Find the functional dependencies of StaffBranch from the data.
Find the functional dependencies of *StaffBranch* from the data.

\[
\begin{align*}
\{\text{staffNo}\} &\rightarrow \{\text{sName, position, salary, branchNo, bAddress}\} \\
\{\text{sName}\} &\rightarrow \{\text{staffNo, position, salary, branchNo, bAddress}\} \\
\{\text{branchNo}\} &\rightarrow \{\text{bAddress}\} \\
\{\text{bAddress}\} &\rightarrow \{\text{branchNo}\} \\
\{\text{branchNo, position}\} &\rightarrow \{\text{salary}\} \\
\{\text{bAddress, position}\} &\rightarrow \{\text{salary}\}
\end{align*}
\]
Find the functional dependencies of StaffBranch that make sense.

\{staffNo\} \rightarrow \{sName, position, salary, branchNo, bAddress\}
\{sName\} \rightarrow \{staffNo, position, salary, branchNo, bAddress\}
\{branchNo\} \rightarrow \{bAddress\}
\{bAddress\} \rightarrow \{branchNo\}
\{branchNo, position\} \rightarrow \{salary\}
\{bAddress, position\} \rightarrow \{salary\}
A functional dependency (FD) on a relation $R$ is a statement of the form:

If two or more tuples of $R$ agree on the attribute values $A_1, A_2, ..., A_n$ (i.e., the tuples have the same values for each of these attributes), then they must also agree on another attribute $B$. We write $\{A_1, A_2, ..., A_n\} \rightarrow \{B\}$

Consider the schema Movies (title, year, length, filmType, studioName, starName)

Possible FDs:
$\{\text{title, year}\} \rightarrow \{\text{length}\}$, $\{\text{title, year}\} \rightarrow \{\text{filmType}\}$, $\{\text{title, year}\} \rightarrow \{\text{studioName}\}$

<table>
<thead>
<tr>
<th>title</th>
<th>year</th>
<th>length</th>
<th>filmType</th>
<th>studioName</th>
<th>starName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>color</td>
<td>Fox</td>
<td>Carrie Fisher</td>
</tr>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>color</td>
<td>Fox</td>
<td>Mark Hamill</td>
</tr>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>color</td>
<td>Fox</td>
<td>Harrison Ford</td>
</tr>
<tr>
<td>Might Ducks</td>
<td>1991</td>
<td>104</td>
<td>color</td>
<td>Disney</td>
<td>Emilio Estevez</td>
</tr>
<tr>
<td>Wayne’s Word</td>
<td>1992</td>
<td>95</td>
<td>color</td>
<td>Paramount</td>
<td>Dana Carvey</td>
</tr>
<tr>
<td>Wayne’s Word</td>
<td>1992</td>
<td>95</td>
<td>color</td>
<td>Paramount</td>
<td>Mike Meyers</td>
</tr>
</tbody>
</table>
We see that we can summarize the right sides:
{title, year} → {length, filmType, studioName}

What about the validity of {title, year} → {starName}? Is it a FD?

A set of one or more attributes \{A_1, A_2, \ldots, A_n\} is a **key** for a relation schema \(R\) if

1. \{A_1, A_2, \ldots, A_n\} → \(R \setminus \{A_1, A_2, \ldots, A_n\}\)

These attributes functionally determine all other attributes of \(R\). We can also say:
\{A_1, A_2, \ldots, A_n\} → \(R\)

2. There is no proper subset \(C \subset \{A_1, A_2, \ldots, A_n\}\) such that \(C \rightarrow R\).

Determine an attribute set that forms a key for Movies.
Checking the preservation of a functional dependency

- **Alternative characterization of a FD** \( A \rightarrow B \)

  Let \( A = \{A_1, ..., A_n\} \), and let \( \text{dom}(A) = \text{dom}(A_1) \times ... \times \text{dom}(A_n) \).

  The FD \( A \rightarrow B \) **holds on** \( R \) if \( \forall \ v \in \text{dom}(A) : |\pi_B(\sigma_{A=v}(R))| \leq 1 \)

  \( (A = v \text{ stands for } A_1 = v_1 \land ... \land A_n = v_n) \)

- This leads to a simple algorithm which computes whether a given relation \( R \) satisfies a given FD \( A \rightarrow B \):

  **algorithm** \( \text{FDPreservation}(R, A \rightarrow B) \)

  // input: relation \( R \) and FD \( A \rightarrow B \)

  // output: \( true \), if \( A \rightarrow B \) holds on \( R \); \( false \) otherwise

  sort \( R \) with respect to \( A \)-values

  if all groups consisting of tuples with equal \( A \)-values also have equal \( B \)-values then
  
  return \( true \)

  else
  
  return \( false \)

  end.
Computation of FDs

- Goal: Compute for a given set $F$ of FDs all logically implied FDs.

- Let $F^+$ be the set of all FDs that can be logically implied from the FDs in $F$. $F^+$ is called the closure of $F$.

- Let $R$ be a relation schema, $F$ a set of FDs and $A, B, C \subseteq R$.

  The following inference rules are used to compute $F^+$ (Armstrong’s axioms):
  - **reflexivity rule**: Let $B \subseteq A$. Then always $A \rightarrow B$ (special case: $A \rightarrow A$) holds.
  - **augmentation rule**: If $A \rightarrow B$ holds, then also $A \cup C \rightarrow B \cup C$ holds.
  - **transitivity rule**: If $A \rightarrow B$ and $B \rightarrow C$ holds, then also $A \rightarrow C$ holds.

- It can be formally shown that these rules are **sound** and **complete**.
  - **soundness**: Inferred FDs hold for all relations of this schema.
  - **completeness**: All valid FDs in $F^+$ can be logically implied with these rules.
Although Armstrong’s axioms are complete, it is comfortable to add three further inference rules:

- **union rule**: If \( A \rightarrow B \) and \( A \rightarrow C \) holds, then also \( A \rightarrow B \cup C \) holds.
- **decomposition rule**: If \( A \rightarrow B \cup C \) holds, then also \( A \rightarrow B \) and \( A \rightarrow C \) holds.
- **pseudotransitivity rule**: If \( A \rightarrow B \) and \( B \cup C \rightarrow D \) holds, then also \( A \cup C \rightarrow D \) holds.

**example:**

- *supplier* relation with the schema *supplier*(sname, saddr, product, price)
- Valid FDs: \( \{\text{name}\} \rightarrow \{\text{saddr}\} \), \( \{\text{name}, \text{product}\} \rightarrow \{\text{price}\} \), \( \{\text{name}\} \rightarrow \{\text{name}\} \), \( \{\text{name}, \text{product}\} \rightarrow \{\text{product}\} \)
- It is to be shown: \( \{\text{name}, \text{product}\} \rightarrow \{\text{saddr}\} \) is also satisfied.

  We have: \( \{\text{name}\} \rightarrow \{\text{saddr}\} \).

  Due to the augmentation rule we obtain: \( \{\text{name}, \text{product}\} \rightarrow \{\text{saddr}, \text{product}\} \).

  Due to the decomposition rule we hence obtain: \( \{\text{name}, \text{product}\} \rightarrow \{\text{saddr}\} \).
computing the closure $F^+$

$F^+ = F$

repeat

for each functional dependency $f$ in $F^+$ do

apply reflexivity and augmentation rules to $F^+$
add the resulting functional dependencies to $F^+$

od;

for each pair of functional dependencies $f_1$ and $f_2$ in $F^+$ do

if $f_1$ and $f_2$ can be combined using transitivity then

add the resulting functional dependency to $F^+$

fi

od;

until $F^+$ does not change any further
Containment of a FD in a closure $F^+$

- question: Let $F$ be a set of FDs and $A \rightarrow B$ a FD. Does $A \rightarrow B \in F^+$ hold?
- problem: explicit calculation of $F^+$ is too expensive
- instead: calculation of the closure $A^+$ of the attribute set $A$ regarding the set $F$
  - $A^+$ consists of all attributes that are functionally determined by $A$.
  - If $B \subseteq A^+$ holds, then also $A \rightarrow B \in F^+$ holds.
- algorithm for inferring $A^+$

```plaintext
algorithm AttrClosure(F, A)
// input: a set F of FDs and a set A of attributes
// output: the complete set $A^+$ of attributes for which holds: $A \rightarrow A^+$

$A^+ := A$;
repeat
    Old$A^+ = A^+$;
    foreach FD $B \rightarrow C \in F$ do
        if $B \subseteq A^+$ then $A^+ := A^+ \cup C$;
    until $A^+ = OldA^+$; return $A^+$
```