NORMAL FORMS

GOOD & BAD DESIGN

BAD

A	B	C	D	E	F
Title	**Director**	**Theatre**	**Address**	**Time**	**Price**
Inferno	Ron Howard	Vue	Omni Centre	20:00	11.50
Inferno	Ron Howard	Vue	Omni Centre	22:30	10.50
Inferno	Ron Howard	Odeon	Lothian Rd	20:00	10.00
Inferno	Ron Howard	Cineworld	Fountain Park	18:20	9.50
Inferno	Ron Howard	Cineworld	Fountain Park	21:00	11.00
Trolls	Mike Mitchell	Vue	Omni Centre	16:10	9.50
Trolls	Mike Mitchell	Vue	Omni Centre	19:30	10.00
Trolls	Mike Mitchell	Odeon	Lothian Rd	15:00	8.50
Trolls	Mike Mitchell	Cineworld	Fountain Park	17:15	9.00

{ Title → Director ; Theatre, Title, Time → Price ; Theatre → Address }

Why is this bad design?

1. Redundancy
   - → Many facts are repeated
     i.e. For every showing, we list both director and title
2. Update anomalies
   - → If the address is changed, you have to update for all movies and showtimes
   - → If a movie stops playing, the association title – director is lost

LHS is the key

Movies: $\boxed{\text{Title}}$ → Director

Title	Director
Inferno	Ron Howard
Trolls	Mike Mitchell

Theatres: $\boxed{\text{Theatre}}$ → Address

Theatre	Address
Vue	Omni Centre
Odeon	Lothian Rd
Cineworld	Fountain Park

Showings: $\boxed{\text{Theatre, Title, Time}}$ → Price

Theatre	Title	Time	Price
Vue	Inferno	20:00	11.50
Vue	Inferno	22:30	10.50
Odeon	Inferno	20:00	10.00
Cineworld	Inferno	18:20	9.50
Cineworld	Inferno	21:00	11.00
Vue	Trolls	16:10	9.50
Vue	Trolls	19:30	10.00
Odeon	Trolls	15:00	8.50
Cineworld	Trolls	17:15	9.00

Why is this good design?

1. No redundancy
   - → Every FD defines a key
2. No information loss
   - → BAD = Movies ⋈ Theatres ⋈ Showings
3. No update anomalies
4. No constraints are lost
   - → All of the original FDs appear as constraints in the new tables

↳ Problems w/ bad designs are caused by FDs X → Y, where X is not a key

Causes redundancy

BOYCE-CODD NORMAL FORM (BCNF)

→ A relation w/ FDs F is in BCNF if for every $X \to Y$ in F

  (a). $Y \subseteq X$ (the FD is trivial)

  OR

  (b). X is a key

→ A database is in BCNF if all relations are in BCNF

i.e. Looking at prev. example, the relation BAD has three FDs and some FDs violate the requirement for BCNF. We can prove this by:

$$\Sigma = \{ A \to B, \quad CAE \to F, \quad C \to D \}$$

↳ Since these FDs are not trivial, we have to check if LHS is a key

↳ $C_\Sigma(A) = AB$ which is not the whole set of attributes, so A is not a key

↳ $C_\Sigma(CAE) = CAEFBD$ so 2nd FD does not violate BCNF, but one violation is enough to conclude the relation is not in BCNF

DECOMPOSITIONS

→ Given a set of attributes $\mathcal{U}$ and a set of FDs F, a decomposition of $\underbrace{(\mathcal{U}, F)}_{\text{Relation schema}}$ is a set $\underbrace{(\mathcal{U}_1, F_1), \ldots, (\mathcal{U}_n, F_n)}$ such tht. $\boxed{\mathcal{U} = \bigcup_{i=1}^{n} \mathcal{U}_i}$ and $\boxed{F_i \text{ is a set of FDs over } \mathcal{U}_i}$

They can't talk about attributes tht. are in another relation schema

- $(ABC, \emptyset)$ is a valid schema

→ **BCNF decomposition** is when each $(\mathcal{U}_i, F_i)$ is in BCNF

→ Criteria for good decompositions: → **Losslessness** — no info. is lost
  → **Dependency preservation** — no constraints are lost

↳ A decomposition of $(\mathcal{U}, F)$ into $(\mathcal{U}_1, F_1), \ldots, (\mathcal{U}_n, F_n)$ is

- Lossless if for every relation R over $\mathcal{U}$ tht. satisfies F,
  1) each $\pi_{\mathcal{U}_i}(R)$ satisfies F_i
  2) $R = \pi_{\mathcal{U}_1}(R) \bowtie \cdots \bowtie \pi_{\mathcal{U}_n}(R)$

- Dependency preserving if $\underbrace{F}$ and $\underbrace{\bigcup_{i=1}^{n} F_i}$ are equivalent

  that is, they have the same closure

PROJECTION OF FDs

→ let F be a set of FDs over attributes $\mathcal{U}$

The $\boxed{\text{projection}}$ of F on $V \subseteq \mathcal{U}$ is the set of all FDs over V that are implied by F

↳ $\pi_V(F) = \{ X \to Y \mid X, Y \subseteq V,\ \underbrace{Y \subseteq C_F(X)}_{\Updownarrow \ F \models X \to Y} \}$

BCNF DECOMPOSITION ALGORITHM

INPUT		OUTPUT
A set of attributes $\mathcal{U}$ and a set of FDs F	→	A database schema S

1 $S := \{(\mathcal{U}, F)\}$

2 While there is $(\mathcal{U}_i, F_i) \in S$ not in BCNF:

3 Replace $(\mathcal{U}_i, F_i)$ by decompose$(\mathcal{U}_i, F_i)$

4 Remove any $(\mathcal{U}_i, F_i)$ for which there is $(\mathcal{U}_j, F_j)$ w/ $\mathcal{U}_i \subseteq \mathcal{U}_j$

5 Return S

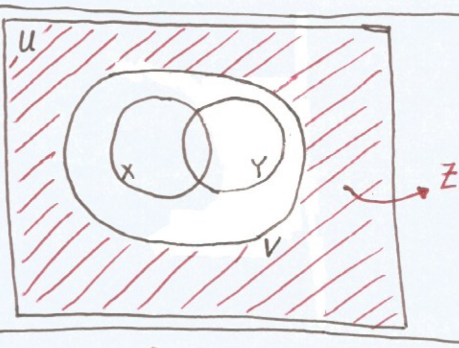
$\underset{\mathcal{U}_1}{V} \cap \underset{\mathcal{U}_2}{XZ} = X$

decompose$(\mathcal{U}, F)$:

1 Choose $(X \to Y) \in F$ that violates BCNF

2 Set $V := C_F(X)$ ($F \models X \to V$) and $Z := \mathcal{U} - V$

3 Return $(V, \pi_V(F))$ and $(XZ, \pi_{XZ}(F))$



→ these are lossless splits of the input tht we are decomposing

$$\begin{array}{ccc} & (\mathcal{U}, F) & \\ \swarrow & & \searrow \\ (\mathcal{U}_1, F_1) & & (\mathcal{U}_2, F_2) \end{array}$$

$\mathcal{U}_1 \cup \mathcal{U}_2 = \mathcal{U}$

For every legal R over $\mathcal{U}$: $R = \pi_{\mathcal{U}_1}(R) \bowtie \pi_{\mathcal{U}_2}(R)$

$\Longleftrightarrow$

$F \models \mathcal{U}_1 \cap \mathcal{U}_2 \to \mathcal{U}_1$ OR $F \models \mathcal{U}_1 \cap \mathcal{U}_2 \to \mathcal{U}_2$

↳ The common attributes between $\mathcal{U}_1$ and $\mathcal{U}_2$ determines all the attributes in one of the pieces

↳ This is exactly what the decompose$(\mathcal{U}, F)$ subprocedure does

- $V := C_F(X) \Leftrightarrow F \models X \to V$ and $V \cap XZ = X$, so $F \models V \cap XZ \to V$ ✓

PROPERTIES OF BCNF ALGO.

1) The decomposed schema is in BCNF and lossless-join

2) The output depends on the FDs chosen to decompose

3) Dependency preservation is not guaranteed → DRAWBACK!

EXAMPLE (1st page)

$\Sigma = \{ \boxed{A \to B}, CAE \to F, \boxed{C \to D} \}$

Not in BCNF

So our first split can use either FDs

$(ABCDEF, \Sigma)$

$A \to B$

$(AB, \{A \to B\})$

$C_\Sigma(A) = AB$

$C_{\{A \to B\}}(A) = AB$ ✓

$(ACDEF, \underbrace{\{CAE \to F, C \to D\}}_{\Sigma_2})$

$Z = \{ABCDEF\} - \{AB\} = \{CDEF\}$

then combine w/ LHS of FD which is A

$C_{\Sigma_2}(CAE) = CAEFD$ ✓ No violation

$C_{\Sigma_2}(C) = CD$ ✗ The FD $C \to D$ violates BCNF

Hence, this relation schema is not in BCNF & we need to decompose further

$C \to D$

$(CD, \{C \to D\})$

$C_{\Sigma_2}(C) = CD$

✓

$(ACEF, \{CAE \to F\})$

$Z = \{ACDEF\} - \{CD\} = \{AEF\}$

$\{AEF\} \cup \{C\} = \{ACEF\}$

✓

THIRD NORMAL FORM (3NF)

2

→ A weaker form of BCNF

→ (U, F) is in 3NF if for every FD $X \to Y$ in F, one of the ff. holds:

1) $Y \subseteq X$ (the FD is trivial)
2) X is a key
3) all of the attributes in Y are prime

NEW!

In 3NF, FDs where the LHS is not a key are allowed as long as the RHS consists only of prime attributes

→ Every schema in BCNF is also in 3NF (but not the other way round)

i.e. Take the relation Lectures: Class, Professor, Time

w/ FDs $F = \{C \to P,\ PT \to C\}$

(CPT, F) is not in BCNF bc. $(C \to P) \in F$ is non-trivial & C is not a key

If we decompose using BCNF algo. we get

$(CP,\ C \to P)$ and $(CT,\ \emptyset)$ → We lose the constraint $PT \to C$!

Important! A professor can't be in 2 diff. classes at the same time

(CPT, F) is in 3NF bc. PT is a candidate key, so P is prime

This schema has more redundancy than in BCNF:

Each time a class appears in a tuple, professor's name is repeated

We tolerate this redundancy bc. there is no BCNF decomposition that preserves the dependencies.

MINIMAL COVERS

→ Let F and G be sets of FDs. G is a $\boxed{\text{cover}}$ of F if $G^+ = F^+$

Same closure.

G is a small representation of all FDs in F

→ It is minimal if:

1) Each FD in G has the form $X \to A$
2) No proper subset of G is a cover
   - We can't remove FDs without losing equivalence to F
3) For $(X \to A) \in G$ and $X' \subset X$, $A \notin C_F(X')$
   - We can't remove attributes from the LHS of FDs in G

Minimal cover is not unique!

FINDING MINIMAL COVERS

1) Put the FDs in standard form → Only 1 attribute on RHS

↳ use Armstrong's decomposition axiom

$X \to A_1 A_2 \dots A_n$ is split into n FDs: $X \to A_1$, $X \to A_2$, …, $X \to A_n$

2) Minimize the LHS of each FD

↳ Check whether attributes in the LHS can be removed:

REPEAT!

- For $(X \to A) \in F$ and $X' \subset X$, check whether $A \in C_F(X')$
- If yes, replace $X \to A$ by $X' \to A$

3) Delete redundant FDs

check if remaining FDs entails that FD

↳ $(X \to A) \in F$, check whether $F - \{X \to A\} \models X \to A$

---

e.g. $\{AB \to BCD\}$

1. $\{AB \to B,\ AB \to C,\ AB \to D\}$
2. For $AB \to B$, $B \subset AB$, so we check if $B \in C_F(B) = B$

   Hence replace $AB \to B$ w/ $B \to B$.
3. The set now becomes $\{B \to B,\ AB \to C,\ AB \to D\}$

   We can remove $B \to B$ bc. $\boxed{\{AB \to C,\ AB \to D\}} \models B \to B$

   (can be derived by reflexivity)

e.g. $\{ABCD \to E,\ E \to D,\ A \to B,\ AC \to D\} = \Sigma$

↳ 1. Already in std. form

2. For $AC \to D$, $C_\Sigma(A) = AB$ and $C_\Sigma(C) = C$. hence cannot minimize

   For $ABCD \to E$, $C_\Sigma(AC) = ACBDE$, $E \in C_\Sigma(AC)$ so we can replace:

   $\Sigma_1 = \{AC \to E,\ E \to D,\ A \to B,\ AC \to D\}$

3. Check for each FDs in Σ_1:

   Σ': $\{\cancel{AC \to E},\ E \to D,\ A \to B,\ AC \to D\} \models AC \to E \to C_{\Sigma'}(AC) = ABD$

   Σ'' $\{AC \to E,\ \cancel{E \to D},\ A \to B,\ AC \to D\} \models E \to D \to C_{\Sigma''}(E) = E$

   ⋮

   Σ''' $\{AC \to E,\ E \to D,\ A \to B,\ \cancel{AC \to D}\} \models AC \to D \to C_{\Sigma'''}(AC) = ACED$ ✓

   REDUNDANT!

$\boxed{\therefore \{AC \to E,\ E \to D,\ A \to B\}}$

3NF SYNTHESIS ALGORITHM

INPUT: (U, F)

OUTPUT: $S = \{(U_1, F_1), \dots, (U_n, F_n)\}$

1. $S := \emptyset$
2. Find a minimal cover G of F
3. Replace all FDs $X \to A_1, \dots, X \to A_n$ in G by $X \to A_1 \cdots A_n$ — same LHS
4. For each FD $(X \to Y) \in G$, add $(XY, \{X \to Y\})$ to S
5. If no (U_i, F_i) in S is such that U_i is key for (U, F) (the original schema), find a key K for (U, F) and add $(K, \emptyset)$ to S
6. If S contains (U_i, F_i) and (U_j, F_j) with $U_i \subseteq U_j$, replace them by $(U_j, F_i \cup F_j)$
7. Output S

↳ The synthesized schema is:
- in 3NF
- lossless-join
- dependency-preserving