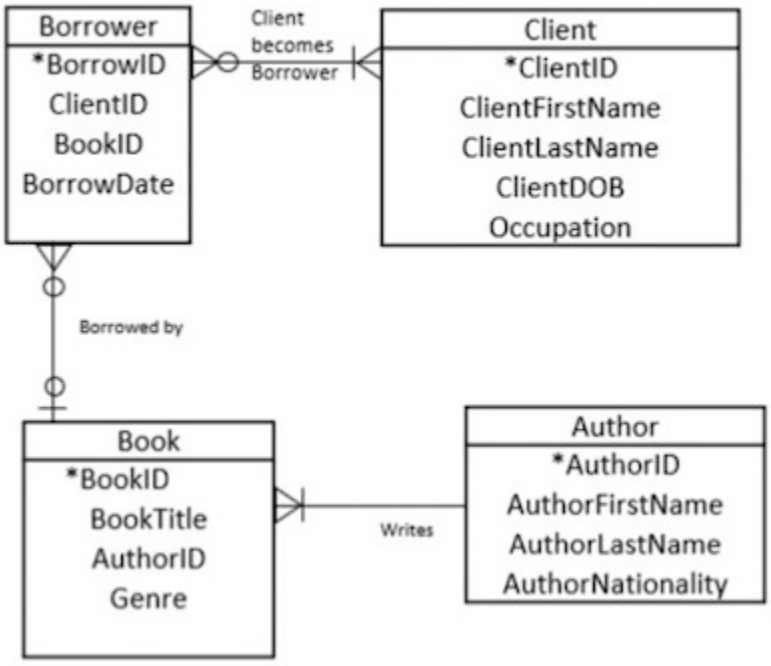
DataBase Fundamentals Assignment - Database Evaluation Essay

Prompt 1: Given the following sample data, you will see that certain tables are linked via primary and foreign keys (e.g., Client and Borrower: ClientId is on both tables).

Write the following SQL statements to retrieve data:

- Select all clients who borrowed books
- Select all books borrowed by borrowers, order by borrow date
- Select all books and include the author's first and last name
- Insert a new client with an occupation of pilot



Database Fundamentals Assignment

1: SELECT DISTINCT Client.ClientId, Client.FirstName, Client.LastName

FROM Client JOIN

Borrower ON Client.ClientId = Borrower.ClientId

JOIN BorrowedBooks ON Borrower.BorrowerId = BorrowedBooks.BorrowerId;

This will gather all the clients who borrowed books by joining the Client, Borrower, and Borrowed Books table

2: SELECT Books.BookId, Books.Title, BorrowedBooks.BorrowDate

FROM Books

JOIN BorrowedBooks ON Books.BookId = BorrowedBooks.BookId

ORDER BY BorrowedBooks.BorrowDate;

This will gather all the books borrowed by borrowers. Adding the Orderby will have it displayed by borrow date

3: SELECT Books.BookId, Books.Title, Author.FirstName, Author.LastName

FROM Books
JOIN Author ON Books.AuthorId = Author.AuthorId;

This will select all the books and the authors full names by joining the Books and Authors tables

4.INSERT INTO Client (ClientId, FirstName, LastName, Occupation)

VALUES ('18282', 'Steve', 'Grey', 'Pilot');

This will take the values in the order above and place the data in the correct tables.

Database Fundamentals Assignment

Prompt 2: Write a brief 500-750 word essay that evaluates the miniature database below. Currently, the database is not in third normal form (3NF). Your analysis should identify the dependencies that exist in the current database setup and explain the steps needed to transform it to third normal form.

Let's begin by defining 3NF. “Third normal form is a key concept of database normalization that removes unwanted dependencies. 3NF builds upon first normal form (1NF) and second normal form (2NF), meaning that it inherits their rules: 1NF requires atomic (indivisible) values in each cell, and 2NF removes partial dependencies on a composite primary key. 3NF takes it further by removing transitive dependencies, a situation where non-key attributes depend indirectly on the primary key.” (Fayard). So what steps do we need to take to remove redundancy in data?

First, check tables for dependencies:

Table Structure for Produce

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID	CHAR (5)	No
SUPPLIERID	CHAR (10)	No
PLUCODE	CHAR (4,2)	No
PRODUCENAME	CHAR (15)	No
TYPE	CHAR (10)	No
STOCKQTY	NUMBER (4,2)	No
NXTDELIVERY	DATE	No

The issue that sticks out here is the SupplierID. Many items can come from the same supplier, which would give us redundant data. To fix this issue, the supplier data will need its own table.

Table Structure for Animal Products:

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID	CHAR (5)	No
SUPPLIERID	CHAR (10)	No
ANPRDNAME	CHAR (15)	No
TYPE	CHAR (10)	No
STOCKQTY	NUMBER(4,2)	No
NXTDELIVERY	DATE	Yes

The Animal Produce table appears to have the same issue. The SupplierID could be redundant. The way to solve this is to have its own table

Table Structure for Grains:

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID	CHAR (5)	No
SUPPLIERID	CHAR (10)	No
GRAINNAME	CHAR (15)	No
TYPE	CHAR (10)	No
STOCKQTY	NUMBER(4,2)	No
NXTDELIVERY	DATE	Yes

Once again, same issue. It could also be worthwhile to create a table to expand on NXTDELIVERY, as this is too vague. If the system only tracks the next delivery date, there's no historical record of previous deliveries or changes in stock quantities over time. This could be

problematic. However, for the scope of the assignment, this may not be necessary to do as I believe it's in 3NF without this step.

Table Structure of Suppliers:

COLUMN_NAME	DATA_TYPE	NULLABLE
SUPPLIERID	CHAR (10)	No
LASTDELIVERY	DATE	Yes
SPECIALTY	CHAR (15)	Yes
ACTIVE	CHAR (1)	No

This table seems to be ok and well put together. There's no information that is redundant or needlessly put on the table. However, the SUPPLIERID must be made into a foreign key in the tables above.

Table Structure of Purchases:

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID	CHAR (5)	No
TOTALBOUGHT	NUMBER (8,2)	Yes
TOTALSOLD	NUMBER(8,2)	Yes
TOTALREV	NUMBER(10,2)	Yes
MARGIN	NUMBER(10,2)	Yes

The Purchases table has potential issues, especially with redundant columns for the total bought, sold, revenue, and margin for different types of products. To normalize, we could split this into separate tables for better structure, where each product type would reference its own purchase records. However, this would just be to make it more robust and might fall out of the scope of

this assignment.

The tables above aren't too far out of reach when it comes to converting to 3NF. With a few changes, many potential data redundancies can be fixed. In the next section, the database will be recreated in order to show the database in 3NF.

Prompt 3: Now that you have identified the steps necessary to bring this database to third normal form (3NF), you will create a diagram that displays the new layout and the relationships between the tables. It should depict how you successfully removed dependencies in the tables to create a relational database model.

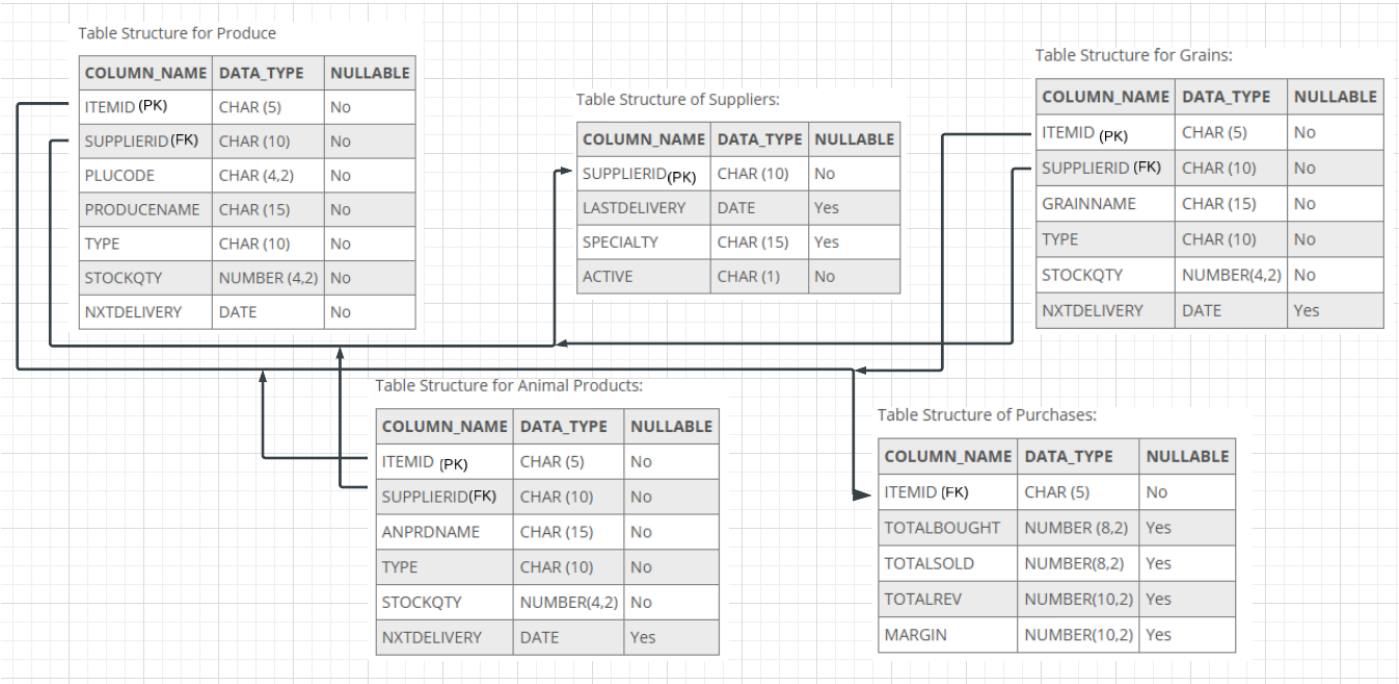


Table Structure for Produce

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID (PK)	CHAR (5)	No
SUPPLIERID (FK)	CHAR (10)	No
PLUCODE	CHAR (4,2)	No
PRODUCENAME	CHAR (15)	No
TYPE	CHAR (10)	No
STOCKQTY	NUMBER (4,2)	No
NXTDELIVERY	DATE	No

Table Structure of Suppliers:

COLUMN_NAME	DATA_TYPE	NULLABLE
SUPPLIERID(PK)	CHAR (10)	No
LASTDELIVERY	DATE	Yes
SPECIALTY	CHAR (15)	Yes
ACTIVE	CHAR (1)	No

Table Structure for Grains:

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID (PK)	CHAR (5)	No
SUPPLIERID (FK)	CHAR (10)	No
GRAINNAME	CHAR (15)	No
TYPE	CHAR (10)	No
STOCKQTY	NUMBER(4,2)	No
NXTDELIVERY	DATE	Yes

Table Structure for Animal Products:

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID (PK)	CHAR (5)	No
SUPPLIERID(FK)	CHAR (10)	No
ANPRDNAME	CHAR (15)	No
TYPE	CHAR (10)	No
STOCKQTY	NUMBER(4,2)	No
NXTDELIVERY	DATE	Yes

Table Structure of Purchases:

COLUMN_NAME	DATA_TYPE	NULLABLE
ITEMID (FK)	CHAR (5)	No
TOTALBOUGHT	NUMBER (8,2)	Yes
TOTALSOLD	NUMBER(8,2)	Yes
TOTALREV	NUMBER(10,2)	Yes
MARGIN	NUMBER(10,2)	Yes

This Database has the proper PKs and FKs. Produce, Animal Products, and Grains tables all reference SUPPLIERID from the Suppliers table (One-to-Many relationship) as there can be many purchases. This should fix the redundancies and put it into 3NF form.

Bibliography

Fayard, Marie. “What Is Third Normal Form (3NF)? A Beginner-Friendly Guide.” *DataCamp*, DataCamp, 18 Nov. 2024, www.datacamp.com/tutorial/third-normal-form.

Gibbs, Martin. “Third Normal Form in DBMS with Examples.” *Study.Com*, study.com/academy/lesson/third-normal-form-in-dbms-with-examples.html.