Database Design

The normal forms...

Defn A file is said to be in First Normal Form if no record has a multi-valued attribute.

Anti-Example..one way to organize the college athletics database could have been a file whose records had attributes:

College_Name, College_Teams.

The file would not have been in 1NF.

Another Anti-Example...in an Employees record, a field "Dependents"

Thrm, First normal form...Given a relation R(ABCD...M) where M is a multivalued attribute and A is a candidate key. Form two relations, R(ABCD...) And RM(AM) where RM has a separate row for each of the distinct entries in M.

E.g. The file described above would have decomposed into a table containing only information about the college and a second containing only information about the teams...just as it exists today

Defn. A FD X→Y is said to be a partial dependency if there is a proper subset of X, Z s.t. Z→Y.

Non eg. The original version of the coaches table contained the following attributes:

Coaches(Id, alma_mater, grad_year, Team_Type, career_wins, career_losses)

The primary key must be Id, Team_Type since one coach can have coached in several sports and therefore have several different career records. This relation has a partial dependency since a coaches alma_mater and grad_year are dependent only on the id.

Defn. A relation is said to be in Second Normal Form (2NF) if it is in 1NF and no attribute is partially dependent on any Primary Key.

You can also see why a non 2NF relation is bad from the example...if a coach did coach two different sports, his/her personal data would appear more than once...the dreaded redundancy.

Defn. A FD X→Y is said to be a transitive dependency if there is a set of attributes Z such that both X→Z and Z→Y are true.

Example: Recall the early version of the teams table:

teams(code,type,wins,losses,coachid, coachln, coachfn, alma_mater, grad_year)

Since every team has exactly one coach,

code,type → coachid

is true, so is

Coachid → coachln, coachfn, alma_mater, grad_year

so

code,type → coachln, coachfn, alma_mater, grad_year

is a transitive dependency.
Defn. An attribute A is called prime or primary if A is part of some candidate key.

Eg. In the colleges table, code, name, city, state are all primary.

Defn A relation is said to be in Third Normal Form (3NF) if it is in 1NF and no non-prime attribute is transitively dependent on the primary key...conversely, transitive dependencies whose left hand side is the primary key are allowed only if the right hand side is a prime attribute..

Non eg. The early version of the teams table is not 3NF.

Defn. A relation is said to be in Boyce-Codd Normal Form if it is in 1NF and the only true FD's are those whose left hand side is a superkey.

Terminology: Given a table with schema R and subsets R₁, R₂,...Rₙ of R, such that R₁∪R₂∪,...∪Rₙ = R. The tables that result from projection onto R₁, R₂,...Rₙ are denoted r₁,r₂,..rn respectively are called a decomposition of R.

Given a table with schema R that violates one or the other of the normal forms, our goal is to find a decomposition of R such that all subtables do satisfy the normal form.

Not just any decomposition will do.

Defn. A decomposition of R is called "lossless" (more properly a lossless join decomposition) if r₁ ◙ r₂ ◙... ◙ rₙ = r.

non-example. If R( A B C D) and r is a b c d
w x y z
then R₁(AB) and R₂(CD) is a legal decomposition, but it is not lossless...

r₁     A  B  r₂     C  D  r₁ ◙ r₂   A  B  C  D
     a   b   c   d  a   b   c   d
     w   x   y   z  a   b   y   z
     w   x   c   d  w   x   y   z

We will insist that any decompositions that we undertake so as to enforce normality be lossless join decompositions..

Testing for a Lossless join decomposition

Input: A relation schema R with a set of FD's which apply to R and a decomposition of R, R₁,R₂,...Rₙ with corresponding sets of FD's Fᵢ
Output: true if the decomposition is a lossless join, false otherwise.
Create a table as follows:
   Each attribute of R will be a label for a column.
   Each subrelation will be the label for a row.

   Place the symbol $a_{ij}$ in row i, column j if the attribute $A_j$ appears in subrelation i.
   Place the symbol $b_{ij}$ in position i,j if attribute $A_j$ does not appear in subrelation i.

E.g. from Ullman:

\[
R(ABCDE) F = \{ A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow C, CE \rightarrow A \}
\]

\[
R_1 (AD) \\
R_2 (AB) \\
R_3 (BE) \\
R_4 (CDE) \\
R_5 (AE)
\]

Leads to the following table

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>a1</td>
<td>b12</td>
<td>b13</td>
<td>a4</td>
<td>b15</td>
</tr>
<tr>
<td>$R_2$</td>
<td>a1</td>
<td>a2</td>
<td>b23</td>
<td>b24</td>
<td>b25</td>
</tr>
<tr>
<td>$R_3$</td>
<td>b31</td>
<td>a2</td>
<td>b33</td>
<td>b34</td>
<td>a5</td>
</tr>
<tr>
<td>$R_4$</td>
<td>b41</td>
<td>b42</td>
<td>a3</td>
<td>a4</td>
<td>a5</td>
</tr>
<tr>
<td>$R_5$</td>
<td>a1</td>
<td>b52</td>
<td>b53</td>
<td>b54</td>
<td>a5</td>
</tr>
</tbody>
</table>

The algorithm proceeds as follows:

For each FD, and each row of the table for which the values on the LHS of the FD agree, modify the table so that the RHS attributes agree as well. If any row has an "a" for a right hand side attribute, make all the equal entries the same “a”. If all the RHS attributes are b's, make the all the same “b”.

Loop around all the functional dependencies until no changes in the matrix are obtained. If at the end, any row of the table is made entirely of “a” symbols, then the decomposition is lossless.

/* In the example, consider the FD $A \rightarrow C$. Three rows of the table contain the symbol a1 in column 1, so change all of the values in column 3 to be b13.
Defn...Let R₁, R₂,...,Rₙ be a decomposition of R and F be a set of FD's that apply to R. Fᵢ denotes the subset of F consisting of those FD's all of whose attributes are in Rᵢ. We say that the decomposition is dependency preserving if

\[ F^+ = (F₁ \cup F₂ \cup ... \cup Fₙ)^+ \]

Our REAL goal is to find lossless, dependency preserving decompositions!
Database Design

One more note...if a relational schema is in 3NF it is automatically in 2NF. If $XY \rightarrow Z$ and $X\rightarrow Z$ are true (a partial dependency) then $XY \rightarrow X$ and $X \rightarrow Z$ forms a transitive dependency. Therefore if we eliminate transitive dependencies we automatically eliminate partial dependencies as well.

Defn. A FD $X \rightarrow A$ is said to be inessential in a set $F$ if it can be removed from $F$ with a result $H$, such that $F^+ = H^+$.

Algorithm: FOR all $X \rightarrow A$ in $F$ /*loop on FDs in F */

$H = F - \{X \rightarrow A\}$ /* remove this FD */

DETERMINE $X^+$ under $H$ /* find new closure of $X$ */

If $A \in X^+$ /* $X \rightarrow A$ is implied by $H$ */

$F \equiv F - \{X \rightarrow A\}$ /* $X \rightarrow A$ is inessential in $H$ */

END-FOR

Defn. Given a relation schema $R$ and set of FD's $F$, we say that another set of FD's, $F_{min}$, is a minimal cover for $F$ if

1. $F_{min}^+ = F^+$, and
2. no proper subset of $F_{min}$ has that property

Finding $F_{min}$ will be an integral part of our algorithm to find lossless join decompositions...

Algorithm: (Find $F_{min}$)

Input: a schema $R$ and set of FD's on $R$, $F$

Output: a minimal cover for $F$, $F_{min}$

1. Rewrite each FD in $F$ (using the projectivity rule) so that the right hand side of every FD contains exactly one attribute.
2. Replace individual FDs with FDs that have a smaller number of attributes on the left-hand side, as long as the result does not change $F^+$. In other words, we cannot replace any dependency $A \rightarrow B$ in $F$ with a dependency $C \rightarrow B$ where $C$ is a proper subset of $A$, and the new set of functional dependencies is equivalent to $F$.
3. From the remaining set of FDs remove any FD's that inessential in $F$.
4. From the remaining set of FDs, collect all FDs with equal left-hand side and use the union rule to create an equivalent set of FDs where all the left-hand sides are unique.

Example: $R(ABCDEFG)$

$$F = \{AB \rightarrow C, D \rightarrow EG, ABC \rightarrow A,\ C \rightarrow A, BE \rightarrow C,\ BC \rightarrow D, CG \rightarrow BD,\ ACD \rightarrow B, CE \rightarrow AG\}$$
Database Design

step 1.

\[
\begin{align*}
\text{AB} & \rightarrow \text{C} & \text{ABC} & \rightarrow \text{A} & \text{D} & \rightarrow \text{E} \\
\text{D} & \rightarrow \text{G} & \text{C} & \rightarrow \text{A} & \text{BE} & \rightarrow \text{C} \\
\text{BC} & \rightarrow \text{D} & \text{CG} & \rightarrow \text{B} & \text{CG} & \rightarrow \text{D} \\
\text{ACD} & \rightarrow \text{B} & \text{CE} & \rightarrow \text{A} & \text{CE} & \rightarrow \text{G}
\end{align*}
\]

step 2.

\[F = \{ \text{AB} \rightarrow \text{C}, \text{ABC} \rightarrow \text{A}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G}, \}
\]

There are only 9 FDs that have more than one attribute in the left-hand side.

Consider \( \text{AB} \rightarrow \text{C} \).

Drop A? Then \( J_1 = \{ \text{B} \rightarrow \text{C}, \text{ABC} \rightarrow \text{A}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( B^+ \) under \( F \) will be the same as \( B^+ \) under \( J_1 \).

\[
\begin{align*}
B^+ (\text{under } F) &= B \\
B^+ (\text{under } J_1) &= BCA
\end{align*}
\]

Drop B? Then \( J_2 = \{ \text{A} \rightarrow \text{C}, \text{ABC} \rightarrow \text{A}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( A^+ \) under \( F \) will be the same as \( A^+ \) under \( J_2 \).

\[
\begin{align*}
A^+ (\text{under } F) &= A \\
A^+ (\text{under } J_2) &= AC
\end{align*}
\]

So we can’t change \( \text{AB} \rightarrow \text{C} \).

Consider \( \text{ABC} \rightarrow \text{A} \).

Drop A? Then \( J_3 = \{ \text{AB} \rightarrow \text{C}, \text{ABC} \rightarrow \text{A}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( BC^+ \) under \( F \) will be the same as \( BC^+ \) under \( J_3 \).

\[
\begin{align*}
BC^+ (\text{under } F) &= BCADGE \\
BC^+ (\text{under } J_3) &= BCADGE
\end{align*}
\]

So we can replace \( \text{ABC} \rightarrow \text{A} \) by \( \text{BC} \rightarrow \text{A} \).

Consider now \( \text{BC} \rightarrow \text{A} \).

Drop B? Then \( J_4 = \{ \text{ABC} \rightarrow \text{A}, \text{C} \rightarrow \text{A}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( C^+ \) under \( F \) will be the same as \( C^+ \) under \( J_4 \).

\[
\begin{align*}
C^+ (\text{under } F) &= CA \\
C^+ (\text{under } J_4) &= CA
\end{align*}
\]

But the new dependency is equivalent to one we already have. So remove it.

Consider \( \text{BE} \rightarrow \text{C} \).

Drop B? Then \( J_5 = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{E} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( E^+ \) under \( F \) will be the same as \( E^+ \) under \( J_5 \).

\[
\begin{align*}
E^+ (\text{under } F) &= E \\
E^+ (\text{under } J_5) &= ECA
\end{align*}
\]
Database Design

Drop E? Then \( J_6 = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{B} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( \text{B}^+ \) under \( F \) will be the same as \( \text{B}^+ \) under \( J_6 \).
\( \text{B}^+ \) (under \( F \)) = B
\( \text{B}^+ \) (under \( J_6 \)) = BCADEG
So we can’t change \( \text{BE} \rightarrow \text{C} \)

Consider \( \text{BC} \rightarrow \text{D} \)

Drop B? Then \( J_7 = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{C} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( \text{C}^+ \) under \( F \) will be the same as \( \text{C}^+ \) under \( J_7 \).
\( \text{C}^+ \) (under \( F \)) = CA
\( \text{C}^+ \) (under \( J_7 \)) = CDGAE

Drop C? Then \( J_8 = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{B} \rightarrow \text{D}, \text{CG} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( \text{B}^+ \) under \( F \) will be the same as \( \text{B}^+ \) under \( J_8 \).
\( \text{B}^+ \) (under \( F \)) = B
\( \text{B}^+ \) (under \( J_8 \)) = BDGE
So we can’t change \( \text{BC} \rightarrow \text{D} \)

Consider \( \text{CG} \rightarrow \text{B} \)

Drop C? Then \( J_9 = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{G} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( \text{G}^+ \) under \( F \) will be the same as \( \text{G}^+ \) under \( J_9 \).
\( \text{G}^+ \) (under \( F \)) = G
\( \text{G}^+ \) (under \( J_9 \)) = GB

Drop G? Then \( J_{10} = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{C} \rightarrow \text{B}, \text{CG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( \text{C}^+ \) under \( F \) will be the same as \( \text{C}^+ \) under \( J_{10} \).
\( \text{C}^+ \) (under \( F \)) = CA
\( \text{C}^+ \) (under \( J_{10} \)) = CABDGE
So we can’t change \( \text{CG} \rightarrow \text{B} \)

Consider \( \text{CG} \rightarrow \text{D} \)

Drop C? Then \( J_{11} = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{BG} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{A}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( \text{G}^+ \) under \( F \) will be the same as \( \text{G}^+ \) under \( J_{11} \).
\( \text{G}^+ \) (under \( F \)) = G
\( \text{G}^+ \) (under \( J_{11} \)) = GDE

Drop G? Then \( J_{12} = \{ \text{AB} \rightarrow \text{C}, \text{D} \rightarrow \text{E}, \text{D} \rightarrow \text{G}, \text{C} \rightarrow \text{A}, \text{BE} \rightarrow \text{C}, \text{BC} \rightarrow \text{D}, \text{C} \rightarrow \text{D}, \text{ACD} \rightarrow \text{B}, \text{CE} \rightarrow \text{G} \} \) should be equivalent to \( F \), i.e., \( \text{C}^+ \) under \( F \) will be the same as \( \text{C}^+ \) under \( J_{12} \).
\( \text{C}^+ \) (under \( F \)) = CA
\( \text{C}^+ \) (under \( J_{12} \)) = CADEG
So we can’t change \( \text{CG} \rightarrow \text{D} \)

Consider \( \text{ACD} \rightarrow \text{B} \)
Drop A? Then $J_{13} = \{ AB \rightarrow C, D \rightarrow E, D \rightarrow G, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow B, CG \rightarrow D, CD \rightarrow B, CE \rightarrow A, CE \rightarrow G \}$ should be equivalent to $F$, i.e., $CD^+ \text{ under } F$ will be the same as $CD^+ \text{ under } J_{13}$.

$CD^+ \text{ (under } F) = \text{CDABEG}$

$CD^+ \text{ (under } J_{13}) = \text{CDABEG}$

So we can drop A

Drop C? Then $J_{14} = \{ AB \rightarrow C, D \rightarrow E, D \rightarrow G, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow B, CG \rightarrow D, AD \rightarrow B, CE \rightarrow A, CE \rightarrow G \}$ should be equivalent to $F$, i.e., $AD^+ \text{ under } F$ will be the same as $AD^+ \text{ under } J_{14}$.

$AD^+ \text{ (under } F) = \text{ADEG}$

$AD^+ \text{ (under } J_{14}) = \text{ADBCEG}$

Drop D? Then $J_{15} = \{ AB \rightarrow C, D \rightarrow E, D \rightarrow G, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow B, CG \rightarrow D, AC \rightarrow B, CE \rightarrow A, CE \rightarrow G \}$ should be equivalent to $F$, i.e., $AC^+ \text{ under } F$ will be the same as $AC^+ \text{ under } J_{15}$.

$AC^+ \text{ (under } F) = AC$

$AC^+ \text{ (under } J_{15}) = \text{ACBDEG}$

So we can change $ACD \rightarrow B$ to $CD \rightarrow B$

Consider CE $\rightarrow$ A

Drop C? Then $J_{16} = \{ AB \rightarrow C, D \rightarrow E, D \rightarrow G, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow B, CG \rightarrow D, CD \rightarrow B, E \rightarrow A, CE \rightarrow G \}$ should be equivalent to $F$, i.e., $E^+ \text{ under } F$ will be the same as $E^+ \text{ under } J_{16}$.

$E^+ \text{ (under } F) = E$

$E^+ \text{ (under } J_{16}) = EA$

Drop E? Then $J_{17} = \{ AB \rightarrow C, D \rightarrow E, D \rightarrow G, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow B, CG \rightarrow D, CD \rightarrow B, C \rightarrow A, CE \rightarrow G \}$ should be equivalent to $F$, i.e., $C^+ \text{ under } F$ will be the same as $C^+ \text{ under } J_{17}$.

$C^+ \text{ (under } F) = CA$

$C^+ \text{ (under } J_{17}) = CA$

But the new dependency is equivalent to one we already have. So remove it.

Consider CE $\rightarrow$ G

Drop C? Then $J_{18} = \{ AB \rightarrow C, D \rightarrow E, D \rightarrow G, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow B, CG \rightarrow D, CD \rightarrow B, E \rightarrow G \}$ should be equivalent to $F$, i.e., $E^+ \text{ under } F$ will be the same as $E^+ \text{ under } J_{18}$.

$E^+ \text{ (under } F) = E$

$E^+ \text{ (under } J_{18}) = EG$

Drop E? Then $J_{19} = \{ AB \rightarrow C, D \rightarrow E, D \rightarrow G, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow B, CG \rightarrow D, CD \rightarrow B, C \rightarrow G \}$ should be equivalent to $F$, i.e., $C^+ \text{ under } F$ will be the same as $C^+ \text{ under } J_{19}$.

$C^+ \text{ (under } F) = CA$

$C^+ \text{ (under } J_{19}) = CGA$

So we can’t change CE $\rightarrow$ G

So the set of functional dependencies after this step is:
Database Design

\[ F = \{ AB \to C, D \to E, D \to G, C \to A, BE \to C, BC \to D, CG \to B, CG \to D, CD \to B, CE \to G \} \]

step 3.

Consider \( AB \to C \). In \( F - \{ AB \to C \} \), \( AB^+ = AB \). So \( AB \to C \) is essential.

Consider \( D \to E \). In \( F - \{ D \to E \} \), \( D^+ = DG \). So \( D \to E \) is essential.

Consider \( D \to G \). In \( F - \{ D \to G \} \), \( D^+ = DE \). So \( D \to G \) is essential.

Consider \( C \to A \). In \( F - \{ C \to A \} \), \( C^+ = C \). So \( C \to A \) is essential.

Consider \( BE \to C \). In \( F - \{ BE \to C \} \), \( BE^+ = BE \), so \( BE \to C \) is essential.

Consider \( BC \to D \). In \( F - \{ BC \to D \} \), \( BC^+ = BCA \), so \( BC \to D \) is essential.

Consider \( CG \to B \). In \( F - \{ CG \to B \} \), \( CG^+ = CGADB \), so \( CG \to B \) is inessential. (Remove it)

Consider \( CD \to B \). In \( F - \{ CD \to B \} \), \( CD^+ = CDAEG \), so \( ACD \to B \) is essential.

Consider \( CE \to G \). In \( F - \{ CE \to G \} \), \( CE^+ = CEA \), so \( CE \to G \) is essential.

So our \( F \) is now

\[
\begin{align*}
AB & \to C & ABC & \to A & D & \to E \\
D & \to G & C & \to A & BE & \to C \\
BC & \to D & CG & \to B & CG & \to D \\
CD & \to B & CE & \to A & CE & \to G
\end{align*}
\]

We call it \( F \)

step 4.

\[ H = \{ AB \to C, D \to E, D \to G, C \to A, BE \to C, BC \to D, CG \to D, CD \to B, CE \to G \} \]

Now we collect FDs to obtain:

\[ F_{\text{min}} = \{ AB \to C, D \to EG, C \to A, BE \to C, BC \to D, CG \to D, CD \to B, CE \to G \} \]