Assignment: Decomposition and Normal Forms

Part One: The analysis of a table
Rational:

In the rational, we use the following text annotations:

an entity The highlighted piece of text resembled an entity-like object.
an attribute The highlighted piece of text resembled an attribute of some entity-like object.
a relationship The highlighted piece of text resembled a relationship between entity-like objects.
a constraint The highlighted piece of text resembled a constraint on data.
~~not relevant~~ The highlighted piece of text is not relevant (e.g., context, application details, ...)

~~In this relational schema,~~ each **pet** has a **unique identifier** ***(pet_id)***, a name ***(name)***, and an age ***(pet_age)***. **Each pet can have multiple owners**~~, e.g., a family that has a single dog~~. We **identify** each **owner** by its identifier ***(owner_id)***. Furthermore, each owner has an age ***(owner_age)***. Furthermore, we keep track of the date at which the **owner** started **ownership** of the pet ***(since)***. Finally, we keep track of **all the other pets that are friends of a pet**. Each such pet is identified by its identifier ***(pet_friend_id)***. ~~Next, an example of an instance of this relational schema~~

pet_id	name	pet_age	owner_id	since	owner_age	pet_friend_id
1	Alicia	12	3	Jan. 12	24	2
1	Alicia	12	3	Jan. 12	24	3
1	Alicia	12	5	Feb. 17	29	2
1	Alicia	12	5	Feb. 17	29	3
2	Bo	9	7	Mar. 5	21	1
2	Bo	9	7	Mar. 5	21	3
3	Celeste	5	1	Dec. 22	99	1

Solutions:

1. The function dependencies of the above table would be:

Functional Dependency	Reasoning
pet_id → name	Each pet (identified via pet_id) has a unique name (identified via name)
pet_id → pet_age	Each pet (identified via pet_id) has a unique age (identified via age)
pet_id, owner_id → since	Each pet (identified via pet_id) and its owner (identified via owner_id) combined, can determine when the owner started owning that pet (identified via since)
owner_id → owner_age	Each owner (identified via owner_id) has a unique age (identified via owner_age)

   So the minimal cover is: pet_id → name, age, owner_id → owner_age, pet_id, owner_id → since

2. One of the multi-valued dependencies (MVD) we have is pet_id, name, pet_age ↠ owner_id, owner_age, since. This MVD holds because each pet is independent of its owner. Secondly, we have that pet_id, name, pet_age ↠ pet_friend_id by using the complementation rule.

3. The relation schema is not 3NF as the function dependency age_rating → child_friendly is a clear violation of 3NF. This is because it is not a trivial dependency, the attribute age_rating is not a super key and child_friendly is not part of any key. So now we decompose this schema using Decompose-3NF to put the relational schema into 3NF.

   Firstly we find the minimal cover, which should be:
   $\mathfrak{S}$ = { 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}.

The for-loop of Decompose-3NF will help construct a relational schema for each $A \rightarrow X$ in the minimal cover with attributes B such that B is in $\mathfrak{S}$'. The decomposition will be:

- We get r_1 (id, user_id, date) from "id → user_id" and "id →date"
- We get r_2 (user_id, user_location) from "user_id → user_location"
- We get r_3 (book_id, title, publisher, age_rating) from "book_id → title", "book_id → publisher" and "book_id → age_rating"
- We get r_4 (id, book_id, format, amount) from "id, book_id, format → amount"
- We get r_5 (book_id, format, isbn) from "book_id, format → isbn"
- We get r_6 (book_id, format, isbn) from "isbn → book_id" and "isbn → format"
- We get r_7 (age_rating, child_friendly) from "age_rating → child_friendly"

The if-condition of Decompose-3NF will construct a relational schema r_8 (id, book_id, format, author, category). Through the use of a while-loop, we can remove redundant relational schemas. So we can remove r_5, as it is a subset of r_6. Therefore we have the following functional dependencies that hold in each relational schema:

Relational Schema	**Functional Dependencies**
r_1 (id, user_id, date)	"id → user_id", "id →date".
r_2 (user_id, user_location)	"user_id → user_location".
r_3 (book_id, title, publisher, age_rating)	"book_id → title", "book_id → publisher", "book_id → age_rating".
r_4 (id, book_id, format, amount)	"id, book_id, format → amount".
r_6 (book_id, format, isbn)	"book_id, format → isbn".
r_7 (age_rating, child_friendly)	"isbn → book_id", "isbn → format".
r_8 (id, book_id, format, author, category)	"age_rating → child_friendly".

Since we have used the Decompose-3NF decomposition algorithm to prove the relational schema, we say that the decomposition is lossless-join and dependency-preserving. So we decompose the dataset and get the following:

r_1	id	user_id	date
	1	1	24 Dec.
	2	2	25 Dec.
	3	1	26 Dec.

r_2	user_id	user_location
	1	Hamilton
	2	Toronto

r_3	book_id	title	publisher	age_rating
	2	Book!	ThePrinter	18+
	5	Comic!	TheCopier	15+

r_4	id	book_id	format	amount
	1	2	paperback	3
	1	2	hardcover	6
	2	2	e-book	1
	3	5	paperback	1

r_6	isbn	book_id	format
	1234	2	paperback
	1237	2	hardcover
	1241	2	e-book
	1298	5	paperback

r_7	age_rating	child_friendly
	18+	no
	5+	yes

r_8	id	book_id	format	author	category
	1	2	paperback	Alicia	CS
	1	2	paperback	Alicia	Theory

	1	2	paperback	Dafni	CS
	1	2	paperback	Dafni	Theory
	1	2	hardcover	Alicia	CS
	1	2	hardcover	Alicia	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
	3	5	paperback	Bo	Comedy

4. Since we have proved that the relational schema is not in 3NF. Thus, we can also conclude that it is not in BCNF. We can show the BNCF schema using Decompose-BCNF.

   As always we start off by making the minimal cover but since we have already done it above, we can just use that. Next, we use the algorithm Decompose-BCNF to put the relational schema into BCNF.

   a. First we start off by R(Id, Ui, Ul, D, B, F, Am, Is, T, Au, P, Ar, Cf, Ca). The attribute "id" is not a key of R which means that id → X is a violation. So we split using "id → user_id, date" into R_1 = Id^+ and R_2 with remaining attributes of R including "id".
      a. R_1(Id, Ui, Ul, D) is still in violation as the user_id, and user_location are in the minimal cover. So we split it into $R_{1,1}$(Ui^+) and $R_{1,2}$
         i. $R_{1,1}$(Ui, Ul) is in BCNF as every dependency is dependent on Ui(key)
         ii. $R_{1,2}$(Id, Ui, D) is also in BCNF as every dependency is dependent on Id(key)
      b. R_2(Id, B, F, Am, Is, T, Au, P, Ar, Cf, Ca) is still in violation as the attribute "book_id" is not a key of R which means that book_id → title, publisher, age_rating is a violation. So we split R_2 into $R_{2,1}$(B^+) and $R_{2,2}$
         i. $R_{2,1}$(B, T, P, Ar, Cf) is still violating BCNF. So we split it into $R_{2,1,1}$(Ar^+) and $R_{2,1,2}$
            1. $R_{2,1,1}$(Ar, Cf) is in BCNF as every dependency is dependent on Ar(key)

2. $R_{2,1,2}$(B, T, P, Ar) is also in BCNF as every dependency is dependent on B(key)

ii. $R_{2,2}$(Id, B, F, Am, Is, Au, Ca) is still violating BCNF. So we split it into $R_{2,2,1}$(Is^+) and $R_{2,2,2}$

1. $R_{2,2,1}$(Is, B, F) is in BCNF as every dependency is dependent on Is(key)
2. $R_{2,2,2}$(Id, Am, Is, Au, Ca) is in BCNF as there are no non-trivial functional dependencies that hold

So in the end, the following functional dependencies hold in each relational schema of the resulting decomposition:

Relational Schema	**Functional Dependencies**
$R_{1,1}$(user_id, user_location)	"user_id → user_location".
$R_{1,2}$(user_id, id, date)	"id → user_id, date ".
$R_{2,1,1}$(age_rating, child_friendly)	"age_rating → child_friendly".
$R_{2,1,2}$(book_id, tile, publisher, age_rating)	"book_id → title, publisher, age_rating".
$R_{2,2,1}$(isbn, book_id, format)	"isbn → book_id", "isbn → format", "book_id, format → isbn".
$R_{2,2,2}$(id, isbn, author, category, amount)	

Since we have used the Decompose-BCNF decomposition algorithm to prove the relational schema, we say that the decomposition is lossless-join. However, since we can not derive id, book_id, format → amount from the above table, the dependency is not preserved. Now we can decompose the dataset and get the following:

$R_{1,1}$	user_id	user_location
	1	Hamilton
	2	Toronto

$R_{1,2}$	id	user_id	date
	1	1	24 Dec.
	2	2	25 Dec.
	3	1	26 Dec.

$R_{2,1,1}$	age_rating	child_friendly
	18+	no
	5+	yes

$R_{2,1,2}$	book_id	title	publisher	age_rating
	2	Book!	ThePrinter	18+
	5	Comic!	TheCopier	15+

$R_{2,2,1}$	id	book_id	format
	1	2	paperback
	1	2	hardcover
	2	2	e-book
	3	5	paperback

$R_{2,2,2}$	id	isbn	author	category	amount
	1	1234	Alicia	CS	3
	1	1234	Alicia	Theory	3
	1	1234	Dafni	CS	3
	1	1234	Dafni	Theory	3
	1	1237	Alicia	CS	6
	1	1237	Alicia	Theory	6
	1	1237	Dafni	CS	6
	1	1237	Dafni	Theory	6
	2	1241	Alicia	CS	1
	2	1241	Alicia	Theory	1
	2	1241	Dafni	CS	1

	2	1241	Dafni	Theory	1
	3	1298	Bo	Comedy	1

5.

Since we have proved that the relational schema is not in BCNF. Thus, we can also conclude that it is not in 4NF. We can show the 4NF schema using Decompose-4NF.

As always we start off by making the minimal cover but since we have already done it above, we can just use that. Next, we use the algorithm Decompose-4NF to put the relational schema into 4NF. We start by applying all splitting steps we can reuse from the BCNF decomposition:

a. The multi-dependency "id ↠ id, user_id, user_location, date, book_id, format, amount, isbn, title, author, publisher, age_rating, child_friendly, category " which we obtain by applying Replication on id → id+, is a 4NF violation. We split using this violation and obtain R_1 and R_2
   i. R_1(Id, Ui, Ul, D) is still in violation as the user_id, and user_location are in the minimal cover. So we split it into $R_{1,1}(Ui^+)$ and $R_{1,2}$
      1. $R_{1,1}$(Ui, Ul) is in 4NF as every dependency is dependent on Ui(key) and there are no multiple dependencies
      2. $R_{1,2}$(Id, Ui, D) is also in 4NF as every dependency is dependent on Id(key)and there are no multiple dependencies
   ii. R_2(Id, B, F, Am, Is, T, Au, P, Ar, Cf, Ca) is still in violation. So we split R_2 into $R_{2,1}(B^+)$ and $R_{2,2}$
      1. $R_{2,1}$(B, T, P, Ar, Cf) is still violating 4NF. So we split it into $R_{2,1,1}(Ar^+)$ and $R_{2,1,2}$
         a. $R_{2,1,1}$(Ar, Cf) is in 4NF as every dependency is dependent on Ar(key) and there are no multiple dependencies
         b. $R_{2,1,2}$(B, T, P, Ar) is also in 4NF as every dependency is dependent on B(key) and there are no multiple dependencies
      2. $R_{2,2}$(Id, B, F, Am, Is, Au, Ca) is still violating 4NF. So we split it into $R_{2,2,1}(Is^+)$ and $R_{2,2,2}$
         a. $R_{2,2,1}$(Is, B, F) is in 4NF as every dependency is dependent on Is(key) and there are no multiple dependencies
         b. $R_{2,2,2}$(Id, Am, Is, Au, Ca) is still in violation as "Am ↠ Id, Is, Am" So we split it into $R_{2,2,2,1}$ (Am^+) and $R_{2,2,2,2}$
            i. $R_{2,2,2,1}$ (Id, Is, Am) is in 4NF as every dependency is dependent on Id(key) and there are no multiple dependencies
            ii. $R_{2,2,2,2}$(Id, Is, Au, Ca) is in 4NF as every dependency is dependent on Id(key) and there are no multiple dependencies

Relational Schema	Functional Dependencies
$R_{1,1}$(user_id, user_location)	“user_id → user_location”.
$R_{1,2}$(user_id, id, date)	“id → user_id, date ”.
$R_{2,1,1}$(age_rating, child_friendly)	“age_rating → child_friendly”.
$R_{2,1,2}$(book_id, tile, publisher, age_rating)	“book_id → title, publisher, age_rating”.
$R_{2,2,1}$(isbn, book_id, format)	“isbn → book_id”, “isbn → format”, “book_id, format → isbn”.
$R_{2,2,2,1}$(id, isbn, amount)	“id, book_id, format → amount”.
$R_{2,2,2,2}$(id, isbn, author, category)	

Since we have used the Decompose-4NF decomposition algorithm to prove the relational schema, we say that the decomposition is a lossless join. However, since we can not derive everything from the above table, the dependency is not preserved. Now we can decompose the dataset and get the following:

$R_{1,1}$	user_id	user_location
	1	Hamilton
	2	Toronto

$R_{1,2}$	id	user_id	date
	1	1	24 Dec.
	2	2	25 Dec.
	3	1	26 Dec.

$R_{2,1,1}$	age_rating	child_friendly
	18+	no
	5+	yes

$R_{2,1,2}$	book_id	title	publisher	age_rating
	2	Book!	ThePrinter	18+

	5	Comic!	TheCopier	15+

$R_{2,2,1}$	id	book_id	format
	1	2	paperback
	1	2	hardcover
	2	2	e-book
	3	5	paperback

$R_{2,2,2,1}$	id	isbn	amount
	1	1234	3
	1	1237	6
	1	1241	1
	1	1298	1

$R_{2,2,2,1}$	id	isbn	author	category
	1	1234	Alicia	CS
	1	1234	Alicia	Theory
	1	1234	Dafni	CS
	1	1234	Dafni	Theory
	1	1237	Alicia	CS
	1	1237	Alicia	Theory
	1	1237	Dafni	CS
	1	1237	Dafni	Theory
	2	1241	Alicia	CS
	2	1241	Alicia	Theory
	2	1241	Dafni	CS
	2	1241	Dafni	Theory
	3	1298	Bo	Comedy

6. Not all issues are resolved: the author and category attributes do not have any dependencies on them so it is hard to derive these attributes. Hence, there could be a dependency for author and category