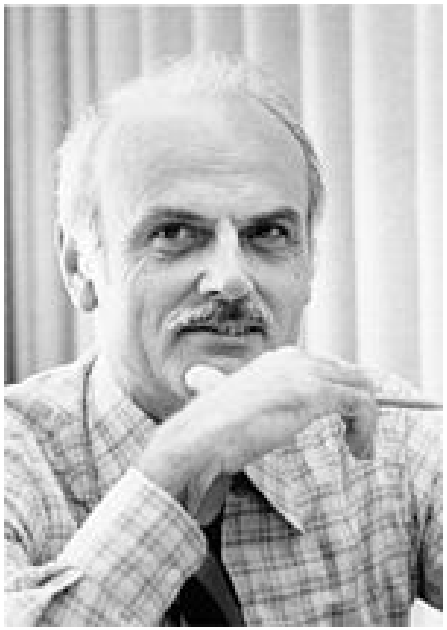
Functional Dependencies – Part 2

Definition and Identification

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.
- He received the ACM Turing Award in 1981 for his contributions on the theoretical foundations of relational databases:
  - **Functional dependencies**
  - **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 well-designed, or why one is better than another.
- FDs are developed to define the **goodness** and **badness** of (relational) database design in a formal way.
  - **Top down**: start with a relation schema and FDs, and produce smaller relation schemas in certain normal form (called *normalisation*).
  - **Bottom up**: start with attributes and FDs, and produce relation schemas (*not popular in practice*).

FDs tell us "relationship between and among attributes"!

Functional Dependencies – Informal Description

- We have two FDs on ENROLMENT:

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
Michael	123458	21/04/1985	COMP2400	2009 S2	6
Michael	123458	21/04/1985	COMP8740	2011 S2	12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

  - StudentID **functionally determines** Name and DoB, i.e.,
    $\{\text{StudentID}\} \rightarrow \{\text{Name, DoB}\}$
  - 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.

Animal	$\rightarrow$	Legs
Ostrich		2
Wombat		4

- If attributes A, B, C determine attributes D, E, then we write

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

  - 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 $X \to Y$ with attribute sets $X, Y \subseteq R$.
  - A relation $r(R)$ **satisfies** $X \to Y$ **on** R if, for any two tuples $t_1, t_2 \in r(R)$, whenever the tuples t_1 and t_2 coincide on values of X, they also coincide on values of Y.

$$t_1[X] = t_2[X] \Downarrow t_1[Y] = t_2[Y]$$

- A FD is **trivial** if it can *always* be satisfied, e.g.,
  - $\{A, B, C\} \to \{C\}$
  - $\{A, B, C\} \to \{A, B\}$
- **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) $C \rightarrow DE$;

$r_1(R)$

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

$r_2(R)$

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

- **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 real-life applications, we often use the following approaches:

  (1) **Analyse data requirements**
  Can be provided in the form of discussion with application users 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:

$$\text{RENTAL}=\{\text{CustID},\text{CustName},\text{PropertyNo},\text{DateStart},\text{Owner}\}\ .$$

- **Data requirements:**
  1. Each customer can be uniquely identified by his or her customer ID.
  $$\{\text{CustID}\} \rightarrow \{\text{CustName}\}$$
  2. A customer cannot rent two or more properties from the same date.
  $$\{\text{CustID, DateStart}\} \rightarrow \{\text{PropertyNo}\}$$
  3. A customer cannot rent the same property more than once.
  $$\{\text{PropertyNo, CustID}\} \rightarrow \{\text{DateStart}\}$$
  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
Tom	123456	25/01/1988	COMP2400	2010 S2	6
Tom	123456	25/01/1988	COMP8740	2011 S2	12
Michael	123458	21/04/1985	COMP2400	2009 S2	6
Michael	123458	21/04/1985	COMP8740	2011 S2	12
Fran	123457	11/09/1987	COMP2400	2009 S2	6

- We may have:
  - $\{\text{StudentID}\} \rightarrow \{\text{Name, DoB}\}$;
  - $\{\text{CourseNo}\} \rightarrow \{\text{Unit}\}$;
  - $\{\text{StudentID, CourseNo, Semester}\} \rightarrow \{\text{Name, DoB, Unit}\}$;
  - $\{\text{Name}\} \rightarrow \{\text{StudentID}\}$ $\times$;
  - $\{\text{DoB}\} \rightarrow \{\text{StudentID}\}$ $\times$;
  - .....

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