week 4

Database Design

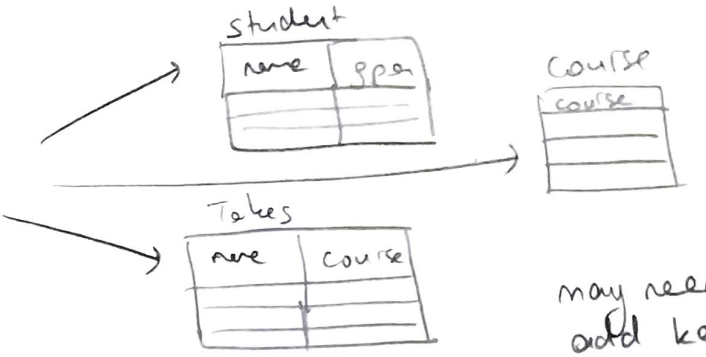
First Normal Form (1NF)

In 1NF, tables are flat

Student

name	GPA	Courses
A	3	
B	3.5	

This is not 1NF



may need to add keys

Data Anomalies

Redundancy → repeating data

Updated an. → need to change several places

Delete an. → lose info we don't want

ex:

name	ssn	phone	city
f	1	P1	S
f	1	P2	S
j	2	P3	C

→ one person can have multiple phones but lives in 1 city

redundancy → f, 1, S

update → F moves to London → need to update 2 rows

delete → F deletes phone. What city is he in?

Soln:

name	ssn	city

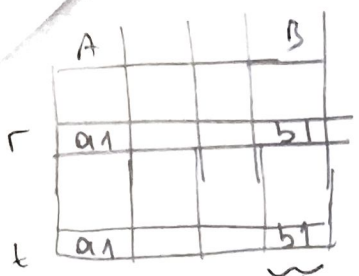
ssn	phone

Functional Dependencies

- A form of constraints

$A \rightarrow B$

- If in your table, 2 rows agree on A column, then they must agree on B.



$A \to B$

t-r agree on A $\Rightarrow$ they agree on B.

ex:

id	name	Phone	Pos
1	S	1234	C
2	M1	9876	S
3	S	9876	S
4	M2	1234	C

FD holds
- id → name, phone, pos
- pos → phone

FD does not hold
- phone → pos

1 violating row is enough for saying it is not F.D.

ex1:

name	cat	color	dept	pri
g	31	gr	toy	49
t	31	gr	toy	39

name → color satisfies
cat → dept satisfies
Color, cat → price **does not**
↳ gr, 31 → 49 ≠
gr, 31 → 39

ex2:

name	cat	color	dept	price
g	31	gr	toy	49
t	31	bl	toy	89
g	51	gr	office	59

name → color ✓
cat → dept ✓
Color, cat → price ✓

Observation:
if name → color, cat → dept, Color, cat → price (F.D ✓) then name, cat → price
Why? → Later

Goal: Find all FD, look for bad ones.

...strong's Rules

Decomposition	$A \to BC$	then	$A \to B$, $A \to C$
Union	$A \to B$	and $A \to C$	then	$A \to BC$
Transivity	$A \to B$	and $B \to C$	then	$A \to C$
Reflexivity	$Y \subseteq X$	then	$X \to Y$	
Augmentation	$X \to Y$	then	$XZ \to YZ$	
Pseudotransitivity	$X \to Y$	and $WY \to Z$	then	$WX \to Z$

1) Splitting and Combine

$A_1, A_2 \to \dots \to B_1, B_2 \dots B_n$

then

$A_1, A_2 \dots A_n \to B_1$,

$A_1, A_2 \dots A_n \to B_2$

$A_1 \dots A_n \to B_n$

2) Trivial

$A_1, A_2 \dots A_n \to A_i$ where $i = 1, 2 \dots n$

name, cat → name

3) Transivitive

ex: Given
1. name → color
2. cat → dept
3. color, cat → price

4. name, cat → name	trivial on 4	
5. name, cat → color	transivity (4,1)	name, cat → name name → color name, cat → color
6. name, cat → cat	trivial 6	
7. name, cat → color, cat	union 5,6	name, cat → color name, cat → cat ∪ name, cat → color, cat
8. name, cat → price	transivity (7,3)	name, cat → color, cat color, cat → price name, cat → price

Hard to list all of them.

Attribute Closures

Given { name → color; cat → dept; color, dept → price }

$(name)^+ = \{ name, color_{name \to color} \}$

$\Rightarrow (name, cat)^+ = \{ name, cat, color_{A \to C}, dept_{B \to D}, price_{color, dept \to price} \}$

$(color)^+ = \{ color \}$

Hence name, cat → color, dept, price

---

ex: R(A, B, C, D, E, F)

Given { A, B → C; A, D → E; B → D; A, F → B }

$(A,B)^+ = ?$

Soln: $(A,B)^+ = \{ A, B, C_{AB \to C}, D_{B \to D}, E_{AD \to E} \}$

$(A,F)^+ = ?$

Soln: $(A,F)^+ = \{ A, F, B_{AF \to B}, C_{AB \to C}, D_{B \to D}, E_{AD \to E} \}$

AF → BCDE (AF) together is a key!

---

ssn	name	addr	hobby
1	J	a	hike
1	J	a	bike

→ redundancy → change in address, more than 1 row

→ delete → suppose a person gives up all hobbies, set hobby = NULL → no, hobby is part of key

delete row → no (lose info)

↳ insert → cannot insert a person w.o. hobby

Decomposition

Person1 (ssn, name, addr)

Hobbies (ssn, hobby)

Suppose we have add → zip

How can we deal with this insertion

We can use assertions

addr		zip
addr		06800
addr		06800
addr		06800

A. FD as

Check **Not exists**

```
Select *
F   Person P1, Person P2
W   P1.add = P2.add and
    P1.zip != P2.zip
```

⇒ It will do after insert.

We can use before / after and new row.

However, it works for only 1 insertion

How about multiple insertion?

Add **this**

---

ex: Suppose we are given the table before)

Is this a F.D ⟹ add → zip.

How can I find this?

Soln: For each addr, group by addr. If there is 2 in the count of the zipcode, it is not FD.

```
Sel  —
—  —
group by addr
having  count(distinct zipcode)
```

Why do we need Attribute Closures?

- We are interested in
  is $X \to A$ implied by the given set of F.D.

Soln: 1st compute X^+
2nd check if A in X^+

ex: $A,B \to C$
$A,D \to B$
$B \to D$

A^+ {A}
B^+ {B, D}
C^+ {C}
D^+ {D}
AD^+ {A, B, C, D}
AC^+ {A, C}
AD^+ {A, D, B, C}

4 attributes
$2^4 - 1$ F.D

$S \to M$
$M, C \to R$
$C \to T$

$\Rightarrow$ $S^+ = \{S, M\}$

$SC^+ = \{S, C, \underset{S \to M}{M}, \underset{C \to T}{T}, \underset{MC \to R}{R}\}$

Superkey → set of attributes $A_1, A_2 \dots A_n \to B$

key → min of superkey

---

ex2: Product (name, price, cat, col)

name, cat → price
cat → color

$(name, cat)^+ = \{name, cat, price, color\}$ (cat → color) → Superkey

$(name)^+ = \{name\}$
$(cat)^+ = \{cat, color\}$ → since they dont cover all attributes, they are not keys.

(name, cat) is key

★ in ex1, we have 5 attributes. 2^5 work, too much time consuming.

Instead, we can look at L.h.s.

$S \to M$
$M, C \to R$
$C \to T$

$S^+ = \{S, M\}$ no key
$M^+ = \{M\}$ //
$C^+ = \{C, T\}$ //
$Sm^+ = \{S, M\}$ //
$SC^+ = \{S, C, M, R, T\}$ ✓

ex: Enrollment (Student, Address, Course, Room, Time)

$S \to A$
$R, T \to C$
$S, C \to R, T$

Find keys

Sol:

$S^+ = S, A$ ×
$R^+ = R$ ×
$T^+ = T$ ×
$C^+ = C$ ×
$SC^+ = S, C, A, R, T$ ✓ (SCART)
$RTS^+ = R, T, S, C, A$ ✓

Which one do we pick?
Depends on the design

either C. T. SC → PK
RTS → unique

or C. T. RTS → PK
SC → unique

minating Anomalies

$X \to A$ is ok F.D if X is (super) key

$X \to A$ is not OK otherwise

ex:

name	ssn	phone	city

ssn → name, city

What is the key?

Soln: Key is (ssn, phone)

Hence, ssn → name, city is bad F.D

ssn	phone

FD = { }

(ssn, phone) is key

no redundancy ✓

ssn	name	city

FD { ssn → name, city }

(ssn) is key

no bad dependency ✓

Key or Keys

Given R(A,B,C) define F.D. st there are 2 or more keys.

Soln: $A \to BC$, $B \to AC$

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

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

A and B are keys

3 keys

Soln: $A \to B$, $B \to C$, $C \to A$

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

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

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

A, B and C are keys

e-Codd Normal Form (BCNF)

$\forall X$, either $X^+ = X$ or $X^+ =$ all attributes

ex: $R(A,B,C)$

$A \to BC$
$B \to AC$

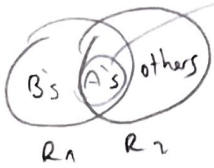
$\Rightarrow$ $A^+ = A,B,C$ ✓
$B^+ = B,A,C$
$C^+ = C$

ex: $R(A,B,C)$

$AB \to C$
$BC \to A$

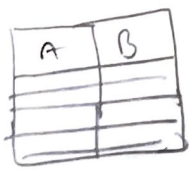
$\Rightarrow$ $A^+ = A$ $\quad AB^+ = ABC$
$B^+ = B$ $\quad BC^+ = BCA$ ✓
$C^+ = C$

→ we need A for joining R_1 and R_2



★ Is there 2 attr. relation that is not BCNF?

-No



Suppose $A \to B$ then $A^+ = A,B$ ✓
suppose $B \to A$ then $B^+ = B,A$ ✓
suppose AB ✓
suppose $\{\}$ // ✓

ex:

name	ssn	phone	city

$ssn \to name, city$

$(ssn, phone)^+ = \{ssn, name, city, phone\}$
$(ssn)^+ = (ssn, name, city)$ X not all attributes
$(phone)^+ = (phone)^*$ X

What is key → $(ssn, phone)$

ssn	name	city

$ssn \to name, city$ ✓
$ssn^+ = (ssn, name, city)$ ✓

ssn	phone

$\{\}$ ✓

Person (name, ssn, age, hair, phone)

ssn → name, age

age → hair

Decompose!

Soln: Identify the key

$(ssn, age)^+ = \{ssn, age, name, hair\}$ phone?

$(ssn, phone)^+ = \{ssn, name, age, hair, phone\}$ ✓

ssn → name
ssn → age
age → either

Both ssn → name, age and age → hair are bad F.D.

1. go! Pick ssn → name age

Person → P1 (ssn, name, age)

ssn → name, age

F.D ✓

Person → P2 (ssn, phone, hair)

geriberlen att + ssn

ssn → hair.

$ssn^+ = \{ssn, hair\}$ phone is missing

P21 (ssn, hair) ✓

P22 (ssn, phone) ✓

ssn → name, age ✓	It is not a mistake
ssn → haircolor ?	It is ok
where is age → hair ?	

2. go! Pick age → hair

Person → P1 (age, hair) ✓

Person → P2 (age, ssn, name, phone)

$ssn^+ \rightarrow \{name, age\}$ phone?

P21 (ssn, name, age) ✓

P22 (ssn, phone) ✓

ssn → name, age ✓

age → hair ✓

1st table, what is the hair color of this age → 2 joins
In 2nd table, — — From P1.

Design depends on the queries.

Q: If I have a big table and only 1 bad FD, should I decompose?

A: If you dont have many update, delete, insert, then you can live with that.

ex:
Student (sid, name, addr, dept)
Transkript (sid, cid, grade)

If I have this query too many times → what is grades of Ayse?
Then this table gives faster results (redundant info but faster)
Transcript (sid, sname, cid, grade)