Designing Databases

- Proposals to design
- How do we know our design is optimal?
- Explore theory and techniques for better design.

- Two goals
  - Eliminate duplication
  - Avoid anomalies

Redundancy and Anomalies

In the Movies Relation below,

- Eliminating Redundancy
  - title, year, length, genre, studioName are repeated for every star name
- Anomalies
  - Update anomalies
    - Change studio name from Fox to something else.
    - Change star's last name
  - Deletion anomalies
    - Remove the genre drama or remove Vivien Leigh

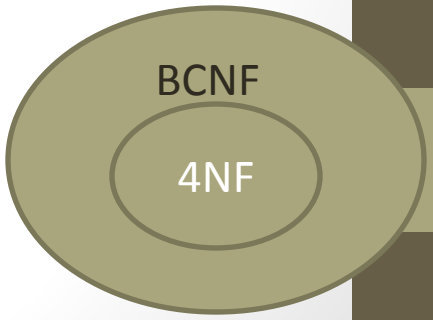
title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992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

Movies Relational Schema

```
Movies(
    title:string,
    year:integer,
    length:integer,
    genre:string,
    studioName:string,
    producerC#:integer
)
MovieStar(
    name:string,
    address:string,
    gender:char,
    birthdate:date
)
StarsIn(
    movieTitle:string,
    movieYear:integer,
    starName:string
)
MovieExec(
    name:string,
    address:string,
    cert#:integer,
    netWorth:integer
)
Studio(
    name:string,
    address:string,
    presC#:integer
)
```

Design by decomposition

- Imagine all the data in one big long table or “mega relation” and apply the different rules or properties to this data
- Design by decomposition
- Decompose based on properties
  - Functional Dependencies (FDs) that would result in Boyce-Codd Normal Form (BCNF)
  - Multi-valued dependencies that would result in Fourth Normal Form
  - 4NF more restrictive than BCNF
- Keep real world data rules in mind as the decomposition is done



Normalization

- Technique used to decompose relations into two or more relations to remove redundancy and anomalies

- First Normal Form → Every component is an atomic value (No multivalued components)
- Second Normal Form → Remove Partial dependencies (Non key to Partial Key) into their own relations.
- Third Normal Form → Remove transitive dependencies (Non Key to Non Key) into their own relations.

- We are interested in BCNF and Fourth Normal form as these will automatically conform to the three normal forms above.

Functional Dependencies (FDs) - I

Consider the following Movies "mega relation"

Movies(title, year, length, genre, studioName, studioAddress, starName, starAddress, starGender, starBirthDate)

Title, year -> length, genre, studioName, studioAddress

Given a title and year, we will get the same data every time. Notice no star info, why?

Title and year functionally determine length, genre, studioName, studioAddress.

Title	Year	Length	Genre	studioName	studioAddress	starName	starAddress	starGender	starBirthDate
Star wars	1977	124	Action	Fox	123 Hollywood Blvd.	Mark Hamill	456 Oak Rd., Malibu	M	1950-01-13
Star wars	1977	124	Action	Fox	123 Hollywood Blvd.	Carrie Fisher	123 Maple St.,Hollywood	F	1960-09-09
Star wars	1977	124	Action	Fox	123 Hollywood Blvd.	Harrison Ford	789 Palm Dr., Beverly Hills	M	1952-04-12

Functional Dependencies (FDs) - II

Consider the following Movies "megarelation"

Movies(title, year, length, genre, studioName, studioAddress, starName, starAddress, starGender, starBirthDate)

Functional dependencies

title, year -> length, genre, studioName, studioAddress

title, year, starName -> starAddress, starGender, starBirthDate

BCNF: If A → B, A is a key.

(If A functionally determines B, A is a key)

Functional Dependency (Formal Def.)

A functional dependency (FD) on a relation R is a statement of the form

If two tuples of R agree on all of the attributes $A_1,A_2,\ldots,A_n$ (i.e., the tuples have the same values in their respective components for each of these attributes), then they must also agree on all of another list of attributes $B_1,B_2,\ldots,B_m$.

We write this FD formally as $A_1,A_2,\ldots,A_n \rightarrow B_1,B_2,\ldots,B_m$ and say that "$A_1,A_2,\ldots,A_n$ functionally determine $B_1,B_2,\ldots,B_m$ "

Movies (title , year , length , genre , studioName , studioLocation, starName)

Functional Dependencies

title, year → length, genre, studioName, studioLocation

title, year → starName (NOT a FD, why?)

Must think about all instances of the data

Keys for a relation

- FDs determine the Keys for a relation
- We say a set of one or more attributes A_1, A_2,…, A_n is a key for a relation R if :
  - Those attributes functionally determine all other attributes of the relation . That is, it is impossible for two distinct tuples of R to agree on all of A_1, A_2,…, A_n.
  - No proper subset of A_1, A_2,…, A_n functionally determines all other attributes of R; i.e., a key must be minimal.

Movies (title , year , length , genre , studioName , studioLocation, starName)

title, year, starName → length, genre, studioName, studioLocation

Multi-valued Dependencies

- Remove multi-valued dependencies to put the relations in Fourth Normal Form

Movies (title , year , length , genre , studioName, studioLocation)

- Redundancy and anomalies not addressed in BCNF
- How many times is studioLocation repeated for each movie?

Decompose Movies into

Movies (title , year , length , genre , studioName)

Studio(studioName, studioLocation)

Superkeys

- A set of attributes that contains a key is called a superkey (also called candidate keys), short for superset of a key.
- Superkey satisfies the first condition of a key: it functionally determines all other attributes of the relation.
- A superkey need not satisfy the second condition : minimality.

Example:

- {title , year , starName} is a super key
- {title , year , starName , length , studioName} is also a super key

FD Exercises(1)

- Consider a relation about people in the United States , including their name, Social Security number, street address , city, state, ZIP code, area code, and phone number (7 digits) .

- What FD's would you expect to hold? What are the keys for the relation? To answer this question, you need to know something about the way these numbers are assigned. For instance , can an area code straddle two states? Can a ZIP code straddle two area codes? Can two people have the same Social Security number? Can they have the same address or phone number?

FD Exercises(2)

- 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 → B
(b) A → C
(c) B → A
(d) B → C
(e) C → A
(f) C → B
(g) AB → C
(h) AC → B
(i) BC → A

Rules for inferring FDs – Armstrong's axioms

Armstrong's axioms

A, B, C are sets of attributes

- Reflexivity
  - If A⊇B, then A→B

  title, year -> title
- Augmentation
  - If A->B, then AC→BC for any C

  studio name -> studio address
  studio name, owner -> studio address, owner
- Transitivity
  - If A→B and B→C, then A→C

  course -> course description and course description -> credits
  course -> credits

- More rules
  - Union: If A→B and A→C, then A→BC
  - Decomposition: If A→BC, then A→B and A→C

Trivial, Nontrivial, completely nontrivial FDs

- Trivial - A constraint of any kind on a relation is said to be trivial if it holds for every instance of the relation, regardless of what other constraints are assumed.
  - If A → B, then B ⊆ A
  - Example: title year → title
- Nontrivial
  - If A→ B, then B $\not\subseteq$ A
  - Example:

  title, year -> length

  (Applying augmentation) title, year → title, year, length

  {title, length} is not a subset of {title, year}
- Completely Nontrivial
  - If A→ B, then A ∩ B = {} (empty set)
  - Example:

  studentID → StudentName but not StudentName → StudentID

Rules for inferring FDs – Splitting/Combining Rule

$A_1,A_2,\ldots,A_n \rightarrow B_1,B_2,\ldots,B_m$

Is equivalent to

$A_1,A_2,\ldots,A_n \rightarrow B_1$

$A_1,A_2,\ldots,A_n \rightarrow B_2$

…

$A_1,A_2,\ldots,A_n \rightarrow B_m$

Can only apply to the right sides.

title year → length genre studioName

title year → length
title year → genre
title year → studioName

Closure of attributes

Let R be a relation with the attributes {A, B, C, D, E, F}
and FDs are AB→ C, BC → AD, D → E, CF → B
What is the closure of {A, B}, or $\{A, B\}^+$?

1) BC → AD is BC → A and BC → D (Using the splitting rule)
2) X = {A, B}
3) Since AB→ C, X = {A, B, C}

Since BC → D, X = {A, B, C, D}
Since D → E, X = {A, B, C, D, E}

$\{A, B\}^+ = \{A, B, C, D, E\}$

What does this tell us about the attributes A,B?

Closure (Formal Def.)

Given relation, FDs, set of attributes $\{A_1,A_2,\ldots,A_n\}$
Find all B such that $\{A_1,A_2,\ldots,A_n\} \rightarrow B$
We denote the closure of a set of attributes $\{A_1,A_2,\ldots,A_n\}$ by $\{A_1,A_2,\ldots,A_n\}^+$.

Algorithm:
INPUT : A set of attributes $\{A_1,A_2,\ldots,A_n\}$ and a set of FD's S.
OUTPUT : The closure $\{A_1,A_2,\ldots,A_n\}^+$.

1. If necessary, split the FD's of S, so each FD in S has a single attribute on the right.

2. Let X be a set of attributes that eventually will become the closure . Initialize X to be $\{A_1,A_2,\ldots,A_n\}$.

3. Repeatedly search for some FD $B_1,B_2,\ldots,B_m \rightarrow C$ such that all $B_1,B_2,\ldots,B_m$ are in the set of attributes X , but C is not . Add C to the set X and repeat the search .

4. The set X , after no more attributes can be added to it , is the correct value of $\{A_1,A_2,\ldots,A_n\}^+$.

Another Closure Example

Using relation

Movies (title , year , length , genre , studioName , studioLocation, starName)

and the following FDs

title, year → length, genre, studioName

studioName → studioLocation

Compute $\{title, year, starName\}^+$

Compute $\{title, year\}^+$

Using closure to determine keys

- Consider the relation R(A,B,C,D,E) and suppose we have the functional dependencies:

  AB → C
  AE → D
  D → B

  Which of the following attribute pairs is a key for R?

1. AB
2. AC
3. AD
4. AE

Closure Exercises

Exercise 3.2.1: Consider a relation with schema $R(A, B, C, D)$ and FD's $AB \to C$, $C \to D$, and $D \to A$.

a) What are all the nontrivial FD's that follow from the given FD's? You should restrict yourself to FD's with single attributes on the right side.

b) What are all the keys of R?

c) What are all the superkeys for R that are not keys?

Exercise 3.2.2: Repeat Exercise 3.2.1 for the following schemas and sets of FD's:

i) $S(A, B, C, D)$ with FD's $A \to B$, $B \to C$, and $B \to D$.

Decomposing Relations

Decomposition of R (a Relation) involves splitting the attributes of R to make the schemas of two new relations.

- Given a relation R (A_1,A_2,…,A_n), we may decompose R into two relations S (B_1,B_2,…,B_m) and T (C_1,C_2,…,C_k) such that :

1 . {A_1,A_2,…,A_n} = {B_1,B_2,…,B_m } U {C_1,C_2,…,C_k}.

2 . $S = \prod_{B1,B2,\ldots,Bm} (R)$.

3. $T = \prod_{C1,C2,\ldots,Ck} (R)$.

Also think of $S \bowtie T = R$ (S natural join T results in R)

Movies Relation decomposed

title	*year*	*length*	*genre*	*studioName*	*starName*
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*	*genre*	*studioName*
Star Wars	1977	124	sciFi	Fox
Gone With the Wind	1939	231	drama	MGM
Wayne's World	1992	95	comedy	Paramount

(b) The relation Movies2.

title	*year*	*starName*
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With the Wind	1939	Vivien Leigh
Wayne's World	1992	Dana Carvey
Wayne's World	1992	Mike Meyers

(b) The relation Movies3.

Movies = Movies2 ⋈ Movies3

Movies = Movies2 U Movies3

Boyce-Codd Normal Form

- A relation R is in BCNF if and only if: whenever there is a nontrivial FD $A_1,A_2,\ldots,A_n \rightarrow B_1,B_2,\ldots,B_m$ for R, it is the case that $\{A_1,A_2,\ldots,A_n\}$ is a superkey for R.

- The following relation is not in BCNF

title	*year*	*length*	*genre*	*studioName*	*starName*
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

- Consider the FD, title year → length genre studioName
- Left side is not a superkey because title and year do not functionally determine starName.
- However, Movies2 from previous slide is in BCNF

BCNF Decomposition Algorithm

INPUT: A relation R with a set of FDs.

OUTPUT: A decomposition of R into a collection of relations into BCNF using “lossless join”.

1. Compute keys for R
2. Repeat the following steps till all relations are in BCNF
   a) Pick any R_0 with A->B that violates BCNF
   b) Decompose into R_1(A, B) and R_2(A, rest)
   c) Compute FDs for R_1 and R_2
   d) Compute keys for R_1 and R_2

BCNF Decomposition Example

Relation R {title , year , studioName , president , presAddr}
FDs are

1. title, year → studioName
2. studioName → president
3. president → presAddr

1. Compute keys - Using closure, {title, year} is the only key.

2. Repeat the following steps till all relations are in BCNF
   a) Relation R is a BCNF violation
   b) Decompose into the following based on FD 2 (studioName → president)
      S {title , year , studioName}
      T {studioName , president , presAddr}
   c) FDs are title, year → studioName, studioName → president, presAddr
   d) Keys are title, year and studioName

2. Decomposing again, here are the three relations:
   a) Relation T is a BCNF violation
   b) Decompose into the following
      S {title , year , studioName}
      T {studioName , president}
      U {president , presAddr}
   c) FDs are title, year → studioName, studioName → president, president → presAddr
   d) Keys are title, year for S, studioName for T and president for U

BCNF Exercise

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 $AB \to C$, $C \to D$, and $D \to A$.

b) $R(A, B, C, D)$ with FD's $B \to C$ and $B \to D$.

c) $R(A, B, C, D)$ with FD's $AB \to C$, $BC \to D$, $CD \to A$, and $AD \to B$.

d) $R(A, B, C, D)$ with FD's $A \to B$, $B \to C$, $C \to D$, and $D \to A$.

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.