Database Normalization

Discussion Sessions 1A and 1B

Session 3

Functional Dependencies

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 $A_1, A_2, \ldots, A_n$, then they must also agree on all of another list of attributes $B_1, B_2, \ldots, B_m$."

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

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	Vivian Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

Possible functional dependencies:

`title year → length genre studioName` ☑

`title year → starName` ✗

Keys of Relations

A set of one or more attributes $\{A_1, A_2, \ldots, A_n\}$ is a **key** of R if:

1. Those attributes functionally determine all other attributes in R.
2. No proper subset of $\{A_1, A_2, \ldots, A_n\}$ functionally determines all other attributes of R (i.e. it is **minimal**).

What is the key of our previous table?

`title year starName`

Superkeys

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

```
title year starName length
```

Trivial Functional Dependencies

- A constraint is said to be **trivial** if it holds for every instance of the relation, regardless of what other constraints are assumed.
- Trivial *FDs* are those $A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_m$ such that

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

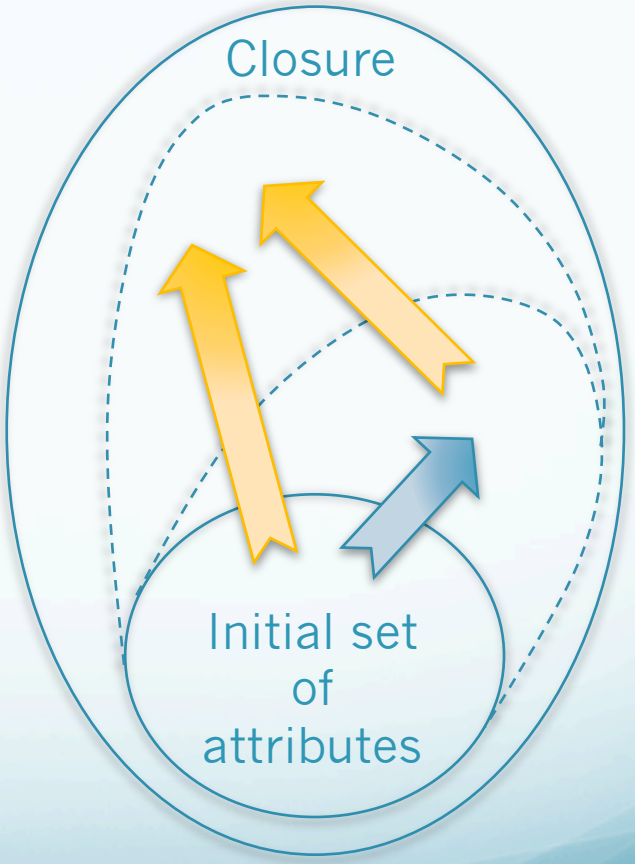
```
title year → title

title → title
```

Closure of Attributes

The **closure** of a set of attributes $\{A_1, A_2, \ldots, A_n\}$ under the *FDs* in S is the set of attributes B, such that $A_1, A_2, \ldots, A_n \rightarrow B$ follows from S.
We denoted it by

$$\{A_1, A_2, \ldots, A_n\}^+$$



Closure of Attributes – The Algorithm

Input: A set of attributes $\{A_1, A_2, \ldots, A_n\}$ and a set of *FDs* S.
Output: The closure $\{A_1, A_2, \ldots, A_n\}^+$

1. If necessary, split the *FDs* of S, so each *FD* 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 of $B_1, B_2, \ldots, B_m$ are in **X**, but C is not. Add C to the **X**, and continue the search.

Armstrong's Axioms

- **Reflexivity**: If $\{B_1, B_2, \ldots, B_m\} \subseteq \{A_1, A_2, \ldots, A_n\}$, then

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

- **Augmentation**: If $A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_m$, then

  $$A_1, A_2, \ldots, A_n, C_1, C_2, \ldots, C_k \rightarrow B_1, B_2, \ldots, B_m, C_1, C_2, \ldots, C_k$$

  for any set of attributes $C_1, C_2, \ldots, C_k$

- **Transitivity**: If $A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_m$ and $B_1, B_2, \ldots, B_m \rightarrow C_1, C_2, \ldots, C_k$, then

  $$A_1, A_2, \ldots, A_n \rightarrow C_1, C_2, \ldots, C_k$$

Exercise

Consider a relation with schema $R(A, B, C, D)$ and *FDs*

$AB \rightarrow C$
$C \rightarrow D$
$D \rightarrow A$

1. What are all the nontrivial *FDs* that follow from the given *FDs*? Restrict yourself to *FDs* with single attributes on the right side.
2. What are all the keys of R?
3. What are all the **superkeys** for R that **are not keys**?

Anomalies and Decomposing Relations

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	Vivian Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

- Any problems with this schema?
  - Redundancy – Update anomalies – Deletion anomalies

Boyce-Codd Normal Form

- A relation *R* is in *Boyce-Codd Normal Form* (**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*.

- In other words, the left side of every nontrivial *FD* **must be a superkey**.
- Our previous relation **is not** in BCNF, since, for the *FD* `title year → length genre studioName`, the set of attributes `{title year}` **is not a superkey**.

BCNF Decomposition Algorithm

Input: A relation R_0 and a set of *FDs* S_0.
Output: A decomposition of R_0 into a collection of relations, all of which are in BCNF.
Method: These steps can be applied recursively to any relation R and *FDs* S. Initially, $R = R_0$ and $S = S_0$.

1. Check whether R is in BCNF. If so, return $\{R\}$ as the answer.
2. Let $X \rightarrow Y$ be one BCNF violation *FD*. Compute X^+. Choose $R_1 = X^+$ and $R_2 = X \cup \{R - X^+\}$
3. Find the sets of *FDs* that hold on R_1 and R_2; let these be S_1 and S_2 respectively.
4. Recursively decompose R_1 and R_2 from **step 1**. *Return the union of the results of these decompositions*.

Exercise

Consider a relation with the following schema

```
{title, year, length, genre, studioName, president,
                                presAddr}
```

Three FDs that we will assume in this relations are

```
title year → length genre studioName
studioName → president
 president → presAddr
```

Decompose this relation, so that the resulting schemas will be in BCNF.

Multivalued Dependencies

A "**multivalued dependency**" is an assertion that two or more attributes or sets of attributes are *independent of one another*.

name	*street*	*city*	*title*	*year*
C. Fisher	123 Maple St.	Hollywood	Star Wars	1977
C. Fisher	5 Locust Ln.	Malibu	Star Wars	1977
C. Fisher	123 Maple St.	Hollywood	Empire Strikes Back	1980
C. Fisher	5 Locust Ln.	Malibu	Empire Strikes Back	1980
C. Fisher	123 Maple St.	Hollywood	Return of the Jedi	1983
C. Fisher	5 Locust Ln.	Malibu	Return of the Jedi	1983

What is the **key** of this table? Is this schema in **BCNF**?

Multivalued Dependencies – Definition

We say a **Multivalued Dependency** (*MVD*)

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

holds in a relation *R* if for each pair of tuples **t** and **u** that agree on all the *A*s, we can find some tuple **v** that agrees:

1. With both **t** and **u** on the *A*s,
2. With **t** on the *B*s, and
3. With **u** on all attributes of *R* that are not *A*s or *B*s.

In other words, for any fixed *A*s, the associated *B*s and all other attributes in *R* appear in all possible combinations in different tuples.

Fourth Normal Form

A relation *R* is in *fourth normal form* (**4NF**) if whenever

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

is a *nontrivial MVD*, $\{A_1, A_2, \ldots, A_n \twoheadrightarrow B_1, B_2, \ldots, B_m\}$ is a **superkey**

So, in our previous table, we have an *MVD*

```
name ↠ street city
```

which is nontrivial. Also, `name` **is not** a superkey of *R*, then, the relation **is not in 4NF**.

4NF Decomposition Algorithm

Input: A relation R_0 and a set of *FDs* and *MVDs* S_0.
Output: A decomposition of R_0 into a collection of relations, all of which are in 4NF.
Method: Do the following steps with $R = R_0$ and $S = S_0$.

1. Find a 4NF violation in R, say $A_1, A_2,..., A_n \twoheadrightarrow B_1, B_2,..., B_m$, where $\{A_1, A_2,..., A_n\}$ is not a superkey. If there is none, return; R by itself is a suitable decomposition.
2. If there is 4NF violation, break R into two schemas:
   a) R_1, whose schema contains the As and Bs.
   b) R_2, whose schema contains the As and all other attributes that are not As or Bs.
3. Find the *MVD* that hold in R_1 and R_2. Recursively decompose R_1 and R_2 with respect to their own dependencies.

Exercise

Consider a relation with the following schema

$$\{name, street, city, title, year\}$$

Where a *MVD* holds

$$name \twoheadrightarrow street\ city$$

Decompose this relation, so that the resulting schemas will be in 4NF