1. Let $R$ be a relation with attributes $ABCD$. Consider the SQL conjunctive query:

$$\text{select } z.D \text{ from } R \times, R \times, R z \text{ where } y.C = 0 \text{ and } x.B = y.B \text{ and } z.B = 5 \text{ and } z.C = y.C$$

(i) (2 points) Construct the pattern corresponding to the query.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>b</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>b</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>5</td>
<td>0</td>
<td>d</td>
</tr>
</tbody>
</table>

(ii) (4 points) Minimize the pattern in (i) knowing that the query is only applied to databases satisfying the fd’s $A \rightarrow D$, $CD \rightarrow B$, $C \rightarrow A$.

Chasing first with $C \rightarrow A$ produces:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>b</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>b</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>5</td>
<td>0</td>
<td>d</td>
</tr>
</tbody>
</table>

We then chase with $A \rightarrow D$, yielding:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>b</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>b</td>
<td>0</td>
<td>d</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>5</td>
<td>0</td>
<td>d</td>
</tr>
</tbody>
</table>

Finally, chasing with $CD \rightarrow B$ produces:
\[
\begin{array}{cccc}
R & A & B & C & D \\
\hline
- & 5 & - & - \\
a & 5 & 0 & d \\
a & 5 & 0 & d \\
\end{array}
\]

Minimizing the above results in:

\[
\begin{array}{cccc}
R & A & B & C & D \\
\hline
a & 5 & 0 & d \\
\end{array}
\quad \text{answer} \quad \begin{array}{c}
D \\
d \\
\end{array}
\]

(iii) (2 points) Construct from the minimized pattern a corresponding minimized SQL query.

\begin{verbatim}
select D from R
where B = 5 and C = 0
\end{verbatim}

2. Let \( R \) be a relation with attributes \( ABCDEG \) and
\[
F = \{ E \rightarrow D, \ C \rightarrow B, \ CBE \rightarrow AG, \ B \rightarrow A, \ G \rightarrow E \}.
\]

(i) (1 point) Find all the keys of \( R \).
Note that \( C \) does not occur on the right-hand side of any FD, so it must belong to every key. Since \( C^+ = CBA \), \( C \) is not a key, so at least one attribute must be added (other than \( A, B \) which are already determined by \( C \)). Continuing the search, \( CD^+ = CDBA \), \( CE^+ = CEABDEG \), and \( CG^+ = CGBABD \) so \( CE \) and \( CG \) are keys. Extending \( CD \) to a superkey requires adding at least one of \( E \) and \( G \), so this would subsume the keys \( CE \) and \( CG \) so no other key can be obtained from \( CD \). Thus, the only keys are \( CE \) and \( CG \).

(ii) (4 points) Find a BCNF decomposition of \( R \) with lossless join with respect to \( F \). (Show how the decomposition is obtained.)
The BCNF decomposition algorithm produces \( \{ ED, CB, GE, CA, CG \} \) (see separate pdf file for details).

(iii) (2 points) Is the decomposition obtained in (ii) dependency preserving with respect to \( F \)?
The FDs \( E \rightarrow D, \ C \rightarrow B, \ G \rightarrow E \) are local so are preserved. The FD \( CBE \rightarrow AG \) is not preserved. To see this, start with
\textit{CBE}. We now have the following starred attributes in the decomposition:

\[ E^*D, C^*B^*, GE^*, C^*A, C^*G \]

Since \( E^+ = ED \) and \( D \) is in \( ED \) we can add \( D \). Since \( C^+ = CBA \) and \( A \) is in \( CA \), we can add \( A \). We now have \( CBEDA \) and the starred attributes

\[ E^*D^*, C^*B^*, GE^*, C^*A^*, C^*G \]

We are now done, because \( E^+ = ED \) and \( C^+ = CBA \) so nothing can be added. Since \( CBEDA \) does not contain \( G \), \( CBE \to AG \) is not preserved. So the decomposition is not dependency preserving.

(iv) (5 points) Find a 3NF decomposition of \( R \) with lossless join and dependency preserving with respect to \( F \) (show the steps). Is the decomposition also in BCNF?

We first eliminate redundancies in \( F \).

- single attributes on right-hand sides of FDs:

\[ E \to D, \ C \to B, \ CBE \to A, \ CBE \to G, \ B \to A, \ G \to E \]

- eliminate redundant FDs. The FDs \( E \to D, \ C \to B, \ CBE \to G, \ G \to E \) are not redundant (because each is the only FD producing the attribute on the right-hand side). The only redundant FD is \( CBE \to A \) (it is implied by \( B \to A \)). We are left with

\[ E \to D, \ C \to B, \ CBE \to G, \ B \to A, \ G \to E \]

- eliminate redundant attributes on left-hand sides of FDs. The only FD with multiple attributes on its left-hand side is \( CBE \to G \). Attribute \( B \) is redundant because \( CE^+ = CEDBGA \) which includes \( G \). \( C \) is not redundant because \( E^+ = E \) and \( E \) is not redundant because \( C^+ = CBA \) (so none determines \( G \)).

The result of eliminating redundancies is

\[ E \to D, \ C \to B, \ CE \to G, \ B \to A, \ G \to E \]
The first-cut 3NF dependency-preserving decomposition is

\[ \{ED, CB, CEG, BA, GE\} \]

Note that CEG is a superkey, so this decomposition also has lossless join, and there is no need to add a key. Also note that GE is included in CEG so it can be removed from the decomposition.

The decomposition is not in BCNF. Indeed, consider CEG (the only component which could violate BCNF). Since \( G^+ = GED \), \( G \rightarrow E \) is a violation of BCNF.