Functional Dependency

Concepts:

- anomalies due to “redundency”: e.g., cannot represent some situation (*insertion anomaly*).
- Armstrong’s Axioms. (Reflexivity, Augmentation, Transitivity) + (Union, Decomposition, Pseudo-transitivity).
- Attribute closure (*very useful tool*)
- FD closure, minimal cover
- Decomposition into BCNF / 3NF
Overview

Legend:
- concept
- key concept

If you really understand what is **the projection of** $F$, you should have no problem in FD-and-NF theories. If not, try Q2 as you have nothing to lose, :p
## Normal Forms

### Comparison

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Meaning</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>3NF</td>
<td>...</td>
<td>Atomic values + No partial or transitive dependency</td>
<td>Decomp. Alg.</td>
</tr>
<tr>
<td>BCNF</td>
<td>...</td>
<td>3NF + Non-trivial FD must be a key constraint</td>
<td>Decomp. Alg.</td>
</tr>
</tbody>
</table>
3NF vs. BCNF

Trade-offs of 3NF and BCNF:
- Decomposition into BCNF might not be dependency preserving.
- Performance consideration.

Examples of *de-normalization*:
- Frequent query, avoid joins.
- Data warehouse.

Special rules for BCNF:
- $AB$ is always in BCNF.
- $XABC \ldots$, while $F = \{X \rightarrow A, X \rightarrow B, \ldots\}$
Below are the patient test results forms from the Getwellquik Hospital.

### Getwellquik Hospital Patient Test Result Form

<table>
<thead>
<tr>
<th>Test ID</th>
<th>Name</th>
<th>Sample Taken</th>
<th>Normal Range</th>
<th>Date/Time</th>
<th>Result</th>
<th>Ordered By</th>
</tr>
</thead>
<tbody>
<tr>
<td>10223</td>
<td>Urea</td>
<td>5ml blood in lithium heparin tube</td>
<td>3–8</td>
<td>12/03/2006 7:50am</td>
<td>3.0</td>
<td>Dr. John Smith</td>
</tr>
<tr>
<td>10334</td>
<td>Hb (male)</td>
<td>5ml blood in EDTA tube</td>
<td>130–180</td>
<td>12/03/2006 8:00am</td>
<td>150.0</td>
<td>Dr. John Smith</td>
</tr>
<tr>
<td>10223</td>
<td>Urea</td>
<td>5ml blood in lithium heparin tube</td>
<td>3–8</td>
<td>15/03/2006 2:00pm</td>
<td>5.1</td>
<td>Dr. Helen Long</td>
</tr>
</tbody>
</table>

### Getwellquik Hospital Patient Test Result Form

<table>
<thead>
<tr>
<th>Test ID</th>
<th>Name</th>
<th>Sample Taken</th>
<th>Normal Range</th>
<th>Date/Time</th>
<th>Result</th>
<th>Ordered By</th>
</tr>
</thead>
<tbody>
<tr>
<td>10665</td>
<td>bilirubin (total)</td>
<td>5ml blood in lithium heparin tube</td>
<td>0–7</td>
<td>25/03/2006 3:00pm</td>
<td>4.0</td>
<td>Dr. Jack Straw</td>
</tr>
</tbody>
</table>
### A Hospital Example II

#### Getwellquik Hospital Patient Test Result Form

<table>
<thead>
<tr>
<th>Test ID</th>
<th>Name</th>
<th>Sample Taken</th>
<th>Normal Range</th>
<th>Date/Time</th>
<th>Result</th>
<th>Ordered By</th>
</tr>
</thead>
<tbody>
<tr>
<td>10334</td>
<td>Hb (female)</td>
<td>5ml blood in EDTA tube</td>
<td>115–165</td>
<td>30/03/2006 4:00pm</td>
<td>150.0</td>
<td>Dr. Ling Woo</td>
</tr>
<tr>
<td>20044</td>
<td>Platelets</td>
<td>5ml blood in EDTA tube</td>
<td>150–400</td>
<td>30/03/2006 4:05pm</td>
<td>23.4</td>
<td>Dr. Ling Woo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test ID</th>
<th>Name</th>
<th>Sample Taken</th>
<th>Normal Range</th>
<th>Date/Time</th>
<th>Result</th>
<th>Ordered By</th>
</tr>
</thead>
<tbody>
<tr>
<td>20044</td>
<td>Platelets</td>
<td>5ml blood in EDTA tube</td>
<td>150–400</td>
<td>24/03/2006 12:00pm</td>
<td>300.0</td>
<td>Dr. Helen Long</td>
</tr>
</tbody>
</table>

1. Identify all the possible attributes to form a universal relation $R$ from the above data.

2. Identify FDs and decompose $R$ into “good forms”. Find the highest normal forms your decomposed relations conform to. (Make reasonable assumptions when identifying FDs)
A possible schema is:

\[ R(\text{pno}, \text{pname}, \text{wno}, \text{bno}, \text{wname}, \text{testid}, \text{tname}, \text{tsample}, \text{trange}, \text{ttime}, \text{tresult}, \text{doctor}) \]

<table>
<thead>
<tr>
<th>pno</th>
<th>pname</th>
<th>wno</th>
<th>bno</th>
<th>wname</th>
<th>testid</th>
<th>tname</th>
<th>tsample</th>
<th>trange</th>
<th>ttime</th>
<th>tresult</th>
<th>doctor</th>
</tr>
</thead>
<tbody>
<tr>
<td>P10034</td>
<td>Robert McDonald</td>
<td>11</td>
<td>84</td>
<td>Orthopedic</td>
<td>10223</td>
<td>Urea</td>
<td>5 mL blood in lithium heparin or plain tube</td>
<td>3 – 8</td>
<td>12/03/2006 7:50am</td>
<td>3.0</td>
<td>Dr. John Smith</td>
</tr>
<tr>
<td>P10034</td>
<td>Robert McDonald</td>
<td>11</td>
<td>84</td>
<td>Orthopedic</td>
<td>10334</td>
<td>Hb (male)</td>
<td>5 mL blood in EDTA tube</td>
<td>130–180</td>
<td>12/03/2006 8:00am</td>
<td>150.0</td>
<td>Dr. John Smith</td>
</tr>
<tr>
<td>P10034</td>
<td>Robert McDonald</td>
<td>11</td>
<td>84</td>
<td>Orthopedic</td>
<td>10223</td>
<td>Urea</td>
<td>5 mL blood in lithium heparin or plain tube</td>
<td>3–8</td>
<td>15/03/2006 2:00pm</td>
<td>5.1</td>
<td>Dr. Helen Long</td>
</tr>
<tr>
<td>P2411</td>
<td>Sue Allen</td>
<td>5</td>
<td>6</td>
<td>Maternity</td>
<td>10665</td>
<td>bilirubin (total)</td>
<td>5 mL blood in lithium heparin or plain tube</td>
<td>0–7</td>
<td>25/03/2006 3:00pm</td>
<td>4.0</td>
<td>Dr. Jack Straw</td>
</tr>
<tr>
<td>P1175</td>
<td>Helen Tan</td>
<td>12</td>
<td>35</td>
<td>Surgical</td>
<td>10334</td>
<td>Hb (female)</td>
<td>5 mL blood in EDTA tube</td>
<td>115–165</td>
<td>30/03/2006 4:00pm</td>
<td>80.0</td>
<td>Dr. Ling Woo</td>
</tr>
<tr>
<td>P1175</td>
<td>Helen Tan</td>
<td>12</td>
<td>35</td>
<td>Surgical</td>
<td>20044</td>
<td>Platelets</td>
<td>5 mL blood in EDTA tube</td>
<td>150–400</td>
<td>30/03/2006 4:05pm</td>
<td>23.4</td>
<td>Dr. Ling Woo</td>
</tr>
<tr>
<td>P1234</td>
<td>George Smith</td>
<td>11</td>
<td>83</td>
<td>Orthopedic</td>
<td>20044</td>
<td>Platelets</td>
<td>5 mL blood in EDTA tube</td>
<td>150–400</td>
<td>24/03/2006 12:00pm</td>
<td>300.0</td>
<td>Dr. Helen Long</td>
</tr>
</tbody>
</table>

The set of FDs we can identify from the data are:
<table>
<thead>
<tr>
<th>FD</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pno \rightarrow pname$</td>
<td></td>
</tr>
<tr>
<td>$pno \rightarrow wno$</td>
<td></td>
</tr>
<tr>
<td>$bno \rightarrow wno$</td>
<td>Seems to be the case from the data and maybe quite reasonable in practice</td>
</tr>
<tr>
<td>$wno \rightarrow wname$</td>
<td>Note: not the other way around</td>
</tr>
<tr>
<td>$tname \rightarrow tid$</td>
<td></td>
</tr>
<tr>
<td>$tname \rightarrow trange, tsample$</td>
<td></td>
</tr>
</tbody>
</table>

(Some FDs are obviously not generalizable, e.g., $tresult \rightarrow doctor$ and thus should not be included)

Based on the above identified FDs, we can obtain the following (lossless-join) decomposition:

Patient $(pno, pname)$
Bed $(bno, wno)$
Ward $(wno, wname)$
Test $(tname, testid, trange, tsample)$
HasTest $(pno, tname, ttime, doctor, tresult)$
Solution III

Note that the last relation include $tresult$ into the PK because we didn't identify the FD:

$$pno, tname, ttime, doctor \rightarrow tresult$$

It is obvious that all the decomposed relation are in BCNF.
Armstrong Axioms describes certain properties about functional dependency.

\[ Y \subseteq X \implies X \rightarrow Y \] \hspace{1cm} (1)
\[ X \rightarrow Y \implies XZ \rightarrow YZ \] \hspace{1cm} (2)
\[ X \rightarrow Y, \ Y \rightarrow Z \implies X \rightarrow Z \] \hspace{1cm} (3)

We can also infer the following:

\[ X \rightarrow YZ \implies X \rightarrow Y, \ X \rightarrow Z \] \hspace{1cm} (4)
\[ X \rightarrow Y, \ X \rightarrow Z \implies X \rightarrow YZ \] \hspace{1cm} (5)

Abstractly, we can view \( \rightarrow \) as a binary function, i.e., a function that takes two parameters and returns a Boolean value. Or formally, \( X \rightarrow Y \) is a shorthand for \( \rightarrow (X, Y) \), just as \( X + Y \) for \( +(X, Y) \).

With this, let’s relate FDs to something we are very familiar with. Now let’s “read” \( \rightarrow \) as the “can be divided by” function, and treat each individual attribute as a positive integer. For example, \( X \rightarrow Y \) is read as “\( X \) can be divided by \( Y \)” ; if \( X = 105 \) and \( Y = 35 \), this function evaluates to TRUE.
Immediately, Equation (3) makes sense now: if $X$ can be divided by $Y$, and $Y$ can be divided by $Z$, then of course $X$ can be divided by $Z$. (Try $X = 105$, $Y = 35$, and $Z = 7$)

Answer the following questions:

- Find a way to define (or “read”) the binary function of union (You can assume $X$, $Y$, $Z$ are all individual attributes for now), so that we can Equation (2) is correct under that interpretation.
- Under your definition of “union”, does Equation (4) hold? Does Equation (5) hold? Can you fix your definition of “union” so that both of them hold?
- Using the interpretation, explain why we still bother to computer $F^+$ even if a set of functional dependencies, $F$, is given. Explain the meaning of projecting $F$ onto some subset of attributes.
Solution 1

- We may define union as multiplication. Hence \( X \cup Y = X \cdot Y \). Then Equation (2) says: if \( X \) can be divided by \( Y \), then for any \( Z \), \( X \cdot Z \) can be divided by \( Y \cdot Z \) (obviously true).

- Under the above interpretation, Equation (4) still holds, but Equation (5) breaks down. E.g., \( 30 \rightarrow 6, 30 \rightarrow 15 \), but \( 30 \not\rightarrow 6 \cdot 15 \).
  To fix this, we need to define “union” as the least common multiple (lcm) function (http://en.wikipedia.org/wiki/Least_common_multiple).
  All the Equations work then. E.g., now we have \( 30 \rightarrow \text{lcm}(6, 15) = 30 \).

- Consider, e.g., \( R = ABCD = \{ A, B, C, D \} \) merely as a collection of four unknown positive integers. \( F \) tells you, in a concise way, all the constraints that holds among any “instance” of them (i.e., any possible tuple of \( R \)). \( F^+ \) then includes all the constraint that must hold! The projection of \( F \) onto any subset of attributes includes all the constraint that must hold on the subset. Now it is obvious that you need to perform simple projections from \( F^+ \), not \( F \).
FYI, this exercise gives you a glimpse into how abstract algebra views the computation: the “names” of the functions (→ and ∪ here) and element (attributes here) do not matter; it is the “structure” and certain “properties” that matter.

Programming Language examples: templates (C++), Generics (Java) and Maybe (Haskell: http://en.wikibooks.org/wiki/Haskell/Understanding_monads)
Consider the relation $R(A, B, C, D, E, F, G)$ and the set of functional dependencies

$$F = \{ \begin{align*}
A & \rightarrow B, \\
BC & \rightarrow F, \\
BD & \rightarrow EG, \\
AD & \rightarrow C, \\
D & \rightarrow F, \\
BEG & \rightarrow FA \} 
\end{align*}$$

Answer the following questions:

- $A^+$
- $ACEG^+$
- $BD^+$
- List all the (candidate) keys of $R$. 

Wei Wang (UNSW)

COMP7640 @ HKBU TUT4

October 17, 2014 15 / 23
Solution I

- To calculate $A^+$:
  
  $$A^+ = \{ A \}$$  
  
  $$= \{ A, B \}$$  

  given  
  $\therefore A \rightarrow B$

- To calculate $ACEG^+$
  
  $$ACEG^+ = \{ ACEG \}$$  
  
  $$= \{ ABCEG \}$$  

  given  
  $\therefore A \rightarrow B$

  $$= \{ ABCEFG \}$$  

  $\therefore BC \rightarrow F$
To calculate $BD^+$

$$BD^+ = \{ BD \}, \quad \text{given}$$
$$= \{ BDEG \}, \quad \therefore BD \rightarrow EG$$
$$= \{ ABDEFG \}, \quad \therefore BEG \rightarrow FA$$
$$= \{ ABCDEFG \}, \quad \therefore AD \rightarrow C$$

Therefore, $BD$ is a candidate key.

List all the (candidate) keys of $R$:

- we can make two important observations: (1) $D$ must be in the key (2) Other keys don’t include $B$. (why?)
- Consider the LHS’s of all FDs. Considering candidates (starting from adding one attribute): e.g., $AD$, … Easy to calculate that $AD^+ = \ldots = ABCDEFG$ and thus it is a candidate key. (why?)
- Try “trivial keys”: $D^+ = DF$, so need to try to add $ABCEG$ yet need to avoid $A$ and $B$, so only need to consider $CEG$. 
Solution III

- $CDEG^+ = \ldots = CDEFG$, so we need $A$ or $B$ on the LHS. Therefore, only two keys: $AD$ and $BD$. 
Consider the following relation $R(A, B, C, D, E)$, with the following set $F$ of functional dependencies:

$$F = \{A \rightarrow BC, \quad CD \rightarrow E, \quad B \rightarrow D, \quad E \rightarrow A\}$$

1. If we decompose it into $R_1 = (A, B, C)$ and $R_2 = (A, D, E)$, is the decomposition lossless join decomposition?

2. Calculate $B^+$. 

3. List all the candidate keys of $R$. 

4. Give a lossless decomposition of $R$ into BCNF.

5. Give a lossless, dependency-preserving decomposition into 3NF of $R$. 
Yes. Because \( R_1 \cap R_2 = A \) and \( A \to R_1 \).

Initially, \( B^+ = B \). Do the loop:

1. \( B^+ = BD \), due to \( B \to D \).

\( B^+ \) cannot be further “grown”. Therefore, \( B^+ = BD \)

Two methods to do this:

1. (Method 1) Note: It is not reasonable to expect someone to enumerate all elements of \( F^+ \). Some shorthand representation of the result should be acceptable as long as the nontrivial members of \( F^+ \) are found.

   1. Starting with \( A \to BC \), we can conclude: \( A \to B \) and \( A \to C \).
   2. Since \( A \to B \) and \( B \to D \), \( A \to D \) (decomposition, transitive)
   3. Since \( A \to CD \) and \( CD \to E \), \( A \to E \) (union, decomposition, transitive)
   4. Since \( A \to A \), we have (reflexive) \( A \to ABCDE \) from the above steps (union)
   5. Since \( E \to A \), \( E \to ABCDE \) (transitive)
   6. Since \( CD \to E \), \( CD \to ABCDE \) (transitive)
   7. Since \( B \to D \) and \( BC \to CD \), \( BC \to ABCDE \) (augmentative, transitive)
Solution II

In addition to the non-trivial FDs listed above, we also have *lots of* trivial FDs in $F^+$, such as $C \rightarrow C$, $D \rightarrow D$, $BD \rightarrow D$, etc. Therefore, any functional dependency with $A$, $E$, $BC$, or $CD$ on the left hand side of the arrow is in $F^+$, no matter which other attributes appear in the FD. Allow * to represent any set of attributes in $R$, then $F^+$ is $BD \rightarrow B$, $BD \rightarrow D$, $C \rightarrow C$, $D \rightarrow D$, $BD \rightarrow BD$, $B \rightarrow D$, $B \rightarrow B$, $B \rightarrow BD$, and all FDs of the form $A^* \rightarrow \alpha$, $BC^* \rightarrow \alpha$, $CD^* \rightarrow \alpha$, $E^* \rightarrow \alpha$, where $\alpha$ is any subset of \{A, B, C, D, E\}. The candidate keys are $A$, $BC$, $CD$, and $E$.

(Method 2) (Optional) since all the attributes appear once on the RHS, there is no “shortcut”. So, we proceed to test each subset of $R$ from bottom up (Question: why not top-down?).

Examine (mainly) 1-attribute set:

- $A^+ = ABCDE = R$.
- Since $E^+ \subseteq R$ and $E \rightarrow A$, we know that $R \supseteq E^+ \supseteq A^+ = R$, so $E^+ = R$ (of course, you can do the attribute closure computation for $E$, but this kind of reasoning is faster).
- Similarly, since $CD \rightarrow E$, . . . , we know $CD^+ = R$ (can do this later, but it comes for free here).
Since $B^+ = BD$, $B$ is not a candidate key. Similarly, since neither $C$ nor $D$ appear on the LHS of FDs, they are not candidate keys either (because their attribute closure cannot be $R$). So $CD$ is the candidate key.

At the end of this round, we know the following facts:
- $A, E, CD$ are already candidate keys. An important corollary is: we do not need to consider any subset of $R$ that contains any of the existing candidate keys, i.e. $A, E, CD$ in this example.
- Since $B, C, D$ are not candidate key, need to examine any superset of them. But according to the above corollary, we do not need to examine $AB$, for example.

2. Examine 2-attribute set: According to the discoveries made in the previous round, we only need to examine the following 2-attribute candidates: $BC$ and $BD$. It is easily computed that $BC^+ = R$ and $BD^+ = BD$. Therefore, $BC$ is also the candidate key.

3. Examine 3-attribute set: We do not have any candidate 3-attribute set to test, as it must be a superset of the existing set of candidate key (Verify this).

Therefore, the candidate keys are $\{A, E, CD, BC\}$. (Note: we can arrive at the same “description” of $F^+$ too)
First, we need to test if \( R \) is in BCNF or not. We know \( R \) is not in BCNF because of the FD \( B \rightarrow D \) (as we have calculated all the candidate keys of \( R \), or just verify if \( B^+ \) is \( R \)). Therefore, we need to do the decomposition as follows:

1. Decompose \( R \) into \( R_1 = BD \) and \( R_2 = ABCE \). \( R_1 \) is definitely in BCNF \( \text{(why?)} \). So we need to test if \( R_2 \) is in BCNF, as follows:
   - By following the definition of the restriction of \( F \) on \( R_2 \), we know that \( A, E, BC \) must still be the candidate keys on \( R_2 \).
   - Test every subset of \( R_2 \) (to see if it is a “bad” FD or not). Given the knowledge about some of the candidate keys on \( R_2 \), we only need to examine \( B, C \) \( \text{(Question: why?)} \). It is easily verified that on \( R_2 \) \( B^+ = B \) and \( C^+ = C \). Therefore, no bad FDs.

Therefore, both \( R_1 \) and \( R_2 \) are in BCNF.

First, we need to test if \( R \) is in 3NF or not. Given the knowledge of all the candidate keys, we know that \( R \) is already in 3NF. So we do not need any decomposition.