

TANDON SCHOOL OF ENGINEERING  
PRINCIPLES OF DATABASE SYSTEMS  
CS-GY 6083 -B FALL 2022

---

GROUP PROJECT PART 2 REPORT  
REAL LIBRARY SYSTEM

---

Haiyue Zhao  
Siyuan Chen  
Haoru Wen

DATE: December 17, 2022

# CONTENTS

<b>Execution Summary:</b>	<b>3</b>
Business Case:	3
Logical Model:	4
Relational Model:	4
Assumptions:	4
Features:	5
Approaches:	6
Benefits:	6
<b>Technology Stack:</b>	<b>7</b>
<b>DDL:</b>	<b>7</b>
<b>List of Tables:</b>	<b>21</b>
<b>Screenshots of Web-Application:</b>	<b>21</b>
Login:	21
Signup:	22
Reset password:	23
Borrow Book:	24
My Borrow List:	24
BookList:	25
Select date for reservation:	25
My Invoice:	26
Search:	26
Rooms:	27
payment:	27
Data visualization: (total number of history of borrowings of each customer)	28
Transaction:	28
<b>Security and Reliability Features:</b>	<b>28</b>
<b>Business Analysis with SQL Queries:</b>	<b>29</b>
Q1 Table Joins with at Least 3 Tables in Join	29
Q2 Multi-row Subquery	30
Q3 Correlated Subquery	31
Q4 SET Operator Query	32



Q5 Query with In-Line View or WITH Clause	33
Q6 TOP-N Query	34
<b>Lesson/Reflection:</b>	<b>34</b>
<b>Extra Features:</b>	<b>35</b>

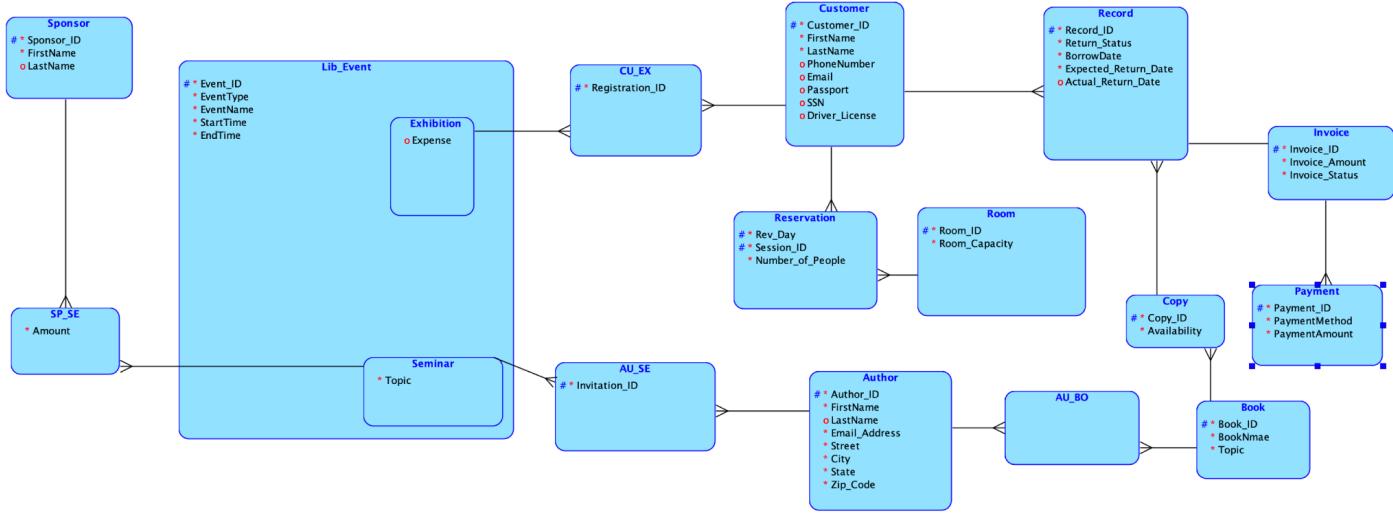
## Execution Summary:

### Business Case:

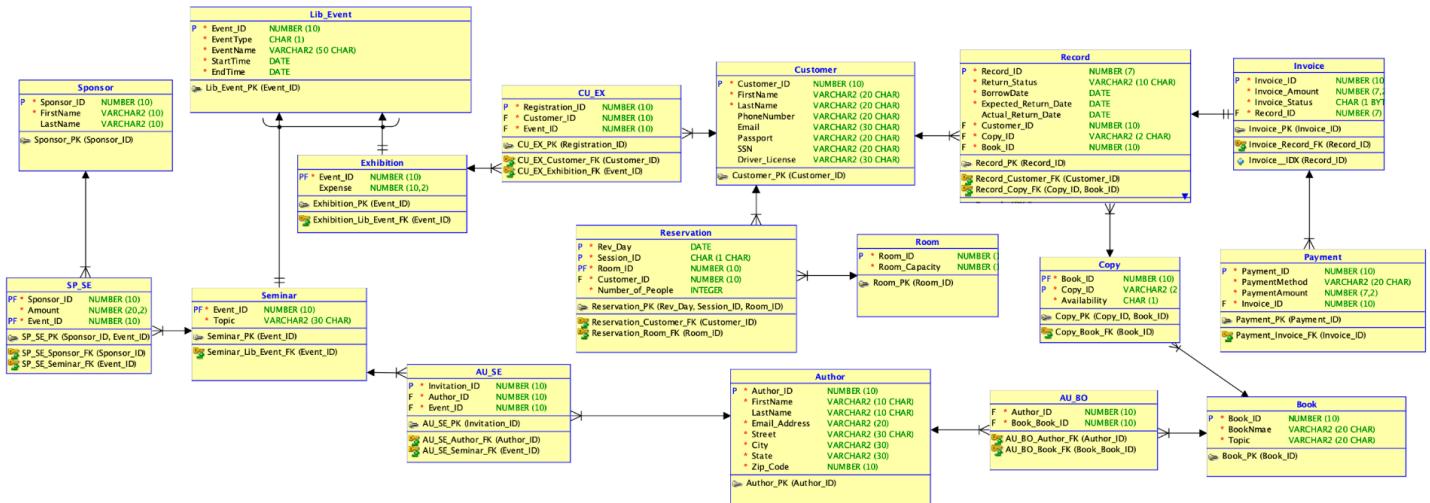
REAL (Read, Edify, And Learn) is a public library. It provides access to a wide range of books at affordable rental services, as well as free room reservation service. REAL is seeking to implement a new library management system to improve the efficiency of its operations. A new sophisticated centralized database system would follow detailed business rules, including:

- a. REAL provides access to a large number of books. Books are classified by topics such as history, science, adventures, drama, etc. Each book has one or more copies, and each copy has its Status (available or not). REAL keeps rental details such as Borrow Date, Expected Return Date, and Actual Return Date.
- b. REAL generates invoices with the Date and Amount. The amount is calculated based on Borrow Date, Expected Return Date, and Actual Return Date. Customers can make payments to the same invoice with a combination of payment methods.
- c. REAL provides Study Rooms for groups. Customers can make reservations with a time slot for a particular date.

## Logical Model:



## Relational Model:



## Assumptions:

1. Sponsor name can be separated into FirstName and LastName.
2. Exhibition and seminar are the subtypes of Lib\_event. Event type is used as the discriminator, and "E" is for exhibition, "R" is for the seminar.

3. Revday, session\_ID, and RoomID can be the primary key to reservations to resolve schedule conflicts. So if the customer reserve the room at a certain time, other people can not reserve the same room at the same time because of the primary key constraint.
4. Author's name can be separated into FirstName and LastName.
5. Actual Return Date can be optional for the records that haven't been returned. When a copy is borrowed, a row will be inserted in the record with the status of "Borrowed" and the actual\_return\_date of NULL. When it is returned, the actual return date and the status will be updated, and the corresponding invoice will be inserted.
6. Return status has three. One for borrowed, one for return, one for late.
7. Availability of the copy is Boolean. It stands for if the copy is available to borrow or not. When people borrow the book if the availability is 0, it will fail. When the copy is borrowed, its availability will be changed from 1 to 0. And when the copy is returned, its availability will be changed from 0 to 1.
8. The relationship between sponsor and seminar is many to many. An intersection table is added.
9. The relationship between the customer and the exhibition is many to many. An intersection table is added.
10. The relationship between the sponsor and seminar is many to many. An intersection table is added.
11. The relationship between the author and the book is many to many. An intersection table is added
12. One record should have one invoice. So, the record to invoice is one to one relationship.
13. The number of people in the reservation should not exceed the room capacity.
14. The expected return date and the actual return date cannot be earlier than the borrow date.
15. The status of the Invoice is boolean, and it is to show if the invoice has been paid off.

## Features:

Our library management system is designed to streamline and automate the process of managing rental and return services, payment, as well as room reservations. The system includes features such as:

- a. CRUD Functionalities (Create, Read, Update, Delete records)
- b. Transaction Concurrency
- c. Password Reset & Multi-factor Authentication
- d. Indexes
- e. Different levels of Authorization on Data (Customers vs. Administrator)
- f. Data Visualisation
- g. Cross-Site Scripting Attacks Protection
- h. SQL Injection Protection

## Approaches:

We implement the system in four phases:

Phase 1: Database setup. We worked on designing a suitable relational schema that can be used to store the data in the system, including analyzing the business cases, creating logical and relational models, generating DDL, adding constraints and triggers, and inserting testing data.

Phase 2: Website setup. During this step, we chose Django as the framework. We set up the connection with the database by pairing the HOST, PORT, NAME, USER, PASSWORD. We use python as the programming language to migrate the MySQL database and make connections with the local server. After the test connection is successful, we go to the next phase.

Phase 3: Website setupWe used HTML, CSS, and Bootstrap as the front-end framework to design the display and the layout of the webpage. For each page that displays records, we implemented tables, search boxes, and a pager to make the data retrieval more user-friendly.

Phase 4: Data communication from front-end to back-end. This part focuses on the user interaction and data communication. For example, after users create the account, the credentials will be encrypted and stored in the database and linked with customer\_id. We modified views.py by setting the parameters in POST requests and routers that determine the action after each click in url.py.

## Benefits:

Our new library management system provides a range of benefits.

First, it improves the efficiency of many tasks, such as book check-out and check-in, late fee tracking, and inventory management. Staff members can use the system to quickly locate and manage inventory and visualize data. This would free up staff time to focus on more high-value activities such as assisting customers.

Second, it provides an enhanced customer experience. The system includes a user-friendly interface for customers to browse and search for books, borrow and return books, and make combinations of payments. Customers are able to find resources easily.

Third, it will increase data accuracy. The system provides a centralized database of all library data, including book records and customer information. Staff can quickly locate and manage data in a safer way.

Overall, the implementation reduces the time and effort required to manage the library's operations, and it provides a more convenient experience for customers and staff.

## Technology Stack:

Design	Oracle Data Modeler
Code Editor	Pycharm
Database	MySQL
Backend	Python, Django
Fronted	jQuery, Bootstrap, HTML
Server	Local Server

## DDL:

```
-- SQLINES DEMO *** le SQL Developer Data Modeler 22.2.0.165.1149
-- SQLINES DEMO *** -11-04 16:10:08 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 au_bo (
    author_id  BIGINT NOT NULL,
    book_book_id BIGINT NOT NULL
);
```

```
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE au_se (
    invitation_id BIGINT NOT NULL COMMENT 'INVITATION ID OF AUTHOR',
```

```

author_id    BIGINT NOT NULL,
event_id     BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN au_se.invitation_id IS
'INVITATION ID OF AUTHOR'; */

ALTER TABLE au_se ADD CONSTRAINT au_se_pk PRIMARY KEY ( invitation_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE author (
    author_id    BIGINT NOT NULL COMMENT 'UNIQUE AUTHOR ID',
    firstname    VARCHAR(10) NOT NULL COMMENT 'FIRST NAME OF AUTHOR',
    lastname     VARCHAR(10) COMMENT 'LAST NAME OF AUTHOR',
    email_address VARCHAR(20) NOT NULL COMMENT 'EMAIL ADDRESS OF THE
AUTHOR',
    street       VARCHAR(30) NOT NULL COMMENT 'STREET ADDRESS',
    city         VARCHAR(30) NOT NULL COMMENT 'CITY ADDRESS',
    state        VARCHAR(30) NOT NULL COMMENT 'STATE',
    zip_code     BIGINT NOT NULL COMMENT 'ZIP CODE OF ADDRESS'
);

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

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.firstname IS
'FIRST NAME OF AUTHOR'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.lastname IS
'LAST NAME OF AUTHOR'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.email_address IS
'EMAIL ADDRESS OF THE AUTHOR'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.street IS
'STREET ADDRESS'; */

/* Moved to CREATE TABLE

```

```
COMMENT ON COLUMN author.city IS
'CITY ADDRESS'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.state IS
'STATE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN author.zip_code IS
'ZIP CODE OF ADDRESS'; */

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

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE book (
    book_id BIGINT NOT NULL COMMENT 'UNIQUE BOOK ID',
    booknmae VARCHAR(20) NOT NULL COMMENT 'NAME OF THE BOOK',
    topic   VARCHAR(20) NOT NULL COMMENT 'TOPIC OF THE BOOK'
);

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

/* Moved to CREATE TABLE
COMMENT ON COLUMN book.booknmae IS
'NAME OF THE BOOK'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN book.topic IS
'TOPIC OF THE BOOK'; */

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

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE copy (
    book_id    BIGINT NOT NULL,
    copy_id    VARCHAR(2) NOT NULL COMMENT 'UNIQUE COPY ID',
    availability DOUBLE NOT NULL COMMENT 'AVAILABILITY STATUS OF THE COPY'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN copy.copy_id IS
'UNIQUE COPY ID'; */
```

```

/* Moved to CREATE TABLE
COMMENT ON COLUMN copy.availability IS
'AVAILABILITY STATUS OF THE COPY'; */

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

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE cu_ex (
    registration_id BIGINT NOT NULL COMMENT 'UNIQUE REGISTRATION ID',
    customer_id    BIGINT NOT NULL,
    event_id       BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN cu_ex.registration_id IS
'UNIQUE REGISTRATION ID'; */

ALTER TABLE cu_ex ADD CONSTRAINT cu_ex_pk PRIMARY KEY ( registration_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE customer (
    customer_id   BIGINT NOT NULL COMMENT 'UNIQUE CUSTOMER ID',
    firstname     VARCHAR(20) NOT NULL COMMENT 'FIRST NAME OF CUSTOMER',
    lastname      VARCHAR(20) NOT NULL COMMENT 'LAST NAME OF CUSTOMER',
    phonenumber   VARCHAR(20) COMMENT 'PHONE NAME OF CUSTOMER',
    email         VARCHAR(30) COMMENT 'EMAIL OF CUSTOMER',
    passport      VARCHAR(20) COMMENT 'PASSPORT OF CUSTOMER',
    ssn          VARCHAR(20) COMMENT 'SSN OF CUSTOMER',
    driver_license VARCHAR(30) COMMENT 'DRIVER'S LICENSE OF CUSTOMER'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.customer_id IS
'UNIQUE CUSTOMER ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.firstname IS
'FIRST NAME OF CUSTOMER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.lastname IS
'LAST NAME OF CUSTOMER'; */

```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.phonenumber IS
'PHONE NAME OF CUSTOMER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.email IS
'EMAIL OF CUSTOMER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.passport IS
'PASSPORT OF CUSTOMER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.ssn IS
'SSN OF CUSTOMER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN customer.driver_license IS
'DRIVER"S LICENSE OF CUSTOMER'; */

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

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE exhibition (
    event_id BIGINT NOT NULL COMMENT 'UNIQUE ID OF THE EVENT',
    expense DECIMAL(10, 2) COMMENT 'EXPENSE OF THE EXHIBITION'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN exhibition.event_id IS
'UNIQUE ID OF THE EVENT'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN exhibition.expense IS
'EXPENSE OF THE EXHIBITION'; */

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

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE invoice (
    invoice_id    BIGINT NOT NULL COMMENT 'UNIQUE INVOICE ID',
    invoice_amount DECIMAL(7, 2) NOT NULL COMMENT 'AMOUNT OF THE INVOICE',
    invoice_status CHAR(1) NOT NULL COMMENT 'STATUS OF THE INVOCIE',
```

```

record_id    INT NOT NULL
);

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

/* Moved to CREATE TABLE
COMMENT ON COLUMN invoice.invoice_amount IS
'AMOUNT OF THE INVOICE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN invoice.invoice_status IS
'STATUS OF THE INVOCIE'; */

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

```

ALTER TABLE invoice ADD CONSTRAINT invoice\_pk PRIMARY KEY ( invoice\_id );

```

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE lib_event (
    event_id  BIGINT NOT NULL COMMENT 'UNIQUE ID OF THE EVENT',
    eventtype CHAR(1) NOT NULL COMMENT 'TYPE OF THE EVENT',
    eventname VARCHAR(50) NOT NULL COMMENT 'NAME OF THE EVENT',
    starttime DATETIME NOT NULL COMMENT 'START TIME OF THE EVENT',
    endtime  DATETIME NOT NULL COMMENT 'END TIME OF THE EVENT'
);

```

ALTER TABLE lib\_event

```

    ADD CONSTRAINT ch_inh_lib_event CHECK ( eventtype IN ( 'E', 'R' ) );

```

```

/* Moved to CREATE TABLE
COMMENT ON COLUMN lib_event.event_id IS
'UNIQUE ID OF THE EVENT'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN lib_event.eventtype IS
'TYPE OF THE EVENT'; */

/* Moved to CREATE TABLE

```

```

COMMENT ON COLUMN lib_event.eventname IS
  'NAME OF THE EVENT'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN lib_event.starttime IS
  'START TIME OF THE EVENT'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN lib_event.endtime IS
  'END TIME OF THE EVENT'; */

ALTER TABLE lib_event ADD CONSTRAINT lib_event_pk PRIMARY KEY ( event_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE payment (
  payment_id    BIGINT NOT NULL COMMENT 'UNIQUE PAYMENT ID',
  paymentmethod VARCHAR(20) NOT NULL COMMENT 'PAYMENT METHOD: CASH,
CREDIT, DEBIT, PAYPAL',
  paymentamount DECIMAL(7, 2) NOT NULL COMMENT 'AMOUNT OF THE PAYMENT',
  invoice_id    BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.payment_id IS
  'UNIQUE PAYMENT ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.paymentmethod IS
  'PAYMENT METHOD: CASH, CREDIT, DEBIT, PAYPAL'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN payment.paymentamount IS
  'AMOUNT OF THE PAYMENT'; */

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

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE record (
  record_id      INT NOT NULL COMMENT 'UNIQUE RECORD ID',
  return_status   VARCHAR(10) NOT NULL COMMENT 'RETURN STATUS OF THE
COPY',
  borrowdate     DATETIME NOT NULL COMMENT 'BORROWED DATE OF COPY',
  expected_return_date DATETIME NOT NULL COMMENT 'EXPECTED RETURN DATE OF
THE COPY',

```

```

actual_return_date DATETIME COMMENT 'ACUTUAL RETURN DATE OF THE COPY',
customer_id      BIGINT NOT NULL,
copy_id          VARCHAR(2) NOT NULL,
book_id          BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN record.record_id IS
'UNIQUE RECORD ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN record.return_status IS
'RETURN STATUS OF THE COPY'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN record.borrowdate IS
'BORROWED DATE OF COPY'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN record.expected_return_date IS
'EXPECTED RETURN DATE OF THE COPY'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN record.actual_return_date IS
'ACUTUAL RETURN DATE OF THE COPY'; */

```

```

ALTER TABLE record ADD CONSTRAINT record_pk PRIMARY KEY ( record_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE reservation (
    rev_day      DATETIME NOT NULL COMMENT 'DATE OF THE RESERVATION',
    session_id   CHAR(1) NOT NULL COMMENT 'TIME SLOT OF THE EVENT',
    room_id      BIGINT NOT NULL,
    customer_id  BIGINT NOT NULL,
    number_of_people INTEGER NOT NULL COMMENT 'NUMBER OF PEOPLE FOR THE
RESERVATION'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN reservation.rev_day IS
'DATE OF THE RESERVATION'; */

```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN reservation.session_id IS
'TIME SLOT OF THE EVENT'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN reservation.number_of_people IS
'NUMBER OF PEOPLE FOR THE RESERVATION'; */

ALTER TABLE reservation
ADD CONSTRAINT reservation_pk PRIMARY KEY ( rev_day,
                                             session_id,
                                             room_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE room (
    room_id      BIGINT NOT NULL COMMENT 'UNIQUE ROOM ID',
    room_capacity BIGINT NOT NULL COMMENT 'CAPACITY OF ROOM'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN room.room_id IS
'UNIQUE ROOM ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN room.room_capacity IS
'CAPACITY OF ROOM'; */

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

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE seminar (
    event_id BIGINT NOT NULL COMMENT 'UNIQUE ID OF THE EVENT',
    topic   VARCHAR(30) NOT NULL COMMENT 'TOPIC OF SEMINAR'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN seminar.event_id IS
'UNIQUE ID OF THE EVENT'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN seminar.topic IS
'TOPIC OF SEMINAR'; */

ALTER TABLE seminar ADD CONSTRAINT seminar_pk PRIMARY KEY ( event_id );
```

```
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE sp_se (
    sponsor_id BIGINT NOT NULL,
    amount    DECIMAL(20, 2) NOT NULL COMMENT 'AMOUNT SPONSOR SUPPORT',
    event_id  BIGINT NOT NULL
);
/* Moved to CREATE TABLE
COMMENT ON COLUMN sp_se.amount IS
'AMOUNT SPONSOR SUPPORT'; */

ALTER TABLE sp_se ADD CONSTRAINT sp_se_pk PRIMARY KEY ( sponsor_id,
    event_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE sponsor (
    sponsor_id BIGINT NOT NULL COMMENT 'UNIQUE SPONSOR ID',
    firstname  VARCHAR(10) NOT NULL COMMENT 'FIRST NAME OF SPONSOR',
    lastname   VARCHAR(10) COMMENT 'LAST NAME OF SPONSOR'
);
/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor.sponsor_id IS
'UNIQUE SPONSOR ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor.firstname IS
'FIRST NAME OF SPONSOR'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN sponsor.lastname IS
'LAST NAME OF SPONSOR'; */

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

ALTER TABLE au_bo
    ADD CONSTRAINT au_bo_author_fk FOREIGN KEY ( author_id )
        REFERENCES author ( author_id );

ALTER TABLE au_bo
    ADD CONSTRAINT au_bo_book_fk FOREIGN KEY ( book_book_id )
        REFERENCES book ( book_id );
```

```
ALTER TABLE au_se
  ADD CONSTRAINT au_se_author_fk FOREIGN KEY ( author_id )
    REFERENCES author ( author_id );
```

```
ALTER TABLE au_se
  ADD CONSTRAINT au_se_seminar_fk FOREIGN KEY ( event_id )
    REFERENCES seminar ( event_id );
```

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

```
ALTER TABLE cu_ex
  ADD CONSTRAINT cu_ex_customer_fk FOREIGN KEY ( customer_id )
    REFERENCES customer ( customer_id );
```

```
ALTER TABLE cu_ex
  ADD CONSTRAINT cu_ex_exhibition_fk FOREIGN KEY ( event_id )
    REFERENCES exhibition ( event_id );
```

```
ALTER TABLE exhibition
  ADD CONSTRAINT exhibition_lib_event_fk FOREIGN KEY ( event_id )
    REFERENCES lib_event ( event_id );
```

```
ALTER TABLE invoice
  ADD CONSTRAINT invoice_record_fk FOREIGN KEY ( record_id )
    REFERENCES record ( record_id );
```

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

```
ALTER TABLE record
  ADD CONSTRAINT record_copy_fk FOREIGN KEY ( copy_id,
    book_id )
    REFERENCES copy ( copy_id,
      book_id );
```

```
ALTER TABLE record
  ADD CONSTRAINT record_customer_fk FOREIGN KEY ( customer_id )
    REFERENCES customer ( customer_id );
```

```
ALTER TABLE reservation
  ADD CONSTRAINT reservation_customer_fk FOREIGN KEY ( customer_id )
```

```

REFERENCES customer ( customer_id );

ALTER TABLE reservation
    ADD CONSTRAINT reservation_room_fk FOREIGN KEY ( room_id )
        REFERENCES room ( room_id );

ALTER TABLE seminar
    ADD CONSTRAINT seminar_lib_event_fk FOREIGN KEY ( event_id )
        REFERENCES lib_event ( event_id );

ALTER TABLE sp_se
    ADD CONSTRAINT sp_se_seminar_fk FOREIGN KEY ( event_id )
        REFERENCES seminar ( event_id );

ALTER TABLE sp_se
    ADD CONSTRAINT sp_se_sponsor_fk FOREIGN KEY ( sponsor_id )
        REFERENCES sponsor ( sponsor_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_3_exhibition;

DELIMITER //

CREATE TRIGGER arc_fkarc_3_exhibition BEFORE
    INSERT OR UPDATE OF event_id ON exhibition
    FOR EACH ROW
    DECLARE d CHAR(1);
BEGIN
    -- SQLINES LICENSE FOR EVALUATION USE ONLY
    SELECT
        a.eventtype
    INTO d
    FROM
        lib_event a
    WHERE
        a.event_id = :new.event_id;

    IF ( d IS NULL OR d <> 'E' ) THEN
        raise_application_error(-20223, 'FK Exhibition.Lib_Event_FK in Table Exhibition violates
Arc constraint on Table Lib_Event - discriminator column EventType doesn''t have value "E"'
    );
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_3_seminar;

DELIMITER //

CREATE TRIGGER arc_fkarc_3_seminar BEFORE
    INSERT OR UPDATE OF event_id ON seminar
    FOR EACH ROW
    DECLARE d CHAR(1);
BEGIN
    -- SQLINES LICENSE FOR EVALUATION USE ONLY
    SELECT
        a.eventtype
    INTO d
    FROM
        lib_event a
    WHERE
        a.event_id = :new.event_id;

    IF ( d IS NULL OR d <> 'R' ) THEN
        raise_application_error(-20223, 'FK Seminar_Lib_Event_FK in Table Seminar violates Arc constraint on Table Lib_Event - discriminator column EventType doesn''t have value "R"');
    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 \*\*\* 17  
-- SQLINES DEMO \*\*\* 1  
-- SQLINES DEMO \*\*\* 34  
-- 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 \*\*\* 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:

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Increm...	Create Time
au_bo	InnoDB	10	Dynamic	10	1638	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2022-11-04
au_se	InnoDB	10	Dynamic	9	1820	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2022-11-04
auth_group	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0	2022-11-27
auth_group_permissions	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2022-11-27
auth_permission	InnoDB	10	Dynamic	24	682	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	24	2022-11-27
auth_user	InnoDB	10	Dynamic	8	2048	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	26	2022-11-27
auth_user_groups	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	1	2022-11-27
auth_user_user_permis...	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	1	2022-11-27
author	InnoDB	10	Dynamic	11	1489	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04
book	InnoDB	10	Dynamic	11	1489	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04
copy	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0	2022-11-27
cu_ex	InnoDB	10	Dynamic	10	1638	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2022-11-04
customer	InnoDB	10	Dynamic	11	1489	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04
django_admin_log	InnoDB	10	Dynamic	4	4096	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	4	2022-11-27
django_content_type	InnoDB	10	Dynamic	8	2048	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	8	2022-11-27
django_migrations	InnoDB	10	Dynamic	18	910	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	19	2022-11-27
django_session	InnoDB	10	Dynamic	2	8192	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0	2022-11-27
exhibition	InnoDB	10	Dynamic	10	1638	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04
Invoice	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0	2022-11-04
lib_event	InnoDB	10	Dynamic	25	655	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04
payment	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0	2022-11-04
record	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2022-11-27
reservation	InnoDB	10	Dynamic	11	1489	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2022-11-04
room	InnoDB	10	Dynamic	11	1489	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04
seminar	InnoDB	10	Dynamic	12	1365	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04
SP_SE	InnoDB	10	Dynamic	10	1638	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0	2022-11-04
sponsor	InnoDB	10	Dynamic	11	1489	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2022-11-04

## Screenshots of Web-Application:

### Login:

localhost:8000/users/login/

USER LOGIN

username: halyue

password:

Remember me

does not have an account? [sign up](#)

Forgot your password? [Reset Password](#)

## Signup:

localhost:8000/users/register/

USER REGISTER

Username:

Password

firstname: Haoru

lastname: Wen

mobile: 9876553444

email: abcd123@nyu.edu

passport: 99884412451234

[register](#)

already have an account? [login](#)

## Reset password:

Django administration

Home > Password reset

Password reset

Forgotten your password? Enter your email address below, and we'll email instructions for setting a new one.

Email address:

### Django: Reset your account password 收件箱 ×

siyuan chen <siyuannyulife@163.com>

12月14日周三 11:39 (3天前) 发送至 我

Hello,

We received a request to reset the password for your account for this email address. To initiate the password reset process for your account, click the link below.

[http://localhost:8000/users/password\\_reset/Mptk/193gf89qhw1h3r834rh9123hr9fht412ht829fh/](http://localhost:8000/users/password_reset/Mptk/193gf89qhw1h3r834rh9123hr9fht412ht829fh/)

This link can only be used once. If you need to reset your password again, please visit request another reset.

If you did not make this request, you can simply ignore this email.

Sincerely,  
The Website Team

## Borrow Book:

The screenshot shows the 'Book List' page of the Library Management System. The left sidebar has links for 'My Borrow List', 'Book List' (which is selected), 'My Invoices', and 'Rooms'. The main content area is titled 'Book List' and shows a table of books with 10 records per page. The columns are 'Book Name', 'Topic', and 'View Copies'. The books listed are: 1984 (Fiction), Asian Culture (Sociology), Data Structure (Science), Database Concepts (Science), Effective C++ (Science), Fluent Python (Science), Game Theory (Science), Hamlet (Fiction), HarryPotter (Fiction), and Introduction to JAVA (Science). A search bar at the top right allows for searching by book name.

Book Name	Topic	View Copies
1984	Fiction	<button>View Copies</button>
Asian Culture	Sociology	<button>View Copies</button>
Data Structure	Science	<button>View Copies</button>
Database Concepts	Science	<button>View Copies</button>
Effective C++	Science	<button>View Copies</button>
Fluent Python	Science	<button>View Copies</button>
Game Theory	Science	<button>View Copies</button>
Hamlet	Fiction	<button>View Copies</button>
HarryPotter	Fiction	<button>View Copies</button>
Introduction to JAVA	Science	<button>View Copies</button>

## My Borrow List:

The screenshot shows the 'My Borrow List' page of the Library Management System. The left sidebar has links for 'My Borrow List' (selected), 'Book List', 'My Invoices', and 'Rooms'. The main content area is titled 'My Borrow List' and shows a table of borrow records with 10 records per page. The columns are 'Copy ID', 'Status', 'Borrow Date', 'Expected Return Date', 'Actual Return Date', and 'Action'. The records listed are: Copy ID 11, Status Return, Borrow Date Dec. 2, 2022, 7:38 p.m., Expected Return Date March 2, 2023, 7:38 p.m., Actual Return Date Dec. 2, 2022, 8:04 p.m., Action Returned; Copy ID 11, Status Return, Borrow Date Dec. 2, 2022, 8:41 p.m., Expected Return Date March 2, 2023, 8:41 p.m., Actual Return Date Dec. 2, 2022, 8:41 p.m., Action Returned; Copy ID 11, Status Return, Borrow Date Dec. 12, 2022, 1:57 a.m., Expected Return Date March 12, 2023, 1:57 a.m., Actual Return Date Dec. 12, 2022, 1:58 a.m., Action Returned; Copy ID 11, Status Return, Borrow Date Dec. 13, 2022, 11:05 p.m., Expected Return Date March 13, 2023, 11:05 p.m., Actual Return Date Dec. 14, 2022, 5:16 p.m., Action Returned; Copy ID 11, Status Return, Borrow Date Dec. 14, 2022, 5:19 p.m., Expected Return Date March 14, 2023, 5:19 p.m., Actual Return Date Dec. 18, 2022, 1:12 a.m., Action Returned; Copy ID 12, Status Return, Borrow Date Dec. 2, 2022, 8:10 p.m., Expected Return Date March 2, 2023, 8:10 p.m., Actual Return Date Dec. 2, 2022, 8:10 p.m., Action Returned; Copy ID 12, Status Return, Borrow Date Dec. 2, 2022, 8:16 p.m., Expected Return Date March 2, 2023, 8:16 p.m., Actual Return Date Dec. 2, 2022, 8:21 p.m., Action Returned; Copy ID 12, Status Return, Borrow Date Dec. 2, 2022, 8:41 p.m., Expected Return Date March 2, 2023, 8:41 p.m., Actual Return Date Dec. 2, 2022, 8:41 p.m., Action Returned; Copy ID 13, Status Return, Borrow Date Jan. 31, 2023, 3:09 a.m., Expected Return Date May 1, 2023, 3:09 a.m., Actual Return Date Jan. 31, 2023, 3:09 a.m., Action Returned; Copy ID 13, Status Borrowed, Borrow Date Dec. 14, 2022, 5:15 p.m., Expected Return Date March 14, 2023, 5:15 p.m., Actual Return Date None, Action Return. A search bar at the top right allows for searching by copy ID.

Copy ID	Status	Borrow Date	Expected Return Date	Actual Return Date	Action
11	Return	Dec. 2, 2022, 7:38 p.m.	March 2, 2023, 7:38 p.m.	Dec. 2, 2022, 8:04 p.m.	<button>Returned</button>
11	Return	Dec. 2, 2022, 8:41 p.m.	March 2, 2023, 8:41 p.m.	Dec. 2, 2022, 8:41 p.m.	<button>Returned</button>
11	Return	Dec. 12, 2022, 1:57 a.m.	March 12, 2023, 1:57 a.m.	Dec. 12, 2022, 1:58 a.m.	<button>Returned</button>
11	Return	Dec. 13, 2022, 11:05 p.m.	March 13, 2023, 11:05 p.m.	Dec. 14, 2022, 5:16 p.m.	<button>Returned</button>
11	Return	Dec. 14, 2022, 5:19 p.m.	March 14, 2023, 5:19 p.m.	Dec. 18, 2022, 1:12 a.m.	<button>Returned</button>
12	Return	Dec. 2, 2022, 8:10 p.m.	March 2, 2023, 8:10 p.m.	Dec. 2, 2022, 8:10 p.m.	<button>Returned</button>
12	Return	Dec. 2, 2022, 8:16 p.m.	March 2, 2023, 8:16 p.m.	Dec. 2, 2022, 8:21 p.m.	<button>Returned</button>
12	Return	Dec. 2, 2022, 8:41 p.m.	March 2, 2023, 8:41 p.m.	Dec. 2, 2022, 8:41 p.m.	<button>Returned</button>
13	Return	Jan. 31, 2023, 3:09 a.m.	May 1, 2023, 3:09 a.m.	Jan. 31, 2023, 3:09 a.m.	<button>Returned</button>
13	Borrowed	Dec. 14, 2022, 5:15 p.m.	March 14, 2023, 5:15 p.m.	None	<button>Return</button>

## BookList:

The screenshot shows a web-based library management system. At the top, there's a header bar with links for 'localhost:8000/my\_library/books/1/' and 'Hello, haiyue. log out books my records my invoices'. On the right, there's a 'My Account' dropdown. Below the header is a dark navigation bar with the text 'Library Management System'. To the left, a sidebar contains links for 'My Borrow List', 'Book List' (which is selected), 'My Invoices', and 'Rooms'. The main content area is titled 'Book List' and includes a dropdown for 'records per page' set to 10, and a search bar. A table lists three book entries: Copy ID 11 (Availability 1) with a 'borrow' button, Copy ID 12 (Availability 0) with an 'Empty' button, and Copy ID 13 (Availability 0) with an 'Empty' button. Below the table, it says 'Showing 1 to 3 of 3 entries'. At the bottom right are buttons for 'Previous', '1' (highlighted in blue), and 'Next'.

## Select date for reservation:

The screenshot shows a 'Room List' page. At the top, there's a header bar with a link for 'localhost:8000/my\_library/rooms/1/' and 'Hello, haiyue. log out books my records my invoices'. On the right, there's a 'My Account' dropdown. Below the header is a dark navigation bar with the text 'Room List'. To the left, a sidebar contains a link for 'Room List'. The main content area is titled 'Room List' and includes a dropdown for 'records per page' set to 10, and a search bar. A table lists ten room entries: Room ID 101 (Capacity 8), 102 (Capacity 12), 103 (Capacity 8), 104 (Capacity 4), 105 (Capacity 4), 201 (Capacity 4), 202 (Capacity 4), 203 (Capacity 4), 204 (Capacity 8), and 205 (Capacity 4). Each entry has fields for 'Date' and 'Period' and a 'Reserve' button. A modal dialog box is open over the table, listing four time periods: A (08:00 - 10:00), B (11:00 - 13:00), C (14:00 - 16:00), and D (17:00 - 19:00). At the bottom, it says 'Showing 1 to 10 of 10 entries' and has buttons for 'Previous', '1' (highlighted in blue), and 'Next'.

## My Invoice:

Hello, haiyue. [log out](#) - [books](#) - [my\\_records](#) - [my\\_invoices](#)

### Library Management System

My Account ▾

[My Borrow List](#)

[Book List](#)

[My Invoices](#)

[Rooms](#)

#### My Invoices

10 records per page

Invoice ID	Date	Balance	Action
11	Jan. 31, 2023, 2:14 a.m.	0.00	All paid
12	Jan. 31, 2023, 3:09 a.m.	0.00	All paid
13	Jan. 31, 2023, 3:14 a.m.	0.00	All paid
14	Dec. 12, 2022, 1:58 a.m.	0.00	All paid
15	Dec. 12, 2022, 1:58 a.m.	0.00	All paid
16	Dec. 14, 2022, 5:16 p.m.	0.00	All paid
17	Dec. 18, 2022, 1:12 a.m.	0.80	<a href="#">Pay for this invoice</a>

Showing 1 to 7 of 7 entries ← Previous 1 Next →

#### My Payments

Invoice ID	Payment ID	Amount	Method
16	7	0.10	Paypal
16	8	0.10	Paypal

## Search:

#### Book List

10 records per page

Book Name	Topic	View Copies
1984	Fiction	<a href="#">View Copies</a>
Hamlet	Fiction	<a href="#">View Copies</a>
HarryPotter	Fiction	<a href="#">View Copies</a>

Showing 1 to 3 of 3 entries (filtered from 11 total entries) ← Previous 1 Next →

## Rooms:

The screenshot shows a web-based library management system. The URL in the address bar is `localhost:8000/my_library/rooms/`. The top navigation bar includes links for "My Account" and other user options. On the left, a sidebar menu lists "My Borrow List", "Book List", "My Invoices", and "Rooms". The main content area is titled "Room List" and displays a table of room information. The table has columns for "Room ID", "Capacity", and "Action". The "Action" column contains buttons for "Reserve", "--Date--", and "--Period--". The room data is as follows:

Room ID	Capacity	Action
101	8	--Date-- --Period-- Reserve
102	12	--Date-- --Period-- Reserve
103	8	--Date-- --Period-- Reserve
104	4	--Date-- --Period-- Reserve
105	4	--Date-- --Period-- Reserve
201	4	--Date-- --Period-- Reserve
202	4	--Date-- --Period-- Reserve
203	4	--Date-- --Period-- Reserve
204	8	--Date-- --Period-- Reserve
205	4	--Date-- --Period-- Reserve

At the bottom, it says "Showing 1 to 10 of 10 entries" and has navigation buttons for "Previous", "1", and "Next".

## payment:

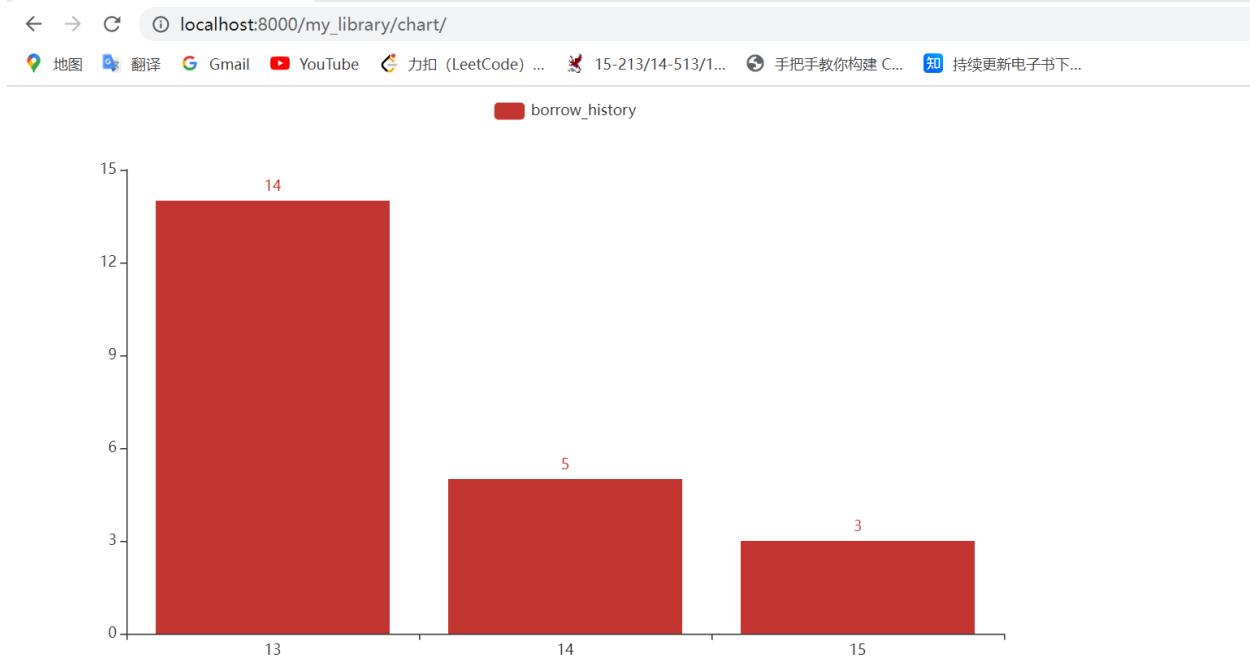
Hello, haiyue. log out- books- my\_records- my\_invoices

The screenshot shows a payment interface within the library management system. The title bar says "Library Management System". The main form area contains fields for "Payment Method" (set to "Paypal"), "Amount" (set to "0.6"), and a "Pay" button.

Payment Method:

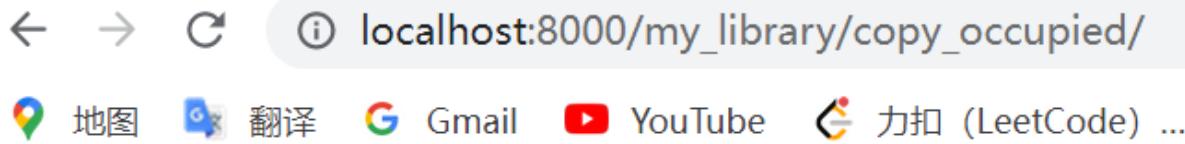
Amount:

## Data visualization: (total number of history of borrowings of each customer)



## Transaction:

We use `django.db.transaction` to make book borrow function, room reservation function and invoice payment function to be transactions. When multiple users are accessing the reservation or borrowing the same copy, we use transactions to prevent data inconsistency.



This copy has just been borrowed! - [back](#)

## Security and Reliability Features:

We choose Django as the framework to prevent SQL injection. Django prevents SQL injection attacks by using parameterized queries. Django's object-relational mapper (ORM) uses parameterized queries to prevent SQL injection when using it to interact with the database. When using the ORM to create, retrieve, update, or delete records in the database, it generates the necessary SQL statements and passes the values specified as parameters to the database driver.

It also has built-in features to prevent CSRF attacks. Django generates a unique token and stores it in the user's session. When the user submits the form, Django checks that the token in the form matches the token in the user's session to determine accepting or rejecting requests.

For reliability features, we implemented `@transaction.atomic` of Django to ensure data consistency when multiple users are accessing as above.

# Business Analysis with SQL Queries:

## Q1 Table Joins with at Least 3 Tables in Join

```
1 •  use errorfree;
2 •  SELECT a.author_id, a.firstname, a.lastname, b.book_id, b.bookname, e.event_id, e.eventname
3   FROM author a
4   INNER JOIN au_bo ab ON a.author_id = ab.author_id
5   INNER JOIN book b ON ab.book_id = b.book_id
6   INNER JOIN au_se ases ON a.author_id = ases.author_id
7   INNER JOIN Lib_Event e ON ases.Event_ID = e.Event_ID
8
9
```

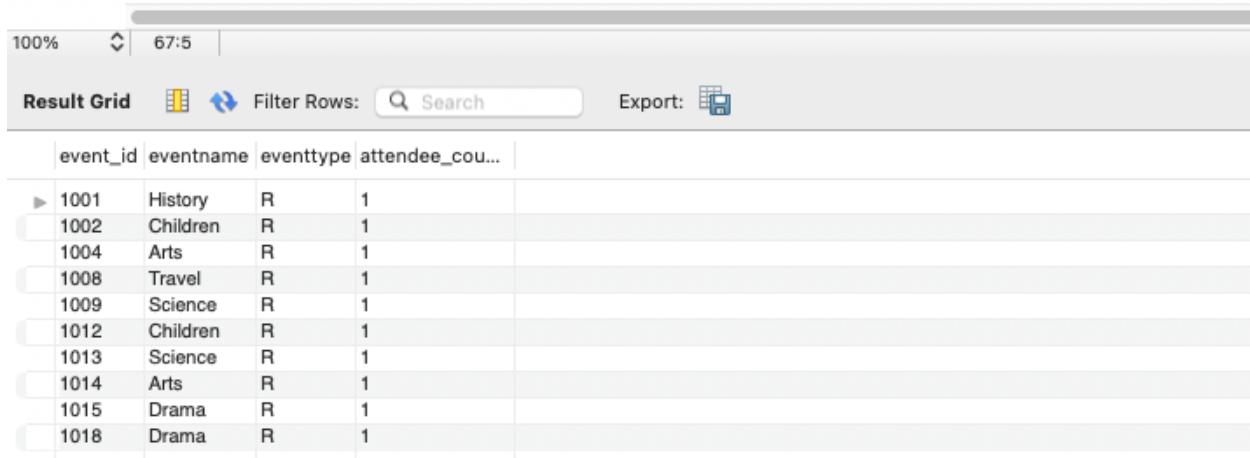
The screenshot shows a database query results grid. At the top, there are navigation controls: a progress bar (00% to 29:5), a refresh icon, a search bar with placeholder 'Search', and an export icon. Below the controls is a table header with columns: author\_id, firstname, lastname, book\_id, bookname, event\_id, and eventname. The data rows are as follows:

	author_id	firstname	lastname	book_id	bookname	event_id	eventname
▶	37890	Nanette	Cambrault	41108	R and J	1013	Science
	34680	Peter	Hall	41107	The Prince	1012	Children
	33099	Kimberely	Grant	41103	The Visual Arts	1004	Arts
	32440	Karen	Partners	41109	Janpan	1014	Arts
	33021	Jonathon	Taylor	41101	The Guns of August	1001	History
	33991	Martha	Sullivan	41104	Treasure Island	1008	Travel
	33213	Amit	Banda	41105	The Merchant	1009	Science
	33056	Jack	Livingston	41102	Heart of Darkness	1002	Children

This query retrieves information about the authors, books, and events that are related to each other through the au\_bo, au\_se, and Lib\_Event tables. The query uses inner joins to combine the tables and returns the author\_id, firstname, lastname, book\_id, bookname, event\_id, and eventname for each author-book-event combination.

## Q2 Multi-row Subquery

```
1 •  use errorfree;
2 •  SELECT e.event_id, e.eventname, e.eventtype,
3      (SELECT COUNT(*) FROM au_se WHERE event_id = e.event_id) AS attendee_count
4  FROM lib_event e
5 WHERE (SELECT COUNT(*) FROM au_se WHERE event_id = e.event_id) > 0
```



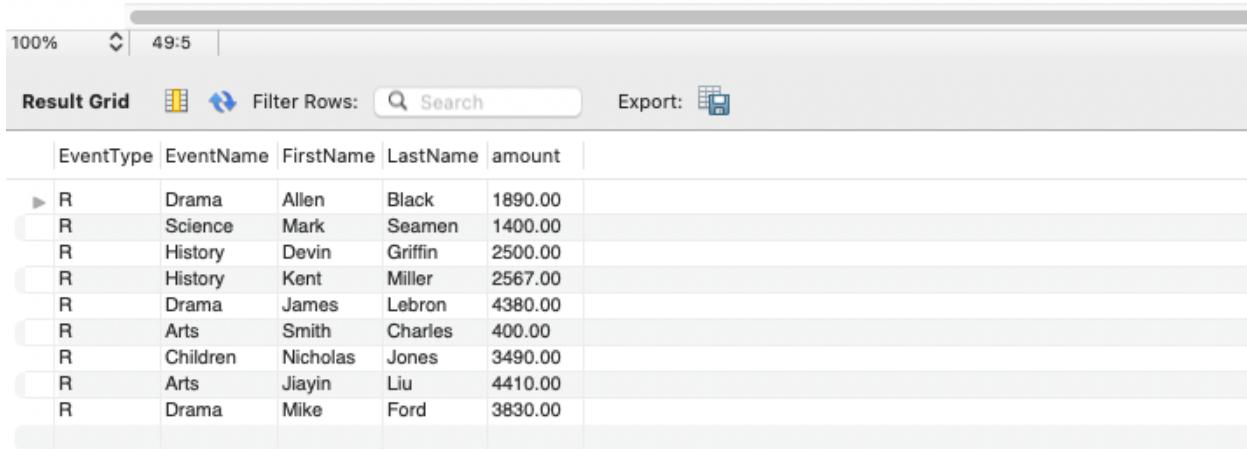
The screenshot shows a MySQL query results grid. At the top, there are zoom controls (100%), a refresh icon, a timestamp (67:5), and navigation buttons. Below that is a toolbar with 'Result Grid' (selected), a refresh icon, 'Filter Rows' with a search bar, and 'Export' options. The main area displays a table with the following data:

	event_id	eventname	eventtype	attendee_cou...
▶	1001	History	R	1
◀	1002	Children	R	1
▶	1004	Arts	R	1
▶	1008	Travel	R	1
▶	1009	Science	R	1
▶	1012	Children	R	1
▶	1013	Science	R	1
▶	1014	Arts	R	1
▶	1015	Drama	R	1
▶	1018	Drama	R	1

This query retrieves information of event\_id, eventname, eventtype and attendee\_count. It selects these columns and uses another multi-row subquery in the `where` clause to filter out lib events that have more than 0 attendees.

## Q3 Correlated Subquery

```
1  SELECT e.EventType, e.EventName, s.FirstName, s.LastName, sp.amount
2  FROM lib_event e
3  INNER JOIN SP_SE sp ON e.Event_ID = sp.Event_ID
4  INNER JOIN sponsor s ON sp.Sponsor_ID = s.Sponsor_ID
5  WHERE sp.Amount = (SELECT MAX(Amount) FROM SP_SE WHERE Event_ID = e.Event_ID)
6
```



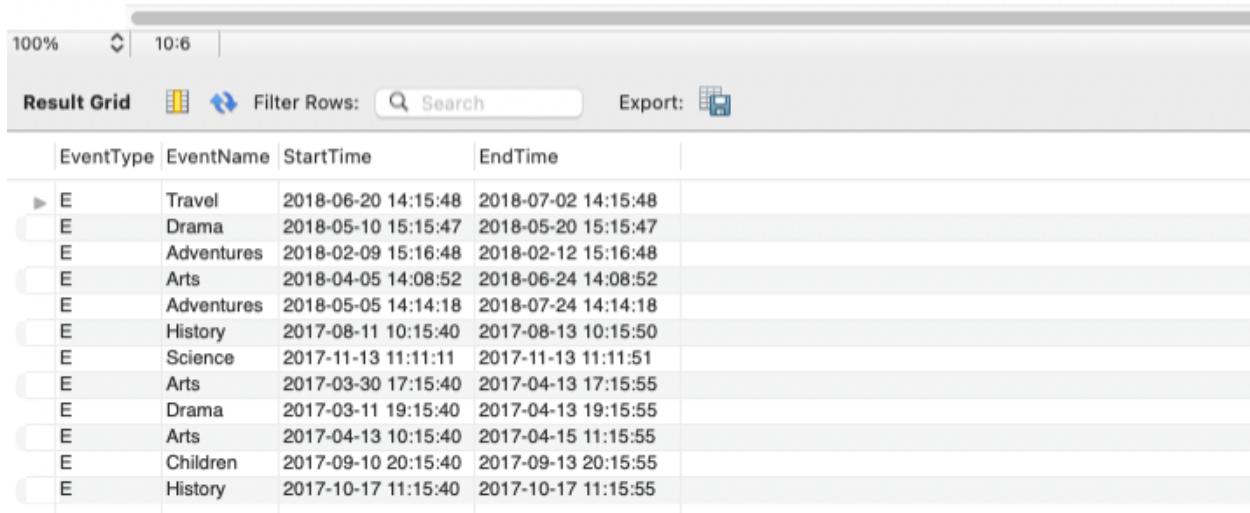
The screenshot shows a database query results grid. At the top, there are navigation controls: a zoom icon (100%), a refresh icon, a search bar with placeholder 'Search', and an export icon. The grid has columns labeled 'EventType', 'EventName', 'FirstName', 'LastName', and 'amount'. The data rows are as follows:

	EventType	EventName	FirstName	LastName	amount
▶	R	Drama	Allen	Black	1890.00
◀	R	Science	Mark	Seamen	1400.00
▶	R	History	Devin	Griffin	2500.00
◀	R	History	Kent	Miller	2567.00
▶	R	Drama	James	Lebron	4380.00
◀	R	Arts	Smith	Charles	400.00
▶	R	Children	Nicholas	Jones	3490.00
◀	R	Arts	Jiayin	Liu	4410.00
▶	R	Drama	Mike	Ford	3830.00

The business information that this query intends to retrieve is the list of events, their sponsors, and the highest sponsorship amount for each event. The query achieves this by using a correlated subquery in the WHERE clause to filter out sponsorship rows that do not have the highest amount for their respective event. The resulting table includes the event type and name from the Event table, the sponsor's first and last name from the Sponsor table, and the sponsorship amount from the SP\_SE table.

## Q4 SET Operator Query

```
1 •  SELECT EventType, EventName, StartTime, EndTime
2   FROM Lib_Event
3   WHERE EventType = 'E'
4 UNION
5   SELECT EventType, EventName, StartTime, EndTime
6   FROM Lib_Event
7   WHERE EventType = 'R' AND StartTime >= '2022-01-01'
8
```



The screenshot shows a database query results grid. At the top, there are navigation controls: a zoom icon (100%), a refresh icon, a search bar with placeholder 'Search', and an export icon. Below the controls is a table header with four columns: 'EventType', 'EventName', 'StartTime', and 'EndTime'. The data rows are as follows:

EventType	EventName	StartTime	EndTime
E	Travel	2018-06-20 14:15:48	2018-07-02 14:15:48
E	Drama	2018-05-10 15:15:47	2018-05-20 15:15:47
E	Adventures	2018-02-09 15:16:48	2018-02-12 15:16:48
E	Arts	2018-04-05 14:08:52	2018-06-24 14:08:52
E	Adventures	2018-05-05 14:14:18	2018-07-24 14:14:18
E	History	2017-08-11 10:15:40	2017-08-13 10:15:50
E	Science	2017-11-13 11:11:11	2017-11-13 11:11:51
E	Arts	2017-03-30 17:15:40	2017-04-13 17:15:55
E	Drama	2017-03-11 19:15:40	2017-04-13 19:15:55
E	Arts	2017-04-13 10:15:40	2017-04-15 11:15:55
E	Children	2017-09-10 20:15:40	2017-09-13 20:15:55
E	History	2017-10-17 11:15:40	2017-10-17 11:15:55

This query is designed to retrieve information about events that are either educational (EventType = 'E') or recreational (EventType = 'R') and start on or after the year 2022 (StartTime >= '2022-01-01'). The UNION operator is used to combine the results of two separate queries, one for each of the two event types. The WHERE clause filters the rows in the Event table to only include those that meet the specified criteria.

## Q5 Query with In-Line View or WITH Clause

```
1 • ① WITH sponsors AS (
2     SELECT Event_ID, COUNT(Sponsor_ID) AS NumSponsors
3     FROM Sp_SE
4     GROUP BY Event_ID
5 )
6     SELECT e.EventType, e.EventName, e.StartTime, s.NumSponsors
7     FROM Lib_Event e
8     INNER JOIN sponsors s ON e.Event_ID = s.Event_ID
9     WHERE s.NumSponsors > 1
10
```

The screenshot shows a database query results grid. At the top, there are navigation controls: a zoom level of 100%, a time indicator of 10:7, and a toolbar with 'Result Grid' selected, followed by 'Filter Rows:', a search bar, and an export icon. The results grid has four columns: EventType, EventName, StartTime, and NumSponsors. There is one visible row of data:

EventType	EventName	StartTime	NumSponsors
R	History	2017-03-13 11:15:40	2

The business information that this query intends to retrieve is the list of events that have more than one sponsor, along with the number of sponsors for each event. The query achieves this by using a WITH clause to define an inline view, sponsors, which counts the number of sponsors for each event. The inline view is then joined to the Event table on the Event\_ID column, and the resulting table is filtered to only include rows where the number of sponsors is greater than 1.

## Q6 TOP-N Query

```
1 •  SELECT e.EventType, e.EventName, s.FirstName, s.LastName, sp.Amount  
2   FROM Lib_Event e  
3   INNER JOIN SP_SE sp ON e.Event_ID = sp.Event_ID  
4   INNER JOIN Sponsor s ON sp.Sponsor_ID = s.Sponsor_ID  
5   ORDER BY sp.Amount DESC  
6   limit 3  
7  |
```

The screenshot shows a database query results grid. At the top, there are navigation controls: a zoom icon (100%), a refresh icon, a search bar with the text 'Search', and export and fetch row options. The results grid has five columns: EventType, EventName, FirstName, LastName, and Amount. The data is as follows:

	EventType	EventName	FirstName	LastName	Amount
▶	R	Arts	Jiayin	Liu	4410.00
▶	R	Drama	James	Lebron	4380.00
▶	R	Drama	Mike	Ford	3830.00

The business information that this query intends to retrieve is the top 3 events in terms of the highest sponsorship amount, along with their sponsors. The query achieves this by using the Limit clause to select the top 3 rows based on the sponsorship amount in descending order. The resulting table includes the event type and name from the Event table, the sponsor's first and last name from the Sponsor table, and the sponsorship amount from the Sponsorship table. This information can be useful for various business purposes, such as identifying which events are most popular with sponsors or allocating resources based on the expected return on investment.

## Lesson/Reflection:

We are glad for the progress we made and the skills we developed. Working with databases was a new experience for us and we found it to be both challenging and rewarding. One of the challenges of the project was learning how to properly structure the database and create proper relationships between tables. Another challenge was how to manipulate and extract data from the database. It took us some time and trial to build the website, but we were eventually able to accomplish this project. Overall, we think this was a great and valuable experience. We are

grateful to take this class. We learned that SQL is a powerful tool for manipulating and extracting data, and it's essential to be patient and persistent when learning new concepts.

## Extra Features:

Following features are applied for extra credit:

- a. Data Visualisation
- b. Security Check on Password Reset
- c. Book search engine for users to find desired books