Relation $R = \text{SNCPXYQ}$ was decomposed into \{SNC\}, \{PXY\}, and \{SPQ\}. Considering the following functional dependencies $S \rightarrow NC$, $P \rightarrow XY$, and $SP \rightarrow Q$, do the Chase Test to assess whether this decomposition had a lossless join.
The Chase Test

• Suppose tuple $t$ comes back in the join.
• Then $t$ is the join of projections of some tuples of $R$, one for each $R_i$ of the decomposition.
• Can we use the given FD’s to show that one of these tuples must be $t$?
The Chase – (2)

• Start by assuming $t = abc$… .
• For each $i$, there is a tuple $s_i$ of $R$ that has $a, b, c, ...$ in the attributes of $R_i$.
• $s_i$ can have any values in other attributes.
• We’ll use the same letter as in $t$, but with a subscript, for these components.
Example: The Chase

• Let $R = AB\overline{C}D$, and the decomposition be $AB$, $BC$, and $CD$.
• Let the given FD’s be $C\rightarrow D$ and $B \rightarrow A$.
• Suppose the tuple $t = abcd$ is the join of tuples projected onto $AB$, $BC$, $CD$. 
The tuples of $R$ projected onto $AB$, $BC$, $CD$.

We've proved the second tuple must be $t$.

Use $B \rightarrow A$

Use $C \rightarrow D$
Answer to Exercise 1

\{SNC\}, \{PXY\}, \{SPQ\}

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>C</th>
<th>P</th>
<th>X</th>
<th>Y</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>n</td>
<td>c</td>
<td>$p_1$</td>
<td>$x_1$</td>
<td>$y_1$</td>
<td>$q_1$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$n_2$</td>
<td>$c_2$</td>
<td>$p$</td>
<td>$x$</td>
<td>$y$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>s</td>
<td>$n_3$</td>
<td>$c_3$</td>
<td>$p$</td>
<td>$x_3$</td>
<td>$y_3$</td>
<td>q</td>
</tr>
</tbody>
</table>
Answer to Exercise 1

Using $S \rightarrow NC$

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>N</th>
<th>C</th>
<th>P</th>
<th>X</th>
<th>Y</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$s$</td>
<td>$n$</td>
<td>$c$</td>
<td>$p_1$</td>
<td>$x_1$</td>
<td>$y_1$</td>
<td>$q_1$</td>
</tr>
<tr>
<td></td>
<td>$s_2$</td>
<td>$n_2$</td>
<td>$c_2$</td>
<td>$p$</td>
<td>$x$</td>
<td>$y$</td>
<td>$q_2$</td>
</tr>
<tr>
<td></td>
<td>$s$</td>
<td>$n_3$</td>
<td>$c_3$</td>
<td>$p$</td>
<td>$x_3$</td>
<td>$y_3$</td>
<td>$q$</td>
</tr>
</tbody>
</table>
Answer to Exercise 1

Using $P \rightarrow XY$

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>N</th>
<th>C</th>
<th>P</th>
<th>X</th>
<th>Y</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$s$</td>
<td>$n$</td>
<td>$c$</td>
<td>$p_1$</td>
<td>$x_1$</td>
<td>$y_1$</td>
<td>$q_1$</td>
</tr>
<tr>
<td></td>
<td>$s_2$</td>
<td>$n_2$</td>
<td>$c_2$</td>
<td>$p$</td>
<td>$x$</td>
<td>$y$</td>
<td>$q_2$</td>
</tr>
<tr>
<td></td>
<td>$s$</td>
<td>$n_\beta$</td>
<td>$c_\beta$</td>
<td>$p$</td>
<td>$x_\beta$</td>
<td>$y_\beta$</td>
<td>$q$</td>
</tr>
</tbody>
</table>

Yes, the decomposition had a lossless join.
Exercise 2

Given a relation $R(A, B, C, D, E)$ and a set of functional dependencies $F = A \rightarrow BC$, $CD \rightarrow E$, $B \rightarrow D$, and $E \rightarrow A$, compute $F^+$. List the candidate keys of $R$. 
Computing Closure of Attributes

- We want to compute \( \{A_1, \ldots, A_n\}^+ \).

  Initialize \( X = \{A_1, \ldots, A_n\} \)

  Repeatedly search for some FD \( B_1, B_2, \ldots, B_n \rightarrow C \)
  such that \( B_i \in X \ \forall i \), and \( C \notin X \)
  \( X = X \cup \{C\} \)
  Until no more attributes can be added to \( X \)

  Return \( X \)
Computing Closure of Attributes: Example

R(A,B,C,D,E,F)
FDs S = \{AB \rightarrow C, BC \rightarrow AD, D \rightarrow E, CF \rightarrow B\}

What is the closure of:
{A,B} + = \{A, B\}  \textit{trivial dependencies}
= \{A,B,C\}  \ AB \rightarrow C
= \{A,B,C,D\}  \ BC \rightarrow D
= \{A,B,C,D,E\}  \ D \rightarrow E

Now, we know that \( AB \rightarrow CDE \)
S ==> AB \rightarrow CDE, or AB \rightarrow CDE follows from S
Closure of Attributes and of FDs

• If we know how to compute the closure of any set of attributes, we can test if any given FD $A_1,\ldots,A_n \rightarrow B$ follows from a set of FDs $S$
  – Compute $\{A_1,\ldots,A_n\}^+$
  – If $B \in \{A_1,\ldots,A_n\}^+$, then $A_1,\ldots,A_n \rightarrow B$
## Closure for A

<table>
<thead>
<tr>
<th>Iteration</th>
<th>result</th>
<th>using</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ABC</td>
<td>A--&gt;BC</td>
</tr>
<tr>
<td>3</td>
<td>ABCD</td>
<td>B--&gt;D</td>
</tr>
<tr>
<td>4</td>
<td>ABCDE</td>
<td>C=--&gt;E</td>
</tr>
<tr>
<td>5</td>
<td>ABCDE</td>
<td></td>
</tr>
</tbody>
</table>

\[ A^+ = ABCDE, \text{ Hence A is a candidate key} \]
## Answer to Exercise 2

**Closure for CD**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>result</th>
<th>using</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CD</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CDE</td>
<td>CD---&gt;E</td>
</tr>
<tr>
<td>3</td>
<td>ACDE</td>
<td>E---&gt;A</td>
</tr>
<tr>
<td>4</td>
<td>ABCDE</td>
<td>A--&gt;BC</td>
</tr>
<tr>
<td>5</td>
<td>ABCDE</td>
<td></td>
</tr>
</tbody>
</table>

CD+ = ABCDE, Hence CD is a candidate key
# Answer to Exercise 2

## Closure for B

<table>
<thead>
<tr>
<th>Iteration</th>
<th>result</th>
<th>Using</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BD</td>
<td>B--&gt;D</td>
</tr>
<tr>
<td>3</td>
<td>BD</td>
<td></td>
</tr>
</tbody>
</table>

B+ = BD, Hence B is NOT a candidate key
Answer to Exercise 2

**Closure for E**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>result</th>
<th>using</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AE</td>
<td>E--&gt;A</td>
</tr>
<tr>
<td>3</td>
<td>ABCE</td>
<td>A--&gt;BC</td>
</tr>
<tr>
<td>4</td>
<td>ABCDE</td>
<td>B--&gt;D</td>
</tr>
<tr>
<td>5</td>
<td>ABCDE</td>
<td></td>
</tr>
</tbody>
</table>

\[ E^+ = ABCDE, \text{ Hence E is a candidate key} \]
Exercise 3

Consider the following set $F$ of functional dependencies on schema $R(A,B,C)$

$A \rightarrow BC$, $B \rightarrow C$, $A \rightarrow B$, $AB \rightarrow C$

Compute the minimal cover for $F$. 
Computing Minimal Covers

• Given a set of FDs F:
  1. Put FDs in standard form
     – Obtain G of equivalent FDs with single attribute on the right side
  2. Minimize left side of each FD
     – For each FD in G, check each attribute on the left side to see if it can be deleted while preserving equivalence to F+
  3. Delete redundant FDs
     – Check each remaining FD in G if it can be deleted while preserving equivalence to F+
Answer to Exercise 3

\[ F = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\} \]

1. Put FDs in standard form
   – \[ M = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow C\} \]

2. Minimize left side of each FD
   – \[ M = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow C\} \]
   – \[ M = \{A \rightarrow B, A \rightarrow C, B \rightarrow C\} \]

3. Delete redundant FDs
   – \[ M = \{A \rightarrow B, A \rightarrow C, B \rightarrow C\} \]
   – \[ M = \{A \rightarrow B, B \rightarrow C\} \]
Exercise 4

Answer the following questions about the table below:

a. List the functional dependencies that might hold.
b. List the candidate keys.
c. List update, insertion, and deletion anomalies associated with this schema.

<table>
<thead>
<tr>
<th>model</th>
<th>make</th>
<th>type</th>
<th>color</th>
<th>country</th>
<th>year</th>
<th>expensive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accord</td>
<td>Honda</td>
<td>sedan</td>
<td>black</td>
<td>Japan</td>
<td>1999</td>
<td>no</td>
</tr>
<tr>
<td>Accord</td>
<td>Honda</td>
<td>coupe</td>
<td>blue</td>
<td>Japan</td>
<td>2000</td>
<td>no</td>
</tr>
<tr>
<td>Accord</td>
<td>Honda</td>
<td>sedan</td>
<td>green</td>
<td>Japan</td>
<td>2001</td>
<td>no</td>
</tr>
<tr>
<td>s2000</td>
<td>Honda</td>
<td>sports</td>
<td>grey</td>
<td>Japan</td>
<td>2000</td>
<td>yes</td>
</tr>
<tr>
<td>Civic</td>
<td>Honda</td>
<td>coupe</td>
<td>white</td>
<td>Japan</td>
<td>2003</td>
<td>no</td>
</tr>
<tr>
<td>Civic</td>
<td>Honda</td>
<td>compact</td>
<td>red</td>
<td>Japan</td>
<td>2000</td>
<td>no</td>
</tr>
<tr>
<td>m3</td>
<td>BMW</td>
<td>sports</td>
<td>blue</td>
<td>Germany</td>
<td>2002</td>
<td>yes</td>
</tr>
<tr>
<td>m3</td>
<td>BMW</td>
<td>sports</td>
<td>black</td>
<td>Germany</td>
<td>2003</td>
<td>yes</td>
</tr>
<tr>
<td>330ci</td>
<td>BMW</td>
<td>compact</td>
<td>black</td>
<td>Germany</td>
<td>2003</td>
<td>no</td>
</tr>
<tr>
<td>E500</td>
<td>Mercedes</td>
<td>luxury</td>
<td>grey</td>
<td>Germany</td>
<td>2003</td>
<td>yes</td>
</tr>
<tr>
<td>ML500</td>
<td>Mercedes</td>
<td>SUV</td>
<td>black</td>
<td>Germany</td>
<td>2004</td>
<td>yes</td>
</tr>
</tbody>
</table>
a. List the functional dependencies that might hold.
   - model → make
   - make → country
   - model → country
   - type → expensive
b. List the candidate keys

(model, type, color, year)+ =
(model, make, type, color, country year, expensive)
c. List update, insertion, and deletion anomalies associated with this schema.

- **Update:** To update model, make and country must be updated as well to keep model → make and make → country.

- **Insert:** To insert model, we must keep the values for make and country the same from previous insertions of model. Otherwise model → make and make → country are violated.

- **Delete:** If we delete all of a particular model, then we lose all the information of make and country.
Given a relation schema $R(A,B,C,D)$ with FDs $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$.

a. Indicate all 3NF violations.

b. Decompose the relation into a collection of relations that are in 3NF.

c. Indicate all the BCNF violations.

d. Decompose relation into a collection of relations that are in BCNF.
Answer to Exercise 5

a. Indicate all 3NF violations.

3NF - all non-key attributes must depend on only the key

\[ AB \rightarrow C, \ C \rightarrow D, \ D \rightarrow A \]

Candidate Keys AB, BC, BD

Assuming the key is AB

b. Decompose the relation into a collection of relations that are in 3NF.

R1(A,B,C)  \quad R2(C,D)
Decomposing into BCNF: Example

- $R = CSJDPQV$
- $F = \{SD \rightarrow P; J \rightarrow S; JP \rightarrow C\}$ Key = $\{C\}$

Need to check for BCNF at each step!
Note: any 2-attribute relation is in BCNF (see pg 89 in textbook)
Dependency-Preserving Decomposition into 3NF: Example

• R = CSJDPQV
• F = \{SD \rightarrow P; J \rightarrow S; JP \rightarrow C\}, Key = \{C\}

JP \rightarrow C \text{ is not preserved}
Since F is a minimal cover, add CJP to the schema
Decomposing into BCNF: Another Example

- Movies = \{title, year, length, studio, starName\}
- F = \{title, year \rightarrow length, studio\},
  Key = \{title, year, starName\}

Movies = \{title, year, length, studio\}
StarsIn = \{title, year, starName\}
Answer to Exercise 5

c. Indicate all BCNF violations.

BCNF - all FDs are implied by the candidate keys

\[ \text{AB} \rightarrow \text{C, C} \rightarrow \text{D, D} \rightarrow \text{A, C} \rightarrow \text{A} \]

Assuming the key is AB

d. Decompose the relation into a collection of relations that are in BCNF.
Answer to Exercise 5

- BCNF violations \{C \rightarrow D; D \rightarrow A; C \rightarrow A\}

Which dependencies were not preserved?
- D \rightarrow A and AB \rightarrow C
Answer to Exercise 5

- BCNF violations \{C \rightarrow D; \ D \rightarrow A; \ C \rightarrow A\}

Which dependencies were not preserved?
AB \rightarrow C