UPPSALA
UNIVERSITET

Lecture 9 and 10 – Functional dependencies and normalization

1DL301 Database Design I

Jan Kudlicka (jan.kudlicka@it.uu.se)

Fall 2019, Period 2

VOX AULAE

INTENDED LEARNING OUTCOMES

Intended learning outcomes for this and the next lecture:

- To understand problems with poorly designed databases.
- To understand concepts of:
  - functional dependencies,
  - prime and non prime attributes,
  - normal forms (1NF, 2NF, 3NF, BCNF).
- To be able to determine normal forms of tables.
- To be able to recognize poorly designed databases and improve them by normalization.

DATABASE QUALITY BASICS

- Each table should represent a single entity or relationship type.
- Each attribute should have an appropriate data type.
- Each attribute should have a NOT NULL constraint if the value cannot be null.
- Interpretability: it should be easy to understand the meaning of tables and attributes from their names (and types).
- Primary key for each table.
- Constraining references to other entities with foreign keys.

EXERCISE 1

EmpProj(eid, lname, fname, pid, project, hours)

Note: eid = employee ID, pid = project ID

eid	lname	fname	pid	project	hours
1	Alnes	Bernt	1	Alfa	100
2	Fjelldal	Mads	3	Charlie	140
4	Longva	Victor	2	Bravo	80
7	Bakke	Alfred	1	Alfa	80
8	Vie	Tor	1	Alfa	910
8	Vie	Tor	3	Charlie	720
9	Westgaard	Sten	3	Charlie	310
10	Liseth	Rakel	3	Charlie	480
11	Norman	Emil	3	Charlie	460
12	Dyrhaug	Atle	2	Bravo	810
15	Kvien	Amalie	2	Bravo	15
15	Kvien	Amalie	3	Charlie	75
16	Tveten	Thomas	2	Bravo	20
17	Lende	Marita	1	Alfa	40

What's "wrong" with this table?

ANOMALIES AND REDUNDANCY

- **Insertion anomaly**: Cannot insert a project unless an employee is working on it. Cannot insert an employee unless assigned to a project.
- **Deletion anomaly**: If we remove all employees from a given project, the information about the project is lost. (And similarly if we remove all projects from a given employee.)
- **Update anomaly**: When some information changes, we need to update all occurrences, otherwise we will introduce **data inconsistency**.
- **Redundancy**: Same information is stored multiple times (a waste of storage and a risk of inconsistency).

NORMALIZATION

Normalization is a step-by-step process to decompose a relation schema into a set of several smaller schemas to avoid redundancy and anomaly problems.

First, we need to learn about:

- Functional dependencies
- Normal forms (we will cover only 1NF, 2NF, 3NF, BCNF)

FUNCTIONAL DEPENDENCY (FD)

Let $R(A_1, A_2, \ldots, A_n)$ denote a relation schema and $X, Y \subseteq \{A_1, A_2, \ldots, A_n\}$ be two subsets of the attributes.

X **functionally determines** Y, denoted as $X \rightarrow Y$, if for any legal instance $r(R)$ and $\forall t_1, t_2 \in r(R)$ the following holds:

$$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

We will also say that Y is functionally dependent on X.

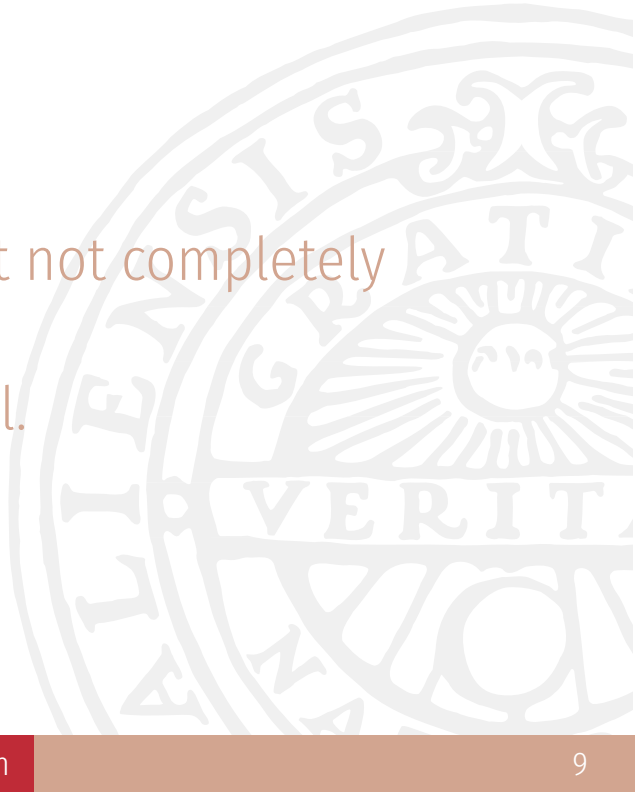
Informally: *"If you tell me the value of each attribute in X, I will be able to tell you what the value of each attribute in Y is".*

TRIVIAL AND NON-TRIVIAL FDS

$X \rightarrow Y$ is

- **trivial** if $Y \subseteq X$
- **non-trivial** if $Y \not\subseteq X$
- **completely non-trivial** if $X \cap Y = \emptyset$

Examples:
{eid, lname} $\rightarrow$ {lname} is trivial.
{eid, lname} $\rightarrow$ {lname, fname} is non-trivial but not completely non-trivial.
{eid, lname} $\rightarrow$ {fname} is completely non-trivial.



ARMSTRONG'S AXIOMS

Reflexivity:

$$Y \subseteq X \Rightarrow X \rightarrow Y$$

Augmentation:

$$X \rightarrow Y \Rightarrow XZ \rightarrow YZ$$

Note: In database theory, the union of two sets, say A and B, is customarily denoted as *AB*.

Transitivity:

$$X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$$

SECONDARY RULES

Union:

$$X \to Y \wedge X \to Z \Rightarrow X \to YZ$$

Decomposition:

$$X \to YZ \Rightarrow X \to Y \wedge X \to Z$$

Composition:

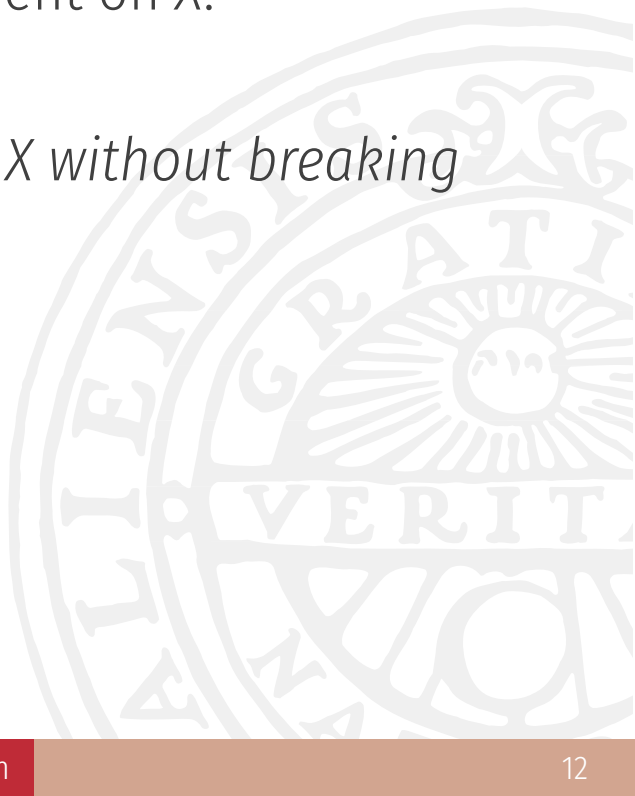
$$X \to Y \wedge Z \to W \Rightarrow XZ \to YW$$

FULL FUNCTIONAL DEPENDENCY (FFD)

A functional dependency $X \rightarrow Y$ is called a **full functional dependency** (FFD) if there is no attribute $A \in X$ such that $X \backslash \{A\} \rightarrow Y$.

We will also say that Y is fully functionally dependent on X.

Informally: *"We cannot remove any attribute from X without breaking the functional dependency."*



CLOSURE

Given $R(A_1, A_2, \ldots A_n)$, the **closure** of $X \subseteq \{A_1, A_2, \ldots, A_n\}$ is the set

$$X^+ = \{A_i \mid X \rightarrow A_i\}$$

Algorithm to determine X^+:

1. Start with $C := X$.
2. Loop through all FFDs $U \rightarrow V$:
   If $U \subseteq C$, add V to C (i.e., $C \leftarrow CV$).
3. If C changed during the last step, go to 2.
4. Return C

EXERCISE 2

Consider a relation $R(A, B, C, D, E)$ with the following FFDs:

- $A, B \rightarrow C, D, E$
- $C \rightarrow A, B, D, E$
- $E \rightarrow D$

Determine the closures $\{A\}^+, \{B\}^+, \{A, B\}^+, \{C\}^+, \{D\}^+, \{C, D\}^+$ and $\{E\}^+$.

SOME REMARKS ON FUNCTIONAL DEPENDENCIES

Functional dependencies are given by real-world constraints and they show the interrelationships of the attributes.

This means we need to know and understand the meaning of the attributes and relationships between them.

We can rule out an FD based on the current instance but not vice versa (ref. “for any legal instance” in the definition)!

SUPERKEY AND CANDIDATE KEY

Let $R(A_1, A_2, \ldots, A_n)$ denote a relation schema.

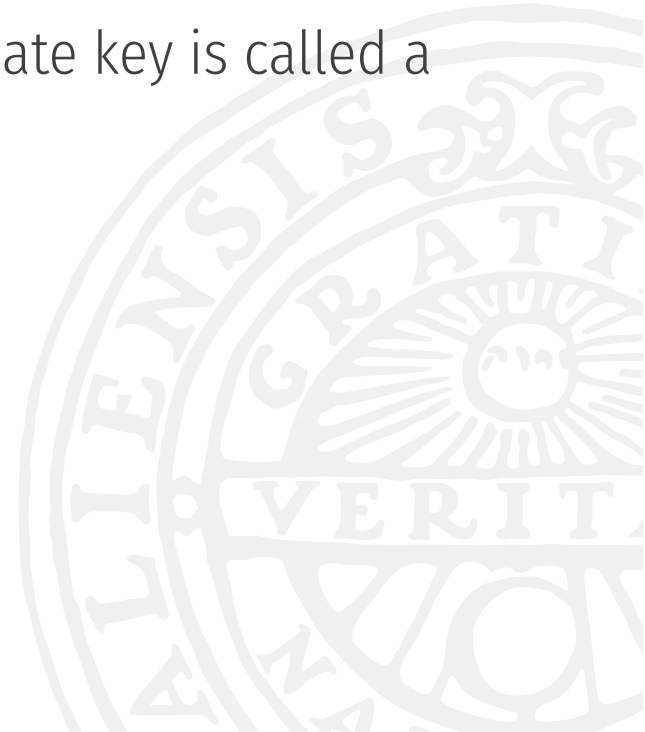
A set $X \subseteq \{A_1, A_2, \ldots, A_n\}$ is

- superkey of R iff $X \rightarrow \{A_1, A_2, \ldots, A_n\}$
- superkey of R iff $X^+ = \{A_1, A_2, \ldots, A_n\}$
- candidate key of R iff $X \rightarrow \{A_1, A_2, \ldots, A_n\}$ is a full functional dependency.

PRIME AND NON-PRIME ATTRIBUTES

An attribute which is a member of any candidate key is called a **prime** attribute.

An attribute which is not a member of any candidate key is called a **non-prime** attribute.



QUIZ 1

$R(A_1, A_2, \ldots, A_n)$ has a candidate key X.

Which of the following options is true?

1. $X \rightarrow \{A_1, A_2, \ldots A_n\} \backslash X$
2. $\{A_1, A_2, \ldots A_n\} \backslash X \rightarrow X$
3. Both previous answers
4. None of the previous answers

QUIZ 2 (PSEUDOTRANSITIVITY)

Consider a relation $R(A_1, \ldots, A_n)$, $X, Y, Z, W \subseteq \{A_1, \ldots, A_n\}$ and

- $X \rightarrow Y$
- $WY \rightarrow Z$

1. $X \rightarrow WZ$
2. $X \rightarrow WY$
3. $WX \rightarrow Z$
4. $Y \rightarrow Z$
5. $WY \rightarrow X$
6. None of the other answers

EXERCISE 3

Find all completely non-trivial full functional dependencies and identify candidate keys, prime and non-prime attributes. (Make your own assumptions if necessary.)

eid	lname	fname	did	dept	pid	project	hours
1	Alnes	Bernt	1	Planning	1	Alfa	100
2	Fjelldal	Mads	1	Planning	3	Charlie	140
4	Longva	Victor	1	Planning	2	Bravo	80
7	Bakke	Alfred	2	Production A	1	Alfa	80
8	Vie	Tor	2	Production A	1	Alfa	910
8	Vie	Tor	2	Production A	3	Charlie	720
9	Westgaard	Sten	2	Production A	3	Charlie	310
10	Liseth	Rakel	3	Production B	3	Charlie	480
11	Norman	Emil	3	Production B	3	Charlie	460
12	Dyrhaug	Atle	3	Production B	2	Bravo	810
15	Kvien	Amalie	4	Sales	2	Bravo	15
15	Kvien	Amalie	4	Sales	3	Charlie	75
16	Tveten	Thomas	4	Sales	2	Bravo	20
17	Lende	Marita	4	Sales	1	Alfa	40

Note: eid = employee ID, did = department ID, pid = project ID

REMINDER: FIRST NORMAL FORM (1NF)

A relation R is in 1NF iff

- the domain (i.e., all possible values) of each attribute contains only atomic (indivisible) values, and
- each attribute value is a single value from the domain of that attribute.

1NF disallows:

- composite attributes
- multivalued attributes
- nested relations

SECOND NORMAL FORM (2NF)

A relation R is in 2NF iff

- R is in 1NF, and
- no non-prime attribute is functionally dependent on any proper subset of any candidate key.

The second part is equivalent to “every non-prime attribute is fully functionally dependent on the whole of every candidate key”.

EXAMPLE OF A RELATION IN 1NF BUT NOT IN 2NF

EmpProj(eid, lname, fname, pid, project, hours)

eid	lname	fname	pid	project	hours
1	Alnes	Bernt	1	Alfa	100
2	Fjelldal	Mads	3	Charlie	140
4	Longva	Victor	2	Bravo	80
7	Bakke	Alfred	1	Alfa	80
8	Vie	Tor	1	Alfa	910
8	Vie	Tor	3	Charlie	720
9	Westgaard	Sten	3	Charlie	310
10	Liseth	Rakel	3	Charlie	480
11	Norman	Emil	3	Charlie	460
12	Dyrhaug	Atle	2	Bravo	810
15	Kvien	Amalie	2	Bravo	15
15	Kvien	Amalie	3	Charlie	75
16	Tveten	Thomas	2	Bravo	20
17	Lende	Marita	1	Alfa	40

Attributes lname and fname are dependent on eid which is a proper subset of a candidate key (eid, pid).

THIRD NORMAL FORM – 3NF

A relation R is in 3NF iff

- R is in 2NF, and
- each non-prime attribute is fully functionally dependent only on the candidate keys.

Ignoring the trivial dependencies.
In the literature the definition of 3NF often uses transitive functional dependencies: A relation is in 3NF iff it is in 2NF and no non-prime attribute is transitively dependent on a candidate key. Attribute A is transitively dependent on X if there exists Y such that $X \rightarrow Y$, $Y \not\rightarrow X$, $Y \rightarrow \{A\}$ and $A \notin Y$.

Lemma (Carlo Zaniolo):
A relation R is in 3NF iff for each FD of R, $X \rightarrow Y$:

- $X \rightarrow Y$ is trivial, or
- X is a superkey, or
- every element of $Y \backslash X$ is a prime attribute.

EXAMPLE OF A RELATION IN 2NF BUT NOT IN 3NF

EmpDept(eid, fname, lname, did, dept)

eid	fname	lname	did	dept
1	Bernt	Alnes	1	Planning
2	Mads	Fjelldal	1	Planning
3	Karoline	Lekve	1	Planning
4	Victor	Longva	1	Planning
5	Ingvar	Nymo	2	Production A
6	Runar	Bodin	2	Production A
7	Alfred	Bakke	2	Production A
8	Tor	Vie	2	Production A
9	Sten	Westgaard	2	Production A
10	Rakel	Liseth	3	Production B
11	Emil	Norman	3	Production B
12	Atle	Dyrhaug	3	Production B
13	Jens	Kvistad	3	Production B
14	Lucas	Ulset	3	Production B
15	Amalie	Kvien	4	Sales
16	Thomas	Tveten	4	Sales
17	Marita	Lende	4	Sales

did → dept, and both did and dept are non-prime attributes.

BOYCE-CODD NORMAL FORM (BCNF)

A relation R is in BCNF iff for each FD of R, $X \rightarrow Y$:

- $X \rightarrow Y$ is trivial, or
- X is a superkey.

Note: In some literature, Boyce-Codd Normal Form is also called 3.5NF.

EXAMPLE OF A RELATION IN 3NF BUT NOT IN BCNF

CourtBooking(court, start, end, rate)

court	start	end	rate
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

There are four distinct rate types:

- SAVER for Court 1 for members
- STANDARD for Court 1 for non-members
- PREMIUM-A for Court 2 for members
- PREMIUM-B for Court 2 for non-members

Source: Wikipedia

EXAMPLE OF A RELATION IN 3NF BUT NOT IN BCNF, CONT'D

Candidate keys:

- {court, start}
- {court, end}
- {rate, start}
- {rate, end}

All attributes are prime $\Rightarrow$ the relation is in 3NF.

It is not however in BCNF since the non-trivial dependency

$$\{\text{rate}\} \rightarrow \{\text{court}\}$$

and {rate} is not a superkey.

NORMAL FORMS

Each NF implies the previous ones:

- If a relation is in 2NF, it is also in 1NF.
- If a relation is in 3NF, it is also in 2NF (and 1NF).
- If a relation is in BCNF, it is also in 3NF (and 2NF and 1NF).

It is not necessary to normalize to the highest normal form. In practice it is usually enough to stop at 3NF or BCNF.

EXAMPLE: NORMALIZATION UNF → 1NF

eid	lname	fname	email
1	Alnes	Bernt	alnes@example.com
2	Fjelldal	Mads	fjelldal@example.com, mads@fjelldal.info
3	Lekve	Karoline	lekve@example.com
4	Longva	Victor	longva@example.com
5	Nymo	Ingvar	nymo@example.com, boss@nymo.com

eid	lname	fname	email
1	Alnes	Bernt	alnes@example.com
2	Fjelldal	Mads	fjelldal@example.com
2	Fjelldal	Mads	mads@fjelldal.info
3	Lekve	Karoline	lekve@example.com
4	Longva	Victor	longva@example.com
5	Nymo	Ingvar	nymo@example.com
5	Nymo	Ingvar	boss@nymo.com

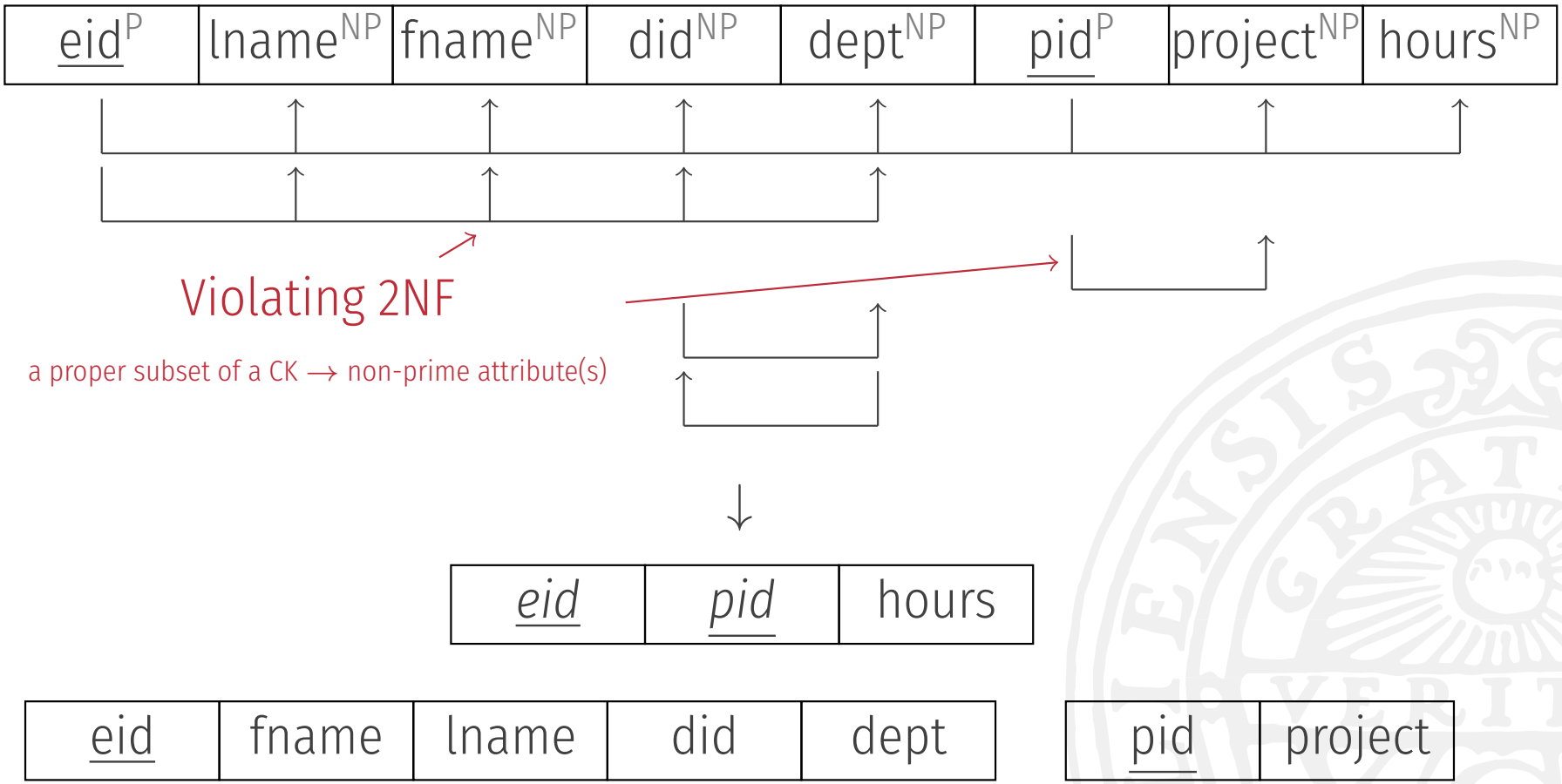
UNF = “unnormalized” form, sometimes denoted by 0NF
Note that change of the primary key.

NORMALIZATION TO 2NF AND 3NF

- Assumption: The relation is in the previous normal form.
- Find a non-trivial FFD that violates the given normal form. Let X denote its determinant.
  - Create a new relation with attributes X^+ and the primary key X.
  - Copy the distinct sets of values for these attributes to the new relation.
  - Remove the $X^+ \setminus X$ attributes from the original relation.
  - Add a FK for X in the original relation (referencing X in the new relation).
- Repeat until the relation conforms to the normal form.

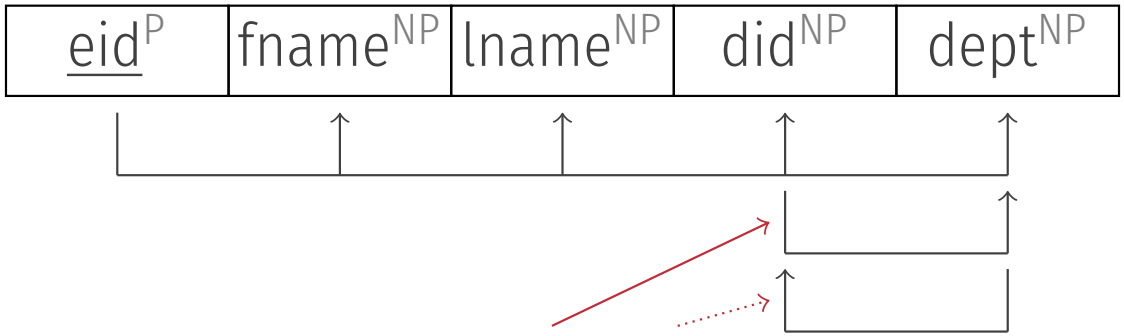
EXAMPLE: NORMALIZATION 1NF → 2NF

Example from Exercise 3 with the following assumptions: Different people might have the same name, the department names are unique and there might be several projects with the same name. There is only one candidate key: {eid, pid}.



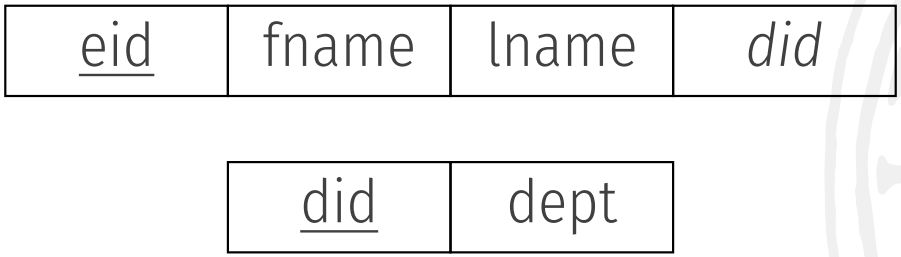
EXAMPLE: NORMALIZATION 2NF → 3NF

The relation with the data about employees is not in 3NF. There is only one CK: {eid}.



Violating 3NF

non-prime attribute → non-prime attribute



NORMALIZATION 3NF $\rightarrow$ BCNF

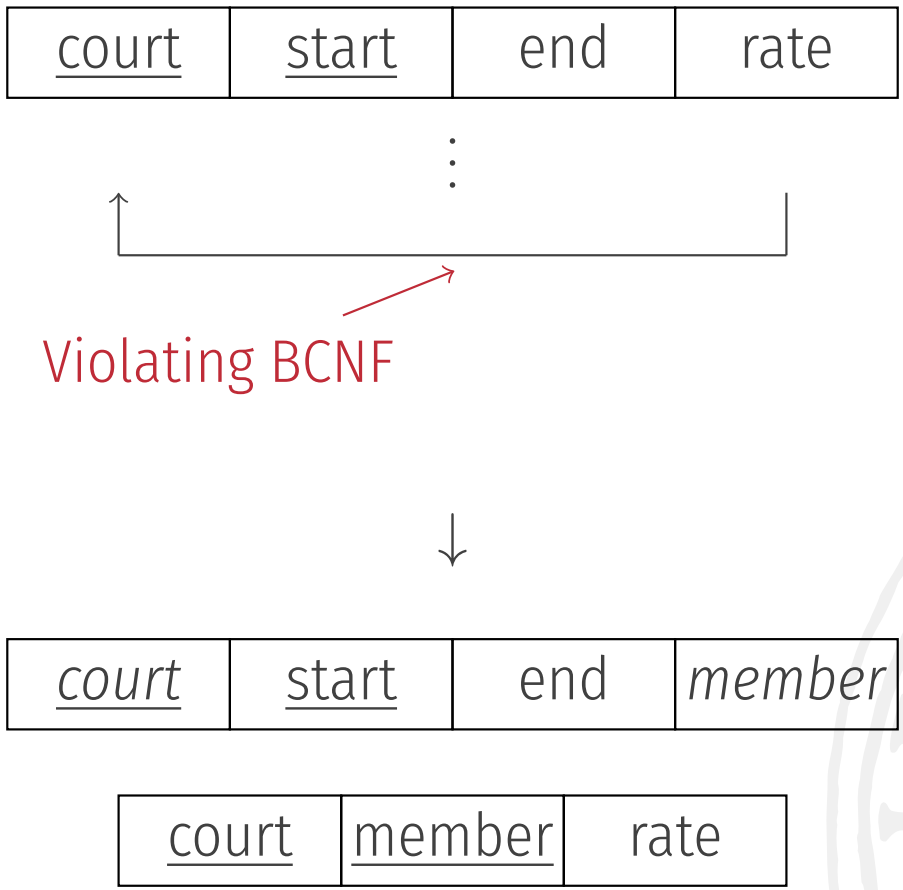
- Find non-trivial FFDs violating the BCNF, i.e. those whose determinant is not a candidate key.
- Decompose the relation (might be more difficult than for previous NFs).

Sometimes it is not possible to decompose the relation and keep the dependencies.

E.g. $R(A, B, C)$ with FFDs $AB \rightarrow C, C \rightarrow B$ cannot be represented by a BCNF schema.

EXAMPLE: NORMALIZATION 3NF → BCNF

CourtBooking:



USEFUL SQL STATEMENTS

Checking if $X \rightarrow \{A\}$ does not hold in the current instance of the table:

```
SELECT X1,X2,...,Xn count(DISTINCT A) AS dcount
FROM table
GROUP BY X1,X2,...,Xn
HAVING dcount > 1
```

Creating a new table for $X \rightarrow Y$ and populating it:

```
CREATE TABLE newtable AS
SELECT DISTINCT X1,X2,...,Xn,Y1,Y2,...,Ym
FROM table
```

or

```
CREATE TABLE newtable (...);

INSERT INTO newtable
SELECT DISTINCT X1,X2,...,Xn,Y1,Y2,...,Ym
FROM table
```

USEFUL SQL STATEMENTS, CONT'D

Adding a primary key to a table:

```
ALTER TABLE newtable
ADD PRIMARY KEY (X1, X2, ..., Xn);
```

Adding a foreign key to a table:

```
ALTER TABLE table
ADD FOREIGN KEY (X1, X2, ..., Xn) REFERENCES newtable(X1, X2, ..., Xn);
```

Dropping a column in a table:

```
ALTER TABLE table
DROP COLUMN A;
```

Note: some of RDBMS (e.g. SQLite) do not allow altering schemas.

QUIZ 3

Consider a relation in 1NF $R(A, B, C, D, E)$ with the following FFDs:

- $A, B \rightarrow C, D, E$
- $C \rightarrow A, B, D, E$
- $E \rightarrow D$

Which of the following is true?

1. R is in BCNF
2. R is in 3NF but not in BCNF
3. R is in 2NF but not in 3NF
4. R is in 3NF but not in 2NF
5. R is in BCNF but not in 3NF
6. R is in 1NF but not in 2NF
7. None of the other answers

QUIZ 4

Consider a relation in 1NF $R(A, B, C, D, E)$ with the following FFDs:

- $A, B \rightarrow C, D, E$
- $C \rightarrow A, B, D, E$
- $E \rightarrow D$

Which of the following normalized databases contains all the information contained in the original table, with all relations in BCNF?

1. $R_1(A, B, E)$, $R_2(C, E)$, $R_3(E, D)$
2. $R_1(A, B, C, D, E)$, $R_2(C, A, B, D, E)$, $R_3(E, D)$
3. $R(A, B, C, D, E)$
4. $R_1(A, B, C, E)$, $R_2(E, D)$
5. $R_1(A, B, D)$, $R_2(C, D)$, $R_3(E, D)$
6. None of the other answers

QUIZ 5

Consider the relation corresponding to the following SQL statement: CREATE TABLE R (A int NOT NULL PRIMARY KEY, B int NOT NULL, C int NOT NULL, D int NOT NULL, UNIQUE(B,C)) and assume that there is a FFD $C \rightarrow D$. Which of the following is true?

1. R is in 1NF but not in 2NF
2. R is in 3NF but not in 2NF
3. R is in 3NF but not in BCNF
4. R is in BCNF
5. R is in 2NF but not in 3NF
6. R is in BCNF but not in 3NF
7. None of the other answers