BCNF
iterative/recursive BCNF decomposition
1 lossless-join
2 dependency preserving

Database Systems

L11: Third Normal Form

$R = \{A, B, C\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$

Is R with Σ in **BCNF**?

① $CK = \{ \{A,B\}, \{A,C\} \}$

② $MC = \{ AB \rightarrow C, \cancel{C \rightarrow B} \}$

$\{A\}^+ = \{A\}$

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

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

consider $C \rightarrow B$

a) non-trivial because $\{B\} \not\subseteq \{C\}$

b) LHS is not a superkey because $\{C\}^+ = \{B,C\} \neq R$

$R = \{A, B, C\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$

Is R with Σ in **BCNF**?

NO

The keys are $\{A,B\}$ and $\{A,C\}$.

Consider $\{C\} \rightarrow \{B\} \in \Sigma$.

Since $\{B\} \not\subseteq \{C\}$, it is *non-trivial*.
Additionally, $\{C\}$ is *not a superkey*.

$R = \{A, B, C\}$

$\Sigma = \{ \{A,B\} \rightarrow \{C\}, \underline{\{C\} \rightarrow \{B\}} \}$

Decompose R into a lossless-join decomposition in **BCNF**.

$\{A, B, C\}$

$\underline{C} \rightarrow B \quad \therefore \quad \underline{\{C\}}^+ = \underline{\{B, C\}}$

$\boxed{\{B, C\}}$

$\Sigma|_{\{B,C\}} = \{ C \rightarrow B \}$

BCNF

$A \quad \cancel{B} \quad \cancel{C}$

C

$\boxed{\{A, C\}}$

$\Sigma|_{\{A,C\}} = \emptyset$

BCNF

$R = \{A, B, C\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$

Decompose R into a lossless-join decomposition in **BCNF**.

Steps

1. Using $\{C\} \rightarrow \{B\}$, computing $\{C\}^+ = \{B,C\}$, we decompose R into

$R_1 = \{B,C\}$	with $\Sigma\vert_{R_1} = \{ \{C\} \rightarrow \{B\} \}$	*(R_1 is in BCNF w.r.t. $\Sigma\vert_{R_1}$)*
$R_2 = \{A,C\}$	with $\Sigma\vert_{R_2} = \varnothing$	*(R_2 is in BCNF w.r.t. $\Sigma\vert_{R_2}$)*

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

$R = \{A, B, C\}$
$\boxed{\Sigma} = \{ \{A,B\} \to \{C\}, \{C\} \to \{B\} \}$

Is the decomposition of R into $\delta = \{ R_1(B,C), R_2(A,C) \}$ a ***dependency preserving*** decomposition?

$\Sigma|_{R_1} = \{C \to B\}$

$\Sigma|_{R_2} = \emptyset$

$\{A,B\}^+_{\Sigma_U} = \{A,B\}$

$\Sigma_U = \{ C \to B \}$

because $C \notin \{A,B\}^+$

then $\{A,B\} \not\to \{C\}$

*Alternative notation is { {B, C}, {A, C} } without naming the relation.

$\{A, B\}$ $\{A, C\}$

$R = \{A, B, C\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$

Is the decomposition of R into $\delta = \{ R_1(B,C), R_2(A,C) \}$ a *dependency preserving* decomposition?

NO

$(\Sigma|_{R_1} \cup \Sigma|_{R_2}) = \{ \{C\} \rightarrow \{B\} \}$
$\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$

Therefore, we lost $\{A,B\} \rightarrow \{C\}$.

1. all fragments are in BCNF
2. lossless join
3. dependency preserving

Note

The situation may happen when there are functional dependencies **among prime attributes**.

Idea of Third Normal Form

Let us relax* BCNF requirements for prime attributes.

*Chronologically, 3NF was defined in 1971 while BCNF was defined in 1974.
So in reality, BCNF is a strengthening of 3NF to solve other issues.

Theorem

Third Normal Form ➥

A relation R with a set of functional dependencies Σ is in *3NF* if and only if for every functional dependency $X \rightarrow \{A\} \in \Sigma^+$:

- $X \rightarrow \{A\}$ is *trivial*, **or**
- X is a *superkey*, **or**
- **A is a prime attribute**

LEMMA 4. *A relation R is 3NF iff for every elementary FD of R, say, X → A,*

(a) *X is a key for R, or*
(b) *A is a key attribute for R.*

PROOF. Easy.

Note

For relation R **before decomposition**, it is sufficient only to look at Σ.

Theorem

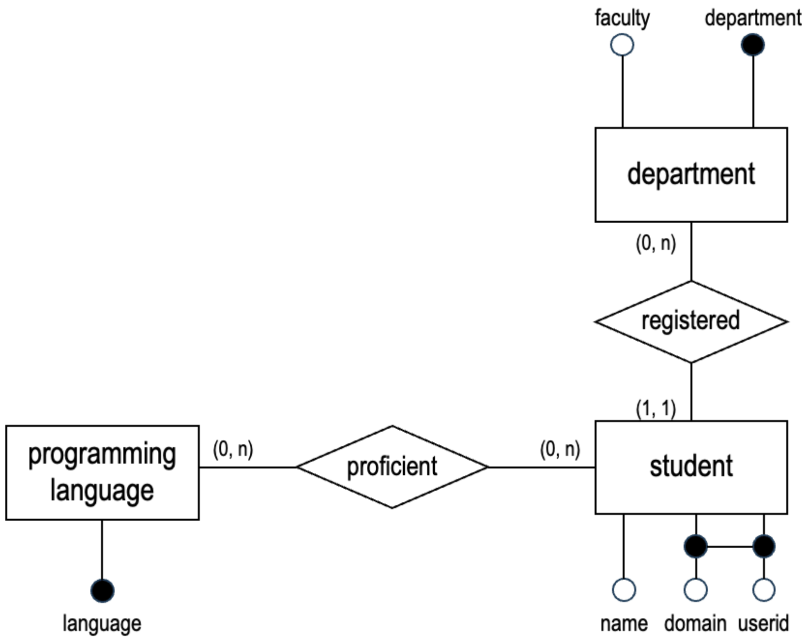
Third Normal Form Violation

A relation R with a set of functional dependencies Σ is in ***3NF*** if and only if for every functional dependency $X \rightarrow \boxed{\{A\}} \in \Sigma^+$:

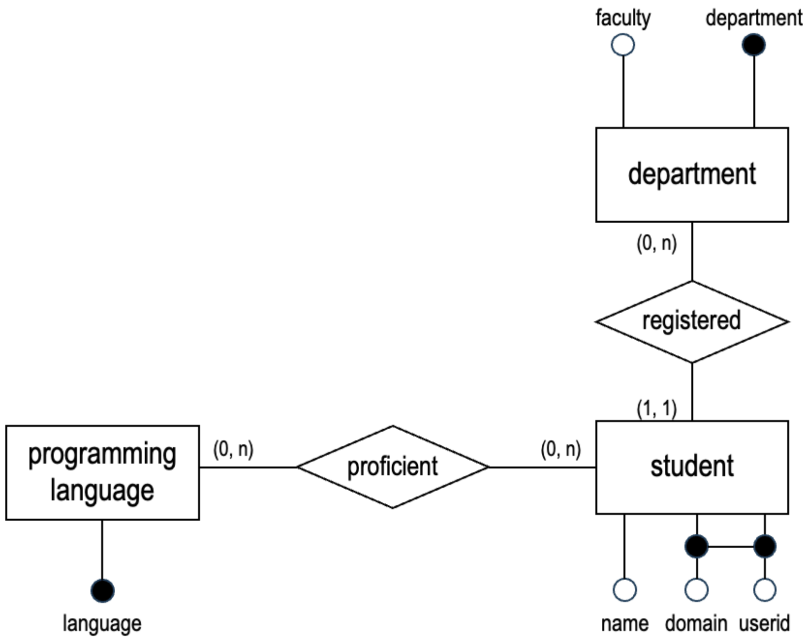
- $X \rightarrow \{A\}$ is NON-***trivial***, **and**
- X is NOT a ***superkey***, **and**
- **A is NOT a prime attribute**

$\Big\} \Longrightarrow$ "more but not all"

Issues


faculty
department
department
(0, n)
registered
(1, 1)
programming language
(0, n)
proficient
(0, n)
student
language
name
domain
userid

Issues

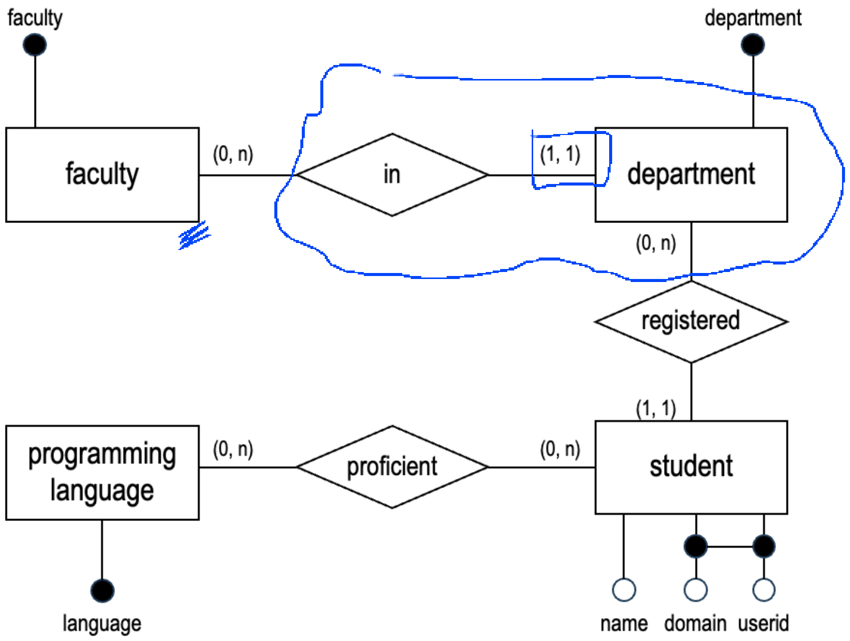


Issue #1

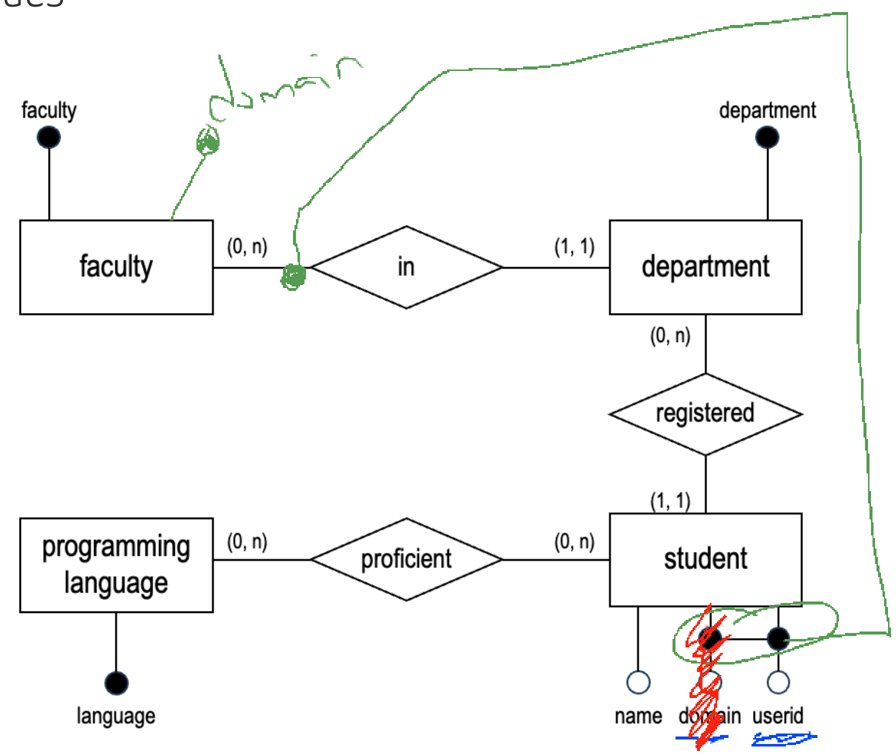
What if we want to know which faculty has **no department**?

E.g. It is a new department.

Issues


faculty
department
faculty
(0, n)
in
(1, 1)
department
(0, n)
registered
(1, 1)
programming language
(0, n)
proficient
(0, n)
student
language
name
domain
userid

Issues



Issue #2

What if each faculty has its own domain?

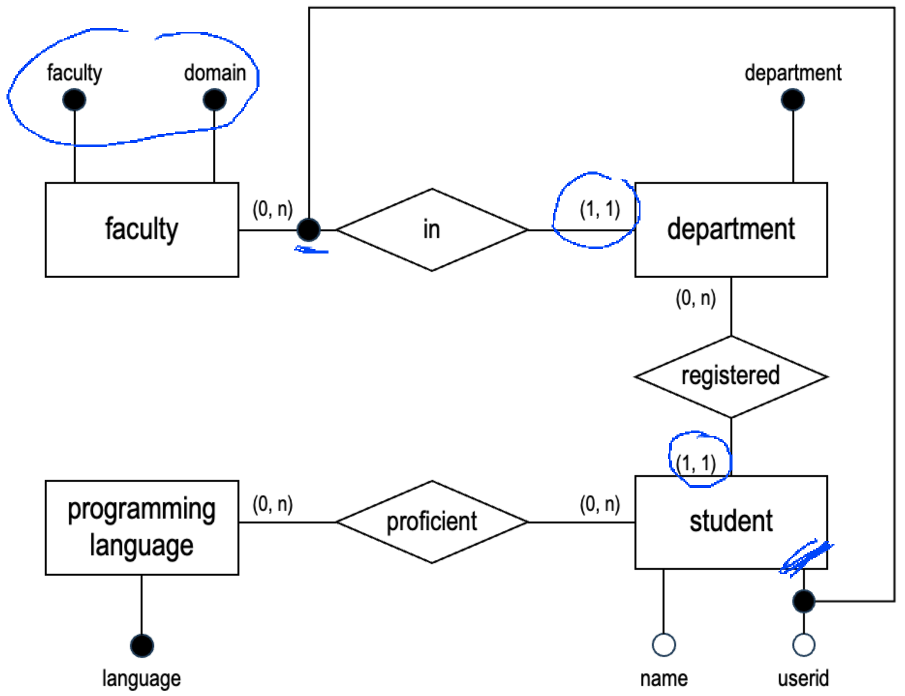
$\{\texttt{faculty}\} \rightarrow \{\texttt{domain}\}$

$\{\texttt{domain}\} \rightarrow \{\texttt{faculty}\}$

Also, student should still be uniquely identified by

$\{\texttt{domain}, \texttt{userid}\}$

ERD



The Strange Case of Far Away Dominant Entity

name, userid,
domain, department

Table

Student

(A) name	(B) userid	(C) domain	(D) department	(E) faculty
Tan Hee Wee	tanh	comp.sut.edu	computer science	computing
Stanley Georgeau	stan	comp.sut.edu	computer science	computing
Goh Jin Wei	goh	comp.sut.edu	information system	computing
Tan Hee Wee	tanhw	eng.sut.edu	computer engineering	engineering
Bjorn Sale	bjorn	eng.sut.edu	computer engineering	engineering
Tan Hooi Ling	tanh	sci.sut.edu	physics	science
Roxana Nassi	rox	sci.sut.edu	mathematics	science
Ami Mokhtar	ami	med.sut.edu	pharmacy	medicine

$$\Sigma|_{R_1} = \{$$
$$\{B,C\} \rightarrow \{A,D\}, \checkmark$$
$$\{D\} \rightarrow \{E\}, \checkmark$$
$$\{E\} \rightarrow \{C\}, \checkmark$$
$$\{C\} \rightarrow \{E\} \checkmark$$
$$\}$$

$\{D\} \rightarrow \{C\}$

Table

Student

(A) name	(B) userid	(C) domain	(D) department	(E) faculty
Tan Hee Wee	tanh	comp.sut.edu	computer science	computing
Stanley Georgeau	stan	comp.sut.edu	computer science	computing
Goh Jin Wei	goh	comp.sut.edu	information system	computing
Tan Hee Wee	tanhw	eng.sut.edu	computer engineering	engineering
Bjorn Sale	bjorn	eng.sut.edu	computer engineering	engineering
Tan Hooi Ling	tanh	sci.sut.edu	physics	science
Roxana Nassi	rox	sci.sut.edu	mathematics	science
Ami Mokhtar	ami	med.sut.edu	pharmacy	medicine

$$\Sigma|_{R_1} = \{ \{B,C\} \rightarrow \{A,D\}, \{D\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{C\} \rightarrow \{E\} \}$$

not in BCNF ✗

D → E is a violation

in 3NF ✓

BC → AD : BC is SK

D → E : E is PA

E → C : C is PA

C → E : E is PA

Candidate Keys

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

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

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

{B, C, D, E}

Table

Student

(A) name	(B) userid	(C) domain	(D) department	(E) faculty
Tan Hee Wee	tanh	comp.sut.edu	computer science	computing
Stanley Georgeau	stan	comp.sut.edu	computer science	computing
Goh Jin Wei	goh	comp.sut.edu	information system	computing
Tan Hee Wee	tanhw	eng.sut.edu	computer engineering	engineering
Bjorn Sale	bjorn	eng.sut.edu	computer engineering	engineering
Tan Hooi Ling	tanh	sci.sut.edu	physics	science
Roxana Nassi	rox	sci.sut.edu	mathematics	science
Ami Mokhtar	ami	med.sut.edu	pharmacy	medicine

$$\Sigma|_{R_1} = \{ \{B,C\} \rightarrow \{A,D\}, \{D\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{C\} \rightarrow \{E\} \}$$

Note

`student` is in 3NF but not in BCNF.

Candidate Keys

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

Synthesis

$\Sigma \equiv \Sigma_C \equiv \Sigma_U$

Algorithm #5: 3NF Synthesis (Bernstein Algorithm)

When a relation is not in 3NF, we can ***synthesize*** a schema in 3NF from a **minimal cover** of the set of functional dependencies.

- For each functional dependency $X \rightarrow Y$ in the minimal cover, create a relation $R_i = X \cup Y$ ⇇ → $X \rightarrow Y \in \Sigma|_{R_i}$

  $X \rightarrow \{A\}$
  $Y \rightarrow \{B\}$
  ⋮

  dep preserve

  Unless it already exists or is ***subsumed*** by another relation *(with some exceptions...)*.
- If none of the created relations contain one of the keys, pick **any** candidate key and create a relation with that candidate key. ⇒ guarantees lossless join

"Synthesizing Third Normal Form relations from functional dependencies"

*We still call the synthesis method a decomposition because we decompose a relation into multiple relations without any loss of attributes.

Intuition

3NF Synthesis Idea

1. **Simplification**: Use of ***minimal cover***.

2. **Partition**: Use of ***canonical cover*** and ***subsumption***.

3. **Synthesis**: Creation of ***relation*** from ***partitioned attributes***.

4. **Candidate Key**: Adding ***candidate key*** as one relation **if it is not yet subsumed**.

Subsumption

$R_1(A, B, C)$

$AB \to C$

$\cancel{R_2(B, C)}$

$\cancel{C \to B}$

$\cancel{R_3(A, B)}$

Notes

Canonical Cover

In order to avoid unnecessary decomposition, it is generally a good idea to use a **canonical cover** instead of **minimal cover** *(we shall do so unless we explicitly identify a problem)*.

Subsumption

We will always perform **subsumption** in our algorithm *(but we will discuss why it does not always work)*.

Theorem 9

The algorithm guarantees *lossless-join*, *dependency-preserving* decomposition in 3NF.

BCNF?

Very often *(but not always)*, the decomposition is also in **BCNF**.

3NF Synthesis

① : CK = {{A, B}, {A, C}} ⇒ PA = {A, B, C}

② : M_C = {B → C, C → B, C → D, C → E}

Example

R = {*A*, *B*, *C*, *D*, *E*}

Σ = { {A,B} → {C,D,E}, {A,C} → {B,D,E}, {B} → {C}, {C} → {B}, {C} → {D}, {B} → {E}, {C} → {E} }

Decompose *R* with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

(1) min
(2) C is not SK
(3) E is not PA

Σ_C = { B → C, C → B, C → D, C → E }

Σ_D = { B → C, C → BDE }

~~R(B, C)~~

R_2(B, C, D, E)

R_K(A, B)

δ = { R_2, R_K }

3NF Synthesis

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Steps

1. Compute **candidate keys**. $\{A, B\}$ and $\{A, C\}$

3NF Synthesis

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Steps

1. Compute **candidate keys**. — $\{A, B\}$ and $\{A, C\}$
2. Compute **minimal cover Σ_C** of Σ. — $\Sigma_C = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{C\} \rightarrow \{E\} \}$

3NF Synthesis

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Steps

1. Compute **candidate keys**. $\{A, B\}$ and $\{A, C\}$
2. Compute **minimal cover** Σ_C of Σ. $\Sigma_C = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{C\} \rightarrow \{E\} \}$
3. Compute **canonical cover** Σ_D of Σ. $\Sigma_D = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B,D,E\} \}$

3NF Synthesis

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Steps

1. Compute **candidate keys**. — $\{A, B\}$ and $\{A, C\}$
2. Compute **minimal cover** Σ_C of Σ. — $\Sigma_C = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{C\} \rightarrow \{E\} \}$
3. Compute **canonical cover** Σ_D of Σ. — $\Sigma_D = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B,D,E\} \}$
4. Synthesize R for each $\sigma \in \Sigma_D$. — $R_1 = \{B, C\}$ and $R_2 = \{B, C, D, E\}$

3NF Synthesis

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Steps

1. Compute **candidate keys**. — $\{A, B\}$ and $\{A, C\}$
2. Compute **minimal cover** Σ_C of Σ. — $\Sigma_C = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{C\} \rightarrow \{E\} \}$
3. Compute **canonical cover** Σ_D of Σ. — $\Sigma_D = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B,D,E\} \}$
4. Synthesize R for each $\sigma \in \Sigma_D$. — $R_1 = \{B, C\}$ and $R_2 = \{B, C, D, E\}$
5. Remove *subsumed* relations. — $R_2 = \{B, C, D, E\}$

3NF Synthesis

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Steps

1. Compute **candidate keys**.	$\{A, B\}$ and $\{A, C\}$
2. Compute **minimal cover** Σ_C of Σ.	$\Sigma_C = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{C\} \rightarrow \{E\} \}$
3. Compute **canonical cover** Σ_D of Σ.	$\Sigma_D = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B,D,E\} \}$
4. Synthesize R for each $\sigma \in \Sigma_D$.	$R_1 = \{B, C\}$ and $R_2 = \{B, C, D, E\}$
5. Remove *subsumed* relations.	$R_2 = \{B, C, D, E\}$
6. Add candidate keys *(if needed)*.	$R_2 = \{B, C, D, E\}$ and $R_3 = \{A, C\}$*

*We can also add $R_3 = \{A, B\}$

3NF Synthesis

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Answer

The resulting decomposition is:

- $R_2 = \{B, C, D, E\}$ with $\Sigma|_{R_1} = \{ \{B\} \rightarrow \{C, D, E\}, \{C\} \rightarrow \{B, D, E\} \}$
  Candidate Keys: $\{B\}$ and $\{C\}$
- $R_3 = \{A, C\}$ with $\Sigma|_{R_2} = \varnothing$

Example

Example

$R = \{A, B, C, D, E\}$
$\Sigma = \{ \{A,B\} \rightarrow \{C,D,E\}, \{A,C\} \rightarrow \{B,D,E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\} \}$

Decompose R with Σ into a lossless-join, dependency-preserving decomposition in **3NF**.

Alternative Answer

Other alternative answers can be obtained from other minimal cover or candidate keys.

- $R_2 = \{B, C, D, E\}$
- $R_3 = \{A, B\}$

- $R_2 = \{B, C, D\}$
- $R_3 = \mathbf{\{B, C, E\}}$
- $R_4 = \{A, C\}$

- $R_2 = \{B, C, D\}$
- $R_3 = \mathbf{\{B, C, E\}}$
- $R_3 = \{A, B\}$ 4

Remove

$\boxed{B} \rightarrow C$ $\quad$ $C \rightarrow BDE$

In the previous example, R_1 = {B, C} is ***subsumed*** by R_2 = *{B, C, D, E}*. The functional dependencies *{B}* → *{C, D, E}* and *{C}* → *{B, D, E}* can still be enforced.

Schema

```
CREATE TABLE R2 (
  B  INT  PRIMARY KEY,       --   {B} -> {C, D, E}
  C  INT  UNIQUE NOT NULL,   --   {C} -> {B, D, E}
  D  INT,
  E  INT

);
```

Keep

In some cases like R = {A, B, C} with Σ = { {A,B} → {C}, {C} → {B} }, we *cannot remove* R_2 = {B, C} even when it is subsumed by R_1 = {A, B, C}.

Only {A,B} → {C}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT,
  PRIMARY KEY (A, B)
);
```

A	B	C
1	1	1
1	2	1

Only {C} → {B}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT PRIMARY KEY
);
```

A	B	C
1	1	1
1	1	2

Both?

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT  UNIQUE NOT NULL,
  PRIMARY KEY (A, B)
);
```

A	B	C
1	1	1
2	1	1

Keep

In some cases like $R = \{A, B, C\}$ with $\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$, we *cannot remove* $R_2 = \{B, C\}$ even when it is subsumed by $R_1 = \{A, B, C\}$.

Only {A,B} → {C}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT,
  PRIMARY KEY (A, B)
);
```

A	B	C
1	1	1
2	2	1

Only {C} → {B}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT PRIMARY KEY

);
```

A	B	C

Both?

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT  UNIQUE NOT NULL,
  PRIMARY KEY (A, B)
);
```

A	B	C

Keep

In some cases like $R = \{A, B, C\}$ with $\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$, we cannot remove $R_2 = \{B, C\}$ even when it is subsumed by $R_1 = \{A, B, C\}$.

Only {A,B} → {C}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT,
  PRIMARY KEY (A, B)
);
```

A	B	C

Only {C} → {B}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT PRIMARY KEY

);
```

A	B	C
1	1	1
1	1	2

Both?

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT  UNIQUE NOT NULL,
  PRIMARY KEY (A, B)
);
```

A	B	C

Keep

In some cases like $R = \{A, B, C\}$ with $\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$, we <u>*cannot remove*</u> $R_2 = \{B, C\}$ even when it is subsumed by $R_1 = \{A, B, C\}$.

Only {A,B} → {C}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT,
  PRIMARY KEY (A, B)
);
```

A	B	C

Only {C} → {B}

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT PRIMARY KEY

);
```

A	B	C

Both?

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT  UNIQUE NOT NULL,
  PRIMARY KEY (A, B)
);
```

A	B	C
1	1	1
2	1	1

Solution

In some cases like $R = \{A, B, C\}$ with $\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$, we *cannot remove* $R_2 = \{B, C\}$ even when it is subsumed by $R_1 = \{A, B, C\}$.

R2(B, C)

```sql
CREATE TABLE R2 (
  B  INT,
  C  INT
    UNIQUE,
  PRIMARY KEY (B, C)

);
```

C → B

R1(A, B, C)

```sql
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT,
  PRIMARY KEY (A, B),
  FOREIGN KEY (B, C) REFERENCES R2(B, C)
);
```

AB → C

Solution

In some cases like $R = \{A, B, C\}$ with $\Sigma = \{ \{A,B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\} \}$, we <u>*cannot remove*</u> $R_2 = \{B, C\}$ even when it is subsumed by $R_1 = \{A, B, C\}$.

R2(B, C)

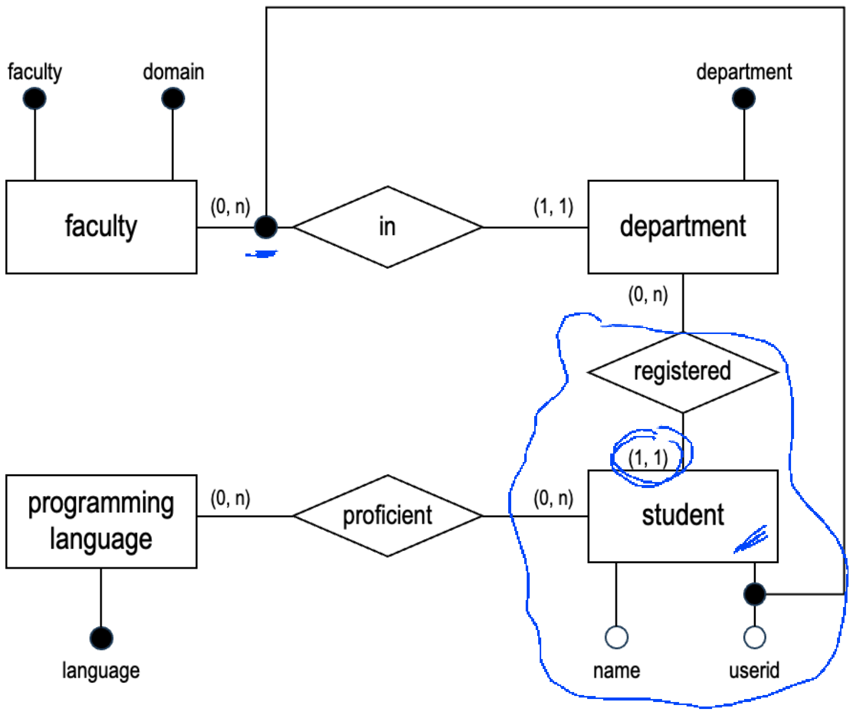
```
CREATE TABLE R2 (
  B  INT,
  C  INT
    UNIQUE,
  PRIMARY KEY (B, C)

);
```

R1(A, B, C)

```
CREATE TABLE R1 (
  A  INT,
  B  INT,
  C  INT,
  PRIMARY KEY (A, B),
  FOREIGN KEY (B, C) REFERENCES R2(B, C)
);
```

ERD



student(name, userid, domain, departmet)

Schema

Tables

`language(language)`	**BCNF**
`faculty(faculty, domain)`	**BCNF**
`department(department, faculty)`	**BCNF**
`student(userid, faculty, name, department)`	**3NF**
`proficiency(userid, faculty, language)`	**BCNF**

3NF Check

```
student(userid, faculty, name, department)
```

Projected Functional Dependency

- {userid, faculty} → {name, department}
- {department} → {faculty}

3NF Check

`student(`<u>`userid, faculty`</u>`, name, department)`

Projected Functional Dependency

- {`userid`, `faculty`} → {`name`, `department`}
- {`department`} → {`faculty`}

Candidate Keys

{`userid`, `faculty`} and {`userid`, `department`}

3NF Check

student(<u>userid, faculty</u>, name, department)

Projected Functional Dependency

- {userid, faculty} → {name, department}
- {department} → {faculty}

{userid, faculty} is **superkey**
{department} is **not a superkey**

Candidate Keys

{userid, faculty} and {userid, department}

3NF Check

student(<u>userid, faculty</u>, name, department)

Projected Functional Dependency

- {userid, faculty} → {name, department}
- {department} → {faculty}

{userid, faculty} is **superkey**
faculty is a **prime attribute**

Candidate Keys

{userid, faculty} and {userid, department}

BCNF Check

B E A D

```
student(userid, faculty, name, department)
```

Projected Functional Dependency

B E A D

- {userid, faculty} → {name, department}
- {department} → {faculty}

D E

{userid, faculty} is **superkey**
{department} is **not a superkey**

Candidate Keys

{userid, faculty} and {userid, department}

$D \rightarrow E$ $\{D\}^+ = \{D, E\}$

Question

$R_1(D, E)$ $R_2(A, B, D)$

Let us decompose using **BCNF decomposition algorithm**.

BCNF Check

`student(userid, faculty, name, department)` *student(B,E,A,D)*

Projected Functional Dependency

- {userid, faculty} → {name, department} — {userid, faculty} is **superkey**
- {department} → {faculty} — {department} is **not a superkey**

BCNF Decomposition

- R_1 = {department, faculty} — $\{D, E\}$ from $\{D\} \rightarrow \{E\}$
- R_2 = {userid, department, name} — $\{B, D, A\}$ from $(\{B, E, A, D\} - \{D, E\}) \cup \{D\}$

BCNF Check

```
student(userid, faculty, name, department)
```

Projected Functional Dependency

- {userid, faculty} → {name, department} ✗
- {department} → {faculty} ✓

$\{uid, fac\}^+ = \{uid, fac\}$

BCNF Decomposition

- R_1 = {department, faculty} $\Sigma|_{R_1}$ = { {department} → {faculty} }
- R_2 = {userid, department, name} $\Sigma|_{R_2}$ = { {userid, department} → {name} }

BCNF Check

```
student(userid, faculty, name, department)
```

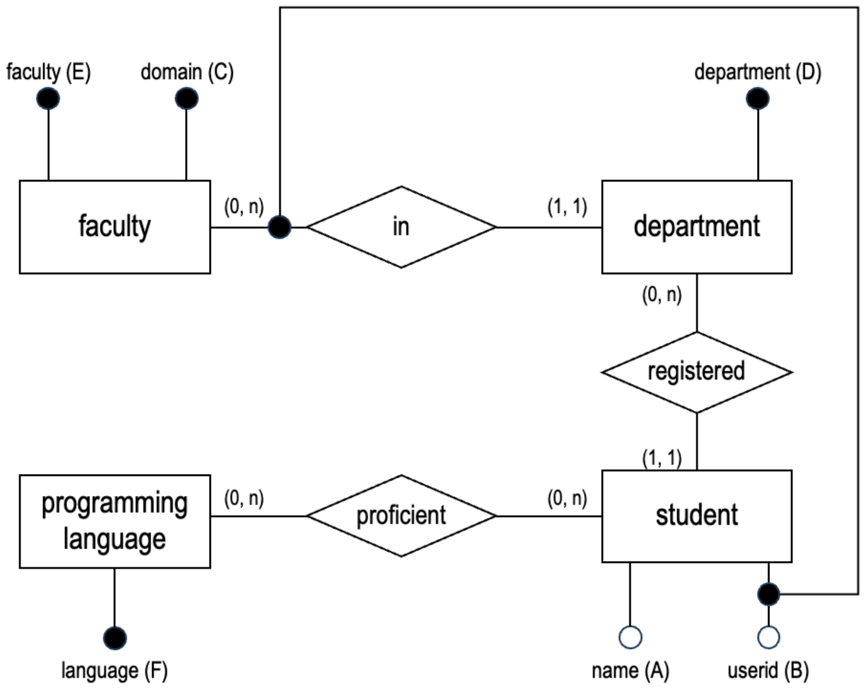
Projected Functional Dependency

- {userid, faculty} → {name, department} **not preserved**
- {department} → {faculty} **preserved**

BCNF Decomposition

- R_1 = {department, faculty} $\Sigma|_{R_1}$ = { {department} → {faculty} }
- R_2 = {userid, department, name} $\Sigma|_{R_2}$ = { {userid, department} → {name} }

FD



Mapping

Attribute	Letter	Attribute	Letter
name	A	department	D
userid	B	faculty	E
domain	C	language	F

$$\Sigma = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$$

Question

What if we do **BCNF decomposition** on the **entire relation**.

Preliminary

$\Sigma = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Candidate Keys

$\{B, C, F\}, \{B, D, F\}, \{B, E, F\}$

Canonical Cover

$\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Is this the only one?

BCNF

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

Min Cover: Σ_D = { {C} → {E}, {E} → {C}, {D} → {E}, {B, C} → {A, D} }

Keys: {B, C, F} {B, D, F} {B, E, F}

BCNF

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: $\{B, C, F\}$ $\{B, D, F\}$ $\{B, E, F\}$

Using {B, C} → {A, D} on R

- $R_1 = \{A, B, C, D, E\}$ with $\Sigma|_{R_1} = \{ \{B, C\} \rightarrow \{A, D, E\}, \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $R_2 = \{B, C, F\}$ with $\Sigma|_{R_2} = \emptyset$

$\cancel{A}$ $\cancel{B}$ $\cancel{C}$ $\cancel{D}$ $\cancel{E}$ F

B C F

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: $\{B, C, F\}$ $\{B, D, F\}$ $\{B, E, F\}$

BCNF

Using {B, C} → {A, D} on R

- $R_1 = \{A, B, C, D, E\}$ with $\Sigma|_{R_1} = \{ \{B, C\} \rightarrow \{A, D, E\}, \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $\boldsymbol{R_2 = \{B, C, F\}}$ with $\Sigma|_{R_2} = \varnothing$

BCNF

Using {B, C} → {A, D} on R

- $R_1 = \{A, B, C, D, E\}$ with $\Sigma|_{R_1} = \{ \{B, C\} \rightarrow \{A, D, E\}, \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $\boldsymbol{R_2 = \{B, C, F\}}$ with $\Sigma|_{R_2} = \varnothing$

Using {D} → {C, E} on R1

- $R_3 = \{C, D, E\}$ with $\Sigma|_{R_3} = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $R_4 = \{A, B, D\}$ with $\Sigma|_{R_4} = \varnothing$

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: $\{B, C, F\}$ $\{B, D, F\}$ $\{B, E, F\}$

BCNF

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: $\{B, C, F\}$ $\{B, D, F\}$ $\{B, E, F\}$

Using {B, C} → {A, D} on R

- $R_1 = \{A, B, C, D, E\}$ with $\Sigma|_{R_1} = \{ \{B, C\} \rightarrow \{A, D, E\}, \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $\boldsymbol{R_2 = \{B, C, F\}}$ with $\Sigma|_{R_2} = \varnothing$

Using {D} → {C, E} on R1

- $R_3 = \{C, D, E\}$ with $\Sigma|_{R_3} = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $\boldsymbol{R_4 = \{A, B, D\}}$ with $\Sigma|_{R_4} = \varnothing$

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: {B, C, F} {B, D, F} {B, E, F}

BCNF

Using {B, C} → {A, D} on R

- $R_1 = \{A, B, C, D, E\}$ with $\Sigma|_{R_1} = \{ \{B, C\} \rightarrow \{A, D, E\}, \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $\boldsymbol{R_2 = \{B, C, F\}}$ with $\Sigma|_{R_2} = \varnothing$

Using {D} → {C, E} on R1

- $R_3 = \{C, D, E\}$ with $\Sigma|_{R_3} = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{C, E\} \}$
- $\boldsymbol{R_4 = \{A, B, D\}}$ with $\Sigma|_{R_4} = \varnothing$

Using {C} → {E} on R3

- $R_5 = \{C, E\}$ with $\Sigma|_{R_5} = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\} \}$
- $R_6 = \{C, D\}$ with $\Sigma|_{R_6} = \{ \{D\} \rightarrow \{C\} \}$

BCNF

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

Min Cover: Σ_D = { {C} → {E}, {E} → {C}, {D} → {E}, {B, C} → {A, D} }

Keys: {B, C, F}   {B, D, F}   {B, E, F}

Using {B, C} → {A, D} on R

- R_1 = {A, B, C, D, E}   with $\Sigma|_{R_1}$ = { {B, C} → {A, D, E}, {C} → {E}, {E} → {C}, {D} → {C, E} }
- **R_2 = {B, C, F}**   with $\Sigma|_{R_2} = \varnothing$

Using {D} → {C, E} on R1

- R_3 = {C, D, E}   with $\Sigma|_{R_3}$ = { {C} → {E}, {E} → {C}, {D} → {C, E} }
- **R_4 = {A, B, D}**   with $\Sigma|_{R_4} = \varnothing$

Using {C} → {E} on R3

- **R_5 = {C, E}**   with $\Sigma|_{R_5}$ = { {C} → {E}, {E} → {C} }
- **R_6 = {C, D}**   with $\Sigma|_{R_6}$ = { {D} → {C} }

BCNF

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

Min Cover: Σ_D = { {C} → {E}, {E} → {C}, {D} → {E}, {B, C} → {A, D} }

Keys: {B, C, F}   {B, D, F}   {B, E, F}

Using {B, C} → {A, D} on R

- R_1 = {A, B, C, D, E}   with $\Sigma|_{R_1}$ = { {B, C} → {A, D, E}, {C} → {E}, {E} → {C}, {D} → {C, E} }
- **R_2 = {B, C, F}**   with $\Sigma|_{R_2}$ = ∅

Using {D} → {C, E} on R1

- R_3 = {C, D, E}   with $\Sigma|_{R_3}$ = { {C} → {E}, {E} → {C}, {D} → {C, E} }
- **R_4 = {A, B, D}**   with $\Sigma|_{R_4}$ = ∅

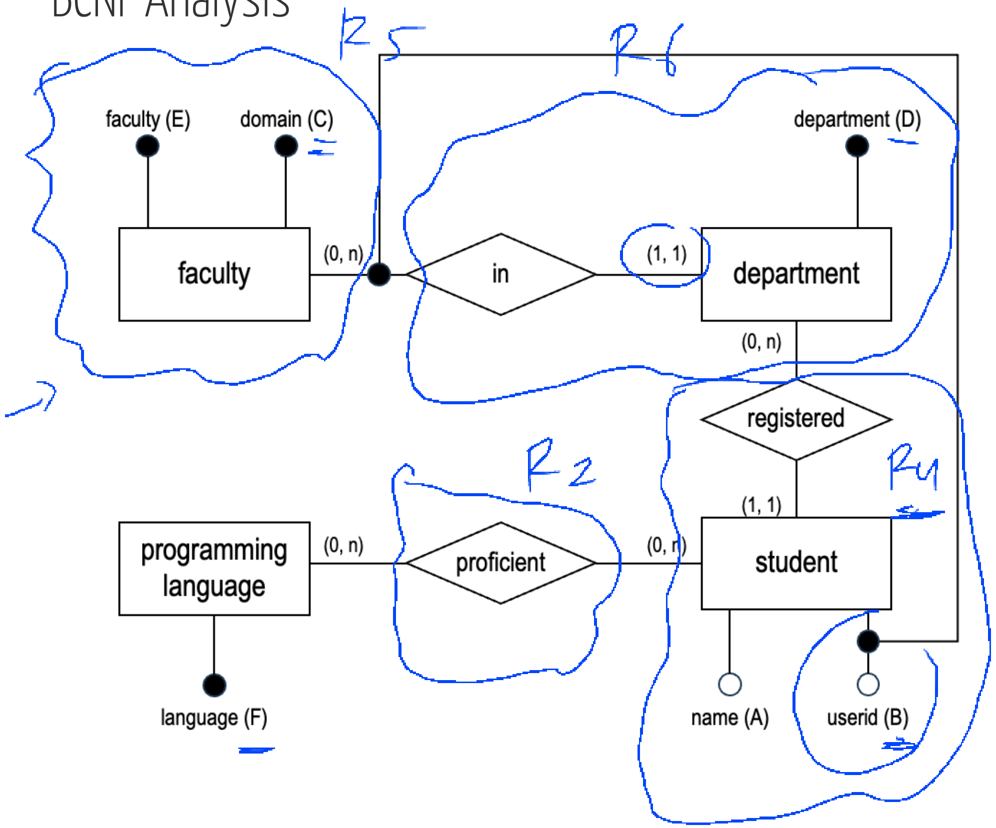
Using {C} → {E} on R3

- **R_5 = {C, E}**   with $\Sigma|_{R_5}$ = { {C} → {E}, {E} → {C} }
- **R_6 = {C, D}**   with $\Sigma|_{R_6}$ = { {D} → {C} }

Fragments

R_2 = {B, C, F} R_5 = {C, E}
R_4 = {A, B, D} R_6 = {C, D}

BCNF Analysis



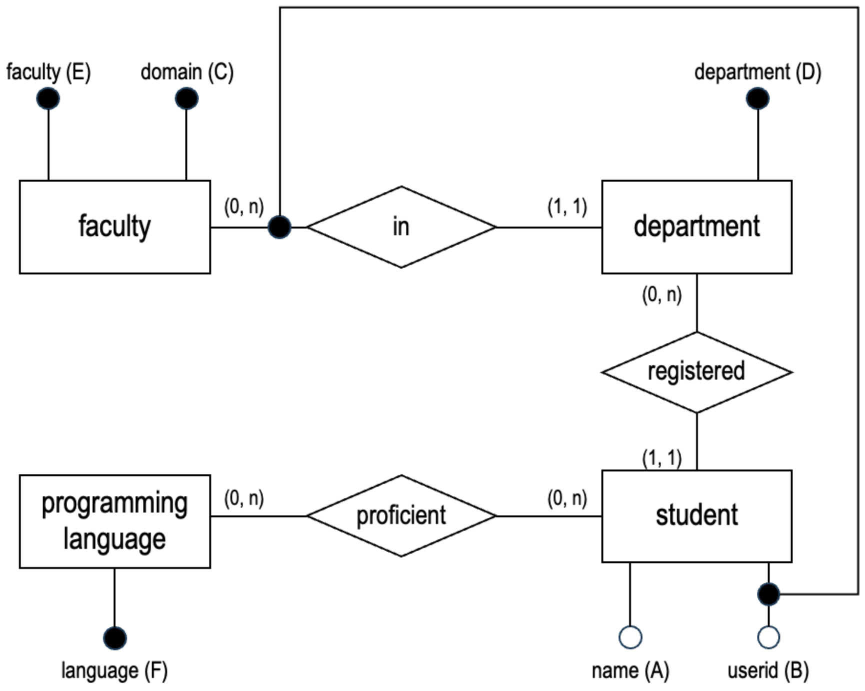
Fragments

- $R_2 = \{B, C, F\}$ $\varnothing$
- $R_4 = \{A, B, D\}$ $\varnothing$
- $R_5 = \{C, E\}$ $\{\{C\} \to \{E\}, \{E\} \to \{C\}\}$
- $R_6 = \{C, D\}$ $\{\{D\} \to \{C\}\}$

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

$$\Sigma_U = \{ \{C\} \to \{E\}, \{E\} \to \{C\}, \{D\} \to \{E\} \}$$

BCNF Analysis



Fragments

- $R_2 = \{B, C, F\}$ $\varnothing$
- $R_4 = \{A, B, D\}$ $\varnothing$
- $R_5 = \{C, E\}$ $\{\{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}\}$
- $R_6 = \{C, D\}$ $\{\{D\} \rightarrow \{C\}\}$

$\Sigma = \{$
$\{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\},$
$\{D\} \rightarrow \{E\}, \mathbf{\{B, C\} \rightarrow \{A, D\}}$
$\}$

$\Sigma_U = \{$
$\{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}$
$\}$

3NF

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: {B, C, F} {B, D, F} {B, E, F}

3NF

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

Canon
~~Min~~ ***Cover:*** Σ_D = { {C} → {E}, {E} → {C}, {D} → {E}, {B, C} → {A, D} }

Keys: {B, C, F} {B, D, F} {B, E, F}

Synthesize

{C} → {E}	{E} → {C}	{D} → {E}	{B,C} → {A,D}
R_1(C, E)	R_2(C, E)	R_3(D, E)	R_4(A, B, C, D)

3NF

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: $\{B, C, F\}$ $\{B, D, F\}$ $\{B, E, F\}$

Synthesize

$\{C\} \rightarrow \{E\} \Downarrow R_1 = \{C, E\}$

$\{E\} \rightarrow \{C\} \Downarrow \cancel{R_2 = \{C, E\}}$

$\{D\} \rightarrow \{E\} \Downarrow R_3 = \{D, E\}$

$\{B,C\} \rightarrow \{A,D\} \Downarrow R_4 = \{A, B, C, D\}$

3NF

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

Min Cover: Σ_D = { {C} → {E}, {E} → {C}, {D} → {E}, {B, C} → {A, D} }

Keys: {B, C, F} {B, D, F} {B, E, F}

Synthesize

{C} → {E}	{E} → {C}	{D} → {E}	{B,C} → {A,D}
⇓	⇓	⇓	⇓
R_1 = {C, E}	R_2 = {C, E}	R_3 = {D, E}	R_4 = {A, B, C, D}

Subsume

- R_1 = {C, E} with $\Sigma|_{R_1}$ = { {C} → {E}, {E} → {C} } *(can we really subsume?)*
- R_3 = {D, E} with $\Sigma|_{R_3}$ = { {D} → {E} }
- R_4 = {A, B, C, D} with $\Sigma|_{R_4}$ = { {B, C} → {A, D} }

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

Min Cover: Σ_D = { {C} → {E}, {E} → {C}, {D} → {E}, {B, C} → {A, D} }

Keys: {B, C, F} {B, D, F} {B, E, F}

3NF

Synthesize

{C} → {E}	{E} → {C}	{D} → {E}	{B,C} → {A,D}
⇓	⇓	⇓	⇓
R_1 = {C, E}	R_2 = {C, E}	R_3 = {D, E}	R_4 = {A, B, C, D}

Subsume

- R_1 = {C, E} with $\Sigma|_{R_1}$ = { {C} → {E}, {E} → {C} } *(can we really subsume?)*
- R_3 = {D, E} with $\Sigma|_{R_3}$ = { {D} → {E} }
- R_4 = {A, B, C, D} with $\Sigma|_{R_4}$ = { {B, C} → {A, D} }

Add Key

- R_5 = {B, C, F} with $\Sigma|_{R_5}$ = ∅

3NF

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

Min Cover: $\Sigma_D = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$

Keys: $\{B, C, F\}$ $\{B, D, F\}$ $\{B, E, F\}$

Synthesize

$\{C\} \rightarrow \{E\}$	$\{E\} \rightarrow \{C\}$	$\{D\} \rightarrow \{E\}$	$\{B,C\} \rightarrow \{A,D\}$
$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
$R_1 = \{C, E\}$	$R_2 = \{C, E\}$	$R_3 = \{D, E\}$	$R_4 = \{A, B, C, D\}$

Subsume

- $R_1 = \{C, E\}$ with $\Sigma|_{R_1} = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\} \}$ *(can we really subsume?)*
- $R_3 = \{D, E\}$ with $\Sigma|_{R_3} = \{ \{D\} \rightarrow \{E\} \}$
- $R_4 = \{A, B, C, D\}$ with $\Sigma|_{R_4} = \{ \{B, C\} \rightarrow \{A, D\} \}$

Add Key

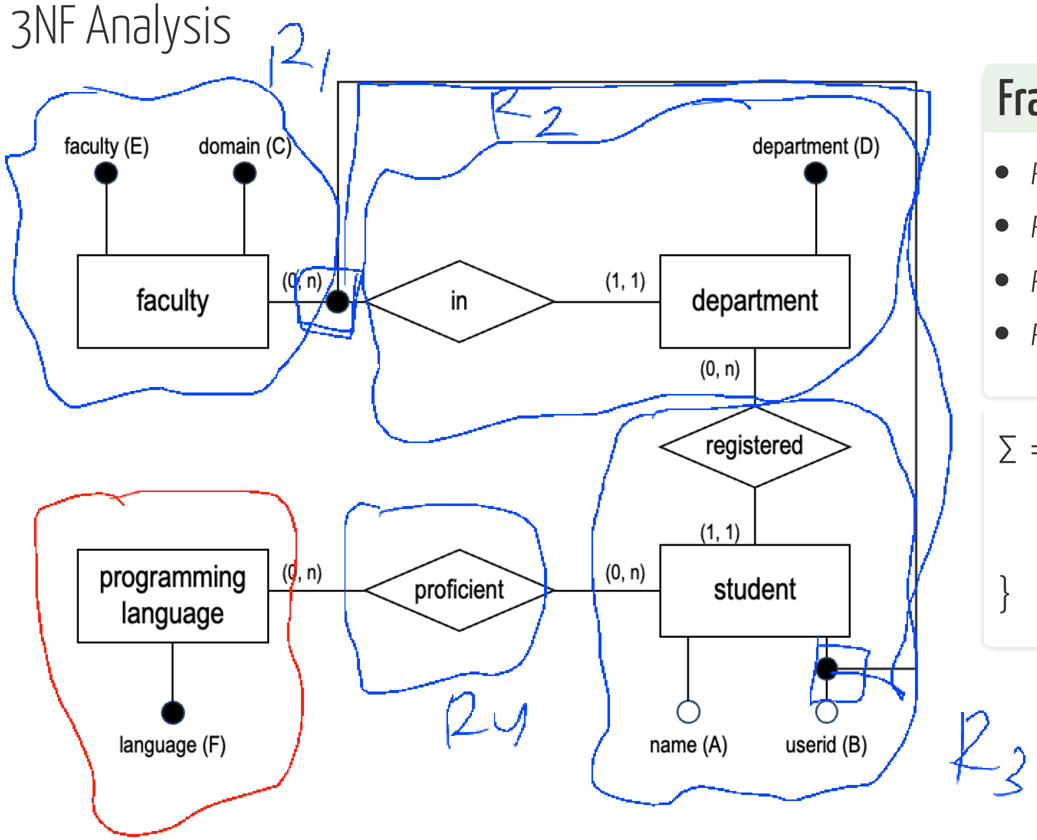
- $R_5 = \{B, C, F\}$ with $\Sigma|_{R_5} = \varnothing$

Fragments

$R_1 = \{C, E\}$ $R_4 = \{A, B, C, D\}$

$R_3 = \{D, E\}$ $R_5 = \{B, C, F\}$

3NF Analysis

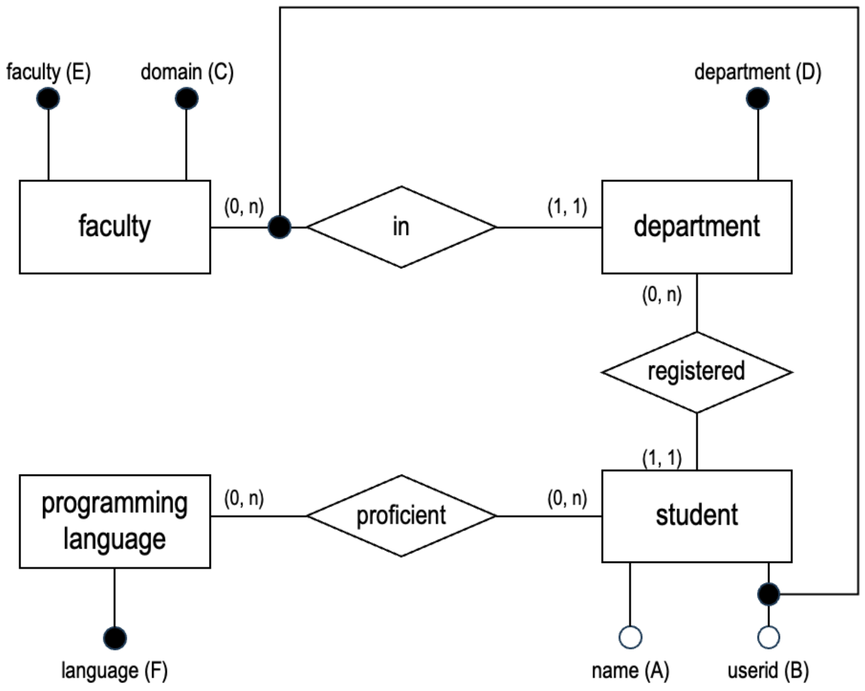


Fragments

- $R_1 = \{C, E\}$ $\{\{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}\}$
- $R_2 = \{D, E\}$ $\{\{D\} \rightarrow \{E\}\}$
- $R_3 = \{A, B, C, D\}$ $\{\{B, C\} \rightarrow \{A, D\}\}$
- $R_4 = \{B, C, F\}$ $\emptyset$

$$\Sigma = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$$

3NF



Fragments

- $R_1 = \{C, E\}$ $\{\{C\}\rightarrow\{E\}, \{E\}\rightarrow\{C\}\}$
- $R_2 = \{D, E\}$ $\{\{D\}\rightarrow\{E\}\}$
- $R_3 = \{A, B, C, D\}$ $\{\{B,C\}\rightarrow\{A,D\}\}$
- $R_4 = \{B, C, F\}$ $\varnothing$

$$\Sigma = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}, \{D\} \rightarrow \{E\}, \{B, C\} \rightarrow \{A, D\} \}$$

Normal Forms ❶ ❷

Theorem 10

$$(4NF) \subseteq \textbf{BCNF} \subseteq \textbf{3NF} \subseteq (2NF) \subseteq "1NF"$$

Theorem 11

$$1NF \neq 2NF \neq 3NF \neq BCNF \neq 4NF$$

Note

There are more normal forms that corresponds to functional dependencies as well as other integrity constraints *(e.g., multi-valued dependency in 4NF)*.

*We are not sketching a proof for these as we have not fully defined them.

Normal Forms ❶ ❷

all values are atomic

Corollary

$$(4NF) \subset \textbf{BCNF} \subset \textbf{3NF} \subset (2NF) \subset \text{"1NF"}$$

There are decompositions in which the fragments are in 3NF *(similarly, BCNF)* but not in BCNF *(similarly, 4NF)*.

Note

In some cases, we do not want **3NF** and may opt for **2NF**. For instance, 3NF will require more *(inner? outer?)* joins to query. This may be an expensive operations for large database.

```
postgres=# exit
Press any key to continue . . .
```

Proof Sketch

Only for Reading ; Not Tested

Theorem #9

Proof ↩

We will focus on dependency-preserving decomposition here.

- Note that we start from a minimal cover Σ_C.
  Since Σ_C is a minimal cover of Σ, we know $\Sigma_C \equiv \Sigma$.
- For each functional dependencies $X \rightarrow Y$ in Σ, we form a relation $R_i = X \cup Y$.
  Hence $X \rightarrow Y$ is in the projected functional dependencies on R_i from R with Σ.
  If it is subsumed, then there must be $R_j \supseteq R_i$. Hence $X \rightarrow Y$ is in the projected functional dependencies on R_j from R with Σ.
- Therefore, the union of all projected functional dependencies is equal to the minimal cover Σ_C.
- Since $\Sigma_C \equiv \Sigma$, the decomposition is guaranteed dependency-preserving by design.

```
postgres=# exit
Press any key to continue . . .
```