Concepts from 3.1-3.2

- Functional dependencies
- Keys & superkeys of a relation
- Reasoning about FDs
- Closure of a set of attributes
- Closure of a set of FDs
- Minimal basis for a set of FDs

Plan

- How can we use FDs to show that a relation has an anomaly (a potential problem)?
- How can we algorithmically fix the problem?

Projecting sets of FDs

- Suppose we have a relation R and set of FDs F
- Let S be a relation obtained by projecting R into a subset of the attributes of R $\pi_{Attributes}(R)$
- The **projection** F_S of F is the set of FDs that follow from F and hold in S
  - Involve only attributes of S

Projecting sets of FDs

- Algorithm for computing F_S :
  - Compute closure F^+
  - F_S is the set of all FDs in F^+ that involve only the attributes in S
- Book describes a different algorithm in section 3.2.8.
- Book's algorithm also shows how to compute a minimal basis of F_S

Projecting sets of FDs

- R(A, B, C, D); F = {A→B, B→C, C→D}
- Which FDs hold in S(A, C, D)?

F^+ is {A→B, B→C, C→D, A→C, A→D, B→D}

F_S is {C→D, A→C, A→D}

Anomalies

- An anomaly is a problem that arises when we try to add too many attributes to a single relation.
- Arises from ***redundancy***: information repeated unnecessarily.
  - When designing schemas, try to ensure you never repeat yourself!

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

Anomalies

- Update anomaly: when you change information in one tuple but leave the same information in a different tuple unchanged.

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

Anomalies

- Deletion anomaly: when deleting one or more tuples removes information that we didn't want to lose.

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

Anomalies

- Insertion anomaly (left out of book): when storing a piece of information forces us to store an unrelated piece of information as well.

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


DECOMPOSITION
a poem by Jordan
In the cold winter time, plant cells freeze and thaw;
This process is just part of nature's law.
The cell walls burst and get all squishy...
That's why this old pumpkin here is so mushy!
(Eventually this pumpkin, like all dead plants, will "decompose," or turn back into soil! The process is called "decomposition.")

Decomposing Relations

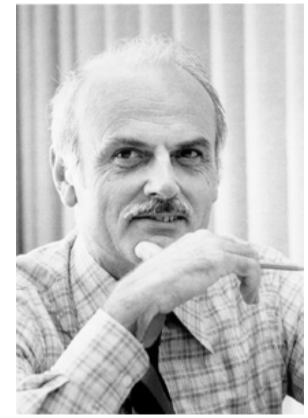
- Given a relation R(A1, A2…, An), two relations S(B1, B2…, Bm) and T(C1, C2…, Ck) form a decomposition of R if:

1. the attributes of S and T together make up the attributes of R, i.e., {A's} = {B's} U {C's}
2. the tuples in S are the projections into {B1…Bm} of the tuples of R i.e. $S \equiv \pi_{B1,B2,..,Bm}(R)$
3. the tuples in T are the projections into {C1…Ck} of the tuples of R i.e. $T \equiv \pi_{C1,C2,..,Ck}(R)$

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

- Decompose into
  - Movies(title, year, length, genre, studio)
  - Stars(title, year, star)
- Are the anomalies removed? Is anything redundant? Why or why not? Do you see a connection to FDs?

BCNF



- Anomalies are guaranteed not to exist when a relation is in ***Boyce-Codd normal form*** (BCNF).
- A relation R is in BCNF iff whenever there is a nontrivial FD $A_1 \ldots A_n$->$B_1 \ldots B_m$ for R, $\{A_1, \ldots, A_n\}$ is a superkey for R.
- Informally, the left side of every nontrivial FD must be a superkey.

Check for BCNF violations

- List all nontrivial FDs in R.
- Ensure left side of each nontrivial FD is a superkey.
- (First have to find all the keys!)

Note: a relation with two attributes is always in BCNF.

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

- Decompose into
  - Movies(title, year, length, genre, studio)
  - Stars(title, year, star)
- What are the new FDs and keys?

Example….

- Is Courses(Number, DepartmentName, CourseName, Classroom, Enrollment, StudentName, Address) in BCNF?
- FDs:
  - Number DepartmentName $\rightarrow$ CourseName
  - Number DepartmentName $\rightarrow$ Classroom
  - Number DepartmentName $\rightarrow$ Enrollment
- What is $\{$Number, DepartmentName$\}^+$ under the FDs?

{Number, DepartmentName, Coursename, Classroom, Enrollment}

- So the key is {Number, DepartmentName, StudentName, Address}
- So the relation is not in BCNF.

Decomposition into BCNF

- Suppose R is a relation schema that violates BCNF
- We can decompose R into a set *S* of new relations such that:
  - each relation in *S* is in BCNF and
  - we can "recover" R from the relations in *S*, i.e., we can reconstruct R exactly from the relations in *S*

Algorithm: Given relation R and set of FDs F:

- Check if R is in BCNF, if not, do:
- If there are FDs that violate BCNF, call one X -> Y. Compute X^+. Let R1 = X^+ and R2 = X and all other attributes not in X^+.
- Compute FDs for R1 and R2 (projection algorithm for FDs).
- Check if R1 and R2 are in BCNF, and repeat if needed.

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

FDs: title year -> length genre studio

Key: {title, year, star}

This relation is not in BCNF (the single FD is a violation because the LHS (title, year) is not a superkey).

Decompose:

- Compute $\{title, year\}^+$ = {title, year, length, genre, studio}
- New relation: R1(title, year, length, genre, studio). Key = {title, year}
- New relation: R2(title, year, star). Key = {title, year, star}
- FDs for R1: Same FD as for original relation. FDs for R2: none
- No BCNF violations in R1 or R2. (LHS of FD in R1 is a superkey.)

- Schema is Courses(Number, DeptName, CourseName, Classroom, Enrollment, StudentName, Address)
- BCNF-violating FD is

Number DeptName → CourseName Classroom Enrollment

- What is {Number, DeptName}+ ?

{Number, DeptName, CourseName, Classroom, Enrollment}

- Decompose Courses into

Courses1(Number, DeptName, CourseName, Classroom, Enrollment)

and

Courses2(Number, DeptName, StudentName, Address)

Are there any BCNF violations in the two new relations?

Students and Profs

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

Warmup for Nov 8

Attributes:

Author,

Address (of a person),

Title (of a book),

Genre (of a book),

Pages (in a book).

Suppose we have the FDs:

Author -> Address

Title -> Genre Pages

Decompose into BCNF.

- There are other types of decomposition besides BCNF. Why should we use this one and not another?
- We'd like a decomposition to:
  1. eliminate anomalies
  2. let us recover the original relation with a join (lossless join property)
  3. let us recover the original FDs when recovering the original relation (dependency preservation property)
- BCNF decomposition gives us 1 & 2, but not 3.

- BCNF decomposition guarantees:
  - There are no redundancy, insertion, update, or deletion anomalies.
  - We can recover the original relation with a *natural join. (lossless join property)*
- However, we might lose some original FDs in the natural join.

Name	Type	Closest Restaurant of Type
Alice	BBQ	Cozy Corner
Alice	Thai	Bhan Thai
Bob	Pizza	Broadway Pizza
Charlie	Doughnuts	Gibson's Donuts
Charlie	Thai	Bangkok Alley
Charlie	BBQ	Cozy Corner

Suppose we want to store information about the closest restaurant to various people that serves a certain type of cuisine.

What are FDs/keys? Is this in BCNF?

Name	Closest
Alice	Cozy Corner
Alice	Bhan Thai
Bob	Broadway Pizza
Charlie	Donald's Donuts
Charlie	Bangkok Alley
Charlie	Cozy Corner

Restaurant	Type
Cozy Corner	BBQ
Bhan Thai	Thai
Broadway Pizza	Pizza
Donald's Donuts	Doughnuts
Bangkok Alley	Thai

Book's example

- Traveling shows:
  - Store theater names, the cities they are in, and the title of the show playing.
  - A show never plays at more than one theater per city.
- theater -> city
- title city -> theater

3rd Normal Form (3NF)

- Allows for lossless joins and dependency preservation.
- Does not fix all anomalies.
- 3NF is a weaker condition than BCNF (anything in BCNF is automatically in 3NF).

3rd Normal Form (3NF)

- A relation R is in 3NF iff for every nontrivial FD A1…An -> B for R, one of the following is true:
  - A1…An is a superkey for R (BCNF test)
  - Each B is a ***prime*** attribute (an attribute in *some* key for R)

Example

- R(C, D, P, S, Y) has FDs
  - PSY → CD
  - CD → S
- Keys are {P, S, Y} and {C, D, P, Y}
- CD → S violates BCNF
- However, R is in 3NF because S is part of a key

3NF Decomposition

- Given a relation R and set F of functional dependencies:

1. Find a minimal basis, G, for F.
2. For each FD X -> A in G, use XA as the schema of one of the relations in the decomposition.
3. If none of the sets of schemas from Step 2 is a superkey for R, add another relation whose schema is a key for R.

Example

- Example:

  R(A, B, C)

  F: {A→B, C→B}

- What is the minimal basis set of FDs?

- What is the decomposition to 3NF?

More redundancy?

Course	Textbook	Prof
ENGL 101	Writing for Dummies	Smith
ENGL 101	Wikipedia Is Not a Primary Source	Smith
ENGL 101	Writing for Dummies	Jones
ENGL 101	Wikipedia Is Not a Primary Source	Jones
COMP 142	How to Program in C++	Smith
COMP 142	How to Program in C++	Jones

Every professor always uses the same set of books.

Is this in BCNF?

- Redundancies can still arise in relations that conform to BCNF.
- Occurs when a single table tries to contain two (or more) many-one (or many-many) relationships.

Course	Textbook	Prof
ENGL 101	Writing for Dummies	Smith
ENGL 101	Wikipedia Is Not a Primary Source	Smith
ENGL 101	Writing for Dummies	Jones
ENGL 101	Wikipedia Is Not a Primary Source	Jones
COMP 142	How to Program in C++	Smith
COMP 142	How to Program in C++	Jones

Multivalued dependencies

- A ***MVD*** is a constraint that two sets of attributes are ***independent*** of each other.
- A MVD A1…An ->-> B1…Bm holds in R if in every instance of R:
  - for every pair of tuples t and u that agree on all the As, we can find a tuple v in R that agrees
    - with both t and u on the As
    - with t on the Bs
    - with u on all those attributes of R that are not As or Bs
- In other words, the information in A1..An determines the values of the set of tuples for B1..Bm ***and*** those tuples are independent of any other attributes in the relation.

- Consider a table with actors/actresses, their street addresses with cities/states/zips, and the movies they've been in (title/year).

- Consider a MVD A1…An ->-> B1…Bm.
- Call attributes not in A's or B's the Cs.
- This MVD holds in R if:
  - whenever we have two tuples of R that agree on the A's but differ on the B's and C's we should be able to find (or create) two new tuples with the same A's but swapped B's and C's.
- Equivalently:
  - If knowing A1…An determines a unique set of tuples for B1..Bm that is independent of the C's.

Course	Textbook	Prof
ENGL 101	Writing for Dummies	Smith
ENGL 101	Wikipedia Is Not a Primary Source	Smith
ENGL 101	Writing for Dummies	Jones
ENGL 101	Wikipedia Is Not a Primary Source	Jones
COMP 142	How to Program in C++	Smith
COMP 142	How to Program in C++	Jones

- Course →→ Textbook is an MVD
- What else?

FDs vs MVDs

- A FD A -> B says "Each A determines a unique B"
  - or, "Each A determines 0 or 1 Bs."
- A MVD A ->-> B says "Each A determines a set of Bs ***where the Bs are independent of anything in the relation that is not an A or a B.***"

Rules for MVDs

- **FD promotion:** Every FD A→B is an MD A→→B
- **Trivial MDs:**

1. If A→→B, then A→→AB
2. If A1, A2…, An and B1, B2, …, Bm make up *all* the attributes of a relation, then A1, A2, …An →→ B1, B2, …Bm holds in the relation

- **Transitive rule:** Given A→→B and B→→C, we can infer A→→C.
- **Complementation rule:** if we know A→→B, then we know A→→C, where all the Cs are attributes not among the As or Bs.

- Note that the **splitting rule does not hold!** If $A \twoheadrightarrow BC$, then it is not true that $A \twoheadrightarrow B$ and $A \twoheadrightarrow C$.

Fourth Normal Form (4NF)

- "Stronger" than BCNF.
- A relation R is in 4NF iff:
  - for all MVDs A1…An ->-> B1…Bm,
    {A1, …, An} is a superkey of R.

4NF Decomposition

- Consider relation R with set of attributes X
- A1 A2 … An $\rightarrow\rightarrow$ B1 B2 … Bm violates 4NF
- Decompose R into two relations whose attributes are:

1. The As and Bs together, i.e., {A1 A2 … An, B1, B2, …, Bm}
2. All the attributes of R which are not Bs, i.e. X – {B1, B2 …, Bm}
3. Recursively check if the new relations are in 4NF and repeat

Example

Course	Textbook	Prof
ENGL 101	Writing for Dummies	Smith
ENGL 101	Wikipedia Is Not a Primary Source	Smith
ENGL 101	Writing for Dummies	Jones
ENGL 101	Wikipedia Is Not a Primary Source	Jones
COMP 142	How to Program in C++	Smith
COMP 142	How to Program in C++	Jones

- Course ->-> Textbook
- Course ->-> Professor

Example

`Drinkers(name,addr,phones,beer)`

- FD: `name` → `addr`
- Nontrivial MVD's:

  `name` →→ `phone` and
  `name` →→ `beer`.
- Only key: {`name, phones, beer`}
- All three dependencies above violate 4NF.
- Successive decomposition yields 4NF relations:

  D1(<u>name</u>, addr)

  D2(<u>name</u>, <u>phones</u>)

  D3(<u>name</u>, <u>beer</u>)

Relationships Among Normal Forms

- 4NF implies BCNF, i.e., if a relation is in 4NF, it is also in BCNF
- BCNF implies 3NF, i.e., if a relation is in BCNF, it is also in 3NF

Property	3NF	BCNF	4NF
Eliminate redundancy due to FDs	Maybe	Yes	Yes
Eliminate redundancy due to MDs	No	No	Yes
Preserves FDs	Yes	Maybe	Maybe
Preserves MDs	Maybe	Maybe	Maybe

Normal Forms

- First Normal Form: each attribute is atomic
- Second Normal Form: No non-trivial FD has a left side that is a proper subset of a key
- Third Normal Form: just discussed it
- Fourth Normal Form: just discussed it
- Fifth Normal Form: outside the scope of this class
- Sixth Normal Form: different versions exist. One version developed for temporal databases
- Seventh Normal Form
  - just kidding ☺

Database Design Mantra

- "everything should depend on the key, the whole key, and nothing but the key"