A.

CUSTOMER table represents customers in real word. The attributes of CUSTOMER table are listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Attribute meaning | Data type | Constraint |
| Email | the email address of customers | character string with length up to 50 | Primary key  Key Constraint |
| Interest | the kinds of books a specific customer likes to read | character string with length up to 30 | NULL is allowed |
| Birth\_date | customers’ birth dates | Character string with length 8 | NOT NULL |
| LName | customers’ last names | character string with length up to 30 | NOT NULL |
| FName | customers’ first names | character string with length up to 30 | NOT NULL |
| Gender | Customers’ genders | Character string with length 1 | NOT NULL |
| Phone | Customers’ phone numbers | Character string with length 10 | NOT NULL |
| CA\_id | The id numbers of customers’ home address | character string with length up to 4 | Foreign key to ADDRESS table’s primary key( Aid)  NOT NULL  Referential integrity constraint |
| CusLicense\_no | Customers’ drive license numbers | Character string with length 8 | NOT NULL  Candidate key |

EMPLOYEE table represents employees in real world, with the attributes table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Ssn | Employee's ssn | Character string with length 11 | Primary key  Key Constraint |
| Salary | Employee’s Salary | Double with two decimal places | NOT NULL |
| Birth\_date | Employee’s birthday | Character string with length 8 | NOT NULL |
| FName | Employee’s first name | Character string with length up to 20 | NOT NULL |
| LName | Employee’s last name | Character string with length up to 20 | NOT NULL |
| Gender | Employee’s gender | Character string with length 1 | character ‘M’ or ‘F’, can be NULL |
| Phone | Employee’s phone | Character string with length 10 | NOT NULL |
| EM\_License\_no | Employee’s driver's license number | Character string with length 8 | UNIQUE |
| EA\_id | Employee’s address id number | Character string with length up to 4 | NOT NULL, Foreign key to ADDRESS’s primary key Aid, referential integrity constraint |

AUTHOR table represents authors in real world, with the attributes table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Id | Author id that uniquely identify each author | Character string with length 9 | Primary key  Key Constraint |
| Birth\_date | Author’s birthday | Character string with length 8 | NOT NULL |
| FName | Author’s first name | Character string with length up to 20 | NOT NULL |
| LName | Author’s last name | Character string with length up to 20 | NOT NULL |
| Gender | Author’s gender | Character string with length 1 | character ‘M’ or ‘F’, NOT NULL |
| Phone | Author’s phone | Character string with length 10 | NOT NULL |
| AU\_License\_no | Author’s driver license number | Character string with length 8 | UNIQUE |
| AA\_id | Author’s address id number | Character string with length up to 4 | NOT NULL, Foreign key to ADDRESS’s primary key Aid, referential integrity constraint |

COUSTOMER\_SERVICE table represents Customer service employee in online book store, with the attributes table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Typing\_speed | Employee’s typing speed when recording customer’s questions | Character string with length up to 20 | NOT NULL |
| Service\_Ssn | Employee’s ssn | Character string with length 11 | NOT NULL,  Primary Key,  Foreign key to EMPLOYEE’s primary key Ssn, referential integrity constraint |

TECHNICIAN table represents technique employee in online book store, with the attributes table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Tech\_Ssn | Technical employee’s ssn | Character string with length 11 | NOT NULL,  Primary Key, Foreign key to EMPLOYEE’s primary key Ssn, referential integrity constraint |
| Certificate | Certificates of being technical in specific area | Character string with length up to 30 | NOT NULL |

MANAGER table represents manager who manage online book store, with the attributes table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Manager\_Ssn | Manager’s ssn | Character string with length 11 | NOT NULL,  Primary Key, Foreign key to EMPLOYEE’s primary key Ssn, referential integrity constraint |
| Skills | Skills that the manager has | Character string with length up to 50 | NOT NULL |

WRITE table represents the relation between AUTHOR and BOOK tables, one author can write many books and one book can have many authors. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Wid | Author id that uniquely identify each author | Character string with length 9 | NOT NULL,  Primary Key,  Foreign key to AUTHOR’s primary key Id, referential integrity constraint |
| WISBN | Book’s ISBN | Character string with length up to 20 | NOT NULL,  Primary Key,  Foreign key to BOOK’s primary key ISBN,  referential integrity constraint |

REQUEST\_SERVICE table represents the relation between CUSTOMER and CUSTOMER\_SERVICE tables. Each tuple is a service record. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| SEmail | Email of customer who requests services | Character string with length up to 30 | NOT NULL,  Foreign key to CUSTOMER’s primary key Email,  referential integrity constraint |
| Service\_code | Unique code that made up by the online book store and uniquely identify each service | Character string with length up to 4 | NOT NULL,  Primary Key. |
| Date | The date that customer requests services. | Character string with length 10 | NOT NULL |
| Request\_Ssn | Ssn of the employee in CUSTOMER\_SERVICE who accepts the request | Character string with length 11 | NOT NULL,  Foreign key to EMPLOYEE’s primary key Ssn,  referential integrity constraint |

CHECK\_ORDER table represents the relation between CUSTOMER\_SERVICE and BOOK\_ORDER tables. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Check\_Ssn | Ssn of Employee who check the order | Character string with length 11 | NOT NULL,  Primary key,  Foreign key to CUSTOMER\_SERVICE’s primary key Service\_Ssn, referential integrity constraint |
| Check\_Quantity | The total quantity of books that is contained in the order that the employee checked | Integer | NOT NULL |
| Check\_Date | The date that customer service employee checked the book | Character string with length 10 | NOT NULL |
| Check\_Order\_no | Order number | Character string with length up to 20 | NOT NULL,  Primary key,  Foreign key to BOOK\_ORDER’s primary key Order\_no,  referential integrity constraint |

BOOK\_ORDER table represents all the book orders that made by customers in the book store. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| CusEmail | Customer’s email who made book order | Character string with length up to 30 | NOT NULL,  Foreign key to CUSTOMER’s primary key Email,  referential integrity constraint |
| Order\_no | Order number | Character string with length 20 | NOT NULL,  primary key, |
| Date | The date that customer made the book order | Character string with length 10 | NOT NULL |

CONTAIN table represents the relationship of tables BOOK and BOOK\_ORDER. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| CISBN | The ISBN of books that customers bought | Character string with length up to 20 | NOT NULL,  Foreign key to BOOK’s primary key ISBN,  referential integrity constraint |
| ConOrder\_no | Order number | Character string with length 20 | NOT NULL,  primary key,  Foreign key to BOOK\_ORDER’s primary key Order\_no,  referential integrity constraint |
| CQuantity | The quantity of the book that is contained in a book order | Integer | NOT NULL |

SALE table represents the sales information about books. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Promo\_code | The code for particular sale made up by the book store. | Character string with length up to 30 | NOT NULL,  primary key, |
| Discount\_percent | Percent off of the original price. | Integer | NOT NULL |

BOOK table represents the books that sold by the book store. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| ISBN | The ISBN of books | Character string with length up to 20 | NOT NULL,  primary key, |
| W\_tax\_id | The tax id of the warehouse which stores the book | Character string with length 4 | NOT NULL,  Foreign key to WAREHOUSE’s primary key Ware\_Tax\_Id,  referential integrity constraint |
| Category | Books’ category | Character string with length up to 20 | NOT NULL |
| Price | The price of the book | Double with two decimal places | NOT NULL |
| Version | Version of the book | Character string with length up to 5 | NUT NULL |
| Title | Title of the book | Character string with length up to 50 | NOT NULL |
| Quantity | Quantity of the book that stored in the warehouse | Integer | NOT NULL |

LIVE table represents the relationship of tables CUSTOMER,AUTHOER,EMPLOYEE and ADDRESS, matching the customer's’ address. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Drive\_no | Customer’s, author’s,employee’s driver's license number | Character string with length 8 | NOT NULL,  primary key,  Foreign key to CUSTOMER’s primary key CusLicense\_no,  Foreign key to EMPLOYEE’s primary key EM\_License\_no,  Foreign key to AUTHOR’s primary key AU\_License\_no, |
| Live\_id | The living area id of customer, made up by the book store | Character string with length 4 | NOT NULL,  primary key,  Foreign key to ADDRESS’s primary key Aid,  referential integrity constraint |

PAYMENT table represents the relationship of tables CUSTOMER and ORDER, matching the customer's’ order number. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Amount | The total amount that customer need to pay | Double with two decimal places | NOT NULL, |
| POrder\_no | The order number that customer paid for | Character string with length up to 20 | NOT NULL,  Primary key,  Foreign key to BOOK\_ORDER’s primary key Order\_no, |
| Security\_code | The security code of customer’s card that they used to pay | Character string with length 3 | NOT NULL |
| Billing\_zip | Customer’s billing address’s zip | Double with two decimal places | NOT NULL |
| Card\_number | The card number that customer used to pay | Character string with length 20 | NUT NULL,  Primary key, |

SHIPMENT table represents the relationship of tables CUSTOMER and ADDRESS, matching the customer's’ address. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Type | The shipping type and speed that customer requested | Character string with length up to 30 | NOT NULL,  Primary key, |
| SOrder\_no | The order number that customer made | Character string with length up to 20 | NOT NULL,  Primary key,  Foreign key to BOOK\_ORDER’s primary key Order\_no, |
| Tax\_Id | Tax id of post office that received this shipment | Character string with length 4 | NOT NULL,  Foreign key to POST\_OFFICE’s primary key Tax\_Id, |
| Last\_location\_tracking | The most recent location of this package | Character string with length up to 50 | NOT NULL, |
| Cost | Shipping cost | Double with two decimal places | NUT NULL, |

SALE\_DETAIL table represents the relationship of tables SALE and BOOK, giving the detail of sales books. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| SPromo\_Code | The code for particular sale made up by the book store. | Character string with length up to 30 | NOT NULL,  primary  key,  Foreign key to SALE’s primary key Promo\_Code, |
| Start\_date | The sale’s start date | Character string with length 10 | NOT NULL, |
| End\_date | The sale’s end date | Character string with length 10 | NOT NULL, |
| SISBN | The sales books’ ISBN | Character string with length up to 20 | NOT NULL,  Primary key,  Foreign key to BOOK’s primary key ISBN, |
| Amount | The amount of books that are sold | Integer | NUT NULL, |

PUBLISH table represents the relationship of tables PUBLISHER and BOOK, giving the detail of  books that published by publishers. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Publish\_Tax\_Id | Tax id for publisher | Character string with length 4 | NOT NULL,  primary key,  Foreign key to PUBLISHER’s primary key Publisher\_Tax\_Id, |
| Date | The date that the book was published | Character string with length 10 | NOT NULL, |
| PubISBN | the ISBN of published books | Character string with length up to 20 | NOT NULL,  Primary key,  Foreign key to BOOK’s primary key ISBN, |
| Amount | The amount of books that going to publish | Integer | NUT NULL, |

PUBLISHER table represents the book publisher in the real world, giving the detail of  books that published by publishers. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Publisher\_Tax\_Id | Tax id for publisher | Character string with length up to 4 | NOT NULL,  primary key, |
| Phone | The phone numebr of the publisher | Character string with length 10 | NOT NULL, |
| Pub\_Name | the name of the publisher | Character string with length up to 30 | NOT NULL, |
| Pub\_Id | The address id of the publisher | Character string with length up to 4 | NUT NULL,Foreign key to ADDRESS’s primary key Aid,  referential integrity constraint |

ADDRESS table represents the addresses in the real world. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Aid | Address id that made up by the book store | Character string with length up to 4 | NOT NULL,  primary key, |
| Zip | Zip code of the address | Character string with length 5 | NOT NULL, |
| Street | Street and number of the address | Character string with length up to 40 | NOT NULL, |

WAREHOUSE table represents all the warehouses that the book store stores book in. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Ware\_Tax\_Id | Warehouse’s tax id | Character string with length up to 4 | NOT NULL,  Primary key, |
| Ware\_Name | Warehouse’s name | Character string with length up to 30 | NOT NULL |
| Stock | Current stocking number of books in warehouse | Integer | NOT NULL |
| Capacity | Maximum storage | Integer | NOT NULL |
| Mgr\_Ssn | Ssn of manager who manages the warehouse | Character string with length 11 | NOT NULL,  Foreign key to EMPLOYEE’s primary key Ssn,  referential integrity constraint |
| Phone | Phone number of warehouse | Character string with length 10 | NOT NULL |
| Wid | Address id of warehouse | Character string with length 4 | NOT NULL,  Foreign key to ADDRESS’s primary key Aid,  referential integrity constraint |

SUPPLIER table represents all the supplies that supply book to the book store. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Supplier\_Tax\_Id | The tax id of the supplier | Character string with length 4 | NOT NULL,  primary key, |
| Sup\_Name | Supplier’s name | Character string with length up to 30 | NOT NULL |
| Phone | Phone number of suppliers | Character string with length 10 | NOT NULL |
| Sup\_Id | Address id of supplyer | Character string with length 4 | NOT NULL, Foreign key to ADDRESS’s primary key Aid,  referential integrity constraint |

SUPPLY table represents the relationship between tables SUPPLIER and WAREHOUSE. SUPPLIER supplies WAREHOUSE books. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Store\_Tax\_Id | The tax id of the warehouse | Character string with length up to 4 | NOT NULL,  primary key,  Foreign key to WAREHOUSE’s primary key Ware\_Tax\_Id |
| S\_Tax\_Id | Supplier’s tax id | Character string with length up to 4 | NOT NULL,  primary key,  Foreign key to SUPPLIER’s primary key Supplier\_Tax\_Id |
| Amount | The amount of books that supplier provided to the warehouse | Integer | NOT NULL |
| Sup\_ISBN | The ISBN of the book that supplier provided to the warehouse | Character string with length up to 20 | NOT NULL,  primary key |

POST\_OFFICE table represents the post offices that make those shipment to customers. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Tax\_Id | Tax id of post office that make the shipment | Character string with length 4 | NOT NULL,  primary key |
| Post\_Name | Post office’s name | Character string with length up to 30 | NOT NULL |
| Phone | Phone number of the post office | Character string with length 10 | NOT NULL |
| Po\_Id | Address id of post office | Character string with length 4 | NOT NULL,  Foreign key to ADDRESS’s primary key Aid,  referential integrity constraint |

PROVIDE table shows the relationship between tables PUBLISHER and SUPPLIER. Publisher provide suppliers number of books. The attributes table is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Attribute Meaning | Data Type | Constraint |
| Sup\_Tax\_Id | The tax id of the supplier | Character string with length 4 | NOT NULL,  primary key,  Foreign key to SUPPLIER’s primary key Supplier\_Tax\_Id,  referential integrity constraint |
| Pub\_Tax\_Id | Tax id for publisher | Character string with length 4 | NOT NULL,  primary key,  Foreign key to PUBLISHER’s primary key Publisher\_Tax\_Id,  referential integrity constraint |
| Amount | The amount of books that supplier provided to the warehouse | Integer | NOT NULL |

B.

a. Find the titles of all books by Pratchett that cost less than $10

Pra←(𝝈LName=Pratchett ( AUTHOR⨝ Id=WId WRITE))

𝝅Title(𝝈Price<10( Pra⨝ Pra.WISBN=BOOK.ISBN BOOK))

SQL: SELECT Title

FROM BOOK AS B, AUTHOR AS A, WRITE AS W

WHERE B.Price<10 AND A.Id=W.Wid AND W.WISBN=B.ISBN

 AND A.LName='Pratchett';

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

assume the email of the customer is a@osu.edu

Allorder←𝝅Order\_no,Date(𝝈Email=a@osu.edu ( CUSTOMER⨝ CUSTOMER.Email=BOOKORDER.CusEmail BOOKORDER))

Allbook←𝝅Date,CISBN(Allorder ⨝ Allorder.Order\_no=CONTAIN.ConOrder\_no CONTAIN)

                𝝅Date,Title(Allbook ⨝Allbook.CISBN=BOOK.ISBN BOOK)

SQL: SELECT  Title, Date

FROM BOOK\_ORDER, CONTAIN, BOOK

            WHERE CusEmail='a@osu.edu' AND Order\_no=ConOrder\_no

                           AND CISBN=ISBN;

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

                    Lesscopy←𝝈Quantity<5(BOOK)

                       𝝅ISBN,Title(Lesscopy)

SQL: SELECT Title, ISBN

FROM  WAREHOUSE, BOOK

            WHERE Ware\_Tax\_id = W\_Tax\_id AND Quantity < 5;

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

Authorid←𝝈LName=Pratchett( AUTHOR)

Bookcode←𝝅WISBN(Authorid ⨝Authorid.Id=WRITE.WId WRITE)

BookName←𝝅ISBN,Title(BOOK ⨝ BOOK.ISBN=Bookcode.WISBN Bookcode)

        Order←𝝅Title,ConOrder\_no(BookName ⨝ BookName.ISBN=CONTAIN.CISBN CONTAIN)

        ED←𝝅Title,CusEmail(Bookorder ⨝Bookorder.Order\_no=ORDER.ConOrder\_no ORDER)

                 𝝅FName,LNameTitle (CUSTOMER ⨝CUSTOMER.Email=ED.CusEmail ED)

SQL: SELECT CUSTOMER.FName, CUSTOMER.LName,Title

            FROM CUSTOMER, BOOK\_ORDER, CONTAIN,BOOK, WRITE,AUTHOR

            WHERE  AUTHOR.LName='Pratchett'  AND Email=CusEmail AND Order\_no= ConOrder\_no AND CISBN=ISBN AND ISBN=WISBN AND Wid=Id;

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

assume the email of the customer is a@osu.edu

 Allorder←𝝅Order\_no(𝝈CusEmail= a@osu.edu ( BOOKORDER))

ƑSUM CQuantity((Allorder ⨝ Allorder.Order\_no=CONTAIN.ConOrder\_no CONTAIN))

SELECT SUM(CQuantity)

FROM BOOK\_ORDER, CONTAIN

WHERE  CusEmail='a@osu.edu' AND Order\_no=ConOrder\_no;

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

 Allorder←𝝅FName,LName,Email,Order\_no(BOOKORDER ⨝BOOKORDER.CusEmail=CUSTOMER.Email CUSTOMER)

Order←𝝅Email,CQuantity, FName,LName, (Allorder ⨝Allorder.Order\_no= CONTAIN.ConOrder\_no CONTAIN)

Total←Email, FName,LName ƑSUM CQuantity(Order)

SQL: SELECT CusEmail,SUM(CQuantity)

FROM CONTAIN,BOOK\_ORDER

WHERE Order\_no=ConOrder\_no

GROUP BY CusEmail

HAVING SUM(CQuantity)=

     (SELECT MAX(max)

               FROM( SELECT SUM(CQuantity) as max

                       FROM BOOK\_ORDER, CONTAIN

                       WHERE Order\_no=ConOrder\_no GROUP BY CusEmail));

Query 1: Find the discount percentage of all the sales and their start date and end date

𝝅Discount\_percent,Start\_date,End\_date(SALE ⨝SALE.Promo\_code=SALE\_DETAIL.SPromo\_code SALE\_DETAIL)

SQL: SELECT Discount\_percent,Start\_date,End\_date

FROM SALE, SALE\_DETAIL

WHERE Promo\_code=SPromo\_code;

Query 2: Find the name of the publisher that provides the supplier with the largest amount of book

Pub←𝝅Publisher\_Tax\_Id,PubName,Amount(PROVIDE ⨝PROVIDE.Pub\_Tax\_Id= PUBLISHER.Publisher\_Tax\_Id PUBLISHER);

Total← Publisher\_Tax\_Id,PubName, ƑSUM Amount(Pub)

𝝅PubName (Total)

SQL: SELECT Pub\_Name,SUM(Amount)

FROM PROVIDE,PUBLISHER

WHERE Publisher\_Tax\_Id=Pub\_Tax\_Id

GROUP BY Pub\_Tax\_Id

HAVING SUM(Amount)=(SELECT MAX(sum)

FROM (SELECT SUM(Amount) AS sum

FROM PROVIDE,PUBLISHER

WHERE Publisher\_Tax\_Id=Pub\_Tax\_Id

GROUP BY Pub\_Tax\_Id));

Query 3: Given the tax id of a warehouse, find the name of the manager who manages it.

People←𝝅Mgr\_Ssn(𝝈Ware\_Tax\_Id=123(WAREHOUSE))

T2←𝝅 Mgr\_Ssn (Manager ⨝Manager\_Ssn=People. Mgr\_Ssn People)

𝝅 FName,LName(T2 ⨝T2. Mgr\_Ssn =EMPLOYEE. EMPLOYEE)

SQL: SELECT FName,LName

FROM WAREHOUSE, MANAGER, EMPLOYEE

WHERE  Ware\_Tax\_id='123' AND Mgr\_Ssn=Manager\_Ssn And Manager\_Ssn=Ssn;

C.

**Instruction for adding new books:**

* Attributes:

1. The ISBN of this book as its primary key;
2. The Tax ID of the warehouse that stores this book, this Tax ID must reference to one of the existed Ware\_Tax\_Id for warehouses;
3. The category of this this book, cannot be null;
4. The price of this book, recorded as a double, cannot be null;
5. The version of this book, i.e the edition of it, cannot be null;
6. The title/ name of this book, cannot be null;
7. The total amount of this book in stock, represented as an integer, cannot be null.

* Restrictions: when adding a book into BOOK, please update AUTHOR, WRITE and PUBLISH according to the information of the new book. The order of the attributes matter when insert into table. If the author of this book is not currently in the AUTHOR table, insert this author’s data and update the related relation LIVE and WRITE according to the instructions below. If the publisher of this book is not existed in the PUBLISHER table, also add the information of this publisher and update the relation PUBLISH.
* Example:

INSERT INTO BOOK

VALUES ('72227710','123','Computer','17.49','1st','How To Do Everything with Your Tablet PC','300');

**Instruction for adding new publisher:**

* Attributes:

1. The tax id of this publisher as its primary key, cannot be null;
2. The name of this publisher, no more than 30 characters, cannot be null;
3. The phone number to contact this publisher, cannot be null;
4. The unique address id to identify the location, must reference to an existed address id in the ADDRESS table

* Restrictions: when adding a new publisher, update the relation table PUBLISH to record all the books published by this publisher. The order of attributes matter when inserting the data. The instruction for add new tuple into PUBLISH is listed below.
* Example:

INSERT INTO PUBLISHER

VALUES (‘568’,'Schmidt Ink, Inc.','3129688488','124');

**Instruction for adding new author:**

* Attributes:

1. The Id of this author, i.e this author’s SSN number as the primary key, cannot be null;
2. The birth date of this author, represented as a string with 8 characters, cannot be null;
3. The author’s first name, no more than20 characters, cannot be null;
4. The author’s last name, no more than 20 characters, cannot be null;
5. The author’s gender, either M or F, cannot be null;
6. The phone that can contact this author, cannot be null;
7. This author’s unique driver license number, represented as a string with 8 characters, cannot be null;
8. The address id identify this author’s address, must references to one of the existed address id in the ADDRESS table.

* Restrictions: when add a new author, update tables WRITE, LIVE if necessary. The order of attributes matter when inserting the data. According to the instructions provided, also add the books written by this author.
* Example:

INSERT INTO AUTHOR

VALUES ('534505273', '19350305', 'Larry', 'Bossidy', 'M', '8577289966', 'US350760', '114');

**Instruction for adding new customer:**

* Attributes:

1. The email that this customer use for registration, recorded as the primary key, no more than 30 characters, cannot be null;
2. This customer’s interests of reading, no more than 30 characters, can be null;
3. This customer’s birth date, represented with 8 characters, cannot be null;
4. Customer’s first name, no more than 20 characters, cannot be null;
5. Customer’s last name, no more than 20 characters, cannot be null;
6. Customer’s gender, either M for male or F for female, cannot be null;
7. The phone number that can contact this customer, cannot be null;
8. The address id that identify this customer’s address, references to one of the existed address id in ADDRESS table;
9. Customer’s unique driver’s license number, represented with 8 characters.

* Restrictions: when adding a new customer, need to update the relation LIVE. The order of attributes matter when inserting data.
* Example:

INSERT INTO CUSTOMER

VALUES (['b@osu.edu](mailto:'b@osu.edu)', 'drama', '19800301', 'Peter', 'Harrt’, 'M', '6141111102', '012', 'US171073');

INSERT INTO LIVE

VALUES ('012','US171073');

**Instruction for adding new relation WRITE:**

* Attributes:

1. The Wid denotes the SSN number of the author, references to one of the existed Id in AUTHOR table, cannot be null;
2. The WISBN denotes the ISBN of a book, references to one of the existed book ISBN in table BOOK, cannot be null;

* Restrictions: the Wid and WISBN together forms the compound primary key. The order of attribute matter inserting a new relation.
* Example:

INSERT INTO WRITE

VALUES ('534505273', '72227710);

**Instruction for adding new relation PUBLISH:**

* Attributes:

1. The tax id of the publisher, references to one of the existed Publisher\_Tax\_Id in table PUBLISHER, cannot be null;
2. The ISBN of the book that gets publisher with the associated publisher, references to an existed ISBN in table BOOK, cannot be null;
3. The date that this book was published, represented in 10 characters, cannot be null;
4. The total number of copies that got published, cannot be null.

* Restrictions: The Publisher\_Tax\_Id and the PubISBN together forms the primary key. The order of attributes matter when inserting data.
* Example:

INSERT INTO PUBLISH

VALUES ('568','72227710','20030218','500');

**Instruction for adding new relation LIVE:**

* Attributes:

1. The address id references to an existed address is in table ADDRESS, cannot be null;
2. The unique driver’s license number references to an existed license number in either CUSTOMER, EMPLOYEE or AUTHOR, cannot be null.

* Restrictions: the address id and the license number together forms the primary key. The order of attributes matter when inserting date.
* Example:

INSERT INTO LIVE

VALUES ('114','US350760');

D.

**Instruction for deleting a book:**

* Attributes:

1. The ISBN of the book that you want to delete, must much with one of the existed ISBN in table BOOK.

* Restrictions: when deleting the book, also delete the tuples in table WRITE and table PUBLISH.
* Example:

DELETE FROM BOOK

WHERE   ISBN = '72227710';

DELETE FROM WRITE

WHERE   WISBN = '72227710';

DELETE FROM PUBLISH

WHERE   PubISBN = '72227710';

**Instruction for deleting a publisher:**

* Attributes:

1. The Publisher\_Tax\_Id that identify this publisher, must match to an existed Publisher\_Tax\_Id in table PUBLISHER.

* Restrictions: when deleting this publisher, delete the tuples in table PUBLISH that display all the books published by this publisher.
* Example:

DELETE FROM PUBLISHER

WHERE   Publisher\_Tax\_Id = '112';

DELETE FROM PUBLISH

WHERE  Publish\_Tax\_Id = '112';

**Instruction for deleting a author:**

* Attributes:

1. The SSN number of the author that you wish to delete, must match to an existed author SSN in AUTHOR table;
2. The unique driver’s license number of this author in order to delete related tuples in LIVE table.

* Restrictions: when delete this author, delete all the tuples in WRITE and LIVE that recorded the information of this author.
* Example:

DELETE FROM AUTHOR

WHERE  Id = '243243723' AND

              AU\_License\_no = 'US350760';

DELETE FROM WRITE

WHERE Wid = '243243723';

DELETE FROM LIVE

WHERE drive\_no = 'US350760';

**Instruction for deleting a customer:**

* Attributes:

1. The email address of the customer that you wish to delete, must match to an existed email address in CUSTOMER table;
2. The unique driver’s license number of this customer in order to delete related tuples in LIVE table.

* Restrictions: when delete this author, delete all the tuples in WRITE and LIVE that recorded the information of this author.
* Example:

DELETE FROM CUSTOMER

WHERE Email = ['b@osu.edu](mailto:'b@osu.edu)' AND

              CusLicense\_no = 'US171073';

DELETE FROM LIVE

WHERE drive\_no = 'US171073';

**Instruction for deleting tuple from relation WRITE:**

* Attributes:

1. The Wid that represent the SSB number of the author;
2. The ISBN of the book.

* Restrictions: either the Wid matches the SSN of the author you want to delete, or the ISBN matched to the book you want to delete.
* Example:

DELETE FROM WRITE

WHERE Wid = '243243723' OR

             WISBN = '72227710';

**Instruction for deleting tuple from relation PUBLISH:**

* Attributes:

1. The Publish\_Tax\_Id that represents the publisher you want to delete;
2. The PubISBN indicate the book you want to delete.

* Restrictions: either the PubISBN matched to the book you want to delete or the Publish\_Tax\_Id match to the publisher you want to delete.
* Example:

DELETE FROM PUBLISH

WHERE Publish\_Tax\_Id = '112' OR

             PubISBN = '72227710';

**Instruction for deleting tuple from relation LIVE:**

* Attributes:

1. The drive\_no represents the person (can be customer or author) you want to delete.

* Restrictions: the driver’s license number has to match either one of the customer’s driver’s license number or one of the author’s driver’s license number.
* Example:

DELETE FROM LIVE

WHERE   drive\_no = 'US171073';