Data Bases

Normalization

Sergio Peignier sergio.peignier@insa-lyon.fr

Associate Professor
INSA Lyon
Biosciences department

Overview

Motivation

Functional Dependencies

Closure

Normal Forms

Motivation

Model choice

- **Same application** $\rightarrow$ **many possible models**.
- **Unadapted model** $\Rightarrow$ **anomalies** :
  - While **Using** the database (e.g., redundancy ...).
  - While **Updating** the databse (e.g., insert, delete, modify).
- How can we **choose** a **suitable model**?

Example

Consider the following relation schema:

FOOD_PRODUCT(Name, Address, Product_name, price)

Which problems do you identify?

Example

- **Redundancy**:
  Address and name **repeated** for each product.

- If a restaurant moves to a **different address**:
  Modify each tuple associated to the restaurant.

- If a restaurant creates a **new product**:
  Check the consistency of the new tuple w.r.t. others.

- If a restaurant **disappears**:
  Search and **delete** each tuple associated to it.

Objective of normalization

Build a **coherent database model**, that avoids **anomalies** by **satisfying** some **properties** called **”normal forms”**

Functional Dependencies

Definition | Functional Dependency (FD)

Notations:

- A and B: two **sets of attributes**.
- x and y: two **tuples**.
- x_A: **values** of **attributes** $\in A$ for **tuple** x.
- $AB = A \cup B$: **Union** of A and B.

Definition:

- If for all x, y; $x_A = y_A \Rightarrow x_B = y_B$ then:
  - B **functionally dependent** on A
  - A **determines** B
  - **Notation**: $A \rightarrow B$
- **FD**: **constrains** describing **links** between a **attributes**.

Normal forms are **based** on **FDs**.

Exercise

Write the **functional dependencies** for the relation:

$$\boxed{Student(ID, Nom, Prenom, Age)}$$

What do you notice?

Functional Dependencies

- **Key** $\rightarrow$ **determines** all other **attributes**.
- **FD**: **Intrinsic** characteristics of attributes.
  (**Never define FD** by checking the relation **instances**)

Example:

Kebab (Name, Address, Product, Price):

Name	Address	Product	Price
Ambiance	Bvd 11 Nov. 1918	galette kebab	5
Univers cafe	Bvd 11 Nov. 1918	maxi kebab	7.5

Does price $\rightarrow$ name, address, product?

Armstrong Axioms

$\forall A, B, C, D$ sets of **attributes** of R; from a set of **FDs**

We can **deduce** other **FDs** using **Armstrong axioms**:

- **Reflexivity**: if $B \subseteq A$, then $A \rightarrow B$.
- **Augmentation**: if $A \rightarrow B$ then $AC \rightarrow BC$.
- **Transitivity**: if $A \rightarrow B$, and $B \rightarrow C$, then $A \rightarrow C$.

Secondary rules:

- **Union**: If $A \rightarrow B$ and $A \rightarrow C$, then $A \rightarrow BC$
- **Pseudo-transitivity**: If $A \rightarrow B$ and $DB \rightarrow C$, then $DA \rightarrow C$
- **Decomposition**: If $A \rightarrow B$ and $C \subseteq B$, then $A \rightarrow C$

Exercise: Deduce secondary rules from axioms.

Armstrong Axioms | Union

$\boxed{A \rightarrow B \quad \text{and} \quad A \rightarrow C}$

- $A \rightarrow B$ by augmentation : $AC \rightarrow BC$
- $A \rightarrow C$ by augmentation : $AA \rightarrow AC$
- And $AA = A$, Then: $A \rightarrow AC$
- By transitivity: $\boxed{A \rightarrow BC}$

Armstrong Axioms | Pseudo-transitivity

$$\boxed{A \rightarrow B \quad \text{and} \quad DB \rightarrow C}$$

- $A \rightarrow B$ by augmentation : $DA \rightarrow DB$
- Since $DB \rightarrow C$ by transitivity: $\boxed{DA \rightarrow C}$

Armstrong Axioms | Decomposition

$$\boxed{A \rightarrow B \quad \text{and} \quad C \subseteq B}$$

- $C \subseteq B$ by reflexivity: $B \rightarrow C$
- Since $A \rightarrow B$, par transitivity: $\boxed{A \rightarrow C}$

Exercise

Consider the following family of FDs:

$$F = \{AB \to C; B \to D; CD \to E; CE \to GH; G \to A\}.$$

Show that $AB \to E$,

Kinds of Functional Dependencies

- **Trivial FD** $\rightarrow$ Obtained by **reflexivity**:
  $A \rightarrow B$ is **trivial** if $B \subseteq A$
- **Simple FD** $\mapsto$ **Single right element**:
  $A \rightarrow B$ is **simple** if $|B| = 1$.
- **Direct FD** $\mapsto$ Is **not** obtained by **transitivity**:
  $A \rightarrow B$ is **direct** if $\nexists C$ s.t. $A \rightarrow C$ et $C \rightarrow B$
- **Full FD** (resp. partial FD) $\mapsto$ **Irreducible left part**:
  $A \rightarrow B$ is **full** if $\nexists C \subset A$ s.t. $C \rightarrow B$.
- **Irreducible or minimal FD** $\rightarrow$ **Simple** and **Full**:
  $A \rightarrow B$ is **EFD** if $\nexists C \subset A$ s.t. $C \rightarrow B$ and if $|B| = 1$.

Closure

Closure

- R: **Relation schema**.
- X: **Subset of attributes** of R.
- F: **family** of **FDs**
- X^+ **closure of** X :
  Set of attributes of R **deduced** from X and F,
  using **Armstrong Axioms**.
- Y is **included** in X^+ if $X \rightarrow Y$.

Closure Computation

Compute the **closure** of a **set of attributes**:

- **Initialization** $X^+ = X$
- **Repeat** until X^+ does not change anymore:
  - **Search:** **FD** $A \rightarrow B$ s.t., $A \subseteq X^+$
  - **Update**: $X^+ \leftarrow X^+ \cup B$.

Exercise

Consider the following set of FD:

$$F = \{A \rightarrow D; AB \rightarrow E; BI \rightarrow E; CD \rightarrow I; E \rightarrow C\}$$

- Compute the closure of AE w.r.t. F.
- Compute the closure of BE w.r.t. F.

Transitive Closure (F^+)

- F: set of **irreducible FDs**.
- F^+ **transitive closure** of F:
  - **Max. non-trivial set** of **irreducible FD** from F.
  - $F \cup \{$ **Irreducible FD** obtained by **transitivity** or **pseudo-transitivity** $\}$
- **2 FD sets** are **equivalent** if they have the **same** F^+

Minimal Cover

- F: set of **irreducible FDs**.

- $MIN(F)$: **Minimal cover** of F
  - **Minimal set** $MIN(F)$ of **FDs** F.
  - Obtained by **removing redundant FDs**.
    i.e., **FDs deductible** from $MIN(F)$.
  - $(MIN(F))^+ = F^+$ and $\nexists F' \subset MIN(F)$ s.t. $F'^+ = F^+$

- **Theorem**:
  Any set of **irreducible FD** has a **minimal cover** (not unique most of the time).

Minimal Cover search Algorithm

- Write all **FDs** as $X \rightarrow A$, with:
  - X **set of attributes**.
  - A **elementary attribute**.

  e.g., $X \rightarrow A_1 A_2 A_n$ is **replaced by** n **FDs** $X \rightarrow A_i$.

- **Remove redundant attributes** e.g., if $ABCD \rightarrow E$ and $A \rightarrow B$ then $ACD \rightarrow E$

- **Remove redundant DFs** (that can be **produced** by **transitivity**).

Example

$$F = \{A \to B, ABCD \to E, EF \to GH, ACDF \to EG\}$$

Example | Step 1

$$F = \{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow GH, ACDF \rightarrow EG\}$$

Write simple FDs:

$$F = \{A \rightarrow B, ABCD \rightarrow E, \mathsf{EF} \rightarrow \mathsf{G}, \mathsf{EF} \rightarrow \mathsf{H}, \mathsf{ACDF} \rightarrow \mathsf{E}, \mathsf{ACDF} \rightarrow \mathsf{G}\}$$

Example | Step 2

Remove redundant attributes:

- Remove each left attribute at a time.
- Check if you can deduce the right side.
- If so, the letter is redundant, you can remove it.

Application:

- $ABCD \rightarrow E$ but $A \rightarrow B$ so B is redundant;
  $ABCD \rightarrow E$ becomes $ACD \rightarrow E$
- $ACDF \rightarrow E$ but $ACD \rightarrow E$ so F is redundant
  $ACDF \rightarrow E$ becomes $ACD \rightarrow E$

Finally:

$$\{A \rightarrow B, \mathbf{ACD} \rightarrow \mathbf{E}, EF \rightarrow G, EF \rightarrow H, \mathbf{ACD} \rightarrow \mathbf{E}, ACDF \rightarrow G\}$$

Example | Step 3

Remove redundant dependencies

- Check each FD.
- Remove it if you can deduce it using the other FDs.

Application:

- $ACD \rightarrow E$ is duplicated so you can remove one FD.
- $ACDF \rightarrow G$ can be deduced from $ACD \rightarrow E$ and $EF \rightarrow G$

Finally:

$$MIN(F) = \{A \rightarrow B, EF \rightarrow G, EF \rightarrow H, \mathsf{ACD} \rightarrow \mathsf{E}\}$$

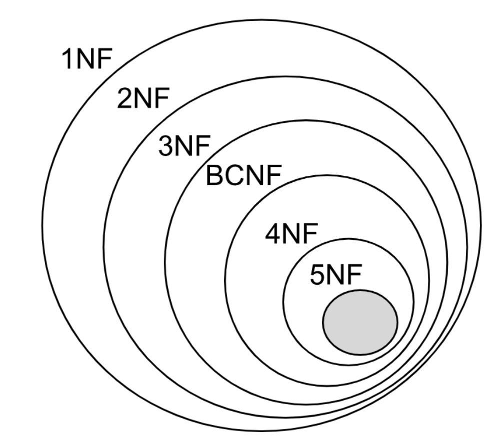
Normal Forms

Normalization

Normalization $\mapsto$ **Decompose** a schema in order to:

- **Avoid anomalies** when **updating** the relation.
- **Avoid loosing** semantic **information** and **FDs**
- **Problem**:
  Normalization $\rightarrow$ **performance degradation**
  (Queries based on join operations).

Normal Forms



In this lecture: NF1, NF2 and NF3 by E.F. Codd in 1972.

Normal Form 1

- Relation R is **NF1** if:
    - R has **attributes** with **atomic values**.
      e.g. of atomic valued date attribute: ”3/1/90”
      e.g. of multi-valued day attribute: ”1/8/98, 1/3/98”
    - R has a **primary key** $\rightarrow$ **No duplicate tuples**.

- **No NF1** $\Rightarrow$ **slower search**
  (one should analyze each attribute content)

Exercise

R

name	phone #	Address
Homer Simpson	06.01, 06.66	Street A, # 65
Ned Flanders	06.02	Street B, # 66
Montgomery Burns	06.03, 06.67	Street H, # 666

Convert *R* to **NF1**.

Anomalies

- **Insert** new person without phone # $\rightarrow$ impossible.
- **Update** address $\rightarrow$ change multiple entries.
- **Delete** phone # $\rightarrow$ delete all info about someone.

Normal Form 2

- **NF2** based on the **full functional dependence**.

- **Relation** R is **NF2** if:
  - R is **NF1**
  - **FDs** between the **key** and **non-key attributes** are **full**: K: primary key of R, $\forall$ attribute $A \notin K$, $\nexists C \subset K$ s.t. $C \rightarrow A$.

  Remark:
  If all **FDs** have an atomic left attribute, then R is **NF2**

- **No NF3** $\Rightarrow$ **Redundancies** (memory burden).

Exercise

Relation:

$Employee(idEmp, idService, name, salary, nameService)$

Key: $[idEmp, idService]$

Functional Dependencies: FD:
$idEmp \rightarrow idService, name, salary, nameService$

Question: Is this relation NF2?

Normal Form 3

- **Relation** R is **NF3** if:
    - R is **NF2**
    - **No FDs** between **non-key attributes**.
      K **primary key**, $\forall$ **FD** $X \to A$; $X = K$ or $A \subseteq K$.
    - **Boyce-Codd Normal Form**:
      K **primary key**, $\forall$ **FD** $X \to A$; $X = K$.
    - **FDs between** $A \subseteq K$ and $X \subseteq K$: allowed
    - **FDs from non-key** to $A \subseteq K$: allowed

- **Convert** R into **NF3**: **remove transitive FDs** by **dispatching attribute** into different **relations**.

- **No NF2** $\Rightarrow$ **Redundancies** (memory burden).

Exercise

Relation: $R(Name_F, Address_F, Product, Price)$

Functional Dependencies:

$FD = \{Name_F \rightarrow Address_F; Name_F, Product \rightarrow Price\}$

Key: $\{Name_F, Product\}$.

Question: Is R in **NF3**?

Join dependency

- R: Relation schema.
- $R_1, R_2, ..., R_n$: **Decomposition** of R.

Relation R satisfies the **join dependency** $*(R_1, ..., R_n)$ if:

$$\boxed{\bowtie_{i=1}^{n} \Pi_{R_i}(R) = R}$$

Decomposition

- R: **universal relation** (with all the attributes)
- F: **set of FDs** over the **set of attributes** of R
- Find its **minimal cover** $MIN(F)$
- Determine the **primary key** of R from $MIN(F)$
- For each **FD**, **decompose** R , using **Heat's Theorem**:
  If $R(A, B, C)$ is **not NF3**, and R has **FD** $B \rightarrow C$:
  Decompose R in $R(A, B)$ and $R2(B, C)$
- Apply this step **iteratively** until each relation is **NF3**.
- **Decomposition representation**: **tree** with **NF3** leaves.

Exercise

Decompose the relation to **NF3**.

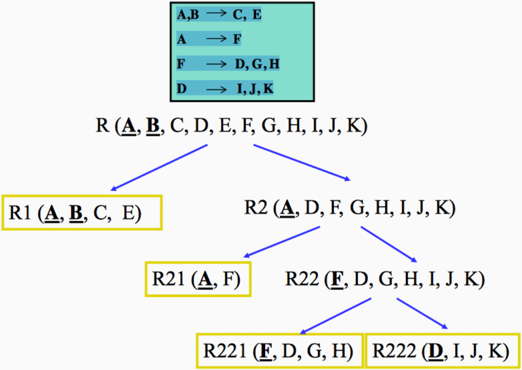
Universal Relation:

$$R(A, B, C, D, E, F, G, H, I, J, K)$$

Minimal cover:

$$F = \{AB \to CE; A \to F; F \to DGH; D \to IJK\}$$

Answer


A,B ⟶ C, E
A ⟶ F
F ⟶ D, G, H
D ⟶ I, J, K
R (A, B, C, D, E, F, G, H, I, J, K)
R1 (A, B, C, E)
R2 (A, D, F, G, H, I, J, K)
R21 (A, F)
R22 (F, D, G, H, I, J, K)
R221 (F, D, G, H)
R222 (D, I, J, K)