Samir Khadka
Assignment_Normalization2
CS457(A) - Data Modeling and Implementation Techniques

1.
- {A, B}→{C} implies that **A** and **B** together determine **C**.
- {A}→{D, E} means that **A** alone determines both **D** and **E**.
- {B}→{F} indicates that **B** determines **F**.
- {F}→{G, H} tells us that **F** determines both **G** and **H**.
- {D}→{I, J} means that **D** determines **I** and **J**.

On the basis of these dependencies, we can say that the attributes A and B together serve as the key to the relation R. No other pair of attributes can define every other attribute uniquely.

Therefore, the key for **R** is **{A, B}**.

Checking if R is in 2NF:

- A relation is in **2NF** if:
  I. It is in **1NF**
  II. Every non-prime attribute is fully functionally dependent on the entire candidate key.
- In our case, the candidate key is **{A, B}**.
- Let's check if all non-prime attributes are fully functionally dependent on **{A, B}**:
  - **C** is fully functionally dependent on **{A, B}** (via **{A, B}**→**{C}**).
  - **D** and **E** are fully functionally dependent on **{A, B}** (via **{A}**→**{D, E}**).
  - **F** is fully functionally dependent on **{A, B}** (via **{B}**→**{F}**).
  - **G** and **H** are fully functionally dependent on **{A, B}** (via **{F}**→**{G, H}**).
  - **I** and **J** are fully functionally dependent on **{A, B}** (via **{D}**→**{I, J}**).

Relation **R** satisfies the **2NF** condition since all non-prime attributes are fully functionally dependent on the candidate key.

2.

Answer

The closure of a specific attribute set is the set of attributes that are functionally determined from that set. We consider the initial set of attributes which are on the left-hand side of the functional dependencies and not on the right-hand side and this is {A, B, D}. The closure of {A, B, D} is: {A, B, D, C, E, F, G, H, I, J}. This implies that {A, B, D} is a superkey.

A candidate key is a minimal key where removing any attribute will not make it more than a super key. We can obtain the candidate key by first removing attributes from the superkey and then verifying the closure set if it contains all the attributes.

We can discover the candidate keys to be the {A, B} and {A, D} after getting rid of B and D one at a time thereby checking the closure.

The prime attributes are components of any candidate key. Firstly, the most significant characteristics are A, B, and D.

Non-prime attributes are those that do not participate in any candidate key. For this example, the non-essential features are C, E, F, G, H, I, and J.

Relation is in 2NF when it is in 1NF and non-prime attribute is fully functionally dependent on each candidate key. Hence, there is no attribute in the relation which is functionally dependent on any candidate key except keys.

3. Answer

a. a)

i. A→B

It is not because of tuple 2 as A has 2 different values of B on 10.

ii. B→C

It holds the functional dependency.

iii. C→B

It does not hold the functional dependency due to tuple 3.

iv. B→A

It does not hold the functional dependency due to tuple 5 and 6.

v. $C \rightarrow A$

It does not hold the functional dependency due to tuples 3,5 and 6.

b. Yes, AB is a potential candidate key.

4. Answer

No, there is no such thing as a candidate key named AB among the list of attributes R (A, B, C, D, and E). A candidate key is a minimal set of attributes that unambiguously identify each tuple in a relation.To check whether ab is a candidate key of the given relation R(A, B, C, D, E), we need to check whether it satisfies the following conditions:

Each tuple in the relation must be unique, which means no two tuples can have the same values for all attributes of the candidate key.

No proper subset of the candidate key can uniquely identify each tuple in the relation.

Now, let's check whether ab satisfies these conditions:

We know that $AB \rightarrow C$, which means two tuples with the same values for ab must have the same value for C. Therefore, AB cannot uniquely identify each tuple in the relation.

We can see that {A, B} is not a proper subset of {A, B, C, D, E}, so we only need to check whether {A, B} is a superkey of the relation.

We will achieve this by determining if all the relation attributes can be expressed without A and B. We are sure that an $AB \rightarrow C$ relationship exists, but that fact will not enable us to notice any specific feature of a C relationship from the set {A,B}.

Hence {AB} is not a superkey of the relation R and as a result may not be able to uniquely identify every tuple in the R. By this logic, we can confirm that {AB} is not a candidate key for the given relation R(A,B,C,D,E).

Now, let's check whether abd is a candidate key of the given relation R(A, B, C, D, E):Uniqueness: As per $ABD \rightarrow C$, all the tuples which have the same values for ABD should have the same value for C.

The same is true for $ABD \rightarrow E$ and also for $ABD \rightarrow B$. Thus, clearly ABD is the key attribute that uniquely denotes each tuple in the relation.

5. Answer

a. Functional Dependencies (F):
   i. $M \rightarrow MP$
   ii. $\{M, Y\} \rightarrow P$
   iii. $MP \rightarrow C$

Now, let’s evaluate each attribute combination as a candidate key:

I. {M}:
   - M → MP: This functional dependency implies that knowing M allows us to determine MP.
   - M → Y: Not given.
   - M → P: Not given.
   - M → C: Not given.

   Since M can determine MP, it is a superkey.
   However, M cannot determine all attributes (Y, P, and C), so it is not a candidate key.

II. {M, Y}:
   - {M, Y} → P: This functional dependency implies that knowing M and Y allows us to determine P.
   - {M, Y} → MP: Not given.
   - {M, Y} → C: Not given.

   Since {M, Y} can determine P, it is a superkey.
   Moreover, {M, Y} can determine all attributes (MP and C), so it is a candidate key.

III. {M, C}:
   - {M, C} → P: Not given.
   - {M, C} → MP: Not given.
   - {M, C} → Y: Not given.

   Since {M, C} cannot determine all attributes (P), it is not a superkey and therefore not a candidate key.

b. The relation REFRIG has a candidate key: {M, Y}.
   Since {M, Y} is the only candidate key, let's examine if the relation REFRIG is in 3NF and BCNF:

   1. 3NF: A relation is in 3NF if the non-trivial functional dependency X→Y satisfies either X is a superkey or Y is a part of, any, candidate key. Under this scenario, consequently all FDs must follow the rule and therefore REFRIG is already in 3NF

   2. BCNF: If there is a non-trivial functional dependency X>Y, superkey X is a superset. Here, the dependent functioning of MP→C is contradicting because MP is not a superkey. Hence, REFRIG doesn’t meet Boyce-Codd’s Normal Forms requirements.

6. Answer
Non-Trivial Functional Dependencies (FDs):

a. Based on the assumptions, we can identify the following non-trivial functional dependencies:
   i. isbn → title, author, publisherName, publisherAdd, totalCopiesOrdered, copiesInStock, publicationDate, category, sellingPrice, cost
   ii. author → title, isbn, publisherName, publisherAdd, totalCopiesOrdered, copiesInStock, publicationDate, category, sellingPrice, cost
   iii. publisherName → publisherAdd
   iv. totalCopiesOrdered → copiesInStock
   v. isbn → sellingPrice, cost

Candidate Keys:

a. The isbn uniquely identifies a book, so it is a candidate key.
b. Additionally, the combination of title and author (since a book may have multiple authors) can also serve as a candidate key.
c. Therefore, the candidate keys are {isbn} and {title, author}.
d. The primary key can be chosen from either of these candidate keys, depending on the specific requirements of the system.

Third Normal Form (3NF):

a. To determine if the relation is in 3NF, we need to check for transitive dependencies.
b. The functional dependency totalCopiesOrdered → copiesInStock is a transitive dependency because copiesInStock depends on totalCopiesOrdered via isbn.
c. To achieve 3NF, we can decompose the relation into smaller relations while preserving dependencies. For example:
   I. Books1 (isbn, title, author, publisherName, publisherAdd, publicationDate, category)
   II. Books2 (isbn, totalCopiesOrdered, copiesInStock, sellingPrice, cost)

This decomposition ensures that each relation has no transitive dependencies.

Boyce-Codd Normal Form (BCNF):

a. To check if the relation is in BCNF, we need to verify that every non-trivial functional dependency has a superkey on the left-hand side.
b. The functional dependency isbn → sellingPrice, cost violates BCNF because isbn is not a superkey.
c. To achieve BCNF, we can further decompose the relation:
   1. Books1 (isbn, title, author, publisherName, publisherAdd, publicationDate, category)
   2. Books2 (isbn, totalCopiesOrdered, copiesInStock)
   3. Books3 (isbn, sellingPrice, cost)

Now each relation satisfies BCNF.