Functional dependencies (FDs)

- An FD is a constraint between two sets of attributes of a relation
- Given R, a set of attributes X in R is said to functionally determine another set of attributes Y in R (denoted $X \rightarrow Y$) iff
  - 2 tuples have the same values of X, then they must have the same values for attribute Y
- For the sake of simplicity we will write the set of attributes as lists:

$$A_1, ..., A_n \rightarrow B_1, ...B_m \tag{1}$$

Example relation

title	year	len	genre	studio	starName
Star Wars	1977	124	SciFi	Fox	Carry Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
The Godfather	1972	175	Drama	Paramount	Robert Duvall
The Godfather	1972	175	Drama	Paramount	Marlon Brando
Moonstruck	1987	102	Comedy	MGM	Cher

Superkey

- A set of attributes $\{A_1, ...A_n\}$ is a superkey of R iff $A_1, ...A_n \rightarrow R$

Candidate key

- A candidate key is a superkey that is minimal:
  - Given a candidate key $K = \{A_1, ..., A_n\}$ there is no proper subset C of K s.t. $C \rightarrow R$

Reasoning about FDs

Given a relation R two sets of FDs A and B are equivalent if:

- the set of instances of R that satisfy A is exactly the same that satisfy B

More generally: a set of FDs A follows from a set of FDs B if:

- every relation instance that satisfies all the FDs in B also satisfies all the FDs in A

Hence:

- Two sets of FDs A and B are equivalent iff A follows B and B follows A.

Armstrong's Axioms

Given relation R with subsets of attributes X, Y and Z:

- Reflexivity (Trivial):

$$Y \subseteq X \text{ then } X \rightarrow Y \tag{2}$$

- Augmentation:

$$\text{If } X \rightarrow Y \text{ then } XZ \rightarrow YZ \quad \text{for any Z} \tag{3}$$

- Transitivity:

$$\text{If } X \rightarrow Y, Y \rightarrow Z \text{ then } X \rightarrow Z \tag{4}$$

Closure of attributes

Given a relation R and a set of FDs, what other FDs can be computed from a set of FDs f?

- The closure of a set of attributes $A_1, ..., A_n$ denoted $\{A_1, ..., A_n\}^+$
- is the set of attributes that can be derived from $A_1, ..., A_n$ using f

Algorithm to compute the closure of attributes

To compute $\{A_1 ... A_n\}^+$:

1. Rewrite FDs in Canonical form
2. $X \leftarrow A_1 ... A_n$
3. For each $B_1 ... B_m \rightarrow C$ in FDs
   - if $B_1 ... B_m \subseteq X$ and $C \notin X$
     - add C to X
4. Repeat (3) until X does not change

At the end: $X = \{A_1 ... A_n\}^+$

Closure of sets of FDs

- do not confuse it with the closure of attributes!!!
- Given a set f of FDs, its closure f^+ is the set of all FDs derived from f
- Two sets of A and B of FDs of a relation R are equivalent iff $\{A\}^+ = \{B\}^+$

Testing if an FD in f is part of $\{f\}^+$

$$X \to Y \in f^+ \text{ iff } Y \subseteq \{X\}^+ \text{ (using f)} \tag{5}$$

Basis of a relation

Given a relation R and FDs f, we say that:

- any set g s.t. $f^+ = g^+$ is a basis of R

Minimal basis (3.2.7)

A minimal basis of a relation R is a basis of R s.t.:

1. All FDs in B are in Canonical Form
2. If, for any FD we remove one or more attr from the left hand side of an FD, the result is no longer a basis.
3. If any FD is removed from B, the result is no longer a basis

(note that the order in the book is different)

Projection of FDs (3.2.8)

Given R and set f of FDs, the projection of f, $R_1 = \Pi_L R$, is:

- the set of FDs that follows from f, and
- that involve only attributes in R_1

Algorithm to compute the projection of FDs

1. $T \leftarrow \emptyset$
2. For each subset $X \subset L$ compute X^+
   - for every attribute A in X^+
     - add $X \to A$ to T iff $A \in L$ and $A \notin X$

Example relation

title	year	len	genre	studio	starName
Star Wars	1977	124	SciFi	Fox	Carry Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
The Godfather	1972	175	Drama	Paramount	Robert Duvall
The Godfather	1972	175	Drama	Paramount	Marlon Brando
Moonstruck	1987	102	Comedy	MGM	Cher

Decomposition

Give $R(A_1, ...A_n)$ can be decomposed into $S(B_1, ..., B_m)$ and $T(C_1, ..., C_k)$ s.t.

1. $\{A_1, ...A_n\} = \{B_1, ..., B_m\} \cup \{C_1, ..., C_k\}$,
2. $S = \Pi_{B_1,...,B_k}(R)$, and
3. $T = \Pi_{C_1,...,C_k}(R)$

Decomposed relation

title	year	len	genre	studio	starName
Star Wars	1977	124	SciFi	Fox	Carry Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
The Godfather	1972	175	Drama	Paramount	Robert Duvall
The Godfather	1972	175	Drama	Paramount	Marlon Brando
Moonstruck	1987	102	Comedy	MGM	Cher

$\Pi_{title,year,len,genre,studio} Movies$

title	year	len	genre	studio
Star Wars	1977	124	SciFi	Fox
The Godfather	1972	175	Drama	Paramount
Moonstruck	1987	102	Comedy	MGM

$\Pi_{title,year,starName} Movies$

title	year	starName
Star Wars	1977	Carry Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
The Godfather	1972	Robert Duvall
The Godfather	1972	Marlon Brando
Moonstruck	1987	Cher

Good decomposition

Given R we want to decompose it into two relations S and T s.t.

1. Lossless join: $R = S \bowtie T$
2. Dependency preserving: The projection of FDs of R into S and T safisfies:

$$\{F_S \cup F_T\}^+ = \{F_R\}^+$$

BCNF (Boyce Codd Normal Form)

A relation R with set of FDs f is in BCNF iff

- for every non trivial FD $A_1, ..., A_n \to B_1, ...B_m$ in f the left-hand side is a superkey

Algorithm to Decompose into BCNF relations

Given R and set F of FDs. R is not BCNF

1. Choose one FD $X \rightarrow Y$ not in BCNF
2. Decompose
   - $R_1 = \{X\}^+$
   - $R_2 = X \cup (R - R_1)$
3. Compute FDs for R_1 and R_2
   - projection of FDs into R_1 and R_2
4. If R_1 or R_2 are not in BCNF, recursively decompose

3rd Normal Form (3NF)

A relation R with set of FDs f is in 3NF iff

- for every non trivial FD $A_1, ..., A_n \rightarrow B_1, ...B_m$ in f
  1. the left-hand side is a superkey, or
  2. every attribute $c \in \{B_1, ..., B_m\}$ is either
     a. $c \in \{A_1, ..., A_n\}$, or
     b. c is part of some candidate key

Decomposition of a Rel. into 3NF Relations (synthesis alg.)

Given R with set of FDs F

1. Find G, a minimal basis of F
2. For each FD $A_1 ... A_n \rightarrow B \in G$
   a. add a relation with schema $A_1 ... A_n B$
   b. with FD $A_1 ... A_n \rightarrow B$
3. If none of the added relations in step 2 is a Superkey or R
   - add another relation whose schema is a key of R