Assume the table on the right contains the **only** set of tuples that may appear in a table R. Which of the following FDs hold in R?

<table>
<thead>
<tr>
<th>tuple</th>
<th>X</th>
<th>Y</th>
<th>V</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x₁</td>
<td>y₁</td>
<td>v₁</td>
<td>w₁</td>
</tr>
<tr>
<td>2</td>
<td>x₁</td>
<td>y₁</td>
<td>v₂</td>
<td>w₂</td>
</tr>
<tr>
<td>3</td>
<td>x₂</td>
<td>y₁</td>
<td>v₁</td>
<td>w₃</td>
</tr>
<tr>
<td>4</td>
<td>x₂</td>
<td>y₁</td>
<td>v₃</td>
<td>w₄</td>
</tr>
</tbody>
</table>

- **{X} → {X}**
  Yes – trivial (holds in any table)

- **{X} → {Y}**
  Yes – all values of Y are identical

- **{X} → {V}**
  No – see first two rows

- **{X} → {W}**
  No – same as before

- **{Y} → {X}**
  No (same for {Y} → {V,W})

- **{W} → {X}**
  Yes – all W values are different

- **{X, V} → {Y}**
  Yes – X alone determines Y

- **{Y, V} → {X}**
  No – see rows 1 and 3
In the previous example, we assumed that we know all possible records in a table, which is not usually true.

In general by looking at an instance of a relation, we can only tell FDs that are **NOT** satisfied.

List 5 FDs that are **not** satisfied in the table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>c1</td>
</tr>
<tr>
<td>a1</td>
<td>b1</td>
<td>c2</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>c1</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>c3</td>
</tr>
</tbody>
</table>

**Solution**

\{B\} → \{A\}  
\{B\} → \{C\}  
\{C\} → \{A\}  
\{A\} → \{C\}  
\{A,B\} → \{C\}
In reality, FDs are given implicitly in the form of constraints when designing a database.

Let a relation $R(\text{title, theater, city})$ where $\text{title}$ is the name of a movie, $\text{theater}$ is the name of a theater playing the movie and $\text{city}$ is the city where the theater is located.

We are given the following constraints:
- Two different cities cannot have theaters with the same name.
- Two different theaters in the same city cannot play the same movie.
- A theater can play many movies (e.g., cineplex).

Write the set of functional dependencies implied by the above assumptions

Solution

$\{\text{Theater}\} \rightarrow \{\text{City}\}$ (if we know the theater, we know the city where it is located – the opposite is not true as a city can have many theaters)

$\{\text{City, Title}\} \rightarrow \{\text{Theater}\}$

Can you identify the candidate key(s)?
City, Title and Theater, Title
We want to create the database for a bank that contains accounts, branches, and customers. We are given the following constraints:

- An account cannot be shared by multiple customers.
- Two different branches do not have the same account.
- Each customer can have at most one account in a branch (but different accounts in different branches).

Write the functional dependencies implied by the above constraints

\[
\begin{align*}
\text{Solution} & \\
\{\text{Account}\} & \rightarrow \{\text{Customer}\} \\
\{\text{Account}\} & \rightarrow \{\text{Branch}\} \\
\{\text{Customer, Branch}\} & \rightarrow \{\text{Account}\}
\end{align*}
\]

Write the candidate key(s):

Customer, Branch and Account
Let R(A,B,C). Assume that we do not know the keys of the table.

How would you test if A is a candidate key of R with a SQL query?

**Solution**

```sql
select A
from R
group by A
having count(*)>1
```

- If this query gives a non-empty result, then A is not a key.
- If the result is empty, we cannot be sure!

What about testing if the dependency \{A\}→\{B\} holds in R?

**Solution**

- Same as before, but replace the last line with `having count(distinct B)>1`
Let the rule: if $X \rightarrow Z$ and $Y \rightarrow Z$, then $X \rightarrow Y$. Show that this rule is not sound (correct) with a counter-example.

Solution

• Let’s use $R(X,Y,Z)$
• We want to find an instance of $R$ where the rule is violated
• Lets say that $R$ contains just two tuples:
  - $(x_1, y_1, z_1)$
  - $(x_1, y_2, z_1)$
• $\{X\} \rightarrow \{Z\}$ and $\{Y\} \rightarrow \{Z\}$ hold, but $\{X\} \rightarrow \{Y\}$ is not true
• Consider a relation R(X,Y,U,V,W) with the following set of dependencies
  - \{\{X\} \rightarrow \{Y\}, \{U,V\} \rightarrow \{W\}, \{V\} \rightarrow \{X\}\}
• Find the closure of each attribute

**Solution**

\[X^+=\{X,Y\}\]
\[Y^+=\{Y\}\]
\[U^+=\{U\}\]
\[V^+=\{V,X,Y\}\]
\[W^+=\{W\}\]

**What is the primary key of R?**

\[UV\]
Exercise #8

- \( R = (A, B, C, G, H, I) \)
  \( F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\} \)

Is \( AG \) a (super)key of \( R \) given \( F \)?

Compute \( AG^+ \) and check if \( AG^+ \) contains the entire \( R \):

1. Result = \( AG \)
2. Result = \( ABCG \) (\( A \rightarrow C; A \rightarrow B \) and \( A \subseteq AG \))
3. Result = \( ABCGH \) (\( CG \rightarrow H \) and \( CG \subseteq AGBC \))
4. Result = \( ABCGHI \) (\( CG \rightarrow I \) and \( CG \subseteq AGBCH \))

\( AG^+ \) is a superkey of \( R \) given \( F \)

Is \( AG \) a candidate key?

1. \( AG \rightarrow R \)
2. Does \( A^+ \rightarrow R \)?
3. Does \( G^+ \rightarrow R \)?

If both are NO, then \( AG \) is a candidate key
Let the relation schema R(A,B,C,D,E) and the set of functional dependencies: \( F = \{ \{A\} \rightarrow \{B\}, \{A,B\} \rightarrow \{C\}, \{D\} \rightarrow \{A,C\}\} \):

Find the canonical cover of \( F \)

\[
\{A \rightarrow B, \ AB \rightarrow C, \ D \rightarrow AC\} \\
\{A \rightarrow B, \ A \rightarrow C, \ D \rightarrow AC\} \\
\{A \rightarrow BC, \ D \rightarrow A\}
\]

Compute the attribute closures

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

What is the primary key of R?

DE