User Manual

1. **TABLE INSTRUCTIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | Classification | Description | Attribute | Description | Constraints |
| Item | Entity | All the items that are sold in the bookstore | barcode | barcode | char (12) not null |
| item\_name | name of item | varchar(50) not null |
| aisle | aisle number | int not null |
| bin | bin number | int not null |
| price | price | float(5, 2) not null |
| publisher\_name | name of publisher | varchar(30) not null |
| Publisher | Entity | The publisher | publisher\_name | name of publisher | varchar(30) not null |
| Store | Entity | Each store in our bookstore chain | store\_ID | store ID | char(10) not null |
| store\_name | store name | varchar(30) not null |
| zip\_code | zip code | char(5) not null |
| country | country | varchar(20) not null |
| Book | Entity | Book | barcode | barcode | CHAR(12) not null |
| book\_category | book category | VARCHAR(20)not null |
| ISBN | ISBN | CHAR(22) not null |
| Software | Entity | Software sold in the store | soft\_ID | ID of this software | INT(8) NOT NULL |
| soft\_category | software category | VARCHAR(20) NOT NULL |
| barcode | barcode | CHAR(12) NOT NULL |
| Journal | Entity | Journal sold in the store | barcode | barcode | CHAR(12) not null |
| ISSN | ISSN | INT(8) not null |
| Issue | Issue | VARCHAR(20) not null |
| CD/DVD | Entity | CD/DVD sold in the store | barcode | barcode | char(12) not null |
| genre | genre | varchar(20) |
| disk\_ID | ID of this disk | varchar(30) not null |
| Author/Artist | Entity | Author of books or artist of CD/DVD | a\_ID | ID of the author or artist | varchar(20) NOT NULL |
| first\_name | first name | varchar(15) NOT NULL |
| last\_name | last name | varchar(15) NOT NULL |
| Transaction | Entity | Each transaction information | transaction\_ID | transaction ID | char(20) not null |
| time | time of the transaction | date not null |
| customer\_ID | customer ID | char(10) not null |
| Customer | Entity | All customers who have shopped and recorded information in our database | customer\_ID | customer ID | char(10) not null |
| first\_name | first name | varchar(15) not null |
| last\_name | last name | varchar(15) not null |
| email | email of the customer | varchar(30) not null |
| Stores | Relationship | Bookshop stores items | stroe\_ID | stroe ID | char(10) not null |
| barcode | barcode | char(12) not null |
| inventory | inventory of the item | varchar(20) not null |
| sold | items sold | int not null |
| Written\_by | Relationship | Books are written by authors | ISBN | ISBN | CHAR(22) NOT NULL |
| a\_ID | ID of the author or artist | VARCHAR(20) NOT NULL |
| Contained\_IN | Relationship | Items are contained in each transaction | quantity | quantity of the item in this transaction | INT NOT NULL |
| barcode | barcode | CHAR(12) NOT NULL |
| transaction\_ID | transaction ID | INT NOT NULL |
| Compsed\_by | Relationship | CD/DVD are composed by artists | disk\_ID | ID of this disk | VARCHAR(30) NOT NULL |
| a\_ID | ID of the author or artist | VARCHAR(20) NOT NULL |

1. **SAMPLE QUERIES**
2. **Find the titles of all books by Pratchett that cost less than $10**

Relational Algebra:

A1 🡨 Written\_by \* Book

A2 🡨 A1 \* Item

A3 🡨 A2 \* Author\_Artist

Result 🡨

SQL:

SELECT I.item\_name

FROM ITEM AS I, AUTHOR\_ARTIST AS A, BOOK AS B, WRITTEN\_BY AS WB

WHERE I.barcode=B.barcode

AND B.ISBN=WB.ISBN

AND A.a\_ID=WB.a\_ID

AND A.last\_name='Pratchett'

AND I.price<10;

1. **Give all the titles and their dates of purchase made by a single customer (you choose how to designate the customer)**

Relational Algebra:

B1 🡨 Contained\_in \* Book

B2 🡨 B1 \* Transaction

B3 🡨 B2 \* Customer

Result 🡨

SQL:

SELECT I.item\_name, T.time

FROM TRANSACTIONS AS T, CONTAINED\_IN AS C, ITEM AS I

WHERE T.transaction\_ID = C.transaction\_ID

AND C.barcode = I.barcode

AND T.customer\_ID = 7938024839;

1. **Find the titles and ISBNs for all books with less than 5 copies in stock**

Relational Algebra:

C1 🡨 (Stores \* Book) \* Item

Result 🡨

SQL:

SELECT I.item\_name, B.ISBN

FROM ITEM AS I, BOOK AS B, STORES AS STS

WHERE I.barcode=B.barcode

AND I.barcode=STS.barcode

AND STS.inventory < 5;

1. **Give all the customers who purchased a book by Pratchett and the titles of Pratchett books they purchased**

Relational Algebra:

D1🡨 Book\*Item

D2🡨 D1\*Written\_by

D3🡨 D2\*Author/Artist

D4🡨 D3\*Contained\_in

D5🡨 D4\*Transaction

Result 🡨

SQL:

SELECT C.customer\_ID, C.first\_name, C.last\_name, I.item\_name

FROM TRANSACTIONS AS T, CONTAINED\_IN AS CI, ITEM AS I, BOOK AS B, AUTHOR\_ARTIST AS AA, WRITTEN\_BY AS WB, CUSTOMER AS C

WHERE T.transaction\_ID = CI.transaction\_ID

AND T.customer\_ID = C.customer\_ID

AND CI.barcode = I.barcode

AND I.barcode = B.barcode

AND AA. a\_ID = WB.a\_ID

AND WB.ISBN = B.ISBN

AND AA.last\_name = 'Pratchett';

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

Relational Algebra:

D1 🡨 Book \* Item

D2 🡨 Written\_by \* (D1 \* Author)

D3 🡨 Transaction \* (D2 \* Contained\_in)

D4 🡨 D3 \* Customer

D5 🡨 -- change id to 7938024839

SQL:

SELECT SUM(CI.quantity)

FROM TRANSACTIONS AS TR, CONTAINED\_IN AS CI, ITEM AS I, BOOK AS B, CUSTOMER AS C

WHERE TR.transaction\_ID = CI.transaction\_ID

AND CI.barcode = I.barcode

AND I.barcode = B.barcode

AND C.customer\_ID = TR.customer\_ID

AND C.customer\_ID= 7938024839;

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

Relational Algebra:

D1 = Book \* Item

D2 = Written by \* D1

D3 = Transaction \* (D2 \* Contained in)

D4 = Customer\_IDFSUM Quantity (D3)

Result = FCustomer\_ID, Max (SUM Quantity) (D4)

SQL:

SELECT MAX(sum\_Quantity), customer\_ID

FROM (

SELECT SUM(quantity) AS sum\_Quantity, customer\_ID

FROM TRANSACTIONS AS TR, CONTAINED\_IN AS CI, ITEM AS I, BOOK AS B

WHERE TR.transaction\_ID = CI.transaction\_ID

AND CI.barcode = I.barcode

AND I.barcode = B.barcode

GROUP BY TR.customer\_ID);

1. **Find The most productive author**

Relational Algebra:

M1 = Author \* Written\_by

M2 = a\_idFCOUNT ISBN, first\_name, last\_name(M1)

Result = F(MAX counted\_ISBN, first\_name, last\_name)(M2)

SQL:

SELECT fname, lname, MAX(ISBN\_count)

FROM (

SELECT AA.first\_name AS fname, AA.last\_name AS lname, COUNT(B.ISBN) AS ISBN\_count

FROM AUTHOR\_ARTIST AS AA, WRITTEN\_BY AS WB, BOOK AS B

WHERE AA.a\_ID = WB.a\_ID AND WB.ISBN = B.ISBN);

1. **Find Inventory of all CD/DVDs in the store named “Buckeye”**

Relational Algebra:

M1 = CD/DVD \* Item

M2 = M1 \* Stores

M3 = M2 \* Store

M4 = σstore\_name = ‘buckeye’(M3)

Result =πdisk\_ID, inventory(M4)

SQL:

SELECT STORES.inventory, CD\_DVD.disk\_ID

FROM STORES, STORE, ITEM, CD\_DVD

WHERE STORES.store\_ID = STORE.store\_ID

AND STORES.barcode = ITEM.barcode

AND ITEM.barcode = CD\_DVD.barcode

AND STORE.store\_name = 'Buckeye';

1. **Find The most popular artist**

Relational Algebra:

M1 = CD/DVD \* Item

M2 = M1 \* Composed\_by

M3 = M2 \* Contained\_in

M4 = M3 \* Author/Artist

M5 = a\_id, first\_name, last\_nameFSUM quantity(M4)

Result = F(a\_id, first\_name, last\_name, MAX sum\_quantity)(M5)

SQL:

SELECT fname, lname, MAX(quantity\_sum)

FROM (

SELECT AA.first\_name AS fname, AA.last\_name AS lname, SUM(CI.quantity) AS quantity\_sum

FROM AUTHOR\_ARTIST AS AA, COMPOSED\_BY AS CB, CD\_DVD AS CD, ITEM AS IT, CONTAINED\_IN AS CI

WHERE AA.a\_ID = CB.a\_ID

AND CB.disk\_ID = CD.disk\_ID

AND CD.barcode = IT.barcode

AND IT.barcode = CI.barcode

GROUP BY AA.a\_ID);

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

SQL:

SELECT C.last\_name,C.first\_name, sum(quantity\*price)

FROM TRANSACTIONS AS T, CUSTOMER AS C, CONTAINED\_IN AS CI, ITEM AS I

WHERE T.transaction\_ID = CI.transaction\_ID

AND C.customer\_ID = T.customer\_ID

AND CI.barcode = I.barcode

GROUP BY C.customer\_ID;

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

SQL:

SELECT C.first\_name, C.last\_name, C.email

FROM TRANSACTIONS AS T, CUSTOMER AS C, ITEM AS I, CONTAINED\_IN AS CI

WHERE T.transaction\_ID = CI.transaction\_ID

AND CI.barcode = I.barcode

AND T.customer\_ID = C.customer\_ID

GROUP BY T.customer\_ID

HAVING quantity\*price > AVG(quantity\*price);

1. **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.**

SQL:

SELECT sum(quantity), IT.item\_name

FROM CONTAINED\_IN AS CI, ITEM AS IT

WHERE IT.barcode = CI.barcode

GROUP BY IT.item\_name

ORDER BY CI.quantity DESC;

1. **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.**

SQL:

SELECT price\*quantity

FROM CONTAINED\_IN AS CI, ITEM AS IT,  BOOK AS B, JOURNAL AS J, SOFTWARE AS S,CD\_DVD AS C

WHERE IT.barcode=B.barcode

AND IT.barcode = J.barcode

AND IT.barcode = S.barcode

AND IT.barcode = CI.barcode

AND IT.barcode = CI.barcode

GROUP BY IT.barcode

ORDER BY price\*quantity DESC;

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

SQL:

SELECT a\_ID, fname, lname, max(quantity\_sum)

FROM (

SELECT A.a\_ID AS a\_ID, sum(CI.quantity) AS quantity\_sum, A.first\_name AS fname, A.last\_name AS lname

FROM AUTHOR\_ARTIST AS A, TRANSACTIONS AS T, CONTAINED\_IN AS CI, ITEM AS I, BOOK AS B, WRITTEN\_BY AS WB

WHERE T.transaction\_ID=CI.transaction\_ID

AND CI.barcode=I.barcode

AND A.a\_ID=WB.a\_ID

AND WB.ISBN=B.ISBN

AND B.barcode=I.barcode

AND CI.barcode=I.barcode

GROUP BY A.a\_ID);

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

SQL:

SELECT fname, lname

FROM (

SELECT a\_ID, max(price\_sum) AS price\_max, fname, lname

FROM (

SELECT A.a\_ID AS a\_ID, sum(CI.quantity\*I.price) AS price\_sum, A.first\_name AS fname, A.last\_name AS lname

FROM AUTHOR\_ARTIST AS A, TRANSACTIONS AS T, CONTAINED\_IN AS CI, ITEM AS I, BOOK AS B, WRITTEN\_BY AS WB

WHERE T.transaction\_ID=CI.transaction\_ID

AND CI.barcode=I.barcode

AND A.a\_ID=WB.a\_ID

AND WB.ISBN=B.ISBN

AND B.barcode=I.barcode

AND CI.barcode=I.barcode

GROUP BY A.a\_ID));

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

SQL:

SELECT c\_ID,fname, lname, email

FROM (

SELECT a\_ID, max(price\_sum) AS price\_max, c\_ID, fname, lname, email

FROM (

SELECT A.a\_ID AS a\_ID, sum(CI.quantity\*I.price) AS price\_sum, CU.customer\_ID as c\_ID, A.first\_name as fname, A.last\_name AS lname, CU.email AS email

FROM AUTHOR\_ARTIST AS A, TRANSACTIONS AS T, CONTAINED\_IN AS CI, ITEM AS I, BOOK AS B, WRITTEN\_BY AS WB, CUSTOMER AS CU

WHERE T.transaction\_ID=CI.transaction\_ID

AND CI.barcode=I.barcode

AND A.a\_ID=WB.a\_ID

AND WB.ISBN=B.ISBN

AND B.barcode=I.barcode

AND CI.barcode=I.barcode

AND CU.customer\_ID = T.customer\_ID

GROUP BY A.a\_ID));

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

SQL:

SELECT AA.first\_name, AA.last\_name

FROM ITEM AS I, CONTAINED\_IN AS CI, TRANSACTIONS AS T, BOOK AS B, WRITTEN\_BY AS WB, AUTHOR\_ARTIST AS AA

WHERE I.barcode = CI.barcode

AND B.barcode = I.barcode

AND WB.ISBN = B.ISBN

AND AA.a\_ID = WB.a\_ID

AND CI.transaction\_ID = T.transaction\_ID

GROUP BY T.customer\_ID

HAVING sum(I.price \* CI.quantity) > (

SELECT AVG(I.price \* CI.quantity) AS avg\_spend

FROM ITEM AS I, CONTAINED\_IN AS CI, TRANSACTIONS AS T

WHERE I.barcode = CI.barcode

AND CI.transaction\_ID = T.transaction\_ID);

1. **INSERT SYNTAX**

**1. When adding a new book, due to the restrictions and Referential Integrity Constraints, you must insert the related values in table BOOK, table Author, table Written\_by, table PUBLISHER , table STORES, table ITEM.**

**SQL syntax:**

INSERT INTO BOOK(barcode, book\_category, ISBN)

VALUES (CHAR(12), VARCHAR(20), CHAR(22));

INSERT INTO WRITTEN\_BY (ISBN, a\_ID)

VALUES (CHAR(22), VARCHAR(20));

INSERT INTO STORES (store\_ID, barcode ,inventory,sold)

VALUES (CHAR(10), CHAR(12),VARCHAR(20),INT);

INSERT INTO ITEM (barcode, item\_name , aisle, bin, price, publisher\_name)

VALUES (CHAR(12), VARCHAR(50),INT,INT,FLOAT(5,2), VARCHAR(30));

INSERT INTO AUTHOR\_ARTIST (a\_ID,firstname,last\_name,)

VALUES (VARCHAR(20), VARCHAR(15), VARCHAR(15));

**Example:**

INSERT INTO BOOK(barcode, book\_category, ISBN)

VALUES (‘602978009201’, ‘Reference’, ‘743222001’);

INSERT INTO WRITTEN\_BY (ISBN, a\_ID)

VALUES (‘743222001’, ‘A23542521’);

INSERT INTO STORES (store\_ID,barcode ,inventory,sold)

VALUES (‘6611445639’, ‘602978009201’, ‘14’ , 1);

INSERT INTO ITEM (barcode, item\_name, aisle, bin, price, publisher\_name)

VALUES (‘602978009201’, ‘special’, 11, 11, 51.11, ‘Sybex’);

INSERT INTO AUTHOR\_ARTIST (a\_ID,firstname,last\_name,)

VALUES (‘A23542521’, ‘Jackson’, ‘Wilshereman’);

**2. When adding a new author, due to the restrictions and Referential Integrity Constraints, you must insert the related values in table BOOK, table Author, table Written\_by, table PUBLISHER, table STORES, table ITEM.**

**SQL syntax:**

INSERT INTO BOOK(barcode, book\_category, ISBN)

VALUES (CHAR(12), VARCHAR(20), CHAR(22));

INSERT INTO WRITTEN\_BY (ISBN, a\_ID)

VALUES (CHAR(22), VARCHAR(20));

INSERT INTO STORES (store\_ID,barcode ,inventory,sold)

VALUES (CHAR(10), CHAR(12),VARCHAR(20),INT);

INSERT INTO ITEM (barcode, item\_name , aisle, bin, price, publisher\_name)

VALUES (CHAR(12), VARCHAR(50),INT,INT,FLOAT(5,2), VARCHAR(30));

INSERT INTO AUTHOR\_ARTIST (a\_ID,firstname,last\_name,)

VALUES (VARCHAR(20), VARCHAR(15), VARCHAR(15));

**Example:**

INSERT INTO BOOK(barcode, book\_category, ISBN)

VALUES (‘602978009201’, ‘Reference’, ‘743222001’);

INSERT INTO WRITTEN\_BY (ISBN, a\_ID)

VALUES (‘743222001’, ‘A23542521’);

INSERT INTO STORES (store\_ID,barcode ,inventory,sold)

VALUES (‘6611445639’, ‘602978009201’, ‘14’ , 1);

INSERT INTO ITEM (barcode, item\_name, aisle, bin, price, publisher\_name)

VALUES (‘602978009201’, ‘special’, 11, 11, 51.11, ‘Sybex’);

INSERT INTO AUTHOR\_ARTIST (a\_ID,firstname,last\_name,)

VALUES (‘A23542521’, ‘Jackson’, ‘Wilshereman’);

3.When adding a new publisher, you must insert the related values in table PUBLISHER

**SQL syntax:**

INSERT INTO PUBLISHER (publisher\_name)

VALUES (VARCHAR(30));

**Example :**

INSERT INTO PUBLISHER (publisher\_name)

VALUES (‘strange’);

**4.When adding a new customer, you must insert the related values in table Customer.**

**SQL syntax:**

INSERT INTO Customer (customer\_ID, first\_name,last\_name,email)

VALUES (CHAR(10),VARCHAR(15), VARCHAR(15), VARCHAR(30));

**Example:**

INSERT INTO Customer (customer\_ID, first\_name,last\_name,email)

VALUES (‘7938024831’, ‘An’, ‘allen’, ‘allen@gmail.com’);

1. **DELETE SYNTAX**

**1.When deleting a book, due to the restrictions and Referential Integrity Constraints, you must delete the related values in table BOOK, table Written\_by, table STORES, table ITEM, table Author**

**SQL syntax:**

DELETE FROM BOOK

WHERE barcode = CHAR(12) AND

book\_category = VARCHAR(20) AND

ISBN = CHAR(22) ;

DELETE FROM WRITTEN\_BY

WHERE ISBN = CHAR(22) AND

a\_ID = VARCHAR(20);

DELETE FROM STORES

WHERE store\_ID = char(10) AND

barcode = char(12) AND

inventory = varchar(20) AND

sold = int;

DELETE FROM ITEM

WHERE barcode =char (12) AND

item\_name =varchar(50) AND

aisle =int AND

bin =int AND

price= float(5, 2) AND

publisher\_name =varchar(30);

**Example:**

DELETE FROM BOOK

WHERE barcode = ‘602978009201’ AND

book\_category = ‘Reference’ AND

ISBN = ‘743222001’;

DELETE FROM WRITTEN\_BY

WHERE ISBN = ‘743222001’AND

a\_ID =‘A23542521’;

DELETE FROM STORES

WHERE store\_ID = ‘6611445639’AND

barcode = ‘602978009201’ AND

inventory = ‘14’ AND

sold = 1;

DELETE FROM ITEM

WHERE barcode =‘602978009201’AND

item\_name = ‘special’AND

aisle =11 AND

bin =11 AND

price= 51.11 AND

publisher\_name =‘Sybex’;

DELETE FROM AUTHOR\_ARTIST

WHERE a\_ID =‘A23542521’AND

firstname =‘Jackson’AND

last\_name = ‘Wilshereman’;

**2.When deleting a author, due to the restrictions and Referential Integrity Constraints, you must delete the related values in table BOOK, table Author, table Written\_by, table table STORES, table ITEM.**

**SQL syntax:**

DELETE FROM BOOK

WHERE barcode = CHAR(12) AND

book\_category = VARCHAR(20) AND

ISBN = CHAR(22) ;

DELETE FROM WRITTEN\_BY

WHERE ISBN = CHAR(22) AND

a\_ID = VARCHAR(20);

DELETE FROM STORES

WHERE store\_ID = char(10) AND

barcode = char(12) AND

inventory = varchar(20) AND

sold = int;

DELETE FROM ITEM

WHERE barcode =char (12) AND

item\_name =varchar(50) AND

aisle =int AND

bin =int AND

price= float(5, 2) AND

publisher\_name =varchar(30);

DELETE FROM AUTHOR\_ARTIST

WHERE a\_ID =varchar(20) AND

firstname =varchar(15) AND

last\_name = varchar(15);

**Example:**

DELETE FROM BOOK

WHERE barcode = ‘602978009201’ AND

book\_category = ‘Reference’ AND

ISBN = ‘743222001’;

DELETE FROM WRITTEN\_BY

WHERE ISBN = ‘743222001’AND

a\_ID =‘A23542521’;

DELETE FROM STORES

WHERE store\_ID = ‘6611445639’AND

barcode = ‘602978009201’ AND

inventory = ‘14’ AND

sold = 1;

DELETE FROM ITEM

WHERE barcode =‘602978009201’AND

item\_name = ‘special’AND

aisle =11 AND

bin =11 AND

price= 51.11 AND

publisher\_name =‘Sybex’;

DELETE FROM AUTHOR\_ARTIST

WHERE a\_ID =‘A23542521’AND

firstname =‘Jackson’AND

last\_name = ‘Wilshereman’;

**3.When deleting a publisher, you must insert the related values in table PUBLISHER**

**SQL syntax:**

DELETE FROM  PUBLISHER

WHERE publisher\_name=varchar(30);

**Example:**

DELETE FROM  PUBLISHER

WHERE publisher\_name=‘strange’;

**4.When deleting a costomer, you must delete the related values in table Customer.**

**SQL syntax:**

DELETE FROM  Custome

WHERE customer\_ID = CHAR(10) AND

firstname =varchar(15) AND

last\_name = varchar(15) AND

email = VARCHAR(30);

**Example:**

DELETE FROM  Custome

WHERE customer\_ID = ‘7938024831’ AND

firstname =‘An’AND

last\_name = ‘allen’AND

email = ‘allen@gmail.com’ ;

**5.The delete statement for each entity in our database.**

DELETE FROM BOOK

WHERE barcode = CHAR(12) AND

book\_category = VARCHAR(20) AND

ISBN = CHAR(22) ;

DELETE FROM WRITTEN\_BY

WHERE ISBN = CHAR(22) AND

a\_ID = VARCHAR(20);

DELETE FROM STORES

WHERE store\_ID = char(10) AND

barcode = char(12) AND

inventory = varchar(20) AND

sold = int;

DELETE FROM STORE

WHERE store\_ID = char(10) AND

store\_name= varchar(30) AND

zip\_code = char(5) AND

country = varchar(20) ;

DELETE FROM ITEM

WHERE barcode =char (12) AND

item\_name =varchar(50) AND

aisle =int AND

bin =int AND

price= float(5, 2) AND

publisher\_name =varchar(30);

DELETE FROM PUBLISHER

WHERE publisher\_name = VARCHAR(30) ;

DELETE FROM AUTHOR\_ARTIST

WHERE a\_ID =varchar(20) AND

firstname =varchar(15) AND

last\_name = varchar(15);

DELETE FROM  Custome

WHERE customer\_ID = CHAR(10)

firstname =varchar(15) AND

last\_name = varchar(15) AND

email = VARCHAR(30);

DELETE FROM  TRANSACTIONS

WHERE transaction\_ID = CHAR(2) AND

Time=ate AND

Customer=ID char(10) AND

email = VARCHAR(30);

DELETE FROM  CONTAINED\_IN

WHERE quantity =INT AND

barcode =CHAR(12) AND

transaction\_ID =INT;

DELETE FROM  COMPOSED\_BY \_IN

WHERE disk\_ID= VARCHAR(30) AND

a\_ID =VARCHAR(20);

DELETE FROM  SOFTWARE

WHERE soft\_ID= INT(8) AND

soft\_category =VARCHAR(20) AND

barcode =CHAR(12);

DELETE FROM  JOURNA

WHERE ISSN=INT(8) AND

issue =ARCHAR(20)AND

barcode =CHAR(12);

DELETE FROM  CD\_DVD

WHERE barcode char(12))AND

genre =archar(20) AND

disk\_ID =archar(30);