Week 3

Theory

Functional dependencies

- A constraint.
- "If two tuples of R agree on all attributes $A_1, A_2, \dots, A_n$ then they must also agree on all of attributes $B_1, B_2, \dots, B_n$"
  - $A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_n$
  - Like … functionally determines …
  - Could also be written by splitting rhs.
    - Splitting and combining only on rhs.
- If every instance of R will be one which a FD is true: R satisfies this FD..
- FD's $A \rightarrow B, B \rightarrow C$
  - Can then deduce $A \rightarrow C$
  - 2 FD's are equivalent if the satisfying set of relation instances is the same
- Trivial vs. non trivial FD's
  - FD is trivial if it holds for every instance of the relation
  - Trivial FD has a rhs that is a subset of the lhs.
    - Eg. $title\ year \rightarrow title$
  - Trivial dependency rule:
    - A non trivial FD, but which could be simplified by removing attributes appearing on both sides

Keys vs super keys

- We say a set of one or more attributes $\{A_1, A_2, \dots, A_n\}$ is a *key* for a relation R if:

  1. Those attributes functionally determine all other attributes of the relation. That is, it is impossible for two distinct tuples of R to agree on all of $A_1, A_2, \dots, A_n$.
  2. No proper subset of $\{A_1, A_2, \dots, A_n\}$ functionally determines all other attributes of R; i.e., a key must be *minimal.*
- Super key:
  - superset of a key
  - Meaning key not satisfying minimality

Computing the closure:

- Given set of attributes, and set of FD's.
- If necessary split FD's we only have FD's with single attribute on RHS.
- Initialize the closure to the given attributes.
- Add new attributes to set which follows from the current set of attributes.

Basis

- Any set of FD's equivalent to some other set of FD's is a basis for S
- Minimal basis B if:
    - All FD's in B have singleton rhs
    - Removing any FD, result is no longer a basis
    - Removing one/more attributes from lhs of any FD in B, result is no longer a basis.

Projecting FD's

- When we project on some relation, which FD's are kept?
- Given FD's S
    - All FD's following from S and
    - Involve only attributes of projected attrs.

INPUT: A relation R and a second relation R_1 computed by the projection $R_1 = \pi_L(R)$. Also, a set of FD's S that hold in R.

OUTPUT: The set of FD's that hold in R_1.

METHOD:

1. Let T be the eventual output set of FD's. Initially, T is empty.
2. For each set of attributes X that is a subset of the attributes of R_1, compute X^+. This computation is performed with respect to the set of FD's S, and may involve attributes that are in the schema of R but not R_1. Add to T all nontrivial FD's $X \rightarrow A$ such that A is both in X^+ and an attribute of R_1.
3. Now, T is a basis for the FD's that hold in R_1, but may not be a minimal basis. We may construct a minimal basis by modifying T as follows:
    (a) If there is an FD F in T that follows from the other FD's in T, remove F from T.
    (b) Let $Y \rightarrow B$ be an FD in T, with at least two attributes in Y, and let Z be Y with one of its attributes removed. If $Z \rightarrow B$ follows from the FD's in T (including $Y \rightarrow B$), then replace $Y \rightarrow B$ by $Z \rightarrow B$.
    (c) Repeat the above steps in all possible ways until no more changes to T can be made.

□

Note:

- Closing the empty set and the set of all attributes cannot yield a nontrivial FD.
- If we already know that the closure of some set X is all attributes, then we cannot discover any new FD's by closing supersets of X.

Design problems (anomalies):

- Happens when we try to cram too much into one relation
- Redundancy:
  - Info may be repeated several times in different tuples.
- Update anomaly:
  - Have to update the same information in many tuples
- Deletion Anomaly:
  - Loosing other information, when deleting a tuple

Decomposing relations:

- Splitting attributes to make multiple relations
- Solves anomalies

Boyce-Codd Normal Form

- Guarentee that none of the above anomalies are present

  A relation R is in BCNF if and only if: whenever there is a nontrivial FD $A_1A_2\cdots A_n \rightarrow B_1B_2\cdots B_m$ for R, it is the case that $\{A_1, A_2, \ldots, A_n\}$ is a superkey for R.
-
  - Meaning all lhs of every nontrivial FD is a superkey
- Decomposing into BCNF
  - Want all subsets to be in BCNF
  - Data in original relation represented faithfully by new relations.

INPUT: A relation R_0 with a set of functional dependencies S_0.

OUTPUT: A decomposition of R_0 into a collection of relations, all of which are in BCNF.

METHOD: The following steps can be applied recursively to any relation R and set of FD's S. Initially, apply them with $R = R_0$ and $S = S_0$.

1. Check whether R is in BCNF. If so, nothing more needs to be done. Return $\{R\}$ as the answer.
2. If there are BCNF violations, let one be $X \rightarrow Y$. Use Algorithm 7 to compute X^+. Choose $R_1 = X^+$ as one relation schema and let R_2 have attributes X and those attributes of R that are not in X^+.
3. Use Algorithm 12 to compute the sets of FD's for R_1 and R_2; let these be S_1 and S_2, respectively.
4. Recursively decompose R_1 and R_2 using this algorithm. Return the union of the results of these decompositions.

- Problems with decomposition
  - Recoverability of information. Can we recover original relation from decomposition?
  - Preservation of dependencies. Joining decomposition, do we have the original FD's?
- Decomposition has a lossless join:
  - We can get the original relation back by joining (natural join)
  - • If we decompose a relation according to Algorithm 20, then the original relation can be recovered exactly by the natural join.
- Chase test for a lossless join:
  - Make tableu like:

    Example 22: Suppose we have relation $R(A, B, C, D)$, which we have decomposed into relations with sets of attributes $S_1 = \{A, D\}$, $S_2 = \{A, C\}$, and $S_3 = \{B, C, D\}$. Then the tableau for this decomposition is shown in Fig. 9.

A	B	C	D
a	b_1	c_1	d
a	b_2	c	d_2
a_3	b	c	d

  - Apply FD's to try to get a row with all letters without subscripts → lossless join
- Sometimes we get both lossless join and dependency preservation when composing into BCNF relations.
  - Solution is to use 3NF

3NF

- Third normal form

- Relaxing BCNF requirement slightly
- Gets both lossless join and dependency preservation properties

A relation R is in *third normal form* (3NF) if:

- Whenever $A_1 A_2 \cdots A_n \to B_1 B_2 \cdots B_m$ is a nontrivial FD, either

$$\{A_1, A_2, \ldots, A_n\}$$

is a superkey, or those of $B_1, B_2, \ldots, B_m$ that are not among the A's, are each a member of some key (not necessarily the same key).

-
    - Attr member of some key aka. is prime.
    - Meaning either left side is superkey or right side consist of prime attributes only.

Algorithm 26: Synthesis of Third-Normal-Form Relations With a Lossless Join and Dependency Preservation.

INPUT: A relation R and a set F of functional dependencies that hold for R.

OUTPUT: A decomposition of R into a collection of relations, each of which is in 3NF. The decomposition has the lossless-join and dependency-preservation properties.

METHOD: Perform the following steps:

1. Find a minimal basis for F, say G.
2. For each functional dependency $X \to A$ in G, use XA as the schema of one of the relations in the decomposition.
3. If none of the relation schemas from Step 2 is a superkey for R, add another relation whose schema is a key for R.

- Checking for minimal basis
    - Checking that no two FD imply the third
        - Eg. Take closure of $\{A, B\}$ (lhs of FD1) using only FD2 and FD3
            - If closure not includes rhs of FD2, this is not implied by others.
    - Check we cannot eliminate attributes from lhs.
        - Remove attribute from a lhs.
        - Check if this new rule is implied by old set of FD's
            - Taking the closure of lhs of new rule
            - If not we cannot remove it.

1.1

Exercise 1.1 : Consider a relation about people in the United States, including their name, Social Security number, street address, city, state, ZIP code, area code, and phone number (7 digits). What FD's would you expect to hold? What are the keys for the relation? To answer this question, you need to know something about the way these numbers are assigned. For instance, can an area code straddle two states? Can a ZIP code straddle two area codes? Can two people have the same Social Security number? Can they have the same address or phone number?

$$R(name, SSN, street, city, state, zip, area\ code, phone)$$

Zip: city based

Area code: for multiple cities

State: multiple cities.

FD's:

- $SSN \rightarrow name, street, city, zip, area, state$
- $zip \rightarrow city$
- $area\ code \rightarrow state$
- $phone \rightarrow name$

Keys:

- $ssn, phone$

2.1

Exercise 2.1 : Consider a relation with schema $R(A, B, C, D)$ and FD's $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$.

a) What are all the nontrivial FD's that follow from the given FD's? You should restrict yourself to FD's with single attributes on the right side.

b) What are all the keys of R?

c) What are all the superkeys for R that are not keys?

a. …

$$AB \rightarrow D$$

$$C \rightarrow A$$

b. …

$$\{AB, BC, BD\}$$

c. …

$$\{ABC, ABD, BCD, ABCD\}$$

2.4

! Exercise 2.4: Show that each of the following are *not* valid rules about FD's by giving example relations that satisfy the given FD's (following the "if") but not the FD that allegedly follows (after the "then").

a) If $A \rightarrow B$ then $B \rightarrow A$.

b) If $AB \rightarrow C$ and $A \rightarrow C$, then $B \rightarrow C$.

c) If $AB \rightarrow C$, then $A \rightarrow C$ or $B \rightarrow C$.

a. …

$$R(SSN, ADDRESS)$$

$$SSN \rightarrow ADDRESS$$

But not the other way around.

b. …

$$SSN, Name \rightarrow Birthday$$

But:

$$name \rightarrow Birthday$$

Does not hold.

c. …

$$R(title, year, genre)$$

$$title, year \rightarrow genre$$

But neither $title \rightarrow genre$ og $year \rightarrow genre$ holds.

3.1

Exercise 3.1: For each of the following relation schemas and sets of FD's:

a) $R(A, B, C, D)$ with FD's $AB \to C$, $C \to D$, and $D \to A$.

b) $R(A, B, C, D)$ with FD's $B \to C$ and $B \to D$.

c) $R(A, B, C, D)$ with FD's $AB \to C$, $BC \to D$, $CD \to A$, and $AD \to B$.

d) $R(A, B, C, D)$ with FD's $A \to B$, $B \to C$, $C \to D$, and $D \to A$.

e) $R(A, B, C, D, E)$ with FD's $AB \to C$, $DE \to C$, and $B \to D$.

f) $R(A, B, C, D, E)$ with FD's $AB \to C$, $C \to D$, $D \to B$, and $D \to E$.

do the following:

$i)$ Indicate all the BCNF violations. Do not forget to consider FD's that are not in the given set, but follow from them. However, it is not necessary to give violations that have more than one attribute on the right side.

$ii)$ Decompose the relations, as necessary, into collections of relations that are in BCNF.

Relation	**FD's**	**Keys**	**Violations**	**Decomposition**
$R(A,B,C,D)$	$AB \to C$ $C \to D$ $D \to A$	AB BC DB	$C \to D$ $D \to A$ $C \to A$	Splitting on $C \to D$. Closure is $\{C, D, A\}$ $R1(C, D, A)$ $R2(C, B)$
$R1(C, D, A)$	$C \to D$ $D \to A$	C	$D \to A$	Splitting on $D \to A$ Closure is $\{D, A\}$ $R11(D, A)$ $R12(D, C)$
$R2(C, B)$	None	CB	None	Done
$R11(D, A)$	$D \to A$	D	None	Done
$R12(D, C)$	$C \to D$	C	None	Done

Meaning we are ending up with:

$$R11(D, A)$$

$$R12(D, C)$$

$$R2(C, B)$$

Relation	**FD's**	**Keys**	**Violations**	**Decomposition**
$R(A,B,C,D)$	$B \rightarrow C$ $B \rightarrow D$	AB	$B \rightarrow C$ $B \rightarrow D$	Splitting on $B \rightarrow C$ Closure is $\{B,C,D\}$ $R1(B,C,D)$ $R2(B,A)$
$R1(B,C,D)$	$B \rightarrow C$ $B \rightarrow D$	B	None	Done
$R2(B,A)$	None	AB	None	Done

Meaning we are ending up with:

$$R1(B,C,D)$$

$$R2(B,A)$$

Relation	**FD's**	**Keys**	**Violations**	**Decomposition**
$R(A,B,C,D)$	...	All pairs	None	Done

Relation	**FD's**	**Keys**	**Violations**	**Decomposition**
$R(A,B,C,D)$	$\ldots$	All singletons	None	Done

Relation	**FD's**	**Keys**	**Violations**	**Decomposition**
$R(A,B,C,D,E)$	$AB \rightarrow C$ $DE \rightarrow C$ $B \rightarrow D$	ABE	$AB \rightarrow C$ $DE \rightarrow C$ $B \rightarrow D$	Splitting on $AB \rightarrow C$ Closure is $\{A,B,C,D\}$ $R1(A,B,C,D)$ $R2(A,B,E)$
$R1(A,B,C,D)$	$AB \rightarrow C$ $B \rightarrow D$	AB	$B \rightarrow D$	Splitting on $B \rightarrow D$ Closure is $\{B,D\}$ $R11(B,D)$ $R12(B,A,C)$
$R2(A,B,E)$	None	ABE	None	Done
$R11(B,D)$	$B \rightarrow D$	B	None	Done
$R12(B,A,C)$	$AB \rightarrow C$	AB	None	Done

Meaning we are ending up with:

$$R2(A,B,E)$$

$$R11(B,D)$$

$$R12(B,A,C)$$

Relation	**FD's**	**Keys**	**Violations**	**Decomposition**
$R(A,B,C,D,E)$	$AB \rightarrow C$ $C \rightarrow D$ $D \rightarrow B$ $D \rightarrow E$	AB AC AD	$C \rightarrow D$ $C \rightarrow E$ $D \rightarrow B$ $D \rightarrow E$	Splitting on $C \rightarrow D$ Closure is $\{C,D,E,B\}$ $R1(B,C,D,E)$ $R2(C,A)$
$R1(B,C,D,E)$	$C \rightarrow D$ $D \rightarrow B$ $D \rightarrow E$	C	$D \rightarrow B$ $D \rightarrow E$	Splitting on $D \rightarrow B$ Closure is $\{D,B,E\}$ $R11(D,B,E)$ $R12(D,C)$
$R2(C,A)$	None	CA	None	Done
$R11(D,B,E)$	$D \rightarrow B$ $D \rightarrow E$	D	None	Done
$R12(D,C)$	$C \rightarrow D$	C	None	Done

Meaning we are ending up with:

$$R2(C,A)$$

$$R11(D,B,E)$$

$$R12(D,C)$$

3.2

Exercise 3.2: We mentioned in Section 3.4 that we would exercise our option to expand the right side of an FD that is a BCNF violation if possible. Consider a relation R whose schema is the set of attributes $\{A, B, C, D\}$ with FD's $A \to B$ and $A \to C$. Either is a BCNF violation, because the only key for R is $\{A, D\}$. Suppose we begin by decomposing R according to $A \to B$. Do we ultimately get the same result as if we first expand the BCNF violation to $A \to BC$? Why or why not?

$A \to B$ gives us the relations:

$$R1(A, B)$$

$$R2(A, C, D)$$

Need to decompose R2 once more:

$$R21(A, C)$$

$$R22(A, D)$$

$A \to BC$ gives us the relations:

$$R1(A, B, C)$$

$$R2(A, D)$$

We are not ending up with the same relations.

4.1

Exercise 4.1: Let $R(A,B,C,D,E)$ be decomposed into relations with the following three sets of attributes: $\{A,B,C\}$, $\{B,C,D\}$, and $\{A,C,E\}$. For each of the following sets of FD's, use the chase test to tell whether the decomposition of R is lossless. For those that are not lossless, give an example of an instance of R that returns more than R when projected onto the decomposed relations and rejoined.

a) $B \to E$ and $CE \to A$.

b) $AC \to E$ and $BC \to D$.

c) $A \to D$, $D \to E$, and $B \to D$.

d) $A \to D$, $CD \to E$, and $E \to D$.

a. …

A	**B**	**C**	**D**	**E**
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c	d_3	e

Using $B \to E$

A	**B**	**C**	**D**	**E**
a	b	c	d_1	e_1
a_2	b	c	d	e_1
a	b_3	c	d_3	e

Using $CE \to A$

A	**B**	**C**	**D**	**E**
a	b	c	d_1	e_1
a	b	c	d	e_1
a	b_3	c	d_3	e

Not lossless

b. …

A	**B**	**C**	**D**	**E**
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c	d_3	e

Using $AC \rightarrow E$

A	B	C	D	E
a	b	c	d_1	e
a_2	b	c	d	e_2
a	b_3	c	d_3	e

Using $BC \rightarrow D$

A	B	C	D	E
a	b	c	d	e
a_2	b	c	d	e_2
a	b_3	c	d_3	e

lossless

c. …

A	B	C	D	E
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c	d_3	e

Using $A \rightarrow D$

A	B	C	D	E
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c	d_1	e

Using $D \rightarrow E$

A	B	C	D	E
a	b	c	d_1	e
a_2	b	c	d	e_2
a	b_3	c	d_1	e

Using $B \rightarrow D$

A	B	C	D	E
a	b	c	d	e
a_2	b	c	d	e_2
a	b_3	c	d_1	e

Lossless

d. …

A	B	C	D	E
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c	d_3	e

Using $A \rightarrow D$

A	B	C	D	E
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c	d_1	e

Using $CD \rightarrow E$

A	B	C	D	E
a	b	c	d_1	e
a_2	b	c	d	e_2
a	b_3	c	d_1	e

Using $E \rightarrow D$

A	B	C	D	E
a	b	c	d_1	e
a_2	b	c	d	e_2
a	b_3	c	d_1	e

Not lossless

5.1

Exercise 5.1 : For each of the relation schemas and sets of FD's of Exercise 3.1:

i) Indicate all the 3NF violations.

ii) Decompose the relations, as necessary, into collections of relations that are in 3NF.

a) $R(A, B, C, D)$ with FD's $AB \to C$, $C \to D$, and $D \to A$.

b) $R(A, B, C, D)$ with FD's $B \to C$ and $B \to D$.

c) $R(A, B, C, D)$ with FD's $AB \to C$, $BC \to D$, $CD \to A$, and $AD \to B$.

d) $R(A, B, C, D)$ with FD's $A \to B$, $B \to C$, $C \to D$, and $D \to A$.

e) $R(A, B, C, D, E)$ with FD's $AB \to C$, $DE \to C$, and $B \to D$.

f) $R(A, B, C, D, E)$ with FD's $AB \to C$, $C \to D$, $D \to B$, and $D \to E$.

	FD's	**Keys**	**Violations**	**Minimal Basis**	**Decomposition**
A	$AB \to C$ $C \to D$ $D \to A$	AB BC DB	None	Done	Done
B	$B \to C$ $B \to D$	AB	$B \to C$ $B \to D$	$B \to C$ $B \to D$	$S_1(B, C)$ $S_2(B, D)$ $S_3(A, B)$
C	…	All pairs	None	Done	Done
D	…	All singletons	None	Done	Done
E	$AB \to C$ $DE \to C$ $B \to D$	ABE	$AB \to C$ $DE \to C$ $B \to D$	$AB \to C$ $DE \to C$ $B \to D$	$S_1(A, B, C)$ $S_2(D, E, C)$ $S_3(B, D)$ $S_4(A, B, E)$
F	$AB \to C$ $C \to D$ $D \to B$ $D \to E$	AB AC AD	$C \to E$ $D \to E$	$AB \to C$ $C \to D$ $D \to B$ $D \to E$	$S_1(A, B, C)$ $S_2(C, D)$ $S_3(D, B)$ $S_4(D, E)$

5.2

Exercise 5.2: Consider the relation `Courses` (C, T, H, R, S, G), whose attributes may be thought of informally as course, teacher, hour, room, student, and grade. Let the set of FD's for `Courses` be $C \to T$, $HR \to C$, $HT \to R$, $HS \to R$, and $CS \to G$. Intuitively, the first says that a course has a unique teacher, and the second says that only one course can meet in a given room at a given hour. The third says that a teacher can be in only one room at a given hour, and the fourth says the same about students. The last says that students get only one grade in a course.

a) What are all the keys for `Courses`?

b) Verify that the given FD's are their own minimal basis.

c) Use the 3NF synthesis algorithm to find a lossless-join, dependency-preserving decomposition of R into 3NF relations. Are any of the relations not in BCNF?

a. …

$$HS$$

b. …

Can we remove a rule?

- Compute closure of lhs without using rule

Can remove an attribute from a lhs?

- Compute closure of new lhs (with removed attr)

c. …

$$S_1(C,T)$$
$$S_2(H,R,C)$$
$$S_3(H,T,R)$$
$$S_4(H,S,R)$$
$$S_5(C,S,G)$$

All in all:

Keys	FD's	Minimal Basis	Decomposition
$\{HS\}$	$C \to T$ $HR \to C$ $HT \to R$ $HS \to R$ $CS \to G$	Same	$S_1(C,T)$ $S_2(H,R,C)$ $S_3(H,T,R)$ $S_4(H,S,R)$ $S_5(C,S,G)$