CS2102 Assignment 2: FD & NF

Deadline: End of Wk 13

1. **[0.5 Mark] Prime Attributes**

   To start your assignment 2, let's begin with something simple. Given a schema `R(A, B, C, D, E)` find the prime attributes given the set of functional dependencies below:

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

   Select all the prime attributes of `R` with respect to the set of functional dependencies above.

   Solution: Prime attributes are {A,C,D}.

2. **[1 Mark] Inference**

   You started a work at an IT company COM2. They have a database DB with a table `R(A, B, C, D, E)`. Unfortunately, the way the database created was lost. What you know is only that the **key** of `R` are: {A,B}, {B,C}, {C,D}, {D,E}, and {A,E}.

   There are a few set of functional dependencies that satisfies this conditions. You know that the following functional dependencies are valid in the database:

   1. $\{A,B\} \rightarrow \{C\}$
   2. $\{A,B\} \rightarrow \{D\}$
   3. $\{A,B\} \rightarrow \{E\}$
   4. $\{B,C\} \rightarrow \{D\}$
   5. $\{B,C\} \rightarrow \{E\}$
   6. $\{B,C\} \rightarrow \{A\}$
   7. $\{C,D\} \rightarrow \{E\}$
   8. $\{C,D\} \rightarrow \{A\}$
   9. $\{C,D\} \rightarrow \{B\}$
   10. $\{D,E\} \rightarrow \{A\}$
   11. $\{D,E\} \rightarrow \{B\}$
   12. $\{D,E\} \rightarrow \{C\}$
   13. $\{A,E\} \rightarrow \{B\}$
   14. $\{A,E\} \rightarrow \{C\}$
   15. $\{A,E\} \rightarrow \{D\}$

   Select **exactly five (5)** functional dependencies that will form a set of functional dependencies such that the schema `R` has exactly the **key** above. Note that there can be more than one possible answers. Obviously, if you choose all, it will definitely be correct by definition of **key**, but you can only choose exactly five.

   Solution: Should be any of the following combinations:

1. $\{A,B\} \rightarrow \{C\}$	1. $\{A,B\} \rightarrow \{E\}$
2. $\{B,C\} \rightarrow \{D\}$	2. $\{B,C\} \rightarrow \{A\}$
3. $\{C,D\} \rightarrow \{E\}$	3. $\{C,D\} \rightarrow \{B\}$
4. $\{D,E\} \rightarrow \{A\}$	4. $\{D,E\} \rightarrow \{C\}$
5. $\{A,E\} \rightarrow \{B\}$	5. $\{A,E\} \rightarrow \{D\}$

   But the actual check is done using an algorithm so as long as your combination produces exactly all the keys (nothing more, nothing less), then it will be accepted.

3. **[1 Mark] Finding Functional Dependencies**

   This question relates the concept of functional dependencies with ER diagram and database schema. In this question, you may ignore the `NULL` value from your analysis.

   (a) Consider the following ER diagram:

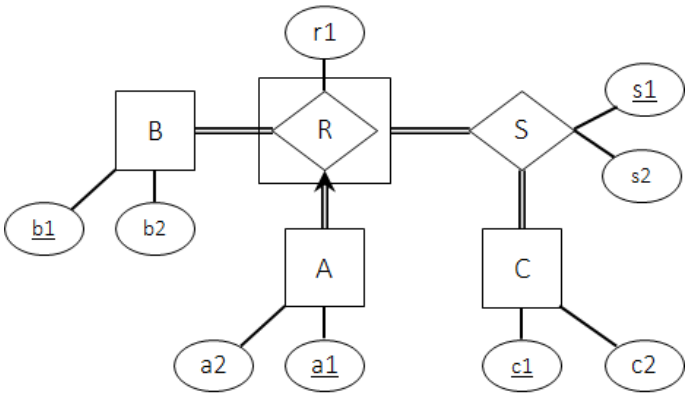

Figure 1: ER Diagram

We know that the attribute `b1` uniquely identifies the attribute `b2`. As such, we can write the following functional dependency: {`b1`} → {`b2`}.

Remember that by definition of functional dependency $A \rightarrow B$, we only care when the value of the attributes A are the same and when the values actually exists in the table. As such, given the following ER diagram:

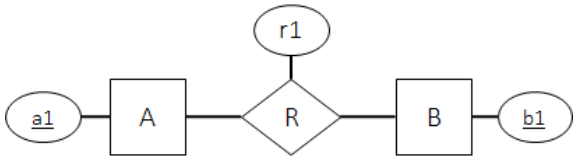

Figure 2: Relationship Set

the following functional dependency is valid: {`a1`,`b1`} → {`r1`}.

First, find all the functional dependencies that are valid based on the ER diagram. Your task is to find a **key** of the entire ER diagram. In other words, find a set of attributes A such that:

$$A \rightarrow \{\texttt{a1},\texttt{a2},\texttt{b1},\texttt{b2},\texttt{c1},\texttt{c2},\texttt{r1},\texttt{s1},\texttt{s2}\}$$

Remember that for the set of attributes A to be a key, there must not be another set of attributes B where $B \subset A$ and B is a superkey.

Select **ALL** the attributes that makes up A.

Solution: $A = \{\texttt{a1,s1,c1}\}$.

Due to the key constraint from A to R, we actually have $\{\texttt{a1}\} \rightarrow \{\texttt{r1}\}$.

(b) Consider the following relational schema which has no connection with the previous question:

```
CREATE TABLE A (
  a1  INTEGER  PRIMARY KEY,
  a2  INTEGER  NOT NULL
);
CREATE TABLE B (
  b1  INTEGER  PRIMARY KEY,
  b2  INTEGER  NOT NULL,
  a1  INTEGER  NOT NULL  REFERENCES A
);
CREATE TABLE C (
  c1  INTEGER  PRIMARY KEY,
  c2  INTEGER  UNIQUE  NOT NULL,
  b1  INTEGER  NOT NULL  REFERENCES B,
);
CREATE TABLE D (
  c1  INTEGER  PRIMARY KEY  REFERENCES C,
  d2  INTEGER  NOT NULL,
);
```

Your task is to find a **key** of the entire database schema. In other words, find a set of attributes A such that:

$$A \rightarrow \{\texttt{a1,a2,b1,b2,c1,c2,}\cancel{\texttt{d1}}\texttt{,d2}\}$$

Remember that for the set of attributes A to be a key, there must not be another set of attributes B where $B \subset A$ and B is a superkey.

Select **ALL** the attributes that makes up A.

Solution: $A = \{\texttt{c1}\}$.

Note that because `b1` is a primary key of `B`, we have $\{\texttt{b1}\} \rightarrow \{\texttt{a1}\}$. Since we also have $\{\texttt{a1}\} \rightarrow \{\texttt{a2}\}$, by transitivity we have $\{\texttt{b1}\} \rightarrow \{\texttt{a2}\}$. Similar reasoning can be made for $\{\texttt{c1}\} \rightarrow \{\texttt{a2}\}$.

Here, we nicely have `c1` in both table `C` and table `D` to be exactly the same attribute. If, for instance, we have the following definition for table `D`:

```
CREATE TABLE D (
  d1  INTEGER  PRIMARY KEY  REFERENCES C,
  d2  INTEGER  NOT NULL,
);
```

Then we **may** have $\{\texttt{c1}\} \rightarrow \{\texttt{d1}\}$ and $\{\texttt{d1}\} \rightarrow \{\texttt{c1}\}$ and we have two possible solutions here. In fact, this comes from an ISA hierarchy.

4. **[2.5 Marks] Normal Forms**

We have learnt two normal forms in Lecture[1]. We will try to use the concepts learnt.

For this question, we will use the following schema `R(A,B,C,D,E)` with the following set of functional dependencies:

$$\{\{\texttt{A,B,D}\} \rightarrow \{\texttt{E}\}, \{\texttt{A,C,E}\} \rightarrow \{\texttt{A,D}\}, \{\texttt{B,D}\} \rightarrow \{\texttt{E}\}, \{\texttt{C,D}\} \rightarrow \{\texttt{B,E}\}, \{\texttt{C,E}\} \rightarrow \{\texttt{B,D}\}\}$$

(a) Find all the keys of `R`. You do not have to show your step. Write the answer in the following format:

$$\{\texttt{A,B,C,D}\}, \{\texttt{B,C,D,E}\}, \{\texttt{C,D,E,A}\}$$

Solution: All the keys are

$$\{\texttt{A,C,E}\},\ \{\texttt{A,C,D}\}$$

(b) Find a lossless-join BCNF decomposition of `R`. You do not have to show your step. Write the answer in the following format:

$$\{\texttt{R1(A,B,C,D), R2(B,C,D,E), R3(C,D,E,A)}\}$$

Solution: One possible decomposition:

$$\{\texttt{R1(B,D,E), R2(B,C,D), R3(C,D,E), R4(A,C,E)}\}$$

This may be formed via the following steps:

- $\{\texttt{C,D,E}\} \rightarrow \{\texttt{B}\}$ violates BCNF of `R(A,B,C,D,E)`. We decompose this into `Ra(B,C,D,E)` and `Rb(A,C,D,E)`
  - $\{\texttt{B,D}\} \rightarrow \{\texttt{E}\}$ violates BCNF of `Ra(B,C,D,E)`. We decompose this into `R1(B,D,E)` and `R2(B,C,D)`
    - `R1(B,D,E)` is in BCNF.
    - `R2(B,C,D)` is in BCNF.
  - $\{\texttt{C,E}\} \rightarrow \{\texttt{D}\}$ violates BCNF of `Rb(A,C,D,E)`. We decompose this into `R3(C,D,E)` and `R4(A,C,E)`
    - `R3(C,D,E)` is in BCNF.
    - `R4(A,C,E)` is in BCNF.

[1] Well, one of them is still not completed as of the release of this assignment

(c) Find a minimal basis of the set of functional dependencies. You do not have to show your step. Write the answer in the following format:

```
{A,B,C} -> {C,D,E}, {B,C,D} -> {D,E,A}, {C,D,E} -> {E,A,B}
```

Solution: One possible minimal basis is:

$$\{\{\texttt{B,D}\} \to \{\texttt{E}\}, \{\texttt{C,D}\} \to \{\texttt{B}\}, \{\texttt{C,E}\} \to \{\texttt{D}\}\}$$

(d) Using the minimal basis you computed in part (c), find a lossless-join and dependency-preserving 3NF decomposition of `R`. Note that you have to first ensure that your answer to part (c) is a minimal basis first. Write the answer in the same format as part (b).

Solution: Using the previous minimal basis, we arrive at the following decomposition:

```
{R1(B,D,E), R2(B,C,D), R3(C,D,E)}
```

The keys are not in the decomposition, so we add the key `R4(A,C,E)`. And voila, we have exactly the same decomposition as our BCNF decomposition.

(e) Is there a lossless-join and dependency-preserving BCNF decomposition of `R`? If **yes**, write the lossless-join and dependency-preserving BCNF decomposition of `R` in the same format as (d). Otherwise, briefly explain the reason why you think that there is no such decomposition.

Solution: Clearly, there is due to the fact that our 3NF synthesis and BCNF decomposition producing the exact same decomposition. In fact, we have shown that 3NF synthesis algorithm may in fact produce a BCNF decomposition!