Functional Dependency

What is wrong in this table?

Roll Number	Name	Department	CGPA
23CS10003	Aryan Yadav	CSE	9.04
23CS10005	Ashutosh Sharma	CSE	8.65
23CS10063	Sanjana Kothi	CSE	9.86
23CS10005	Ashutosh Ranjan	CSE	8.65
23CS10091	Sibashis Som	CSE	9.83

Functional Dependencies

- There are usually a variety of constraints (rules) on the data in the real world.
- For example, some of the constraints that are expected to hold in a university database are:
  - Students and instructors are uniquely identified by their ID.
  - Each student and instructor has only one name.
  - Each instructor and student is (primarily) associated with only one department.
  - Each department has only one value for its budget, and only one associated building.

Expressing the Constraints

- Roll number → Name (Roll number determines Name)
- Roll number → CGPA
- Roll number → Department

Functional Dependencies (Cont.)

- An instance of a relation that satisfies all such real-world constraints is called a **legal instance** of the relation;
- A legal instance of a database is one where all the relation instances are legal instances
- Constraints on the set of legal relations.
- Require that the value for a certain set of attributes determines uniquely the value for another set of attributes.
- A functional dependency is a generalization of the notion of a *key.*

Legal Table Instance Satisfying the Constraints

Roll Number	Name	Department	CGPA
23CS10003	Aryan Yadav	CSE	9.04
23CS10005	Ashutosh Sharma	CSE	8.65
23CS10063	Sanjana Kothi	CSE	9.86
23CS10006	Ashutosh Ranjan	CSE	8.65
23CS10091	Sibashis Som	CSE	9.83

Functional Dependencies Definition

- Let *R* be a relation schema

$$\alpha \subseteq R \ \ and \ \ \beta \subseteq R$$

- The **functional dependency**

$$\alpha \rightarrow \beta$$

holds on *R* if and only if for any legal relations *r*(R), whenever any two tuples t_1 and t_2 of *r* agree on the attributes α, they also agree on the attributes β. That is,

$$t_1[\alpha] = t_2[\alpha] \ \Rightarrow \ t_1[\beta] = t_2[\beta]$$

- Example: Consider *r*(A,*B*) with the following instance of *r.*

1	4
1	5
3	7

- On this instance, $B \rightarrow A$ hold; $A \rightarrow B$ does **NOT** hold,

Closure of a Set of Functional Dependencies

- Given a set *F* set of functional dependencies, there are certain other functional dependencies that are logically implied by *F*.
  - If $A \rightarrow B$ and $B \rightarrow C$, then we can infer that $A \rightarrow C$
  - etc.
- The set of **all** functional dependencies logically implied by *F* is the **closure** of *F*.
- We denote the *closure* of *F* by F^+.

Keys and Functional Dependencies

- K is a superkey for relation schema R if and only if $K \rightarrow R$
- K is a candidate key for R if and only if
  - $K \rightarrow R$, and
  - for no $\alpha \subset K$, $\alpha \rightarrow R$
- Functional dependencies allow us to express constraints that cannot be expressed using superkeys. Consider the schema:

  in_dep (<u>*ID,*</u> *name, salary*<u>*, dept_name,*</u> *building, budget*).

  We expect these functional dependencies to hold:

  $$dept\_name \rightarrow building$$

  $$ID \rightarrow building$$

  but would not expect the following to hold:

  $$dept\_name \rightarrow salary$$

Use of Functional Dependencies

- We use functional dependencies to:
  - To test relations to see if they are legal under a given set of functional dependencies.
    - If a relation *r* is legal under a set *F* of functional dependencies, we say that *r* **satisfies** *F.*
  - To specify constraints on the set of legal relations
    - We say that *F* **holds on** *R* if all legal relations on *R* satisfy the set of functional dependencies *F.*
- Note: A specific instance of a relation schema may satisfy a functional dependency even if the functional dependency does not hold on all legal instances.
  - For example, a specific instance of *instructor* may, by chance, satisfy

    $$name \rightarrow ID.$$

Trivial Functional Dependencies

- *A* functional dependency is **trivial** if it is satisfied by all instances of a relation
- Example*:*
  - *ID, name* $\rightarrow$ *ID*
  - *name* $\rightarrow$ *name*
- In general, $\alpha \rightarrow \beta$ is trivial if $\beta \subseteq \alpha$

Closure of a Set of Functional Dependencies

- Given a set *F* set of functional dependencies, there are certain other functional dependencies that are logically implied by *F*.
  - If $A \rightarrow B$ and $B \rightarrow C$, then we can infer that $A \rightarrow C$
  - etc.
- The set of **all** functional dependencies logically implied by *F* is the **closure** of *F*.
- We denote the *closure* of *F* by F^+.

Closure of a Set of Functional Dependencies

- We can compute F^+, the closure of F, by repeatedly applying **Armstrong's Axioms:**
  - **Reflexive rule:** if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$
  - **Augmentation rule:** if $\alpha \rightarrow \beta$, then $\gamma \alpha \rightarrow \gamma \beta$
  - **Transitivity rule:** if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$
- These rules are
  - **Sound** -- generate only functional dependencies that actually hold, and
  - **Complete** -- generate all functional dependencies that hold.

Example of F^+

- $R = (A, B, C, G, H, I)$
  $F = \{ A \rightarrow B$
  $A \rightarrow C$
  $CG \rightarrow H$
  $CG \rightarrow I$
  $B \rightarrow H\}$
- Some members of F^+
  - $A \rightarrow H$
    - by transitivity from $A \rightarrow B$ *and* $B \rightarrow H$
  - $AG \rightarrow I$
    - by augmenting $A \rightarrow C$ with G, to get $AG \rightarrow CG$
      and then transitivity with $CG \rightarrow I$
  - $CG \rightarrow HI$
    - by augmenting $CG \rightarrow I$ to infer $CG \rightarrow CGI$,
      and augmenting of $CG \rightarrow H$ to infer $CGI \rightarrow HI$,
      and then transitivity

Closure of Functional Dependencies (Cont.)

- Additional rules:
  - **Union rule**: If $\alpha \rightarrow \beta$ holds *a*nd $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds.
  - **Decomposition rule**: If $\alpha \rightarrow \beta\gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds.
  - **Pseudotransitivity rule**:If $\alpha \rightarrow \beta$ holds *a*nd $\gamma\beta \rightarrow \delta$ holds, then $\alpha\gamma \rightarrow \delta$ holds.
- The above rules can be inferred from Armstrong's axioms.

Procedure for Computing F^+

- To compute the closure of a set of functional dependencies F:

$F^+ = F$
repeat
    for each functional dependency f in F^+
        apply reflexivity and augmentation rules on f
        add the resulting functional dependencies to F^+
    for each pair of functional dependencies f_1 and f_2 in F^+
        if f_1 and f_2 can be combined using transitivity
            then add the resulting functional dependency to F^+
until F^+ does not change any further

- **NOTE**: We shall see an alternative procedure for this task later

Closure of Attribute Sets

- Given a set of attributes α, define the ***closure*** of α **under** *F* (denoted by α^+) as the set of attributes that are functionally determined by α under *F*
- Algorithm to compute α^+, the closure of α under *F*

```
result := α;
while (changes to result) do
    for each β → γ in F do
      begin
        if β ⊆ result then  result := result ∪ γ
      end
```

Example of Attribute Set Closure

- $R = (A, B, C, G, H, I)$
- $F = \{A \rightarrow B$
  $A \rightarrow C$
  $CG \rightarrow H$
  $CG \rightarrow I$
  $B \rightarrow H\}$
- $(AG)^+$
  1. *result = AG*
  2. *result = ABCG* ($A \rightarrow C$ and $A \rightarrow B$)
  3. *result = ABCGH* ($CG \rightarrow H$ and $CG \subseteq AGBC$)
  4. *result = ABCGHI* ($CG \rightarrow I$ and $CG \subseteq AGBCH$)
- Is *AG* a candidate key?
  1. Is AG a super key?
     1. Does $AG \rightarrow R$? == Is $R \supseteq (AG)^+$
  2. Is any subset of AG a superkey?
     1. Does $A \rightarrow R$? == Is $R \supseteq (A)^+$
     2. Does $G \rightarrow R$? == Is $R \supseteq (G)^+$
     3. In general: check for each subset of size *n-1*

Uses of Attribute Closure

There are several uses of the attribute closure algorithm:

- Testing for superkey:
  - To test if α is a superkey, we compute α^{+}, and check if α^{+} contains all attributes of *R*.
- Testing functional dependencies
  - To check if a functional dependency $\alpha \rightarrow \beta$ holds (or, in other words, is in F^{+}), just check if $\beta \subseteq \alpha^{+}$.
  - That is, we compute α^{+} by using attribute closure, and then check if it contains β.
  - Is a simple and cheap test, and very useful
- Computing closure of F
  - For each $\gamma \subseteq R$, we find the closure γ^{+}, and for each $S \subseteq \gamma^{+}$, we output a functional dependency $\gamma \rightarrow S$.

Canonical Cover

- Suppose that we have a set of functional dependencies *F* on a relation schema. Whenever a user performs an update on the relation, the database system must ensure that the update does not violate any functional dependencies; that is, all the functional dependencies in *F* are satisfied in the new database state.
- If an update violates any functional dependencies in the set *F*, the system must roll back the update.
- We can reduce the effort spent in checking for violations by testing a simplified set of functional dependencies that has the same closure as the given set.
- This simplified set is termed the **canonical cover**
- To define canonical cover we must first define **extraneous attributes.**
    - An attribute of a functional dependency in *F* is **extraneous** if we can remove it without changing F^+

Extraneous Attributes

- Removing an attribute from the left side of a functional dependency could make it a stronger constraint.
  - For example, if we have $AB \rightarrow C$ and remove B, we get the possibly stronger result $A \rightarrow C$. It may be stronger because $A \rightarrow C$ logically implies $AB \rightarrow C$, but $AB \rightarrow C$ does not, on its own, logically imply $A \rightarrow C$
- But, depending on what our set F of functional dependencies happens to be, we may be able to remove B from $AB \rightarrow C$ safely.
  - For example, suppose that
  - $F = \{AB \rightarrow C, A \rightarrow D, D \rightarrow C\}$
  - Then we can show that F logically implies $A \rightarrow C$, making extraneous in $AB \rightarrow C$.

Extraneous Attributes (Cont.)

- Removing an attribute from the right side of a functional dependency could make it a weaker constraint.
  - For example, if we have $AB \rightarrow CD$ and remove C, we get the possibly weaker result $AB \rightarrow D$. It may be weaker because using just $AB \rightarrow D$, we can no longer infer $AB \rightarrow C$.
- But, depending on what our set F of functional dependencies happens to be, we may be able to remove C from $AB \rightarrow CD$ safely.
  - For example, suppose that

    F = { $AB \rightarrow CD$, $A \rightarrow C$.
  - Then we can show that even after replacing $AB \rightarrow CD$ by $AB \rightarrow D$, we can still infer \$$AB \rightarrow C$ and thus $AB \rightarrow CD$.

Extraneous Attributes

- An attribute of a functional dependency in *F* is **extraneous** if we can remove it without changing F^+
- Consider a set *F* of functional dependencies and the functional dependency $\alpha \rightarrow \beta$ in *F*.
  - **Remove from the left side**: Attribute A is **extraneous** in α if
    - $A \in \alpha$ and
    - *F* logically implies $(F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - A) \rightarrow \beta\}$.
  - **Remove from the right side**: Attribute *A* is **extraneous** in β if
    - $A \in \beta$ and
    - The set of functional dependencies $(F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$ logically implies *F.*
- *Note:* implication in the opposite direction is trivial in each of the cases above, since a "stronger" functional dependency always implies a weaker one

Testing if an Attribute is Extraneous

- Let *R* be a relation schema and let *F* be a set of functional dependencies that hold on *R* . Consider an attribute in the functional dependency $\alpha \rightarrow \beta$.
- To test if attribute $A \in \beta$ is extraneous in β
  - Consider the set:
    $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$,
  - check that α^+ contains *A;* if it does, *A* is extraneous in β
- To test if attribute $A \in \alpha$ is extraneous in α
  - Let $\gamma = \alpha - \{A\}$. Check if $\gamma \rightarrow \beta$ can be inferred from *F.*
    - Compute γ^+ using the dependencies in *F*
    - If γ^+ includes all attributes in β then , *A* is extraneous in α

Examples of Extraneous Attributes

- Let $F = \{AB \rightarrow CD, A \rightarrow E, E \rightarrow C\}$
- To check if C is extraneous in $AB \rightarrow CD$, we:
  - Compute the attribute closure of AB under $F' = \{AB \rightarrow D, A \rightarrow E, E \rightarrow C\}$
  - The closure is *ABCDE*, which includes *CD*
  - This implies that *C* is extraneous

Canonical Cover

A **canonical cover** for F is a set of dependencies F_c such that

- F logically implies all dependencies in F_c , and
- F_c logically implies all dependencies in F, and
- No functional dependency in F_c contains an extraneous attribute, and
- Each left side of functional dependency in F_c is unique. That is, there are no two dependencies in F_c
  - $\alpha_1 \rightarrow \beta_1$ and $\alpha_2 \rightarrow \beta_2$ such that
  - $\alpha_1 = \alpha_2$

Canonical Cover

- To compute a canonical cover for *F*:

  repeat

  Use the union rule to replace any dependencies in *F* of the form

  $\alpha_1 \rightarrow \beta_1$ and $\alpha_1 \rightarrow \beta_2$ with $\alpha_1 \rightarrow \beta_1 \beta_2$

  Find a functional dependency $\alpha \rightarrow \beta$ in F_c with an extraneous attribute either in α or in β

  /* Note: test for extraneous attributes done using $F_{c,}$ not F*/

  If an extraneous attribute is found, delete it from $\alpha \rightarrow \beta$

  until (F_c not change

- Note: Union rule may become applicable after some extraneous attributes have been deleted, so it has to be re-applied

Example: Computing a Canonical Cover

- $R = (A, B, C)$
  $F = \{A \rightarrow BC$
  $B \rightarrow C$
  $A \rightarrow B$
  $AB \rightarrow C\}$
- Combine $A \rightarrow BC$ and $A \rightarrow B$ into $A \rightarrow BC$
  - Set is now $\{A \rightarrow BC, B \rightarrow C, AB \rightarrow C\}$
- A is extraneous in $AB \rightarrow C$
  - Check if the result of deleting A from $AB \rightarrow C$ is implied by the other dependencies
    - Yes: in fact, $B \rightarrow C$ is already present!
  - Set is now $\{A \rightarrow BC, B \rightarrow C\}$
- C is extraneous in $A \rightarrow BC$
  - Check if $A \rightarrow C$ is logically implied by $A \rightarrow B$ and the other dependencies
    - Yes: using transitivity on $A \rightarrow B$ *and* $B \rightarrow$ C.
      - Can use attribute closure of A in more complex cases
- The canonical cover is: $A \rightarrow B$
  $B \rightarrow C$

Summary of Functional Dependencies

- Roll → Name
- Roll → CGPA
- Roll → Department

Roll	Name	Department	CGPA
23CS10003	Aryan Yadav	CSE	9.04
23CS10005	Ashutosh Sharma	CSE	8.65
23CS10063	Sanjana Kothi	CSE	9.86
23CS10006	Ashutosh Ranjan	CSE	8.65
23CS10091	Sibashis Som	CSE	9.83

If two rows have same values of Roll it SHOULD have same value for Name

Closure of a set of FDs

- ***F^+*:** Set of functional dependencies logically implied by *F*
- Need to check if a table is legal wrt the closure ***F^+*** and not just *F*
- *Armstrong's Axioms*
  - **Reflexive rule:** if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$
  - **Augmentation rule:** if $\alpha \rightarrow \beta$, then $\gamma\,\alpha \rightarrow \gamma\,\beta$
  - **Transitivity rule:** if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$

Example

Example. Assume that there are 4 attributes A, B, C, D, and that $F = \{A \to B, B \to C\}$. Then, F^+ includes all the following FDs:

$A \to A$, $A \to B$, $A \to C$, $B \to B$, $B \to C$, $C \to C$, $D \to D$, $AB \to A$, $AB \to B$, $AB \to C$, $AC \to A$, $AC \to B$, $AC \to C$, $AD \to A$, $AD \to B$, $AD \to C$, $AD \to D$, $BC \to B$, $BC \to C$, $BD \to B$, $BD \to C$, $BD \to D$, $CD \to C$, $CD \to D$, $ABC \to A$, $ABC \to B$, $ABC \to C$, $ABD \to A$, $ABD \to B$, $ABD \to C$, $ABD \to D$, $BCD \to B$, $BCD \to C$, $BCD \to D$, $ABCD \to A$, $ABCD \to B$, $ABCD \to C$, $ABCD \to D$.

Attribute Closure

Attribute Closure

If X is an attribute set, the closure X^+ is the set of all attributes B such that:

$$X \longrightarrow B$$

In other words, X^+ includes all attributes that are functionally determined from X

Example

Roll	Name	Department	CGP A
23CS10003	Aryan Yadav	CSE	9.04
23CS10005	Ashutosh Sharma	CSE	8.65
23CS10063	Sanjana Kothi	CSE	9.86
23CS10006	Ashutosh Ranjan	CSE	8.65
23CS10091	Sibashis Som	CSE	9.83

- Roll → Name, CGPA, Department

Attribute Closure Algorithm

- Let $X = \{A_1, A_2, \ldots, A_n\}$
- **UNTIL** X doesn't change **REPEAT**:

  IF $B_1, B_2, \ldots, B_m \longrightarrow C$ is an FD **AND**

  $B_1, B_2, \ldots, B_m$ are all in X

  THEN add C to X

Example

R(A, B, C, D, E, F)

- $A, B \longrightarrow C$
- $A, D \longrightarrow E$
- $B \longrightarrow D$
- $A, F \longrightarrow B$

Compute the attribute closures:

- $\{A, B\}^{+} = \{A, B, C, D, E\}$
- $\{A, F\}^{+} = \{A, F, B, D, E, C\}$

Example

Example: Let F be the set of following FDs:

$$\text{cid} \rightarrow \text{title}$$
$$\text{title} \rightarrow \text{dept}$$
$$\text{cid, year} \rightarrow \text{dept}$$
$$\text{cid, year} \rightarrow \text{cid, dept}$$

What is the closure of $X = \{\text{cid, year}\}$?

Keys and Attribute Closures

- Compute X^+ for all sets of attributes X
- If $X^+ = all\ attributes$, then X is a superkey
- If no subset of X is a superkey, then X is also a key

Example

Product(name, category, price, color)

- $name \longrightarrow color$
- $color, category \longrightarrow price$

Superkeys:

- $\{name, category\}, \{name, category, price\}$
  $\{name, category, color\}, \{name, category, price, color\}$

Keys:

- $\{name, category\}$

FD Closure Algorithm

algorithm (F)
/* F is a set of FDs */
1. $F^+ = \emptyset$
2. **for** each possible attribute set X
3. compute the closure X^+ of X on F
4. **for** each attribute $A \in X^+$
5. add to F^+ the FD: $X \to A$
5. **return** F^+

Example. Assume that there are 4 attributes A, B, C, D, and that $F = \{A \rightarrow B, B \rightarrow C\}$. To compute F^+, we first get:

- $A^+ = AB^+ = AC^+ = ABC^+ = \{A, B, C\}$
- $B^+ = BC^+ = \{B, C\}$
- $C^+ = \{C\}$
- $D^+ = \{D\}$
- $AD^+ = \{A, D\}$
- $BC^+ = \{B, C\}$
- $BD^+ = BCD^+ = \{B, C, D\}$
- $ABD^+ = ABCD^+ = \{A, B, C, D\}$
- $ACD^+ = \{A, C, D\}$

It is easy to generate the FDs in F^+ from the closures of the above attribute sets.

Cover (minimal basis) of a Set of FD

S is a **minimal basis** for a set F if FDs if:

- $S^+ = F^+$
- every FD in S has one attribute on the right side
- if we remove any FD from S, the closure is not F^+
- if for any FD in S we remove one or more attributes from the left side, the closure is not F^+

Example

F

- $A \longrightarrow B$
- $A, B, C, D \longrightarrow E$
- $E, F \longrightarrow G, H$
- $A, C, D, F \longrightarrow E, G$

Minimal Cover

- $A \longrightarrow B$
- $A, C, D \longrightarrow E$
- $E, F \longrightarrow G$
- $E, F \longrightarrow H$

Multi-Valued Dependencies

Name →→ Phone

If the first two tuples are present, the last two tuple MUST be present in the table

Name	Dept	Phone	Research Area
Pabitra	CSE	9434724097	Machine Learning
Pabitra	CSE	9434024099	Image processing
Pabitra	CSE	9434724097	Image Processing
Pabitra	CSE	9434024099	Machine Learning

Name	Dept	Phone	Research Area
Pabitra	CSE	9434724097, 9434024099	Machine Learning, Image Processing

Definition MVD

Any attribute say a multiple define another attribute b; if any legal relation r(R), for all pairs of tuples t1 and t2 in r, such that,

```
t1[a] = t2[a]
```

Then there exists t3 and t4 in r such that.

```
t1[a] = t2[a] = t3[a] = t4[a]
t1[b] = t3[b]; t2[b] = t4[b]
t1[c] = t4[c]; t2[c] = t3[c]
```