Normalization in Relational Database

Learning Objectives

- To Minimize data redundancy and inconsistency in a database by using normalization.

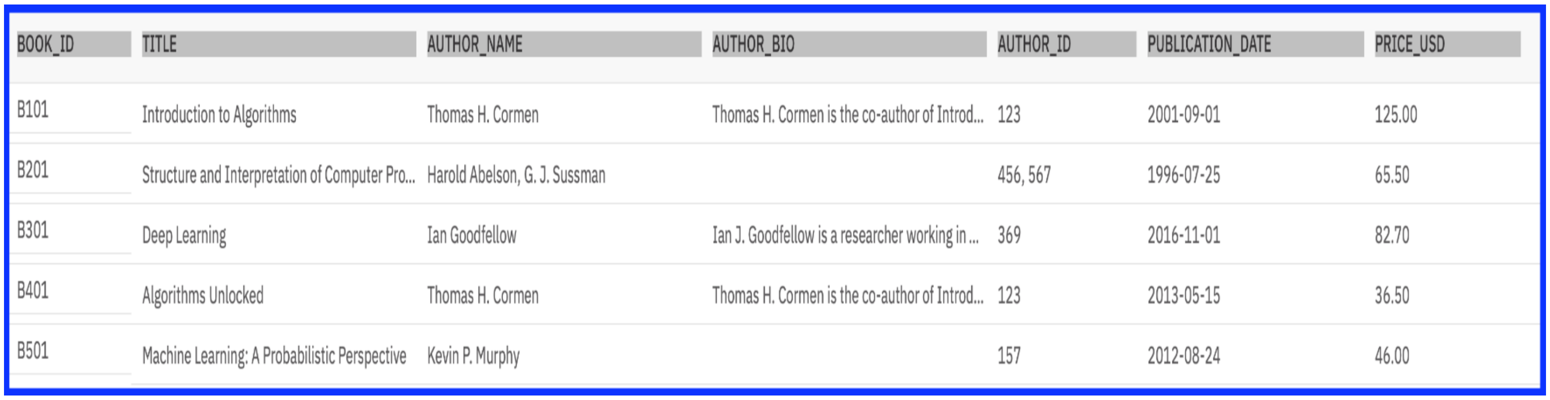
In this lab, you will learn about normalization. First, you will learn how to minimize data redundancy and inconsistency in a database by normalizing tables.

Instructions

In this exercise, you will learn about first normal form (1NF) and implement second normal form (2NF).

Task A: First normal form (1NF)

In this task of normalization, you will be working with the **BookShop** table. The following image shows the **BookShop** table:



BOOK_ID	TITLE	AUTHOR_NAME	AUTHOR_BIO	AUTHOR_ID	PUBLICATION_DATE	PRICE_USD
B101	Introduction to Algorithms	Thomas H. Cormen	Thomas H. Cormen is the co-author of Introd...	123	2001-09-01	125.00
B201	Structure and Interpretation of Computer Pro...	Harold Abelson, G. J. Sussman		456, 567	1996-07-25	65.50
B301	Deep Learning	Ian Goodfellow	Ian J. Goodfellow is a researcher working in ...	369	2016-11-01	82.70
B401	Algorithms Unlocked	Thomas H. Cormen	Thomas H. Cormen is the co-author of Introd...	123	2013-05-15	36.50
B501	Machine Learning: A Probabilistic Perspective	Kevin P. Murphy		157	2012-08-24	46.00

You will answer some questions to determine if the table above is in 1NF.

1. Does the above table have unique rows?

2. Does each cell of the above table have single/atomic value?

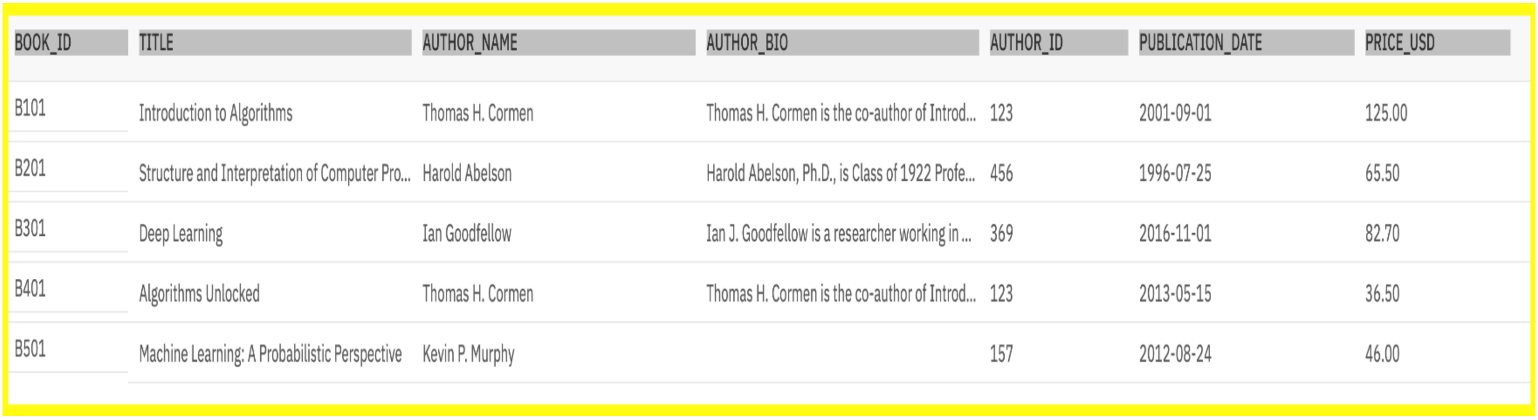
3. By definition, a table is in 1NF if every attribute in that relation contains single valued data and every tuple in that relation is unique. Does the above table fall in first normal form?

4. If your answer to question 3 is No, how can you normalize the table to ensure first normal form?

Task B: Second normal form (2NF)

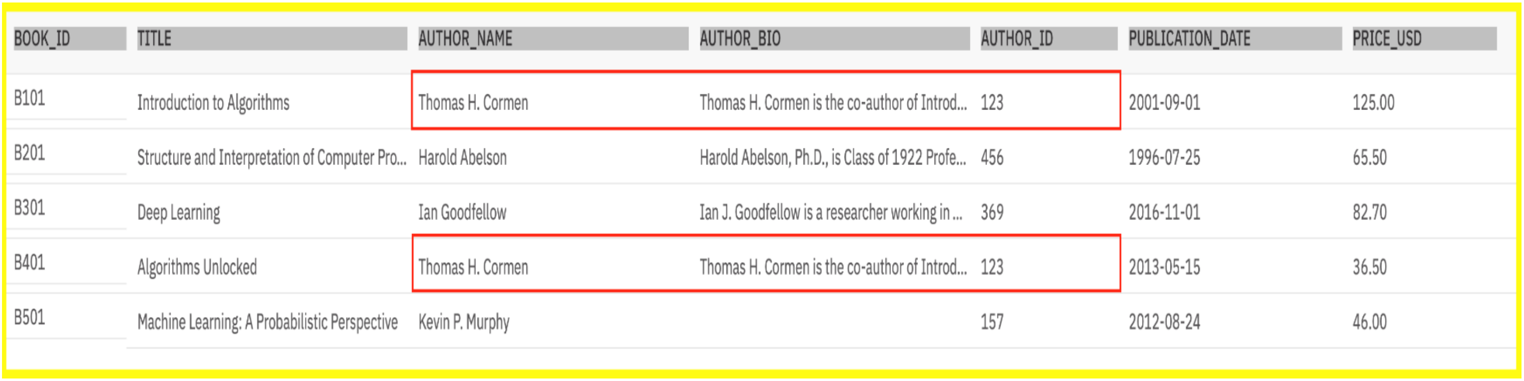
1. Starting from this task of normalization, this lab requires you to have a version of **BookShop** table (shown below). Download the BookShop-CREATE-INSERT.sql script below, upload it to the SQL server, and run it. The script will drop any previous BookShop table that exists, create the new BookShop table, and populate it with the sample data required for this lab.

- BookShop-CREATE-INSERT.sql



BOOK_ID	TITLE	AUTHOR_NAME	AUTHOR_BIO	AUTHOR_ID	PUBLICATION_DATE	PRICE_USD
B101	Introduction to Algorithms	Thomas H. Cormen	Thomas H. Cormen is the co-author of Introd...	123	2001-09-01	125.00
B201	Structure and Interpretation of Computer Pro...	Harold Abelson	Harold Abelson, Ph.D., is Class of 1922 Profe...	456	1996-07-25	65.50
B301	Deep Learning	Ian Goodfellow	Ian J. Goodfellow is a researcher working in ...	369	2016-11-01	82.70
B401	Algorithms Unlocked	Thomas H. Cormen	Thomas H. Cormen is the co-author of Introd...	123	2013-05-15	36.50
B501	Machine Learning: A Probabilistic Perspective	Kevin P. Murphy		157	2012-08-24	46.00

2. By definition, a relation is in second normal form if it is already in 1NF and does not contain any partial dependencies. If you look at the BookShop table, you will find every column in the table is single or atomic valued, but it has multiple books by the same author.



BOOK_ID	TITLE	AUTHOR_NAME	AUTHOR_BIO	AUTHOR_ID	PUBLICATION_DATE	PRICE_USD
B101	Introduction to Algorithms	Thomas H. Cormen	Thomas H. Cormen is the co-author of Introd...	123	2001-09-01	125.00
B201	Structure and Interpretation of Computer Pro...	Harold Abelson	Harold Abelson, Ph.D., is Class of 1922 Profe...	456	1996-07-25	65.50
B301	Deep Learning	Ian Goodfellow	Ian J. Goodfellow is a researcher working in ...	369	2016-11-01	82.70
B401	Algorithms Unlocked	Thomas H. Cormen	Thomas H. Cormen is the co-author of Introd...	123	2013-05-15	36.50
B501	Machine Learning: A Probabilistic Perspective	Kevin P. Murphy		157	2012-08-24	46.00

This means that the AUTHOR_ID, AUTHOR_NAME and AUTHOR_BIO details for BOOK_ID B101 and B401 are the same. As the number of rows in the table increase, you will be needlessly storing more and more occurrences of these same pieces of information. And if an author updates their bio, you must update all of these occurrences.

3. In this scenario, to enforce 2NF you can take the author information such as AUTHOR_ID, AUTHOR_NAME and AUTHOR_BIO out of the BookShop table into another table, for example a table named **BookShop_AuthorDetails**. You then link each book in the BookShop table to the relevant row in the BookShop_AuthorDetails table, using a unique common column such as AUTHOR_ID to link the tables. To create the new **BookShop_AuthorDetails** table, copy the code below and paste it to the textbox of the

```
CREATE TABLE BookShop_AuthorDetails
AS
(SELECT DISTINCT AUTHOR_ID, AUTHOR_NAME, AUTHOR_BIO FROM BookShop)
WITH DATA;

SELECT * FROM BookShop_AuthorDetails;
```

Run SQL page after adding a new blank script. Click **Execute**.

```
CREATE TABLE BookShop_AuthorDetails
AS
(SELECT DISTINCT AUTHOR_ID, AUTHOR_NAME, AUTHOR_BIO FROM
BookShop)
WITH DATA
```

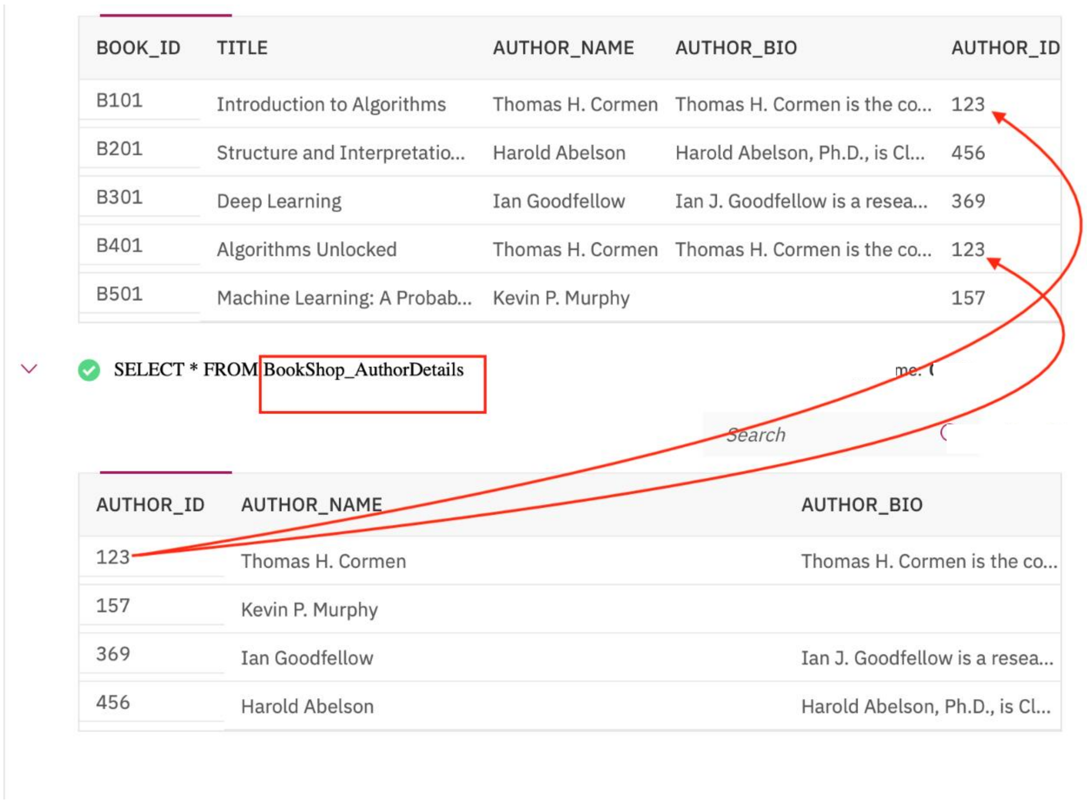
If this query doesn't run, create a table manually and put data from this query.

```
INSERT INTO BookShop_AuthorDetails (AUTHOR_NAME, AUTHOR_BIO, AUTHOR_ID) (SELECT
AUTHOR_NAME, AUTHOR_BIO, AUTHOR_ID FROM BookShop)
```

4. Now you can drop the redundant author information related columns from the BookShop table. Do not drop the AUTHOR_ID column though, because that will be used to maintain the relationship between the two tables. To drop the redundant author information related columns from the BookShop table, copy the code below and paste it to the current script textbox of the **Run SQL** page replacing the existing code with this new code. Click **Execute.**

```
ALTER TABLE BookShop
DROP COLUMN AUTHOR_NAME
ALTER TABLE BookShop
DROP COLUMN AUTHOR_BIO;
```

5. Now you are only storing the author information once per author and only have to update it in one place: reducing redundancy and increasing consistency of data. Thus, 2NF is ensured.



BOOK_ID	TITLE	AUTHOR_NAME	AUTHOR_BIO	AUTHOR_ID
B101	Introduction to Algorithms	Thomas H. Cormen	Thomas H. Cormen is the co...	123
B201	Structure and Interpretatio...	Harold Abelson	Harold Abelson, Ph.D., is Cl...	456
B301	Deep Learning	Ian Goodfellow	Ian J. Goodfellow is a resea...	369
B401	Algorithms Unlocked	Thomas H. Cormen	Thomas H. Cormen is the co...	123
B501	Machine Learning: A Probab...	Kevin P. Murphy		157

```
SELECT * FROM BookShop_AuthorDetails
```

AUTHOR_ID	AUTHOR_NAME	AUTHOR_BIO
123	Thomas H. Cormen	Thomas H. Cormen is the co...
157	Kevin P. Murphy	
369	Ian Goodfellow	Ian J. Goodfellow is a resea...
456	Harold Abelson	Harold Abelson, Ph.D., is Cl...

Task C: Third normal form (3NF)

Third Normal Form (3NF) is a database normalization technique that reduces data redundancy and inconsistencies by eliminating transitive dependencies. It ensures that each column in a table depends only on the primary key and not on any non-key attributes.

In simple terms, a table is in 3NF if every non-key column is functionally dependent on the primary key, and there are no transitive dependencies.
To achieve 3NF in SQL Server, you can follow these steps:

1- Identify the primary key(s) of the table

2- Identify any non-key columns that depend on only part of the primary key. *These are partial dependencies and should be moved to a separate table*
3- Identify any non-key columns that depend on other non-key columns. *These are transitive dependencies and should be moved to a separate table*
4- Create new tables for the partial and transitive dependencies
5- Create foreign key constraints to link the new tables to the original table
6- Remove the redundant columns from the original table.

Let’s extend our previous example to a table like this:

BOOK _ID	TITLE	CATEGORY	AUTHOR _ID	PUBLICATION _DATE	PRICE_USD	PUBLISHER_NAME
B101	Introduction to Algorithms	CS&E	1	09-03-2008	09$	MIT Press
B201	Structure and Interpretation	ME	2	18-11-2020	12$	HarperCollins
B301	Deep Learning	AI	4	02-02-2022	10$	Penguin
B401	Algorithms Unlocked	CS&E	1	18-05-2012	10$	MIT Press
B501	Machine Learning	AI	4	05-08-2019	08$	Mastery Publishers

In this example, the BOOK_ID is the primary key, and there is no partial dependency on any column, so the table satisfies the 2NF requirements.

However, there is a transitive dependency between the **publisher** columns and **category** since the category depends on the publisher. This means that the table does not satisfy the 3NF requirements.

To bring this table into 3NF, we can separate the publisher and category columns into a separate table, like this:

BOOK _ID	TITLE	AUTHOR _ID	PRICE _USD	PUBLICATION_ DATE	PUBLISHER_ID
B101	Introduction to Algorithms	1	09$	09-03-2008	1
B201	Structure and Interpretation	2	12$	18-11-2020	2
B301	Deep Learning	4	10$	02-02-2022	3
B401	Algorithms Unlocked	1	10$	18-05-2012	1
B501	Machine Learning	4	08$	05-08-2019	4

PUBLISHER _ID	CATEGORY	PUBLISHER_NAME
1	CS&E	MIT Press
2	ME	HarperCollins
3	AI	Penguin
4	AI	Mastery Publishers

Summary:

1nf	**2nf**	**3nf**
Identify and remove multiple entries in a column	Identify and remove redundant entries in a table, by creating another table and linking both by Pk-Fk	Identify and remove entries that depend on non-key columns of a table, by creating another table.
Each cell in the table must contain only a single value	No Partial Dependency	No Transitive Dependency
	PD: Non-key attributes depending on a part of Primary Key (Pk is a composite PK)	TD: Non-key attributes depend on non-key attributes.

Lab Tasks:

1- Create a table with redundant data and normalize it to 3NF:

Create a table named ***orders*** with columns **order_id, customer_name, customer_email, customer_address, product_name, product_description, product_price,** and **quantity**.

Insert some sample data into the table, making sure to ***include redundant data***. Normalize the table to third normal form by creating additional tables for customers and products and linking them to the orders table with foreign keys.

2- Create a table with partial dependencies and normalize it to 3NF:

Create a table named ***employees*** with columns **employee_id, employee_name, employee_address, department_name, department_location,** and **manager_name.**

Insert some sample data into the table, making sure to include ***partial dependencies***. Normalize the table to second normal form by creating additional tables for departments and managers and linking them to the employees table with foreign keys.

3- Create a table with transitive dependencies and normalize it to 3NF:

Create a table named ***courses*** with columns **course_id, course_name, department_name,** and **instructor_name.**

Insert some sample data into the table, making sure to include ***transitive dependencies***. Normalize the table to third normal form by creating additional tables for departments and instructors and linking them to the courses table with foreign keys.