Analysis, Query and Normalization of Entity Relation Diagrams

Nicholas D. Mensah

Western Governors University

Computer Science 107: Database Fundamentals

Study.com

8/13/2022

Author Note *(*

gitHub: https://github.com/mensahTribeWeb/DB_Fundamentals_Com_Sci_107.git

)

Abstract

This analysis of Entity Relationship Diagrams will analyze schemas to determine their Normalization. I will determine dependencies of attributes of each table, and determine their connections to other tables. I identify steps to remove redundant material which will limit the optimization of the table's design. We will walk through the design to produce designs that are in third normal form. Before Normalization of the tables I will Query existing tables to produce datasets. Material was utilized from my public library project in order to maximize on time and to actually produce results from the queries requested.

Analysis, Query and Normalization of Entity Relation Diagrams

Part 1 - Retrieve Data From a Database. Will utilize Structured query language to demonstrate the ability to query Data. while following the Create Retrieval Update Delete processes which are known as C.R.U.D allow knowledge seekers to utilize data. ERDs assist in minimizing the confusion of disjointed tables which allows the SQL to be utilized effectively. The syntax of SQL reads like english which guides me through relational databases. SQL comes with to sasic clauses of SELECT and FROM to specifically target tables to query. When the SQL statements of at least two clauses are used, accessing a set of data becomes easier.

Part 1 - Retrieve Data From a Database. The initial query requests produce a result set of all columns(fields) from the Borrower table. Second, I will add a title instead of a wildcard to select all books from the book table, which will be joined with the borrower table using the bookid. Third, the first name and last name of the author is queried from the author table which is aliased with an “a” . an inner join JOINS book to the author table using bookid and book author. Finally, a record that contains a pilot will be inserted into the client table.Therefore, all queries are successfully producing the requested result-sets.

Part 1 - Retrieve Data From a Database

Creating DataBase & Tables (optional)

```sql
---Create Database
CREATE DATABASE IF NOT EXISTS libraryDB ;

--Create Tables
--Barrower TABLE
CREATE TABLE Borrower
(
    BorrowID SERIAL PRIMARY KEY,
    ClientID INTEGER,
    BookID INTEGER,
    BorrowDate Date,
    FOREIGN KEY(ClientID) REFERENCES Client(ClientID),
    FOREIGN KEY(BookID) REFERENCES BooK(BookID)
);

--Client TABLE
CREATE TABLE Client
(
    ClientID SERIAL PRIMARY KEY,
    ClientFirstName VARCHAR(30),
    ClientLastName VARCHAR(30),
    ClientDOB INTEGER,
    Occupation VARCHAR(50)
);
```

```sql
--Book TABLE
CREATE TABLE Book
(
    BookID SERIAL PRIMARY KEY,
    BookTitle VARCHAR(50),
    BookAuthor INTEGER,
    Genre VARCHAR(30),
    FOREIGN KEY(BookAuthor) REFERENCES Author(AuthorID)
);


--Author TABLE
CREATE TABLE Author
(
    AuthorID SERIAL PRIMARY KEY,
    AuthorFirstName VARCHAR(30),
    AuthorLastName VARCHAR(30),
    AuthorNationality VARCHAR(50));
```

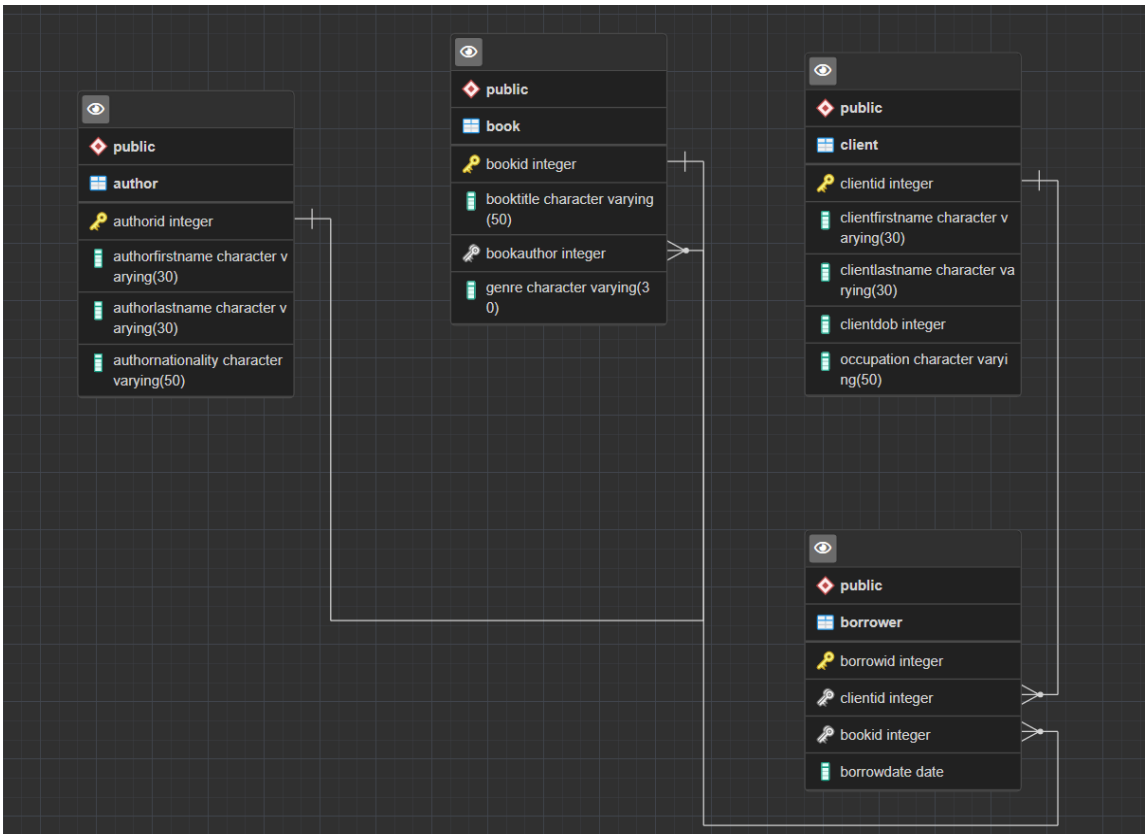
Populating TABLES

```
ulate Borrower table:

INSERT INTO Borrower (BorrowID, ClientID,
                      BookID, BorrowDate)
VALUES
    (1, 35, 17,   '2016-07-20'),
    (2, 1,   3,   '2016-07-21'),
    (3, 42,  8,   '2016-07-22'),
    (4, 62, 16,   '2016-07-23'),
    (5, 53, 13,   '2016-07-24'),
    (6, 33, 15,   '2016-07-25'),
    (7, 40, 14,   '2016-07-26'),
    (8, 64,  2,   '2016-07-27'),
    (9, 56, 30,   '2016-07-28'),
    (10, 23, 2,   '2016-07-29'),
    (11, 46, 19,  '2016-07-30'),
    (12, 61, 20,  '2016-07-31'),
    (13, 58,  7,  '2016-08-01'),
    (14, 46, 16,  '2016-08-02'),
    (15, 80, 21,  '2016-08-03'),
    (16, 51, 23,  '2016-08-04'),
    (17, 49, 18,  '2016-08-05'),
    (18, 43, 18,  '2016-08-06'),
    (19, 30,  2,  '2016-08-07'),
    (20, 48, 24,  '2016-08-08'),
    (21, 71,  5,  '2016-08-09'),
    (22, 35,  3,  '2016-08-10'),
    (23, 57,  1,  '2016-08-11'),
    (24, 23, 25,  '2016-08-12'),
    (25, 20, 12,  '2016-08-13'),
    (26, 25,  7,  '2016-08-14'),
    (27, 72, 29,  '2016-08-15'),
    (28, 74, 20,  '2016-08-16'),
    (29, 53, 14,  '2016-08-17'),
    (30, 32, 10,  '2016-08-18') ;
```

```
INSERT INTO Book (BookID, BookTitle, BookAuthor, Genre)
VALUES
    (1, 'Build your database system', 1, 'Science'),
    (2, 'The red wall', 2, 'Fiction'),
    (3, 'The perfect match', 3, 'Fiction'),
    (4, 'Digital Logic', 4, 'Science'),
    (5, 'How to be a great lawyer', 5, 'Law'),
    (6, 'Manage successful negotiations', 6, 'Society'),
    (7, 'Pollution today', 7, 'Science'),
    (8, 'A gray park', 2, 'Fiction'),
    (9, 'How to be rich in one year', 8, 'Humor'),
    (10, 'Their bright fate', 9, 'Fiction');

-- Populate Client table:

INSERT INTO Client(ClientID, ClientFirstName,
                   ClientLastName, Occupation, ClientDOB)
VALUES
    (1, 'Kaiden', 'Hill', 'Student', 2006),
    (2, 'Alina', 'Morton','Student', 2010),
    (3, 'Fania', 'Brooks', 'Food Scientist', 1983),
    (4, 'Courtney', 'Jensen', 'Student', 2006),
    (5, 'Brittany', 'Hill', 'Firefighter', 1983),
    (6, 'Max', 'Rogers', 'Student', 2005),
    (7, 'Margaret', 'McCarthy', 'School Psychologist', 1981),
    (8, 'Julie', 'McCarthy', 'Professor', 1973),
    (9, 'Ken', 'McCarthy', 'Securities Clerk', 1974),
    (10, 'Britany', 'O Quinn', 'Violinist', 1984),
    (11, 'Conner', 'Gardner', 'Licensed Massage Therapist', 1998),
    (12, 'Mya', 'Austin', 'Parquet Floor Layer', 1960),
    (13, 'Thierry', 'Rogers', 'Student', 2004),
    (14, 'Eloise', 'Rogers', 'Computer Security Manager', 1984),
    (15, 'Gerard', 'Jackson', 'Oil Exploration Engineer', 1979),
    (16, 'Randy', 'Day', 'Aircraft Electrician', 1986);

--populate Author TABLE
```

```sql
--populate Author TABLE

INSERT INTO Author (AuthorID, AuthorFirstName,
                    AuthorLastName, AuthorNationality)
VALUES
     (1,'Sofia','Smith','Canada'),
     (2, 'Maria','Brown','Brazil'),
     (3, 'Elena','Martin','Mexico'),
     (4, 'Zoe', 'Roy','France'),
     (5, 'Sebastian','Lavoie','Canada'),
     (6, 'Dylan', 'Garcia', 'Spain'),
     (7, 'Ian', 'Cruz', 'Mexico'),
     (8, 'Lucas', 'Smith', 'USA'),
     (9, 'Fabian', 'Wilson', 'USA'),
     (10, 'Liam', 'Taylor', 'Canada'),
     (11, 'William', 'Thomas', 'Great Britain'),
     (12, 'Logan', 'Moore', 'Canada');
```

libraryDB ERD



Write the following SQL statements to retrieve data:

```sql
--Write the following SQL statements to retrieve data:

-- Select all borrowers
  SELECT *
  FROM Borrower;

-- Select all books borrowed by borrowers, order by borrow date
SELECT *
FROM book
JOIN Borrower USING(bookID)
ORDER BY borrowID;

-- Select all books and include the author first and last name
SELECT AuthorFirstName, AuthorLastName
FROM AUTHOR AS a
INNER JOIN book AS b ON a.AuthorID = b.bookAuthor;

-- Insert a new client with an occupation of pilot
INSERT INTO client
VALUES(1, 'Kaiden', 'Hill', 'piolt', 2006);
```

Part Two - Database Normalization . Allows me to analyze Data to determine if the material is in normal form and what that and how I will arrive at that conclusion. Normalization of the data allows the redundancies to be removed to optimize the data that will be present in the tables. We will use three methodologies to determine normalcy, the first normal form, second normal form, and third normal form. When I first look to normalize a database base I will initially utilize the first Normal Form(1NF) to reduce redundancy. I will determine if each table

has a primary key which usually is an id that is unique to that table. Next, fields must have unique names in order to reduce repeating fields. To elaborate, fields will not store similar data throughout any column. Third , data will not be repeated across fields and there will result in no redundant data. Next, will determine if the table is in Second Normal Form (2NF) by ensuring that each table server's one purpose and will be connected to a singular primary key and possibly joins others tables with a foreign key. A table will be in third normal form when it meets requirements of 1NF and 2NF. The tables only have columns that will not depend on each other in a way if it was removed the table will fail, this would mean the column would have a transitive dependency. All the fields will be dependent on the primary key. Therefore, these are the procedures I will use to analyze the table to determine normalcy.

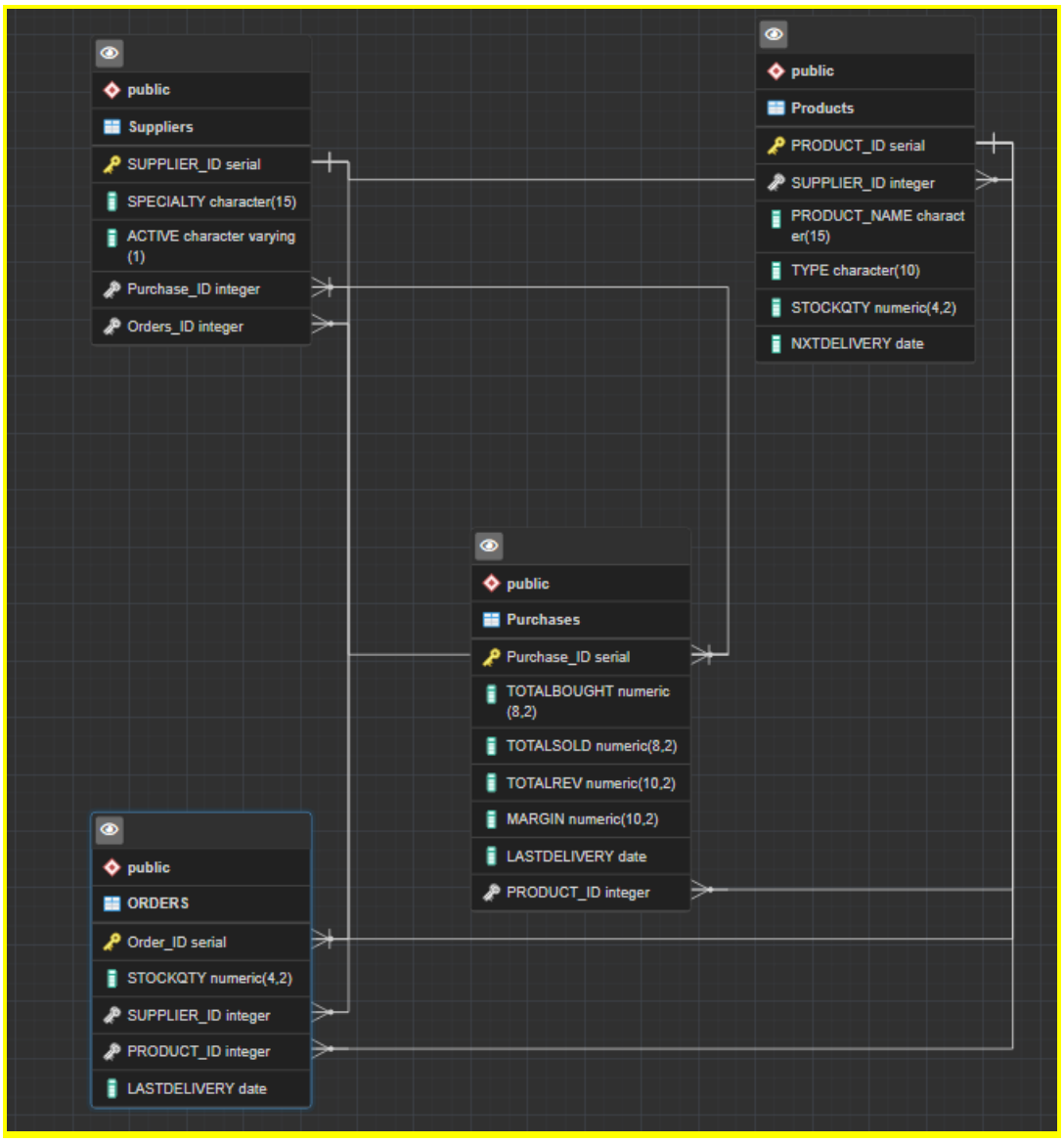
```
BEGIN;


CREATE TABLE IF NOT EXISTS public."Produce"
(
    "ITEMID" character varying(5) NOT NULL,
    "SUPPLIERID" character varying(10) NOT NULL,
    "PLUCODE" numeric(4, 2) NOT NULL,
    "PRODUCENAME" character varying(15) NOT NULL DEFAULT 15,
    "TYPE" character varying(10) NOT NULL DEFAULT 10,
    "STOCKQTY" numeric(4, 2) NOT NULL,
    "NXTDELIVERY" date NOT NULL,
    PRIMARY KEY ("ITEMID")
);

CREATE TABLE IF NOT EXISTS public."Animal Products"
(
    "ITEMID" character varying(5) NOT NULL,
    "SUPPLIERID" character varying(10)[] NOT NULL,
    "ANPRDNAME" character varying(15) NOT NULL,
    "TYPE" character varying(10)[] NOT NULL,
    "STOCKQTY" numeric(4, 2) NOT NULL,
    "NXTDELIVERY" date,
    PRIMARY KEY ("ITEMID")
);

CREATE TABLE IF NOT EXISTS public."Grains"
(
    "ITEMID" character varying(5) NOT NULL,
    "SUPPLIERID" character(10) NOT NULL,
    "GRAINNAME" character(15) NOT NULL,
    "TYPE" character(10) NOT NULL,
    "STOCKQTY" numeric(4, 2) NOT NULL,
    "NXTDELIVERY" date,
    PRIMARY KEY ("ITEMID")
);

CREATE TABLE IF NOT EXISTS public."Suppliers"
(
    "SUPPLIERID" character varying(10),
    "LASTDELIVERY" date NOT NULL,
    "SPECIALTY" character(15) NOT NULL,
    "ACTIVE" character varying(1),
    PRIMARY KEY ("SUPPLIERID")
);

CREATE TABLE IF NOT EXISTS public."Purchases"
(
    "ITEMID" character varying(5),
    "TOTALBOUGHT" numeric(8, 2) NOT NULL,
    "TOTALSOLD" numeric(8, 2) NOT NULL,
    "TOTALREV" numeric(10, 2) NOT NULL,
    "MARGIN" numeric(10, 2) NOT NULL,
    PRIMARY KEY ("ITEMID")
);
END;
```


public
Suppliers
SUPPLIERID character varying(10)
LASTDELIVERY date
SPECIALTY character(15)
ACTIVE character varying(1)
public
Animal Products
ITEMID character varying(5)
SUPPLIERID character varying[](10)
ANPRDNAME character varying(15)
TYPE character varying[](10)
STOCKQTY numeric(4,2)
NXTDELIVERY date
public
Produce
ITEMID character varying(5)
SUPPLIERID character varying(10)
PLUCODE numeric(4,2)
PRODUCENAME character varying(15)
TYPE character varying(10)
STOCKQTY numeric(4,2)
NXTDELIVERY date
public
Grains
ITEMID character varying(5)
SUPPLIERID character(10)
GRAINNAME character(15)
TYPE character(10)
STOCKQTY numeric(4,2)
NXTDELIVERY date
public
Purchases
ITEMID character varying(5)
TOTALBOUGHT numeric(8,2)
TOTALSOLD numeric(8,2)
TOTALREV numeric(10,2)
MARGIN numeric(10,2)

Part Two - Database Normalization. Upon initial ERD analysis I wanted to determine if the database was In 1NF. The names were not unique, based on analyzing id they all were some type of item id which was ambiguous. Next, I checked if any of the columns needed to be broken down any further so this was fine. The Product table has redundant data, the plucode, the id can help distinguish the row of data. But the plucode will assist on the store level.

Part three - Develop an ERD- Entity Relationship Diagram. We will develop an ERD which meets the standard to be in Third Normal form. To determine if the table stores information about a single topic which depends on the id of the table. When the first normal form and second normal form has been established removal of transitive properties will Normalize and optimize the final ERD for this project.

Part three - Develop an ERD- Entity Relationship Diagram. The item names are given unique names like ANPRD_ID, SUPPLIER_ID, PROD_ID, PURCHASE_ID, GRAIN_ID, SUPPLIER_ID this will assist in establishing primary keys and foreign keys. To elaborate, two conditions are met: each TABLE has a primary key and each field has unique names. Each table now has a plu code.I created an associate table name order to allow the other tables to have singular purposes and to remove transient dependencies. Also I just use one product table the varieties of products will be established by type and primary keys.



Results

Outcome 1

I will now discuss the result of querying a relational database while utilizing SQL. Next, I will address the solution to the anomalies of the data table, which resulted in the tables to be In 3NF.

Outcome 2

Reduced clutter and redundancies by allowing each table to serve a purpose.each record within a table will depend on the primary key.

Outcome 3

The ERD is produced in 3NF, by reducing redundancy and assuring 1NF and 2NF are satisfied. Removal of transient properties. Establishing primary and foreign keys.

Discussion

I have Normalized the Third ERD by assuring that the data is in 1NF and 2NF before ensuring it was In 3NF. Tables have clear and concise fields which allows for optimized queries. The queries have resulted in result sets and utilizes the language of SQL. I analyzed the initial data because it was not in 3NF. I used records from my previous assignment to save on time producing new data when data was established. I did not believe this would cause any issues being that this section was optional. This project helped build on my foundation of SQL and ERDs. This project assisted with understanding Normalcy a bit better when analyzing and populating tables with data.

References

Basic SQL Query Syntax. (2016, October 13). Retrieved from https://study.com/academy/lesson/basic-sql-query-syntax.html.

First Normal Form in DBMS with Examples. (2020, March 16). Retrieved from https://study.com/academy/lesson/first-normal-form-in-dbms-with-examples.html.

Third Normal Form in DBMS with Examples. (2020, March 14). Retrieved from https://study.com/academy/lesson/third-normal-form-in-dbms-with-examples.html.

Designing a Relational Database: Tutorial & Overview. (2020, March 17). Retrieved from https://study.com/academy/lesson/designing-a-relational-database-tutorial-overview.html.

Relationships in SQL: Examples & Explanation. (2020, August 13). Retrieved from https://study.com/academy/lesson/relationships-in-sql-examples-explanation.html.