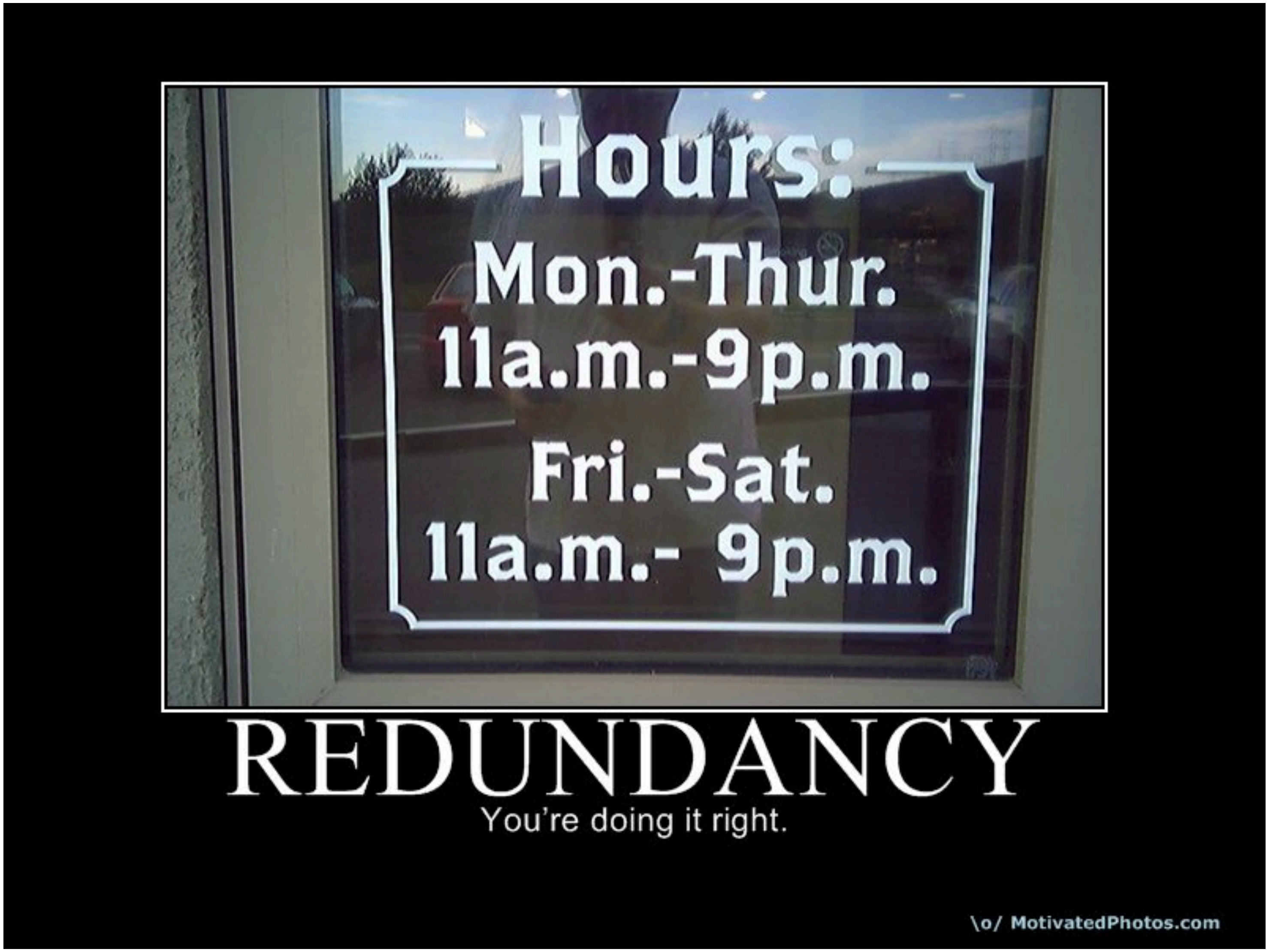
Functional Dependencies

Chapter 3


Hours:
Mon.-Thur.
11a.m.-9p.m.
Fri.-Sat.
11a.m.- 9p.m.
REDUNDANCY
You're doing it right.
\o/ MotivatedPhotos.com


LINDA L. FRALEY
2013
DO NOT TOP OFF
1 INSERT NOZZLE
2 TOPPING OFF AFTER AUTOMATIC SHUT-OFF MAY RESULT IN SPILLAGE AND MAY CAUSE A HAZARDOUS CONDITION.
Please Prepay in Advance
YOU COULD INSTANTLY WIN...
FREE GAS FOR A YEAR!
OTHER PRIZES
CircleKGreatLakes.com

*Often, our first attempts at DB schemas can be improved, especially by eliminating **redundancy**.*

Functional dependencies help us do this.

What is a FD?

- X -> Y
  - X and Y are sets of attributes from a relation.
  - Read: "X functionally determines Y"
- Intuitive definitions:
  - "If you know X, you can determine Y."
  - "For each X, there can be only one Y."


HIGHLANDER™
THERE CAN BE ONLY ONE

What is a FD?

- X -> Y means

  If two tuples agree on all attributes in X, then they must agree on all attributes in Y.

- If we can be sure every instance of a relation will make a FD true, then the relation ***satisfies*** the FD.

What is an FD?

- An FD is a constraint on a single relational schema (one table).
  - It must hold on every instance of the relation.
  - Therefore, you cannot deduce an FD from a relational instance.

title	year	length	genre	studio	star
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	Drama	MGM	Vivien Leigh
Wayne's World	1992	95	Comedy	Paramount	Dana Carvey
Wayne's World	1992	95	Comedy	Paramount	Mike Meyers

What are the FDs?

title	year	length	genre	studio	star
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	Drama	MGM	Vivien Leigh
Wayne's World	1992	95	Comedy	Paramount	Dana Carvey
Wayne's World	1992	95	Comedy	Paramount	Mike Meyers

title year -> length

title year -> genre

title year -> studio

title year -> length genre studio

title	year	length	genre	studio	star
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	Drama	MGM	Vivien Leigh
Wayne's World	1992	95	Comedy	Paramount	Dana Carvey
Wayne's World	1992	95	Comedy	Paramount	Mike Meyers

Does title year -> star?

Where do FDs come from?

- "Key-ness" of attributes
- Domain and application constraints
- Real world constraints

Definition of Keys

- FDs allow us to formally define keys
- A set of attributes {A1, A2, …, An} is a ***key*** for relation R if:

Uniqueness: {A1, A2, …, An} functionally determine all the other attributes of R

Minimality: no proper subset of {A1, A2, …, An} functionally determines all other attributes of R.

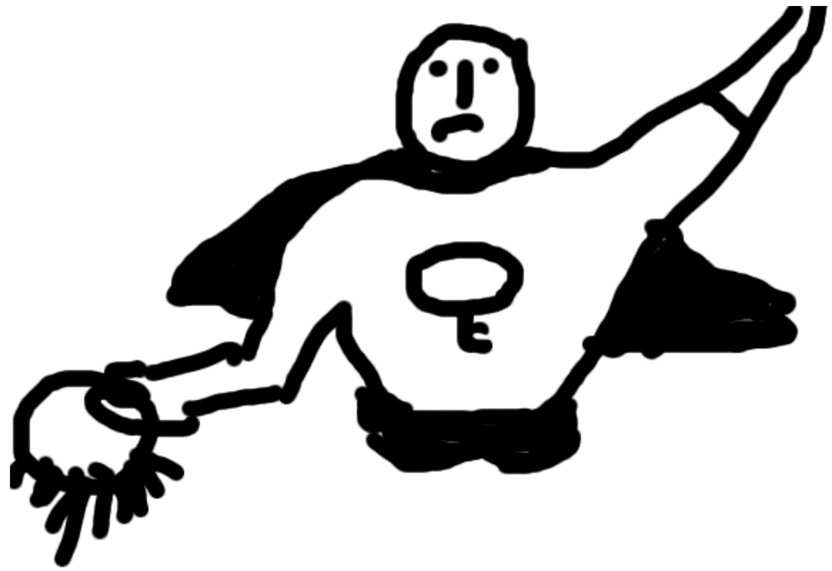
title	year	length	genre	studio	star
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	Drama	MGM	Vivien Leigh
Wayne's World	1992	95	Comedy	Paramount	Dana Carvey
Wayne's World	1992	95	Comedy	Paramount	Mike Meyers

- What are the keys?

Two things you already know and one thing you don't:

- A relation can have more than one key.
- Usually one key is known as the primary key.
- FDs have nothing to do with primary keys, just keys.

Superkeys



Superkeys

- A ***superkey*** (superset of a key) is a set of attributes that contains a key.

- In other words, a superkey satisfies the uniqueness part of the key definition, but may not satisfy the minimality part.

Find the keys and superkeys

title	year	length	genre	studio	star
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	Drama	MGM	Vivien Leigh
Wayne's World	1992	95	Comedy	Paramount	Dana Carvey
Wayne's World	1992	95	Comedy	Paramount	Mike Meyers

With a partner

- Consider a relation about people in the USA, including name, SSN, street address, city, state, zip code, area code, and 7-digit phone number.
- What FDs would you expect to hold?
- What are the keys for this relation?
- Hints: Can an area code straddle two states? Can a zip code straddle two area codes?

Rules for Manipulating FDs

- Learn how to reason about FDs
- Define rules for deriving new FDs from a given set of FDs
- Example: R (A, B, C) satisfies FDs A->B, B->C.
  - What others does it satisfy?
  - A -> C
  - What is the key for R?
  - A (because A->B and A->C)

Equivalence of FDs

- Why?
  – To derive new FDs from a set of FDs
- An FD F follows from a set of FDs T if every relation instance that satisfies all the FDs in T also satisfies F
  – A → C follows from T = {A→B, B→C}
- Two sets of FDs S and T are equivalent if each FD in S follows from T and each FD in T follows from S
  – S = {A→B, B→C, A→C} and T = {A→B, B→C} are equivalent

Splitting and Combining FDs

- The set of FDs
  - A1 A2 A3…An → B1
  - A1 A2 A3…An → B2
  - …

  is equivalent to the FD
  - A1 A2 A3…An → B1 B2 B3 … Bm
- This equivalence implies two rules:
  - Splitting rule
  - Combining rule
  - These rules work because all the FDs in S and T have identical left hand sides

Splitting and Combining FDs

- Can we split and combine left hand sides of FDs?
- For the Movies relation, is the FD
  - title year -> length

  equivalent to the set of FDs {title -> length, year -> length}?
  - (No!)

Triviality of FDs

- A FD A1 A2…An → B1 B2…Bm is
  - Trivial if the B's are a subset of the A's

    $$\{B_1, B_2, \ldots B_n\} \subseteq \{A_1, A_2, \ldots A_n\}$$

  - Non-trivial if at least one B is not among the A's

    $$\{B_1, B_2, \ldots B_n\} - \{A_1, A_2, \ldots A_n\} \neq \emptyset$$

  - Completely non-trivial if none of the B's are among the A's

    $$\{B_1, B_2, \ldots B_n\} \cap \{A_1, A_2, \ldots A_n\} = \emptyset$$

Triviality of FDs

- What good are trivial and non-trivial FDs?
  - Trivial dependencies are always true
  - They help simplify reasoning about FDs
- Trivial dependency rule: The FD A1 A2…An → B1 B2…Bm is equivalent to the FD A1 A2…An → C1 C2..Ck, where the C's are those B's that are not A's, i.e.

$$\{C_1, C_2, \ldots, C_k\} = \{B_1, B_2, \ldots, B_m\} - \{A_1, A_2, \ldots, A_n\}$$

- Find a trivial FD:

title	year	length	genre	studio	star
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	Drama	MGM	Vivien Leigh
Wayne's World	1992	95	Comedy	Paramount	Dana Carvey
Wayne's World	1992	95	Comedy	Paramount	Mike Meyers

Review

- **FD: X -> Y**: for each X, there is only one Y.
- **Superkey**: a set of attributes that functionally determines all of the other attributes of a relation.
- **Key**: a superkey that is also minimal (can't remove any attributes from it and still functionally determine all the other attributes).
- **Trivial FD**: where X and Y have an attribute in common.

Review

- A set of FDs S **follows** from another set of FDs T iff all the FDs in S are implied by those in T.
  - (e.g., through the splitting/combining rule, transitivity, etc)
- Two sets of FDs are **equivalent** if each set follows from the other.

Closure of a set of attributes

- Suppose you have a set of attributes {A1, …, An} and a set of FDs S.
- The closure of {A1, …, An} under S is the set of attributes B such that
  – every relation in S also satisfies A1…An -> B.
- Intuitive def'n: B is the set of attributes that we can deduce from knowing A1, …, An.
- Closure of {A1,…An} denoted by $\{A1,\ldots An\}^+$

Closure of Attributes: Algorithm

1. Use the splitting rule so that each FD in S has one attribute on the right.
2. Set X = {A1, A2 …, An}
3. Find FD B1 B2…Bk → C in S such that

{B1 B2 … Bk} $\subseteq$ X but C $\notin$ X

4. Add C to X
5. Repeat the last two steps until you can't find C

Why is the algorithm correct?
Read 3.2.5 in textbook

Closure of Attributes: Example

- Suppose a relation R (A, B, C, D, E, F) has FDs:
  - AB → C, BC → AD, D → E, CF → B
- Find the closures of:
  - {A, B}
  - {B, C, F}
  - {A, F}

  under the FDs above.

Why compute closures?

- Can test whether any FD follows from a set of other FDs.
  - Say we know a set of FDs S, and we want to check if a "new" FD A1…An -> B follows from S.
  - Simply check if B is in {A1, A2, …, An} closed under S.
- To prove the correctness of rules for manipulating FDs.
- Can compute keys algorithmically.

Algorithm for computing keys

- Recall a superkey is a set of attributes that functionally determines all the other attributes.
- The closure of a set of attributes A1…An under a set of FDs gives you all the other attributes in R that can be functionally determined from knowing A1…An.
- What is the connection between superkeys and attribute closure?

Students and Profs

- Suppose we have one single relation with attributes:
  - R#
  - Student Name
  - ProfID (ID of professor teaching a class with the student)
  - ProfName
  - AdvisorID
  - AdvisorName

Armstrong’s Axioms

- We can use closures of attributes to determine if any FD follows from a given set of FDs
- Armstrong's axioms: complete set of inference rules from which it is possible to derive every FD that follows from a given set.

Not the right W. W. Armstrong. This is Warwick Windridge Armstrong, an Australian cricketer. He did not invent these axioms. They were originated by William Ward Armstrong, who is Canadian, and does not have a picture on Wikipedia.

Armstrong's Axioms

X, Y, and Z are sets of attributes.

- **Reflexivity**

$$Y \subseteq X \Rightarrow X \rightarrow Y$$

  – E.g. ssn name → ssn

- **Augmentation**

$$X \rightarrow Y \Rightarrow XW \rightarrow YW$$

  – E.g. ssn → name then ssn grade → name grade

Armstrong's Axioms

X, Y, and Z are sets of attributes.

- **Transitivity**

$$\left.\begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \end{array}\right\} \Rightarrow X \rightarrow Z$$

e.g. if ssn → address and address → tax-rate
then

ssn → address

Note on notation

- Relation Schema: R(A1, A2, A3): parentheses surround attributes, attributes separated by commas.
- Set of attributes: {A1, A2, A3}: curly braces surround attributes, attributes separated by commas
- FD: A1 A2 → A3: no parentheses or curly braces, attributes separated by spaces, arrows separates left hand side and right hand side
- Set of FDs: {A1 A2 → A3, A2 → A1}: curly braces surround FDs, FDs separated by commas

Computing Closures of FDs

- To compute the closure of a set of FDs, repeatedly apply Armstrong's Axioms until you cannot find any new FDs.

Examples of Computing Closures of FDs

- (Let us include only completely non-trivial FDs in these examples, with a single attribute on the right)
- $F = \{A \rightarrow B, B \rightarrow C\}$
- $\{F\}^+ = ??$

Examples of Computing Closures of FDs

- (Let us include only completely non-trivial FDs in these examples, with a single attribute on the right)
- F = {A→B, B→C}
- $\{F\}^+$ = {A→B, B→C, A→C, AC→B, AB→C}

Examples of Computing Closures of FDs

- (Let us include only completely non-trivial FDs in these examples, with a single attribute on the right)
- F = {AB→C, BC→A, AC→B}
- $\{F\}^+$ = ??

Examples of Computing Closures of FDs

- (Let us include only completely non-trivial FDs in these examples, with a single attribute on the right)
- $F = \{AB \rightarrow C, BC \rightarrow A, AC \rightarrow B\}$
- $\{F\}^+ = \{AB \rightarrow C, BC \rightarrow A, AC \rightarrow B\}$

Examples of Computing Closures of FDs

- (Let us include only completely non-trivial FDs in these examples, with a single attribute on the right)
- $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$
- $\{F\}^+ = ??$

Examples of Computing Closures of FDs

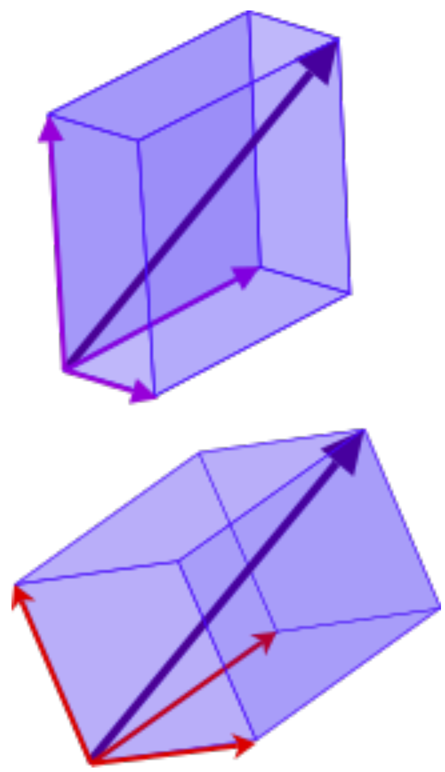
- (Let us include only completely non-trivial FDs in these examples, with a single attribute on the right)
- $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$
- $\{F\}^+ = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, A \rightarrow C, A \rightarrow D, B \rightarrow D, \ldots\}$

Closures of Attributes vs Closure of FDs

- Closure of attributes:
  - Takes a set of attributes A and a set of FDs S.
  - Produces a set of attributes (all the attribs that can be functionally determined from A, given S).
  - Used for computing keys, checking if an FD follows from a set of FDs.
- Closure of a set of FDs:
  - Takes a set of FDs.
  - Produces a set of FDs (all the FDs that follow from S).
  - Can be for verifying a minimal basis, but can use closure of attributes as well.

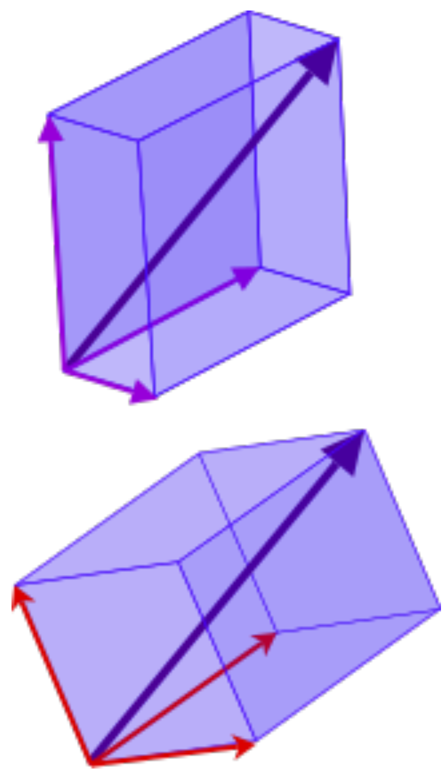
Basis Set of FDs

- In linear algebra, a ***basis*** is the smallest set of linearly independent vectors such that you can build any other vector out of the basis vectors.



Basis Set of FDs

- In databases, a ***(minimal) basis*** for a set of FDs S is the smallest set of FDs that is equivalent to S.
  - That is, all of the FDs in S follow from the basis set of FDs.



Minimal basis

- Given a set of FDs S, a minimal basis for S is another set of FDs B where:
  - All the FDs in B have singleton right sides.
  - If any FD is removed from B, the result is no longer a basis.
  - If we remove any attribute from the left side of any FD in B, the result is no longer a basis.
- Like in linear algebra, there can be multiple minimal bases for a set of FDs, though unlike in linear algebra, two minimal bases for a set of FDs may be different sizes.

Example of Minimal Basis

- R(A, B, C) is a relation such that each attribute functionally determines the other two attributes
- What are the FDs that hold in R and what are the minimal bases?
  - (Assume only one attribute on the right-hand side, only non-trivial FDs)

Example of Minimal Basis

- R(A, B, C) is a relation such that each attribute functionally determines the other two attributes
- What are the FDs that hold in R and what are the minimal bases?
  - (Assume only one attribute on the right-hand side, only non-trivial FDs)
- FDs: A→B, A→C, B→A, B→C, C→A, C→B, AB→C, BC→A, AC→B
- Minimal Bases: {A→B, B→A, B→C, C→B}, {A→B, B→C, C→A}, etc.