# Database

## Theory

### Third Normal Forms

# Motivation

- › Schema
- Decomposition
- Dependencies
- Idea

## Schema

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

Is $R$ with $\Sigma$ in **BCNF**?

# Motivation

› Schema
Decomposition
Dependencies
Idea

## Schema

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

Is $R$ with $\Sigma$ 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***.

# Motivation

## Decomposition

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

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

# Motivation

## Decomposition

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

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 $\Sigma|_{R_1} = \{ \{C\} \rightarrow \{B\} \}$ *($R_1$ is in BCNF w.r.t. $\Sigma|_{R_1}$)*

   $R_2 = \{A,C\}$ with $\Sigma|_{R_2} = \varnothing$ *($R_2$ is in BCNF w.r.t. $\Sigma|_{R_2}$)*

# Motivation

Schema
Decomposition
❱ Dependencies
Idea

## Dependencies

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

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

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

# Motivation

## Dependencies

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

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

# Motivation

## Idea

### 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.

# Third Normal Form

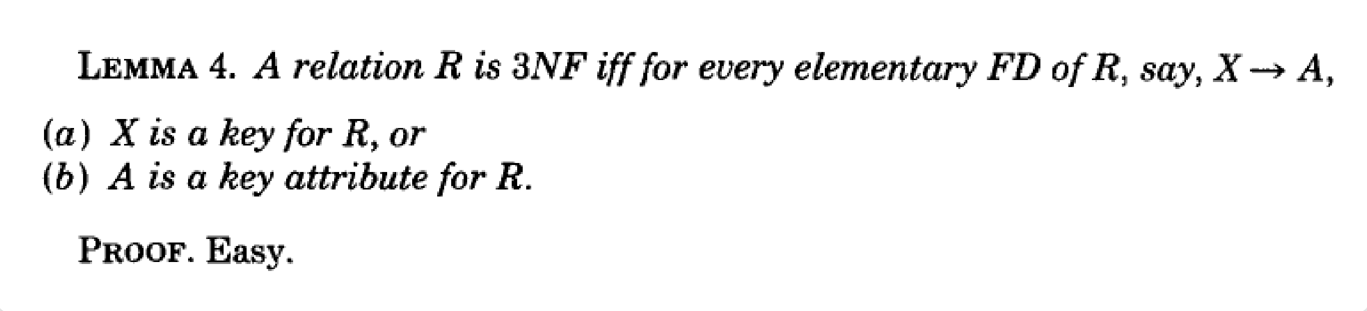
## 3NF

### Theorem

**Third Normal Form**

A relation $R$ with a set of functional dependencies $\Sigma$ 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, $X \rightarrow A$,*

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

PROOF. Easy.

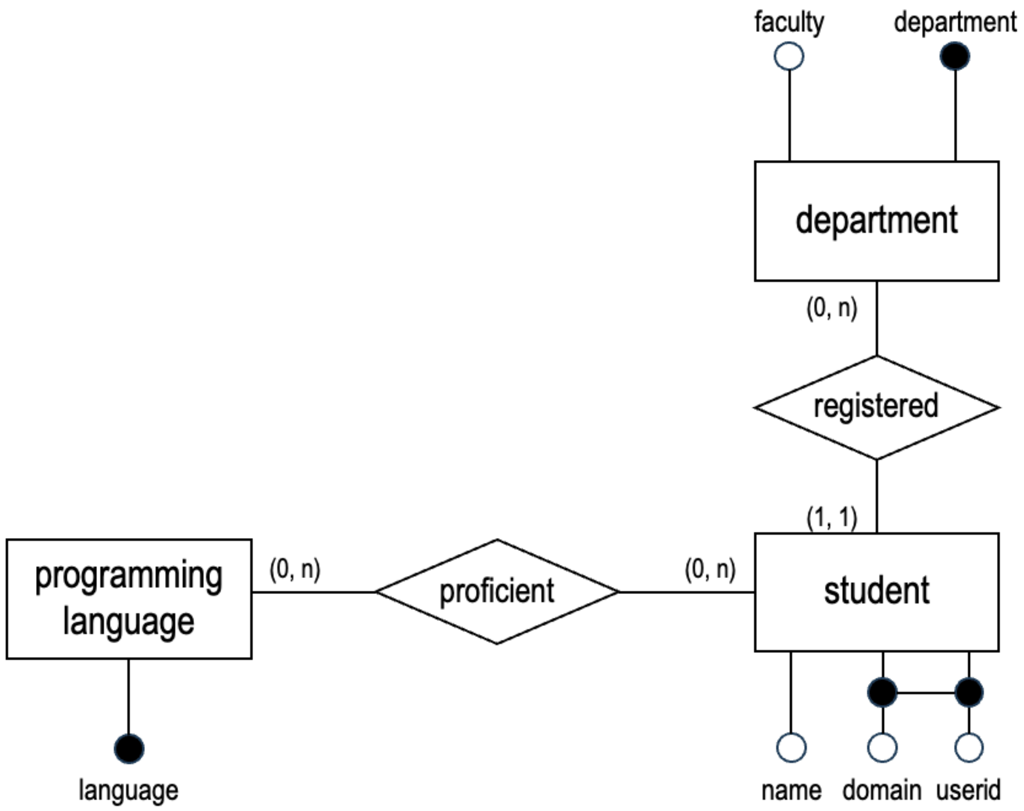
**Note**

For relation $R$ **before decomposition**, it is sufficient only to look at $\Sigma$.

# Third Normal Form

## Example

### ERD



**Issue #1**

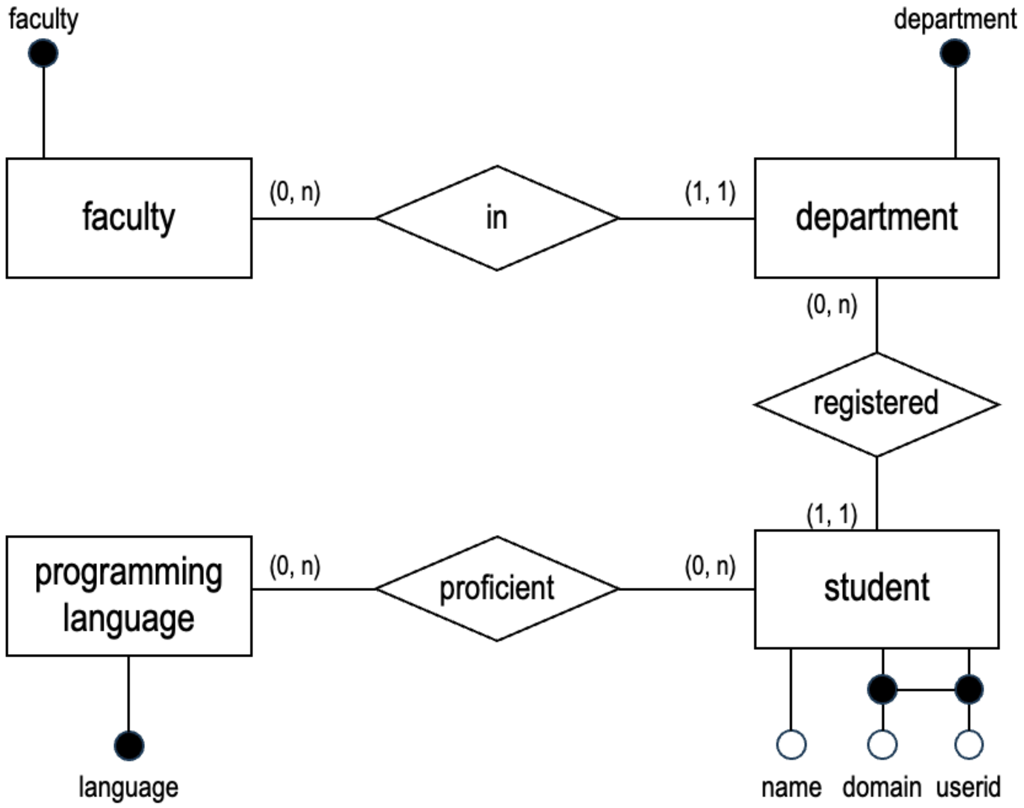
What if we want to know which faculty has **no department**?

*E.g. It is a new department.*

# Third Normal Form

## Example

ERD



### Issue #2

What if each faculty has its own domain?

{faculty} → {domain}

{domain} → {faculty}

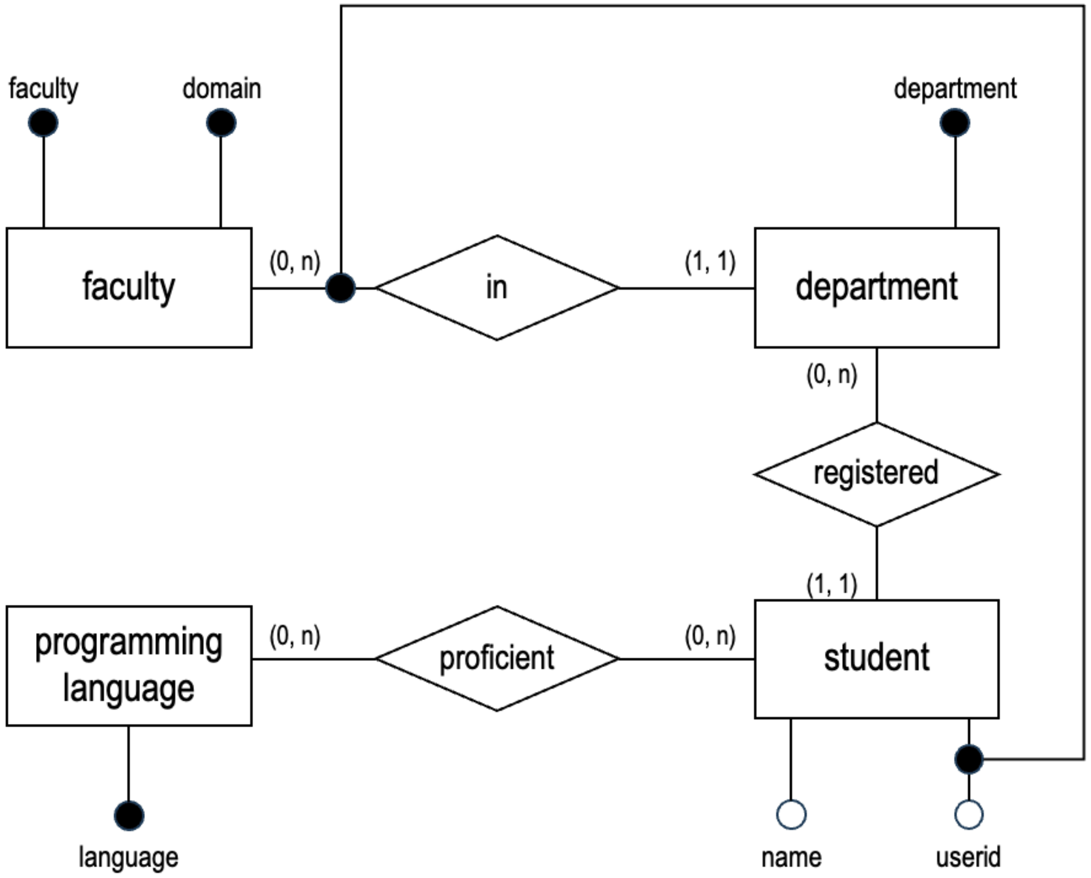
Also, student should still be uniquely identified by

{domain, userid}

# Third Normal Form

## Example

### ERD



The Strange Case of Far Away Dominant Entity

# Third Normal Form

## Example

### 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 |

$$\Sigma|_{R_1} = \{$$

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

$$\{D\} \rightarrow \{E\},$$

$$\{E\} \rightarrow \{C\},$$

$$\{C\} \rightarrow \{E\}$$

$$\}$$

# Third Normal Form

3NF
❱ Example
*ERD*
*Table*
Algorithm
Subsumption

## Example

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 |

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

### Candidate Keys

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

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

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

# Third Normal Form

## Example

### 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 |

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

### Issue

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

### Candidate Keys

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

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

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

# Third Normal Form

## Algorithm

Synthesis

### 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

  $$R_i = X \cup Y$$

  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.

**"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.

# Algorithm

## Synthesis

### 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**.

# Third Normal Form

## Algorithm

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)*.

### Theorem 9

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

### BCNF?

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

# Algorithm

## 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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

# Third Normal Form

## Algorithm

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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

### Steps

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

# Third Normal Form

## Algorithm

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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

### Steps

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

# Third Normal Form

## Algorithm

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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

### Steps

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

# Third Normal Form

## Algorithm

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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

### Steps

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

# Third Normal Form

## Algorithm

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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

### Steps

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

# Third Normal Form

## Algorithm

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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

### Steps

| | |
|---|---|
| 1. Compute **candidate keys**. | $\{A, B\}$ and $\{A, C\}$ |
| 2. Compute **minimal cover** $\Sigma_C$ of Σ. | $\Sigma_C = \{ \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{C\} \rightarrow \{E\} \}$ |
| 3. Compute **canonical cover** $\Sigma_D$ 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 $R_3 = \{A, B\}$

# Third Normal Form

## Algorithm

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

# Third Normal Form

## Algorithm

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 $\Sigma$ into a lossless-join, dependency-preserving decomposition in **3NF**.

### Alternative Answer

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

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

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

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

# Third Normal Form

3NF
Example
Algorithm
❱ Subsumption
*Remove*
*Keep*

## Subsumption

### Remove

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

## Schema

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

);
```

# Third Normal Form

## Subsumption

### 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)

);
```

### Only {C} → {B}

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


);
```

### Both?

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

);
```

# Third Normal Form

| A | B | C |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 1 |

3NF
Example
Algorithm
❱ Subsumption
*Remove*
***Keep***

## Subsumption

### 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}

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

);
```

### Only {C} → {B}

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


);
```

### Both?

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

);
```

# Third Normal Form

| A | B | C |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 2 |

3NF
Example
Algorithm
❯ Subsumption
*Remove*
*Keep*

## Subsumption

### 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}

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


);
```

#### Only {C} → {B}

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



);
```

#### Both?

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


);
```

# Third Normal Form

| A | B | C |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 1 |

## Subsumption

### 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}

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


);
```

### Only {C} → {B}

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



);
```

### Both?

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


);
```

# Third Normal Form

3NF
Example
Algorithm
❯ Subsumption
*Remove*
*Keep*

## Subsumption

### 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 $R_1 = \{A, B, C\}$.

### R2(B, C)

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

);
```

### 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)
);
```

# Third Normal Form

## Subsumption

### 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 $R_1 = \{A, B, C\}$.

### R2(B, C)

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

);
```

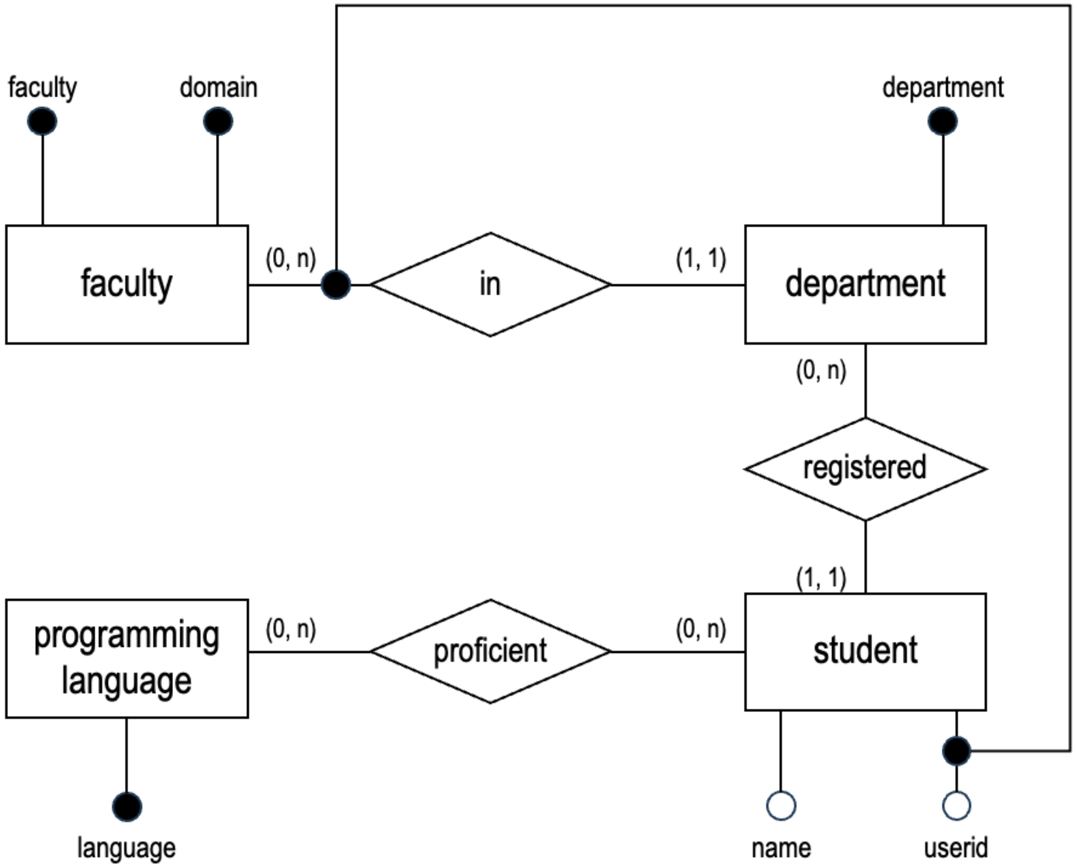
### R1(A, B, C)

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

# Back to Our Case

## Case

### ERD



# Back to Our Case

## Case

### Schema

### Tables

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

# Back to Our Case

## Case

### 3NF

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

### Projected Functional Dependency

- $\{\texttt{userid}, \texttt{faculty}\} \rightarrow \{\texttt{name}, \texttt{department}\}$
- $\{\texttt{department}\} \rightarrow \{\texttt{faculty}\}$

# Back to Our Case

## Case

3NF

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

### Projected Functional Dependency

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

### Candidate Keys

{userid, faculty} and {userid, department}

# Back to Our Case

## Case

### 3NF

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

### Projected Functional Dependency

- {userid, faculty} → {name, department} — {userid, faculty} is **superkey**
- {department} → {faculty} — faculty is a **prime attribute**

### Candidate Keys

{userid, faculty} and {userid, department}

# Back to Our Case

- **Case**
  - *ERD*
  - *Schema*
  - *3NF*
  - ***BCNF***
- Decomposition

## Case

BCNF

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

### Projected Functional Dependency

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

### Candidate Keys

{userid, faculty} and {userid, department}

# Back to Our Case

## Case

### BCNF

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

#### Projected Functional Dependency

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

#### BCNF Decomposition

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

# Back to Our Case

## Case

### BCNF

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

### Projected Functional Dependency

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

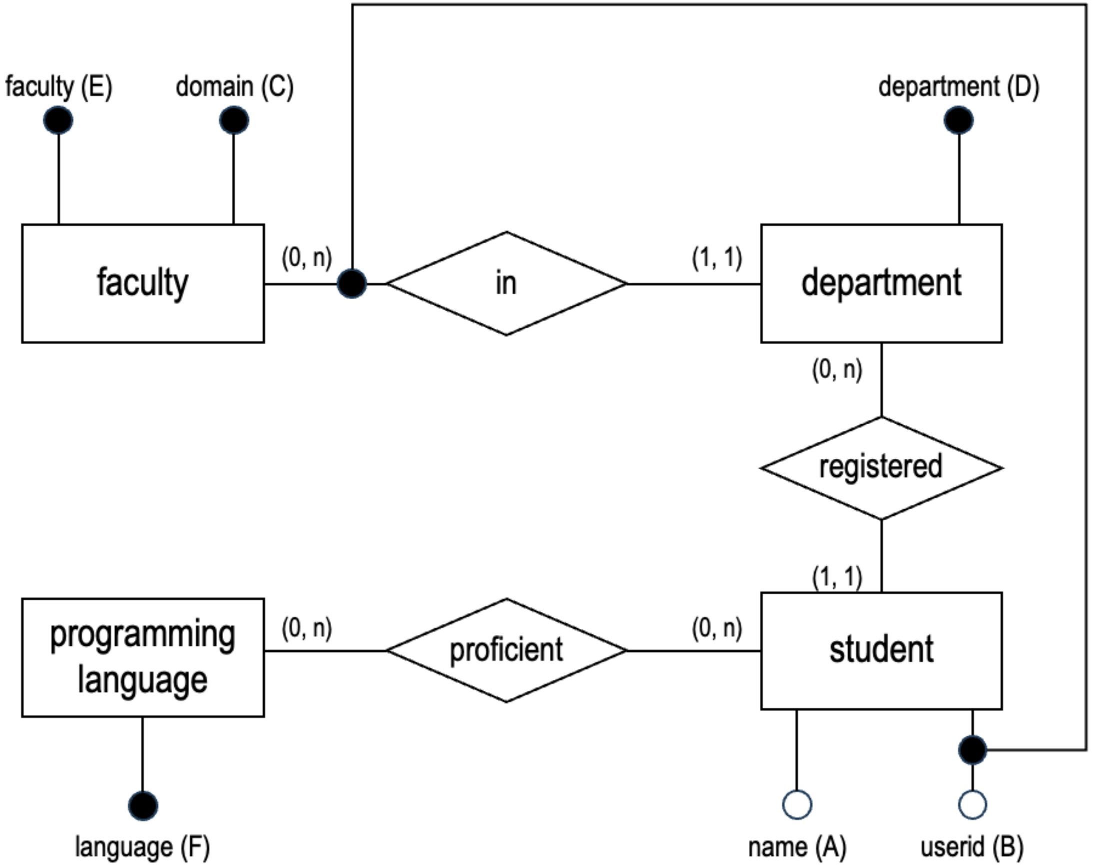
### BCNF Decomposition

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

# Back to Our Case

## Decomposition

FD



### Mapping

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

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

# Back to Our Case

## Decomposition

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?

# Back to Our Case

### Canonical Cover

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

### Candidate Keys

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

## Decomposition

BCNF

### Using {B, C} → {A, D} on R

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

# Back to Our Case

### Canonical Cover

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

### Candidate Keys

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

## Decomposition

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

# Back to Our Case

### Canonical Cover

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

### Candidate Keys

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

## Decomposition

BCNF

### 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 $\Sigma|_{R_4} = \varnothing$

# Back to Our Case

### Canonical Cover

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

### Candidate Keys

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

## Decomposition

BCNF

### Using {B, C} → {A, D} on R

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

### 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 $\Sigma|_{R_4} = \emptyset$

# Back to Our Case

### Canonical Cover

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

### Candidate Keys

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

Case
- Decomposition
  - *FD*
  - *Preliminary*
  - *BCNF*
  - *3NF*

## Decomposition

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 $\Sigma|_{R_6} = \{ \{D\} \rightarrow \{C\} \}$

# Back to Our Case

## Canonical Cover

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

## Candidate Keys

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

## Decomposition

BCNF

### Using {B, C} → {A, D} on R

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

### Using {D} → {C, E} on R1

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

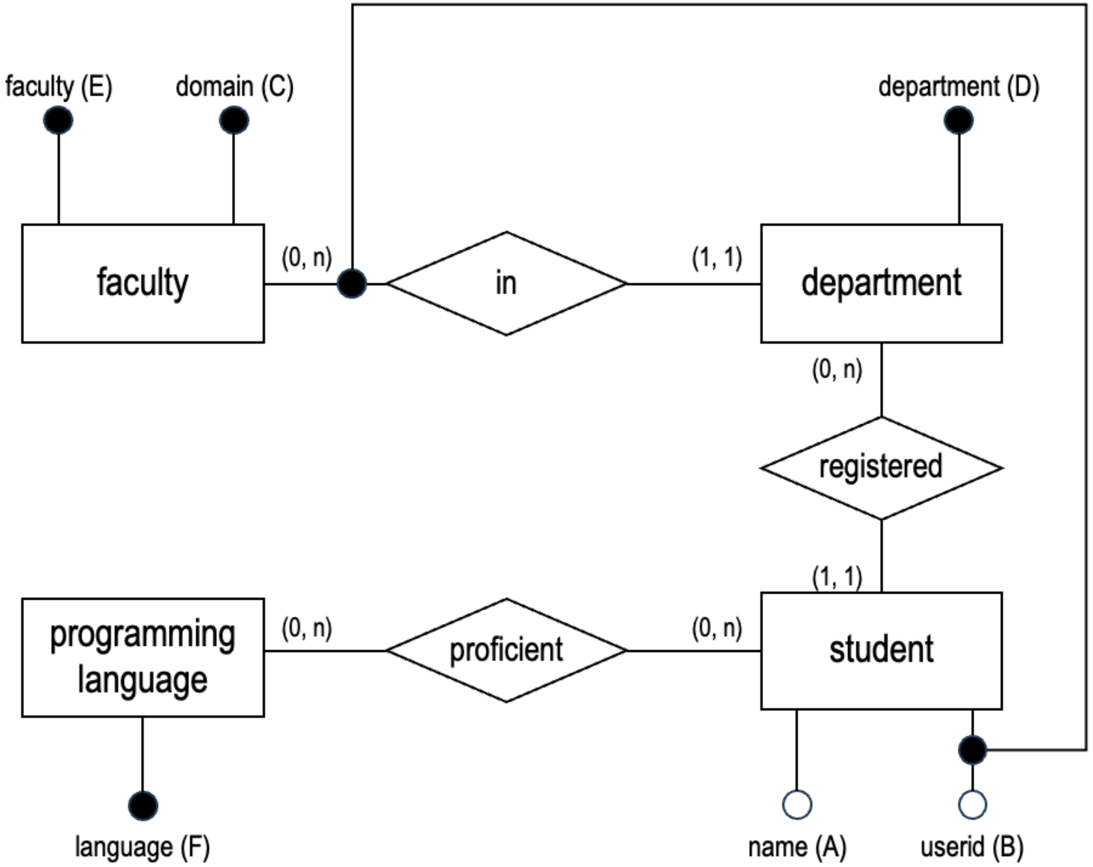
### Using {C} → {E} on R3

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

# Back to Our Case

## Decomposition

BCNF



### Fragments

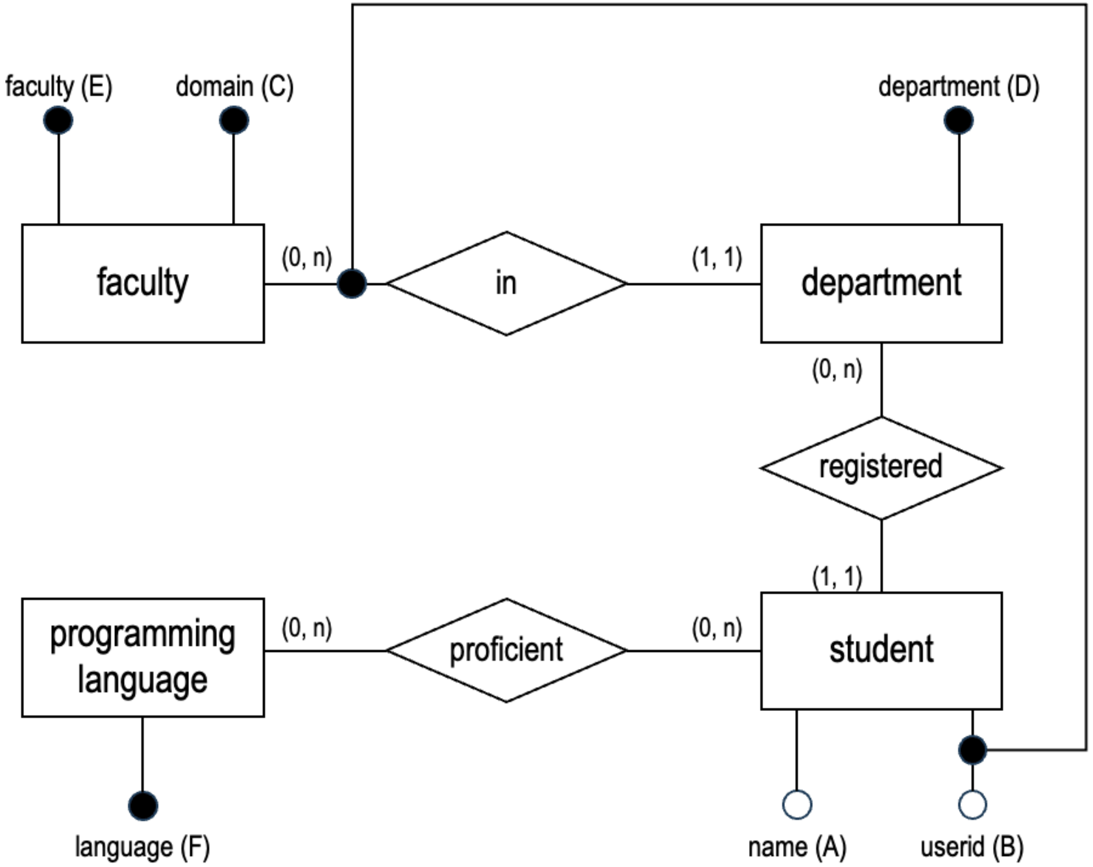
- $R_2 = \{B, C, F\}$ $\varnothing$
- $R_4 = \{A, B, D\}$ $\varnothing$
- $R_5 = \{C, E\}$ $\{\{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}\}$
- $R_6 = \{C, D\}$ $\{\{D\} \rightarrow \{C\}\}$

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

# Back to Our Case

## Decomposition

### BCNF



**Fragments**

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

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

# Back to Our Case

### Canonical Cover

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

### Candidate Keys

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

Case
- Decomposition
  - FD
  - Preliminary
  - BCNF
  - 3NF

## Decomposition

3NF

### Synthesize

- $R_1 = \{C, E\}$ with $\Sigma|_{R_1} = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\} \}$ *(can we really subsume?)*
- $R_2 = \{D, E\}$ with $\Sigma|_{R_2} = \{ \{D\} \rightarrow \{E\} \}$
- $R_3 = \{A, B, C, D\}$ with $\Sigma|_{R_3} = \{ \{B, C\} \rightarrow \{A, D\} \}$

# Back to Our Case

## Canonical Cover

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

## Candidate Keys

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

## Decomposition

3NF

### Synthesize

- $R_1 = \{C, E\}$ with $\Sigma|_{R_1} = \{ \{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\} \}$ *(can we really subsume?)*
- $R_2 = \{D, E\}$ with $\Sigma|_{R_2} = \{ \{D\} \rightarrow \{E\} \}$
- $R_3 = \{A, B, C, D\}$ with $\Sigma|_{R_3} = \{ \{B, C\} \rightarrow \{A, D\} \}$

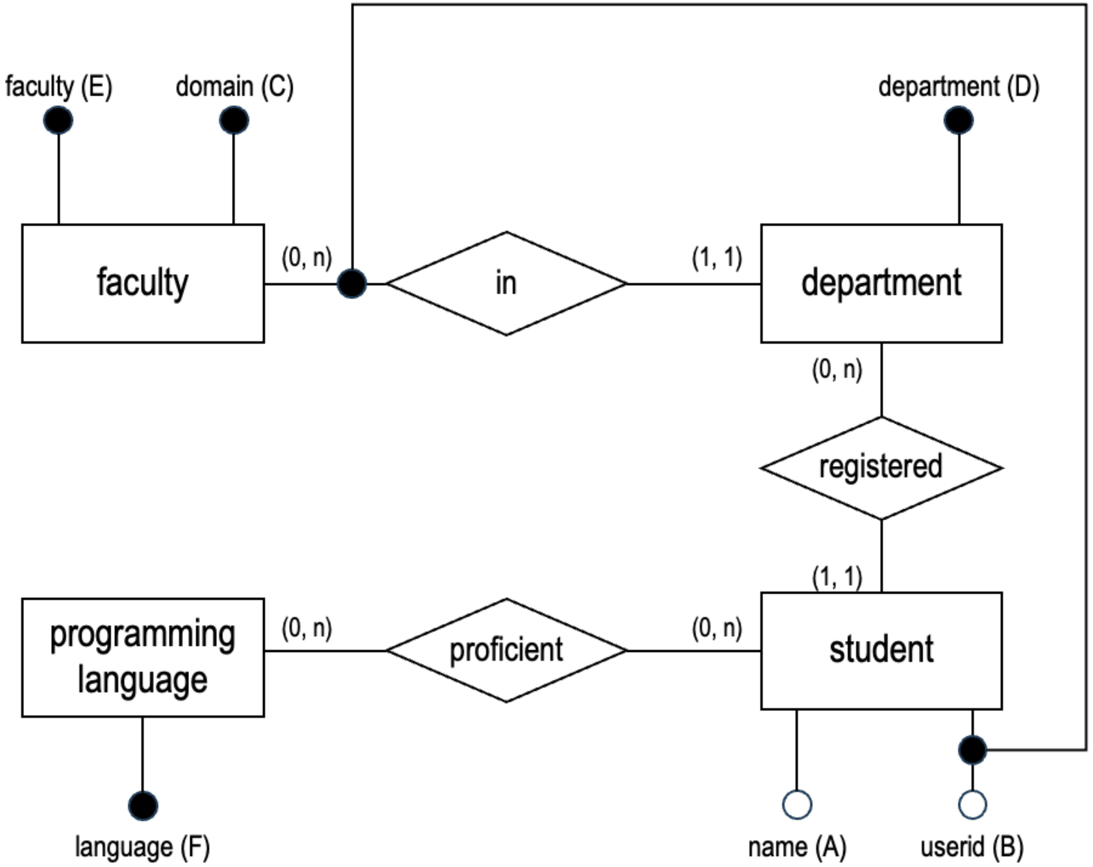
### Add Key

- $R_4 = \{B, C, F\}$ with $\Sigma|_{R_4} = \varnothing$

# Back to Our Case

## Decomposition

3NF



### Fragments

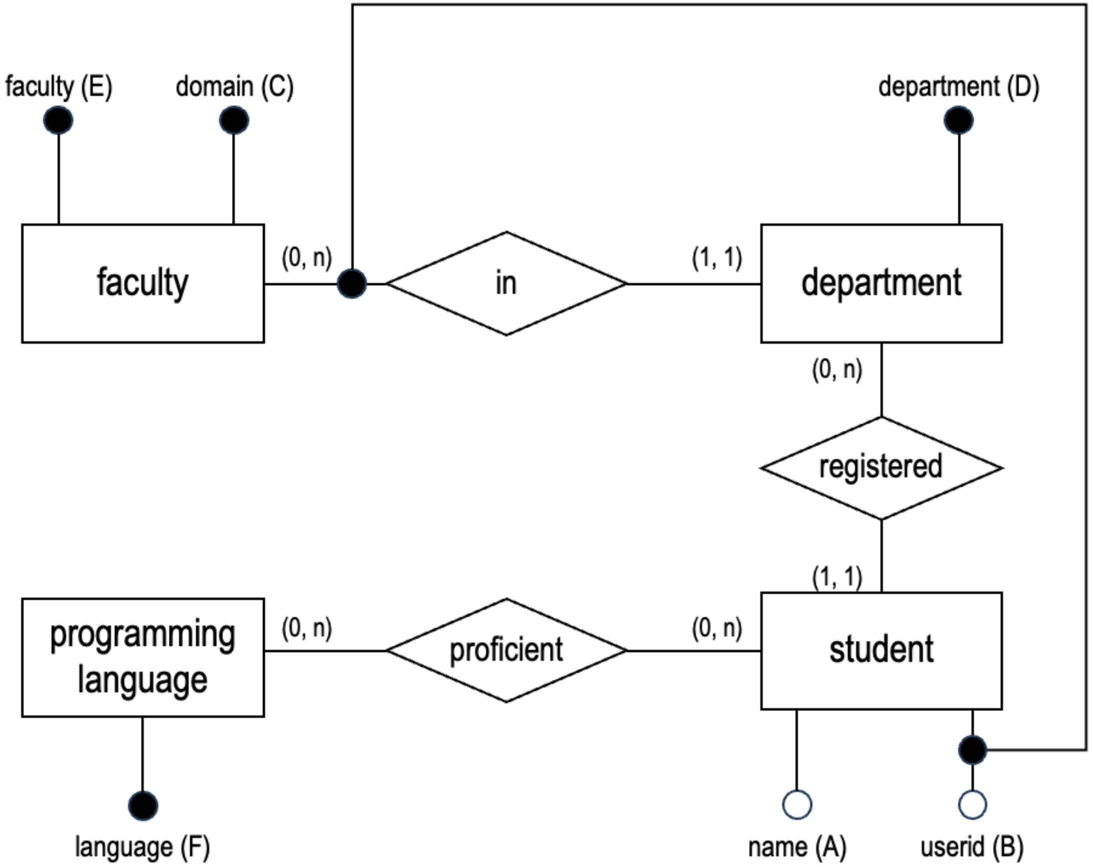
- $R_1 = \{C, E\}$ $\{\{C\} \rightarrow \{E\}, \{E\} \rightarrow \{C\}\}$
- $R_2 = \{D, E\}$ $\{\{D\} \rightarrow \{E\}\}$
- $R_3 = \{A, B, C, D\}$ $\{\{B,C\} \rightarrow \{A,D\}\}$
- $R_4 = \{B, C, F\}$ $\varnothing$

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

# Back to Our Case

## Decomposition

### 3NF



**Fragments**

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

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

# End of Story?

## Normal Forms

### Theorem 10

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

### 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.

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

# Proof Sketch

## Only for Reading ; Not Tested

# Proof Sketch

## Theorem #9

### Proof

We will focus on dependency-preserving decomposition here.

- Note that we start from a minimal cover $\Sigma_C$.

  Since $\Sigma_C$ is a minimal cover of $\Sigma$, we know $\Sigma_C \equiv \Sigma$.
- For each functional dependencies $X \rightarrow Y$ in $\Sigma$, 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 $\Sigma$.

  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 $\Sigma$.
- Therefore, the union of all projected functional dependencies is equal to the minimal cover $\Sigma_C$.
- Since $\Sigma_C \equiv \Sigma$, the decomposition is guaranteed dependency-preserving by design.

```
postgres=# \q
Press any key to continue . . .
```