Database Management System: Assignment 4

Total Marks : 20

July 26, 2023

Question 1

Consider the relational schema `OfficeList(Sector, Company, Building, Floor, Employees)` with the following Functional Dependency set:
{`Sector, Company`} → {`Building, Floor`}
{`Company, Building`} → {`Employees, Sector`}
`Building` → `Floor`
Which attribute is present in all the composite candidate keys of `OfficeList`?

Marks: 2 **MCQ**

a) `Sector`

b) `Company`

c) `Building`

d) `Floor`

Answer: b)
Explanation: The candidate keys of the relation `OfficeList` are {`Company, Sector`} and {`Company, Building`}. Here, the common attribute is `Company`.
Hence, option b) is correct.

Question 2

Consider the relational schema:
`DeviceLogs(Device, OperatingSystem, Logins, DateOfRecord, DeviceLocation)` with the following Functional Dependency set:
{`Device, OperatingSystem`} → `DeviceLocation`
{`Logins, DateOfRecord`} → `OperatingSystem`
`DeviceLocation` → `DateOfRecord`
If `X` is the number of candidate keys of `DeviceLogs` and `Y` is its highest Normal Form, find the values of `X` and `Y`. *Marks: 2* **MCQ**

a) `X=1, Y=2`

b) `X=2, Y=1`

c) `X=3, Y=3`

d) `X=2, Y=3`

Answer: c)
Explanation: The candidate keys of the relation `DeviceLogs` are
{`Device, Logins, OperatingSystem`}, {`Device, Logins, DateOfRecord`}, {`Device, Logins, DeviceLocation`}. For a relation to be in 2NF, the absence of partial dependency can be ensured if none of the non-prime attributes (considering all candidate keys) is functionally dependent on any subset of the candidate keys. The relation is in 2NF as none of the non-prime attributes is functionally dependent on any sub-set of the candidate keys (as there is no non-prime attribute in the relation). Since all the attributes to the R.H.S of the functional dependencies are prime attributes, the relation is in 3NF.
Hence, option c) is correct.

Question 3

Consider the relational schema `DataLabelling(DataSetName, DataFeature1, DataFeature2, Label, Confidence, Annotators)` with the following Functional Dependency set:
`DataSetName` $\rightarrow$ {`Annotators, DataFeature2`}
{`DataFeature1, DataFeature2`} $\rightarrow$ `Label`
{`Label, DataFeature1`} $\rightarrow$ `Confidence`
`DataFeature2` $\rightarrow$ `DataSetName`
Which attribute can NOT be derived directly or indirectly from `DataFeature2`?

Marks: 2 **MSQ**

a) `DataSetName`

b) `DataFeature1`

c) `Label`

d) `Annotators`

Answer: b), c)
Explanation: The closure of $\texttt{DataFeature2}^{+}$={`DataFeature2, DataSetName, Annotators`}.
Thus, `DataFeature1` and `Label` cannot be derived from it.
Hence, options b) and c) are correct.

Question 4

Consider the relational schema `LuckyDraw(Box, Item, Price, Picked, WinningPrize)`. Which of the following set of functional dependencies should be chosen so that `LuckyDraw` can be in 2NF but not in 3NF? *Marks: 2* **MCQ**

a) `Box` → {`Item, Price`}
   `Price` → `WinningPrize`
   `Item` → `Picked`

b) {`Box, Item`} → `Price`
   `Price` → `WinningPrize`
   `Item` → `Picked`

c) {`Item, Box`} → `Price`
   {`Item,Price`} → `WinningPrize`

d) {`Item, Box, Picked`} → `Price`
   {`Item, Price`} → `WinningPrize`
   {`WinningPrize, Price`} → {`Item, Box`}

Answer: a)
Explanation: In option (a), the primary key is `Box`. Thus, there can not be any partial dependency. However, transitive dependency exists in the second and third functional dependencies. Option (b) is in 1NF as `Picked` is dependent on partial key `Item`. Similarly, for (c), the primary key is {`Box, Item, Picked`} and thus, partial dependency exists. Option (d) is in 3NF as all the attributes of the schema are prime attributes.
Hence, option (a) is correct.

Question 5

Consider the relational schema Tournament(Champion, Category, Participants, Judges, Day) with the following Functional Dependency set F.
{Category, Day} → Champion
{Category, Champion} → {Participants, Judges}
Day → {Judges, Category}
What is the canonical cover of F? *Marks: 2* **MCQ**

a) Category → Champion
   Champion → Participants
   Champion → Judges
   Day → Category

b) Day → Champion
   {Category, Champion} → Participants
   {Category, Champion} → Judges

c) Day → Champion
   {Category, Champion} → Participants
   {Category, Champion} → Judges
   Day → Category

d) {Category, Day} → Champion
   {Category, Champion} → Participants
   {Category, Champion} → Judges
   Day → { Judges, Category }

Answer: c)
Explanation: Category in the L.H.S of the first Functional dependency is extraneous as Day^+ contains Category using set F. Similarly, Judges in the R.H.S of the last Functional dependency is extraneous as Day^+ contains Judges using the set {F-Day → {Judges, Category}}∪{Day → {Category}}.
Hence, option (c) is correct.

Question 6

Consider the relational schema `Gallery(GallerySection, ArtistID, ArtID, Sold, Theme)` with the following Functional Dependency set:
{`GallerySection, ArtistID`} → `ArtID`
`ArtID` → `Sold`
{`ArtID, ArtistID`} → `Theme`
The relation is decomposed into the following:
`Gallery1(GallerySection, ArtistID, ArtID)`
`Gallery2(GallerySection, ArtistID, Sold)`
Which of the following is true about the decomposition? *Marks: 2* **MCQ**

a) Both lossless and dependency preserving.

b) Neither lossless nor dependency preserving.

c) Lossless but not dependency preserving.

d) Lossy but dependency preserving.

Answer: b)
Explanation: From the given decomposition, we can see that the union between the attributes of `Gallery1` and `Gallery2` does not contain `Theme`. Hence, it is not lossless. Similarly, {`ArtID, ArtistID`} → `Theme` can not be derived from the decomposition. Hence, it is not dependency-preserving.
Hence, option b) is correct.

Question 7

Consider the relational schema `Gallery(GallerySection, ArtistID, ArtID, Sold, Theme)` with the following Functional Dependency sets:

`S1={`
`{GallerySection, ArtistID}` $\rightarrow$ `ArtID`
`Sold` $\rightarrow$ `{ArtID, Theme}`
`}`
`S2={`
`GallerySection` $\rightarrow$ `{ArtistID, ArtID}`
`Sold` $\rightarrow$ `{GallerySection, Theme}`
`}`

Which of the following is true? *Marks: 2* **MCQ**

a) Neither `S1` covers `S2` nor `S2` covers `S1`

b) `S1` covers `S2` but `S2` does not cover `S1`

c) `S2` covers `S1` but `S1` does not cover `S2`

d) Both `S1` covers `S2` and `S2` covers `S1`

Answer: c)
Explanation: `GallerySection` $\rightarrow$ `{ArtistID, ArtID}` cannot be derived from S1 as `GallerySection`$^+$ does not contain `{ArtistID, ArtID}`. All FDs of S1 can be derived from S2.
Hence, option (c) is correct.

Question 8

Consider the relational schema `Gallery(GallerySection, ArtistID, ArtID, Sold, Theme)` with the following Functional Dependency set:
{`GallerySection, ArtistID`} → `ArtID`
`ArtID` → `Sold`
{`ArtID, ArtistID`} → `Theme`
Which of the following is true? *Marks: 2* **MCQ**

a) `Gallery` has 1 candidate key and is in 1NF.

b) `Gallery` has 1 candidate key and is in 2NF.

c) `Gallery` has 2 candidate key and is in 3NF.

d) `Gallery` has 2 candidate key and is in 1NF.

Answer: b)
Explanation: The candidate key is {`GallerySection, ArtistID`}. Although {`ArtID, ArtistID`} → `Theme`, {`ArtID, ArtistID`} is not a proper subset of {`GallerySection, ArtistID`}. Hence, the relation is in 2NF.
Hence, option b) is correct.

Question 9

In a relation `FurnitureStore(FurnitureNo, FurnitureType, Price, Width, Height, Weight, DeliveryCharge)`, `FurnitureNo` identifies `FurnitureType` and `Price`. Also, `FurnitureType, Width, Height` and `Weight` combined determines the `DeliveryCharge`. `Width` and `Height` of the furniture are dependent on `FurnitureNo` and `Weight` together. Which of the following are the non-prime attributes of `FurnitureStore`? *Marks: 2* **MSQ**

a) `FurnitureNo`

b) `Height`

c) `Weight`

d) `DeliveryCharge`

Answer: b), d)
Explanation: The given Functional dependencies are:
`FurnitureNo` $\rightarrow$ {`FurnitureType, Price`}
{`FurnitureType, Width, Height, Weight`} $\rightarrow$ `DeliveryCharge`
{`FurnitureNo, Weight`} $\rightarrow$ {`Width, Height`}
The candidate key is {`FurnitureNo, Weight`}.
Hence, options (b), and (d) are correct.

Question 10

In a relation `FurnitureStore(FurnitureNo, FurnitureType, Price, Width, Height, Weight, DeliveryCharge)`, `FurnitureNo` identifies `FurnitureType` and `Price`. Also, `FurnitureType, Width, Height` and `Weight` combined determines the `DeliveryCharge`. `Width` and `Height` of the furniture are dependent on `FurnitureNo` and `Weight` together. The highest normal form of `FurnitureStore` is `n`. If the highest normal form of the relation has to be increased to `n+1`, which of the following changes should be made in its current functional dependency set?

Marks: 2 **MCQ**

a) `FurnitureNo` and `Weight` together, should identify `FurnitureType` and `Price`

b) Only `Width, Height` and `Weight` should identify the `DeliveryCharge`

c) Only `FurnitureType, Height` and `Weight` should identify the `DeliveryCharge`

d) Only `Weight` should identify the `Width` and `Height`

Answer: a)
Explanation: The given Functional dependencies are:
`FurnitureNo` $\rightarrow$ {`FurnitureType, Price`}
{`FurnitureType, Width, Height, Weight`} $\rightarrow$ `DeliveryCharge`
{`FurnitureNo, Weight`} $\rightarrow$ {`Width, Height`}
The candidate key is {`FurnitureNo, Weight`}. The relation is in 1NF, as partial dependency exists in `FurnitureNo` $\rightarrow$ {`FurnitureType, Price`}. If `Weight` is added to the L.H.S of this dependency, the relation reaches 2NF.
Hence, option (a) is correct.