Schema Refinement

Problem: Redundancy

Replicated data + change = Trouble.

- Leads to wasted storage
- Insert/delete/update anomalies

Solution: Functional Dependencies + Decomposition

Functional Dependencies are a form of integrity constraints that help identify redundancy in schemas and help refine the database

Decompose or split a table into two tables in a way that eliminates duplicates but does not lose any of the information and preserves the integrity constraints

Functional Dependency

$\mathbb{A} \longrightarrow \mathbb{B}$

$\{A_1, \dots, A_n\} \longrightarrow \{B_1, \dots, B_m\}$

Given any two tuples, t_1, t_2 in table R with attribute sets $\mathbb{A}, \mathbb{B}$

if their $\mathbb{A}$ values are the same, then their $\mathbb{B}$ values must be the same.

$$\pi_{\mathbb{A}} t_1 = \pi_{\mathbb{A}} t_2 \implies \pi_{\mathbb{B}} t_1 = \pi_{\mathbb{B}} t_2$$

	model	year	color	price	mileage
t_1	Ford Fission	2010	blue	20000	25
t_2	Ford Fission	2010	red	21000	25
t_3	Ford Fission	2020	blue	30000	30
t_4	Ford Passion	2000	purple	40000	25

$\mathbb{A} \coloneqq \{model, year\}; \mathbb{B} \coloneqq \{mileage\}$ $\quad$ $\mathbb{A} \longrightarrow \mathbb{B}$

$\{model, year\} \longrightarrow \{mileage\}$

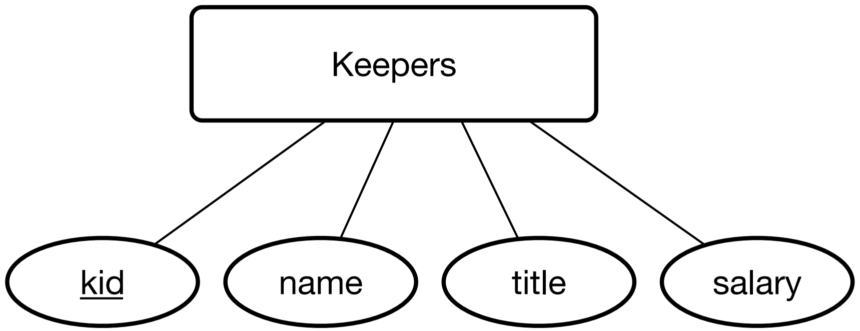
$\pi_{\mathbb{A}} t_1 = \pi_{\mathbb{A}} t_2$ means $\pi_{\mathbb{B}} t_1 = \pi_{\mathbb{B}} t_2$: same mileage

$\pi_{\mathbb{A}} t_1 \neq \pi_{\mathbb{A}} t_3$ says nothing about $\pi_{\mathbb{B}} t_1, \pi_{\mathbb{B}} t_3$: the mileage could be different or the same

$\pi_{\mathbb{B}} t_1 = \pi_{\mathbb{B}} t_4$ says nothing about $\pi_{\mathbb{A}} t_1, \pi_{\mathbb{A}} t_4$: the FD says nothing about model and year when mileage is different

Where do FDs come from?

- *Hold true over all allowable instances* not just ones that currently exist in the database
- Come from application semantics.
- Not learned from data, but you might learn suggestions for FDs
- Help us think about redundancies and their anomalies



The ER model doesn't capture FDs!

$$kid \longrightarrow \{name, title, salary\}$$

$$title \longrightarrow salary$$

Update Anomalies

$title \longrightarrow salary$

kid	name	title	salary
872	Azza Abouzied	senior	5,000
452	Hazem Ibrahim	junior	3,000
672	Miro Mannino	junior	3,000
981	Benjamin Mee	senior	5,000
666	Joe Exotic	junior	3,000
321	Jane Goodall	chief	10,000

Can we update Miro's salary?

No, it will be inconsistent with Hazem's and Joe's salaries who are also "junior" keepers

Deletion Anomalies

$title \longrightarrow salary$

kid	name	title	salary
872	Azza Abouzied	senior	5,000
452	Hazem Ibrahim	junior	3,000
672	Miro Mannino	junior	3,000
981	Benjamin Mee	senior	5,000
666	Joe Exotic	junior	3,000
321	Jane Goodall	chief	10,000

Can we delete Jane Goodall?

We will lose all information on what the salary is for chief keepers!

Insertion Anomalies

$title \longrightarrow salary$

kid	name	title	salary
872	Azza Abouzied	senior	5,000
452	Hazem Ibrahim	junior	3,000
672	Miro Mannino	junior	3,000
981	Benjamin Mee	senior	5,000
666	Joe Exotic	junior	3,000
321	Jane Goodall	chief	10,000
209	Ian Malcolm	intern	?

Can we insert a keeper with a title for which we don't know the salary?

Then you might invent a value without reference to the true rule!

Why are some functional dependencies problematic?

$title \longrightarrow salary$ 

$kid \longrightarrow salary$ 

kid	name	title	salary
872	Azza Abouzied	senior	5,000
452	Hazem Ibrahim	junior	3,000
672	Miro Mannino	junior	3,000
981	Benjamin Mee	senior	5,000
666	Joe Exotic	junior	3,000
321	Jane Goodall	chief	10,000

title is not a key so pairs of (title, salary) e.g. (senior, 5000) appear many times

kid is a key, so each pair of (kid, salary) e.g. (872, 5000) appears exactly once

kid	name	title	salary
872	Azza Abouzied	senior	5,000
452	Hazem Ibrahim	junior	3,000
672	Miro Mannino	junior	3,000
981	Benjamin Mee	senior	5,000
666	Joe Exotic	junior	3,000
321	Jane Goodall	chief	10,000

kid	name	title
872	Azza Abouzied	senior
452	Hazem Ibrahim	junior
672	Miro Mannino	junior
981	Benjamin Mee	senior
666	Joe Exotic	junior
321	Jane Goodall	chief

title	salary
senior	5,000
junior	3,000
intern	1,000
chief	10,000

Eliminate Redundancy by decomposing the relation along the problematic FDs!

Armstrong's Axioms

Trivial

$$A_1 \rightarrow A_1$$

$$A_1, A_2, \ldots, A_k \rightarrow A_i$$

An attribute determines itself;
A set of attributes determine any one of the attributes in the set

$$\text{model}, \text{year}, \text{color} \longrightarrow \text{year}$$

Split & Combine

Split →

$$A_1, A_2, \ldots, A_n \longrightarrow B_1, B_2, \ldots, B_k \iff \begin{bmatrix} A_1, A_2, \ldots A_n \longrightarrow B_1 \\ A_1, A_2, \ldots A_n \longrightarrow B_2 \\ \ldots \\ A_1, A_2, \ldots A_n \longrightarrow B_k \end{bmatrix}$$

← Combine

- If a set of attributes $\mathbb{A}$ determines a set $\mathbb{B}$, then $\mathbb{A}$ also determines every attribute within $\mathbb{B}$.
- If a set of attributes $\mathbb{A}$ determines sets $\mathbb{B}$ and $\mathbb{C}$, then it also determines their union $\mathbb{B} \cup \mathbb{C}$

$$\text{model, year, color} \longrightarrow \text{price, mileage} \iff \begin{bmatrix} \text{model, year, color} \longrightarrow \text{price} \\ \text{model, year, color} \longrightarrow \text{mileage} \end{bmatrix}$$

Transitive

$$\begin{matrix} A_1, A_2, \dots, A_n \longrightarrow B_1, B_2, \dots, B_m \\ B_1, B_2, \dots, B_m \longrightarrow C_1, C_2, \dots, C_p \end{matrix} \Longrightarrow A_1, A_2, \dots, A_n \longrightarrow C_1, C_2, \dots, C_p$$

If a set of attributes $\mathbb{A}$ determines a set $\mathbb{B}$,
and $\mathbb{B}$ determines a set $\mathbb{C}$,
then $\mathbb{A}$ determines $\mathbb{C}$

$$\begin{matrix} \text{model, year, color} \longrightarrow \text{mileage} \\ \text{mileage} \longrightarrow \text{tax} \end{matrix} \Longrightarrow \text{model, year, color} \longrightarrow \text{tax}$$

Given this:

F_1: $\text{model, color, year} \rightarrow \text{price}$

F_2: $\text{model, year} \rightarrow \text{mileage}$

F_3: $\text{mileage} \rightarrow \text{tax}$

Can you derive this? $model, color, year \rightarrow price, mileage, tax$

F_4: $model, year, color \rightarrow mileage, color$

Trivial F_2 $A_1 \longrightarrow A_1$

F_5: $model, year, color \rightarrow mileage$

Split F_4 $A_1, \ldots, A_n \longrightarrow B_1, B_2 \Longleftrightarrow \begin{array}{l} A_1, \ldots, A_n \longrightarrow B_1 \\ A_1, \ldots, A_n \longrightarrow B_2 \end{array}$

F_6: $model, year, color \rightarrow tax$

Transitivity F_3 F_5 $A_1, \ldots, A_n \longrightarrow B_1, B_2 \Longleftrightarrow \begin{array}{l} A_1, \ldots, A_n \longrightarrow B_1 \\ A_1, \ldots, A_n \longrightarrow B_2 \end{array}$

$model, color, year \rightarrow model, year, price$

Combine F_1 F_5 F_6 $_1, \ldots, A_n \longrightarrow B_1, B_2 \Longleftrightarrow \begin{array}{l} A_1, \ldots, A_n \longrightarrow B_1 \\ A_1, \ldots, A_n \longrightarrow B_2 \end{array}$

Given this:

F_1: $\text{model}, \text{color}, \text{year} \rightarrow \text{price}$

F_2: $\text{model}, \text{year} \rightarrow \text{mileage}$

F_3: $\text{mileage} \rightarrow \text{tax}$

Can you derive this?

$$color, tax \rightarrow price$$

Closures & Keys

Attribute Closure

Given a set of attributes $A_1, A_2, \ldots, A_n$

The *closure* $\{A_1, A_2, \ldots, A_n\}^+ := \{B_1, \ldots, B_m\} \mid A_1, A_2, \ldots, A_n \longrightarrow B_i$

Given this:

F_1: $\text{model, color, year} \rightarrow \text{price}$

F_2: $\text{model, year} \rightarrow \text{mileage}$

F_3: $\text{mileage} \rightarrow \text{tax}$

What is? $\{model, color, year\}^+$

$\{model, color, year, \ldots\}$	Trivial
$\{model, color, year, price, \ldots\}$	By F_1
$\{model, color, year, price, mileage, \ldots\}$	By F_2
$\{model, color, year, price, mileage, tax\}$	By Transitivity F_3

With closures, we can easily verify a functional dependency.

To check if $\mathbb{A} \longrightarrow \mathbb{B}$

Compute $\mathbb{A}^+$

Check if $\mathbb{B} \subset \mathbb{A}^+$

So what?

Given this:

F_1: $\text{model, color, year} \rightarrow \text{price}$

F_2: $\text{model, year} \rightarrow \text{mileage}$

F_3: $\text{mileage} \rightarrow \text{tax}$

Can you derive this?

$color, tax \rightarrow price$

What is? $\{color, tax\}^+$

$\{color, tax\}$ Trivial

And that's it!

Since $price \notin \{color, tax\}^+$

$\{color, tax\} \nrightarrow price$

Keys

Superkey	Any set of attributes that functionally determine all attributes in a relation. $\{A_1, \ldots A_k\}^+ = R$
Candidate Key	A superkey for which no strict subset is a superkey! A minimal superkey $\mathbb{A}^+$ **is a candidate key iff** $\forall \mathbb{A}' \subset \mathbb{A}, \mathbb{A}'^+ \neq R$
Primary Key	A candidate key for the relation.

Normalization

Boyce-Codd Normal Form

Decomposition & Normal Forms

Decompose (defn): replace R by two or more relations $R_1, \ldots, R_n$ such that:

- $Attr(R_i) \subseteq Attr(R)$
- $\bigcup_i Attr(R_i) = Attr(R)$

1 *What is a good decomposition?*

GOALS

Is it Lossless?

Does it Eliminates Anomalies?

Is it Dependency Preserving?

2 *When to stop decomposing?*

NORMAL FORMS

If a relation is a normal form, we know it avoids certain/reduces certain problems

e.g. *BCNF* ensures a lossless decomposition that eliminates redundancy

3 *How to decompose?*

$kid \longrightarrow \{name, title, salary\},\ title \longrightarrow salary$

Keepers

kid	name	title	salary
872	Azza Abouzied	senior	5,000
452	Hazem Ibrahim	junior	3,000
672	Miro Mannino	junior	3,000
981	Benjamin Mee	senior	5,000
666	Joe Exotic	junior	3,000
321	Jane Goodall	chief	10,000

Keepers’

kid	name	title
872	Azza Abouzied	senior
452	Hazem Ibrahim	junior
672	Miro Mannino	junior
981	Benjamin Mee	senior
666	Joe Exotic	junior
321	Jane Goodall	chief

$kid \longrightarrow \{name, title\}$

Salary

title	salary
senior	5,000
junior	3,000
intern	1,000
chief	10,000

$title \longrightarrow salary$

1 *Is this a good decomposition?*

GOALS

Is it Lossless?

- *Yes!* Keepers = Keepers’ ⋈ Salary

Does it Eliminate Anomalies?

- *Yes!* No redundancies

Is it Dependency Preserving?

- *Yes!*

$$\left(F_{\text{Keepers}'} \cup F_{\text{Salary}}\right)^{+} = F^{+}_{\text{Keepers}}$$

$A \rightarrow B$
$C \rightarrow B$

A	B	C
1	2	3
4	5	6
7	2	9

Lossy decomposition …

A	B
1	2
4	5
7	2

$A \rightarrow B$

$\bowtie$

B	C
2	3
5	6
2	9

$C \rightarrow B$

=

A	B	C
1	2	3
1	2	9
4	5	6
7	2	3
7	2	9

… but dependency preserving $[(A \rightarrow B) \cup (C \rightarrow B)]^{+} = [A \rightarrow B, C \rightarrow B]^{+}$

Lossless decomposition …

A	C
1	3
4	6
7	9

$\bowtie$

B	C
2	3
5	6
2	9

$C \rightarrow B$

=

A	B	C
1	2	3
4	5	6
7	2	9

… but not dependency preserving $[C \rightarrow B]^{+} \neq [A \rightarrow B, C \rightarrow B]^{+}$

Boyce-Codd Normal Form (BCNF)

1 *Is it a good form?*

GOALS

Is Lossless

Eliminates Anomalies

But it may not always be dependency preserving

2 *When to stop decomposing?*

Is R in BCNF?

A relation R is in BCNF if $\{A_1, \ldots, A_n\} \longrightarrow B$ is a non-trivial dependency in R (i.e. $\mathrm{B} \neq \mathrm{A_i}$), then $\{A_1, \ldots, A_n\}$ is a superkey for R

Another way to think of it:
For all sets of attributes $\mathbb{A}$ of R, either $\mathbb{A} = \mathbb{A}^+$ or $\mathbb{A}^+$ = $\{\text{all attributes of } R\}$

BCNF = no "problematic" FDs

Boyce-Codd Normal Form (BCNF)

3 *How to decompose?*

The LHS of a "bad" FD that is not a superkey of R

```
BCNFy(R):
    find X s.t. X ≠ X+ ≠ [all attributes]
    if (not found) then
        R is in BCNF
    else
        Let Y = X+ - X
        Let Z = [all attributes] - X+
        Let R_1 = (X ∪ Y)
        Let R_2 = (X ∪ Z)
        BCNFy(R_1)
        BCNFy(R_2)
```

Decompose along the "bad" FD

Recursively decompose

BCNF Example Decomposition

R (id, name, major, sig, dues)

Functional dependencies in R

- F_1 id, sig $\rightarrow$ id, name, major, sig, dues
- F_2 id $\rightarrow$ name, major
- F_3 sig $\rightarrow$ dues