Faculty of Engineering and Computer Science
Expectations of Originality

This form has been created to ensure that all students in the Faculty of Engineering and Computer Science comply with principles of academic integrity prior to submitting coursework to their instructors for evaluation: namely reports, assignments, lab reports and/or software. All students should become familiar with the University's Code of Conduct (Academic) located at http://web2.concordia.ca/Legal_Counsel/policies/english/AC/Code.html

Please read the back of this document carefully before completing the section below. This form must be attached to the front of all coursework submitted to instructors in the Faculty of Engineering and Computer Science.

Course Number: COMP 5531 **Instructor:** Nagi Basha

Type of Submission (Please check off reponses to both a & b)

a. __ Report __ Assignment X Lab Report X Software

b. __ Individual submission X Group Submission (All members of the team must sign below)

Having read both sides of this form, I certify that I/we have conformed to the Faculty's expectations of originality and standards of academic integrity.

Name: Adnan Ali (please print clearly) ID No: 40181614 Signature: AAli Date: 6/20/21

Name: Daejung Bae (please print clearly) ID No: 40013623 Signature: Jasonbae Date: 6/20/21

Name: Patrick Drummond (please print clearly) ID No: 40185198 Signature: Patrick Drummond Date: 6/20/21

Name: Seyedsina Mirmaghferaty (please print clearly) ID No: 40124936 Signature: Seyedsina Mirmaghferaty Date: 6/20/21

Name: _____ (please print clearly) ID No: _____ Signature: _____ Date: _____

Name: _____ (please print clearly) ID No: _____ Signature: _____ Date: _____

Do Not Write in this Space – Reserved for Instructor

EXPECTATIONS OF ORIGINALITY & STANDARDS OF ACADEMIC INTEGRITY

ALL SUBMISSIONS must meet the following requirements:

1. The decision on whether a submission is a group or individual submission is determined by the instructor. Individual submissions are done alone and should not be identical to the submission made by any other student. In the case of group submissions, all individuals in the group must be listed on and must sign this form prior to its submission to the instructor.
2. All individual and group submissions constitute original work by the individual(s) signing this form.
3. Direct quotations make up a very small proportion of the text, i.e., not exceeding 5% of the word count.
4. Material paraphrased from a source (e.g., print sources, multimedia sources, web-based sources, course notes or personal interviews) has been identified by a numerical reference citation.
5. All of the sources consulted and/or included in the report have been listed in the Reference section of the document.
6. All drawings, diagrams, photos, maps or other visual items derived from other sources have been identified by numerical reference citations in the caption.
7. No part of the document has been submitted for any other course.
8. Any exception to these requirements are indicated on an attached page for the instructor's review.

REPORTS and ASSIGNMENTS must also meet the following additional requirements:

1. A report or assignment consists entirely of ideas, observations, information and conclusions composed by the student(s), except for statements contained within quotation marks and attributed to the best of the student's/students' knowledge to their proper source in footnotes or references.
2. An assignment may not use solutions to assignments of other past or present students/instructors of this course or of any other course.
3. The document has not been revised or edited by another student who is not an author.
4. For reports, the guidelines found in Form and Style, by Patrick MacDonagh and Jack Borden (Fourth Edition: May 2000, available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf) have been used for this submission.

LAB REPORTS must also meet the following requirements:

1. The data in a lab report represents the results of the experimental work by the student(s), derived only from the experiment itself. There are no additions or modifications derived from any outside source.
2. In preparing and completing the attached lab report, the labs of other past or present students of this course or any other course have not been consulted, used, copied, paraphrased or relied upon in any manner whatsoever.

SOFTWARE must also meet the following requirements:

1. The software represents independent work of the student(s).
2. No other past or present student work (in this course or any other course) has been used in writing this software, except as explicitly documented.
3. The software consists entirely of code written by the undersigned, except for the use of functions and libraries in the public domain, all of which have been documented on an attached page.
4. No part of the software has been used in previous submissions except as identified in the documentation.
5. The documentation of the software includes a reference to any component that the student(s) did not write.
6. All of the sources consulted while writing this code are listed in the documentation.

Important: Should you require clarification on any of the above items please contact your instructor.

COMP 5531

WARM-UP PROJECT 1

TEAM WIC55311

Team email – wic55311@encs.concordia.ca

Name	ID
Adnan Ali	40181614
Daejung Bae	40013623
Patrick Drummond	40185198
Seyedsina Mirmaghferaty	40124936

Schema

Customer (customer_id, first_name, last_name, email, phone_number, total_purchases, address, city, province, postal_code)

Bookstore (bookstore_id, name)

Author (author_id, first_name, last_name)

Reader Interest (email, customer_id, genre)

Book (isbn, title, author, cost_price, sell_price, subject, quantity_on_hand, sold_per_year)

Publisher (publisher_id, name, phone_number, email, website, address, city, province, postal_code)

Branch (branch_id, publisher_id, name, phone, address, province, postal_code, branch_manager, branch_manager_email, head_office_id)

Book (isbn, title, author_name, author_id, genre, publisher_id, cost_price, sell_price, quantity_on_hand, sold_per_year)

BooksInABranch (isbn, branch_id, inventory)

Orders (order_id, order_date, isbn, quantity_ordered, publisher_id, branch_id)

CustomersOrder (details_id, isbn, quantity, price, customer_id, bookstore_id, order_date)

Assumptions

5531 - Bookstore is given an id to distinguish it from other bookstores in the area and to allow for scalability in the database, if required.

Reader-Interest is based on a user's/customers previous book history or cookies. It is tracked by the email listed on the site and the subject is the genre of the books purchased/viewed.

Customer is a normal consumer who is separate from the bookstore entity.

Orders is a portal, created by the selling companies, that is used by both customers and bookstores to place their orders. Once either entity places their orders with the

respective details, the order is transferred to the publishing company or the book warehouse according to the order type.

Order_details record everything related to order made from the portal. It distinguishes between the Customer and Bookstore order by asking for their ID.

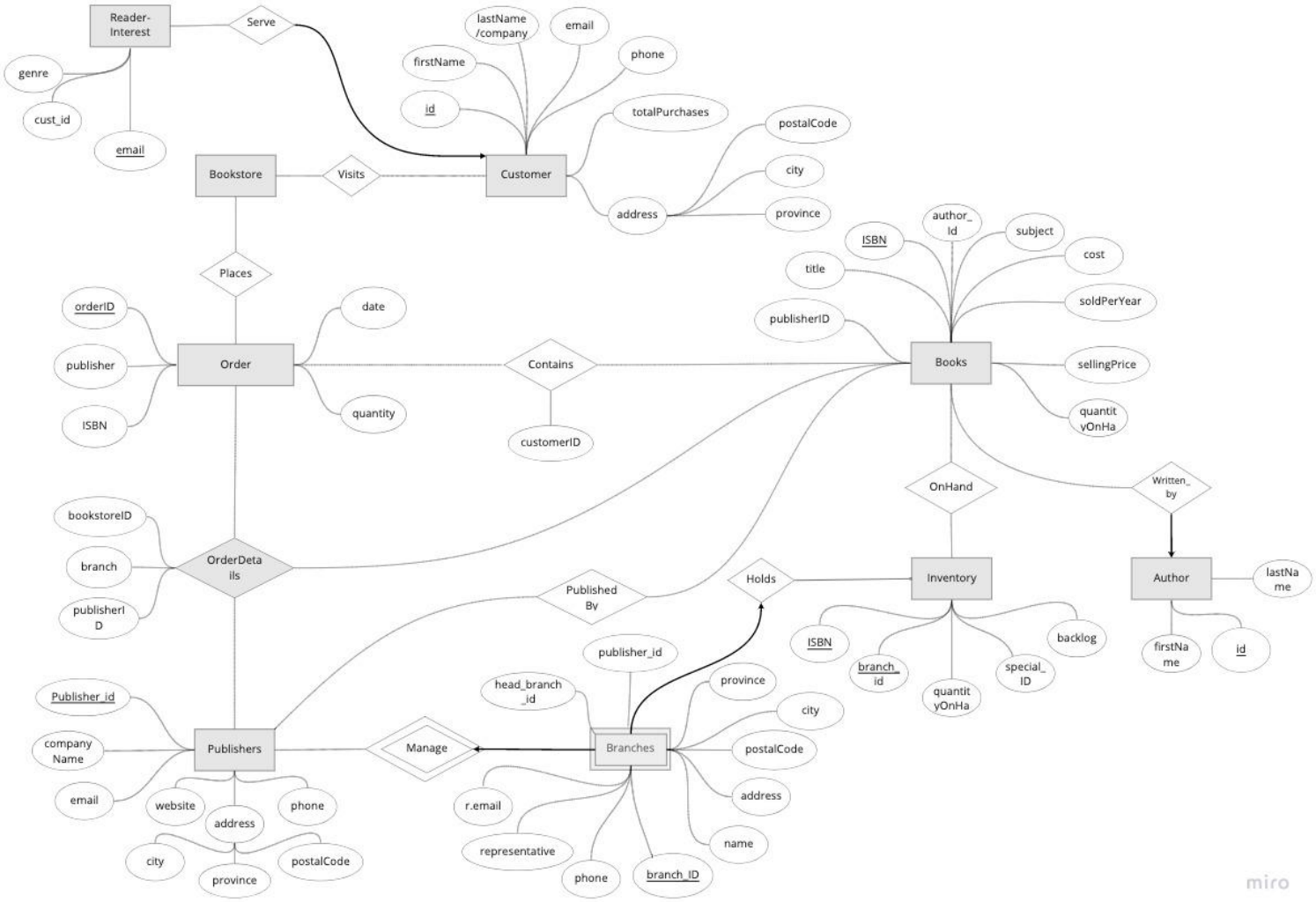
For those publishers in the area that have multiple branches, the head-office branch has a special ID given to that specific branch.

Orders portal sends the book to a specific publisher, who sends it to a specific branch.

Subject is stored as a genre for the books in the database.

Book_branch lists inventory of books that are available at a specific branch (assigned to a branch_id).

E/R Diagram


Reader-Interest
Serve
genre
cust_id
email
lastName /company
email
firstName
phone
id
totalPurchases
postalCode
city
address
province
Customer
Bookstore
Visits
Places
orderID
publisher
ISBN
Order
date
quantity
Contains
customerID
author_ Id
subject
ISBN
title
cost
soldPerYear
publisherID
Books
sellingPrice
quantit yOnHa
OnHand
Written_ by
bookstoreID
branch
publisherI D
OrderDeta ils
Published By
Holds
Inventory
Author
lastNa me
ISBN
branch_ id
quantit yOnHa
special_ ID
backlog
firstNa me
id
publisher_id
head_branch _id
province
city
postalCode
address
name
branch_ID
phone
representative
r.email
Publisher_id
company Name
email
website
address
phone
city
province
postalCode
Publishers
Manage
Branches
miro

Running the code

Our team has been assigned the wic55311 database, where we implemented our DBMS along with a generic database for queries. To login, just use the following credentials.

Your MYSQL username is wic55311
The name of the MYSQL server is wic5531.encs.concordia.ca
The name of the database you can use is also wic55311
The password for your database is ASBP5531 (case sensitive)
You cannot change this password.

The script for the schema has been stored in Procedures > abc.
The script for the data has been stored in Procedures > data.
The script for each query has been stored in Procedures > queries.

Each script can be accessed by using the call function. Details of each script are mentioned below with the following results.

Query1: Get details of all books in stock.

isbn	title	author	author_id	subject	publisher_id	cost_price	sell_price	quantity_on_hand	sold_per_year
5110737231234	Real Genius	Christie	10	mystery	10	12.00	26.00	7	98
9780140447934	War and Peace	Tolstoy	2	Novel	2	17.00	20.99	15	400
9780140707342	Hamlet	Shakespeare	3	Tragic Novel	3	18.00	20.00	20	800
9780807281918	Harry Potter and the Chamber of Secrets	Rowling	1	Novel	1	19.80	22.50	20	500
9781260548006	Software Engineering a practitioners Approach	Pressman	7	Computer Science	7	84.24	90.24	125	1250
9781400136315	The Adventures Of Huckleberry Finn	Twiain	8	American Literature	8	113.50	120.00	5	899
9781400156337	Pride and Prejudice	Austen	4	Romantic Novel	4	51.00	58.46	30	184
9781405032735	Murder on the Orient Express	Christie	10	Novel	10	85.00	90.00	2	401
9781478902089	Along Came A Spider	Christie	10	Novel	10	60.23	65.23	200	0
9781600242304	The Quickie	Patterson	9	Novel	9	15.50	18.00	160	1500

Query2: Get details of all back orders.

order_id	order_date	isbn	quantity_ordered	publisher_id	branch_id
1	2020-01-15	9780807281918	1	1	1

Query3: For a given customer, get details of all his/her back orders

details_id	isbn	quantity	price	customer_id	bookstore_id	order_date
1	9780807281918	1	19.80	1	1	2020-01-15

Query4: For a given customer, get details of all his/her purchases made during a specific period of time

first_name	last_name	details_id	isbn	quantity	price	customer_id	bookstore_id	order_date
Hlbert	Hinstein	3	9780140707342	3	18.00	8	1	2020-03-17
Hlbert	Hinstein	8	9781260548006	7	84.20	8	1	2020-08-22

Query5: Give a report of sales during a specific period of time

isbn	quantity_ordered	sell_price	cost_price	profit
9780807281918	1	22.50	19.80	2.70
9780140447934	2	20.99	17.00	7.98
9780140707342	3	20.00	18.00	6.00
9781400156337	4	58.46	51.00	29.84
9781551117584	5	12.00	10.00	10.00

Query6: Find the title of book(s) that have the highest back-order

title	back_order
Murder on the Orient Express	10

Query7: Give details of books that are supplied by a given publisher

isbn	title	author	author_id	subject	publisher_id	cost_price	sell_price	quantity_on_hand	sold_per_year
5110737231234	Real Genius	Christie	10	mystery	10	12.00	26.00	7	98
9781405032735	Murder on the Orient Express	Christie	10	Novel	10	85.00	90.00	2	401
9781478902089	Along Came A Spider	Christie	10	Novel	10	60.23	65.23	200	0

Query8: For a given publisher, get details of the head office and all the branches for that publisher

branch_id	publisher_id	name	branch_manager	branch_manager_email	head_office_id	phone_number	address	province	postal_code	publisher_id	name	phone_number	email	website
1	10	First Branch	Donald Trump	donald.trump@gmail.com	0	12345678910	First Branch Street	ON	A5D2B2	10	Bloomsburry Publishing	20191817161	blooms@gmail.com	Bloomsburry.com
2	10	Second Branch	Melania Trump	melania.trump@gmail.com	0	12345678910	Second Branch Street	ON	A5D2B2	10	Bloomsburry Publishing	20191817161	blooms@gmail.com	Bloomsburry.com
3	10	Third Branch	Trump Jr	junior.trump@gmail.com	0	12345678910	Third Branch Street	ON	A5D2B2	10	Bloomsburry Publishing	20191817161	blooms@gmail.com	Bloomsburry.com
4	10	Fourth Branch	Ivanka Trump	ivanka.trump@gmail.com	1	12345678910	Foruth Branch Street	ON	A5D2B2	10	Bloomsburry Publishing	20191817161	blooms@gmail.com	Bloomsburry.com

Query9: Get details of books that are in the inventory but there have never been a purchase for that specific book.

isbn	title	author	author_id	subject	publisher_id	cost_price	sell_price	quantity_on_hand	sold_per_year
9781478902089	Along Came A Spider	Christie	10	Novel	10	60.23	65.23	200	0

Query10: Get details of all books that are in the inventory for a given author

isbn	title	author	author_id	subject	publisher_id	cost_price	sell_price	quantity_on_hand	sold_per_year
5110737231234	Real Genius	Christie	10	mystery	10	12.00	26.00	7	98
9781405032735	Murder on the Orient Express	Christie	10	Novel	10	85.00	90.00	2	401
9781478902089	Along Came A Spider	Christie	10	Novel	10	60.23	65.23	200	0

Procedures / abc / source

```
CREATE DEFINER=`wic55311`@`132.205.%.%` PROCEDURE `wic55311`.`abc`()
BEGIN

SET FOREIGN_KEY_CHECKS=0; -- Please keep drop table in order
drop table if exists customer;
drop table if exists bookstore;
drop table if exists author;
drop table if exists reader_interest;
drop table if exists publisher;
drop table if exists branch;
drop table if exists book;
drop table if exists inventory;
drop table if exists book_branch;
drop table if exists orders;
drop table if exists order_details;
SET FOREIGN_KEY_CHECKS=1;

create table customer
(
	customer_id int not null,
	first_name varchar(20) not null,
	last_name varchar(20) not null,
	email varchar(20) not null,
	phone_number varchar(11) not null,
	address varchar(50) not null,
	city varchar(20) not null,
	postal_code varchar(6) not null,
	province varchar(2) not null,
	total_purchases smallint,
	company_name varchar (50), --  check
	primary key (customer_id)
);

create table bookstore
(
	bookstore_id int not null,
	customer_id int,
	primary key (bookstore_id),
	foreign key (customer_id) references customer(customer_id)
);

create table author
(
	author_id int not null,
	first_name varchar(50),
```

```
        last_name varchar(50),
        primary key (author_id)
);

create table reader_interest
(
        customer_id int not null,
        email varchar(20) not null,
        genre varchar(20) not null,
        primary key (email),
        foreign key (customer_id) references customer(customer_id)
);

create table publisher
(
        publisher_id int not null,
        name varchar(50) not null,
        phone_number varchar(11) not null,
        email varchar (50) not null,
        website varchar(50) not null,
        address varchar (50) not null,
        city varchar (20) not null,
        province varchar (2) not null,
        postal_code varchar(6) not null,
        primary key (publisher_id)
);

create table branch
(
        branch_id int not null,
        publisher_id int not null,
        name varchar(50) not null,
        branch_manager varchar(50) not null,
        branch_manager_email varchar(50) not null,
        head_office_id smallint not null,
        phone_number varchar(11) not null,
        address varchar(50) not null,
        province varchar(2) not null,
        postal_code varchar(6) not null,
        primary key (branch_id),
        foreign key (publisher_id) references publisher(publisher_id)
);

create table book
(
        isbn varchar(13) not null,
        title varchar (100) not null,
        author varchar (50) not null,
```

```
        author_id int not null,
        subject varchar (50) not null,
        publisher_id int not null,
        cost_price decimal (6,2) not null,
        sell_price decimal (6,2) not null,
        quantity_on_hand smallint,
        sold_per_year smallint,
        primary key (isbn),
        foreign key (author_id) references author(author_id),
        foreign key (publisher_id) references publisher(publisher_id)
);

create table inventory
(
        bookstore_id int not null,
        isbn varchar(13) not null,
        quantity_on_hand smallint,
        primary key (bookstore_id, isbn),
        foreign key (bookstore_id) references bookstore(bookstore_id),
        foreign key (isbn) references book(isbn)
);


create table book_branch -- books available in a branch
(
        isbn varchar(13) not null,
        branch_id int not null,
        inventory int not null,
        primary key (branch_id, isbn),
        foreign key (isbn) references book (isbn),
        foreign key (branch_id) references branch (branch_id)
);

create table orders -- books ordered by publishers
(
        order_id int not null,
        order_date date, -- YYYY-MM-DD
        isbn varchar (13) not null,
        quantity_ordered int not null,
        publisher_id int not null,
        branch_id int not null,
        primary key (order_id),
        foreign key(isbn) references book(isbn),
        foreign key(publisher_id) references publisher(publisher_id),
        foreign key(branch_id) references branch(branch_id)
);

create table order_details  -- books ordered by customers (includes bookstores)
```

```
(
        details_id int not null,
        isbn varchar (13) not null,
        quantity smallint not null,
        price decimal (6,2) not null,
        customer_id int not null,
        bookstore_id int not null,
        order_date date, -- YYYY-MM-DD
        primary key (details_id),
        foreign key (isbn) references book(isbn),
        foreign key (customer_id) references customer(customer_id),
        foreign key (bookstore_id) references bookstore(bookstore_id)
);


END
```

Procedures / data / source

```
CREATE DEFINER=`wic55311`@`132.205.%.%` PROCEDURE `wic55311`.`data`()
BEGIN

insert into bookstore (bookstore_id, customer_id)
values (1, null);

insert into customer (customer_id, first_name, last_name, email, phone_number, address, city,
postal_code, province, total_purchases, company_name)
values
        (1, 'Albert', 'Ainstein', 'albert@gmail.com', 12345678910, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 40, null),
        (2, 'Blbert', 'Binstein', 'blbert@gmail.com', 12345678911, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 39, null),
        (3, 'Clbert', 'Cinstein', 'clbert@gmail.com', 12345678912, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 38, null),
        (4, 'Dlbert', 'Dinstein', 'dlbert@gmail.com', 12345678913, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 37, null),
        (5, 'Elbert', 'Einstein', 'elbert@gmail.com', 12345678914, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 36, null),
        (6, 'Flbert', 'Finstein', 'flbert@gmail.com', 12345678915, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 35, null),
        (7, 'Glbert', 'Ginstein', 'glbert@gmail.com', 12345678916, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 34, null),
        (8, 'Hlbert', 'Hinstein', 'hlbert@gmail.com', 12345678917, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 33, null),
        (9, 'Ilbert', 'Iinstein', 'ilbert@gmail.com', 12345678918, 'Albert Street', 'Toronto',
'M4C1A1', 'ON', 32, null),
        (10, 'John', 'Smith', 'john.smith@gmail.com', 19987654321, 'Hohn Street', 'Toronto',
'A5D2B2', 'BC', 25, null);


insert into author (author_id, first_name, last_name)
values
        (1, 'Jk',          'Rowling'),
        (2, 'Leo' ,        'Tolstoy'),
        (3, 'Wiliam',  'Shakespeare'),
        (4, 'Jane',        'Austen'),
        (5, 'Charles', 'Dickens'),
        (6, 'Abraham', 'Silberschatz'),
        (7, 'RogerS',  'Pressman'),
        (8, 'Mark' ,   'Twiain'),
        (9, 'James' ,  'Patterson'),
        (10, 'Agatha', 'Christie'),
        (11, 'James' , 'Guyer'),
        (12, 'William', 'Ditty'),
```

```
        (13, 'Lucia', 'Yushachkov');


insert into reader_interest (customer_id, email, genre)
values
        (1, 'albert@gmail.com', 'Horror'),
        (2, 'blbert@gmail.com', 'Horror'),
        (3, 'clbert@gmail.com', 'Horror'),
        (4, 'dlbert@gmail.com', 'Horror'),
        (5, 'elbert@gmail.com', 'Horror'),
        (6, 'flbert@gmail.com', 'Horror'),
        (7, 'glbert@gmail.com', 'Horror'),
        (8, 'hlbert@gmail.com', 'Horror'),
        (9, 'ilbert@gmail.com', 'Horror'),
        (10, 'john.smith@gmail.com', 'Suspense');


insert into publisher (publisher_id, name, phone_number, email, website, address, city,
province, postal_code)
values
        (1, 'publisher1', 12345678911, 'publisher1@gmail.com', 'publisher1.com', 'Publisher1
Street', 'Toronto', 'ON', 'M4C1A1'),
        (2, 'publisher2', 12345678912, 'publisher1@gmail.com', 'publisher2.com', 'Publisher2
Street', 'Toronto', 'ON', 'M4C1A1'),
        (3, 'publisher3', 12345678913, 'publisher1@gmail.com', 'publisher3.com', 'Publisher3
Street', 'Toronto', 'ON', 'M4C1A1'),
        (4, 'publisher4', 12345678914, 'publisher1@gmail.com', 'publisher4.com', 'Publisher4
Street', 'Toronto', 'ON', 'M4C1A1'),
        (5, 'publisher5', 12345678915, 'publisher1@gmail.com', 'publisher5.com', 'Publisher5
Street', 'Toronto', 'ON', 'M4C1A1'),
        (6, 'publisher6', 12345678916, 'publisher1@gmail.com', 'publisher6.com', 'Publisher6
Street', 'Toronto', 'ON', 'M4C1A1'),
        (7, 'publisher7', 12345678917, 'publisher1@gmail.com', 'publisher7.com', 'Publisher7
Street', 'Toronto', 'ON', 'M4C1A1'),
        (8, 'publisher8', 12345678918, 'publisher1@gmail.com', 'publisher8.com', 'Publisher8
Street', 'Toronto', 'ON', 'M4C1A1'),
        (9, 'publisher9', 12345678919, 'publisher1@gmail.com', 'publisher9.com', 'Publisher9
Street', 'Toronto', 'ON', 'M4C1A1'),
        (10, 'Bloomsburry Publishing', 20191817161, 'blooms@gmail.com', 'Bloomsburry.com',
'Publisher1 Street', 'Toronto', 'ON', 'NAC24R');


insert into branch (branch_id, publisher_id, name, branch_manager, branch_manager_email,
head_office_id, phone_number, address, province, postal_code)
values
        (1, 10, 'First Branch', 'Donald Trump', 'donald.trump@gmail.com', 0, 12345678910, 'First
Branch Street', 'ON', 'A5D2B2'),
```

```
        (2, 10, 'Second Branch', 'Melania Trump', 'melania.trump@gmail.com', 0, 12345678910,
'Second Branch Street', 'ON', 'A5D2B2'),
        (3, 10, 'Third Branch', 'Trump Jr', 'junior.trump@gmail.com', 0, 12345678910, 'Third
Branch Street', 'ON', 'A5D2B2'),
        (4, 10, 'Fourth Branch', 'Ivanka Trump', 'ivanka.trump@gmail.com', 1, 12345678910,
'Foruth Branch Street', 'ON', 'A5D2B2'),
        (5, 1, 'Branch1', 'Manager1', 'manager1@gmail.com', 0, 12345678910, 'Branch1 Street',
'ON', 'M4C1A1'),
        (6, 4, 'Branch2', 'Manager2', 'manager2@gmail.com', 0, 12345678910, 'Branch2 Street',
'ON', 'M4C1A1'),
        (7, 6, 'Branch3', 'Manager3', 'manager3@gmail.com', 0, 12345678910, 'Branch3 Street',
'ON', 'M4C1A1'),
        (8, 5, 'Branch4', 'Manager4', 'manager4@gmail.com', 0, 12345678910, 'Branch4 Street',
'ON', 'M4C1A1'),
        (9, 2, 'Branch5', 'Manager5', 'manager5@gmail.com', 0, 12345678910, 'Branch5 Street',
'ON', 'M4C1A1'),
        (10, 3, 'Branch6', 'Manager6', 'manager6@gmail.com', 0, 12345678910, 'Branch6
Street', 'ON', 'M4C1A1');


insert into book (isbn, title, author, author_id, subject, publisher_id, cost_price, sell_price,
quantity_on_hand, sold_per_year)
values
        ('9780807281918', 'Harry Potter and the Chamber of Secrets','Rowling', 1, 'Novel', 1,
19.8, 22.5, 20 , 500),
        ('9780140447934', 'War and Peace','Tolstoy', 2, 'Novel', 2, 17, 20.99,  15  , 400),
        ('9780140707342', 'Hamlet','Shakespeare', 3, 'Tragic Novel', 3, 18, 20,  20 , 800),
        ('9781400156337', 'Pride and Prejudice','Austen', 4, 'Romantic Novel', 4, 51, 58.46,  30 ,
184),
        ('9781551117584', 'Oliver Twist', 'Dickens', 5, 'Novel', 5, 10, 12, 0  , 870),
        ('9781119320913', 'Operating Systems Concepts','Silberschatz', 6, 'Computer Education',
6, 100, 120,  0 , 100),
        ('9781260548006', 'Software Engineering a practitioners Approach', 'Pressman', 7,
'Computer Science', 7, 84.24, 90.24, 125 , 1250),
        ('9781400136315', 'The Adventures Of Huckleberry Finn', 'Twiain', 8, 'American
Literature', 8, 113.5, 120, 5  , 899),
        ('9781600242304', 'The Quickie','Patterson', 9, 'Novel', 9, 15.5, 18,  160  , 1500),
        ('9781405032735', 'Murder on the Orient Express','Christie', 10, 'Novel', 10, 85, 90,  2  ,
401),
        ('9781478902089', 'Along Came A Spider' , 'Patterson', 10, 'Novel', 10, 60.23, 65.23, 200
, 0),
        ('5110737231234', 'Real Genius', 'Guyer', 10, 'mystery', 10,  12.00, 26.00 , 7, 98);


insert into inventory (bookstore_id, isbn, quantity_on_hand)
values
        (1, '9780807281918', 40),
        (1, '9780140447934', 21),
```

```
    (1, '9780140707342', 0),
    (1, '9781400156337', 27),
    (1, '9781551117584', 20),
    (1, '9781119320913', 32),
    (1, '9781260548006', 15),
    (1, '9781400136315', 21),
    (1, '9781600242304', 22),
    (1, '9781405032735', 19),
    (1, '9781478902089', 20),
    (1, '5110737231234', 30);


insert into book_branch (isbn, branch_id, inventory)
values
    ('9780807281918', 1, 0),
    ('9780140447934', 2, 15),
    ('9780140707342', 3, 20),
    ('9781400156337', 4, 25),
    ('9781551117584', 5, 30),
    ('9781119320913', 6, 35),
    ('9781260548006', 7, 40),
    ('9781400136315', 8, 45),
    ('9781600242304', 9, 50),
    ('9781405032735', 10, 55),
    ('9781478902089', 10, 60),
    ('5110737231234', 10, 65);


insert into orders (order_id, order_date, isbn, quantity_ordered, publisher_id, branch_id)
values
    (1, "2020-01-15", '9780807281918', 1, 1, 1),
    (2, "2020-02-16", '9780140447934', 2, 2, 2),
    (3, "2020-03-17", '9780140707342', 3, 3, 3),
    (4, "2020-04-18", '9781400156337', 4, 4, 4),
    (5, "2020-05-19", '9781551117584', 5, 5, 5),
    (6, "2020-06-20", '9781119320913', 6, 6, 6),
    (7, "2020-07-21", '9781260548006', 7, 7, 7),
    (8, "2020-08-22", '9781400136315', 8, 8, 8),
    (9, "2020-09-23", '9781405032735', 9, 9, 9),
    (10, "2020-10-24", '9781405032735', 10, 10, 10);


insert into order_details (details_id, isbn, quantity, price, customer_id, bookstore_id,
order_date)
values
    (1, '9780807281918', 1, 19.8, 1, 1, "2020-01-15"),
    (2, '9780140447934', 2, 17, 2, 1, "2020-02-16"),
    (3, '9780140707342', 3, 18, 8, 1, "2020-03-17"),
```

```
(4, '9781400156337', 4, 51, 4, 1,"2020-04-18"),
(5, '9781551117584', 5, 10, 5, 1, "2020-05-19"),
(6, '9781119320913', 6, 100, 6, 1, "2020-06-20"),
(7, '9781260548006', 7, 84.2, 7, 1, "2020-07-21"),
(8, '9781260548006', 7, 84.2, 8, 1, "2020-08-22"),
(9, '9781405032735', 9, 15.5, 9, 1, "2020-09-23"),
(10, '9781405032735', 10, 60.23, 10, 1, "2020-10-24");
```

END

Procedures / queries / source

```
CREATE DEFINER=`wic55311`@`132.205.%.%` PROCEDURE `wic55311`.`queries`()
BEGIN

-- i.	Get details of all books in stock.
select *
from book
where quantity_on_hand > 0;

-- ii.	Get details of all back orders.
select orders.*
from orders
where
	branch_id in
				(select branch_id
				from book_branch
				where inventory = 0);

-- iii.	For a given customer, get details of all his/her back orders.
select order_details.*
from order_details
where
	order_details.isbn = '9780807281918'; -- took book from answer (ii) as it was in
backorder

-- iv.	For a given customer, get details of all his/her purchases made during a
specific period of time.
select customer.first_name, customer.last_name, order_details.*
from order_details, customer
where order_details.customer_id = customer.customer_id
	and order_details.order_date between "2020-01-15" and "2020-09-23"
	and order_details.customer_id = 8;

-- v.	Give a report of sales during a specific period of time.
select book.isbn, orders.quantity_ordered, book.sell_price, book.cost_price,
(orders.quantity_ordered * (book.sell_price - book.cost_price)) as profit
from book, orders
where book.isbn = orders.isbn
	and orders.order_date between "2020-01-15" and "2020-05-19";

-- vi.	Find the title of book(s) that have the highest back-order.
select book.title, (orders.quantity_ordered - inventory.quantity_on_hand) as back_order
from book, orders, order_details, inventory
where book.isbn = orders.isbn
	and orders.isbn = order_details.isbn
```

```
        and (orders.quantity_ordered - inventory.quantity_on_hand) > 0
        group by book.title
        order by back_order desc
        limit 1;

-- vii.  Give details of books that are supplied by a given publisher.
select book.*
from book, publisher
where book.publisher_id = publisher.publisher_id
        and publisher.publisher_id = 10;

-- viii. For a given publisher, get details of the head office and all the branches for
that publisher.
select *
from branch, publisher
where publisher.publisher_id = branch.publisher_id and branch.publisher_id = 10;

-- ix.   Get details of books that are in the inventory but there have never been a
purchase for that specific book.
select *
from book
where sold_per_year = 0;

-- x.    Get details of all books that are in the inventory for a given author.
select book.*
from inventory, author, book
where inventory.isbn = book.isbn
        and author.author_id = book.author_id
        and author.author_id = 10;

END
```