Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

Databases

Malte Luttermann – Institute for Humanities-Centered Artificial Intelligence (CHAI)

November 13, 2025

Version from November 13, 2025

Topics for Today

- Decomposition procedure
- Third normal form (3NF)

Decomposition into Boyce-Codd Normal Form (BCNF)

Are the following statements correct?

- Every relational schema can be decomposed into BCNF relations in a lossless way.
- Every relational schema can be decomposed into BCNF relations in a dependency preserving way.

Decomposition into Boyce-Codd Normal Form (BCNF)

Are the following statements correct?

- Every relational schema can be decomposed into BCNF relations in a lossless way. ✓
- Every relational schema can be decomposed into BCNF relations in a dependency preserving way.

Decomposition into Boyce-Codd Normal Form (BCNF)

Are the following statements correct?

- Every relational schema can be decomposed into BCNF relations in a lossless way. ✓
- Every relational schema can be decomposed into BCNF relations in a dependency preserving way. ✗

Decomposition Procedure

Let the schema $\mathcal{R} = \{A, B, C, D\}$ with the functional dependencies

$$\{D\} \to \{B, C\}, \quad \{A, B, C\} \to \{D\}$$

be given. Apply the decomposition procedure to decompose $\mathcal{R}$.

```
decomposition($\mathcal{R}$, $F$)
  Z := $\{\mathcal{R}\}$
  while there exists a schema $\mathcal{R}_i \in$ Z that is not in BCNF:
    let $\alpha \to \beta$ be a non-trivial FD in $F$ holding for $\mathcal{R}_i$ such that
      1. $\alpha \cap \beta = \emptyset$ and
      2. $\alpha \to \mathcal{R}_i$ does not hold
    decompose $\mathcal{R}_i$ into $\mathcal{R}_{i_1} := \alpha \cup \beta$ and $\mathcal{R}_{i_2} := \mathcal{R}_i \setminus \beta$
    Z := Z $\setminus \{\mathcal{R}_i\} \cup \{\mathcal{R}_{i_1}, \mathcal{R}_{i_2}\}$
  return Z
```

Decomposition Procedure

Schema $\mathcal{R} = \{A, B, C, D\}$, FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

```
decomposition(ℛ, F)
  Z := {ℛ}
  while there exists a schema ℛ_i ∈ Z that is not in BCNF:
    let α → β be a non-trivial FD in F holding for ℛ_i such that
      1. α ∩ β = ∅ and
      2. α → ℛ_i does not hold
    decompose ℛ_i into ℛ_{i_1} := α ∪ β and ℛ_{i_2} := ℛ_i \ β
    Z := Z \{ℛ_i} ∪ {ℛ_{i_1}, ℛ_{i_2}}
  return Z
```

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and
(2) for each functional dependency $\alpha \to \beta$ that holds in $\mathcal{R}$, it holds that
   (a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or
   (b) α is a super key of $\mathcal{R}$.

Decomposition Procedure

Schema $\mathcal{R} = \{A, B, C, D\}$, FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

```
decomposition(R, F)
  Z := {R}
  while there exists a schema R_i ∈ Z that is not in BCNF:
    let α → β be a non-trivial FD in F holding for R_i such that
      1. α ∩ β = ∅ and
      2. α → R_i does not hold
    decompose R_i into R_i1 := α ∪ β and R_i2 := R_i \ β
    Z := Z \{R_i} ∪ {R_i1, R_i2}
  return Z
```

- FD $\{D\} \to \{B, C\}$ with $\{D\} \cap \{B, C\} = \emptyset$ and $\{D\} \to \mathcal{R}$ does not hold

Decomposition Procedure

Schema $\mathcal{R} = \{A, B, C, D\}$, FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

```
decomposition(R, F)
  Z := {R}
  while there exists a schema R_i ∈ Z that is not in BCNF:
    let α → β be a non-trivial FD in F holding for R_i such that
      1. α ∩ β = ∅ and
      2. α → R_i does not hold
    decompose R_i into R_i1 := α ∪ β and R_i2 := R_i \ β
    Z := Z \{R_i} ∪ {R_i1, R_i2}
  return Z
```

- FD $\{D\} \to \{B, C\}$ with $\{D\} \cap \{B, C\} = \emptyset$ and $\{D\} \to \mathcal{R}$ does not hold
- Decompose $\mathcal{R} = \{A, B, C, D\}$ into $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$

Decomposition Procedure

- Decomposition: $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$
- FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$
- Check whether $\mathcal{R}_1$ and $\mathcal{R}_2$ are in BCNF

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to \beta$ that holds in $\mathcal{R}$, it holds that

   (a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or

   (b) α is a super key of $\mathcal{R}$.

Decomposition Procedure

- Decomposition: $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$
- FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$
- Check whether $\mathcal{R}_1$ and $\mathcal{R}_2$ are in BCNF
- $\mathcal{R}_1 = \{B, C, D\}$ in BCNF?

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to \beta$ that holds in $\mathcal{R}$, it holds that

  (a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or

  (b) α is a super key of $\mathcal{R}$.

Decomposition Procedure

- Decomposition: $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$
- FDs $\{D\} \rightarrow \{B, C\}$ and $\{A, B, C\} \rightarrow \{D\}$
- Check whether $\mathcal{R}_1$ and $\mathcal{R}_2$ are in BCNF
- $\mathcal{R}_1 = \{B, C, D\}$ in BCNF? Yes. (the only nontrivial FD is $\{D\} \rightarrow \{B, C\}$)

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \rightarrow \beta$ that holds in $\mathcal{R}$, it holds that

(a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or

(b) α is a super key of $\mathcal{R}$.

Decomposition Procedure

- Decomposition: $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$
- FDs $\{D\} \rightarrow \{B, C\}$ and $\{A, B, C\} \rightarrow \{D\}$
- Check whether $\mathcal{R}_1$ and $\mathcal{R}_2$ are in BCNF
- $\mathcal{R}_1 = \{B, C, D\}$ in BCNF? Yes. (the only nontrivial FD is $\{D\} \rightarrow \{B, C\}$)
- $\mathcal{R}_2 = \{A, D\}$ in BCNF?

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \rightarrow \beta$ that holds in $\mathcal{R}$, it holds that

  (a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or

  (b) α is a super key of $\mathcal{R}$.

Decomposition Procedure

- Decomposition: $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$
- FDs $\{D\} \rightarrow \{B, C\}$ and $\{A, B, C\} \rightarrow \{D\}$
- Check whether $\mathcal{R}_1$ and $\mathcal{R}_2$ are in BCNF
- $\mathcal{R}_1 = \{B, C, D\}$ in BCNF? Yes. (the only nontrivial FD is $\{D\} \rightarrow \{B, C\}$)
- $\mathcal{R}_2 = \{A, D\}$ in BCNF? Yes. (no nontrivial FDs hold and the only key is $\{A, D\}$)

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \rightarrow \beta$ that holds in $\mathcal{R}$, it holds that

   (a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or

   (b) α is a super key of $\mathcal{R}$.

Decomposition Procedure

Schema $\mathcal{R} = \{A, B, C, D\}$, FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

```
decomposition($\mathcal{R}$, $F$)
  Z := {$\mathcal{R}$}
  while there exists a schema $\mathcal{R}_i \in$ Z that is not in BCNF:
    let $\alpha \to \beta$ be a non-trivial FD in $F$ holding for $\mathcal{R}_i$ such that
      1. $\alpha \cap \beta = \emptyset$ and
      2. $\alpha \to \mathcal{R}_i$ does not hold
    decompose $\mathcal{R}_i$ into $\mathcal{R}_{i_1} := \alpha \cup \beta$ and $\mathcal{R}_{i_2} := \mathcal{R}_i \setminus \beta$
    Z := Z $\setminus \{\mathcal{R}_i\} \cup \{\mathcal{R}_{i_1}, \mathcal{R}_{i_2}\}$
  return Z
```

- Decomposition: $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$

Decomposition Procedure

Schema $\mathcal{R} = \{A, B, C, D\}$, FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

```
decomposition(𝓡, F)
  Z := {𝓡}
  while there exists a schema 𝓡_i ∈ Z that is not in BCNF:
    let α → β be a non-trivial FD in F holding for 𝓡_i such that
      1. α ∩ β = ∅ and
      2. α → 𝓡_i does not hold
    decompose 𝓡_i into 𝓡_{i_1} := α ∪ β and 𝓡_{i_2} := 𝓡_i \ β
    Z := Z \{𝓡_i} ∪ {𝓡_{i_1}, 𝓡_{i_2}}
  return Z
```

- Decomposition: $\mathcal{R}_1 = \{B, C, D\}$ and $\mathcal{R}_2 = \{A, D\}$
- Lossless but not dependency preserving (the FD $\{A, B, C\} \to \{D\}$ cannot be associated with a sub-relation)!

Normal Forms – Third Normal Form

Let the schema $\mathcal{R} = \{A, B, C, D\}$ with the functional dependencies

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

be given. Is $\mathcal{R}$ in 3NF?

Normal Forms – Third Normal Form

Let the schema $\mathcal{R} = \{A, B, C, D\}$ with the functional dependencies

$$\{D\} \to \{B, C\}, \quad \{A, B, C\} \to \{D\}$$

be given. Is $\mathcal{R}$ in 3NF?

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and
(2) for each functional dependency $\alpha \to B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that
  (a) $B \in \alpha$ (i.e., the dependency is trivial), or
  (b) α is a super key of $\mathcal{R}$, or
  (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{A, B, C, D\}$ with FDs $\{D\} \rightarrow \{B, C\}$ and $\{A, B, C\} \rightarrow \{D\}$.

- Candidate keys: $\{A, B, C\}$, $\{A, D\}$

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \rightarrow B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

   (a) $B \in \alpha$ (i.e., the dependency is trivial), or

   (b) α is a super key of $\mathcal{R}$, or

   (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{A, B, C, D\}$ with FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

- Candidate keys: $\{A, B, C\}$, $\{A, D\}$
- All attributes are prime (i.e., contained in a candidate key), so $\mathcal{R}$ is in 2NF

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

  (a) $B \in \alpha$ (i.e., the dependency is trivial), or

  (b) α is a super key of $\mathcal{R}$, or

  (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{A, B, C, D\}$ with FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

- Candidate keys: $\{A, B, C\}$, $\{A, D\}$
- All attributes are prime (i.e., contained in a candidate key), so $\mathcal{R}$ is in 2NF
- FD $\{A, B, C\} \to \{D\}$: It holds that $\{A, B, C\}$ is a super key and D is prime

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

  (a) $B \in \alpha$ (i.e., the dependency is trivial), or

  (b) α is a super key of $\mathcal{R}$, or

  (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{A, B, C, D\}$ with FDs $\{D\} \to \{B, C\}$ and $\{A, B, C\} \to \{D\}$.

- Candidate keys: $\{A, B, C\}$, $\{A, D\}$
- All attributes are prime (i.e., contained in a candidate key), so $\mathcal{R}$ is in 2NF
- FD $\{A, B, C\} \to \{D\}$: It holds that $\{A, B, C\}$ is a super key and D is prime

$\mathcal{R} = \{A, B, C, D\}$ is in 3NF!

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

  (a) $B \in \alpha$ (i.e., the dependency is trivial), or

  (b) α is a super key of $\mathcal{R}$, or

  (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Normal Forms – Third Normal Form

Let the schema $\mathcal{R} = \{[\mathit{Street}, \mathit{Place}, \mathit{FState}, \mathit{PostCode}]\}$ with the FDs

$$\{\mathit{PostCode}\} \to \{\mathit{Place}, \mathit{FState}\}, \quad \{\mathit{Street}, \mathit{Place}, \mathit{FState}\} \to \{\mathit{PostCode}\}$$

be given. Is $\mathcal{R}$ in 3NF?

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

   (a) $B \in \alpha$ (i.e., the dependency is trivial), or

   (b) α is a super key of $\mathcal{R}$, or

   (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Normal Forms – Third Normal Form

Let the schema $\mathcal{R} = \{[\mathit{Street}, \mathit{Place}, \mathit{FState}, \mathit{PostCode}]\}$ with the FDs

$$\{\mathit{PostCode}\} \to \{\mathit{Place}, \mathit{FState}\}, \quad \{\mathit{Street}, \mathit{Place}, \mathit{FState}\} \to \{\mathit{PostCode}\}$$

be given. Is $\mathcal{R}$ in 3NF? Yes.

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

- (a) $B \in \alpha$ (i.e., the dependency is trivial), or
- (b) α is a super key of $\mathcal{R}$, or
- (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Normal Forms – Third Normal Form

Let the schema $\mathcal{R} = \{[Street, Place, FState, PostCode]\}$ with the FDs

$$\{PostCode\} \rightarrow \{Place, FState\}, \quad \{Street, Place, FState\} \rightarrow \{PostCode\}$$

be given. Is $\mathcal{R}$ in BCNF?

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \rightarrow \beta$ that holds in $\mathcal{R}$, it holds that

   (a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or

   (b) α is a super key of $\mathcal{R}$.

Normal Forms – Third Normal Form

Let the schema $\mathcal{R} = \{[\mathit{Street}, \mathit{Place}, \mathit{FState}, \mathit{PostCode}]\}$ with the FDs

$$\{\mathit{PostCode}\} \to \{\mathit{Place}, \mathit{FState}\}, \quad \{\mathit{Street}, \mathit{Place}, \mathit{FState}\} \to \{\mathit{PostCode}\}$$

be given. Is $\mathcal{R}$ in BCNF? No.

A relation $\mathcal{R}$ is in BCNF if

(1) $\mathcal{R}$ is in 2NF (each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to \beta$ that holds in $\mathcal{R}$, it holds that

  (a) $\beta \subseteq \alpha$ (i.e., the dependency is trivial), or

  (b) α is a super key of $\mathcal{R}$.

Normal Forms – Third Normal Form

Let the schema $\mathcal{R} = \{[\mathit{Street}, \mathit{Place}, \mathit{FState}, \mathit{PostCode}]\}$ with the FDs

$$\{\mathit{PostCode}\} \rightarrow \{\mathit{Place}, \mathit{FState}\}, \quad \{\mathit{Street}, \mathit{Place}, \mathit{FState}\} \rightarrow \{\mathit{PostCode}\}$$

be given. Is $\mathcal{R}$ in BCNF? No.

Problem

Decomposition into BCNF relations in a dependency preserving way is not always possible!

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{[Street, Place, FState, PostCode]\}$ with FDs

$$\{PostCode\} \rightarrow \{Place, FState\}, \quad \{Street, Place, FState\} \rightarrow \{PostCode\}.$$

Candidate keys: $\{Street, Place, FState\}$ and $\{Street, PostCode\}$.

```sql
create table PostCodeDirectory (
  Street ...,
  Place ...,
  FState ...,
  PostCode ...,
  primary key (Street, Place, FState),
  unique (Street, PostCode)
);
```

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{[Street, Place, FState, PostCode]\}$ with FDs

$$\{PostCode\} \rightarrow \{Place, FState\}, \quad \{Street, Place, FState\} \rightarrow \{PostCode\}.$$

Candidate keys: $\{Street, Place, FState\}$ and $\{Street, PostCode\}$.

```
create table PostCodeDirectory (
  Street ...,
  Place ...,
  FState ...,
  PostCode ...,
  primary key (Street, Place, FState),
  unique (Street, PostCode)
);
```

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{[\mathit{Street}, \mathit{Place}, \mathit{FState}, \mathit{PostCode}]\}$ with FDs

$$\{\mathit{PostCode}\} \to \{\mathit{Place}, \mathit{FState}\}, \quad \{\mathit{Street}, \mathit{Place}, \mathit{FState}\} \to \{\mathit{PostCode}\}.$$

Candidate keys: $\{\mathit{Street}, \mathit{Place}, \mathit{FState}\}$ and $\{\mathit{Street}, \mathit{PostCode}\}$.

```sql
create table PostCodeDirectory (
  Street ...,
  Place ...,
  FState ...,
  PostCode ...,
  primary key (Street, Place, FState),
  unique (Street, PostCode),
  unique (PostCode)
);
```

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{[Street, Place, FState, PostCode]\}$ with FDs

$$\{PostCode\} \rightarrow \{Place, FState\}, \quad \{Street, Place, FState\} \rightarrow \{PostCode\}.$$

Candidate keys: $\{Street, Place, FState\}$ and $\{Street, PostCode\}$.

```
create table PostCodeDirectory (
  Street ...,
  Place ...,
  FState ...,
  PostCode ...,
  primary key (Street, Place, FState),
  unique (Street, PostCode),
  unique (PostCode, Place, FState)
);
```

Normal Forms – Third Normal Form

Schema $\mathcal{R} = \{[Street, Place, FState, PostCode]\}$ with FDs

$$\{PostCode\} \rightarrow \{Place, FState\}, \quad \{Street, Place, FState\} \rightarrow \{PostCode\}.$$

Candidate keys: $\{Street, Place, FState\}$ and $\{Street, PostCode\}$.

```
create table PostCodeDirectory (
  Street ...,
  Place ...,
  FState ...,
  PostCode ...,
  primary key (Street, Place, FState),
  unique (Street, PostCode),
  check(all x, all y: x.PostCode <> y.PostCode or
    (x.Place = y.Place and x.FState = y.FState))
);
```

Another Example

Consider $\mathcal{R} = \{[\mathit{EmployeeID}, \mathit{DeptID}, \mathit{DeptName}, \mathit{Salary}]\}$ with the FD

$$\{\mathit{DeptID}\} \rightarrow \{\mathit{DeptName}\}.$$

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \rightarrow B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

  (a) $B \in \alpha$ (i.e., the dependency is trivial), or

  (b) α is a super key of $\mathcal{R}$, or

  (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Another Example

Consider $\mathcal{R} = \{[\mathit{EmployeeID}, \mathit{DeptID}, \mathit{DeptName}, \mathit{Salary}]\}$ with the FD

$$\{\mathit{DeptID}\} \to \{\mathit{DeptName}\}.$$

$\mathcal{R}$ is not in 3NF (and hence also not in BCNF).

A relation $\mathcal{R}$ is in 3NF if

(1) $\mathcal{R}$ is in 2NF (i.e., each non-key attribute $A \in \mathcal{R}$ is fully functionally dependent on every candidate key of $\mathcal{R}$), and

(2) for each functional dependency $\alpha \to B$ with $\alpha \subseteq \mathcal{R}$ and $B \in \mathcal{R}$, it holds that

- (a) $B \in \alpha$ (i.e., the dependency is trivial), or
- (b) α is a super key of $\mathcal{R}$, or
- (c) the attribute B is contained in a candidate key of $\mathcal{R}$ (i.e., B is prime).

Another Example

Consider $\mathcal{R} = \{[\mathit{EmployeeID}, \mathit{DeptID}, \mathit{DeptName}, \mathit{Salary}]\}$ with the FD

$$\{\mathit{DeptID}\} \rightarrow \{\mathit{DeptName}\}.$$

$\mathcal{R}$ is not in 3NF (and hence also not in BCNF).

Possible remedy:

- Decompose $\mathcal{R}$ into $\mathcal{R}_1 = \{[\mathit{EmployeeID}, \mathit{DeptID}, \mathit{Salary}]\}$ and $\mathcal{R}_2 = \{[\mathit{DeptID}, \mathit{DeptName}]\}$
- Then, both $\mathcal{R}_1$ and $\mathcal{R}_2$ are in BCNF
- Whenever we need the department name, we have to perform a join (costly)!

Another Example

Consider $\mathcal{R} = \{[\mathit{EmployeeID}, \mathit{DeptID}, \mathit{DeptName}, \mathit{Salary}]\}$ with the FD

$$\{\mathit{DeptID}\} \rightarrow \{\mathit{DeptName}\}.$$

$\mathcal{R}$ is not in 3NF (and hence also not in BCNF).

Alternative: Redesign schema such that $\{\mathit{DeptID}, \mathit{DeptName}\}$ is a candidate key

- Then, $\mathcal{R}$ is in 3NF
- No decomposition required (avoids joins)
- What extra checks do we need?

Another Example

Candidate keys $\{EmployeeID\}$ and $\{DeptID, DeptName\}$, FD $\{DeptID\} \rightarrow \{DeptName\}$

```sql
CREATE TABLE Employees (
  EmpID       INT PRIMARY KEY,
  DeptID      INT NOT NULL,
  DeptName    VARCHAR(50) NOT NULL,
  Salary      DECIMAL(10,2),

  -- 1) Make DeptID a candidate key component (together with DeptName)
  CONSTRAINT uq_DeptID_DeptName UNIQUE (DeptID, DeptName),

  -- 2) Enforce the FD {DeptID} → {DeptName}
  CONSTRAINT uq_DeptID UNIQUE (DeptID)
);
```

Another Example

Candidate keys $\{EmployeeID\}$ and $\{DeptID, DeptName\}$, FD $\{DeptID\} \rightarrow \{DeptName\}$

```
CREATE TABLE Employees (
  EmpID       INT PRIMARY KEY,
  DeptID      INT NOT NULL,
  DeptName    VARCHAR(50) NOT NULL,
  Salary      DECIMAL(10,2),

  -- 1) Make DeptID a candidate key component (together with DeptName)
  CONSTRAINT uq_DeptID_DeptName UNIQUE (DeptID, DeptName),

  -- 2) Enforce the FD {DeptID} → {DeptName}
  CONSTRAINT uq_DeptID UNIQUE (DeptID)
);
```

Enforces FDs but might not be useful in an application! (Because each department can only have a single employee...)