INFS1200/7900 Module 4 Assignment - Solutions

Due: 28 May 2021 @ 04:00 PM AEST

Weighting: 15%

Full Name	Student ID
SOLUTIONS	SOLUTIONS

1. Overview

The purpose of this assignment is to test your ability to use and apply functional dependencies and normalisation in the database design process. You will gain experience in isolating problems with relational schema by determining the appropriate normal form and finding examples of potential anomalies in relation instances. Finally, you will generate an efficient database schema using both the BCNF and 3NF approach to normalisation.

This assignment **must** be completed individually.

2. Submission

All submissions must be made through an electronic marking tool called Gradescope, which will also be used for providing feedback. You **must** record all your answers in the spaces provided in this document. Altering the format or layout of this document in any way will attract penalties**.**

3. Marking

The Module 4 assignment is worth **15 course marks** (of 60 course marks total for all four assignments).

4. Plagiarism

The University has strict policies regarding plagiarism. Penalties for engaging in unacceptable behaviour can range from cash fines or loss of grades in a course, through to expulsion from UQ. You are required to read and understand the policies on academic integrity and plagiarism in the course profile (Section 6.1).

If you have any questions regarding what is considered an acceptable level of collaboration with your peers, please see either the lecturer or your tutor for guidance. Remember that ignorance is not a defence!

5. Task

This assignment has been split into five sections which each examine a key topic covered in Module 4 of this course. Each section will have several questions with appropriate instructions to assist you in completing that question. You must complete **all** sections of this assignment and submit your answers using the boxes provided.

Section A – Anomalies & Functional Dependencies

Question 1

A local computer store *Apples & Pears*^{TM} uses a database to store transactions related to customer purchases. A sample of a table from their database has been provided below. For simplicity, the attribute names have been simplified to single letters.

A	B	C	D	E	F
1	iPear 11	Jack	10/05/2020	12	2000
2	Apple Server 2019	Jack	12/04/2021	14	3000
3	Pear Mini	Grace	10/05/2021	13	2000
4	Apple Server 2019	Andrew	18/04/2021	14	3000
5	Pear Mini	XiaoMing	11/05/2021	13	2000
6	PearPods	Naomi	18/04/2021	15	1500

Based on the data above, provide a list of *all possible* non-trivial functional dependencies for this table. You do not need to justify your answers.

Note: For this question, non-trivial FD's with more than one attribute on the left-hand side should also not be included. For example, the following two FD's should not be included:

{Z} → {Z} {Z, Y} → {Z}

Additionally, all FD's with the same values on the left hand side must be simplified into one FD. For example:

{Z} → {X}, {Z} →{W} ⛔ {Z} → {X, W} ✓

{A} → {B, C, D, E, F}
{B} → {E, F}
{E} → {B, F}
{B, C, D, E, F} → {A}

Question 2

Cafe64 has implemented a database system to record the Coffees made by each Barista. The schema for their Transactions table is provided below including a snapshot of some sample data.

ID	BaristaID	CoffeeID	BaristaName	Bean	GrindTime
70	25	1435	Yeseul	Arabica	30
72	25	1435	Yeseul	Robusta	38
73	23	1348	Patrick	Liberica	25
74	24	2345	Jessie	Arabica	30
75	24	1348	Jessie	Robusta	38
76	25	2345	Yeseul	Liberica	25

This table contains the following non-trivial functional dependencies:

{ID} → {BaristaID, CoffeeID, BaristaName, Bean, GrindTime}
{BaristaID} → {BaristaName}
{Bean} → {GrindTime}

Using the table above, you must provide a brief example and explanation of database operations that which would cause an insertion, modification and deletion anomaly. Your explanation should be brief and not exceed 50 words. An example of the format your answers should be written in can be seen on the next page.

Example format:

Operation:
Insert <71, 25, 1435, “Yeseul”, “Arabica”, 30> into Transactions

Explanation:
This operation would cause an ____ anomaly to occur because … (Note the above operation is simply an example and does not cause an anomaly.)

Insertion Anomaly:

Operation:
Insert <NULL, NULL, NULL, NULL, “Excelsa”, 45> into *Transactions*

Explanation:
If we wanted to store a new bean and associated grind time in our system, we would need to enter a tuple with an empty value for ID which would cause an entity integrity error hence making this operation impossible to perform.

EXAMPLE ONLY

Modification Anomaly:

Operation:
Update the tuple <70, 25, 1435, “Yeseul”, “Arabica”, 30> in *Transactions* to <70, 25, 1435, “Grace”, “Arabica”, 30>

Explanation:
If we wanted to update the name of the barista with ID 70 as in the example operation above, then we would need to manually update each other tuple in the instance data for which the old value appears. Otherwise, this would lead to an inconsistent state in our database.

EXAMPLE ONLY

Deletion Anomaly:

Operation:
Delete from *Transactions* where ID = 73

Explanation:
Deleting the transaction with ID 73 from the relation would cause us to lose all information regarding barista with ID 23 as this is the only tuple in which that barista is referenced. Hence, this operation results in unintentional data loss for the database.

ONLY CORRECT ANSWER

SEE NEXT PAGE FOR SECTION B

Section B – Keys

For each of the questions in this section, you are required to list all possible candidate keys for the given schema based on the functional dependencies provided. You may wish to compute the closure of your key(s) to confirms they are valid.

Question 1

R [A, B, C, D, E, F, G, H, I, J]

{A, B, C} → {D, E, F}

{E, F} → {D, C}

{D} → {G, H, I}

{E, G} → {I, J}

Candidate Key(s):
{A, B, C}, {A, B, E, F}

Question 2

R [A, B, C, D, E, F, G, H]

{A} → {B, F}

{B, F} → {A}

{A, B} → {D, E, G}

{E} → {F}

Candidate Key(s):
{A, C, H}, {C, H, B, E}, {C, H, B, F}

Question 3

R [A, B, C, D, E, F, G, H, I, J, K]

{A, B, E} → {G, J, K, D}

{J, G} → {E, H, B, C}

{H, I} → {F, J}

{D, C, F} → {A, B, E}

Candidate Key(s):
{A, B, E, I}, {A, G, H, I}, {A, G, I, J}, {C, D, F, I}, {C, D, H, I}, {D, G, H, I}, {D, G, I, J}

Section C – Highest Normal Form

For each question in this section, you are required to state and justify the highest normal form of the relation given a schema and functional dependencies. Your explanations should be brief and concise. *Hint*: It may be useful to identify the candidate keys for each relation.

Question 1

R [A, B, C, D, E]

{C, D} → {A, E, B}

{A} → {C}

Highest Normal Form:
3NF

Explanation:
The candidate keys for this table are {C, D} and {A, D}. The right-hand side of the second functional dependency is a prime attribute so both 2NF and 3NF conditions are meet. However, the LHS of the second functional dependency is not a super key and violates the BCNF definition.

Question 2

R [A, B, C, D, E, F, G, H]

{E, F} → {D, G, H}

{G, H} → {C}

Highest Normal Form:
1NF

Explanation:
The candidate keys for this table are {A, B, C}, {A, B, E, F} and {A, B, G, H}. Hence the set of attributes on the left-hand side of the first functional dependency are a proper subset of one of the candidate keys. This violates 2NF meaning 1NF is the highest normal form.

Question 3

R [A, B, C, D, E, F, G, H, I]

{A, D, F} → {B, E, C, G, H, I}

{D, C} → {A, F}

{G, E} → {D}

{H} → {I}

Highest Normal Form:
2NF

Explanation:
The candidate keys for this table are {C, D}, {A, D, F}, {A, E, F, G} and {C, E, G}. The LHS of the first two functional dependencies are super key and the RHS are non-prime attributes hence they do not violate 2NF or 3NF. The RHS of the third functional dependency is also a prime attribute so it also does not violate 2NF or 3NF. The LHS of the final functional dependency is not a proper subset of any candidate key and thus does not violate 2NF. However, it is not a super key and this does violate 3NF.

Section D – BCNF Decomposition

For each question in this section, you are required to decompose the given relation into BCNF form and state any new relations created in the process with their functional dependencies and identify any functional dependencies which are fully lost during the decomposition. You must show your work using the **tree method** presented in tutorials. Consider the functional dependencies in the order presented in the question.

Question 1

R [A, B, C, D, E, F, G, H, I, J]

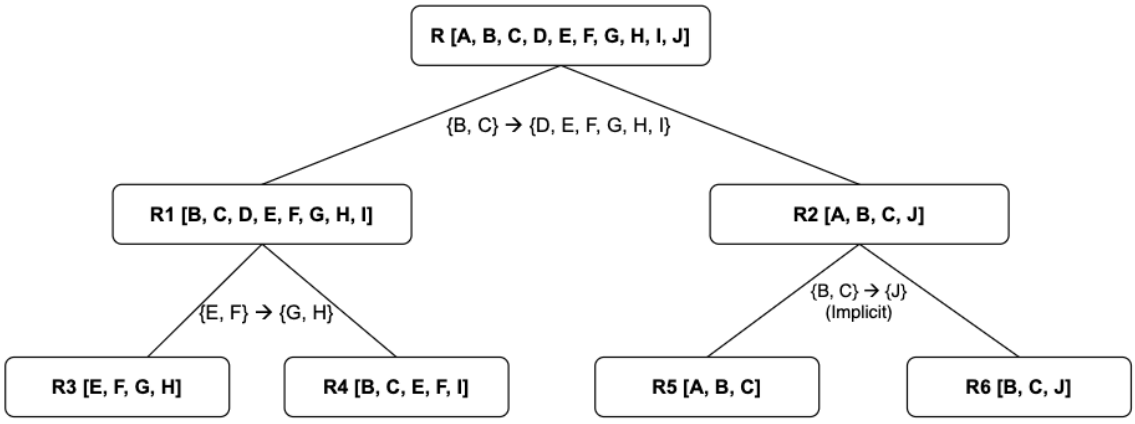
{A} → {B, C, D, E, F, G, H, I, J}

{B, C} → {D, E, F, G, H, I}

{E, F} → {G, H}

{I} → {J}

Candidate Key: {A}



New Relationships:

R3 [E, F, G, H]
{E, F} → {G, H}

R4 [B, C, E, F, I]
{B, C} → {E, F, I}

R5 [A, B, C]
{A} → {B, C}

R6 [B, C, J]
{B, C} → {J} (Implicit)

Functional Dependencies Fully Lost:
{I} → {J}

Question 2

R [A, B, C, D, E, F, G, H]

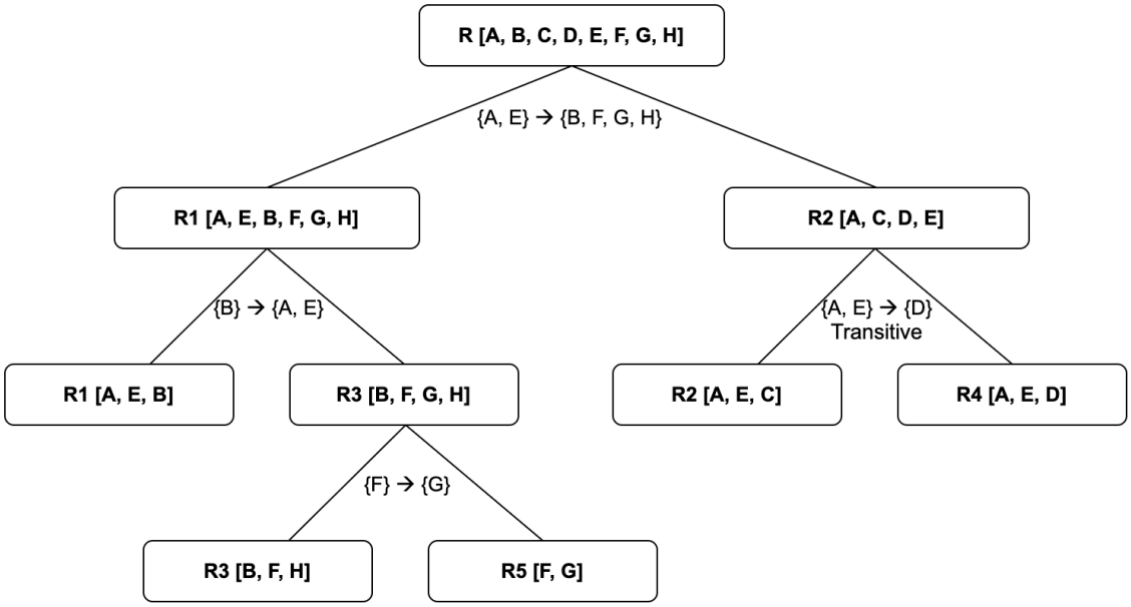
{A, E} → {B, F, G, H}

{B} → {A, D, E}

{F} → {G}

{G} → {F}

Candidate Keys: {C, B}, {C, A, E}



New Relationships:

R1 [A, E, B]
{A, E} → {B}
{B} → {A. E}

R2 [A, E, C]
No non-trivial FD's

R3 [B, F, H]
No non-trivial FD's

R4 [A, E, D]
{A, E} → {D} (Transitivity)

R5 [F, G]
{F} → {G}
{G} → {F}

Functional Dependencies Fully Lost:
None

Section E – 3NF Decomposition

Question 1

Based on the following relational schema and functional dependencies, find minimal cover for relation R with union.

R [A, B, C, D, E, F, G, H, I, J, K, L, M, N]

{A, B, C} → {F, E}

{D} → {G, H, M}

{H, A} → {I, J}

{K, L, D} → {M, H, B}

{I, J} → {N}

{H} → {M}

Minimal Cover:

Step 1: (RHS Simplification)	Step 2: (LHS Simplification)	Step 3: (FD Set Simplification)
{A, B, C} → {F}	{A, B, C} → {F}	{A, B, C} → {F}
{A, B, C} → {E}	{A, B, C} → {E}	{A, B, C} → {E}
{D} → {G}	{D} → {G}	{D} → {G}
{D} → {H}	{D} → {H}	{D} → {H}
{D} → {M}	{D} → {M}	
{H, A} → {I}	{H, A} → {I}	{H, A} → {I}
{H, A} → {J}	{H, A} → {J}	{H, A} → {J}
{K, L, D} → {M}	**{D} → {M}**	
{K, L, D} → {H}	**{D} → {H}**	
{K, L, D} → {B}	{K, L, D} → {B}	{K, L, D} → {B}
{I, J} → {N}	{I, J} → {N}	{I, J} → {N}
{H} → {M}	{H} → {M}	{H} → {M}

Minimal Cover:
{
{A, B, C} → {F}, {A, B, C} → {E}, {D} → {G}, {D} → {H}, {H, A} → {I}, {H, A} → {J}, {K, L, D} → {B}, {I, J} → {N}, {H} → {M}
}

Minimal Cover with Union:
{
{A, B, C} → {F, E}, {D} → {G, H}, {H, A} → {I, J}, {K, L, D} → {B}, {I, J} → {N}, {H} → {M}
}

Question 2

The minimal cover has been provided below for a given relation with a set of functional dependencies. Using the minimal cover, normalise the relation to 3NF such that all functional dependencies and candidate keys are preserved. Your final answer must include any new relations created and their functional dependencies.

R [A, B, C, D, E, F, G, H]

{B} → {A, C, D}

{E} → {F, G}

{G} → {H, E}

Minimal Cover without Union: {

{B} → {A}, {B} → {C}, {B} → {D}, {E} → {F}, {E} → {G}, {G} → {H}, {G} → {E}

}

Candidate Keys: {B, E}, {B, G}

Step 1: Combine all FD's with the same LHS:
{B} → {A, C, D}
{E} → {F, G}
{G} → {H, E}

Step 2: Create relation for all FD's X → Y such that X union Y does not exists in a previous table
R1 [B, A, C, D]
R2 [E, F, G]
R3 [G, H, E]

Step 3: Create a relation for any candidate keys which are missing
R4 [B, E, G]

Step 4: Eliminate redundant relations
-

Final Relational Schema:
R1 [B, A, C, D]
{B} → {A, C, D}

R2 [E, F, G]
{E} → {F, G}

R3 [G, H, E]
{G} → {H, E}

R4 [B, E, G]
{B, E} → {G}
{B, G} → {E}

Question 3

Based on the following relational schema and functional dependencies, find minimal cover for relation R and then decompose R to 3NF such that all functional dependencies and candidate keys are preserved. Your final answer must include any new relations created and their functional dependencies.

R [A, B, C, D, E, F, G, H, I, J, K]

{A, B} → {H, I, J, K, E}

{K} → {A, F, G, B}

{I} → {J, K, G}

{B} → {D, E}

Minimal Cover:

Step 1: (RHS Simplification)	Step 2: (LHS Simplification)	Step 3: (FD Set Simplification)
{A, B} → {H}	{A, B} → {H}	{A, B} → {H}
{A, B} → {I}	{A, B} → {I}	{A, B} → {I}
{A, B} → {J}	{A, B} → {J}	
{A, B} → {K}	{A, B} → {K}	
{A, B} → {E}	**{B} → {E}**	
{K} → {A}	{K} → {A}	{K} → {A}
{K} → {F}	{K} → {F}	{K} → {F}
{K} → {G}	{K} → {G}	
{K} → {B}	{K} → {B}	{K} → {B}
{I} → {J}	{I} → {J}	{I} → {J}
{I} → {K}	{I} → {K}	{I} → {K}
{I} → {G}	{I} → {G}	{I} → {G} – See note below
{B} → {D}	{B} → {D}	{B} → {D}
{B} → {E}	{B} → {E}	{B} → {E}

Minimal Cover:
{
{A, B} → {H}, {A, B} → {I}, {K} → {A}, {K} → {F}, {K} → {B}, {I} → {J}, {I} → {K}, {I} → {G}, {B} → {D}, {B} → {E}
}

Candidate Keys: {C, I}, {C, K}, {C, A, B}

Step 1: Combine all FD's with the same LHS:
{A, B} → {H, I}
{K} → {A, F, B} or {K} → {A, F, G, B}
{I} → {J, K, G} or {I} → {J, K}
{B} → {D, E}

For this question you could remove *either* {K} → {G} or {I} → {K} but **not** both. Both methods were accepted.

Step 2: Create relation for all FD's X → Y such that X union Y does not exists in a previous table
R1 [A, B, H, I]
R2 [K, A, F, B] or **R2 [K, A, F, G, B]**
R3 [I, J, K, G] or **R3 [I, J, K]**
R4 [B, D, E]

Step 3: Create a relation for any candidate keys which are missing
R5 [C, I, K, A, B]

Step 4: Eliminate redundant relations
-

Final Relational Schema (Option 1):

R1 [A, B, H, I]	**R2 [K, A, F, G, B]**	**R3 [I, J, K, G]**
{A, B} → {H, I}	{K} → {A, F, G, B}	{I} → {J, K, G}
R4 [B, D, E]	**R5 [C, I, K, A, B]**	
{B} → {D, E}	{C, I} → {K, A, B}, {C, A} → {K, I, B}, {C, A, B} → {K, I}	