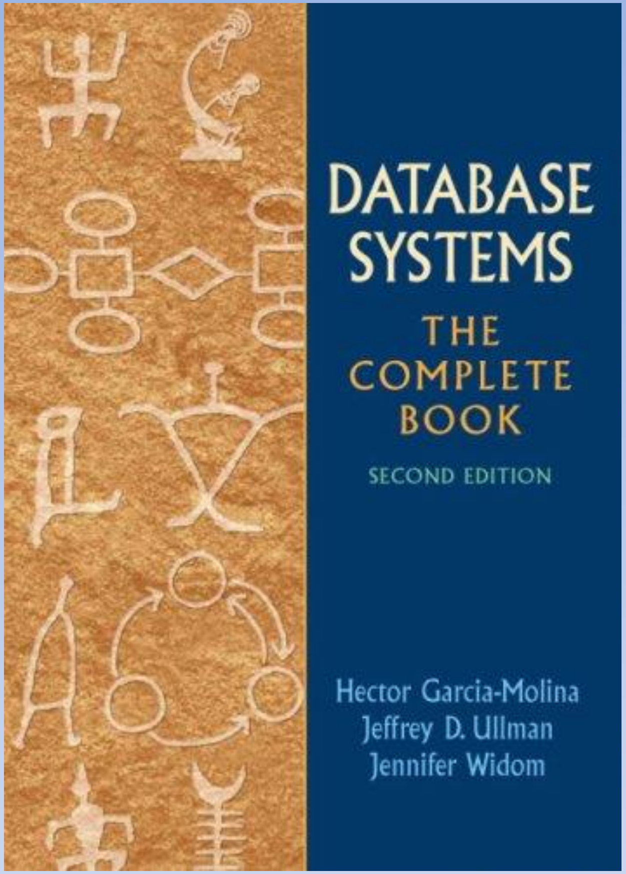
Database Systems: The Complete Book(3rd)

by Hector Garcia-Molina,

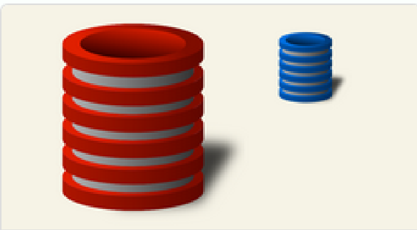
Jeffrey D. Ullman & Jennifer Widom



Design Theory

- Chapter 3: Design Theory for Relational Databases
  - Normal Forms

Next Assignment



Relational Design Theory

Databases - DB8

Started - Jun 09, 2014

View Course

Your final grade: **92%**.

Download Statement (PDF)

What Makes a Good Database?

What Makes a Good Database?

- What makes a **BAD** database?

DATABASE ANOMALIES

Database Anomalies

- What is a DATABASE ANOMALY?
- What is Database?
  - Collection of information that exists over some extended period of time.
  - Database is a collection of data managed by a Database Management System.

- What is Anomaly?

www.merriam-webster.com

Anomaly

- 1: the angular distance of a planet from its perihelion as seen from the sun

Anomaly

- 1: the angular distance of a planet from its perihelion as seen from the sun
- 2: deviation from the common rule : irregularity
- 3: something anomalous : something different, abnormal, peculiar, or not easily classified

Database Anomalies: Redundancy

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	Ain't Wastin' Time No More	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	Melissa	3:51	Capricorn
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

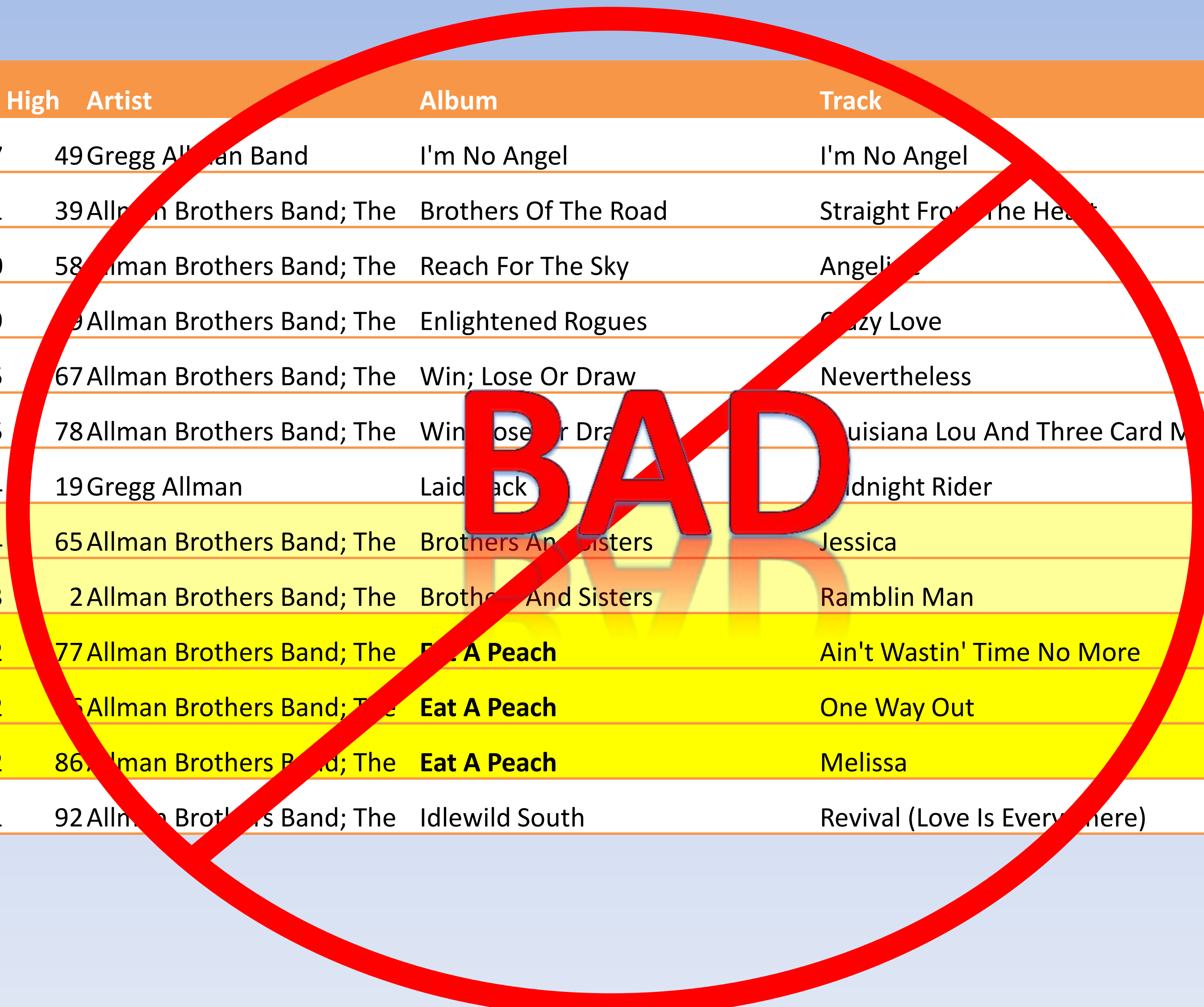
Database Anomalies: Redundancy

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win ose r Dra	uisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid ack	idnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	Ain't Wastin' Time No More	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	Melissa	3:51	Capricorn
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

BAD

Database Anomalies: Redundancy

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	[illegible]	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	Ain't Wastin' Time No More	3:40	Capricorn
1972	[illegible]	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	Melissa	3:51	Capricorn
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn



Database Anomalies: Redundancy

- Information repeated in multiple tuples
  - Wasteful
  - Introduces inconsistency potentials!

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	Ain't Wastin' Time No More	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	Melissa	3:51	Capricorn
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

Database Anomalies: Update Anomaly

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	Ain't Wastin' Time No More	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	Melissa	3:51	**Epic**
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

Database Anomalies: Update Anomaly

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	am[illegible]n Ma[illegible]	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	[illegible]in't [illegible] No [illegible]ore	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	Melissa	3:51	**Epic**
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

BAD

Database Anomalies: Update Anomaly

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy L[illegible]	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	[illegible]:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:2[illegible]	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	[illegible]	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	[illegible]	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	[illegible]elissa	3:5[illegible]	**Epic**
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	[illegible]39	Capricorn

BAD

Database Anomalies: Deletion Anomaly

AYear	High	Artist	Album	Track	ATime	Label
~~1987~~	~~49~~	~~Gregg Allman Band~~	~~I'm No Angel~~	~~I'm No Angel~~	~~3:43~~	~~Epic~~
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	Eat A Peach	Ain't Wastin' Time No More	3:40	Capricorn
1972	86	Allman Brothers Band; The	Eat A Peach	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	Eat A Peach	Melissa	3:51	Capricorn
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

Database Anomalies: Deletion Anomaly

- Deleting Track Deletes:
  - Entire Album
  - Entire Label

AYear	High	Artist	Album	Track	ATime	Label
~~1987~~	~~49~~	~~Gregg Allman Band~~	~~I'm No Angel~~	~~I'm No Angel~~	~~3:43~~	~~Epic~~
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	Eat A Peach	Ain't Wastin' Time No More	3:40	Capricorn
1972	86	Allman Brothers Band; The	Eat A Peach	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	Eat A Peach	Melissa	3:51	Capricorn
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

Database Anomalies
THE PROBLEM

- Redundancy
- Update Anomalies
- Deletion Anomalies

Database Anomalies
THE SOLUTION

- Redundancy
- Update Anomalies
- Deletion Anomalies

NORMAL FORMS

Normal Forms

- First some theory (definitions)….

Functional Dependencies

- Captures Constraint:
  - Whenever a value of one attribute (or set) occurs
  - the value of another attribute (or set) is always the same.
- Given attribute **A** with a value **a**, the attribute **B** functionally determined by A will always have the same value, say **b**

Database Anomalies: Functional Dependency

AYear	High	Artist	Album	Track	ATime	Label
1987	49	Gregg Allman Band	I'm No Angel	I'm No Angel	3:43	Epic
1981	39	Allman Brothers Band; The	Brothers Of The Road	Straight From The Heart	3:18	Arista
1980	58	Allman Brothers Band; The	Reach For The Sky	Angeline	3:15	Arista
1979	29	Allman Brothers Band; The	Enlightened Rogues	Crazy Love	3:07	Capricorn
1975	67	Allman Brothers Band; The	Win; Lose Or Draw	Nevertheless	3:34	Capricorn
1975	78	Allman Brothers Band; The	Win; Lose Or Draw	Louisiana Lou And Three Card M	3:10	Capricorn
1974	19	Gregg Allman	Laid Back	Midnight Rider	4:26	Capricorn
1974	65	Allman Brothers Band; The	Brothers And Sisters	Jessica	4:00	Capricorn
1973	2	Allman Brothers Band; The	Brothers And Sisters	Ramblin Man	4:58	Capricorn
1972	77	Allman Brothers Band; The	**Eat A Peach**	Ain't Wastin' Time No More	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	One Way Out	3:40	Capricorn
1972	86	Allman Brothers Band; The	**Eat A Peach**	Melissa	3:51	Capricorn
1971	92	Allman Brothers Band; The	Idlewild South	Revival (Love Is Everywhere)	2:39	Capricorn

Album
Functionally Determines
Label

Functional Dependency

- *Album* -> Label
  - IF: two tuples agree in their ALBUM value
    - (the song is on the same album)
  - THEN: the two tuples will agree in their LABEL value.
- $A \rightarrow B$
  - *IF: two tuples agree in their A values.*
  - *THEN: the two tuples agree in their B values.*

Functional Dependency

- Based on information taken from the real world.
- Must be true for all possible tuples in the relation.

Reasoning w/ FD's

- IF:
  - ❖ $A \rightarrow B$ & $B \rightarrow C$
  - ❖ *We can Prove:* $A \rightarrow C$
- *Proof By Construction*
- *Any 2 tuples with same value for A:*
  - (a, b_1, c_1) & (a, b_2, c_2)
  - *Since* $A \rightarrow B$: (a, b, c_1) & (a, b, c_2)
  - *Since* $B \rightarrow C$: c_1 *must equal* c_2
  - *QED*

Keys & FDs

- Now more formally define a KEY for a Relation.

- $\{A_1, A_2, \ldots, A_n\}$ is a set of attributes
- $\{A_1, A_2, \ldots, A_n\}$ is a KEY for relation R IF:
  - $\{A_1, A_2, \ldots, A_n\}$ Functionally Determine all other attributes of R.
  - No proper subset of $\{A_1, A_2, \ldots, A_n\}$ also Functionally Determine all other attributes of R.

Computers Domain

- Product (maker, model, type)

- Assert FDs:
  - model $\rightarrow$ maker
  - model $\rightarrow$ type
- {model} is a key

Superkeys

- A set that contains a KEY as a subset is called a Superkey!

Computers Domain

- Product (maker, model, type)

- Assert FDs:
  - o model $\rightarrow$ maker
  - o model $\rightarrow$ type
- {model} is a key
- {model, maker} is a superkey
- {model, maker} IS NOT a key.
  - – Not minimal

Question

1. Consider a relation *R*(A,B,C) and suppose *R* contains the following four tuples:

A	B	C
1	2	2
1	3	2
1	4	2
2	5	2

For each of the following functional dependencies, state whether or not the dependency is satisfied by this relation instance.

(a) $A \rightarrow B$
(b) $A \rightarrow C$
(c) $B \rightarrow A$
(d) $B \rightarrow C$
(e) $C \rightarrow A$
(f) $C \rightarrow B$
(g) $AB \rightarrow C$
(h) $AC \rightarrow B$
(i) $BC \rightarrow A$

Question

1. Consider a relation R(A,B,C) and suppose R contains the following four tuples:

A	**B**	**C**
1	2	2
1	3	2
1	4	2
2	5	2

For each of the following functional d

ncy is satisfied by this relation instance.

(a) $A \rightarrow B$
(b) $A \rightarrow C$
(c) $B \rightarrow A$
(d) $B \rightarrow C$
(e) $C \rightarrow A$
(f) $C \rightarrow B$
(g) $AB \rightarrow C$
(h) $AC \rightarrow B$
(i) $BC \rightarrow A$

1.
(a) not satisfied
(b) satisfied
(c) satisfied
(d) satisfied
(e) not satisfied
(f) not satisfied
(g) satisfied
(h) not satisfied
(i) satisfied

Reasoning w/ FD's

- Usually lean towards constructions

- Two sets of FD's *S* and *T* are equivalent IF:
  - The set of relation instances satisfying S is exactly the same as the set of relation instances satisfying T.

- *S* FOLLOWS from *T* IF:
  - ALL **relation instances** that satisfy *T* also satisfy *S*.

Splitting/Combining FD's

- SPLITTING
- $A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_n$

- *Set of Rules:*
  - $A_1, A_2, \ldots, A_n \rightarrow B_1$
  - $A_1, A_2, \ldots, A_n \rightarrow B_2$

  …
  - $A_1, A_2, \ldots, A_n \rightarrow B_n$
- Not left hand side!

Splitting/Combining FD's

- PersonID, StartDate $\rightarrow$ *PhoneNumber*
  - *?? PersonID* $\rightarrow$ *PhoneNumber*
  - *??* StartDate $\rightarrow$ *PhoneNumber*

- Not left hand side!

Splitting/Combining FD's

- COMBINING
- *Set of Rules:*
  - $A_1, A_2, \ldots, A_n \rightarrow B_1$
  - $A_1, A_2, \ldots, A_n \rightarrow B_2$

  …
  - $A_1, A_2, \ldots, A_n \rightarrow B_n$

- $A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_n$

Simplifying FD's w/ Trivial-Dependency Rule

- A trivial FD is true for every possible relation instance.
  - Any constraint is trivial if it holds for all possible relation instances.
- $A_1, A_2, \ldots, A_n \rightarrow A_1$
  - Provably trivial since any two tuples $(a_1, a_2, \ldots a_n)$ & $(b_1, b_2, \ldots, b_n)$ that agree on $a_1, a_2, \ldots, a_n$, MUST agree on a_1.
- By same token in a Non-Trivial FD:
  - If some attributes appear on both left and right side of FD, they can be removed from right side.

Question

2. Which of the following rules for functional dependencies are correct (i.e., the rule holds over all databases) and which are incorrect (i.e., the rule does not hold over some database)? For incorrect rules, give the simplest example relation instance you can come up with where the rule does not hold.

(a) If $A \rightarrow B$ and $BC \rightarrow D$, then $AC \rightarrow D$
(b) If $AB \rightarrow C$ then $A \rightarrow C$
(c) If $A \rightarrow B1,..,Bn$ and $C1,..,Cm \rightarrow D$ and $\{C1,...,Cm\}$ is a subset of $\{B1,..,Bn\}$, then $A \rightarrow D$
(d) If $A \rightarrow C$ and $B \rightarrow C$ and $ABC \rightarrow D$, then $A \rightarrow D$

Question

2. Which of the following rules for functional dependencies are correct (i.e., the rule holds over all databases) and which are incorrect (i.e., the rule does not hold over some database)? For incorrect rules, give the simplest example relation instance you can come up with where the rule does not hold.

(a) If $A \rightarrow B$ and $BC \rightarrow D$, then $AC \rightarrow D$
(b) If $AB \rightarrow C$ then $A \rightarrow C$
(c) If $A \rightarrow B1,..,Bn$ and $C1,..,Cm \rightarrow D$ and $\{C1,...,Cm\}$ is a subset of $\{B1,..,Bn\}$, then $A \rightarrow D$
(d) If $A \rightarrow C$ and $B \rightarrow C$ and $ABC \rightarrow D$, then $A \rightarrow D$

2.
(a) is correct
(b) is incorrect - R(A,B,C) with R={(1,2,3),(1,4,5)}
(c) is correct
(d) is incorrect - R(A,B,C,D) with R={(1,2,3,4),(1,5,3,6)}

Question

3. Consider a relation R(A,B,C,D,E) with the following functional dependencies:

$A \rightarrow B$
$CD \rightarrow E$
$E \rightarrow A$
$B \rightarrow D$

Specify all minimal keys for R.

Question

3. Consider a relation *R*(A,B,C,D,E) with the following functional dependencies:

A → B
CD → E
E → A
B → D

Specify all minimal keys for *R*.

- AC, BC, CD, CE

Question

4. Consider a relation R(A,B,C,D,E,F,G,H) with the following functional dependencies:

$A \rightarrow BCD$
$AD \rightarrow E$
$EFG \rightarrow H$
$F \rightarrow GH$

(a) Based on these functional dependencies, there is one minimal key for R. What is it?
(b) One of the four functional dependencies can be removed without altering the key. Which one?

Question

4. Consider a relation R(A,B,C,D,E,F,G,H) with the following functional dependencies:

A → BCD
AD → E
EFG → H
F → GH

(a) Based on these functional dependencies, there is one minimal key for R. What is it?
(b) One of the four functional dependencies can be removed without altering the key. Which one?

- (a) AF
- (b) EFG -> H
  - F -> GH

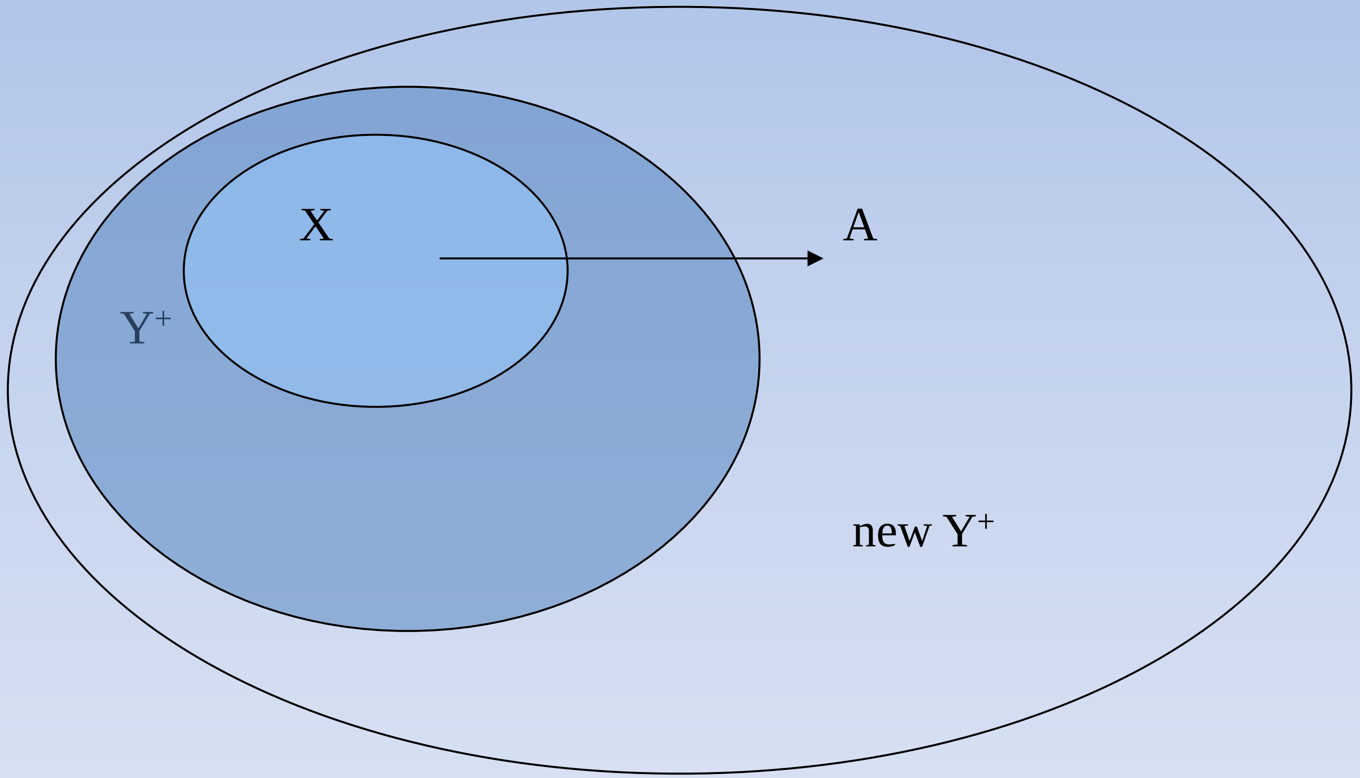
Functional Dependency & Closure

- Given:
  - Set of attributes Ā , Relation, & FD’s
  - Find the set of all attributes B, functionally determined by Ā.
    - Find all B, such that $\bar{A} \to B$
- Closure of Ā designated as $\bar{A}^{+}$

Algorithm 3.7

1. Insure that FD's have singleton RHS (split if needed).
2. Initialize X (eventual output) to Set of attributes $\bar{A}$
3. Repeatedly search for through set of FD's for FD:
   - ❖ $A_1, A_2, \ldots, A_n \rightarrow B_1$
   - ❖ *LHS* ($A_1, A_2, \ldots, A_n$) are all in X
   - ❖ B is not yet in X
   - ❖ Add B to X
4. X is closure set when no more attributes can be added.

Closure Algorithm



Computing Closure w/ Python
Representing FD's

- FD Representation
  - ('AB', 'C')
  - A, B $\rightarrow C$
- Set of FD's is a Python List of FD's
  - [FD1, FD2, FD3]
    - FD1 = ('AB', 'C')

Computing Closure w/ Python

```
A='ABCD'
FDs=[('AB','C'), ('C','D'), ('D','A')]

def ComputeClosure(A, S):
    '''A: Set of attributes 'ABCDE'
       S: Set of FD's [ ['AB', 'C'], ... ]
    '''
    if not A: return []
    X = list(A)
    last = 0
    while last != len(X):
        last = len(X)
        for s in S:
            if s[1] not in X and all([a in X for a in s[0]]):
                X.append(s[1])
    return sorted(X)
```

Computing Closure w/ Python

```
A='ABCD'
FDs=[('AB','C'), ('C','D'), ('D','A')]

def AC():
    nFD = []
    for a in A:
        cc = ComputeClosure(a, FDs)
        print a, '=>', cc
        for z in cc:
            if not checkFDs( (a,z), FDs ): nFD.append( (a,z) )
    print 'New FDs', nFD

    for a in [a+b for a in A for b in A if a<b]:
        if a< b:
            cc = ComputeClosure(a, FDs)
            print a, '=>', cc
        for z in cc:
            if not checkFDs( (a,z), FDs ): nFD.append( (a,z) )
    print 'New FDs', nFD
```

A=

FD

de

```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
>>> AC()
A => ['A']
B => ['B']
C => ['A', 'C', 'D']
D => ['A', 'D']
New FDs [('C', 'A')]
AB => ['A', 'B', 'C', 'D']
AC => ['A', 'C', 'D']
AD => ['A', 'D']
BC => ['A', 'B', 'C', 'D']
BD => ['A', 'B', 'C', 'D']
CD => ['A', 'C', 'D']
New FDs [('C', 'A'), ('AB', 'D'), ('AC', 'D'), ('BC', 'A'), ('BC', 'D'), ('BD',
'A'), ('BD', 'C'), ('CD', 'A')]
>>> 
```

FD's from Closure Set

- Closure of $\bar{A}$ = $\bar{A}^+ = (A_1, A_2, \ldots, A_n)^+$ implies FD's
  - $A_1, A_2, \ldots, A_n \rightarrow B$
  - For all B in $\bar{A}^+$

Closure Algorithm Test Proof (part 1)

- Closure Algorithm claims only true FDs.
- Basis: *After 0 steps true for trivial FD*
  - $A_1, A_2, \ldots, A_n \rightarrow A_1, A_2, \ldots, A_n$
- Induction: Suppose D added from
  - FD F = $B_1, B_2, \ldots, B_n \rightarrow D$
  - Inductive Hypothesis (IH): $A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_n$
  - If two tuples agree on $A_1, A_2, \ldots, A_n$ then from IH they also agree on $B_1, B_2, \ldots, B_n$
  - Since the tuples agree on $B_1, B_2, \ldots, B_n$ then from FD F they agree on D proving $A_1, A_2, \ldots, A_n \rightarrow D$

Closure Test Proof (part 2)

- Need to prove that if a Functional Dependency is not claimed from Closure Algorithm, then it does not follow from input FDs.
- Proof by construction of two tuples that agree in $\bar{A}^+$ but disagree in all other attributes.

WHY
Closure Algorithm Discovers ALL True FD's

- Suppose $A_1, A_2, \ldots, A_n \rightarrow B$ were an FD not found by algorithm 3.7!
- This Means:
  - $(A_1, A_2, \ldots, A_n)^+$ using the set of FD's does not include B.
- Our proof requires showing :
  - $A_1, A_2, \ldots, A_n \rightarrow B$
  - DOES NOT FOLLOW from FD's

WHY
Closure Algorithm Discovers ALL True FD's

- How are we going to show:
  - $A_1, A_2, \ldots, A_n \rightarrow B$
  - DOES NOT FOLLOW from FD's

- Construct an Instance I of the relation R that satisfies all FD's BUT does not satisfy the FD:
  - $A_1, A_2, \ldots, A_n \rightarrow B$

WHY
Closure Algorithm Discovers ALL True FD's

- Construct an Instance I of the relation R that satisfies all FD's BUT does not satisfy the FD:
  - $A_1, A_2, \ldots, A_n \rightarrow B$

	$\{A_1, A_2, \ldots, A_n\}^+$	Other Attributes
t:	$1\,1\,1 \cdots 1\,1$	$0\,0\,0 \cdots 0\,0$
s:	$1\,1\,1 \cdots 1\,1$	$1\,1\,1 \cdots 1\,1$

- I Consists of Two Tuples: t, s
- t & s agree in $(A_1, A_2, \ldots, A_n)^+$
- t & s disagree in all attributes NOT IN $(A_1, A_2, \ldots, A_n)^+$

Maybe there's a FD in our set that is not satisfied by t & s???

- $C_1, C_2, \ldots, C_k \rightarrow D$
  - Not Satisfied by t, s
  - t & s agree in $C_1, C_2, \ldots, C_k$
  - t & s DO NOT agree in D!

	$\{A_1, A_2, \ldots, A_n\}^+$	Other Attributes
t:	$1\,1\,1 \cdots 1\,1$	$0\,0\,0 \cdots 0\,0$
s:	$1\,1\,1 \cdots 1\,1$	$1\,1\,1 \cdots 1\,1$

- $C_1, C_2, \ldots, C_k$ must be in $(A_1, A_2, \ldots, A_n)^+$
- D must NOT be in $(A_1, A_2, \ldots, A_n)^+$
  - t & s agree in $C_1, C_2, \ldots, C_k$
  - t & s DO NOT agree in D!

Maybe there's a FD in our set that is not satisfied by t & s???

	$\{A_1, A_2, \ldots, A_n\}^+$	Other Attributes
t:	$1\,1\,1 \cdots 1\,1$	$0\,0\,0 \cdots 0\,0$
s:	$1\,1\,1 \cdots 1\,1$	$1\,1\,1 \cdots 1\,1$

- $C_1, C_2, \ldots, C_k$ must be in $(A_1, A_2, \ldots, A_n)^+$
- D must NOT be in $(A_1, A_2, \ldots, A_n)^+$
  - t & s agree in $C_1, C_2, \ldots, C_k$
  - t & s DO NOT agree in D!
- Closure Algorithm ERROR/Contradiction!
  - $C_1, C_2, \ldots, C_k \rightarrow D$ Should have caused D to be added to $(A_1, A_2, \ldots, A_n)^+$ when X was equal to $A_1, A_2, \ldots, A_n$
- $C_1, C_2, \ldots, C_k \rightarrow D$ CANNOT EXIST!

NOW SHOW:
I does not Satisfy $A_1, A_2, \ldots, A_n \rightarrow B$

	$\{A_1, A_2, \ldots, A_n\}^+$	Other Attributes
t:	$1\,1\,1 \cdots 1\,1$	$0\,0\,0 \cdots 0\,0$
s:	$1\,1\,1 \cdots 1\,1$	$1\,1\,1 \cdots 1\,1$

- $A_1, A_2, \ldots, A_n$ are part of $(A_1, A_2, \ldots, A_n)^+$
- B is NOT part of $(A_1, A_2, \ldots, A_n)^+$
- t & s therefore disagree in B!
- THEREFORE, Instance I (t & S) does not satisfy $A_1, A_2, \ldots, A_n \rightarrow B$

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- What are all non-trivial FD's that follow from the given FD's (single right side)?
- What are all the keys?
- What are all the superkeys (that are not also keys)?

Functional Dependency & Keys

- Assume a relation has no duplicate tuples.
  - IF: $\bar{A} \rightarrow$ all attributes of relation
  - THEN: $\bar{A}$ is a KEY for the relation
    - $\bar{A}$ must have a different value for every tuple.
    - $\bar{A}$ must not be null.
- If $\bar{A}^+$ is all attributes of a relation:
  - $\bar{A}$ is a superkey.
  - $\bar{A}$ is also a key if no set X formed by removing an attribute from $\bar{A}$ is X^+ also a key.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$

- What are all non-trivial FD's that follow from the given FD's (single right side)?
- What are all the keys?
- What are all the superkeys (that are not also keys)?

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- What are all non-trivial FD's that follow from the given FD's (single right side)?

- Need to computer closure of all 15 non-empty sets of attributes:
  - Single Attributes: A, B, C, D
  - Pairs: AB, AC, AD, BC, BD, CD
  - Triples: ABC, ABD, ACD, BCD
  - All: ABCD

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$

- What are all non-trivial FD's that follow from the given FD's (single right side)?

- Single Attributes: A, B, C, D
  - $\{A\}^+ = A$, $\{B\}^+ = B$, $\{C\}^+ = ACD$, and $\{D\}^+ = AD$
    - Yields: $C \rightarrow A$

Computing Closure

- Relation: R(A, B, C, D)
- FD's: $AB \to C$, $C \to D$, and $D \to A$
- Pairs: AB, AC, AD, BC, BD, CD
  - $\{AB\}^+ = ABCD$: yielding $AB \to D$.
  - $\{AC\}^+ = ACD$: yielding $AC \to D$
  - $\{AD\}^+ = AD$, so nothing new.
  - $\{BC\}^+ = ABCD$: yielding: $BC \to A$, $BC \to D$.
  - $\{BD\}^+ = ABCD$: yielding $BD \to A$, $BD \to C$.
  - $\{CD\}^+ = ACD$: Yielding $CD \to A$.

Computing Closure

- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$
- Triples: ABC, ABD, ACD, BCD
  - $\{ACD\}^+ = ACD$
  - $\{ABD\}^+ = \{ACD\}^+ = \{BCD\}^+ = ABCD$: yielding:
    - $ABC \rightarrow D$
    - $ABD \rightarrow C$
    - $BCD \rightarrow A$

Computing Closure

- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$
- All: ABCD
  - $\{ABCD\}^+ = ABCD$, so no new dependencies.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- What are all non-trivial FD's that follow from the given FD's (single right side)?

- C→A, AB→D, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→A.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- What are all non-trivial FD's that follow from the given FD's (single right side)?
- **What are all the keys?**
- What are all the superkeys (that are not also keys)?

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- **What are all the keys?**
- Pairs: AB, AC, AD, BC, BD, CD
  - **$\{AB\}^+$ = ABCD**: yielding AB→D.
  - $\{AC\}^+$ = ACD: yielding AC→D
  - $\{AD\}^+$ = AD, so nothing new.
  - **$\{BC\}^+$ = ABCD**: yielding: BC→A, BC→D.
  - **$\{BD\}^+$ = ABCD**: yielding BD→A, BD→C.
  - $\{CD\}^+$ = ACD: Yielding CD→A.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- **What are all the keys?**
  – AB, BC, BD

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$

- What are all the superkeys (that are not also keys)?
  - Keys: AB, BC, BD
  - Superkeys: ABC, ABD, BCD, ABCD

Projecting a Set of FD's Algorithm 3.12

- INPUT :
  - A relation R and a second relation R_1 computed by the projection $R_1 = \pi_L(R)$.
  - a set of FD's S that hold in R.
- OUTPUT : The set of FD's that hold in R_1.

Projecting a Set of FD's
Algorithm 3.12

1. Let T be the eventual output set of FD's.
   - T starts empty.
2. For each set of attributes X that is a subset of the attributes of R_1,
   - Compute X^+.
   - Using set of FD's S
   - May involve attributes that are in the schema of R but not R_1.
   - Add to T all nontrivial FD's X -> A such that A is both in X^+ and an attribute of R_1.
3. T is now basis for the FD's that hold in R_1 , but may not be minimal basis.

Projecting a Set of FD's
Algorithm 3.12

3. T is now basis for the FD's that hold in R_1 , but may not be minimal basis.
   - We may construct a minimal basis by modifying T as follows :
     1. First
        - If there is an FD F in T that follows from the other FD's in T, remove F from T. (transitive)
     2. Second
        - Let Y -> B be an FD in T, with at least two attributes in Y,
        - let Z be Y with one of its attributes removed.
        - If Z -> B follows from the FD' s in T (including Y -> B) , then replace Y -> B by Z -> B.

Algorithm 3.12 In Action

- R(A, B, C, D)
- FD's:
  - $A \rightarrow B$
  - $B \rightarrow C$
  - $C \rightarrow D$
- Now we want to project out the attribute B
  - Finding the FD's for the relation R_1(A, C, D)
- Now in theory we'll need to look at the closure of all subsets of the attributes {A, C, D}
  - 8 in total!

Algorithm 3.12
In Action

- Some Optimizations:
  - Closing the empty set and the set of all attributes will generate nothing but trivial FD's!
  - Once we know a set of attributes is a key, no need to check supersets!
- So Start with singleton sets!
- Move On to doubleton (and beyond) as needed!

Algorithm 3.12
In Action

- For each closure of some set of attributes X
  - IF X^+ includes E
  - AND E is in the schema of R_1
  - THEN: ADD FD $X \rightarrow E$
- $\{A\}^+$ includes {A, B, C, D}
  - SO ADD: $A \rightarrow C$, $A \rightarrow D$
  - DO NOT ADD: $A \rightarrow B$
    - B is not in R_1 so it doesn't make sense!

Algorithm 3.12
In Action

- $\{C\}^+$ includes $\{C, D\}$
  - SO ADD: $C \rightarrow D$

- $\{D\}^+$ includes $\{D\}$
  - SO NO NEW FD's

Doubletons

- Now since $\{A\}^+$ includes all attributes no supersets need to be considered!
- $\{C, D\}^+$ only doubleton needing consideration!
- $\{C, D\}^+ = \{C, D\}$
  - Nothing to ADD!

Final FD's

- $R_1(A, C, D)$
  - $A \rightarrow C$
  - $A \rightarrow D$
  - $C \rightarrow D$
- NOTE:
  - $A \rightarrow B$ follow from $A \rightarrow C$, $C \rightarrow D$
  - We can remove $A \rightarrow B$ (transitivity)
- Minimal Basis: $A \rightarrow C$, $C \rightarrow D$

Now Normal Forms

- Now we can define a set of requirements for our schema that insure we will not have the anomalies :
  - Redundancy
  - Update Anomalies
  - Deletion Anomalies
- Boyce-Codd Normal Form is that definition!

Boyce-Codd Normal Form

- We say a relation *R* is in *BCNF*
  - IF: given nontrivial FD *X* ->*Y* that holds in *R*,
  - THEN: *X* is a superkey.
- Note: *nontrivial* means *Y* is not contained in *X*.
- Note: a *superkey* is any superset of a key (not necessarily a proper superset).

Unnormalized Relation

Patient #	Surgeon #	Surg. date	Patient Name	Patient Addr	Surgeon	Surgery	Postop drug	ug side effec
1111	145 311	Jan 1, 1995; June 12, 1995	John White	15 New St. New York, NY	Beth Little Michael Diamond	Gallstones removal; Kidney stones removal	Penicillin, none-	rash none
1234	243 467	Apr 5, 1994 May 10, 1995	Mary Jones	10 Main St. Rye, NY	Charles Field Patricia Gold	Eye Cataract removal Thrombosis removal	Tetracycline none	Fever none
2345	189	Jan 8, 1996	Charles Brown	Dogwood Lane Harrison, NY	David Rosen	Open Heart Surgery	Cephalosporin	none
4876	145	Nov 5, 1995	Hal Kane	55 Boston Post Road, Chester, CN	Beth Little	Cholecystectomy	Demicillin	none
5123	145	May 10, 1995	Paul Kosher	Blind Brook Mamaroneck, NY	Beth Little	Gallstones Removal	none	none
6845	243	Apr 5, 1994 Dec 15, 1984	Ann Hood	Hilton Road Larchmont, NY	Charles Field	Eye Cornea Replacement Eye cataract removal	Tetracycline	Fever

First Normal Form – Atomic Entities

Patient #	Surgeon #	Surgery Date	Patient Name	Patient Addr	Surgeon Name	Surgery	Drug admin	Side Effects
1111	145	01-Jan-95	John White	15 New St. New York, NY	Beth Little	Gallstones removal	Penicillin	rash
1111	311	12-Jun-95	John White	15 New St. New York, NY	Michael Diamond	Kidney stones removal	none	none
1234	243	05-Apr-94	Mary Jones	10 Main St. Rye, NY	Charles Field	Eye Cataract removal	Tetracycline	Fever
1234	467	10-May-95	Mary Jones	10 Main St. Rye, NY	Patricia Gold	Thrombosis removal	none	none
2345	189	08-Jan-96	Charles Brown	Dogwood Lane Harrison, NY	David Rosen	Open Heart Surgery	Cephalosporin	none
4876	145	05-Nov-95	Hal Kane	55 Boston Post Road, Chester, CN	Beth Little	Cholecystectomy	Demicillin	none
5123	145	10-May-95	Paul Kosher	Blind Brook Mamaroneck, NY	Beth Little	Gallstones Removal	none	none
6845	243	05-Apr-94	Ann Hood	Hilton Road Larchmont, NY	Charles Field	Eye Cornea Replacement	Tetracycline	Fever
6845	243	15-Dec-84	Ann Hood	Hilton Road Larchmont, NY	Charles Field	Eye cataract removal	none	none

Third Normal Form -- Motivation

- There is one structure of FD's that causes trouble when we decompose.
- *AB ->C* and *C ->B*.
  - Example: *A* = street address, *B* = city, *C* = zip code.
- There are two keys, {*A,B* } and {*A,C* }.
- *C ->B* is a BCNF violation, so we must decompose into *AC*, *BC*.

We Cannot Enforce FD’s

- The problem is that if we use *AC* and *BC* as our database schema, we cannot enforce the FD *AB ->C* by checking FD’s in these decomposed relations.
- Example with *A* = street, *B* = city, and *C* = zip on the next slide.

An Unenforceable FD

street	zip
545 Tech Sq.	02138
545 Tech Sq.	02139

city	zip
Cambridge	02138
Cambridge	02139

Join tuples with equal zip codes.

street	city	zip
545 Tech Sq.	Cambridge	02138
545 Tech Sq.	Cambridge	02139

Although no FD's were violated in the decomposed relations, FD street city -> zip is violated by the database as a whole.

3NF Let’ s Us Avoid This Problem

- 3rd Normal Form (3NF) modifies the BCNF condition so we do not have to decompose in this problem situation.
- An attribute is *prime* if it is a member of any key.
- *X* ->*A* violates 3NF if and only if *X* is not a superkey, and also *A* is not prime.

What 3NF and BCNF Give You

- There are two important properties of a decomposition:

1. *Lossless Join* : it should be possible to project the original relations onto the decomposed schema, and then reconstruct the original.
2. *Dependency Preservation* : it should be possible to check in the projected relations whether all the given FD's are satisfied.

3NF and BCNF -- Continued

- We can get (1) with a BCNF decomposition.
- We can get both (1) and (2) with a 3NF decomposition.
- But we can't always get (1) and (2) with a BCNF decomposition.
  - street-city-zip is an example.

Algorithm 3.20

Algorithm 3.20: BCNF Decomposition Algorithm.

INPUT: A relation R_0 with a set of functional dependencies S_0.

OUTPUT: A decomposition of R_0 into a collection of relations, all of which are in BCNF.

METHOD: The following steps can be applied recursively to any relation R and set of FD's S. Initially, apply them with $R = R_0$ and $S = S_0$.

1. Check whether R is in BCNF. If so, nothing more needs to be done. Return $\{R\}$ as the answer.

2. If there are BCNF violations, let one be $X \to Y$. Use Algorithm 3.7 to compute X^+. Choose $R_1 = X^+$ as one relation schema and let R_2 have attributes X and those attributes of R that are not in X^+.

3. Use Algorithm 3.12 to compute the sets of FD's for R_1 and R_2; let these be S_1 and S_2, respectively.

4. Recursively decompose R_1 and R_2 using this algorithm. Return the union of the results of these decompositions.

Question

Q1 (1 point possible)

For the relation Apply(SSN,cName,state,date,major), suppose college names are unique and students may apply to each college only once, so we have two FDs: cName $\rightarrow$ state and SSN,cName $\rightarrow$ date,major. Is Apply in BCNF?

- Yes
- No

Save Submit *You have used 0 of 4 submissions*

Question

Q1 (1/1 point)

For the relation Apply(SSN,cName,state,date,major), suppose college names are unique and students may apply to each college only once, so we have two FDs: cName $\rightarrow$ state and SSN,cName $\rightarrow$ date,major. Is Apply in BCNF?

- ◯ Yes
- ◉ No ✔

EXPLANATION

{SSN,cName} is a key so only cName $\rightarrow$ state is a BCNF violation. Based on this violation we decompose into A1(cName,state), A2(SSN,cName,date,major). Now both FDs have keys on their left-hand-side so we're done.

Save | Submit | Hide Answer | *You have used 1 of 4 submissions*

Question

Q2 (1 point possible)

Consider relation Apply(SSN,cName,state,date,major) with FDs cName $\rightarrow$ state and SSN,cName $\rightarrow$ date,major. What schema would be produced by the BCNF decomposition algorithm?

- Apply(SSN,cName,state,date,major)
- A1(cName,state), A2(SSN,cName,date,major)
- A1(cName,state), A2(SSN,date,major)
- A1(cName,state), A2(SSN,cName,date), A3(SSN,cName,major)

Save Submit *You have used 0 of 4 submissions*

Question

Q2 (1/1 point)

Consider relation Apply(SSN,cName,state,date,major) with FDs cName $\rightarrow$ state and SSN,cName $\rightarrow$ date,major. What schema would be produced by the BCNF decomposition algorithm?

- ○ Apply(SSN,cName,state,date,major)
- ● A1(cName,state), A2(SSN,cName,date,major) ✔
- ○ A1(cName,state), A2(SSN,date,major)
- ○ A1(cName,state), A2(SSN,cName,date), A3(SSN,cName,major)

EXPLANATION

{SSN,cName} is a key so only cName $\rightarrow$ state is a BCNF violation. Based on this violation we decompose into A1(cName,state), A2(SSN,cName,date,major). Now both FDs have keys on their left-hand-side so we're done.

Exercise 3.3.1

- For each of the following relation schemas and sets of FD ' s :
  a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A
- do the following :
  – i) Indicate all the BCNF violations . Do not forget to consider FD's that are not in the given set, but follow from them. However, it is not necessary to give violations that have more than one attribute on the right side.
  – ii) Decompose the relations, as necessary, into collections of relations that are in BCNF.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$

- What are all non-trivial FD's that follow from the given FD's (single right side)?
- What are all the keys?

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- What are all non-trivial FD's that follow from the given FD's (single right side)?

- Need to computer closure of all 15 non-empty sets of attributes:
  – Single Attributes: A, B, C, D
  – Pairs: AB, AC, AD, BC, BD, CD
  – Triples: ABC, ABD, ACD, BCD
  – All: ABCD

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- What are all non-trivial FD's that follow from the given FD's (single right side)?

- Single Attributes: A, B, C, D
  - $\{A\}^+ = A$, $\{B\}^+ = B$, $\{C\}^+ = ACD$, and $\{D\}^+ = AD$
    - Yields: C $\rightarrow$ A

Computing Closure

- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A
- Pairs: AB, AC, AD, BC, BD, CD
  - $\{AB\}^+$ = ABCD: yielding AB→D.
  - $\{AC\}^+$ = ACD: yielding AC→D
  - $\{AD\}^+$ = AD, so nothing new.
  - $\{BC\}^+$ = ABCD: yielding: BC→A, BC→D.
  - $\{BD\}^+$ = ABCD: yielding BD→A, BD→C.
  - $\{CD\}^+$ = ACD: Yielding CD→A.

Computing Closure

- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$
- Triples: ABC, ABD, ACD, BCD
  - $\{ACD\}^+ = ACD$
  - $\{ABD\}^+ = \{ACD\}^+ = \{BCD\}^+ = ABCD$: yielding:
    - $ABC \rightarrow D$
    - $ABD \rightarrow C$
    - $BCD \rightarrow A$

Computing Closure

- Relation: R(A, B, C, D)
- FD’s: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A
- All: ABCD
  - $\{ABCD\}^+$ = ABCD, so no new dependencies.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- What are all non-trivial FD's that follow from the given FD's (single right side)?

- **C→A, AB→D, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→A.**

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - i) Indicate all the BCNF violations .
- **FD’S that Follow: C→A, AB→D, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→A.**

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - i) Indicate all the BCNF violations .
- **Step 1 FD’S:**
  - AB -> C, C -> D, D -> A
  - C→A, AB→D, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→A.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$

- What are all non-trivial FD's that follow from the given FD's (single right side)?
- **What are all the keys?**

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- **What are all the keys?**
- Pairs: AB, AC, AD, BC, BD, CD
  - **$\{AB\}^+$ = ABCD**: yielding AB→D.
  - $\{AC\}^+$ = ACD: yielding AC→D
  - $\{AD\}^+$ = AD, so nothing new.
  - **$\{BC\}^+$ = ABCD**: yielding: BC→A, BC→D.
  - **$\{BD\}^+$ = ABCD**: yielding BD→A, BD→C.
  - $\{CD\}^+$ = ACD: Yielding CD→A.

Computing Closure

- Exercise 3.2.1
- Relation: R(A, B, C, D)
- FD's: AB $\rightarrow$ C, C $\rightarrow$ D, and D $\rightarrow$ A

- **Step 2: What are all the keys?**
  - AB, BC, BD

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - i) Indicate all the BCNF violations .
- **FD'S:**
  - AB -> C , C -> D, D -> A
  - C→A, AB→D, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→A.
- **Keys:**
  - AB, BC, BD

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - i) Indicate all the BCNF violations .
- Any FD that does not have **Key on the Left** is a BCNF violation.
- **FD'S:**
  - AB -> C , C -> D, D -> A
  - C→A, AB→D, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→A.
- **Keys:** AB, BC, BD

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - i) Indicate all the BCNF violations .
- Any FD that does not have **Key on the Left** is a BCNF violation.
- **FD’S Violations:**
  - AB -> C , **C -> D**, and **D -> A**
  - **C→A**, AB→D, **AC→D**, BC→A, BC→D, BD→A, BD→C, **CD→A**, ABC→D, ABD→C, and BCD→A.
- Keys: AB, BC, BD

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - i) Indicate all the BCNF violations .
- **FD'S Violations:**
  - AB -> C , **C -> D**, and **D -> A**
  - **C→A**, AB→D, **AC→D**, BC→A, BC→D, BD→A, BD→C, **CD→A**, ABC→D, ABD→C, and BCD→A.
- **Keys: AB, BC, BD**
- Any FD that does not have **Key on the Left** is a BCNF violation.
- **Needed all FD's to easily check for Violations!**

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - ii) Decompose the relations, as necessary, into collections of relations that are in BCNF.
- **FD'S Violations:**
  - AB -> C , **C -> D**, and **D -> A**
  - **C→A**, AB→D, **AC→D**, BC→A, BC→D, BD→A, BD→C, **CD→A**, ABC→D, ABD→C, and BCD→A.
- **Keys: AB, BC, BD**
- Decompose using violation C->D

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - ii) Decompose the relations, as necessary, into collections of relations that are in BCNF.
- **FD'S Violations:**
  - AB -> C , **C -> D**, and **D -> A**
  - **C→A**, AB→D, **AC→D**, BC→A, BC→D, BD→A, BD→C, **CD→A**, ABC→D, ABD→C, and BCD→A.
- **Keys: AB, BC, BD**
- Decompose using violation C->D
- C^+={A, C, D}, R_1= R1(A, C, D)
- R_2 attributes = C U {A, B, C, D} – C^+ = {B, C} = R2(B, C)

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - ii) Decompose the relations, as necessary, into collections of relations that are in BCNF.
- **FD'S Violations:**
  - AB -> C , **C -> D**, and **D -> A**
  - **C→A**, AB→D, **AC→D**, BC→A, BC→D, BD→A, BD→C, **CD→A**, ABC→D, ABD→C, and BCD→A.
- **Keys: AB, BC, BD**
- Decompose using violation C->D
- C^+={A, C, D}, R_1= R1(A, C, D)
- R_2 attributes = C U {A, B, C, D} – C^+ = {B, C} = R2(B, C)
- R1(A, C, D), R2(B, C)
  - Violations?

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - ii) Decompose the relations, as necessary, into collections of relations that are in BCNF.
- Decomposed Relations:
  - R1(A, C, D), R2(B, C)
- **FD’S Violations:**
  - AB -> C , C -> D, and **D -> A w/ R1(A, C, D)**

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - ii) Decompose the relations, as necessary, into collections of relations that are in BCNF.
- Decomposed Relations:
  - R1(A, C, D), R2(B, C)
- **FD'S Violations:**
  - AB -> C , C -> D, and **D -> A w/ R1(A, C, D)**
  - **D^+ = {A, D} so R_3 = R3(A, D)**
  - R_4 attributes = D ∪ {A, C, D} – D^+ = {C, D} = R4(C, D)

Exercise 3.3.1

a) R (A, B, C, D) with FD ' s AB -> C , C -> D , and D -> A

- do the following :
  - ii) Decompose the relations, as necessary, into collections of relations that are in BCNF.
- **Decomposed Relations in BCNF:**
  - **R2(B, C), R3(A, D), R4(C, D)**

Definition of MVD

- A *multivalued dependency* (MVD) on *R*, *X* ->->*Y* , says that if two tuples of *R* agree on all the attributes of *X*, then their components in *Y* may be swapped, and the result will be two tuples that are also in the relation.
- i.e., for each value of *X*, the values of *Y* are independent of the values of *R*-*X*-*Y*.

Example: MVD

LikesMovie(name, addr, phones, MovieLiked)

- A movie goer's phones are independent of the movies they like.
  - name->->phones and name ->->MovieLiked.
- Thus, each of a movie goer's phones appears with each of the movies they like in all combinations.
- This repetition is unlike FD redundancy.
  - name->addr is the only FD.

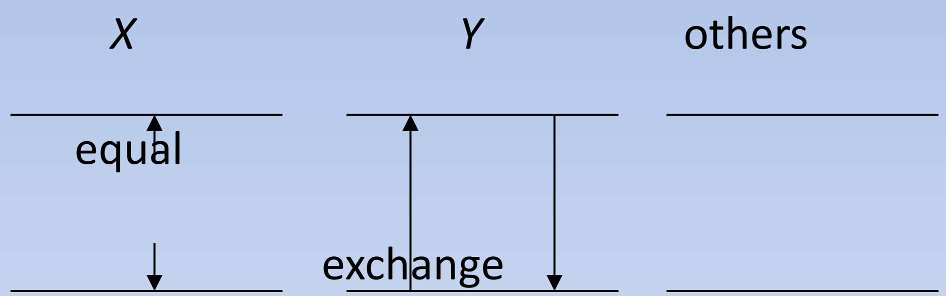
Tuples Implied by name->->phones

If we have tuples:

name	addr	phones	movieLiked
sue	a	p1	b1
sue	a	p2	b2
sue	a	p2	b1
sue	a	p1	b2

Then these tuples must also be in the relation.
(MVD's are Tuple Generators)

Picture of MVD *X* ->->*Y*



MVD Rules: Trivial MVD's

- X->>Y is a Trivial MVD if
  - Attributes of X are equal or a subset of attributes of Y
  - For relation R where Attributes of X combined with attributes of Y contain all attributes of relation R

MVD Rules

- Every FD is an MVD (*promotion*).
  - If *X* ->*Y*, then swapping *Y* 's between two tuples that agree on *X* doesn't change the tuples.
  - Therefore, the "new" tuples are surely in the relation, and we know *X* ->->*Y*.
- *Complementation* : If *X* ->->*Y*, and *Z* is all the other attributes, then *X* ->->*Z*.

Fourth Normal Form

- The redundancy that comes from MVD’s is not removable by putting the database schema in BCNF.
- There is a stronger normal form, called 4NF, that (intuitively) treats MVD’s as FD’s when it comes to decomposition, but not when determining keys of the relation.

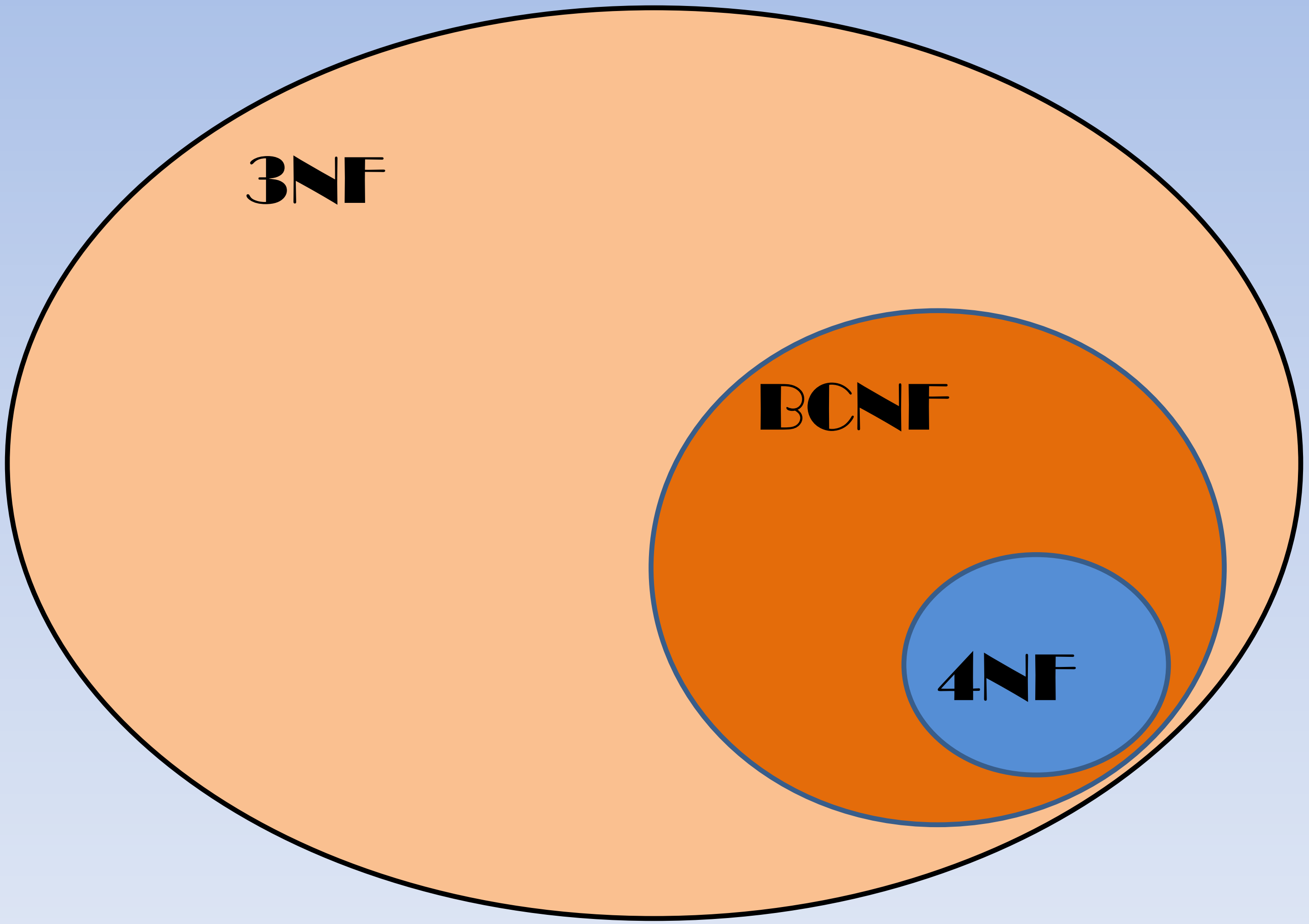
4NF Definition

- A relation *R* is in *4NF* if: whenever $X \twoheadrightarrow Y$ is a nontrivial MVD, then *X* is a superkey.
  - *Nontrivial MVD* means that:
    1. *Y* is not a subset of *X*, and
    2. *X* and *Y* are not, together, all the attributes.
  - Note that the definition of "superkey" still depends on FD's only.

BCNF Versus 4NF

- Remember that every FD *X* ->*Y* is also an MVD, *X* ->->*Y*.
- Thus, if *R* is in 4NF, it is certainly in BCNF.
  - Because any BCNF violation is a 4NF violation (after conversion to an MVD).
- But *R* could be in BCNF and not 4NF, because MVD's are "invisible" to BCNF.

3NF vs. BCNF vs. 4NF



Question

Q1 (1 point possible)

Consider a relation R(A,B,C) with multivalued dependency $A \twoheadrightarrow B$. Suppose there at least 3 different values for A, and each value of A is associated with at least 4 different B values and at least 5 different C values. What is the minimum number of tuples in R?

- 60
- 15
- 12
- 27

Save Submit *You have used 0 of 4 submissions*

Question

Q1 (1/1 point)

Consider a relation R(A,B,C) with multivalued dependency A ↠ B. Suppose there at least 3 different values for A, and each value of A is associated with at least 4 different B values and at least 5 different C values. What is the minimum number of tuples in R?

- (selected) 60 ✓
- 15
- 12
- 27

EXPLANATION

Multivalued dependency A ↠ B says that for each value of A, we must have every combination of B and C values. So for each of the 3 values of A we must have at least 4*5=20 different tuples.