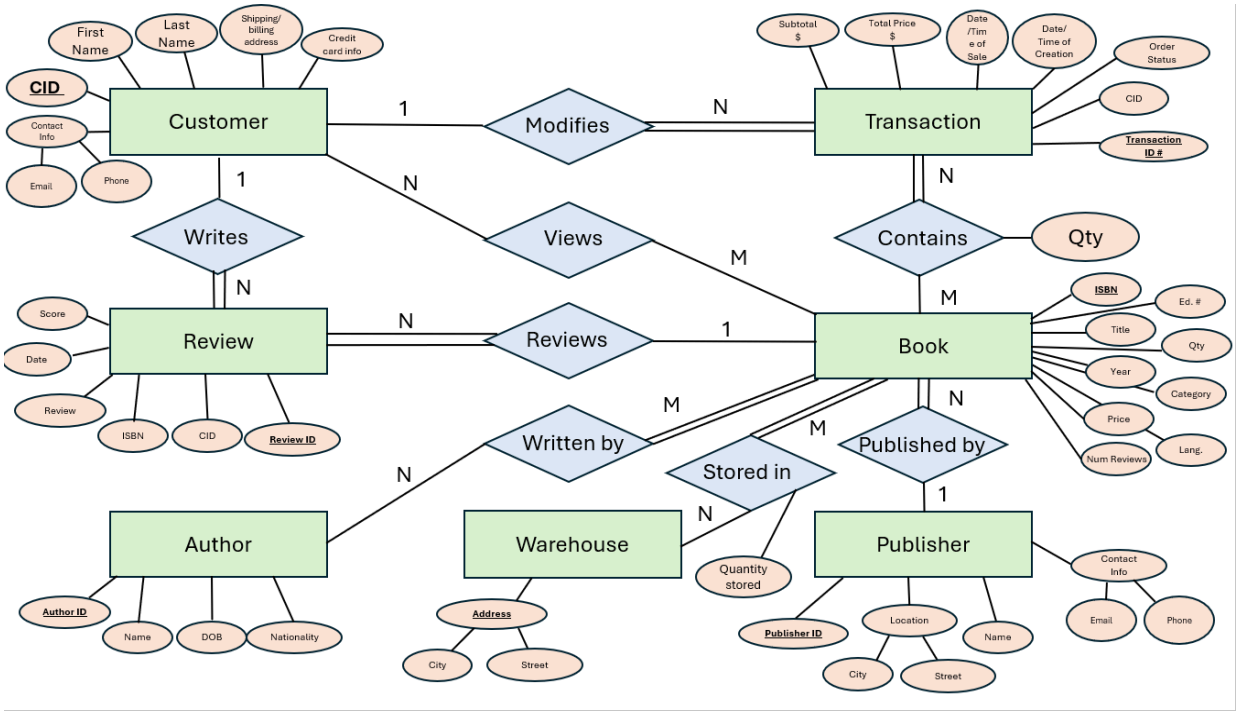
CSE 3241 Project Checkpoint 03 – SQL and More SQL

Names: Ethan Conley, Connor Fricke, Sierra Reis

Date: 03/19/2025

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 02. If you were instructed to change the model for Project Checkpoint 02, make sure you use the revised versions of your models.



Primary keys are underlined, {foreign keys are curly braces}

CUSTOMER (CID, First name, Last name, shipping/billing address, credit card info, email, phone)
REVIEW (Review ID, {CID}, {ISBN}, Review, Date, Score)
AUTHOR (AuthorID, Name, DOB, Nationality)
WAREHOUSE (City, Street)
PUBLISHER (PublisherID, City, Street, Name, Email, Phone)
BOOK (ISBN, Edition_#, Title, Quantity, Publication_Year, Category, Price, Language, Num_Reviews)
TRANSACTION (Transaction_ID, {CID}, Order_Status, Date/Time_of_Creation, Date/Time_of_Sale, Subtotal, Total_Price)
WRITTEN_BY({ISBN},{authorID})
VIEWS({CID}, {ISBN})
CONTAINS({transactionID}, {ISBN}, qty)
STORED_IN({address}, {ISBN}, Quantity_stored)

2. Given your relational schema, create a text file containing the SQL code to create your database schema. Use this SQL to create a database in SQLite. Populate this database with the data provided for the project as well as 20 sample records for each table that does not contain data provided in the original project documents.

```
CREATE TABLE Customer(
        CID                     INT             not null,
        FirstName               VARCHAR(16)     not null,
        LastName                VARCHAR(32)     not null,
        ShippingBillingAddress  VARCHAR(64)     not null,
        CreditCardInfo          VARCHAR(64)     not null,
        Email                   VARCHAR(32),
        Phone                   VARCHAR(16),
        Primary Key(CID)
);

CREATE TABLE Transactions(
        TransactionID           INT             NOT NULL,
        CID                     INT             NOT NULL,
        OrderStatus             VARCHAR(16)     NOT NULL,
        DateTimeCreation        DATETIME,
        DateTimeSale            DATETIME,
        Subtotal                DECIMAL(2, 2),
        TotalPrice              DECIMAL(2, 2),
        PRIMARY KEY(Transaction_ID),
        FOREIGN KEY (CID) REFERENCES Customer(CID)
);

CREATE TABLE Review(
        ReviewID                int             not null,
        CID                     int             not null,
        ISBN                    char(10)        not null,
        Review                  text,
        ReviewDate              date,
        Score                   int,
        Primary Key(ReviewID),
        Foreign Key(CID) references Customer(CID),
        Foreign Key(ISBN) references Book(ISBN)
);

CREATE TABLE Book (
        ISBN                    char(10)        not null,
        EditionNumber           int,
        Title                   varchar(32)     not null,
```

```
        Quantity                int,
        PublicationYear         int,
        Category                varchar(32),
        Price                   DECIMAL(2,2)        not null,
        Language                varchar(16),
        NumReviews              int,
        PublisherID             int                 not null,
        Primary Key(ISBN),
        FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID)
);

CREATE TABLE Author (
        AuthorID                int                 not null,
        Name                    varchar(64)         not null,
        DOB                     date,
        Nationality             varchar(64),
        Primary Key(AuthorID)
);

CREATE TABLE Warehouse (
        City                    varchar(32)         not null,
        Street                  varchar(32)         not null,
        Primary Key(City, Street)
);

CREATE TABLE Publisher(
        PublisherID             INT                 NOT NULL,
        City                    VARCHAR(16),
        Street                  VARCHAR(64),
        Name                    VARCHAR(32)         NOT NULL,
        Email                   VARCHAR(32),
        Phone                   CHAR(10)            NOT NULL,
        PRIMARY KEY  (PublisherID)
);

CREATE TABLE WrittenBy (
        ISBN                    CHAR(10)            NOT NULL,
        AuthorID                INT                 NOT NULL,
        FOREIGN KEY (ISBN) REFERENCES Book(ISBN),
        FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID)
);

CREATE TABLE Views (
        CID                     INT                 NOT NULL,
```

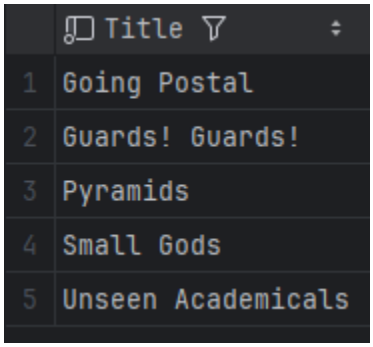
```
        ISBN                    CHAR(10)        NOT NULL,
        FOREIGN KEY(CID) REFERENCES Customer(CID),
        FOREIGN KEY(ISBN) REFERENCES Book(ISBN)
);

CREATE TABLE Contains (
        TransactionID           INT             NOT NULL,
        ISBN                    CHAR(10)        NOT NULL,
        Qty                     INT             NOT NULL,
        FOREIGN KEY (TransactionID) REFERENCES Transactions(TransactionID)
);

CREATE TABLE StoredIn (
        City                    VARCHAR(16)     NOT NULL,
        Street                  VARCHAR(64)     NOT NULL,
        ISBN                    CHAR(10)        NOT NULL,
        QuantityStored          INT             NOT NULL,
        FOREIGN KEY(City) REFERENCES Warehouse(City),
        FOREIGN KEY(Street) REFERENCES Warehouse(Street),
        FOREIGN KEY(ISBN) REFERENCES Book(ISBN)
);
```

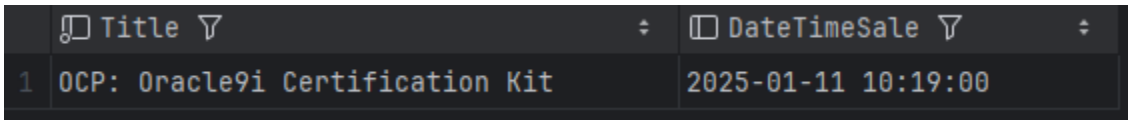
3. Given your relational schema, provide the SQL to perform the following queries. If your schema cannot provide answers to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries. These queries should be provided in a plain text file named “WorksheetTwoSimpleQueries.txt”:

a. Find the titles of all books by Pratchett that cost less than $10
SELECT Title FROM Book
NATURAL JOIN WrittenBy
NATURAL JOIN Author
WHERE AuthorName LIKE ‘%Pratchett%’ AND Price < 10;



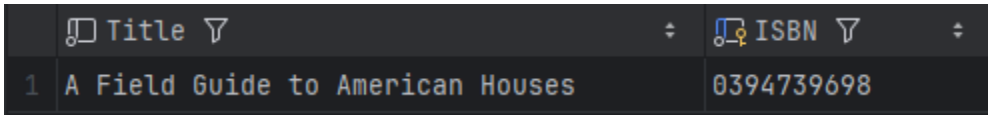
	Title
1	Going Postal
2	Guards! Guards!
3	Pyramids
4	Small Gods
5	Unseen Academicals

b. Give all the titles and their dates of purchase made by a single customer (you choose how to designate the customer)
SELECT Title, DateTimeSale
FROM Book NATURAL JOIN Contains
NATURAL JOIN Transactions
WHERE CID = 1;



	Title	DateTimeSale
1	OCP: Oracle9i Certification Kit	2025-01-11 10:19:00

c. Find the titles and ISBNs for all books with less than 5 copies in stock
SELECT Title, ISBN
FROM Book
WHERE Quantity < 5;

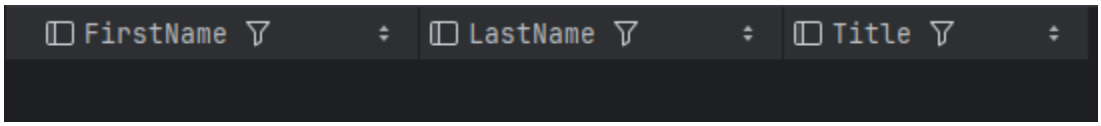


	Title	ISBN
1	A Field Guide to American Houses	0394739698

d. Give all the customers who purchased a book by Pratchett and the titles of Pratchett books they purchased
SELECT FirstName, LastName, Title

```
FROM Customer NATURAL JOIN Transactions
NATURAL JOIN Contains
NATURAL JOIN Book
NATURAL JOIN WrittenBy
NATURAL JOIN Author
WHERE AuthorName LIKE '%Pratchett%';
```

No output for our sample data,



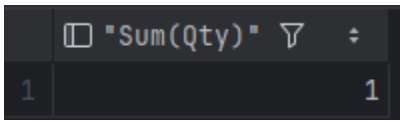
but would expect something like:

FirstName	LastName	Title
Jane	Smith	Going Postal

etc…

e. Find the total number of books purchased by a single customer (you choose how to designate the customer)

```
SELECT Sum(Qty)
FROM Transactions NATURAL JOIN Contains
WHERE CID = 1;
```



f. Find the customer who has purchased the most books and the total number of books they have purchased

```
SELECT FirstName, LastName, TotalBooks
From Customer NATURAL JOIN (
        SELECT CID, Sum(Qty) as TotalBooks
        FROM Transactions NATURAL JOIN Contains
        GROUP BY CID
)
WHERE TotalBooks = (
        SELECT MAX (TotalBooks)
        FROM (
                SELECT CID, Sum(Qty) as TotalBooks
                FROM Transactions NATURAL JOIN Contains
```

```
            GROUP BY CID
        )
)
```

	FirstName	LastName	TotalBooks
1	Liana	Brewer	5

4. For Project Checkpoint 02, you were asked to come up with three additional interesting queries that your database can provide. Give what those queries are supposed to retrieve in plain English, as relational algebra and then as SQL. Your queries should include joins and at least one should include an aggregate function, and they should be the same as the queries you outlined for Worksheet 02. If you were instructed to fix the queries in Checkpoint 02, make sure you use the fixed queries here. These queries should be provided in a plain text file named "WorksheetTwoExtraQueries.txt".

a) Find all Warehouses that store books written by Pratchett

```
SELECT City, Street
FROM Warehouse
NATURAL JOIN StoredIn
NATURAL JOIN WrittenBy
NATURAL JOIN Author
WHERE AuthorName LIKE '%Pratchett%';
```

No output for our sample data,

City	Street

But would expect something like:

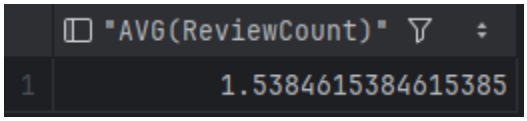
City	**Street**
Savannah	**123 Oak Avenue**

etc…

b) Find the average number of reviews written by customers

```
SELECT AVG(ReviewCount)
FROM (
        SELECT CID, COUNT(*) AS ReviewCount
        FROM Review
```

```
        GROUP BY CID
);
```

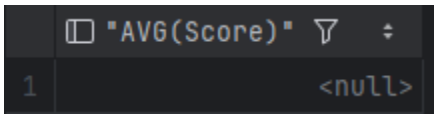


	"AVG(ReviewCount)"
1	1.5384615384615385

c) Find the average review score for books written by Pratchett

```
SELECT AVG(Score)
FROM Review
NATURAL JOIN Book
NATURAL JOIN WrittenBy
NATURAL JOIN Author
WHERE AuthorName LIKE '%Pratchett%';
```

No output for our sample data,



	"AVG(Score)"
1	<null>

But would expect something like:

AVG(Score)
4.7

5. Given your relational schema, provide the SQL for the following more advanced queries. These queries may require you to use techniques such as nesting, aggregation using HAVING clauses, and other techniques . If your database schema does not contain the information to answer to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries. Note that if your database does contain the information but in non-aggregated form, you should NOT revise your model but instead figure out how to aggregate it for the query! These queries should be provided in a plain text file named "WorksheetTwoAdvancedQueries.txt".

a. Provide a list of customer names, along with the total dollar amount each customer has spent.

```
SELECT FirstName, LastName, SUM(Transactions.TotalPrice) as TotalSpent
FROM Customer
```

NATURAL JOIN Transactions
GROUP BY CID;

	FirstName	LastName	TotalSpent
1	John	Smith	52.41
2	Stella	Sims	32.2
3	Sebastian	Orr	46.85
4	Alexia	Schultz	34.44
5	Allen	Travis	54.87
6	Sanaa	Small	31.75
7	Bella	Holland	58.35
8	Ian	Greene	33.48
9	Hailee	Munoz	20.37
10	Isaiah	Murray	42.99
11	Eliana	Ray	60.39
12	Alma	Skinner	21.97
13	Liana	Brewer	21.73
14	Sarahi	Morales	44.81
15	Clay	Strickland	11.05
16	Holden	Hester	11.9
17	Arielle	Mata	35.62
18	Harper	Haynes	36.01
19	Davis	Clayton	26.73
20	Jasmine	Richard	5.84

b. Provide a list of customer names and e-mail addresses for customers who have spent more than the average customer.

```
SELECT FirstName, LastName, Email
FROM Customer
NATURAL JOIN Transactions
GROUP BY CID
HAVING SUM(Transactions.TotalPrice) > (
        SELECT AVG(TotalSpent)
        FROM (
                SELECT SUM(Transactions.TotalPrice) as TotalSpent
```

```
                FROM Transactions
                GROUP BY CID
        )
);
```

	FirstName	LastName	Email
1	John	Smith	mohammed24@hotmail.com
2	Sebastian	Orr	efren42@hotmail.com
3	Alexia	Schultz	candace.ritchie22@hotmail.com
4	Allen	Travis	ashton63@gmail.com
5	Bella	Holland	miracle_oconnell4@yahoo.com
6	Isaiah	Murray	pascale_oreilly@hotmail.com
7	Eliana	Ray	burnice79@yahoo.com
8	Sarahi	Morales	alvera79@gmail.com
9	Arielle	Mata	garrick.franecki19@hotmail.com
10	Harper	Haynes	stanford54@gmail.com

(Note: The emails do not correspond to the persons actual name, they were randomly generated)

c. Provide a list of the titles in the database and associated total copies sold to customers, sorted from the title that has sold the most individual copies to the title that has sold the least.

```
SELECT Title, SUM(Contains.Qty) as TotalSold
FROM Book
NATURAL JOIN Contains
GROUP BY Title
ORDER BY TotalSold DESC;
```

	Title	TotalSold
1	Cerulean Sins	5
2	SQL Server 2000 for Experienced	3
3	Execution: The Discipline of Get	3
4	The Pianist	2
5	The Data Warehouse Toolkit: The	2
6	Fundamentals of Database Systems	2
7	ColdFusion MX Web Application Co	2
8	The Secret Life of Bees	1
9	The Hours	1
10	The Guru's Guide to Transact-SQL	1
11	The Girl in the Red Coat: A Memo	1
12	Professional SQL Server 2000 Pro	1
13	OCP: Oracle9i Certification Kit	1
14	MySQL	1
15	Investing in Fixer-Uppers : A Co	1
16	How To Do Everything with Your T	1
17	Google Hacks	1
18	Data Mining: Practical Machine L	1
19	Creating Documents with Business	1
20	Access 2002 Developer's Handbook	1

d. Provide a list of the titles in the database and associated dollar totals for copies sold to customers, sorted from the title that has sold the highest dollar amount to the title that has sold the smallest.

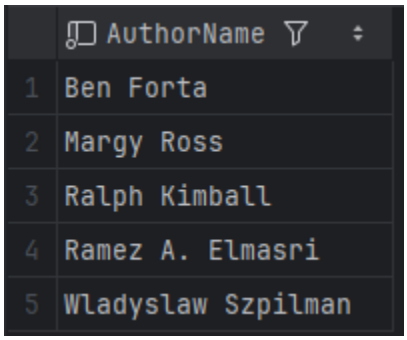
```
SELECT Title, SUM(Transactions.TotalPrice) as TotalSold
FROM Book
NATURAL JOIN Contains
NATURAL JOIN Transactions
GROUP BY Title
ORDER BY TotalSold DESC;
```

	Title	TotalSold
1	Professional SQL Server 2000 Pro	60.39
2	The Guru's Guide to Transact-SQL	58.35
3	Data Mining: Practical Machine L	54.87
4	OCP: Oracle9i Certification Kit	52.41
5	The Data Warehouse Toolkit: The	46.85
6	The Secret Life of Bees	44.81
7	Fundamentals of Database Systems	42.99
8	The Girl in the Red Coat: A Memo	36.01
9	The Pianist	35.62
10	How To Do Everything with Your T	34.44
11	Access 2002 Developer's Handbook	33.48
12	SQL Server 2000 for Experienced	32.2
13	Creating Documents with Business	31.75
14	The Hours	26.73
15	MySQL	21.97
16	Cerulean Sins	21.73
17	ColdFusion MX Web Application Co	20.37
18	Investing in Fixer-Uppers : A Co	11.9
19	Google Hacks	11.05
20	Execution: The Discipline of Get	5.84

e. Find the most popular author in the database (i.e. the one who has sold the most books)

```
SELECT AuthorName
FROM Author
NATURAL JOIN WrittenBy
NATURAL JOIN Book
NATURAL JOIN Contains
GROUP BY AuthorName
HAVING Contains.Qty = (
        SELECT MAX(Contains.Qty)
        FROM Author
        NATURAL JOIN WrittenBy
```

```
        NATURAL JOIN Book
        NATURAL JOIN Contains
        GROUP BY AuthorName
);
```



	AuthorName
1	Ben Forta
2	Margy Ross
3	Ralph Kimball
4	Ramez A. Elmasri
5	Wladyslaw Szpilman

(Note: In our data they have each sold two books:

	AuthorName	Qty
1	Ben Forta	2
2	Margy Ross	2
3	Ralph Kimball	2
4	Ramez A. Elmasri	2
5	Wladyslaw Szpilman	2

)

f. Find the most profitable author in the database for this store (i.e. the one who has brought in the most money)

```
SELECT AuthorName
FROM Author
NATURAL JOIN WrittenBy
NATURAL JOIN Book
NATURAL JOIN Contains
NATURAL JOIN Transactions
GROUP BY AuthorName
HAVING TotalPrice = (
        SELECT MAX(TotalPrice)
        FROM Author
        NATURAL JOIN WrittenBy
        NATURAL JOIN Book
        NATURAL JOIN Contains
        NATURAL JOIN Transactions
```

```
        GROUP BY AuthorName
);
```

	AuthorName
1	Ben Forta

g. Provide a list of customer information for customers who purchased anything written by the most profitable author in the database.

```
SELECT FirstName, LastName, Email, Phone, ShippingBillingAddress, CreditCardInfo
FROM Customer
NATURAL JOIN Transactions
NATURAL JOIN Contains
NATURAL JOIN Book
NATURAL JOIN WrittenBy
NATURAL JOIN Author
WHERE AuthorName = (
        SELECT AuthorName
        FROM Author
        NATURAL JOIN WrittenBy
        NATURAL JOIN Book
        NATURAL JOIN Contains
        NATURAL JOIN Transactions
        GROUP BY AuthorName
        HAVING TotalPrice =  (
                SELECT MAX(TotalPrice)
                FROM Author
                NATURAL JOIN WrittenBy
                NATURAL JOIN Book
                NATURAL JOIN Contains
                NATURAL JOIN Transactions
                GROUP BY AuthorName
        )
);
```

	FirstName	LastName	Email	Phone	ShippingBillingAddress	CreditCardInfo
1	Hailee	Munoz	noel97@gmail.com	5896513207	3310 Co RD S	1531143416542315

h. Provide the list of authors who wrote the books purchased by the customers who have spent more than the average customer.

```
SELECT AuthorName
FROM Author
```

```
NATURAL JOIN WrittenBy
NATURAL JOIN Book
NATURAL JOIN Contains
NATURAL JOIN Transactions
NATURAL JOIN Customer
WHERE CID IN (
        SELECT CID
        FROM Customer
        NATURAL JOIN Transactions
        GROUP BY CID
        HAVING SUM(TotalPrice) > (
                SELECT AVG(TotalSpent)
                FROM (
                        SELECT SUM(TotalPrice) as TotalSpent
                        FROM Transactions
                        GROUP BY CID
                )
        )
);
```

	AuthorName
1	Chip Dawes
2	Ralph Kimball
3	Margy Ross
4	Bill Mann
5	Ian H. Witten
6	Ken Henderson
7	Ramez A. Elmasri
8	Rob Vieira
9	Sue Monk Kidd
10	Wladyslaw Szpilman
11	Roma Ligocka

Once you have completed all of the questions for Part Two, create a ZIP archive containing the binary SQLite file and the three text files and submit this to the Carmen Dropbox. Make sure your queries work against your database and provide your expected output before you submit them!