UNIVERSITÉ Concordia UNIVERSITY

Warm Up Project

Brenden Bissessar 4044016

Pierre Watine 40027675

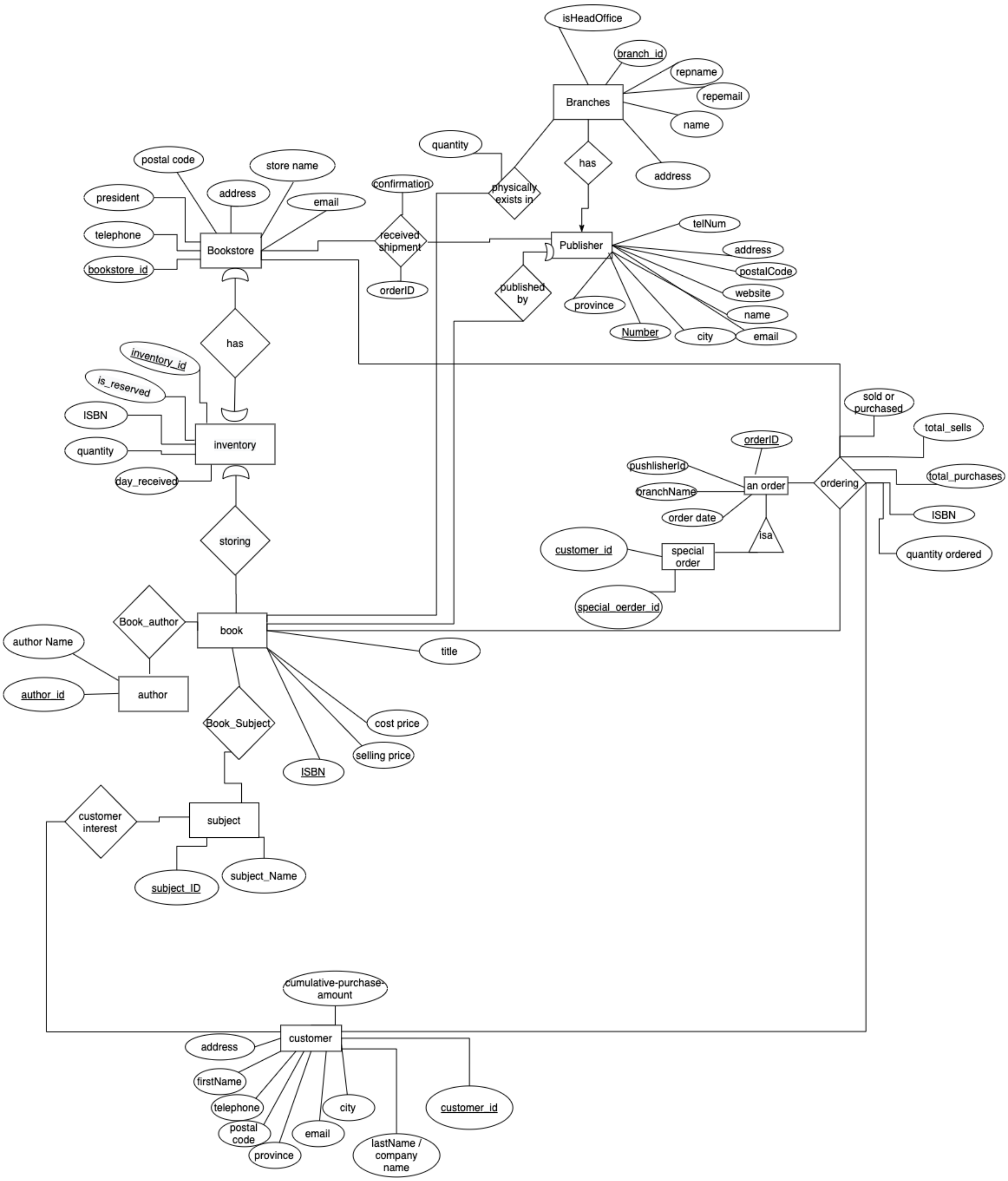
Kasra Laamerad 40020876

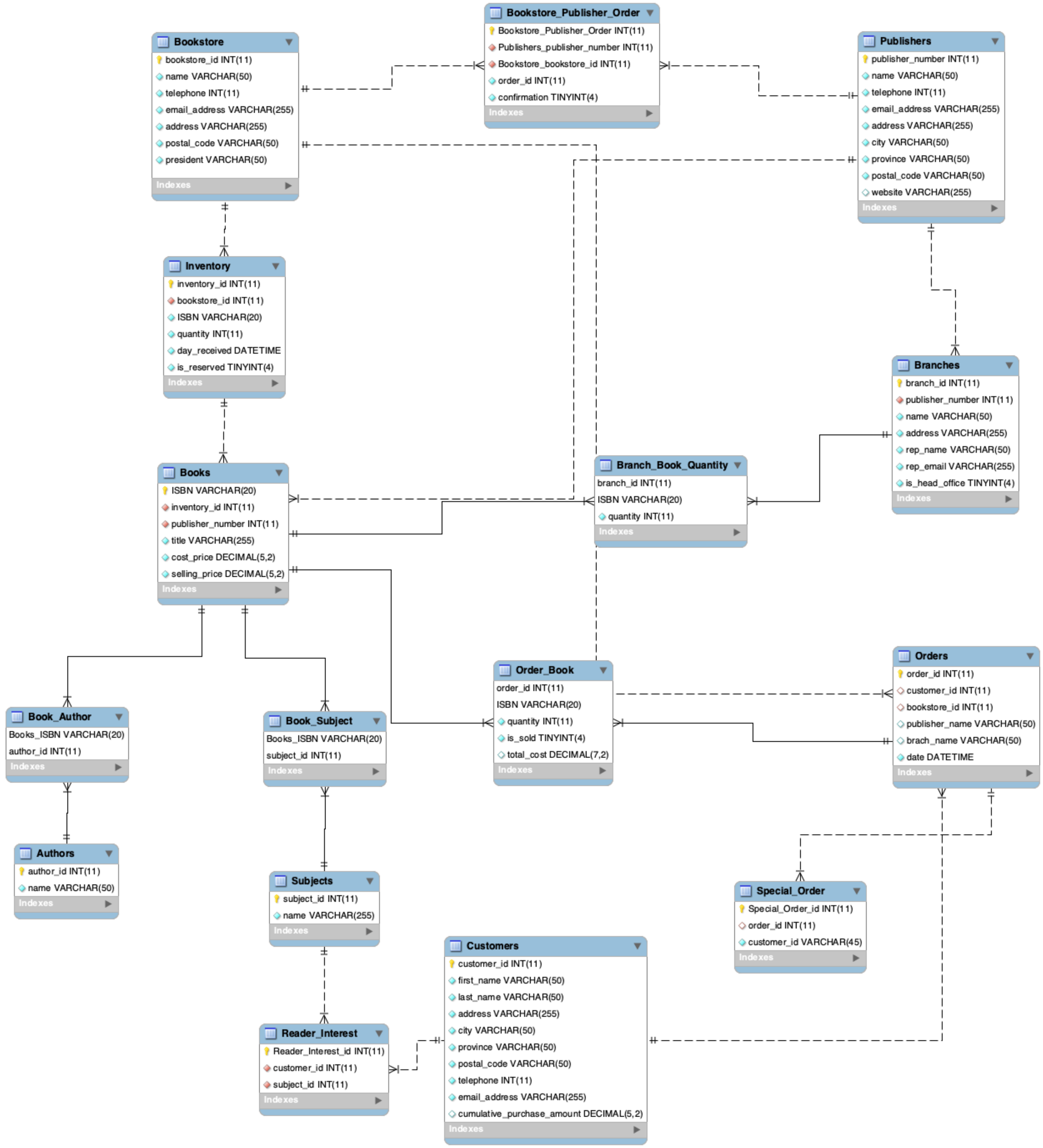
Evan Lamenta 27240007

Kien Nguyen 40055738

Khaled Jababo July 20th 2020

« We certify that this submission is our original work and meets the Faculty’s Expectations of Originality”


isHeadOffice
branch_id
repname
repemail
Branches
name
address
has
quantity
physically exists in
postal code
store name
president
address
email
confirmation
telephone
received shipment
Publisher
telNum
address
Bookstore
postalCode
bookstore_id
orderID
published by
website
province
name
Number
city
email
has
inventory_id
is_reserved
sold or purchased
total_sells
ISBN
inventory
orderID
quantity
pushlisherId
an order
ordering
total_purchases
day_received
branchName
order date
ISBN
isa
storing
customer_id
special order
quantity ordered
special_oerder_id
Book_author
book
author Name
title
author_id
author
Book_Subject
cost price
selling price
ISBN
customer interest
subject
subject_ID
subject_Name
cumulative-purchase-amount
customer
address
firstName
telephone
city
customer_id
postal code
email
province
lastName / company name


Bookstore_Publisher_Order
Bookstore_Publisher_Order INT(11)
Publishers_publisher_number INT(11)
Bookstore_bookstore_id INT(11)
order_id INT(11)
confirmation TINYINT(4)
Indexes
Bookstore
bookstore_id INT(11)
name VARCHAR(50)
telephone INT(11)
email_address VARCHAR(255)
address VARCHAR(255)
postal_code VARCHAR(50)
president VARCHAR(50)
Indexes
Publishers
publisher_number INT(11)
name VARCHAR(50)
telephone INT(11)
email_address VARCHAR(255)
address VARCHAR(255)
city VARCHAR(50)
province VARCHAR(50)
postal_code VARCHAR(50)
website VARCHAR(255)
Indexes
Inventory
inventory_id INT(11)
bookstore_id INT(11)
ISBN VARCHAR(20)
quantity INT(11)
day_received DATETIME
is_reserved TINYINT(4)
Indexes
Branches
branch_id INT(11)
publisher_number INT(11)
name VARCHAR(50)
address VARCHAR(255)
rep_name VARCHAR(50)
rep_email VARCHAR(255)
is_head_office TINYINT(4)
Indexes
Books
ISBN VARCHAR(20)
inventory_id INT(11)
publisher_number INT(11)
title VARCHAR(255)
cost_price DECIMAL(5,2)
selling_price DECIMAL(5,2)
Indexes
Branch_Book_Quantity
branch_id INT(11)
ISBN VARCHAR(20)
quantity INT(11)
Indexes
Orders
order_id INT(11)
customer_id INT(11)
bookstore_id INT(11)
publisher_name VARCHAR(50)
brach_name VARCHAR(50)
date DATETIME
Indexes
Order_Book
order_id INT(11)
ISBN VARCHAR(20)
quantity INT(11)
is_sold TINYINT(4)
total_cost DECIMAL(7,2)
Indexes
Book_Author
Books_ISBN VARCHAR(20)
author_id INT(11)
Indexes
Book_Subject
Books_ISBN VARCHAR(20)
subject_id INT(11)
Indexes
Authors
author_id INT(11)
name VARCHAR(50)
Indexes
Subjects
subject_id INT(11)
name VARCHAR(255)
Indexes
Special_Order
Special_Order_id INT(11)
order_id INT(11)
customer_id VARCHAR(45)
Indexes
Customers
customer_id INT(11)
first_name VARCHAR(50)
last_name VARCHAR(50)
address VARCHAR(255)
city VARCHAR(50)
province VARCHAR(50)
postal_code VARCHAR(50)
telephone INT(11)
email_address VARCHAR(255)
cumulative_purchase_amount DECIMAL(5,2)
Indexes
Reader_Interest
Reader_Interest_id INT(11)
customer_id INT(11)
subject_id INT(11)
Indexes

Functions

This function checks the quantities of the book in the inventory and puts the customer inside the special_Order table if the bookstore does not have that book inside its inventory.
It will also store the customers’ interest in the subject of the book that they bought.

```
DELIMITER $$

CREATE FUNCTION CustomerOrder(orderID INT, customerID INT)
RETURNS INT
DETERMINISTIC
BEGIN
  DECLARE order_status INT;

  SET order_status = (
                        SELECT EXISTS(
                                SELECT *
                                FROM (
                                        SELECT CASE
                                                WHEN inventory.quantity >= Order_Book.quantity
                                                        THEN 10
                                                        ELSE 20
                                                        END as Saleable
                                        FROM Inventory
                                        JOIN Books ON Books.ISBN = Inventory.ISBN
                                        JOIN Order_Book ON Books.ISBN = Order_Book.ISBN
                                        WHERE order_id = orderID
                                        ) AS a
                                WHERE 15 > ALL(SELECT Saleable)
                                )
                        )
        ;

  IF order_status = 0 THEN
    INSERT INTO Special_Order(order_id,customer_id)
    VALUES(orderID,customerID);
  END IF;

  INSERT INTO Reader_Interest(customer_id,subject_id)
        SELECT customerID, subject_id
        FROM Book_Subject
        JOIN Books ON Book_Subject.Books_ISBN = Books.ISBN
        JOIN Order_Book ON Books.ISBN = Order_Book.ISBN;

        RETURN (customerID);
END$$
DELIMITER ;
```

The following function calculates the total cost based on the customers' or bookstore's order.

```
DELIMITER $$
CREATE FUNCTION sell(isbn1 VARCHAR(20), qty INT, status INT)
RETURNS DECIMAL(7,2)
READS SQL DATA
DETERMINISTIC
BEGIN
   DECLARE totalCost DECIMAL(7,2);

          IF status = 1 THEN
                    SET totalCost = (SELECT qty * selling_price
                                                        FROM Books
                                                        WHERE Books.ISBN = isbn1);
   ELSE
                    SET totalCost = (SELECT qty * cost_price
                                                                  FROM Books
                                                                  WHERE Books.ISBN = isbn1);
   END IF;

          RETURN (totalCost);
END$$
DELIMITER ;

DELIMITER $$
```

Create Tables:

```
#CREATE DATABASE eyc353_1;

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema store
-- -----------------------------------------------------


-- -----------------------------------------------------
-- Schema store
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `store` DEFAULT CHARACTER SET utf8 ;
USE `eyc353_1` ;

-- -----------------------------------------------------
-- Table `store`.`Bookstore`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Bookstore` (
  `bookstore_id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  `telephone` INT NOT NULL,
  `email_address` VARCHAR(255) NOT NULL,
  `address` VARCHAR(255) NOT NULL,
  `postal_code` VARCHAR(50) NOT NULL,
  `president` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`bookstore_id`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `store`.`Inventory`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Inventory` (
  `inventory_id` INT NOT NULL AUTO_INCREMENT,
  `bookstore_id` INT NOT NULL,
  `ISBN` VARCHAR(20) NOT NULL,
  `quantity` INT NOT NULL,
  `day_received` DATETIME NOT NULL,
  `is_reserved` TINYINT NOT NULL,
  PRIMARY KEY (`inventory_id`),
  CONSTRAINT `fk_Inventory_Bookstore`
    FOREIGN KEY (`bookstore_id`)
    REFERENCES `store`.`Bookstore` (`bookstore_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Publishers`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Publishers` (
  `publisher_number` INT NOT NULL,
  `name` VARCHAR(50) NOT NULL,
  `telephone` INT NOT NULL,
  `email_address` VARCHAR(255) NOT NULL,
  `address` VARCHAR(255) NOT NULL,
  `city` VARCHAR(50) NOT NULL,
  `province` VARCHAR(50) NOT NULL,
  `postal_code` VARCHAR(50) NOT NULL,
  `website` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`publisher_number`))
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Books`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Books` (
  `ISBN` VARCHAR(20) NOT NULL,
  `inventory_id` INT NOT NULL,
  `publisher_number` INT NOT NULL,
  `title` VARCHAR(255) NOT NULL,
  `cost_price` DECIMAL(5,2) NOT NULL,
  `selling_price` DECIMAL(5,2) NOT NULL,
  INDEX `fk_Books_Intentory1_idx` (`inventory_id` ASC),
  PRIMARY KEY (`ISBN`),
  INDEX `fk_Books_Publishers1_idx` (`publisher_number` ASC),
  CONSTRAINT `fk_Books_Intentory1`
    FOREIGN KEY (`inventory_id`)
    REFERENCES `store`.`Intentory` (`inventory_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Books_Publishers1`
    FOREIGN KEY (`publisher_number`)
    REFERENCES `store`.`Publishers` (`publisher_number`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Authors`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Authors` (
  `author_id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`author_id`))
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Subjects`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Subjects` (
  `subject_id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`subject_id`))
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Book_Author`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Book_Author` (
  `Books_ISBN` VARCHAR(20) NOT NULL,
  `author_id` INT NOT NULL,
  PRIMARY KEY (`Books_ISBN`, `author_id`),
  INDEX `fk_Book_Author_Books1_idx` (`Books_ISBN` ASC),
  INDEX `fk_Book_Author_Authors1_idx` (`author_id` ASC),
  CONSTRAINT `fk_Book_Author_Books1`
    FOREIGN KEY (`Books_ISBN`)
    REFERENCES `store`.`Books` (`ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Book_Author_Authors1`
    FOREIGN KEY (`author_id`)
    REFERENCES `store`.`Authors` (`author_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Book_Subject`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Book_Subject` (
  `Books_ISBN` VARCHAR(20) NOT NULL,
  `subject_id` INT NOT NULL,
  PRIMARY KEY (`Books_ISBN`, `subject_id`),
  INDEX `fk_Book_Subject_Books1_idx` (`Books_ISBN` ASC),
  INDEX `fk_Book_Subject_Subjects1_idx` (`subject_id` ASC),
  CONSTRAINT `fk_Book_Subject_Books1`
    FOREIGN KEY (`Books_ISBN`)
    REFERENCES `store`.`Books` (`ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Book_Subject_Subjects1`
    FOREIGN KEY (`subject_id`)
    REFERENCES `store`.`Subjects` (`subject_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Customers`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Customers` (
  `customer_id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(50) NOT NULL,
  `last_name` VARCHAR(50) NOT NULL,
  `address` VARCHAR(255) NOT NULL,
  `city` VARCHAR(50) NOT NULL,
  `province` VARCHAR(50) NOT NULL,
  `postal_code` VARCHAR(50) NOT NULL,
  `telephone` INT NOT NULL,
  `email_address` VARCHAR(255) NOT NULL,
  `cumulative_purchase_amount` DECIMAL(5,2) NULL DEFAULT 0,
  PRIMARY KEY (`customer_id`))
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Orders` (
  `order_id` INT NOT NULL AUTO_INCREMENT,
  `customer_id` INT NULL,
  `bookstore_id` INT NULL,
  `publisher_name` VARCHAR(50) NULL DEFAULT NULL,
  `brach_name` VARCHAR(50) NULL DEFAULT NULL,
  `date` DATETIME NOT NULL,
  PRIMARY KEY (`order_id`),
  INDEX `fk_Orders_Customers1_idx` (`customer_id` ASC),
  INDEX `fk_Orders_Bookstore1_idx` (`bookstore_id` ASC),
  CONSTRAINT `fk_Orders_Customers1`
    FOREIGN KEY (`customer_id`)
    REFERENCES `store`.`Customers` (`customer_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Orders_Bookstore1`
    FOREIGN KEY (`bookstore_id`)
    REFERENCES `store`.`Bookstore` (`bookstore_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Order_Book`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Order_Book` (
  `order_id` INT NOT NULL,
  `ISBN` VARCHAR(20) NOT NULL,
  `quantity` INT NOT NULL,
  `is_sold` TINYINT NOT NULL,
  `total_cost` DECIMAL(7,2) NULL DEFAULT 0,
  INDEX `fk_Order_Book_Orders1_idx` (`order_id` ASC),
  PRIMARY KEY (`order_id`, `ISBN`),
  INDEX `fk_Purchase_Book_Books1_idx` (`ISBN` ASC),
  CONSTRAINT `fk_Order_Book_Orders1`
    FOREIGN KEY (`order_id`)
    REFERENCES `store`.`Orders` (`order_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Purchase_Book_Books1`
    FOREIGN KEY (`ISBN`)
    REFERENCES `store`.`Books` (`ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Bookstore_Publisher_Order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Bookstore_Publisher_Order` (
  `Bookstore_Publisher_Order` INT NOT NULL AUTO_INCREMENT,
  `Publishers_publisher_number` INT NOT NULL,
  `Bookstore_bookstore_id` INT NOT NULL,
  `order_id` INT NOT NULL,
  `confirmation` TINYINT NOT NULL,
  INDEX `fk_Bookstore_Publisher_Order_Publishers1_idx` (`Publishers_publisher_number` ASC),
  INDEX `fk_Bookstore_Publisher_Order_Bookstore1_idx` (`Bookstore_bookstore_id` ASC),
  PRIMARY KEY (`Bookstore_Publisher_Order`),
  CONSTRAINT `fk_Bookstore_Publisher_Order_Publishers1`
    FOREIGN KEY (`Publishers_publisher_number`)
    REFERENCES `store`.`Publishers` (`publisher_number`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Bookstore_Publisher_Order_Bookstore1`
    FOREIGN KEY (`Bookstore_bookstore_id`)
    REFERENCES `store`.`Bookstore` (`bookstore_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `store`.`Special_Order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Special_Order` (
  `Special_Order_id` INT NOT NULL AUTO_INCREMENT,
  `order_id` INT NULL,
  `customer_id` VARCHAR(45) NOT NULL,
  INDEX `fk_Special_Order_Orders1_idx` (`order_id` ASC),
  PRIMARY KEY (`Special_Order_id`),
  CONSTRAINT `fk_Special_Order_Orders1`
    FOREIGN KEY (`order_id`)
    REFERENCES `store`.`Orders` (`order_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Branches`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Branches` (
  `branch_id` INT NOT NULL AUTO_INCREMENT,
  `publisher_number` INT NOT NULL,
  `name` VARCHAR(50) NOT NULL,
  `address` VARCHAR(255) NOT NULL,
  `rep_name` VARCHAR(50) NOT NULL,
  `rep_email` VARCHAR(255) NOT NULL,
  `is_head_office` TINYINT NOT NULL,
  PRIMARY KEY (`branch_id`),
  INDEX `fk_Branches_Publishers1_idx` (`publisher_number` ASC),
  CONSTRAINT `fk_Branches_Publishers1`
    FOREIGN KEY (`publisher_number`)
    REFERENCES `store`.`Publishers` (`publisher_number`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Branch_Book_Quantity`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Branch_Book_Quantity` (
  `branch_id` INT NOT NULL,
  `ISBN` VARCHAR(20) NOT NULL,
  `quantity` INT NOT NULL,
  INDEX `fk_Branch_Book_Quantity_Branches1_idx` (`branch_id` ASC),
  PRIMARY KEY (`ISBN`, `branch_id`),
  CONSTRAINT `fk_Branch_Book_Quantity_Branches1`
    FOREIGN KEY (`branch_id`)
    REFERENCES `store`.`Branches` (`branch_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Branch_Book_Quantity_Books1`
    FOREIGN KEY (`ISBN`)
    REFERENCES `store`.`Books` (`ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `store`.`Reader_Interest`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `store`.`Reader_Interest` (
  `Reader_Interest_id` INT NOT NULL AUTO_INCREMENT,
  `customer_id` INT NOT NULL,
  `subject_id` INT NOT NULL,
  PRIMARY KEY (`Reader_Interest_id`),
  INDEX `fk_Customer_Subject_Customers1_idx` (`customer_id` ASC),
  INDEX `fk_Customer_Subject_Subjects1_idx` (`subject_id` ASC),
  CONSTRAINT `fk_Customer_Subject_Customers1`
    FOREIGN KEY (`customer_id`)
    REFERENCES `store`.`Customers` (`customer_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Customer_Subject_Subjects1`
    FOREIGN KEY (`subject_id`)
    REFERENCES `store`.`Subjects` (`subject_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Populating Tables 10 Entries:

```
USE eyc353_1 ;

-- insert authors********************************************************************
 insert into Authors (name)
	values
		("A"),("B"),("C"),("D"),("E"),("F"),("G"),("H"),("I"),("M");

-- *************************************************************

insert into Bookstore (name, telephone, email_address, address, postal_code, president)
VALUES
	('c', 3333333, 'c@hotmail.com', '333 street', 'c11222', 'Yugi Motto'),
	('d', 4444444, 'd@hotmail.com', '444 street', 'd11222', 'Evan'),
	('e', 5555555, 'e@hotmail.com', '555 street', 'e11222', 'Goku'),
	('f', 6666666, 'f@hotmail.com', '666 street', 'f11222', 'Kien'),
	('g', 7777777, 'g@hotmail.com', '777 street', 'g11222', 'Kasra'),
	('h', 8888888, 'h@hotmail.com', '888 street', 'h11222', 'Pierre'),
	('i', 9999999, 'i@hotmail.com', '999 street', 'i11222', 'Kobe'),
   ('j', 0, 'j@hotmail.com', '000 street', 'j11222', 'Brenden');

-- insert publisher ******************************************************************

insert into
Publishers(publisher_number,name,telephone,email_address,address,city,province,postal_code,websit
e)
	values
		(1,"Ken_News1",438999990,"ken@publisher1.ca","123
westmount","westmount","quebec","h3h 1r7","www.kennews1.ca"),
		(2,"Ken_News2",438999991,"ken@publisher2.ca","124
westmount","westmount","quebec","h3h 1r8","www.kennews2.ca"),
      (3,"Ken_News3",438999992,"ken@publisher3.ca","153 westmount","westmount","quebec","h3h
1r9","www.kennews3.ca"),
      (4,"Ken_News4",438999993,"ken@publisher4.ca","163 westmount","westmount","quebec","h3h
1t7","www.kennews4.ca"),
      (5,"Ken_News5",438999994,"ken@publisher5.ca","173 westmount","westmount","quebec","h3h
1x1","www.kennews5.ca"),
      (6,"Ken_News6",438999995,"ken@publisher6.ca","183 westmount","westmount","quebec","h3h
1r3","www.kennews6.ca"),
      (7,"Ken_News7",438999996,"ken@publisher7.ca","193 westmount","westmount","quebec","h3h
1r4","www.kennews7.ca"),
        (8,"Ken_News8",438999997,"ken@publisher8.ca","423 westmount","westmount","quebec","h3h
1a1","www.kennews8.ca"),
        (9,"Ken_News9",438999998,"ken@publisher9.ca","223 westmount","westmount","quebec","h3h
1b2","www.kennews9.ca"),
        (10,"Ken_News10",438999999,"ken@publisher10.ca","495
westmount","westmount","quebec","h3h 1c9","www.kennews10.ca");

INSERT INTO Inventory(inventory_id, bookstore_id, ISBN, quantity, day_received, is_reserved)
 VALUES
            ('1', '1', '0-2237-3192-7', '1', '2020-01-02 00:00:00', '1'),
            ('2', '1', '0-2710-5457-3', '1', '2020-01-03 00:00:00', '1'),
            ('3', '1', '0-2838-9190-4', '1', '2020-01-04 00:00:00', '0'),
            ('4', '1', '0-3432-6145-6', '1', '2020-01-05 00:00:00', '0'),
            ('5', '1', '0-4193-9813-9', '1', '2020-01-06 00:00:00', '1'),
            ('6', '1', '0-4809-1234-3', '1', '2020-01-07 00:00:00', '0'),
            ('7', '1', '0-4838-6782-9', '1', '2020-01-08 00:00:00', '1'),
            ('8', '1', '0-6034-3910-1', '1', '2020-01-09 00:00:00', '1'),
            ('9', '1', '0-6521-1546-2', '2', '2020-01-10 00:00:00', '1'),
            ('10', '1', '0-1188-0050-7', '1', '2020-01-01 00:00:00', '0');


-- insert books info ******************
 insert into Books(ISBN,inventory_id,publisher_number,title,cost_price,selling_price)
            values
            ("0-7915-3557-6",1,5,"How to become millionare at 10",11.5,20),
        ("0-2838-9190-4",1,1,"How to become millionare at 11",12.5,21),
        ("0-4838-6782-9",1,3,"How to become millionare at 12",13.5,22),
        ("0-6521-1546-2",1,2,"How to become millionare at 13",14.5,23),
        ("0-7225-8364-8",1,1,"How to become millionare at 14",15.5,24),
        ("0-9583-3711-X",1,2,"How to become millionare at 15",16.5,25),
        ("0-8229-1390-9",1,3,"How to become millionare at 16",17.5,26),
        ("0-4193-9813-9",1,8,"How to become millionare at 17",18.5,27),
        ("0-2237-3192-7",1,9,"How to become millionare at 18",19.5,28),
        ("0-6950-6671-4",1,7,"How to become millionare at 19",20.5,29),
        ("0-1188-0050-7",1,8,"How to become millionare at 20",21.5,30),
        ("0-2710-5457-3",1,6,"How to become millionare at 21",22.5,31),
        ("0-4809-1234-3",1,7,"How to become millionare at 22",23.5,32),
        ("0-6034-3910-1",1,8,"How to become millionare at 23",24.5,33),
        ("0-3432-6145-6",1,1,"How to become millionare at 24",25.5,34),
        ("0-6904-3082-5",1,9,"How to become millionare at 25",26.5,35);
```

```
-- ************************************************************************************\
-- insert author for the books

 insert into Book_Author(Books_ISBN,author_id)
             values
                          ("0-7915-3557-6",1),
        ("0-7915-3557-6",2),
        ("0-7915-3557-6",3),
        ("0-2838-9190-4",1),
        ("0-4838-6782-9",4),
        ("0-6521-1546-2",4),
        ("0-7225-8364-8",5),
        ("0-9583-3711-X",5),
        ("0-9583-3711-X",6),
        ("0-9583-3711-X",7),
        ("0-8229-1390-9",6),
        ("0-4193-9813-9",7),
        ("0-2237-3192-7",8),
        ("0-6950-6671-4",8),
        ("0-1188-0050-7",8),
        ("0-2710-5457-3",9),
        ("0-2710-5457-3",8),
        ("0-2710-5457-3",7),
        ("0-2710-5457-3",6),
        ("0-4809-1234-3",3),
        ("0-6034-3910-1",8),
        ("0-3432-6145-6",9),
        ("0-6904-3082-5",10);
-- *******************************************************************
-- insert subjects table
 insert into Subjects(name)
             values
                          ("science"),
        ("science-fiction"),
        ("fiction"),
        ("self-help"),
        ("romance"),
        ("action"),
        ("adventure"),
        ("classic"),
        ("novel"),
        ("autobiography");
```

```
-- -- **************************************************

-- -- added subject to books
 insert into Book_Subject(books_isbn,subject_id)
            values
                         ("0-7915-3557-6",1),
        ("0-7915-3557-6",2),
        ("0-7915-3557-6",3),
        ("0-2838-9190-4",1),
        ("0-4838-6782-9",4),
        ("0-6521-1546-2",4),
        ("0-7225-8364-8",5),
        ("0-9583-3711-X",5),
                         ("0-9583-3711-X",6),
                         ("0-9583-3711-X",7),
        ("0-8229-1390-9",6),
        ("0-4193-9813-9",7),
        ("0-2237-3192-7",8),
        ("0-6950-6671-4",8),
        ("0-1188-0050-7",8),
        ("0-2710-5457-3",9),
        ("0-2710-5457-3",8),
        ("0-2710-5457-3",7),
        ("0-2710-5457-3",6),
        ("0-4809-1234-3",3),
        ("0-6034-3910-1",8),
        ("0-3432-6145-6",9),
        ("0-6904-3082-5",10);

-- ************************************************************

-- insert Branches
 insert into Branches(branch_id,publisher_number, name, address, rep_name, rep_email,
is_head_office)
  values (1,1,"Konami", "102 Greenview St. Winston Salem, NC 27103", "Hideo Kojima",
"h_kojima@gmail.com", 0),
   (2,1,"EA", "1 Arrowhead Ave. Bangor, ME 04401", "John Madden", "jm@ea.com", 0),
   (3,1,"Funimation", "8059 Prince St. Great Falls, MT 59404", "Janet Jackson",
"head_chief_jackson@funimation.com", 1),
   (4,1,"Penguin Killers", "85 Poplar St. Noblesville, IN 46060", "Michael Myers",
"totally_not_a_stalker@penguinkillers.uk", 0),
    (5,2,"Fake Acid Pools", "8840 West Dogwood St. Valdosta, GA 31601", "Rick Sanchez",
"the_rick@plumbus.org", 1),
    (6,2,"Nacho Libre", "288 Shipley Street Vienna, VA 22180", "Jack Black", "blackjack76@gmail.com", 0),
    (7,3,"Cow Chop", "35 North Court Torrance, CA 90505", "James Wilson",
"uberhaxornova@hotmail.com", 0),
    (8,3,"Flex Tape", "159 Cardinal St. Garfield, NJ 07026", "Phil Swift", "alottadamage@flex_seal.com",
1),
    (9,4,"I love trains", "9022 Iroquois St. Chapel Hill, NC 27516", "Sheldon Cooper",
"perfect_being@msn.com", 1),
    (10,5,"Speedwagon Inc.", "8232 Honey Creek St. Cedar Falls, IA 50613", "Robert E. O. Speedwagon",
"checkered_hat@speedwagon.us", 1);

-- *************************************************************

-- insert Branch_Book_Quantity
 insert into Branch_Book_Quantity(branch_id,ISBN,quantity)
  values (1,"0-1188-0050-7",10),
    (1,"0-4809-1234-3",10),
    (3,"0-2838-9190-4",30),
    (9,"0-3432-6145-6",5),
    (4,"0-8229-1390-9",12),
    (10,"0-6521-1546-2",10),
    (4,"0-1188-0050-7",2),
    (5,"0-3432-6145-6",45);

-- insert Bookstore_Publisher_Order
*****************************************************************

 insert into Bookstore_Publisher_Order(Publishers_publisher_number, Bookstore_bookstore_id,
order_id,confirmation)
  values (1,1,1,0),
    (2,1,2,1),
    (1,1,3,0),
    (9,1,4,0),
    (4,1,5,1),
    (10,1,6,0),
    (6,1,7,1),
    (5,1,8,1);

INSERT INTO Customers(first_name, last_name, address, city, province, postal_code, telephone,
email_address, cumulative_purchase_amount)
VALUES ('John', 'Smith', 'aaa', 'Montreal', 'Quebec', 'H3P1V2', '1234567', 'a@gmail.com', 70.5),
    ('Kasra', 'Rad', 'bbb', 'Montreal', 'Quebec', 'H3H1V2', '1234562', 'b@gmail.com', 80.5),
    ('Robert', 'Wilson', 'ccc', 'Montreal', 'Quebec', 'H2H1V2', '1234563', 'c@gmail.com', 20.5),
    ('Michael', 'Anderson', 'ddd', 'Montreal', 'Quebec', 'H3V1V2', '1234564', 'd@gmail.com', 10.2),
    ('William', 'Jackson', 'eee', 'Montreal', 'Quebec', 'H4V1V2', '1234565', 'e@gmail.com', 7),
    ('David', 'Perez', 'fff', 'Montreal', 'Quebec', 'H5V1V2', '1234566', 'f@gmail.com', 170.4),
    ('Richard', 'Lee', 'ggg', 'Montreal', 'Quebec', 'H2P1V2', '1234569', 'g@gmail.com', 90.2),
    ('Joseph', 'Lewis', 'hhh', 'Montreal', 'Quebec', 'H3P1V2', '1234539', 'h@gmail.com', 40),
    ('Thomas', 'Young', 'iii', 'Montreal', 'Quebec', 'H4P1V2', '1234549', 'i@gmail.com', 300),
    ('Daniel', 'Baker', 'jjj', 'Montreal', 'Quebec', 'H3H2P2', '1234559', 'j@gmail.com', 400.58);

INSERT INTO Order_Book(order_id, ISBN, quantity, is_sold, total_cost)
VALUES (1, '0-2838-9190-4', 4, 0, sell('0-2838-9190-4', 4, 0)),
    (1, '0-6521-1546-2', 3, 0, sell('0-6521-1546-2', 3, 0)),
    (1, '0-7225-8364-8', 2, 0, sell('0-7225-8364-8', 2, 0)),
    (2, '0-2838-9190-4', 2, 1, sell('0-2838-9190-4',2, 1)),
    (3, '0-8229-1390-9', 1, 1, sell('0-8229-1390-9',1, 1)),
    (4, '0-2838-9190-4', 1, 1, sell('0-2838-9190-4',1, 1)),
    (4, '0-2237-3192-7', 10, 1, sell('0-2237-3192-7',10, 1)),
    (5, '0-2710-5457-3', 1, 1, sell('0-2710-5457-3',1, 1)),
    (6, '0-2838-9190-4', 1, 1, sell('0-2838-9190-4',1, 1)),
    (6, '0-2710-5457-3', 2, 1, sell('0-2710-5457-3',2, 1)),
    (7, '0-7225-8364-8', 5, 0, sell('0-7225-8364-8',5, 0)),
    (8, '0-2710-5457-3', 1, 0, sell('0-2710-5457-3',1, 0)),
    (8, '0-2237-3192-7', 4, 0, sell('0-2237-3192-7',4, 0)),
    (9, '0-2710-5457-3', 2, 0, sell('0-2710-5457-3',2, 0)),
    (10, '0-2710-5457-3', 20, 1, sell('0-2710-5457-3',20, 1));

INSERT INTO Orders(bookstore_id, publisher_name, brach_name, date)
VALUES (1, 2, 1, "2018-03-02");
INSERT INTO Orders(customer_id, date)
VALUES (CustomerOrder(2,1), "2018-09-05"),
     (CustomerOrder(3,2), "2018-10-07"),
    (CustomerOrder(4,1), "2019-01-05"),
    (CustomerOrder(5,3), "2019-02-05"),
    (CustomerOrder(6,4), "2019-03-05");
INSERT INTO Orders(bookstore_id, publisher_name, brach_name, date)
VALUES (1, 2, 1, "2020-01-05"),
        (1, 3, 2, "2020-02-05"),
        (1, 4, 3, "2020-03-05");
INSERT INTO Orders(customer_id, date)
VALUES (CustomerOrder(10,7), "2020-04-01");
```

Queries:

```
USE eyc353_1 ;

/* Query 1 */
SELECT
Books.ISBN,Books.publisher_number,Books.title,Books.cost_price,Books.selling_price,Order_Book.is_sold,
SUM(Order_Book.quantity) as ytd_sold
FROM Books LEFT JOIN Order_Book
        ON Books.ISBN = Order_Book.ISBN
GROUP BY Books.ISBN,Order_Book.is_sold
        HAVING Order_Book.is_sold = 1;


/* Query 2 */
SELECT
Books.ISBN,Books.publisher_number,Books.title,Books.cost_price,Books.selling_price,Branch_Book_Quantity.branch_id,
Branch_Book_Quantity.quantity,Order_Book.quantity as order_quantity,
(Order_Book.quantity - Branch_Book_Quantity.quantity) as back_order
FROM Books LEFT JOIN Branch_Book_Quantity
        ON Books.ISBN = Branch_Book_Quantity.ISBN
  LEFT JOIN Order_Book
                ON Books.ISBN = Order_Book.ISBN
WHERE Branch_Book_Quantity.branch_id > 0 AND
        (Order_Book.quantity - Branch_Book_Quantity.quantity) > 0
        GROUP BY Books.ISBN;


/* Query 3 */
SELECT
Customers.customer_id,Customers.first_name,Customers.last_name,Orders.date,Order_Book.order_id,
        Order_Book.ISBN,Order_Book.quantity
FROM Customers LEFT JOIN Special_Order
        ON Customers.customer_id = Special_Order.customer_id
  LEFT JOIN Orders
                ON Special_Order.order_id = Orders.order_id
    LEFT JOIN Order_Book
                        ON Orders.order_id = Order_Book.order_id
WHERE Customers.customer_id = 3;
```

```sql
/* Query 4 */
SELECT
Customers.customer_id,Customers.first_name,Customers.last_name,Orders.date,Order_Book.order_id,
        Order_Book.ISBN,Order_Book.quantity,Branch_Book_Quantity.branch_id
FROM Customers LEFT JOIN Orders
                ON Customers.customer_id = Orders.customer_id
    LEFT JOIN Order_Book
        ON Orders.order_id = Order_Book.order_id
    LEFT JOIN Branch_Book_Quantity
        ON Order_Book.ISBN = Branch_Book_Quantity.ISBN
WHERE Customers.customer_id = 3
        AND branch_id = 5
        AND date BETWEEN "2019-01-01" AND "2019-12-31";


/* Query 5 */
SELECT Branch_Book_Quantity.branch_id,Branch_Book_Quantity.ISBN,Books.cost_price as
branch_sell_price,
Order_Book.quantity,Orders.date,(Books.cost_price * Order_Book.quantity) as total_sales
FROM Branch_Book_Quantity LEFT JOIN Books
        ON Branch_Book_Quantity.ISBN = Books.ISBN
        LEFT JOIN Order_Book
                ON Branch_Book_Quantity.ISBN = Order_Book.ISBN
    LEFT JOIN Orders
                ON Order_Book.order_id = Orders.order_id
WHERE Branch_Book_Quantity.branch_id = 1
                AND Orders.date BETWEEN "2019-01-01" AND "2019-12-31"
    AND Order_Book.is_sold = 0;/*this means it has been sold to bookstore */


/* Query 6 */
SELECT
        Results.publisher_number AS title,
        Results.company_name
FROM (
SELECT
  publisher_number,
  name AS company_name,
  COUNT(Publishers_publisher_number) AS total
FROM Publishers
JOIN Bookstore_Publisher_Order ON Publishers_publisher_number = publisher_number
WHERE confirmation = 0
GROUP BY Publishers_publisher_number
ORDER BY total
) AS Results
LIMIT 1;


/* Query 7 */

SELECT *
FROM Books
WHERE publisher_number = 1
ORDER BY selling_price;


/* Query 8 */

SELECT *
FROM Branches
WHERE publisher_number IN (
                            SELECT publisher_number
                            FROM (
                                   SELECT
                                   publisher_number,
                                   COUNT(publisher_number) AS number_of_branches
                                   FROM Branches
                                   GROUP BY publisher_number
                                   HAVING number_of_branches >= 3) AS publisher_id);


/* Query 9 */

SELECT *
FROM Books
JOIN Inventory USING (inventory_id)
WHERE day_received <= "2019-07-19" AND Books.ISBN = ANY (
                                                         SELECT ISBN
                                                         FROM Orders
                                                         JOIN Order_Book USING (order_id)
                                                         WHERE is_sold = 1);
```

```
/* Query 10 */

SELECT *
FROM Books
WHERE ISBN IN ( SELECT Books_ISBN
                FROM Book_Author
                 JOIN Authors USING (author_id)
                 WHERE Authors.name = "C");
```

Relationship Count query

1. SELECT Count(*) FROM eyc353_1.Reader_Interest;

   Output: 180

2. SELECT Count(*) FROM eyc353_1.Orders;

   Output: 10

3. SELECT Count(*) FROM eyc353_1.Book_Subject;

   Output: 23

4. SELECT Count(*) FROM eyc353_1.Book_Author;

   Output: 23

5. SELECT Count(*) FROM eyc353_1. Bookstore_Publisher_Order;

   Output: 8

6. SELECT Count(*) FROM eyc353_1. Order_Book;

   Output: 15

7. SELECT Count(*) FROM eyc353_1. Special_Order;

   Output: 3