CS2102: Database Systems (AY2021-2022 – Sem 1)

Final Exam

Instructions

1. Please read **ALL** instructions carefully.
2. This assessment contains **16** questions (*and 1 bonus question*):
   (a) There are 10 Multiple Response Questions (MRQ): 1.1 - 1.5, 2.1 - 2.5
   (b) There is 1 Short Essay Question: 2.6
   (c) There is 1 Multiple Choice Question (MCQ): 2.7
   (d) There is 1 Hot Spot Question: 3.1 (*5 questions on Examplify*)
   (e) There are 3 Fill in the Blank Questions (FITB): 4.1, 5.1 - 5.2
3. All the assessment is be done using Examplify:
   (a) This is a secure assessment:
      i. Your Internet connection will be blocked.
      ii. You will not be able to access any other software besides Examplify.
   (b) This is a closed-book exam
4. Use the question number shown on Examplify when asking via Zoom chat.
   - If the answer is clear from the question pdf/Examplify, we will reply with "No Comment".
5. Failure to follow each of the instructions above may result in deduction of your marks.

Good Luck!

Bonus

You will only get this bonus mark if you have less than 40 marks.

0.1 Revision Time (1 mark). Consider the following topics:

1. Relational Algebra
2. Entity-Relationship
3. Armstrong's Axioms
4. Closure
5. Minimal Basis
6. BCNF
7. 3NF
8. Lossless Join
9. Dependency Preserving

Which of the above are **NOT** tested in this exam? Write the number on Examplify after reading all the questions below.

Solution: 3

NOTE: Only the *number* as per instruction.

1 Functional Dependencies

In this section, we consider the schema `R(A, B, C, D, E)` with the following set of functional dependencies:

$$\texttt{F} = \{\{\texttt{A}\} \to \{\texttt{B}\},\ \{\texttt{A,E}\} \to \{\texttt{D}\},\ \{\texttt{B}\} \to \{\texttt{E}\},\ \{\texttt{A,D}\} \to \{\texttt{E}\},\ \{\texttt{C,D}\} \to \{\texttt{A}\},\ \{\texttt{A,B}\} \to \{\texttt{D}\},\ \{\texttt{E}\} \to \{\texttt{B}\}\}$$

1.1 Closure (2 marks). Select **ALL** the attributes in the closure of $\{\texttt{A,E}\}$ (*i.e.*, $\{\texttt{A,E}\}^+$) with respect to `F`.

(A) `A` (B) `B` (C) `C` (D) `D` (E) `E`

Solution: A, B, D, E

- $\{\texttt{A}\} \to \{\texttt{B}\}$: Gives `B`.
- $\{\texttt{B}\} \to \{\texttt{E}\}$: Gives `E`.
- $\{\texttt{A,E}\} \to \{\texttt{D}\}$: Gives `D`.
- There are no $\{\texttt{C}\}$ on the *right-hand side*.

$\{\texttt{A,E}\}^+ = \{\texttt{A,B,D,E}\}$

1.2 Implied FD (2 marks). functional dependencies that are implied by F (*i.e.,* can be derived from F) from the choices below. If none of the choices can be implied, you should choose "None of the above".

(A) $\{A\} \rightarrow \{D\}$ (*this was* $\{A\} \rightarrow \{B\}$ in Examplify but it is still can be derived)

(B) $\{D,E\} \rightarrow \{A\}$

(C) $\{C,D\} \rightarrow \{E\}$

(D) $\{A,D\} \rightarrow \{C\}$

(E) None of the above

Solution: A, C

$\{A\}^+ = \{A,B,D,E\}$ $\therefore \{A\} \rightarrow \{D\}$ or $\{A\} \rightarrow \{B\}$

$\{D,E\}^+ = \{B,D,E\}$ $\therefore \{D,E\} \nrightarrow \{A\}$ since $A \notin \{D,E\}^+$

$\{C,D\}^+ = \{A,B,C,D,E\}$ $\therefore \{C,D\} \rightarrow \{E\}$

$\{A,D\}^+ = \{A,B,D,E\}$ $\therefore \{A,D\} \nrightarrow \{E\}$ since $C \notin \{A,D\}^+$

1.3 Superkeys (2 marks). Select **ALL** the superkeys of R with respect to F from the choices below. If none of the choices is a superkey, you should choose "None of the above".

(A) $\{A,B,C\}$

(B) $\{B,C,D\}$

(C) $\{C,D,E\}$

(D) $\{A,B,E\}$

(E) None of the above

Solution: A, B, C

$\{A,B,C\}^+ = \{A,B,C,D,E\}$ so it's a superkey

$\{B,C,D\}^+ = \{A,B,C,D,E\}$ so it's a superkey

$\{C,D,E\}^+ = \{A,B,C,D,E\}$ so it's a superkey

$\{A,B,E\}^+ = \{A,B,D,E\}$ so it's **NOT** a superkey

1.4 Prime Attributes (2 marks). Select **ALL** prime attributes of R with respect to F.

(A) A (B) B (C) C (D) D (E) E

Solution: A, C, D
Keys = {A,C}, {C,D}, so, prime attributes are {A, C, D}.

1.5 Functional Equivalence (2 marks). We say that two attributes X and Y are *functionally equivalent* if and only if we can show both: {X} → {Y} and {Y} → {X}. Select **ALL** attributes that are functionally equivalent to B with respect to F. For ***obvious*** reason, your answer should include B.

(A) A (B) B (C) C (D) D (E) E

Solution: B, E
$\{A\}^+ = \{A,B,D,E\}$
$\{B\}^+ = \{B,E\}$
$\{C\}^+ = \{C\}$
$\{D\}^+ = \{D\}$
$\{E\}^+ = \{B,E\}$
$\therefore \{B\} \rightarrow \{E\} \wedge \{E\} \rightarrow \{B\}$

NOTE: As per instruction, B should be selected as part of the answer. This is because if we let X = B and Y = B, we satisfy the condition {B} → {B} and {B} → {B} for which both are trivial.

2 Normal Forms

In this section, we consider the same schema as before `R(A, B, C, D, E)` with the same set of functional dependencies as before:

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

We further consider the following decomposition for `R`:

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

2.1 Properties (2 marks). Note that `R` is **NOT** in BCNF. Select **ALL** the functional dependencies below that violates the BCNF property of `R` with respect to `F`. In other words, the functional dependencies must also be implied by `F`. If none of the choices is a violation that is implied by `F`, you should choose "None of the above".

(A) $\{\texttt{A,E}\} \rightarrow \{\texttt{B}\}$

(B) $\{\texttt{A,B,D}\} \rightarrow \{\texttt{C}\}$

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

(D) $\{\texttt{A,C}\} \rightarrow \{\texttt{E}\}$

(E) None of the above

Solution: A, C
$\{\texttt{A,E}\}^+ = \{\texttt{A,B,D,E}\}$
$\{\texttt{A,B,D}\}^+ = \{\texttt{A,B,D,E}\}$ ($\{\texttt{A,B,D}\} \rightarrow \{\texttt{C}\}$ *is not an implied FD*)
$\{\texttt{A,D,E}\}^+ = \{\texttt{A,B,D,E}\}$
$\{\texttt{A,C}\}^+ = \{\texttt{A,B,C,D,E}\}$

2.2 BCNF (2 marks). Select **ALL** the decomposed schema that are in BCNF with respect to `F`. If none of the choices is in BCNF, you should choose "None of the above".

(A) `R1` (B) `R2` (C) `R3` (D) `R4` (E) None of the above

Solution: E
Violations

- `R1`: $\{\texttt{A}\} \rightarrow \{\texttt{B}\}$
- `R2`: $\{\texttt{A}\} \rightarrow \{\texttt{D}\}$
- `R3`: $\{\texttt{A}\} \rightarrow \{\texttt{E}\}$
- `R4`: $\{\texttt{B}\} \rightarrow \{\texttt{E}\}$

2.3 3NF (2 marks). Select **ALL** the decomposed schema that are **NOT** in 3NF with respect to F. If all of the choices are in 3NF, you should choose "None of the above".

(A) R1 (B) R2 (C) R3 (D) R4 (E) None of the above

Solution: A, C
Violations

- R1: $\{A\} \rightarrow \{B\}$
- R2: *no violations since all attributes are prime*
  - $\{A\}^+ = \{A,B,D,E\}$, so $\{A,C\}$ is a key of R2R2
  - $\{C,D\} \rightarrow \{A\}$, so $\{C,D\}$ is a key of R2 too
- R3: $\{A\} \rightarrow \{E\}$
- R4: *no violations since all attributes are prime*
  - Note that from Q1.5, we know that B is functionally equivalent to E
  - Since $\{B,D\}$ is a key, therefore $\{D,E\}$ is also a key

2.4 Lossless Join (2 marks). Select **ALL** the lossless join decomposition of R with respect to F. If none of the choices is a lossless join decomposition, you should choose "None of the above". Note that you do not have to take into account any other properties beside lossless join decomposition.

(A) { R1(A,B), R2(B,C,D), R3(A,E) }
(B) { R1(A,B,D,E), R2(A,C,E), R3(B,C) }
(C) { R1(A,B,C), R2(A,C,D,E) }
(D) { R1(A,B,E), R2(A,C,D) }
(E) None of the above

Solution: B, C, D

- A: *not lossless join*
  - Try Rt(A,B,E) = R1(A,B) $\bowtie$ R3(A,E) since $\{A\} \rightarrow \{B\}$. But R(A,B,C,D,E) = Rt(A,B,E) $\bowtie$ R2(B,C,D) is not lossless join because $\{B\}^+ = \{B,E\}$
  - On the other hand, it is clear that neither Rt(A,B,C,D) = R1(A,B) $\bowtie$ R2(B,C,D) nor Rt(A,B,C,D,E) = R2(B,C,D) $\bowtie$ R3(A,E) are lossless join

- B: *lossless join*
  - Rt(A,B,C,D,E) = R1(A,B,D,E) $\bowtie$ R2(A,C,E) is lossless join because $\{A,E\}^+ = \{A,B,D,E\}$
  - At this point, adding new schema will not make it lossy
- C: *lossless join by simple step*
- D: *lossless join by simple step*

2.5 Dependency Preserving (2 marks). Select **ALL** the dependency preserving decomposition of R with respect to F. If none of the choices is a dependency preserving decomposition, you should choose "None of the above". Note that you do not have to take into account any other properties beside dependency preserving decomposition.

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

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

(C) { R1(A,B,C), R2(A,C,D,E) }

(D) { R1(A,B,E), R2(A,C,D) }

(E) None of the above

Solution: D
Might be easier to compare with the *canonical cover*:
$\{\{A\} \to \{D,E\}, \{B\} \to \{E\}, \{C,D\} \to \{A\}, \{E\} \to \{B\}\}$
Write down the *canonical cover* of each projection:

- A: $\{\{A\} \to \{B\}\} \cup \{\{C,D\} \to \{B\}\} \cup \{\{A\} \to \{E\}\}$
- B: $\{\{A\} \to \{D,E\}, \{B\} \to \{E\}, \{E\} \to \{B\}\} \cup \{\{A\} \to \{E\}\} \cup \{\}$
- C: $\{\{A\} \to \{B\}\} \cup \{\{A\} \to \{D,E\}, \{C,D\} \to \{A\}\}$
- D: $\{\{A\} \to \{E\}, \{B\} \to \{E\}, \{E\} \to \{B\}\}$
  $\cup \{\{C,D\} \to \{A\}, \{A\} \to \{D\}\}$

The prove for equivalence is left as an exercise.

2.6 Optimal Decomposition (3 marks). Find a lossless join and dependency preserving BCNF decomposition of R with respect to F *if it exists*. Otherwise, find a lossless join and dependency preserving 3NF decomposition of R with respect to F.

Furthermore, your answer should not contain any **redundant schema**. We say that a schema Ri in a decomposition is redundant if and only if there is another schema Rj in the decomposition such that all attributes in Ri exists in Rj. For instance, R1(A,B,C) is redundant if there is R2(A,B,C,D).

Your answer should be in the same syntax as Assignment 2 (*relevant example reproduced below*). Any deviation from the syntax may be penalised.

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

Note that whitespaces are ignored and may be omitted for simplicity. However, brackets are important and the correct use of brackets as well as the correct type of brackets are expected.

Do **NOT** any working and/or other texts besides the answer in the correct format. Otherwise, additional penalty may be applied.

Your answer:

Solution: One *possible* answer is: `{ R1(A,B,D), R2(A,C,D), R3(B,E) }`

2.7 With Complete Information (1 mark). Is your answer in the previous question BCNF or 3NF? Note, you will get this mark even if your answer for the previous question is wrong as long as you correctly choose the appropriate answer. However, if you do not have any answer for the previous question, you will not get any mark for this one.

(A) `BCNF`

(B) `3NF`

Solution: B

NOTE: BCNF solution is **unfortunately** not possible and **none** of the BCNF answers given satisfies all conditions. Choosing 3NF when your previous answer actually satisfies BCNF *may* be penalised. However, if your previous answer satisfies BCNF and you answer BCNF here, you will get the mark, but you will likely not get lots of marks for the previous question.

3 Database Modeling with ER Diagrams

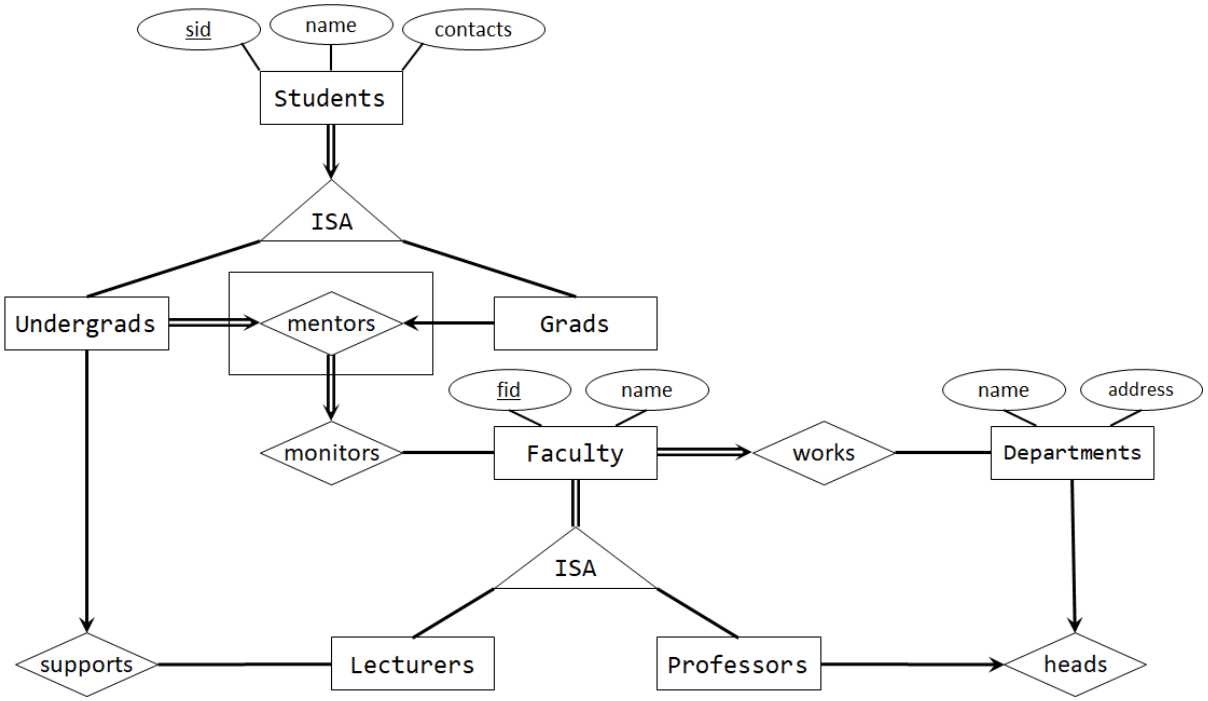
3.1 ER Diagram Errors (5 marks). The ER diagram shown below represents a simplified university database and contains **at most 5 errors**. An error may be due to the use of incorrect notations or due to incorrectly modeling the following constraints:

- Each students must be either an Undergrad or Grad student but not both; each student has a name, a set of contact numbers, and is uniquely identified by an id (*sid*).
- A faculty member may be a Lecturer or Professor; each faulty member has a name and is uniquely identified by an id (fid).
- Each faculty member works for exactly one department; each department has an address and is uniquely identified by its name.
- Each Undergrad student is mentored by a Grad student but some Grad student may decide not to mentor any Undergrad student. If they do mentor, they can only mentor one Undergrad.
- Each Undergrad-Grad mentorship is monitored by a faculty member.
- Each department must be headed by exactly one professor; a professor can only be a head to at most one department.
- An Undergrad student might receive additional support from a Lecturer.

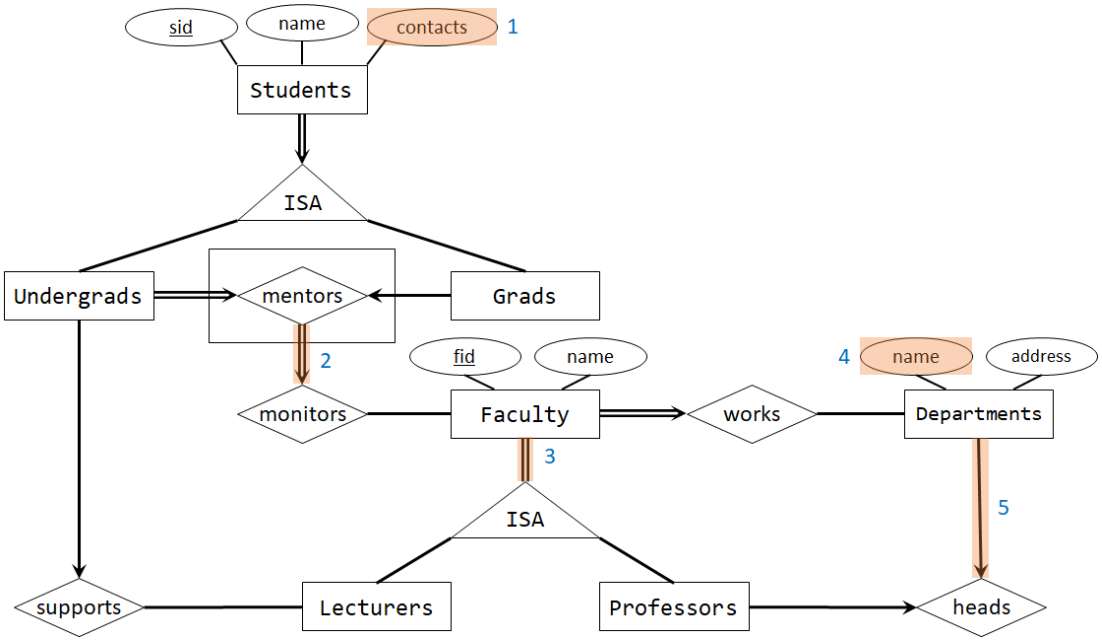
In the following 5 questions on Examplify, you will be presented with the following ER diagram (*on the next page*). In each question, place a pin over 1 of the 5 errors. Note the order in which you find and pin the 5 errors does not matter. If there are **fewer than 5 errors**, you should leave it *blank* instead of answering with duplicate errors (*i.e.,* the same error used before). You should place your pin according to the following rule:

- If the mistake is due to attribute, your pin should be at the center of the oval.
- If the mistake is due to entity set, your pin should be at the center of the rectangle.
- If the mistake is due to relationship set or aggregate, your pin should be at the center of the diamond.
- If the mistake is such that the ISA hierarchy is not supposed to be used, your pin should be at the center of the the triangle. However, if the mistake is due to the overlap and/or covering constraints, your pin should be on the line.
- If the mistake is due to the connection (*i.e., lines/arrows*), your pin should be on the line (if possible, outside of any rectangle unless the mistake is due to aggregate).

ER diagram:



Solution: ER diagram with orange boxes indicating the the problematic parts:



1. `contacts` should be a multi-valued attribute.
2. The arrow should not touch the diamond (`<mentors>`) but should end at the rectangle.
3. `Faculty` *may* be `Lecturers` or `Professors` but not necessary. In other words, they may also be *neither*.
4. `name` should be a key (*i.e.*, underlined).

5. `Departments` → `heads` should be `Departments` ⇒ `heads` since there is a total participation constraint as well in "A department *must* be headed by exactly one professor".

NOTE: Duplicate answer *may* be penalised. The actual bounding box *may* be smaller than what is shown especially for attributes.

4 Cardinalities

4.1 Cardinalities of RA operations (5 Marks). You are given 2 relations R and S, with m being the number of tuples in R (i.e., $|R| = m$) and n being the number of tuples in S (i.e., $|S| = n$). Assume that $m > n > 0$, R and S are union-compatible and both relations do not contain any *null* values.

Fill in the blank below with the minimum and maximum number of tuples in the resulting relation. Specifies the minimum and maximum number of tuples in the result relation – denoted as tuples [*minimum*, *maximum*] to mean that the result is *inclusive* of both *minimum* and *maximum*. There are no other assumption on the schema of R and S!

Please use the following expressions for your answer:

- Values: 0, 1, 2, ...
- Variables: m or n
- Operations:
  - Addition: $m + n$ or $n + m$
  - Subtraction: $m - n$ or $n - m$
  - Multiplication: mn or nm
  - Division: m/n or n/m

You should not use any other expressions for your answer.

(a) $R \cup S$: [☐ , ☐]

Solution: $[m, m + n]$

- **Minimum**: when $S \subset R$ (*since S is smaller*).
- **Maximum**: when $S \cap R = \emptyset$.

(b) $R \cap S$: [☐ , ☐]

Solution: $[0, n]$

- **Minimum**: when $S \cap R = \emptyset$.
- **Maximum**: when $S \subset R$ (*since S is smaller*).

(c) $R - S$: [☐ , ☐]

Solution: $[m - n, m]$

- **Minimum**: when $S \subset R$.
- **Maximum**: when $S \cap R = \emptyset$.

NOTE: Should this be $S - R$, the minimum should be 0 since we cannot have negative size.

(d) $R \bowtie S$: [☐ , ☐]

Solution: $[0, mn]$

- **Minimum**: consider a common attribute `A`, the minimum is when there are no tuples such that the value of attributes `A` are the same in both R and S.
- **Maximum**: there are many cases such as (1) no common attribute so it is equivalent to $\times$ or (2) all tuples contain the same value for the common attribute `A`.

(e) $R \rtimes S$: [☐ , ☐]

Solution: $[m, mn]$

- **Minimum**: when none of the tuples on S are dangling so all dangling tuples are given by R which will be smaller than $m + n$ when all are dangling
- **Maximum**: similar to natural join case (2)

5 Theory

In this section, your answer should follow the same syntax as Assignment 2 (*examples reproduced below*). Any deviation from the syntax may be penalised.

- Schema: `R(A, B, C, D, E)`
- Functional Dependency: `{A,B} -> {C,D}`
- Set of Schema: `{R1(A, B, C), R2(D, E, F)}`
  - The name of the schema is irrelevant but must be written.
- Set of Functional Dependency: `{ {A,B} -> {C,D}, {E,G} -> {H,I} }`

Note that whitespaces are ignored and may be omitted for simplicity. However, brackets and arrows are important and the correct use of brackets as well as the correct type of brackets are expected.

5.1 Keys and Equivalence (3 marks). Consider any schema `R` with an arbitrary set of functional dependencies `F1` and `F2`. Bob claimed that if the key of `R` with respect to `F1` is *the same as* the key of `R` with respect to `F2`, it means that `F1` is *equivalent* to `F2`.

Alice says that it is not. Using as few attributes as possible and as few functional dependencies as possible provide a counter-example to Bob's claim. In other words, provide the concrete example of `R`, `F1` and `F2` to counter Bob's claim.

Do **NOT** any working and/or other texts besides the answer in the correct format. Otherwise, additional penalty may be applied.

- Answer for `R`:
- `F1 =`
- `F2 =`

Solution: First, note that there is no answer with 2 attributes by *enumeration*. This is a similar "proof" as 2 attributes schema are in BCNF. The idea here is to create an fd with LHS as part of the key with the other attribute missing.
`R(A, B, C)`
`F1 = {{A} → {B}}`
`F2 = {{C} → {B}}`

NOTE: There are many possible answers, we will check for the following conditions.

- Key of `R` with respect to `F1` is the same as key of `R` with respect to `F2`.
- `F1` $\not\equiv$ `F2`.

The following penalties *may* apply with varying degrees.

- Invalid syntax (*since the syntax has been used in Assignment 2*).
- Using more than 3 attributes in `R`.
- Using more than 1 functional dependency on `F1` and/or `F2`.
- Using more than 1 attributes on each of the left-hand side and right-hand side of functional dependencies on `F1` and/or `F2`.

5.2 Minimal Basis (3 marks). Consider an arbitrary set of functional dependencies `F` such that `F` contains only non-trivial and decomposed functional dependencies. Alice claims that *any* minimal basis `Fb` of `F` must share **at least one** functional dependencies with `F`. In other words, the intersection of `F` and `Fb` must have at least one element (*i.e.,* $\texttt{F} \cap \texttt{Fb} \neq \emptyset$).

Now Bob says that it is, in fact, not. Using as few attributes as possible and as few functional dependencies as possible provide a counter-example to Alice's claim. In other words, provide the concrete example of `F` and `Fb` to counter Alice's claim.

Do **NOT** any working and/or other texts besides the answer in the correct format. Otherwise, additional penalty may be applied.

- `F` =
- `Fb` =

Solution: First, note that there is no answer with 2 attributes because we require functional equivalence. The trick here is to have 3 attributes forming a cycle. Then we simply have the cycle be read clockwise and counter-clockwise.

$\texttt{F} = \{\{\texttt{A}\} \rightarrow \{\texttt{B}\}, \{\texttt{B}\} \rightarrow \{\texttt{C}\}, \{\texttt{C}\} \rightarrow \{\texttt{A}\}\}$
$\texttt{Fb} = \{\{\texttt{A}\} \rightarrow \{\texttt{C}\}, \{\texttt{B}\} \rightarrow \{\texttt{A}\}, \{\texttt{C}\} \rightarrow \{\texttt{B}\}\}$

NOTE: Two functional dependencies are *equal* if and only if the **set of attributes** on both the left-hand side and right-hand side of $\rightarrow$ are the same. Since this is an equality on **set**, $\{\texttt{A,B}\} \rightarrow \{\texttt{C}\}$ is equal to $\{\texttt{B,A}\} \rightarrow \{\texttt{C}\}$.

Using $\emptyset$ on one or both of `F` and `Fb` will not receive full mark as we do not consider (1) functional dependencies with empty left-hand/right-hand side and (2) empty set of functional dependencies.

The following penalties *may* apply with varying degrees.

- Invalid syntax (*since the syntax has been used in Assignment 2*).
- Using more than 3 attribute names.
- Using more than 3 functional dependencies on `F1` and/or `F2`.
- Using more than 1 attributes on each of the left-hand side and right-hand side of functional dependencies on `F1` and/or `F2`.