{X} \rightarrow \boldsymbol{Z}$ (**decomposition rule** or **reduce right hand side rule**)
4. If $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ and $\boldsymbol{W}, \boldsymbol{Y} \rightarrow \boldsymbol{Z}$ then $\boldsymbol{W}, \boldsymbol{X} \rightarrow \boldsymbol{Z}$ (**pseudo transitivity rule**)
5. If $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ then $\boldsymbol{X}, \boldsymbol{Z} \rightarrow \boldsymbol{Y}$ **(extend left hand side rule)**

Using inference rules

Let $\boldsymbol{R} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ be a relational schema

Given set of functional dependencies

$\boldsymbol{F} = \{\boldsymbol{A} \rightarrow \boldsymbol{B}, \boldsymbol{B} \rightarrow \boldsymbol{C}\}$ valid in $\boldsymbol{R}$

Is it true that $\boldsymbol{A} \rightarrow \boldsymbol{C}$?

If $\boldsymbol{A} \rightarrow \boldsymbol{B}$ and $\boldsymbol{B} \rightarrow \boldsymbol{C}$ then application of **transitivity axiom** provides $\boldsymbol{A} \rightarrow \boldsymbol{C}$

Using inference rules

Let $\boldsymbol{R} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ be a relational schema

Given set of functional dependencies

$\boldsymbol{F} = \{\boldsymbol{A} \rightarrow \boldsymbol{B}, \boldsymbol{C}\}$ valid in $\boldsymbol{R}$

Is it true that $\boldsymbol{A} \rightarrow \boldsymbol{B}$ and $\boldsymbol{A} \rightarrow \boldsymbol{C}$?

Reflexivity axiom provides $\boldsymbol{B}, \boldsymbol{C} \rightarrow \boldsymbol{C}$

If $\boldsymbol{A} \rightarrow \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{B}, \boldsymbol{C} \rightarrow \boldsymbol{C}$ then **transitivity axiom** provides $\boldsymbol{A} \rightarrow \boldsymbol{C}$

Reflexivity axiom provides $\boldsymbol{B}, \boldsymbol{C} \rightarrow \boldsymbol{B}$

If $\boldsymbol{A} \rightarrow \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{B}, \boldsymbol{C} \rightarrow \boldsymbol{B}$ then **transitivity axiom** provides $\boldsymbol{A} \rightarrow \boldsymbol{B}$

Using inference rules

Let $\boldsymbol{R} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ be a relational schema

Given set of functional dependencies

$\boldsymbol{F} = \{\boldsymbol{A} \rightarrow \boldsymbol{B}, \boldsymbol{A} \rightarrow \boldsymbol{C}\}$ valid in $\boldsymbol{R}$

Is it true that $\boldsymbol{A} \rightarrow \boldsymbol{B}, \boldsymbol{C}$?

If $\boldsymbol{A} \rightarrow \boldsymbol{B}$ then **augmentation axiom** provides

$\boldsymbol{A} \rightarrow \boldsymbol{A}, \boldsymbol{B}$

If $\boldsymbol{A} \rightarrow \boldsymbol{C}$ then **augmentation axiom** provides

$\boldsymbol{A}, \boldsymbol{B} \rightarrow \boldsymbol{B}, \boldsymbol{C}$

If $\boldsymbol{A} \rightarrow \boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{A}, \boldsymbol{B} \rightarrow \boldsymbol{B}, \boldsymbol{C}$ then **transitivity axiom** provides

$\boldsymbol{A} \rightarrow \boldsymbol{B}, \boldsymbol{C}$

Using inference rules

Let $\boldsymbol{R} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ be a relational schema
Given set of functional dependencies $\boldsymbol{F} = \{\boldsymbol{A} \rightarrow \boldsymbol{B}\}$ valid in $\boldsymbol{R}$

Is it true that $\boldsymbol{A}, \boldsymbol{C} \rightarrow \boldsymbol{B}$?

Reflexivity axiom provides $\boldsymbol{A}, \boldsymbol{C} \rightarrow \boldsymbol{A}$
If $\boldsymbol{A}, \boldsymbol{C} \rightarrow \boldsymbol{A}$ and $\boldsymbol{A} \rightarrow \boldsymbol{B}$ then **transitivity axiom** provides $\boldsymbol{A}, \boldsymbol{C} \rightarrow \boldsymbol{B}$

Using inference rules

A relational schema
$\boldsymbol{STUDENT(s\#, fname, lname, dob, average)}$ validates (satisfies) the following functional dependencies:

$\boldsymbol{s\# \rightarrow fname}$

$\boldsymbol{s\# \rightarrow lname}$

$\boldsymbol{s\# \rightarrow dob}$

$\boldsymbol{s\# \rightarrow average}$

$\boldsymbol{fname, lname, dob \rightarrow s\#}$

$\boldsymbol{fname, lname, dob \rightarrow average}$

Using inference rules

Hence,

$s\# \rightarrow fname, lname, dob, average$ and ...
$fname, lname, dob \rightarrow s\#, average$

Note, that both functional dependencies **cover** entire relational schema and **no other** functional dependencies that **do not cover** entire relational schema validate in the schema e.g.
$fname \rightarrow s\#$ or $lname \rightarrow s\#$ or $dob \rightarrow s\#$ etc.

The relational schema
$STUDENT(s\#, fname, lname, dob, average)$ is now normalised.

References

T. Connoly, C. Begg, Database Systems, A Practical Approach to Design, Implementation, and Management, Chapter 14.4 Functional Dependencies, Chapter 15.1 More on Functional Dependencies, Pearson Education Ltd, 2015