Ian Kenny

November 14, 2016

Databases

Lecture 7

In this lecture

- Functional dependencies continued.

Functional dependencies

We saw in the previous session that a functional dependency (FD) is a relationship between sets of attributes.

A functional dependency exists between a set of attributes $\bar{A}$ and a set of attributes $\bar{B}$ if, for all tuples, where the tuples have equal values for attributes $\bar{A}$ then they will have equal values for attributes $\bar{B}$.

For two tuples t and u in relation R, $\bar{A} \rightarrow \bar{B}$ means

$\forall t, u \in R$

$$t[A_1, A_2, ..., A_n] = u[A_1, A_2, ..., A_n] \implies t[B_1, B_2, ..., B_m] = u[B_1, B_2, ..., B_m]$$

Functional dependencies

Functional dependencies can be real or apparent. Apparent FDs arise from the data but could be violated by new data. Real FDs hold for all data. Real FDs are a constraint on a relation.

We can analyse the data itself for FDs. Some of those arising from this process will be real and some apparent. However, this process will enable us to find all FDs but many of them may say little of interest about a specific domain.

In order to identify the FDs that we are genuinely interested in, we also need to use domain knowledge (knowledge of the enterprise).

The closure of a set of attributes

Given a set of attributes and a set of FDs, we can compute the *closure* of those attributes.

The closure of a set of attributes $\bar{A}$, denoted A^+ is the set of attributes that are functionally determined by $\bar{A}$.

Closure example

Consider the following relation

$R(A, B, C)$

And the set of FDs on the relation

$A \rightarrow B$ (there's only one FD in this set for now).

We compute the closure of each subset of attributes of R by, for each subset, adding to a set the attributes functionally determined by that subset of attributes. An example should make that clearer.

Closure of a set of attributes: example

$R(A, B, C)$; $A \rightarrow B$

$\{A\}^+ = \{A, B\}$
$\{B\}^+ = \{B\}$
$\{C\}^+ = \{C\}$
$\{A, B\}^+ = \{A, B\}$
$\{A, C\}^+ = \{A, B, C\}$
$\{B, C\}^+ = \{B, C\}$
$\{A, B, C\}^+ = \{A, B, C\}$

Closure of a set of attributes: example

Which *other* FDs are implied by this closure?

We are given $A \rightarrow B$

From the closure we can see

$A \rightarrow A, B$
$A \rightarrow A$ (splitting rule)
$B \rightarrow B$
$C \rightarrow C$
$A, B \rightarrow A, B$
$A, C \rightarrow A, B, C$
$A, C \rightarrow A$ (splitting rule)
$A, C \rightarrow B$ (splitting rule)
$A, C \rightarrow C$ (splitting rule)
$A, B, C \rightarrow A, B, C$

Keys

There is a relationship between FDs and keys.

A *superkey* is any set of attributes in a relation that can be used to identify each row uniquely.

A *minimal superkey* is a superkey that has no redundant attributes in the key. A minimal superkey is called a *candidate key*. Thus, candidate keys are the superkeys that have the fewest attributes. A candidate key can be selected as the primary key for a table.

Clearly, a key needs to be able to functionally determine all of the attributes of a relation. That is the purpose of a key.

Thus, speaking in terms of FDs, a candidate key for a relation is a minimal set of attributes that form the left-hand side of an FD that has all other attributes on the right-hand side.

Closure of a set of attributes: example

The following keys are implied by the closure we computed

$\{A\}^+ = \{A, B\}$
$\{B\}^+ = \{B\}$
$\{C\}^+ = \{C\}$
$\{A, B\}^+ = \{A, B\}$
$\{A, C\}^+ = \{A, B, C\}$ minimal superkey $=$ candidate key
$\{B, C\}^+ = \{B, C\}$
$\{A, B, C\}^+ = \{A, B, C\}$ superkey

The FD for the candidate key is

$A, C \rightarrow A, B, C$

This makes sense. A determines B so we can get both A and B from A. C is not determined by any other attribute so must be included in the key.

Another closure example

Consider a more complex example with relation

$R(A, B, C, D)$

And a set of FDs on R

$\{A \rightarrow B, B \rightarrow D\}$

Another closure example

The closures:

$\{A\}^+ = \{A, B, D\}$ (D via the transitive rule)
$\{B\}^+ = \{B, D\}$
$\{C\}^+ = \{C\}$
$\{D\}^+ = \{D\}$
$\{A, B\}^+ = \{A, B, D\}$
$\{A, C\}^+ = \{A, B, C, D\}$
$\{A, D\}^+ = \{A, B, D\}$
$\{B, D\}^+ = \{B, D\}$
$\{B, C\}^+ = \{B, D, C\}$
$\{A, B, C\}^+ = \{A, B, C, D\}$
$\{A, B, D\}^+ = \{A, B, D\}$
$\{A, C, D\}^+ = \{A, B, C, D\}$
$\{B, C, D\}^+ = \{B, C, D\}$
$\{A, B, C, D\}^+ = \{A, B, C, D\}$

Another closure example

Which FDs are implied?

$R(A, B, C, D)$

We start with

$\{A \to B, B \to D\}$

Another closure example

$A \rightarrow B$
$B \rightarrow D$
$A \rightarrow D$ (transitive rule)
$A \rightarrow B, D$ (combining rule)
$A, C \rightarrow B, C$ (augmentation)
$A, C \rightarrow B$ (splitting rule)
$A, C \rightarrow D$ (transitive rule)
$A, C \rightarrow B, D$ (combining rule)

This means that A,C functionally determines B,D, hence AC is a key for the relation since

If $A, C \rightarrow B, D$ then

$A, C \rightarrow A, B, C, D$

This confirms the result of the closure.

Another closure example

The keys:

$\{A\}^+ = \{A, B, D\}$
$\{B\}^+ = \{B, D\}$
$\{C\}^+ = \{C\}$
$\{D\}^+ = \{D\}$
$\{A, B\}^+ = \{A, B, D\}$
$\{A, C\}^+ = \{A, B, C, D\}$ candidate key
$\{A, D\}^+ = \{A, B, D\}$
$\{B, D\}^+ = \{B, D\}$
$\{B, C\}^+ = \{B, D, C\}$
$\{A, B, C\}^+ = \{A, B, C, D\}$ superkey
$\{A, B, D\}^+ = \{A, B, D\}$
$\{A, C, D\}^+ = \{A, B, C, D\}$ superkey
$\{B, C, D\}^+ = \{B, C, D\}$
$\{A, B, C, D\}^+ = \{A, B, C, D\}$ superkey

Finding an implied FD

We may need to find if a specific FD is implied. Consider relation R

$R(A, B, C, D, E)$

And the set of FDs

$A \rightarrow B$
$A \rightarrow C$
$B, C \rightarrow D, E$

We want to determine if $A \rightarrow E$.

Finding an implied FD

We can compute the closure of A.

The FDs again

$A \rightarrow B$
$A \rightarrow C$
$B, C \rightarrow D, E$

The closure of A (shown in steps. Only the final line is the closure)

1. $\{A\}^+ = \{A\}$
2. $\{A\}^+ = \{A, B\}$
3. $\{A\}^+ = \{A, B, C\}$
4. $\{A\}^+ = \{A, B, C, D, E\}$

Therefore A implies E hence $A \rightarrow E$.

A is, in fact, a candidate key for the relation.

Finding an implied FD

We can also follow the rules for manipulating FDs.

We start with

$A \rightarrow B$
$A \rightarrow C$
$B, C \rightarrow D, E$

The following FDs are implied

$A \rightarrow B, C$ (combining rule)
$A \rightarrow D, E$ (transitive rule)
$A \rightarrow D$ (splitting rule)
$A \rightarrow E$ (splitting rule)

Therefore $A \rightarrow E$.

Another approach to finding a key

Consider the relation R

$R(A, B, C, D, E)$

And the FDs

$A \rightarrow C$
$B \rightarrow D$

We want to find the keys.

Another approach to finding a key

Clearly, $\{A, B, C, D, E\}$ is a superkey since it contains all of the attributes.

But which attributes do **not** appear on the right-hand side of any FD?

$\{A, B, E\}$

The closure of $\{A, B, E\}$ is

1. $\{A, B, E\}^+ = \{A, B, E\}$
2. $\{A, B, E\}^+ = \{A, B, C, E\}$
3. $\{A, B, E\}^+ = \{A, B, C, D, E\}$

The closure includes all of the attributes hence it is a key.

Is it minimal? Try removing attributes from the key and see if it is still a key.

The Employee table

We can apply these processes to the Employee table. However, we saw that there are a lot of functional dependencies on that table. Indeed, one online tool calculated that there are 175 non-trivial functional dependencies.

The closure of the attributes also requires a very large number of operations. There are eight columns and therefore we would need to process the set of all subsets (the power set) of the attributes. This is a large number.

This process can be fully automated, however. But, since we are considering how to do these 'by hand' for now we can select some FDs from the domain.

The Employee table

empid	name	job	salary	project	manager	prlength	prcost
1	Jones	engineer	35000	Gherkin	Davis	5	500m
2	Wilson	sales	28000	Tunnel	Fulton	6	250m
1	Jones	engineer	35000	Tunnel	Fulton	6	250m
3	Peters	tech	24000	Gherkin	Davis	5	500m
3	Peters	tech	24000	Dome	Mandelson	2	2000m
4	Price	runner	17500	Dome	Mandelson	2	2000m
5	Dollis	designer	45000	Airport	Craig	5	1000m

Employee

The Employee table

We could select the following since they seem fairly genuine.

$project \rightarrow manager$
$job \rightarrow salary$
$empid \rightarrow name$
$project \rightarrow prlength, prcost$
$prlength \rightarrow prcost$

The Employee table

$project \rightarrow manager$
$job \rightarrow salary$
$empid \rightarrow name$
$project \rightarrow prlength, prcost$
$prlength \rightarrow prcost$

$\{project\}^+ = \{project, manager, prlength, prcost\}$
$\{job\}^+ = \{job, salary\}$
$\{empid\}^+ = \{empid, name\}$
$\{prlength\}^+ = \{prlength, prcost\}$
$\{project, job\}^+ = \{job, salary, project, manager, prlength, prcost\}$
$\{project, empid\}^+ =$
$\{empid, name, project, manager, prlength, prcost\}$
$\{project, empid, job\}^+ =$
$\{empid, name, job, salary, project, manager, prlength, prcost\}$

Keys

There are many superkeys in the table. However, a minimal superkey, i.e. candidate key was found on the previous slide. It is

$\{empid, job, project\}$

Normalisation

In the case of the Employee table, this process simply reveals the flaws in the table. It is possible to find a primary key for the table, as just demonstrated, but that doesn't mean that it is a good table. It is always possible to find a primary key for a table. In the worst case, one simply selects the entire set of attributes. Since no two rows can be the same in the relational model, this is guaranteed to work.

Next we will consider the process of normalisation to improve this table and to remove the anomalies we discussed earlier.