Functional Dependencies – Part 2

Codd and Functional Dependencies

- **Functional dependencies** (FDs) were introduced by Codd in 1971.[1]
- Edgar F. Codd of IBM Research (1923-2003) invented the **relational data model** for data management in 1970.
- H
  th

  - **Normalization**
    - Boyce–Codd Normal Form (BCNF)

  - **Query languages**
    - Relational Calculus
    - Relational Algebra

[1] *Further Normalization of the Data Base Relational Model*. E. F. Codd, IBM Research Report, San Jose, California, 1971.

Why Functional Dependencies?

- We need some **formal way** of analysing whether a database schema is w
- F tiona
  d

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

FDs tell us "relationship between and among attributes"!

Functional Dependencies – Informal Description

- We have two FDs on ENROLMENT:

ENROLMENT					
	_____		_____	_____	Unit
					6
					12
					6
Michael	123458	21/04/1985	COMP8740	2011 S2	12
Fran	123457	11/09/1987			

- StudentID **functionally determines**

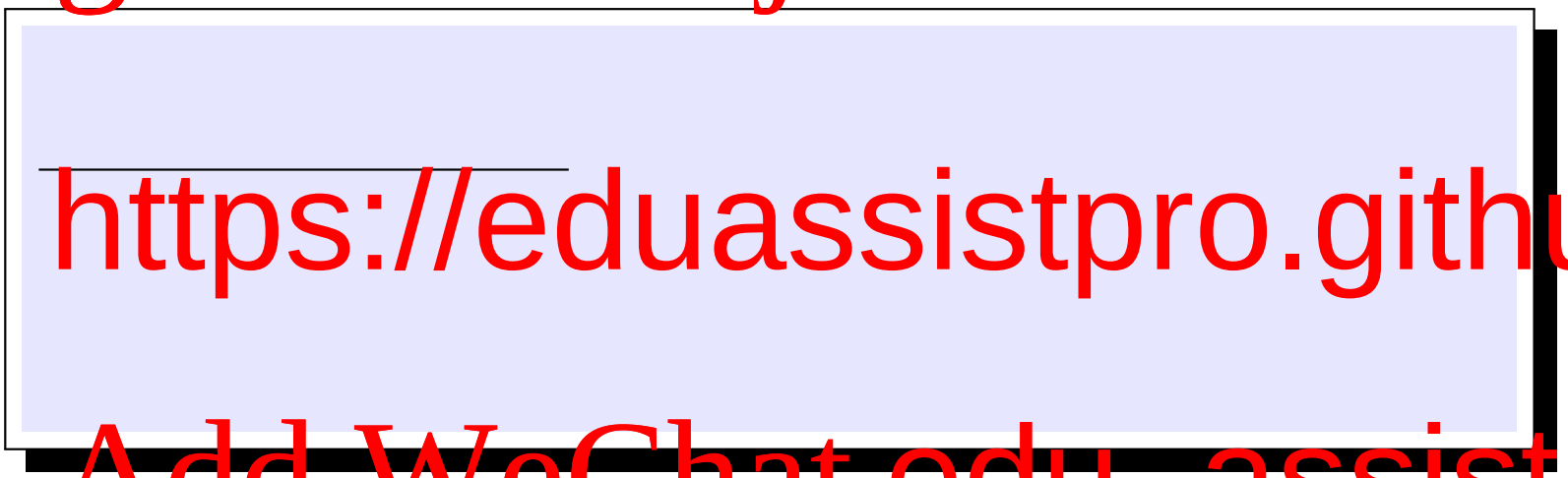
  $\{\text{StudentID}\} \rightarrow \{\text{Name, DoB}\}$

- CourseNo **functionally determines** Unit, i.e.,

  $\{\text{CourseNo}\} \rightarrow \{\text{Unit}\}$

Functional Dependencies – Informal Description

- A FD says that, within a relation, the values of some attributes determine the values of other attributes.



- If attributes A, B, C determine attributes

$$\{A, B, C\} \rightarrow \{D, E\}$$

  - This means, if two tuples have the same values for A, B and C, then they must also have the same values for D and E.
  - A, B and C are the **determinant**, while D and E are the **dependent**.

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,

$$\Downarrow$$

$$t_1[Y] = t_2$$

- A FD is **trivial** if it can *always* be satisfied.
  - $\{A, B, C\} \rightarrow \{C\}$
  - $\{A, B, C\} \rightarrow \{A, B\}$
- **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 on Relations

- Consider the following relations with attributes {A,B,C,D,E}. Do they satisfy: (1) $AB \rightarrow E$; (2) $C \rightarrow DE$;

A	B	C	D	E
1	4	1	9	4
1	4	2	8	9
1	4	3	8	9

A	B	C	D	E
1	3	1	3	8

- **Check:**

	$r_1(R)$	$r_2(R)$
(1) $AB \rightarrow E$	no	yes
(2) $C \rightarrow DE$	yes	no

How to Identify FDs in General?

- A functional dependency specifies a constraint on the relation schema that must hold **at all times**.
- 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) Identifying FDs - Analyse Data Requirements

- Consider the following relation schema:

  RENTAL={CustID,CustName,PropertyNo,DateStart,Owner} .

- **D**

  1. $\{\quad\} \rightarrow \{\quad\}$
  2. A customer cannot rent two or more properti

     {CustID, DateStart
  3. A customer cannot rent the same property m

     $\{\text{PropertyNo, CustID}\} \rightarrow \{\text{DateStart}\}$
  4. Each property can be uniquely identified by its owner.

     $\{\text{Owner}\} \rightarrow \{\text{PropertyNo}\}$

(2) Identifying FDs - 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} $\rightarrow$ {Name, DoB};
  - {CourseNo} $\rightarrow$ {Unit};
  - {StudentID, CourseNo, Semester} $\rightarrow$ {Name, DoB, Unit};
  - {Name} $\rightarrow$ {StudentID} $\times$;
  - {DoB} $\rightarrow$ {StudentID} $\times$;
  - .....

Limitations: Sample data needs to be a true representation of **all possible values** that the database may hold.