CS-GY 6083 B - Principles of Database Systems Report for Project Part II

- Section: B
- Date: December 15, 2022
- Participants:
    - Junda Ai, net ID ja4426
    - Jayant Raj, net ID jr5993
    - Yuyang Liu, net ID yl8019

Table of Contents

Executive Summary

The goal of this project is to build a web-based user interface for the database schema designed in Part I, which models a library management system that provide three main services to the customer:

- An inventory for book rental
- A events management system for exhibitions and seminars
- A rooms management system for group study room reservation

In addition to the services mentioned above the library would also allow customers to signup and login to their account and manage access authorizations of different groups of users.

Tech Stack

- Server side: Django, Python, HTML
- Database: MySQL
- CSS: Bootstrap
- Deployment: Docker, Docker Compose

Given the tight deadline and short of stuff Junda decided to use the ***Django*** framework in the Python programming language to facilitate rapid prototyping of the application with tons of security features out of the box, and the ***Bootstrap*** framework to style the HTML pages. And ***Docker compose*** to orchestrate the deployment of the Django application and the MySQL instance.

Django is a full stack web framework that provides an ORM (Object Relational Mapper) to interact with the database, a templating engine to render the HTML pages, and a web server to serve the pages. The ORM is a layer that sits between the application and the database and provides a high level interface to interact with the database. The templating engine is a layer that sits between the application and the web server and provides a high level interface to render the HTML pages. The web server is a layer that sits between the application and the client and provides a high level interface to serve the pages. Django also comes with an admin page to conveniently manage database models out of the box and some useful security middlewares, please see Security Features for more details. Bootstrap provides a set of CSS classes to plug-and-play with the HTML pages.

DDL File Content

```
-- SQLINES DEMO *** le SQL Developer Data Modeler 22.2.0.165.1149
-- SQLINES DEMO *** -11-05 15:10:28 EDT
-- SQLINES DEMO *** le Database 21c
-- SQLINES DEMO *** le Database 21c



-- SQLINES DEMO *** no DDL - MDSYS.SDO_GEOMETRY

-- SQLINES DEMO *** no DDL - XMLTYPE

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE author (
    auth_id INT NOT NULL COMMENT 'AUTHOR ID',
    fname   VARCHAR(30) NOT NULL COMMENT 'First name',
    lname   VARCHAR(30) NOT NULL COMMENT 'Last name',
    email   VARCHAR(50) NOT NULL COMMENT 'Author''s email',
    street  VARCHAR(30) NOT NULL COMMENT 'Street address',
    city    VARCHAR(30) NOT NULL COMMENT 'Author''s city',
    state   VARCHAR(2) NOT NULL COMMENT 'Author''s state code',
    zip     INT NOT NULL COMMENT 'Author''s zip code'
);
```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN author.auth_id IS
    'AUTHOR ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.fname IS
    'First name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.lname IS
    'Last name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.email IS
    'Author''s email'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.street IS
    'Street address'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.city IS
    'Author''s city'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.state IS
    'Author''s state code'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.zip IS
    'Author''s zip code'; */

ALTER TABLE author ADD CONSTRAINT author_pk PRIMARY KEY ( auth_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE author_seminar (
    inv_id  INT NOT NULL COMMENT 'Invitation ID',
    auth_id INT NOT NULL,
    e_id    INT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN author_seminar.inv_id IS
    'Invitation ID'; */

ALTER TABLE author_seminar
    ADD CONSTRAINT author_seminar_pk PRIMARY KEY ( inv_id,
                                                   auth_id,
                                                   e_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE book (
    book_id  INT NOT NULL COMMENT 'Book ID',
    name     VARCHAR(50) NOT NULL COMMENT 'Book''s name',
```

```
    topic_id SMALLINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN book.book_id IS
    'Book ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN book.name IS
    'Book''s name'; */

ALTER TABLE book ADD CONSTRAINT book_pk PRIMARY KEY ( book_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE book_author (
    auth_id INT NOT NULL,
    book_id INT NOT NULL
);

ALTER TABLE book_author ADD CONSTRAINT book_author_pk PRIMARY KEY (
book_id,

auth_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE copy (
    copy_id BIGINT NOT NULL COMMENT 'Unique Copy ID for a book',
    status  CHAR(1) NOT NULL COMMENT 'Status of copy - A for available, R
for on rent',
    book_id INT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN copy.copy_id IS
    'Unique Copy ID for a book'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN copy.status IS
    'Status of copy - A for available, R for on rent'; */

ALTER TABLE copy ADD CONSTRAINT copy_pk PRIMARY KEY ( copy_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE customer (
    cust_id   BIGINT NOT NULL COMMENT 'Customer ID',
    fname     VARCHAR(30) NOT NULL COMMENT 'First name',
    lname     VARCHAR(30) NOT NULL,
    phone     VARCHAR(13) NOT NULL COMMENT 'Customer''s phone number',
    email     VARCHAR(50) NOT NULL COMMENT 'Customer''s email',
    id_type   CHAR(1) NOT NULL COMMENT 'Identification type - one of
passport (P), SSN (S) or driver''s license (D)',
    id_number VARCHAR(15) NOT NULL COMMENT 'ID_NUMBER (can be
alphanumeric) from passport, SSN-driver''s license.'
);
```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.cust_id IS
    'Customer ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.fname IS
    'First name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.phone IS
    'Customer''s phone number'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.email IS
    'Customer''s email'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.id_type IS
    'Identification type - one of passport (P), SSN (S) or driver''s
license (D)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.id_number IS
    'ID_NUMBER (can be alphanumeric) from passport, SSN-driver''s
license.'; */

ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( cust_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE customer_exhibition (
    reg_id  INT NOT NULL COMMENT 'Registration ID or invitation ID
depending on whether the customer is an author participating in seminar or
in exhibition.',
    cust_id BIGINT NOT NULL,
    e_id    DOUBLE NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer_exhibition.reg_id IS
    'Registration ID or invitation ID depending on whether the customer is
an author participating in seminar or in exhibition.'; */

ALTER TABLE customer_exhibition
    ADD CONSTRAINT customer_exhibition_pk PRIMARY KEY ( reg_id,
                                                        cust_id,
                                                        e_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE customer_room (
    day_booked DATETIME NOT NULL COMMENT 'Day the room is booked',
    timeslot   TINYINT NOT NULL COMMENT 'Timeslot the room is booked - one
of four, with ''1'' for 8-10AM, ''2'' for 11am-1pm, ''3'' for 1-3pm and
''4'' for 4-6pm.',
```

```
    cust_id    BIGINT NOT NULL,
    room_id    SMALLINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer_room.day_booked IS
    'Day the room is booked'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer_room.timeslot IS
    'Timeslot the room is booked - one of four, with ''1'' for 8-10AM,
''2'' for 11am-1pm, ''3'' for 1-3pm and ''4'' for 4-6pm.'; */

ALTER TABLE customer_room ADD CONSTRAINT customer_room_pk PRIMARY KEY (
room_id,

cust_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE event (
    e_id       INT NOT NULL COMMENT 'Unique Event ID',
    name       VARCHAR(30) NOT NULL COMMENT 'Event''s name',
    type       VARCHAR(10) NOT NULL COMMENT 'Event type - can either be
Seminar or Exhibition',
    start_time DATETIME NOT NULL COMMENT 'Start datetime',
    stop_time  DATETIME NOT NULL COMMENT 'Stop datetime',
    topic_id   SMALLINT
);

ALTER TABLE event
    ADD CONSTRAINT ch_inh_event CHECK ( type IN ( 'Exhibition', 'Seminar'
) );

/* Moved to CREATE TABLE
COMMENT ON COLUMN event.e_id IS
    'Unique Event ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN event.name IS
    'Event''s name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN event.type IS
    'Event type - can either be Seminar or Exhibition'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN event.start_time IS
    'Start datetime'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN event.stop_time IS
    'Stop datetime'; */

ALTER TABLE event ADD CONSTRAINT event_pk PRIMARY KEY ( e_id );
```

```
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE exhibition (
    e_id     INT NOT NULL COMMENT 'Unique Event ID',
    expenses DECIMAL(8, 2) NOT NULL COMMENT 'Exhibition expenses'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN exhibition.e_id IS
    'Unique Event ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN exhibition.expenses IS
    'Exhibition expenses'; */

ALTER TABLE exhibition ADD CONSTRAINT exhibition_pk PRIMARY KEY ( e_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE invoice (
    invoice_id BIGINT NOT NULL COMMENT 'Invoice ID',
    inv_date   DATETIME NOT NULL COMMENT 'Date the invoice is generated',
    inv_amt    DECIMAL(8, 2) NOT NULL COMMENT 'Invoice amount (calculated
based on formula).',
    rent_id    INT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN invoice.invoice_id IS
    'Invoice ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN invoice.inv_date IS
    'Date the invoice is generated'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN invoice.inv_amt IS
    'Invoice amount (calculated based on formula).'; */

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX invoice__idx ON
    invoice (
        rent_id
    ASC );

ALTER TABLE invoice ADD CONSTRAINT invoice_pk PRIMARY KEY ( invoice_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE payment (
    pay_id     BIGINT NOT NULL COMMENT 'Unique Payment ID for paying a
part (or full) of that invoice',
    pay_date   DATETIME NOT NULL COMMENT 'Date payment is done',
    pay_method CHAR(2) NOT NULL COMMENT 'Payment method - ''P'' for
PayPal, ''CC'' for credit card, ''CA'' for cash and ''DC'' for debit
card.',
```

```
    fname      VARCHAR(30) NOT NULL COMMENT 'Cardholder''s first name',
    lname      VARCHAR(30) NOT NULL COMMENT 'Cardholder''s last name',
    pay_amt    DECIMAL(8, 2) NOT NULL COMMENT 'Payment amount',
    invoice_id BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.pay_id IS
    'Unique Payment ID for paying a part (or full) of that invoice'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.pay_date IS
    'Date payment is done'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.pay_method IS
    'Payment method - ''P'' for PayPal, ''CC'' for credit card, ''CA'' for
cash and ''DC'' for debit card.'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.fname IS
    'Cardholder''s first name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.lname IS
    'Cardholder''s last name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.pay_amt IS
    'Payment amount'; */

ALTER TABLE payment ADD CONSTRAINT payment_pk PRIMARY KEY ( pay_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE rental (
    rent_id       INT NOT NULL COMMENT 'Unique rental service ID for each
time a copy is borrowed',
    r_status      CHAR(1) NOT NULL COMMENT 'Rental status - ''B'' for
borrowed, ''R'' for returned, ''L'' for late',
    borrow_date   DATETIME NOT NULL COMMENT 'Date borrowed',
    exp_return    DATETIME NOT NULL COMMENT 'Date the rental is expected
to return',
    actual_return DATETIME NOT NULL COMMENT 'Date the rental is actually
returned',
    copy_id       BIGINT
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN rental.rent_id IS
    'Unique rental service ID for each time a copy is borrowed'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN rental.r_status IS
    'Rental status - ''B'' for borrowed, ''R'' for returned, ''L'' for
```

```
late'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN rental.borrow_date IS
    'Date borrowed'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN rental.exp_return IS
    'Date the rental is expected to return'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN rental.actual_return IS
    'Date the rental is actually returned'; */

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX rental__idx ON
    rental (
        rent_id
    ASC );

ALTER TABLE rental ADD CONSTRAINT rental_pk PRIMARY KEY ( rent_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE room (
    room_id       SMALLINT NOT NULL COMMENT 'Unique ID for each room',
    room_capacity TINYINT NOT NULL COMMENT 'Max capacity for each room'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN room.room_id IS
    'Unique ID for each room'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN room.room_capacity IS
    'Max capacity for each room'; */

ALTER TABLE room ADD CONSTRAINT room_pk PRIMARY KEY ( room_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE seminar (
    e_id INT NOT NULL COMMENT 'Unique Event ID'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN seminar.e_id IS
    'Unique Event ID'; */

ALTER TABLE seminar ADD CONSTRAINT seminar_pk PRIMARY KEY ( e_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE sponsor (
    sponsor_id INT NOT NULL COMMENT 'Unique Sponsor ID',
    fname      VARCHAR(30) NOT NULL COMMENT 'Sponsor First name if
individual. Org name if org.',
```

```
    lname      VARCHAR(30) COMMENT 'Sponsor''s last name (optional)',
    type       CHAR(1) NOT NULL COMMENT 'Sponsor type. I for individual,
and O for organisation.'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor.sponsor_id IS
    'Unique Sponsor ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor.fname IS
    'Sponsor First name if individual. Org name if org.'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor.lname IS
    'Sponsor''s last name (optional)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor.type IS
    'Sponsor type. I for individual, and O for organisation.'; */

ALTER TABLE sponsor ADD CONSTRAINT sponsor_pk PRIMARY KEY ( sponsor_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE sponsor_seminar (
    amt_supp   DECIMAL(8, 2) NOT NULL COMMENT 'Amount each sponsor
supports for each seminar',
    sponsor_id INT NOT NULL,
    e_id       INT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor_seminar.amt_supp IS
    'Amount each sponsor supports for each seminar'; */

ALTER TABLE sponsor_seminar ADD CONSTRAINT sponsor_seminar_pk PRIMARY KEY
( sponsor_id,

e_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE topic (
    topic_id    SMALLINT NOT NULL COMMENT 'Topic ID',
    name        VARCHAR(30) NOT NULL COMMENT 'Topic name',
    description VARCHAR(100) NOT NULL COMMENT 'Topic description'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN topic.topic_id IS
    'Topic ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN topic.name IS
    'Topic name'; */
```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN topic.description IS
    'Topic description'; */

ALTER TABLE topic ADD CONSTRAINT topic_pk PRIMARY KEY ( topic_id );

ALTER TABLE author_seminar
    ADD CONSTRAINT author_seminar_author_fk FOREIGN KEY ( auth_id )
        REFERENCES author ( auth_id );

ALTER TABLE author_seminar
    ADD CONSTRAINT author_seminar_seminar_fk FOREIGN KEY ( e_id )
        REFERENCES seminar ( e_id );

ALTER TABLE book_author
    ADD CONSTRAINT book_author_author_fk FOREIGN KEY ( auth_id )
        REFERENCES author ( auth_id );

ALTER TABLE book_author
    ADD CONSTRAINT book_author_book_fk FOREIGN KEY ( book_id )
        REFERENCES book ( book_id );

ALTER TABLE book
    ADD CONSTRAINT book_topic_fk FOREIGN KEY ( topic_id )
        REFERENCES topic ( topic_id );

ALTER TABLE copy
    ADD CONSTRAINT copy_book_fk FOREIGN KEY ( book_id )
        REFERENCES book ( book_id );

ALTER TABLE customer_exhibition
    ADD CONSTRAINT customer_fk FOREIGN KEY ( cust_id )
        REFERENCES customer ( cust_id );

ALTER TABLE customer_room
    ADD CONSTRAINT customer_room_customer_fk FOREIGN KEY ( cust_id )
        REFERENCES customer ( cust_id );

ALTER TABLE customer_room
    ADD CONSTRAINT customer_room_room_fk FOREIGN KEY ( room_id )
        REFERENCES room ( room_id );

ALTER TABLE event
    ADD CONSTRAINT event_topic_fk FOREIGN KEY ( topic_id )
        REFERENCES topic ( topic_id );

ALTER TABLE exhibition
    ADD CONSTRAINT exhibition_event_fk FOREIGN KEY ( e_id )
        REFERENCES event ( e_id );

ALTER TABLE invoice
    ADD CONSTRAINT invoice_rental_fk FOREIGN KEY ( rent_id )
        REFERENCES rental ( rent_id );
```

```
ALTER TABLE payment
    ADD CONSTRAINT payment_invoice_fk FOREIGN KEY ( invoice_id )
        REFERENCES invoice ( invoice_id );

ALTER TABLE rental
    ADD CONSTRAINT rental_copy_fk FOREIGN KEY ( copy_id )
        REFERENCES copy ( copy_id );

ALTER TABLE seminar
    ADD CONSTRAINT seminar_event_fk FOREIGN KEY ( e_id )
        REFERENCES event ( e_id );

ALTER TABLE sponsor_seminar
    ADD CONSTRAINT sponsor_seminar_seminar_fk FOREIGN KEY ( e_id )
        REFERENCES seminar ( e_id );

ALTER TABLE sponsor_seminar
    ADD CONSTRAINT sponsor_seminar_sponsor_fk FOREIGN KEY ( sponsor_id )
        REFERENCES sponsor ( sponsor_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_4_seminar;

DELIMITER //

CREATE TRIGGER arc_fkarc_4_seminar BEFORE
    INSERT OR UPDATE OF e_id ON seminar
    FOR EACH ROW
    DECLARE d VARCHAR(10);
BEGIN
    -- SQLINES LICENSE FOR EVALUATION USE ONLY
    SELECT
        a.type
    INTO d
    FROM
        event a
    WHERE
        a.e_id = :new.e_id;

    IF ( d IS NULL OR d <> 'Seminar' ) THEN
        raise_application_error(-20223, 'FK SEMINAR_EVENT_FK in Table
SEMINAR violates Arc constraint on Table EVENT - discriminator column TYPE
doesn''t have value ''Seminar'''
        );
    END IF;

    DECLARE EXIT HANDLER FOR not found BEGIN
        NULL;
    END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;
END;
```

```
/

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_4_exhibition;

DELIMITER //

CREATE TRIGGER arc_fkarc_4_exhibition BEFORE
    INSERT OR UPDATE OF e_id ON exhibition
    FOR EACH ROW
    DECLARE d VARCHAR(10);
BEGIN
    -- SQLINES LICENSE FOR EVALUATION USE ONLY
    SELECT
        a.type
    INTO d
    FROM
        event a
    WHERE
        a.e_id = :new.e_id;

    IF ( d IS NULL OR d <> 'Exhibition' ) THEN
        raise_application_error(-20223, 'FK EXHIBITION_EVENT_FK in Table
EXHIBITION violates Arc constraint on Table EVENT - discriminator column
TYPE doesn''t have value ''Exhibition'''
        );
    END IF;

    DECLARE EXIT HANDLER FOR not found BEGIN
        NULL;
    END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;
END;
/



-- SQLINES DEMO *** per Data Modeler Summary Report:
--
-- SQLINES DEMO ***                          18
-- SQLINES DEMO ***                           2
-- SQLINES DEMO ***                          36
-- SQLINES DEMO ***                           0
-- SQLINES DEMO ***                           0
-- SQLINES DEMO ***                           0
-- SQLINES DEMO *** DY                        0
-- SQLINES DEMO ***                           0
-- SQLINES DEMO ***                           0
-- SQLINES DEMO ***                           2
-- SQLINES DEMO ***                           0
-- SQLINES DEMO ***  TYPE                     0
-- SQLINES DEMO ***  TYPE                     0
```

```
-- SQLINES DEMO ***  TYPE BODY              0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO *** EGMENT                  0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO *** ED VIEW                 0
-- SQLINES DEMO *** ED VIEW LOG             0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
--
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
--
-- SQLINES DEMO ***                         0
--
-- SQLINES DEMO ***                         0
-- SQLINES DEMO *** A                       0
-- SQLINES DEMO *** T                       0
--
-- SQLINES DEMO ***                         0
-- SQLINES DEMO ***                         0
```

List of Tables

Listed below are tables generated by the Django ORM:

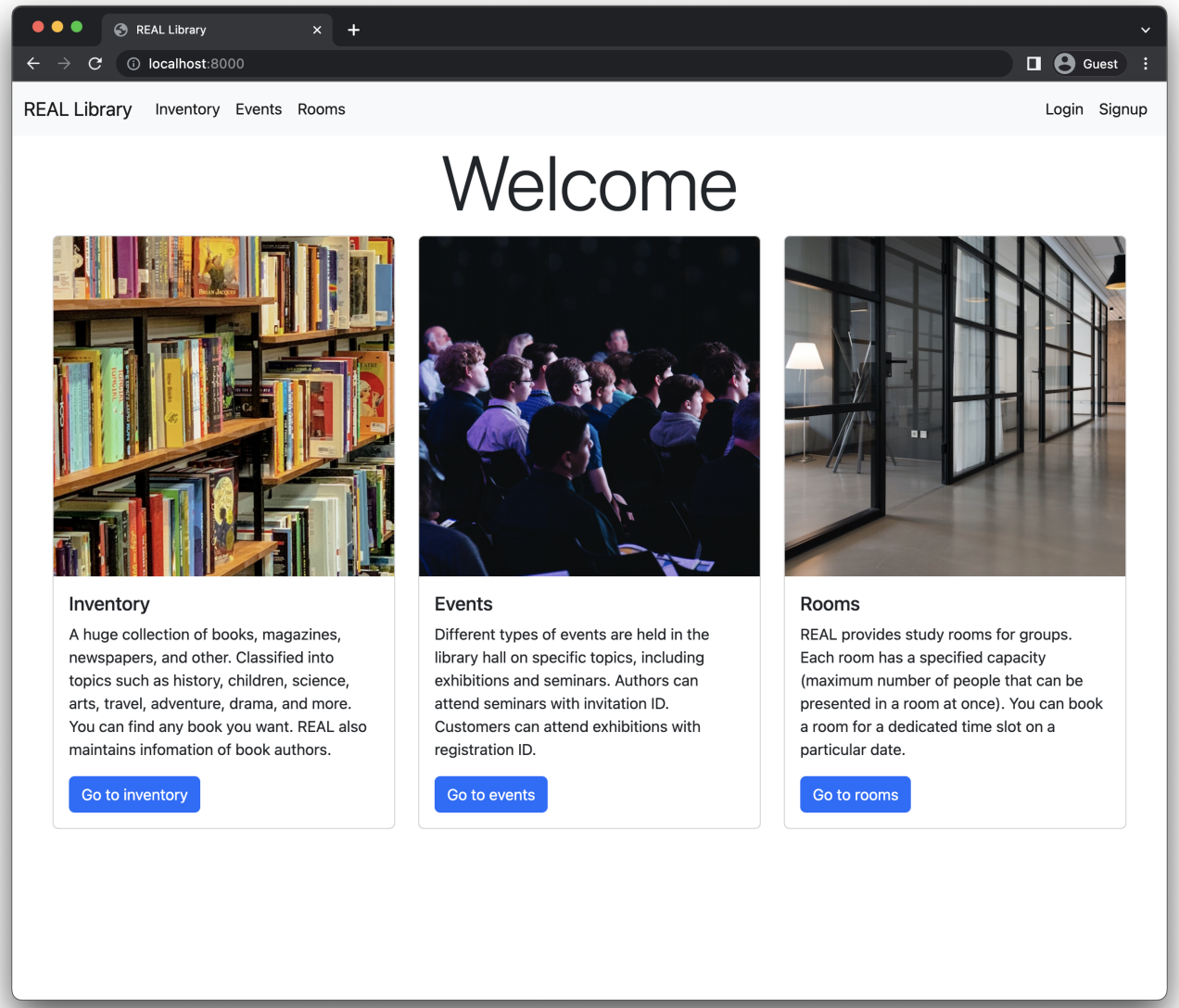
```
mysql> show tables;
+----------------------------+
| Tables_in_reallibrary      |
+----------------------------+
| auth_group                 |
| auth_group_permissions     |
| auth_permission            |
| auth_user                  |
| auth_user_groups           |
| auth_user_user_permissions |
| django_admin_log           |
| django_content_type        |
| django_migrations          |
| django_session             |
| events_exhibition          |
| events_exhibitionuser      |
| events_seminar             |
| events_seminarauthor       |
| events_seminarsponsor      |
| events_sponsor             |
| home_customer              |
| inventory_author           |
| inventory_book             |
| inventory_bookauthor       |
| inventory_copy             |
| inventory_invoice          |
| inventory_payment          |
| inventory_rental           |
| inventory_topic            |
| rooms_room                 |
| rooms_roomcustomer         |
+----------------------------+
27 rows in set (0.00 sec)
```

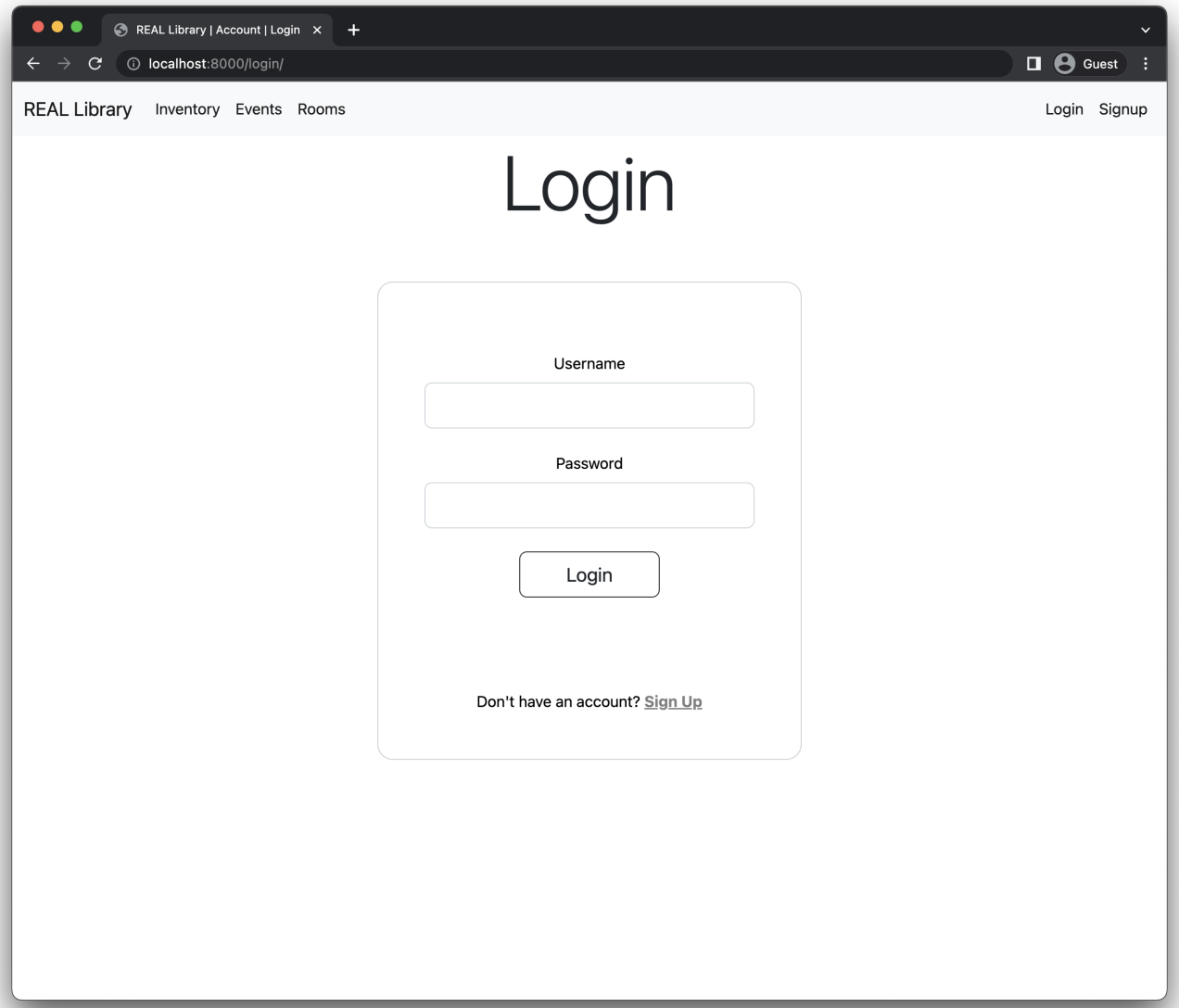
```
mysql> select count(*) from events_exhibition;
+----------+
| count(*) |
+----------+
|       12 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from events_exhibitionuser;;
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.01 sec)

ERROR:
No query specified

mysql> select count(*) from events_seminar;
+----------+
| count(*) |
+----------+
|        9 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from events_seminarauthor;
+----------+
| count(*) |
+----------+
|       10 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from events_seminarsponsor;
+----------+
| count(*) |
+----------+
|        6 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from events_sponsor;
+----------+
| count(*) |
+----------+
|       20 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from home_customer;
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from inventory_author;
+----------+
| count(*) |
+----------+
|       20 |
+----------+
1 row in set (0.00 sec)
```

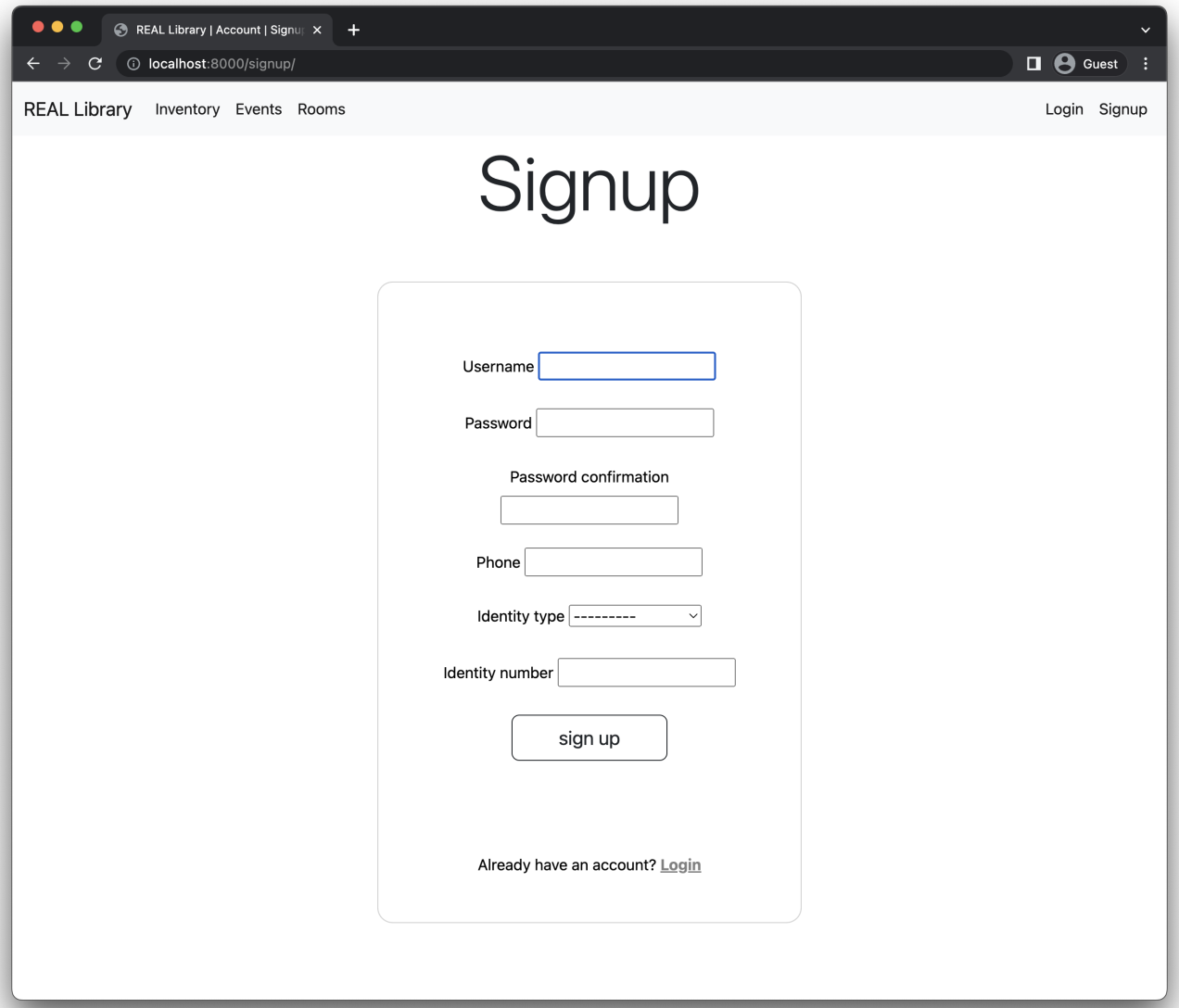
```
mysql> select count(*) from inventory_book;
+----------+
| count(*) |
+----------+
|       28 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from inventory_bookauthor;
+----------+
| count(*) |
+----------+
|       28 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from inventory_copy;
+----------+
| count(*) |
+----------+
|       79 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from inventory_invoice;
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from inventory_payment;
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from inventory_rental;
+----------+
| count(*) |
+----------+
|        8 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from inventory_topic;
+----------+
| count(*) |
+----------+
|       10 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from rooms_room;
+----------+
| count(*) |
+----------+
|       10 |
+----------+
1 row in set (0.01 sec)
```

```
mysql> select count(*) from rooms_roomcustomer;
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)
```

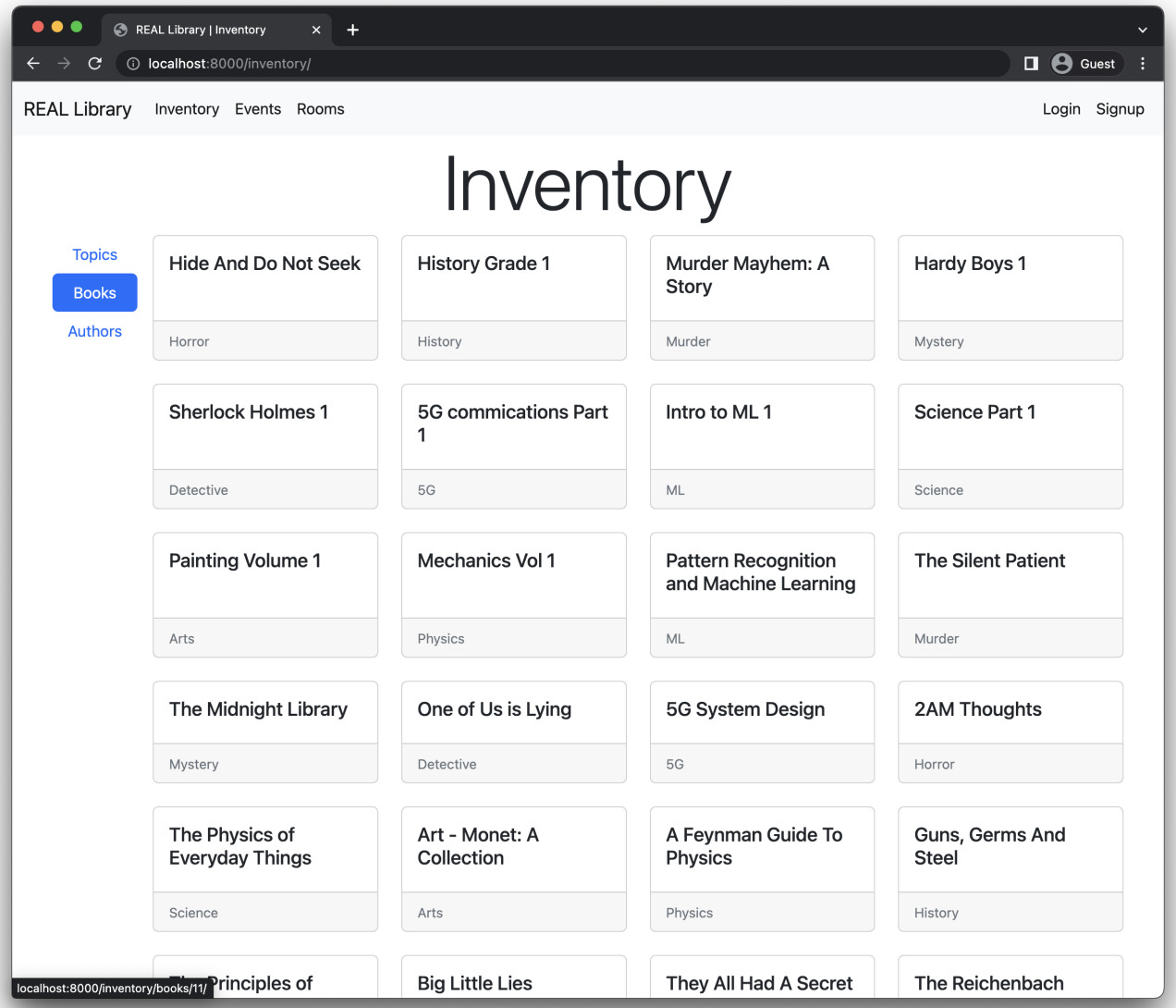
Screenshots

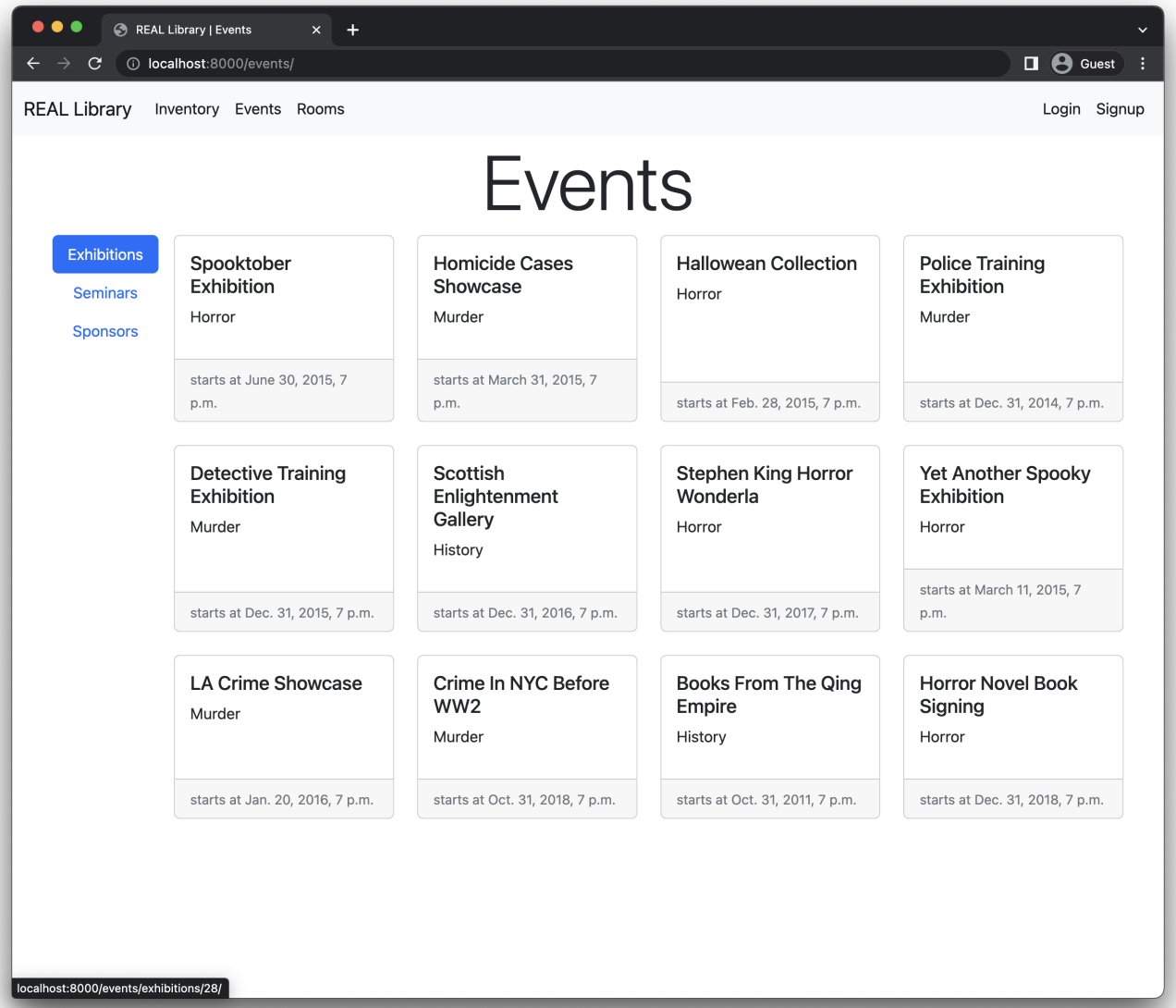

REAL Library
localhost:8000
Guest
REAL Library
Inventory
Events
Rooms
Login
Signup
Welcome
Inventory
A huge collection of books, magazines, newspapers, and other. Classified into topics such as history, children, science, arts, travel, adventure, drama, and more. You can find any book you want. REAL also maintains infomation of book authors.
Go to inventory
Events
Different types of events are held in the library hall on specific topics, including exhibitions and seminars. Authors can attend seminars with invitation ID. Customers can attend exhibitions with registration ID.
Go to events
Rooms
REAL provides study rooms for groups. Each room has a specified capacity (maximum number of people that can be presented in a room at once). You can book a room for a dedicated time slot on a particular date.
Go to rooms

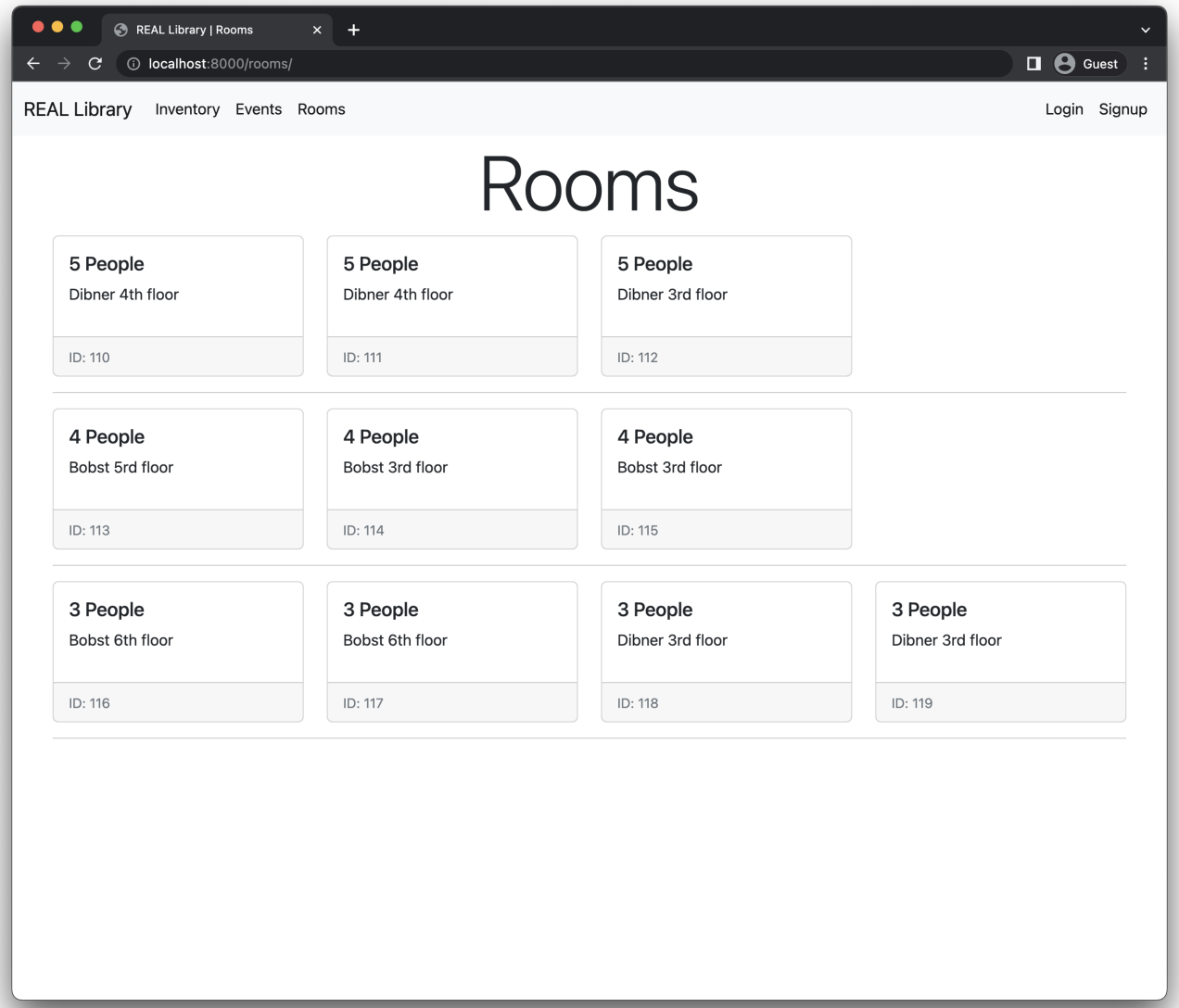

REAL Library | Account | Login
localhost:8000/login/
Guest
REAL Library
Inventory
Events
Rooms
Login
Signup
Login
Username
Password
Login
Don't have an account? Sign Up

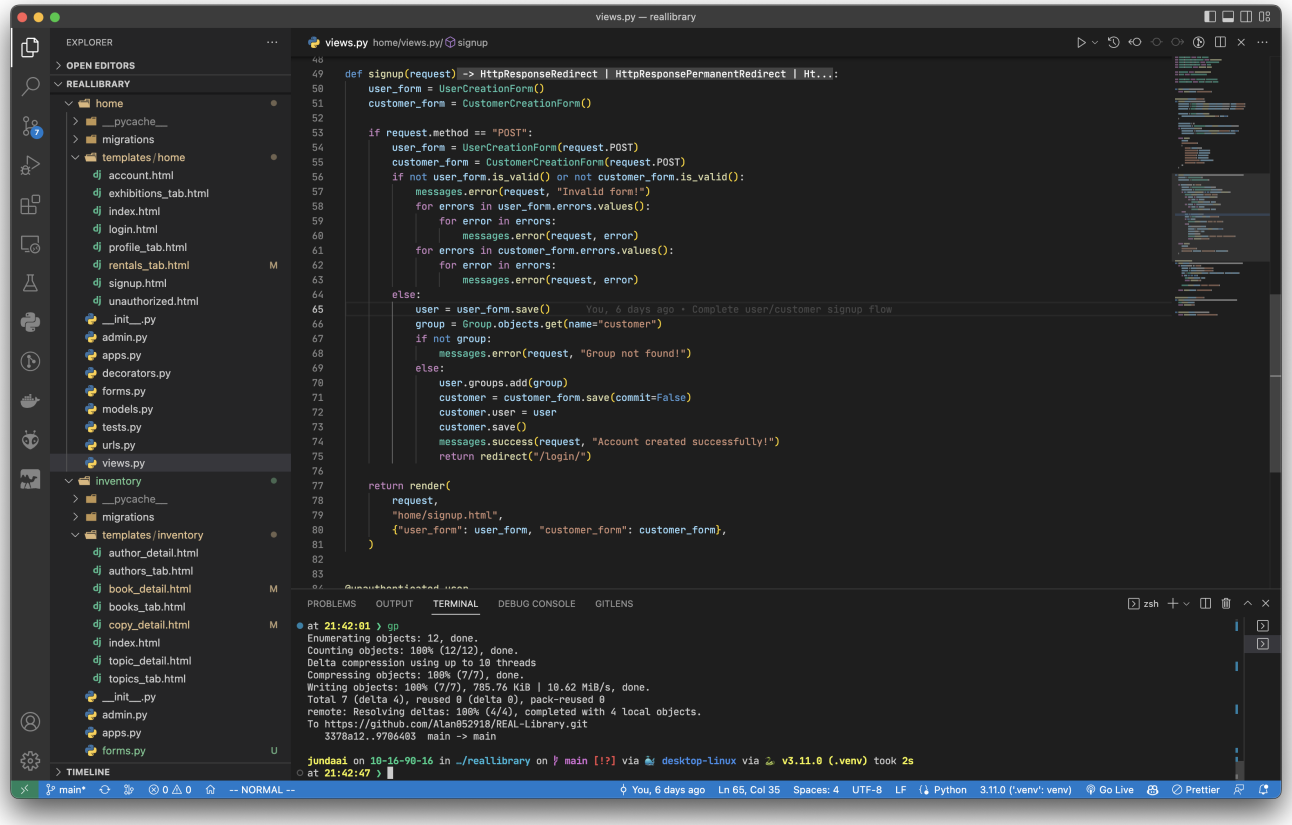

REAL Library | Account | Signup
localhost:8000/signup/
Guest
REAL Library
Inventory
Events
Rooms
Login
Signup
Signup
Username
Password
Password confirmation
Phone
Identity type

Identity number
sign up
Already have an account? Login


REAL Library | Inventory
localhost:8000/inventory/
Guest
REAL Library
Inventory
Events
Rooms
Login
Signup
Inventory
Topics
Books
Authors
Hide And Do Not Seek
Horror
History Grade 1
History
Murder Mayhem: A Story
Murder
Hardy Boys 1
Mystery
Sherlock Holmes 1
Detective
5G commications Part 1
5G
Intro to ML 1
ML
Science Part 1
Science
Painting Volume 1
Arts
Mechanics Vol 1
Physics
Pattern Recognition and Machine Learning
ML
The Silent Patient
Murder
The Midnight Library
Mystery
One of Us is Lying
Detective
5G System Design
5G
2AM Thoughts
Horror
The Physics of Everyday Things
Science
Art - Monet: A Collection
Arts
A Feynman Guide To Physics
Physics
Guns, Germs And Steel
History
Big Little Lies
They All Had A Secret
The Reichenbach
localhost:8000/inventory/books/11/


REAL Library | Events
localhost:8000/events/
Guest
REAL Library
Inventory
Events
Rooms
Login
Signup
Events
Exhibitions
Seminars
Sponsors
Spooktober Exhibition
Horror
starts at June 30, 2015, 7 p.m.
Homicide Cases Showcase
Murder
starts at March 31, 2015, 7 p.m.
Hallowean Collection
Horror
starts at Feb. 28, 2015, 7 p.m.
Police Training Exhibition
Murder
starts at Dec. 31, 2014, 7 p.m.
Detective Training Exhibition
Murder
starts at Dec. 31, 2015, 7 p.m.
Scottish Enlightenment Gallery
History
starts at Dec. 31, 2016, 7 p.m.
Stephen King Horror Wonderla
Horror
starts at Dec. 31, 2017, 7 p.m.
Yet Another Spooky Exhibition
Horror
starts at March 11, 2015, 7 p.m.
LA Crime Showcase
Murder
starts at Jan. 20, 2016, 7 p.m.
Crime In NYC Before WW2
Murder
starts at Oct. 31, 2018, 7 p.m.
Books From The Qing Empire
History
starts at Oct. 31, 2011, 7 p.m.
Horror Novel Book Signing
Horror
starts at Dec. 31, 2018, 7 p.m.
localhost:8000/events/exhibitions/28/


REAL Library | Rooms
localhost:8000/rooms/
Guest
REAL Library
Inventory
Events
Rooms
Login
Signup
Rooms
5 People
Dibner 4th floor
ID: 110
5 People
Dibner 4th floor
ID: 111
5 People
Dibner 3rd floor
ID: 112
4 People
Bobst 5rd floor
ID: 113
4 People
Bobst 3rd floor
ID: 114
4 People
Bobst 3rd floor
ID: 115
3 People
Bobst 6th floor
ID: 116
3 People
Bobst 6th floor
ID: 117
3 People
Dibner 3rd floor
ID: 118
3 People
Dibner 3rd floor
ID: 119



Security Features

CSRF Protection

Cross-site request forgery (CSRF) is a type of attack that occurs when a malicious web site, email, blog, instant message, or program causes a user's web browser to perform an unwanted action on a trusted site when the user is authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.

Django provides a CSRF protection mechanism that is enabled by default. It works by checking that the request contains a valid CSRF token, which is a randomly generated value that is tied to the user's session. The token is included in the page as a hidden field, and is also set as a cookie. When the user submits a form, the token is checked against the value in the cookie. If they match, the request is accepted. If they don't match, the request is rejected.

dj login.html home/templates/home/login.html

```
11  <div class="container py-5 h-100">
12    <div class="row d-flex justify-content-center align-items-center h-100">
13      <div class="col-12 col-md-8 col-lg-6 col-xl-5">
14        <div class="card bg-white text-black" style="border-radius: 1rem">
15          <div class="card-body p-5 text-center">
16            <form method="POST">
17              {% csrf_token %}
18              <div class="mb-md-5 mt-md-4 pb-5">
```

SQL Injection Protection

SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

Django's querysets are protected from SQL injection since their queries are constructed using query parameterization. This means that the query parameters are not embedded in the query string, but are passed as separate arguments to the database backend and escaped by the underlying database driver. This prevents the database backend from having to parse the query string, which is vulnerable to SQL injection attacks.

Clickjacking Protection

Clickjacking is a type of malicious exploit of a website where malicious code is injected into a legitimate web page to trick the user into clicking on a button or link that they would not otherwise click. The malicious code is invisible to the user and is rendered on top of the legitimate web page. The malicious code is executed when the user clicks on the button or link.

Django contains clickjacking protection in the form of the X-Frame-Options middleware which in a supporting browser will prevent a page from being loaded in a frame or iframe unless the page comes from the same origin as the parent page. This is done by setting the X-Frame-Options header to DENY. This is the default setting for the middleware.

SSL/HTTPS

HTTPS is a protocol for secure communication over a computer network, and is widely used on the Internet. HTTPS is an extension of the Hypertext Transfer Protocol (HTTP). In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS), or, formerly, its predecessor, Secure Sockets Layer (SSL). The protocol is therefore also often referred to as HTTP over TLS, or HTTP over SSL. With a few configuration steps Django can enable HTTPS for your site.

Lesson Learned

I learned a great deal of Django, MySQL, Bootstrap, and Docker in the process of doing this project. I learned how to model many-to-many relationships in Django with the `models.ManyToManyField()` and adding additional fields to the through model. And model inheritance with an abstract parent model. And how to filter model objects in Django views.

I also bumpped into a few issues with Docker compose, Django constantly failed to connect to MySQL server at first, after two days of research I found out that the problem was that the MySQL server was not ready when Django tried to connect to it. I solved this by adding a `healthcheck` option to the `db` service and a `depends_on` option to the `web` service in the `docker-compose.yml` file and specified that `web` not start until `db` service is healthy. So the Django app will not start until the MySQL server is ready for connection.

Finally I learned to style the template pages with Bootstrap, which is a very powerful and popular CSS framework that could be used in a plug-and-play fashion.

Business Analysis with Project Data

1. FOR QUERY 1 (Table joins with at least 3 tables in join)

Business information to retrieve: For each book that has a copy, display the name of the book, its topic name and the number of copies there are, in descending order of the number of copies.

```sql
-- Q1:

SELECT inventory_book.name,
       inventory_topic.name,
       COUNT(inventory_copy.book_id) as "Number of copies"
FROM inventory_book
         INNER JOIN inventory_topic ON inventory_book.topic_id = inventory_topic.id
         INNER JOIN inventory_copy ON inventory_book.id = inventory_copy.book_id
GROUP BY inventory_book.name, inventory_topic.name
ORDER BY COUNT(inventory_copy.book_id) DESC;
```

Services

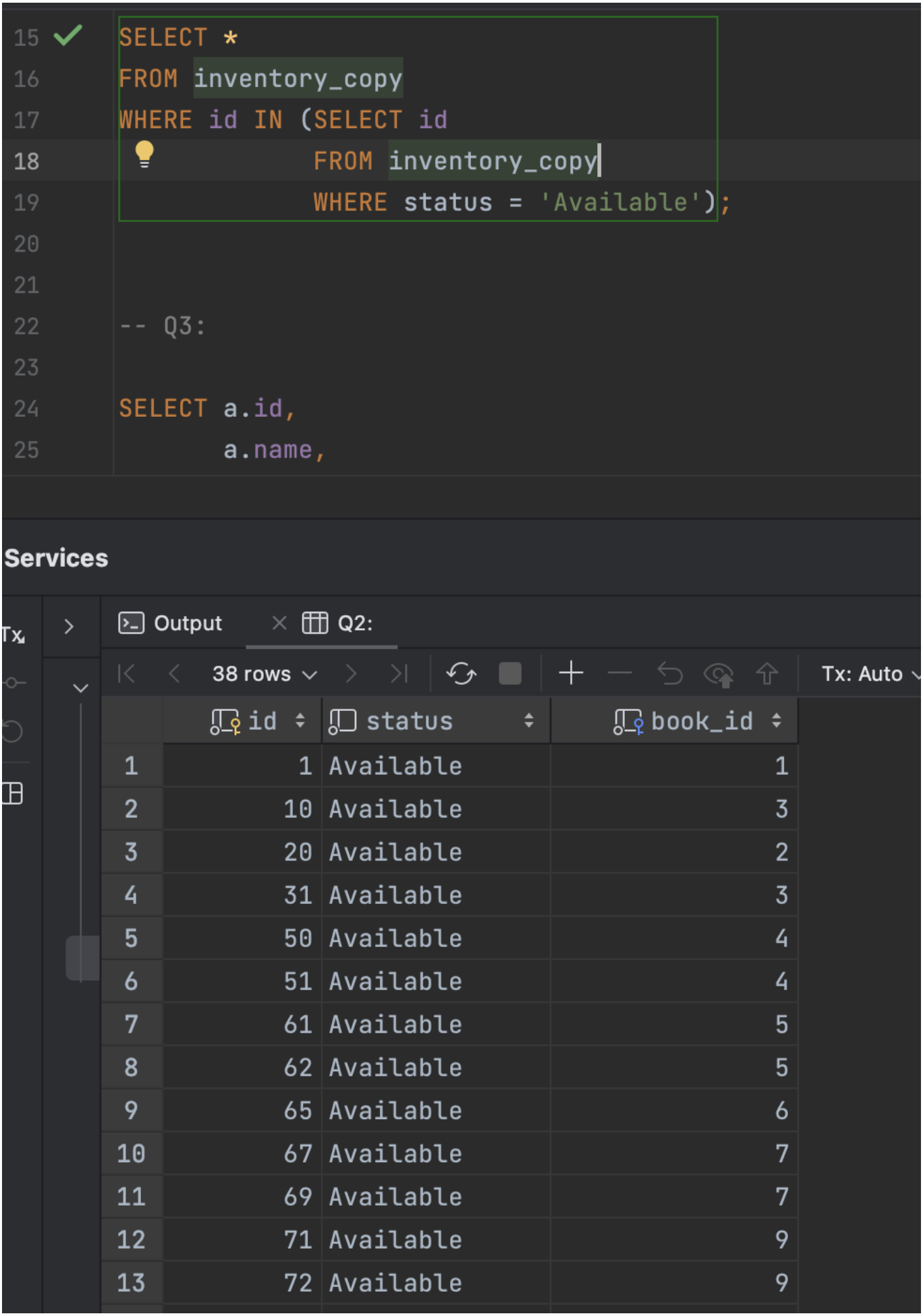
Output Q1:

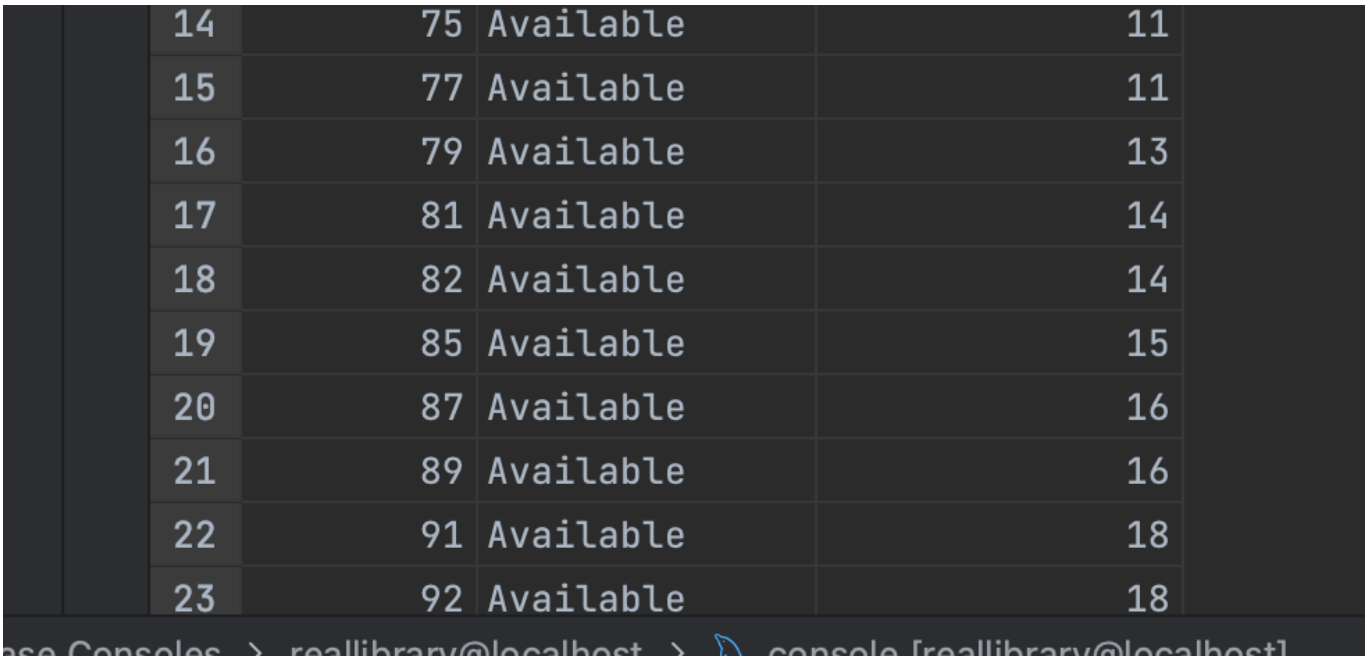
28 rows CSV

	inventory_book.name	inventory_topic.name	`Number of copies`
1	Sapiens: A Brief History of H…	History	5
2	Murder Mayhem: A Story	Murder	5
3	Hardy Boys 1	Mystery	5
4	One of Us is Lying	Detective	4
5	The Reichenbach Falls: A Sher…	Detective	4
6	The Selfish Gene	Science	4
7	Fundamentals Of Physics	Physics	4
8	5G commications Part 1	5G	3
9	2AM Thoughts	Horror	3
10	Pattern Recognition and Machi…	ML	3
11	Art - Monet: A Collection	Arts	3
12	They All Had A Secret	Mystery	3
13	The Principles of Deep Learni…	ML	3
14	A Feynman Guide To Physics	Physics	3
15	Hide And Do Not Seek	Horror	3
16	Intro to ML 1	ML	3
17	Sherlock Holmes 1	Detective	3
18	Painting Volume 1	Arts	2
19	The Midnight Library	Mystery	2
20	Mechanics Vol 1	Physics	2
21	Big Little Lies	Murder	2
22	5G System Design	5G	2
23	History Grade 1	History	2

2. FOR QUERY 2 (MULTI-ROW SUBQUERY)

Business information to retrieve: Find all the book copies, of any book, in the library that are still available to borrow

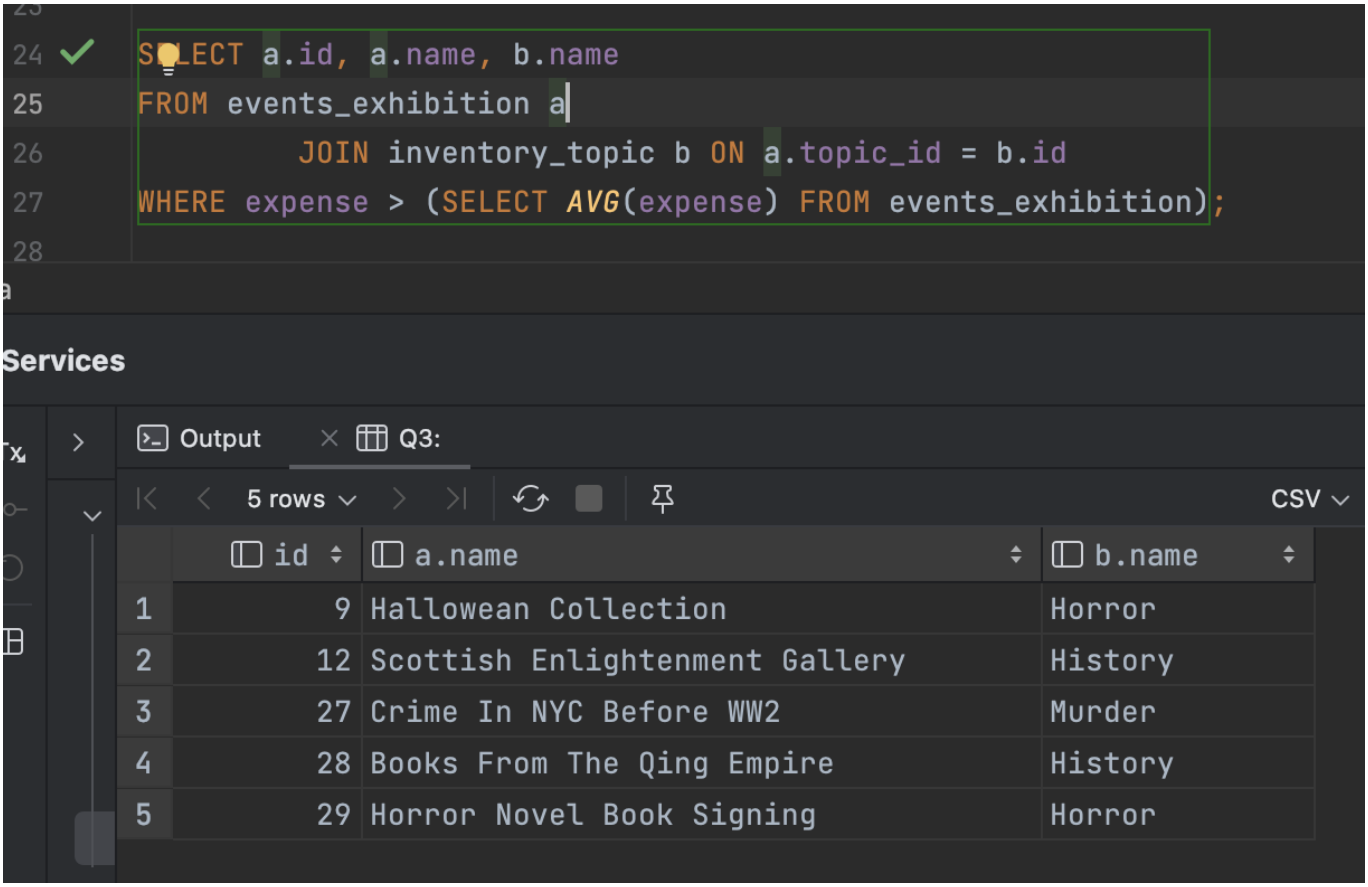




14	75	Available	11
15	77	Available	11
16	79	Available	13
17	81	Available	14
18	82	Available	14
19	85	Available	15
20	87	Available	16
21	89	Available	16
22	91	Available	18
23	92	Available	18

3. FOR QUERY 3 (CORRELATED QUERY)

Business information to retrieve: The following query finds exhibitions whose expense is greater than the average expense of all exhibitions, and lists its ID, name and topic name.



```
SELECT a.id, a.name, b.name
FROM events_exhibition a
        JOIN inventory_topic b ON a.topic_id = b.id
WHERE expense > (SELECT AVG(expense) FROM events_exhibition);
```

Services

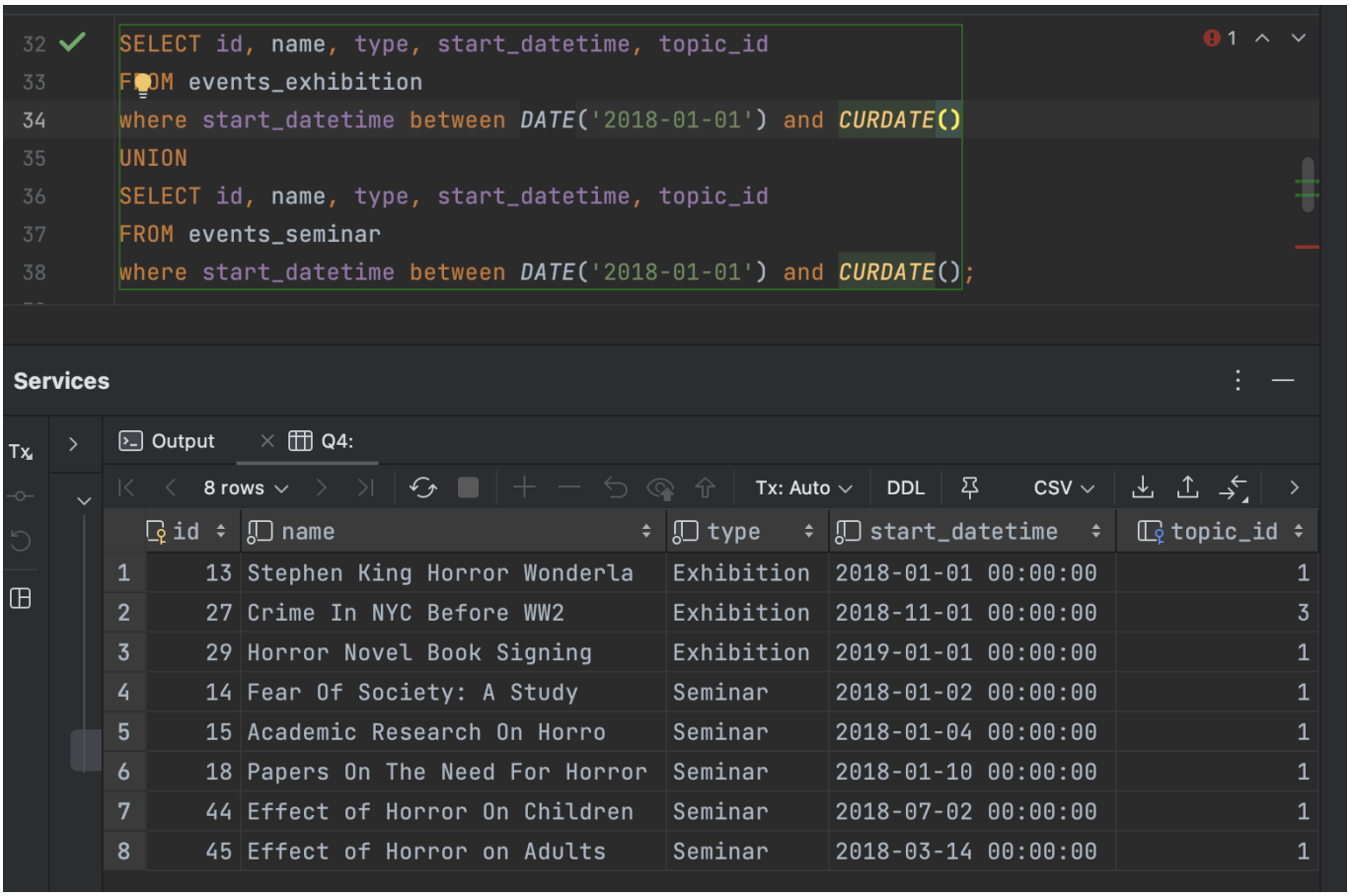
Output Q3:

5 rows CSV

	id	a.name	b.name
1	9	Hallowean Collection	Horror
2	12	Scottish Enlightenment Gallery	History
3	27	Crime In NYC Before WW2	Murder
4	28	Books From The Qing Empire	History
5	29	Horror Novel Book Signing	Horror

4. FOR QUERY 4 (SET-OPERATOR QUERY)

Business information to retrieve: Find and display the ID number, type, start time and topic ID for the events, be it an exhibition or a seminar, that start earliest on the 1st of January, 2018.



```sql
SELECT id, name, type, start_datetime, topic_id
FROM events_exhibition
where start_datetime between DATE('2018-01-01') and CURDATE()
UNION
SELECT id, name, type, start_datetime, topic_id
FROM events_seminar
where start_datetime between DATE('2018-01-01') and CURDATE();
```

Services

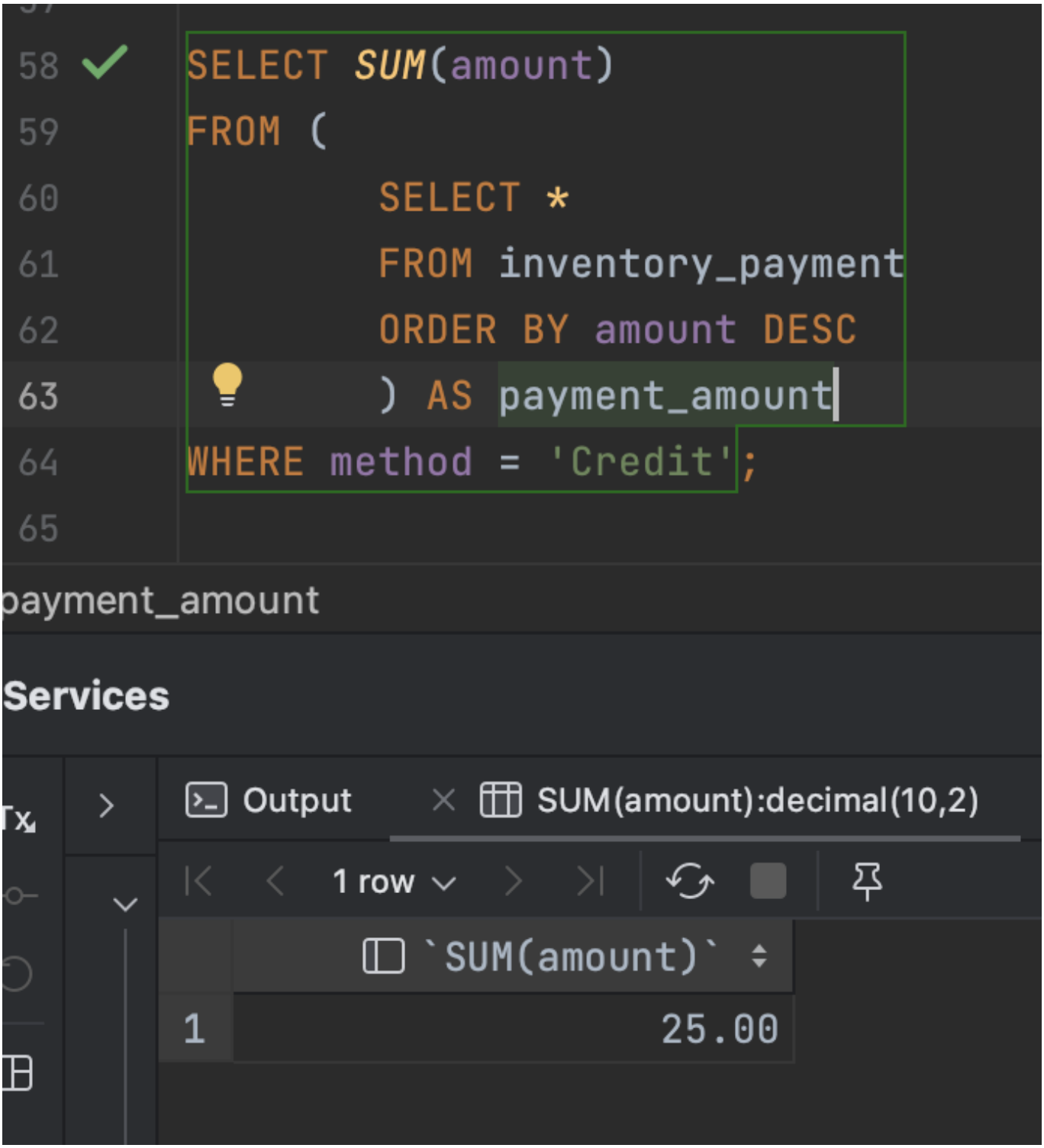
Output Q4:

8 rows Tx: Auto DDL CSV

	id	name	type	start_datetime	topic_id
1	13	Stephen King Horror Wonderla	Exhibition	2018-01-01 00:00:00	1
2	27	Crime In NYC Before WW2	Exhibition	2018-11-01 00:00:00	3
3	29	Horror Novel Book Signing	Exhibition	2019-01-01 00:00:00	1
4	14	Fear Of Society: A Study	Seminar	2018-01-02 00:00:00	1
5	15	Academic Research On Horro	Seminar	2018-01-04 00:00:00	1
6	18	Papers On The Need For Horror	Seminar	2018-01-10 00:00:00	1
7	44	Effect of Horror On Children	Seminar	2018-07-02 00:00:00	1
8	45	Effect of Horror on Adults	Seminar	2018-03-14 00:00:00	1

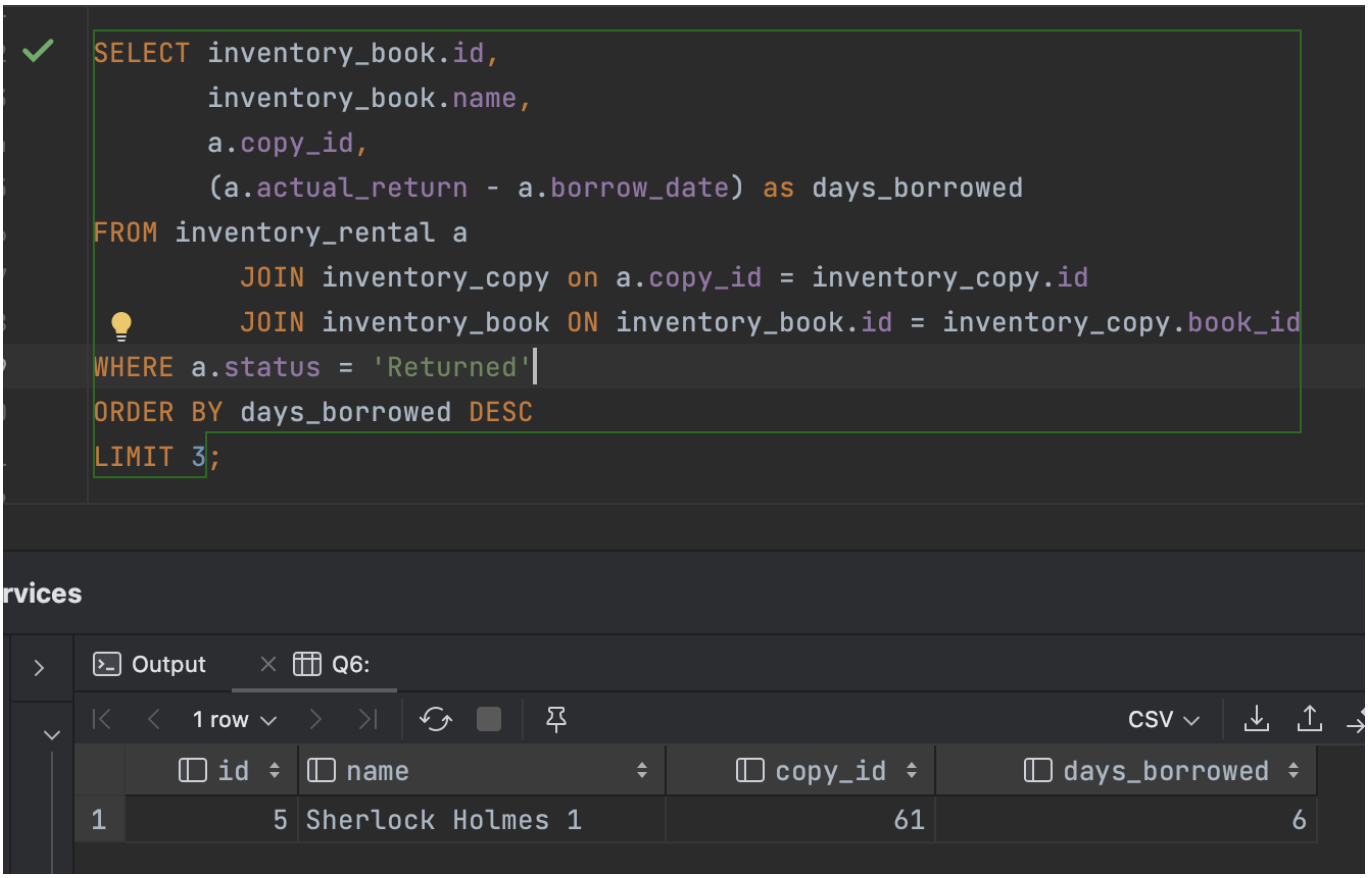
5. FOR QUERY 5 (Query with in line view or WITH clause)

Business information to retrieve: The following query finds, for the payments that were done using credit cards, the total amount paid or collected till now (useful for accounting).



6. FOR QUERY 6 (TOP-N)

Business information to retrieve: Find the top 3 book copies that were borrowed for the longest amount of time, and display the book id, book name, copy id and the number of days it was borrowed for.



Extra Features

- We added indexes to the Rental and Invoice tables to speed up the queries.
- We also orchestrated the app and database with Docker Compose.