COMPSCI 2DB3 Assignment 6

Elite Lu lue13

Question 1

The minimal cover is shown below with the explanations:

Functional Dependency	Reasoning
pet_id → name	Each pet has a unique name associated with them.
pet_id → pet_age	Each pet only has exactly one age. Having multiple would not be plausible.
pet_id, owner_id → since	For each owner_id and pet_id, there is one start date for when the person became an owner to the pet.
owner_id → owner_age	Each owner has exactly one age.

Question 2

The pets are independent of the owners that are present in the table. This independence is expressed via either the multivalued dependency "pet_id, name, pet_age ↠ owner_id, owner_age, since", and the multivalued dependency "pet_id, name, pet_age ↠ pet_friend_id" or the join dependency "⋈{X, Y}" where X = {pet_id, name, pet_age, owner_id, owner_age } and Y = {pet_id, name, pet_age, pet_friend_id}

Question 3

This table is not in 3NF. An example of a dependency that violates this property is "user_id → user_location". This dependency is not trivial because "user_id" is not a key nor a superkey of the relational schema and along with this, "user_location" is not part of any other key.
In order to decompose this schema, I will use the algorithm DECOMPOSE-3NF to put the relational schema into 3NF. This can be achieved by determining a minimal cover for all the provided functional dependencies.

Note that there is one problem where in order to achieve the minimal cover, I must remove all the redundant schemas such that they will not impact the keys and dependencies. This is present with id → user_location, id → user_id, and user_id → user_location. This is a scenario where A → B, A → C, and B → C. in order to preserve the dependencies and make this the minimal cover, I will remove A → C since removing this will not impact the keys and superkeys along with the dependencies. The same can be said for book_id → age_rating, book_id → child_friendly, and age_rating → child_friendly. The minimal cover is shown as such:

$\mathfrak{S}'$ = {id → user_id, id → date,
user_id → user_location,
book_id → title, book_id → publisher, book_id → age_rating,
id, book_id, format → amount,
book_id, format → isbn,
isbn → book_id, isbn → format,
age_rating → child_friendly}

With this, I will then use the for-loop of the decomposition algorithm and construct a relational schema A → X in the minimal cover with attributes $A \cup B$, where $B = \{Y \mid A \dashrightarrow Y \in \mathfrak{S}'\}$. From this, I will get the following:

Schemas	Dependencies

r_0	r_0(id, user_id, date) from id → user_id, id → date
r_1	r_1(user_id, user_location) from user_id → user_location
r_2	r_2(book_id, title, publisher, age_rating, child_friendly) from book_id → title, book_id → publisher, book_id → age_rating
r_3	r_3(id, book_id, format, amount) from id, book_id, format → amount
r_4	r_4(book_id, format, isbn) from book_id, format → isbn
r_5	r_5(book_id, format, isbn) from isbn → book_id, isbn → format
r_6	r_6(age_rating, child_friendly) from age_rating → child_friendly

With this, we must consider the keys that have not been used and any redundant relational schemas. The keys that have not been used include category and author. Because of this. I will create another relational schema, shown below:

r_7	r_7(id, book_id, format, author, category)

Next, when I will remove the redundant schemas. This would include r_4. This is because r_4 is the same as r_5. Because of this, the following functional schemas hold each relational schema of the resulting decomposition:

Relational Scheme	Functional Dependencies
r_0(id, user_id, date)	"id → user_id, date".
r_1(user_id, user_location)	"user_id → user_location"
r_2(book_id, title, publisher, age_rating)	"book_id → title, publisher, age_rating".
r_3(id, book_id, format, amount)	"id, book_id, format → amount".
r_5(book_id, format, isbn)	"isbn → book_id, format". "isbn → book_id, isbn → format".
r_6(age_rating, child_friendly)	"age_rating → child_friendly".
r_7(id, book_id, format, author, category)	

Not that the decomposition is lossless-join and dependency-preserving, as we have used the DECOMPOSE-3NF algorithm, which guarantees lossless-join and dependency-preservation.

Finally, the decomposition of the dataset is shown below. Note that the naming convention for the columns is with respect to the original document:

r_0

Id	Ui	D
1	1	24 Dec.
2	2	25 Dec.
3	1	26 Dec.

r_1

Ui	Ul
1	Hamilton
2	Toronto

r_2

B	T	P	Ar
2	Book!	ThePrinter	18+
5	Comic!	TheCopier	5+

r_3

Id	B	F	Am
1	2	paperback	3
1	2	hardcover	6
2	2	e-book	1
3	5	paperback	1

r_5

B	F	Is
2	paperback	1234
2	hardcover	1237
2	e-book	1241
5	paperback	1298

r_6

Ar	Cf
18+	no
5+	yes

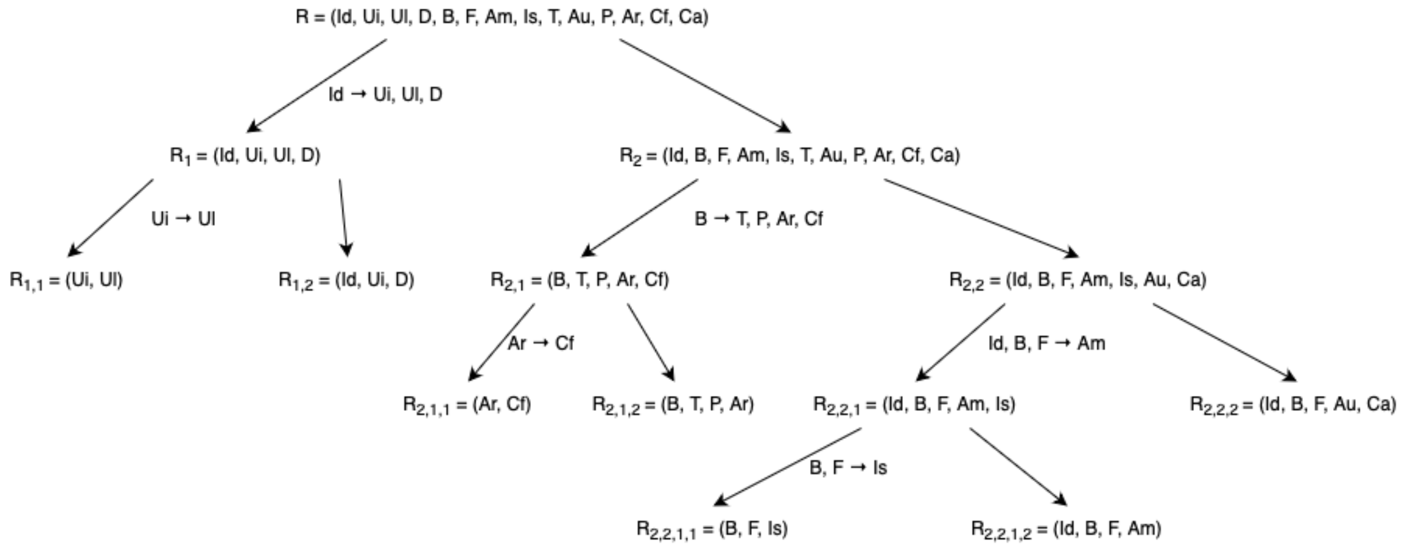
r_7

Id	B	F	Au	Ca
1	2	paperback	Alicia	CS
1	2	paperback	Alicia	Theory
1	2	paperback	Dafni	CS
1	2	paperback	Dafni	Theory
1	2	hardcover	Dafni	CS
1	2	hardcover	Dafni	Theory
1	2	hardcover	Dafni	CS
1	2	hardcover	Dafni	Theory
2	2	e-book	Alicia	CS
2	2	e-book	Alicia	Theory
2	2	e-book	Dafni	CS
2	2	e-book	Dafni	Theory
2	5	paperback	Bo	Comedy

Question 4

The schema Order is not in BCNF. This is because it is not in 3NF a BCNF is also in BNF.

Because of this, I will use the algorithm DECOMPOSE-BCNF to put the schema into BCNF. First, I will construct the table and then explain each of the splits and show how the steps make the schema become a composition of binary relations. Here are the following steps:



R = (Id, Ui, Ul, D, B, F, Am, Is, T, Au, P, Ar, Cf, Ca)
Id → Ui, Ul, D
R_1 = (Id, Ui, Ul, D)
R_2 = (Id, B, F, Am, Is, T, Au, P, Ar, Cf, Ca)
Ui → Ul
B → T, P, Ar, Cf
$R_{1,1}$ = (Ui, Ul)
$R_{1,2}$ = (Id, Ui, D)
$R_{2,1}$ = (B, T, P, Ar, Cf)
$R_{2,2}$ = (Id, B, F, Am, Is, Au, Ca)
Ar → Cf
Id, B, F → Am
$R_{2,1,1}$ = (Ar, Cf)
$R_{2,1,2}$ = (B, T, P, Ar)
$R_{2,2,1}$ = (Id, B, F, Am, Is)
$R_{2,2,2}$ = (Id, B, F, Au, Ca)
B, F → Is
$R_{2,2,1,1}$ = (B, F, Is)
$R_{2,2,1,2}$ = (Id, B, F, Am)

a) We begin with R = (Id, Ui, Ul, D, B, F, Am, Is, T, Au, P, AR, Cf, Ca). Because 'Id' is not a key of this R, this means we must split. The relationship that will be used for the split is id → user_id, user_location, date. With this, I am left two relations R_1 and R_2 where R_1 = (Id, Ui, Ul, D) and R_2 = (Id, B, F, Am, is, T, Au, P, Ar, Cf, Ca).

b) The relational schema R_1 is still not in BCNF. An example of a violation is user_id → user_location. As a result, we split again to get $R_{1,1}$ and $R_{1,2}$ where $R_{1,1}$ = (Ui, Ul) and $R_{1,2}$ = (Id, Ui, D). Both $R_{1,1}$ and $R_{1,2}$ are in BCNF as $R_{1,1}$ is binary and $R_{1,2}$ has the key Id.

c) R_2 is not in BCNF. The relational dependency book_id → title, publisher, age_rating, child_friendly violates BCNF. Another split occurs here to get $R_{2,1}$ and $R_{2,2}$ where $R_{2,1}$ = (B, T, P, Ar, Cl) and $R_{1,2}$ = (Id, B, F, Am, Is, Au, Ca).

d) With the previous split, I will split on $R_{2,1}$ because it is not in BCNF as age_rating → child_friendly violates it. A split occurs here to get $R_{2,1,1}$ and $R_{2,1,2}$ where $R_{2,1,1}$ = (Ar, Cf) and $R_{2,1,2}$ = (B, T, P, Ar). $R_{2,1,1}$ is binary, so it is in BCNF. $R_{2,1,2}$ is in BCNF now because it has a key which is B.

e) On the rightmost part of the graph, $R_{2,2}$ is still not in BCNF. Because of this, I will split on id, book_id, format → amount. However, this split will be a bit different since I will be including isbn in the dependency that contains the previously aforementioned relation because a later dependency. As a result, I will get $R_{2,2,1}$ and $R_{2,2,2}$ where $R_{2,2,1}$ = (Id, B, F, Am, Is) and $R_{2,2,2}$ = (Id, B, F, Au, Ca). $R_{2,2,2}$ is in BCNF.

f) $R_{2,2,1}$ is still not in BCNF. Using the dependency book_id, format → isbn, I get $R_{2,2,1,1}$ and $R_{2,2,1,2}$ where $R_{2,2,1,1}$ = (B, F, Is) and $R_{2,2,1,2}$ = (Id, B, F, Am). Finally, both are in BCNF since they each have a key, which are B, F and Id, B, F respectively.

With this, the following dependencies hold in the relational schema of the resulting decomposition:

Relation Scheme	Functional Dependencies
$R_{1,1}$(id, location)	user_id → user_location
$R_{1,2}$(id, user_id, date)	From split and id → user_id, user_location, date
$R_{2,1,1}$(age_rating, child_friendly)	age_rating → child_friendly

$R_{2,1,2}$(book_id, title, publisher, age_rating)	From split and dependency book_id → title, publisher, age_rating, child_friendly
$R_{2,2,1,1}$(book_id, format, isbn)	book_id, format → isbn isbn → book_id, format
$R_{2,2,1,2}$(id, book_id, format, amount)	From split and id, book_id, format → amount
$R_{2,2,2}$(id, book_id, format, amount, category)	

The decomposition is a lossless-join since we used the DECOMPOSE-BCNF decomposition algorithm which guarantees a lossless-join decomposition. Surprisingly, this was also dependency preserving. This is because any dependency can be traced back. However, this is not always the case as DECOMPOSE-BCNF does not guarantee this.

Finally, the example dataset is decomposed with accordance to the information above:

$R_{1,1}$

Ui	Ul
1	Hamilton
2	Toronto
1	Hamilton

$R_{1,2}$

Id	Ui	D
1	1	24 Dec.
2	2	25 Dec.
3	1	26 Dec.

$R_{2,1,1}$

Ar	Cf
18+	no
5+	yes

$R_{2,1,2}$

B	T	P	Ar
2	Book!	ThePrinter	18+
5	Comic!	TheCopier	5+

$R_{2,2,1,1}$

B	F	Is
2	paperback	1234
2	hardcover	1237
2	e-book	1241
5	paperback	1298

$R_{2,2,1,2}$

Id	B	F	Am
1	2	paperback	3
1	2	paperback	6
2	2	e-book	1
3	5	paperback	1

$R_{2,2,2}$

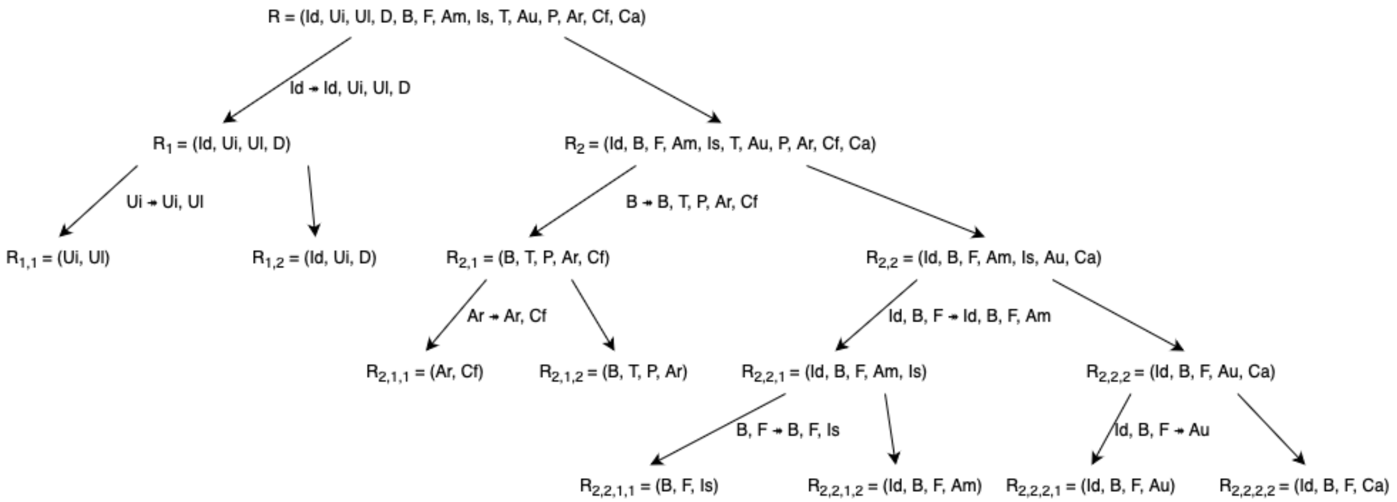
Id	B	F	Au	Ca
1	2	paperback	Alicia	CS
1	2	paperback	Alicia	Theory
1	2	paperback	Dafni	CS
1	2	paperback	Dafni	Theory
1	2	hardcover	Dafni	CS
1	2	hardcover	Dafni	Theory
1	2	hardcover	Dafni	CS
1	2	hardcover	Dafni	Theory
2	2	e-book	Alicia	CS
2	2	e-book	Alicia	Theory
2	2	e-book	Dafni	CS
2	2	e-book	Dafni	Theory
2	5	paperback	Bo	Comedy

Question 5

Like the previous two question, this question is not in 4NF. This is because the table is not in BCNF. For something to be a 4NF, it must also be in BCNF as all 4NFs are BCNF.

To make this 4NF, I will use the DECOMPOSE-4NF algorithm to make this a 4NF. We use the violation $\alpha \twoheadrightarrow \alpha^+$, where α^+ is the attribute closure of α with respect to only the functional dependencies. In addition to this, I will use the attribute that $\alpha \rightarrow \alpha^+$ implies $\alpha \twoheadrightarrow \alpha^+$ because of replication.

I will reuse some of BCNF and along with that, I will evaluate non-binary relation schemes to determine if they introduce new dependencies for relational schemas.



R = (Id, Ui, Ul, D, B, F, Am, Is, T, Au, P, Ar, Cf, Ca)
Id ↠ Id, Ui, Ul, D
R_1 = (Id, Ui, Ul, D)
R_2 = (Id, B, F, Am, Is, T, Au, P, Ar, Cf, Ca)
Ui ↠ Ui, Ul
B ↠ B, T, P, Ar, Cf
$R_{1,1}$ = (Ui, Ul)
$R_{1,2}$ = (Id, Ui, D)
$R_{2,1}$ = (B, T, P, Ar, Cf)
$R_{2,2}$ = (Id, B, F, Am, Is, Au, Ca)
Ar ↠ Ar, Cf
Id, B, F ↠ Id, B, F, Am
$R_{2,1,1}$ = (Ar, Cf)
$R_{2,1,2}$ = (B, T, P, Ar)
$R_{2,2,1}$ = (Id, B, F, Am, Is)
$R_{2,2,2}$ = (Id, B, F, Au, Ca)
B, F ↠ B, F, Is
Id, B, F ↠ Au
$R_{2,2,1,1}$ = (B, F, Is)
$R_{2,2,1,2}$ = (Id, B, F, Am)
$R_{2,2,2,1}$ = (Id, B, F, Au)
$R_{2,2,2,2}$ = (Id, B, F, Ca)

The beginning is very similar to the BCNF as shown below with the steps and method.

a) We begin with R = (Id, Ui, Ul, D, B, F, Am, Is, T, Au, P, AR, Cf, Ca). Because 'Id' is not a key of this R, this means we must split. The relationship that will be used for the split is id ↠ id, user_id, user_location, which is obtained through replication of id → id$^+$. With this, I am left two relations R_1 and R_2 where R_1 = (Id, Ui, Ul, D) and R_2 = (Id, B, F, Am, is, T, Au, P, Ar, Cf, Ca).

b) The relational schema R_1 is still not in 4NF. An example of a violation is user_id ↠ user_id, user_location, obtained from the replication of user_id → user_id$^+$. As a result, we split again to get $R_{1,1}$ and $R_{1,2}$ where $R_{1,1}$ = (Ui, Ul) and $R_{1,2}$ = (Id, Ui, D). $R_{1,1}$ is binary, so it is 4NF. $R_{1,2}$ will be discussed later.

c) R_2 is not in 4NF. The relational dependency book_id ↠ book_id, title, publisher, age_rating, child_friendly, derived from the replication of book_id → book_id$^+$ violates 4NF. Another split occurs here to get $R_{2,1}$ and $R_{2,2}$ where $R_{2,1}$ = (B, T, P, Ar, Cl) and $R_{1,2}$ = (Id, B, F, Am, Is, Au, Ca).

d) With the previous split, I will split on $R_{2,1}$ because it is not in 4NF as age_rating ↠ age_rating, child_friendly violates it, derived from the replication of age_rating → age_rating$^+$. A split occurs here to get $R_{2,1,1}$ and $R_{2,1,2}$ where $R_{2,1,1}$ = (Ar, Cf) and $R_{2,1,2}$ = (B, T, P, Ar). $R_{2,1,1}$ is in 4NF since it is binary. I will discuss $R_{2,1,2}$ later.

e) On the rightmost part of the graph, $R_{2,2}$ is still not in 4NF. Because of this, I will split on id, book_id, format ↠ id, book_id, format, amount, derived from the replication of id, book_id, format → (id, book_id, format)$^+$. However, this split will be a bit different since I will be including isbn in the dependency that contains the previously aforementioned relation because a later dependency. As a result, I will get $R_{2,2,1}$ and $R_{2,2,2}$ where $R_{2,2,1}$ = (Id, B, F, Am, Is) and $R_{2,2,2}$ = (Id, B, F, Au, Ca). $R_{2,2,2}$ will be discussed later.

f) $R_{2,2,1}$ is still not in 4NF. Using the dependency book_id, format ↠ book_id, format, isbn which was derived from replication of book_id, format → (book_id, format)$^+$, I get $R_{2,2,1,1}$ and $R_{2,2,1,2}$ where $R_{2,2,1,1}$ = (B, F, Is) and $R_{2,2,1,2}$ = (Id, B, F, Am). Both will be discussed later.

From here one, I must analyze each of the non-binary scheme and discuss if it is in 4NF.

g) $R_{2,2,2}$ = (Id, B, F, Au, Ca). We have "id → $\mathfrak{J}$", which we can apply replication on this to get "id, B, F ↠ $\mathfrak{J}$". Using transitivity on "id, B, F ↠ $\mathfrak{J}$" and "$\mathfrak{J}$ ↠ author" to get "id, B, F ↠ author". However, id, B, F are not keys, which means I must split to get $R_{2,2,2,1}$ = (Id, B, F, Au) and $R_{2,2,2,2}$ = (Id, B, F, Ca).

h) With the other multivalued dependencies, they do not introduce any new dependencies for the relational schemes. This includes $R_{1,2}$ = (Id, Ui, D), $R_{2,1,2}$ = (B, T, P, Ar), $R_{2,2,1,1}$ = (B, F, Is), and $R_{2,2,1,2}$ = (Id, B, F, Am). Because of this, all these schemes are in 4NF.

Relation Scheme	Functional Dependencies
$R_{1,1}$(id, location)	user_id → user_location
$R_{1,2}$(id, user_id, date)	From split and id → user_id, user_location, date
$R_{2,1,1}$(age_rating, child_friendly)	age_rating → child_friendly
$R_{2,1,2}$(book_id, title, publisher, age_rating)	From split and dependency book_id → title, publisher, age_rating, child_friendly
$R_{2,2,1,1}$(book_id, format, isbn)	book_id, format → isbn isbn → book_id, format
$R_{2,2,1,2}$(id, book_id, format, amount)	From split and id, book_id, format → amount
$R_{2,2,2,1}$(id, book_id, format, author)	
$R_{2,2,2,2}$(id, book_id, format, category)	

It is noted that non-trivial multivalued dependencies hold in this decomposition as that cannot be derived from the above functional dependencies. This decomposition is lossless-join since we strictly abided by the algorithm. However, this is not dependency preserving since “id, book_id, format ↠ author” and “id, book_id, format ↠ category” are not preserved since they cannot be derived from the functional dependencies in the above table.

Finally, here are the tables shown below:

$R_{1,1}$

Ui	Ul
1	Hamilton
2	Toronto
1	Hamilton

$R_{1,2}$

Id	Ui	D
1	1	24 Dec.
2	2	25 Dec.
3	1	26 Dec.

$R_{2,1,1}$

Ar	Cf
18+	no
5+	no

$R_{2,1,2}$

B	T	P	Ar
2	Book!	ThePrinter	18+
5	Comic!	TheCopier	5+

$R_{2,2,1,1}$

B	F	Is
2	paperback	1234
2	hardcover	1237
2	e-book	1241
5	paperback	1298

$R_{2,2,1,2}$

Id	B	F	Am
1	2	paperback	3
1	2	paperback	6
2	2	e-book	1
3	5	paperback	1

$R_{2,2,2,1}$

Id	B	F	Au
1	2	paperback	Alicia

1	2	paperback	Dafni
1	2	hardcover	Dafni
1	2	hardcover	Dafni
2	2	e-book	Alicia
2	2	e-book	Dafni
3	5	paperback	Bo

$R_{2,2,2,2}$

Id	B	F	Ca
1	2	paperback	CS
1	2	paperback	Theory
1	2	hardcover	CS
1	2	hardcover	Theory
2	2	e-book	CS
2	2	e-book	Theory
3	5	paperback	Comedy

Question 6

For each of these decompositions, not all issues are resolved. For the 4NF, author and category are in the same table as id, however there may not be a correlation between them so it would be better to model these with book_id and format.

One small issue of redundancy is the child_friendly and age_rating. This is because if something is rated 18+, then it is not child friendly and if it is 5+, then it is. It is very redundant to use both and if you remove one, you can derive the it with the other field.