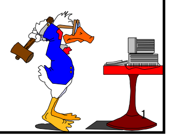
Functional Dependencies



Functional Dependency

- Functional dependency describes the relationship between attributes in a relation.
- Eg. if A and B are attributes of relation R, B is functionally dependent on A (denoted A → B), if each value of A in R is associated with exactly one value of B in R.

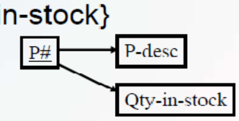
FD Diagram

Stock

P#	P-desc.	Qty-in-stock
P2	nut	5000
P1	bolt	8300
P3	washer	9750
P4	nut	2326

(P#, P-desc, Qty-in-stock)

P# ➔ {P-desc, Qty-in-stock}



Motivation:

- Functional dependencies help in accomplishing the following two goals:
- (a) controlling redundancy and
- (b) enhancing data reliability.

Definition

- Let U_j be a subset of the universal set of attributes u .
- A functional dependencies (fd) is a constraint of U_j of the form X→Y where X, $Y \subseteq U_j$.
- Relation $R(U_j)$ satisfies FD X→ Y or X→Y holds in $R(U_j)$ if for every two tuples in $R(U_j)$, say t1 and t2, we have:
- if t1[X] = t2[X] => t2[Y] = t1[Y]

Definition

- if two tuples agree on the 'X' attribute, they *must* agree on the 'Y' attribute, too
- If X →Y we say X functionally determines Y.
- Notice that X→Y implies many-to-one or one-to-one mapping.

Example

- Consider the Emp schema below:
- EMP (name, salary, dept, mgr)

Example

- Consider the following data dependencies
- 1. Each employee has one salary
  - Name $\longrightarrow$ salary
- 2. Each employee works in only one department
  - name $\longrightarrow$ dept

Example

- Some employees may work in more than one department
- name -/$\longrightarrow$ dept
  - e1[name] = e2[name] but
  - e1[dept] $\neq$ e2[dept]

Inference Axioms

- (A-axioms or Armstrong's Axioms)
  - An inference axiom is a rule that states
  - if a relation satisfies certain FDs then it must satisfy certain other FDs.

Inference Axioms

- The *closure of F* (usually written as ***F+***) is the set of all functional dependencies that may be logically derived from *F*.
- Often *F* is the set of most obvious and important functional dependencies and
- ***F+***, the closure, is the set of all the functional dependencies including *F* and those that can be deduced from *F*.

Inference Axioms

- The closure is important and may, for example, be needed in finding one or more candidate keys of the relation.
- A set of inference rules, called *Armstrong's axioms*, specifies how new functional dependencies can be inferred from given ones.

Armstrong's axioms

- Let A, B, and C be subsets of the attributes of the relation R. Armstrong's F axioms are as follows:

(F1) *Reflexivity*

If B is a subset of A, then A → B

(F2) *Augmentation*

If A → B, then A,C → B,C

(F3) *Transitivity*

If A → B and B → C, then A → C

13

Armstrong's axioms

- Further rules can be derived from the first three rules that simplify the practical task of computing X+.
- Let D be another subset of the attributes of relation R, then:

(F4) *Self-determination*

A → A

(F5) *Decomposition*

If A → B,C, then A → B and A → C

14

Armstrong's axioms

(F6) *Union*

If A → B and A → C, then A → B,C

(F7) *Composition*

If A → B and C → D then A,C → B,D

15

Examples of the use of Armstrong's Axioms

- Consider R = (Street, Zip, City) ;
- F = {City Street ⟶ Zip, Zip ⟶ City}
- Show that : Street Zip ⟶ Street Zip City

16

Proof:

1. Zip ⟶ City – Given
2. Street Zip ⟶ Street City – Augmentation of (1) by Street
3. City Street ⟶ Zip – Given
4. City Street ⟶ City Street Zip
   - Augmentation of (3) by City Street
5. Street Zip ⟶ City Street Zip
   – Transitivity of (2) and (4)

17

Example 2

- Let R = (ABCDEGHI)
- F = {AB ⟶ E, AG ⟶ J, BE ⟶ I, E ⟶ G GI ⟶ H}
- Show that AB ⟶ GH is derived by F

18

Proof

1. AB ⟶ E - Given
2. AB ⟶ AB – Reflexivity
3. AB ⟶ B - Projectivity from (2)
4. AB ⟶ BE – Union from (1) and (3)
5. BE ⟶ I - Given
6. AB ⟶ I – Transitivity from (4) and (5)

19

Proof Cont:

1. E ⟶ G – Given
2. AB ⟶ G – Transitivity from (1) and (7)
3. AB ⟶ GI – Union from (6) and (8)
4. GI ⟶ H – Given
5. AB ⟶ H – Transitivity from (9) and (10)
6. AB ⟶ GH – Union from (8) and (11)

20

B-Axioms

- **Definition:** Set of inference axioms which are not a subset of F1 to F7.
- Let r(R) with WXY and Z be subsets of R, and C an attribute in R then:

B1. Reflexivity – X ⟶ X

B2. Accumulation – If (X ⟶ YZ) and (Z ⟶ CW) then X ⟶ YZC

B3. Projectivity – If (X ⟶ YZ) then X ⟶ Y

21

Example

- Let R = (ABCDEGHI) F = {AB ⟶ E, AG ⟶ J, BE ⟶ I, E ⟶ G, GI ⟶ H}
- Show F |= AB ⟶ GH using only B-axioms

22

Example

- EI ⟶ EI – Reflexivity (B1)
- E ⟶ G – Given
- EI ⟶ EIG – Accumulation (B2)
- EI ⟶ GI – Projectivity (B3) from (3)
- GI ⟶ H – Given

23

Example

- EI ⟶ GHI – Accumulation from (4) and (5)
- EI ⟶ GH – Projectivity from (6)
- AB ⟶ AB – Reflexivity
- AB ⟶ E – Given
- AB ⟶ ABE – Accumulation from (8) and (9)
- BE ⟶ I – Given

24

Example

- AB → ABEI – Accumulation from (10) and (11)
- AB → ABEIG – Accumulation from (4) and (12)
- AB → ABEGHI – Accumulation from (7) and (13)
- AB → GH – Projectivity from (14)
  - Therefore we have found a derivation sequence for AB → GH using only the B-axioms

25

RAP – Derivation Sequence

- RAP: Reflexivity, Augmentation, Projectivity
- Definition: Consider derivation sequences for X → Y on a set F of f.d.s using the B-axioms that satisfy the following constraints:
  - The first f.d. is X → Y
  - The last f.d. is X → Y

26

RAP – Derivation Sequence

- Every FD other than the first and last is either an f.d. in F (given) or and f.d. of the form X → Z that was derived using axiom B2

27

RAP – Derivation Sequence

- Such a derivation is called a RAP-derivation sequence, for the order in which the axioms are used.

28

Example:

- Let R = (ABCDEGHI) F = {AB → E AG → J, BE → I, E → G, GI → H}
- Find a RAP-sequence for AB → GH

29

Example:

- AB → AB (B1)
- AB → E - Given *
- AB → ABE (B2)
- BE → I - Given *
- AB → ABEI (B2)

30

Example:

- $E \rightarrow G$ - Given *
- $AB \rightarrow ABEIG$ (B2)
- $GI \rightarrow H$ - Given *
- $AB \rightarrow ABIGH$ (B2)
- $AB \rightarrow GH$ (B3)

31

Covers for Functional Dependencies

- If every set of FDs, F can be inferred from another set of FDs, G, then G is said to cover F.
- *Also E* is covered by *F* if every FD in *E* is also in F^{+}.
- *E* and *F* are equivalent if $E^{+} = F^{+}$, i.e, E covers F and F covers E.

32

Minimal Cover for a set of FDs

- It is always useful to identify a simplified set of FDs, G_c, that is equivalent to F.
- This means that they have the same closure (F+) as F and its no further reducible.
- We try to get the set G where $F \equiv G$.
- This means that we could enforce G or F and the valid database states will remain the same.

33

Minimal Cover for a set of FDs (cont.)

- In practice the minimal cover is useful because the effort required to check for violations in the database is minimized therefore improving the database performance
- F can be its own minimal cover also known as canonical cover.
- There can be several minimal covers of F.

34

Minimal Cover for a set of FDs (cont.)

- Formally G_c is the minimal cover of F if:
  - $G_c \equiv F$ (G_c and F are equivalent)
  - The dependant (RHS) in every FD in G_c is a singleton attribute. This is called standard or canonical form.
  - No FD in G_c is redundant. In other words, if any FD in G_c is discarded, then G_c would be no longer equivalent to F.

35

Minimal Cover for a set of FDs (cont.)

- The determinant (LHS) of every FD in G_c is irreducible. In other words, if any attribute is discarded from the determinant of any FD in G_c, then G_c would be no longer equivalent to F.

36

Algorithm to compute the minimal cover

1. Set G to F.
2. Convert all FDs into standard (canonical) form.
3. Remove all redundant attributes from the determinant (LHS) of the FDs from G
4. Remove all redundant FDs from G.

Two Notes:

- This algorithm might produce different results based on the order of candidates removal.
- Steps 3 and 4 aren't interchangeable.

Algorithm to compute the minimal cover

1. G := f;
2. Replace each FD $X \rightarrow A_1, A_2, \ldots, A_K$ in G by the *k* FDs $X \rightarrow A_1$, $X \rightarrow A_2$, $X \rightarrow A_K$;
3. for each FD $X \rightarrow A$ in G

   for each attribute B X

   if $(X - B)^+$ with-respect-to G contains A

   then replace $X \rightarrow A$ with $X - \{B\} \rightarrow A$ in G;

Algorithm to compute the minimal cover

4. For each FD $X \rightarrow A$ in G

   if X^+ with-respect-to G-$\{X \rightarrow A\}$ contains A

   then remove $X \rightarrow A$ from G;

- There is at least one minimal cover for any F, maybe several

Examples

- Consider a set of attributes {ABC} and set of FDs F:

  fd1: A->C　　fd2: (AC)->B

  fd3: B->A　　fd4: C->(AB)

- Rewrite in standard form fd4:
  - fd4a: C->A　　fd4b: C->B

Examples (cont.)

- Based on fd4b, A in fd2 is redundant. We remove it. Now we remove fd4b because is identical to fd2.
- We are left with the minimal cover of F (G_c):

  fd1: A->B　　fd2: B->C

  fd3: C->A

Examples (cont.)

- Consider the set of attributes {Student,Advisor,Subject,Grade} and a set of FDs F:

fd1:{Student,Advisor}->{Grade,Subject}

fd2: Advisor->Subject

fd3: {Student, Subject}->{Grade,Advisor}

Examples (cont.)

- Rewrite in standard form:

fd1a: {Student,Advisor}->Grade
fd1b: {Student,Advisor}->Subject
fd2: Advisor->Subject
fd3a: {Student,Subject}->Grade
fd3b: {{Student,Subject}->Advisor

Examples (cont.)

- Given fd2, Student is redundant in fd1b. We remove it. Now we remove fd1b since its identical to fd2.
- Next, fd1a is redundant because it's contained by the set {fd2, fd3a}. We remove it.

Examples (cont.)

- We are left with the minimal cover of F (G_c):
  fd2:Advisor->Subject
  fd3a: {Student,Subject}->Grade
  fd3b: {Student,Subject}->Advisor
- So the idea is to remove the fd's which are derivable from the others, and keep those fd's used in the process of derivation.