Functional Dependencies

Overview

- Basic Definitions – FD
- Inference Rules
- Trivial and nontrivial dependencies
- Closure of a set of dependencies
- Closure of a set of attributes – finding key K
- Irreducible sets of dependencies – minimal cover

Functional Depencency

- A functional dependency is a property of the semantics or meaning of the attributes – that is how they relate to one another.
- Whenever two sets of attributes indicate that a FD should hold, dependency is specified as a constraint.
- Main use of FD: to describe the relation schema by specifying constraints on its attributes that must *hold at all times.*
- For example:
  1) SSN --> ename   2) Pnumber --> {Pname, Plocation}
  3) {SSN, Pnumber} --> Hours

FD : Basic definitions

- Two cases:
- Case a:

  the value of a given relvar

  at a given point in time.
- Case b:

  the set of all possible values

  that the given relvar might

  assume at different times.

SCP

SUPPLIER_ NUMBER	CITY	PART_ NUMBER	QUANTITY
S1	Bombay	P1	100
S1	Bombay	P2	100
S2	Chennai	P1	200
S2	Chennai	P2	200
S3	Chennai	P2	300
S4	Bombay	P2	400
S4	Bombay	P4	400
S4	Bombay	P5	400

FD : Case A

- Let *r* be a relation, and let X and Y be arbitrary subsets of the set of attributes of *r*.

  Then we say that Y is functionally dependent on X – in symbols,

  $X \rightarrow Y$

  X **functionally determines** Y or X arrow Y

  – if and only if each X value in *r* has associated with it precisely one Y value in *r*.
- Ex: { SUPPLIER_NUMBER } --> { CITY }

FD

- Thus, X determines Y if, and only if, whenever two tuples agree on their X-value, they must necessarily agree on their Y-value.
  - If a constraint on R states that, there can not be more than one tuple with a given X-value in any instance r(R) – that is, X is a candidate key of R.
  - If X --> Y in R, does not say whether or not Y --> X in R.

FD : Case A

- {SUPPLIER_NUMBER} --> { CITY }

  {SUPPLIER_NUMBER, PART_NUMBER} -> { QUANTITY }

  {SUPPLIER_NUMBER, PART_NUMBER} -> { CITY }

  {SUPPLIER_NUMBER, PART_NUMBER} ->{CITY,QUANTITY}

  {SUPPLIER_NUMBER} -> {QUANTITY}

  {QUANTITY} -> {SUPPLIER_NUMBER}

  {SUPPLIER_NUMBER, PART_NUMBER} -> {SUPPLIER_NUMBER}
- Left and right sides of an FD are called the **determinant** and the **dependent** respectively.

Basic definitions

- Not interested in FDs that hold for the particular value that the relvar have at some particular time.
- Rather those FDs that hold for *all possible values* of that relvar.
- For example, any two tuples appearing in SCP at same time with the same supplier number must necessarily have the same city as well.

```
CONSTRAINT SUPPLIER_NUMBER_CITY_FD
      FORALL SCPX FORALL SCPY
            (IF SCPX.SUPPLIER_NUMBER = SCPY.SUPPLIER_NUMBER
                  THEN SCPX.CITY=SCPY.CITY  END IF );

-------------       equivalent to ----------------
{ SUPPLIER_NUMBER } → { CITY }
```

FD : Case B

- Let *R* be a relation variable, and let X and Y be arbitrary subsets of the set of attributes of *R*.

  Then we say that Y is functionally dependent on X – in symbols,

  $$X \longrightarrow Y$$

  X **functionally determines** Y or X arrow Y

  – if and only if **in every possible legal value of *R***, each X value has associated with it precisely one Y value.

- Ex: { SUPPLIER_NUMBER } -> { CITY }

FD : Case B

- Some (time-independent) FDs that apply to SCP:

  SUPPLIER_NUMBER -> CITY

  {SUPPLIER_NUMBER, PART_NUMBER} -> CITY

  {SUPPLIER_NUMBER, PART_NUMBER} -> QUANTITY

  {SUPPLIER_NUMBER, PART_NUMBER}->{CITY,QUANTITY}

  {SUPPLIER_NUMBER, PART_NUMBER } -> {SUPPLIER_NUMBER }

- Following FD do no hold *for all time*:

  SUPPLIER_NUMBER -> QUANTITY

  QUANTITY -> SUPPLIER_NUMBER

Trivial and nontrivial FD

- An FD is trivial if and only if the right side is a subset of the left side.

  {SUPPLIER_NUMBER, PART_NUMBER} -> { SUPPLIER_NUMBER }

- Trivial dependencies are not very interesting in practice.

Inference

- Some FDs might imply others:

  {SUPPLIER_NUMBER, PART_NUMBER}->{CITY,QUANTITY}

  implies

  {SUPPLIER_NUMBER, PART_NUMBER} → CITY

  {SUPPLIER_NUMBER, PART_NUMBER} → QUANTITY
- Its impossible to specify all possible FDs for a given situation.

Inference

- An FD $X \rightarrow Y$ is *inferred from* or *implied by* a set of dependencies F specified on R, if $X \rightarrow Y$ holds in every legal relation state r of R.
- The *inferred* FD's need not be explicitly stated in addition to the given FD.
- Closure – includes all possible dependencies that can be inferred from the given set F.

Closure of FD F

- The set of all dependencies that include F as well as all dependencies that can be inferred from F is called the closure of F; it is denoted by F+.
- To determine a systematic way to infer dependencies , set of **inference rules** are used to infer new dependencies from a given set of dependencies.
- Armstrong's axioms provides set of *inference rules* by which new FDs can be inferred from given ones.

Armstrong Axioms

- Let A, B, and C be arbitrary subsets of the set of attributes of the given relvar R, then

IR 1. Reflexivity: If B is a subset of A, then $A \rightarrow B$.

IR 2. Augmentation: If $A \rightarrow B$, then $AC \rightarrow BC$.

IR 3. Transitivity: If $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$.

- The rules are **complete**, in a sense that, given a set S of FDs, all FDs implied by S can be derived from S using the rules.
- They are also **sound**, in a sense that, no additional FDs can be so derived.

Inference Rules

- IR1 is trivial; a FD $A \rightarrow B$ is trivial, if A is superset of B; otherwise it is nontrivial.
- Several rules can be derived from the three:

4. Self-determination: $A \rightarrow A$.

5. Decomposition: If $A \rightarrow BC$, then $A \rightarrow B$ and $A \rightarrow C$.

6. Union: If $A \rightarrow B$ and $A \rightarrow C$, then $A \rightarrow BC$.

7. Composition: If $A \rightarrow B$ and $C \rightarrow D$, then $AC \rightarrow BD$.

Inference Rules

- Proof IR 6:

1.	$A \to B$	(given)
2.	$A \to C$	(given)
3.	$A \to AB$	(using IR2 on 1)
4.	$AB \to BC$	(using IR2 on 2)
5.	$A \to BC$	(using IR3 on 3 and 4)

Closure of F

- Suppose given a relvar R with attributes A, B, C, D, E, F and the FDs:

  $A \rightarrow BC$, $B \rightarrow E$, $CD \rightarrow EF$

  show that the FD $AD \rightarrow F$ holds for R, and is thus a member of the closure of the given set:

Closure of F

- Suppose given a relvar R with attributes A, B, C, D, E, F

  and the FDs:

  $A \rightarrow BC$, $B \rightarrow E$, $CD \rightarrow EF$

  show that the FD $AD \rightarrow F$ holds for R, and is thus a

  member of the closure of the given set:

1.	$A \rightarrow BC$	(given)
2.	$A \rightarrow C$	(use IR5 on 1)
3.	$AD \rightarrow CD$	(use IR2 on 2)
4.	$CD \rightarrow EF$	(given)
5.	$AD \rightarrow EF$	(use IR3 on 3 and 4)
6.	$AD \rightarrow F$	(use IR5 on 5)

Closure of a set of attributes

- Given a relvar R, a set X of attributes of R, and a set F of FDs that hold for R,
  we can determine the set of all attributes of R that are functionally dependent on X– the closure X^+ of X under F.
- For each such set of attributes X, determine the set X^+ of attributes that are functionally determined by X based on F; X^+ is called the **closure of X under F**.

Closure of a set of attributes

- Algorithm: the closure of X under F

```
CLOSURE [X,F] := X;
do forever ;
    for each FD Y → Z in F
        do ;
            if Y ⊆ CLOSURE [X,F]
            then CLOSURE [X,F] := closure [X,F] ∪ Z;
        end
    if CLOSURE[X,F] did not change on this iteration then
    leave the loop;
end;
```

Closure of a set of attributes

- Suppose given a relvar R with attributes A, B, C, D, E, F and FDs:

  $A \rightarrow BC$, $E \rightarrow CF$, $B \rightarrow E$, $CD \rightarrow EF$

  Compute the closure {A,B}+ of the set of attributes {A,B} under the set of FDs :

Closure of a set of attributes

- $A \rightarrow BC$, $E \rightarrow CF$, $B \rightarrow E$, $CD \rightarrow EF$

  Compute the closure $\{A,B\}^+$ of the set of attributes $\{A,B\}$

  under the set of FDs :

CLOSURE [Z,S] := {A,B} ;

1. $A \subseteq \{A, B\}$ then CLOSURE [Z,S] := {A, B, C} ;
2. $E \not\subseteq \{A, B, C\}$ --- no change
3. $B \subseteq \{A, B, C\}$ then CLOSURE [Z,S] := {A, B, C, E} ;
4. $CD \not\subseteq \{A, B, C, E\}$ --- no change
5. $A \subseteq \{A, B, C, E\}$ ---- right side attributes already in closure
6. $E \subseteq \{A, B, C, E\}$ then CLOSURE [Z,S] := {A, B, C, E, F} ;
7. $B \subseteq \{A, B, C, E, F\}$ --- no change
8. $CD \not\subseteq \{A, B, C, E, F\}$ --- no change

no change in CLOSURE[Z,S] for next iteration; terminate;

therefore Closure of {A,B} is $\{A,B\}+ = \{A, B, C, E, F\}$

Key K for R given F

- Superkey for a given relvar R are subsets K of the attributes of R such that the FD:

  $$K \rightarrow A$$

  holds true for every attribute A of R.
- K is a *superkey* if and only if the closure K+ of K – under the given set of FDs – is precisely the set of all attributes of R.
- K is a candidate key if and only if it is an irreducible superkey.

Key K for R given F

- Finding a key K for R

> A relation R ans set of FD F
>
> 1. Set K := R
>
> 2. For each attribute A in K
>
>   Compute (K-A)+ with respect to F;
>
>   If (K-A)+ contains all attributes in R, then
>
>   set K:= K-{A};

Key K for R given F

- Given the set of FD F: A $\rightarrow$ BC, E $\rightarrow$ CF, B $\rightarrow$ E, CD $\rightarrow$ EF for a relation R(A,B,C,D,E,F).
- Compute the key K for R given F.

Corollary

- 1. Is K – a super key? (uniqueness property)

  1. Does $K \rightarrow R$? == Is $(K)^+ \supseteq R$

- 2. Is any subset of K – a superkey ? (Irreducible property)

  Let K=(AB)

  1.Does $A \rightarrow R$? == Is $(A)^+ \supseteq R$

  2.Does $B \rightarrow R$? == Is $(B)^+ \supseteq R$

Equivalence of Sets of FD

- Let S1 and S2 be two sets of FD. If every FD in S2 can be inferred from S1 or every FD in S2 is also in $S1^+$, then we say that S1 is a **cover** for S2.
- If S1 is a cover for S2 and S2 is a cover for S1 – i.e. $S1^+ = S2^+$, we say that S1 and S2 are **equivalent**.

Irreducible sets of dependencies

- A set S of FDs to be irreducible if and only if it satisfies the following three properties:
- 1. the right side of every FD in S involves just one attribute.
- 2. the left side of every FD in S is irreducible – no attribute can be discarded from the left side without changing the closure S^+

  Ex: {supplier_number, part_number} --> qty
- 3. no FD in S can be discarded from S without changing the closure S+
- Also called as Minimal cover F for a set of FDs S.

Irreducible sets of dependencies

- Minimal cover F for a set of FD

1. Set F := E

2. Replace each FD X → {A1, A2, …, An} by the n functional dependencies X → A1, X → A2, …. , X → An.

* this places FD in Canonical form.

3. For each FD X → A in F

    for each attribute B that is an element of X

        if { {F – {X→A}} U {(X-{B}) → A } } is equivalent to F

            then replace X → A with (X- {B}) → A in F

            * removal of extraneous attribute B in X.

Irreducible sets of dependencies

- Minimal cover F for a set of FD

4. For each remaining FD $X \rightarrow A$ in F

   if { {F – {X→A}} is equivalent to F

   then remove $X \rightarrow A$ from F

   * removal of redundant FD from F

Irreducible sets of dependencies

- Given a R with attributes A,B,C,D, and FDs:
  1. A -> BC
  2. B -> C
  3. A -> B
  4. AB -> C
  5. AC -> D
- (1) can be rewritten as A -> B and A -> C
- Now the set of FDs are:
  1. A -> B
  2. A -> C
  3. B -> C
  4. AB -> C
  5. AC -> D

Irreducible sets of dependencies

- C can be eliminated from AC -> D:
  from (2) A -> AC
  AC -> D --given (5)
  hence A -> D
- Now FDs are:
  1. A -> B
  2. A -> C
  3. B -> C
  4. AB -> C
  5. A -> D
- Next, AB -> C can be eliminated:
  from (2), AB -> CB – by augmentation
  AB -> C -- by decomposition

Irreducible sets of dependencies

- Now FDs are:
  1. A -> B
  2. A -> C
  3. B -> C
  4. A -> D
- A -> C is implied by A -> B and B -> C
- Now FDs are:
  1. A -> B
  2. B -> C
  3. A -> D
- The above set is **irreducible!**

References

- *Chapter 11: Functional Dependencies*

  An introduction to database systems, *CJ. Date*
- *Fundamentals of Database Systems, 7th Edition,*

  Elmasri and Navathe, Pearson