Project1

Course Section: Principles of Database Systems, 6083 Section B

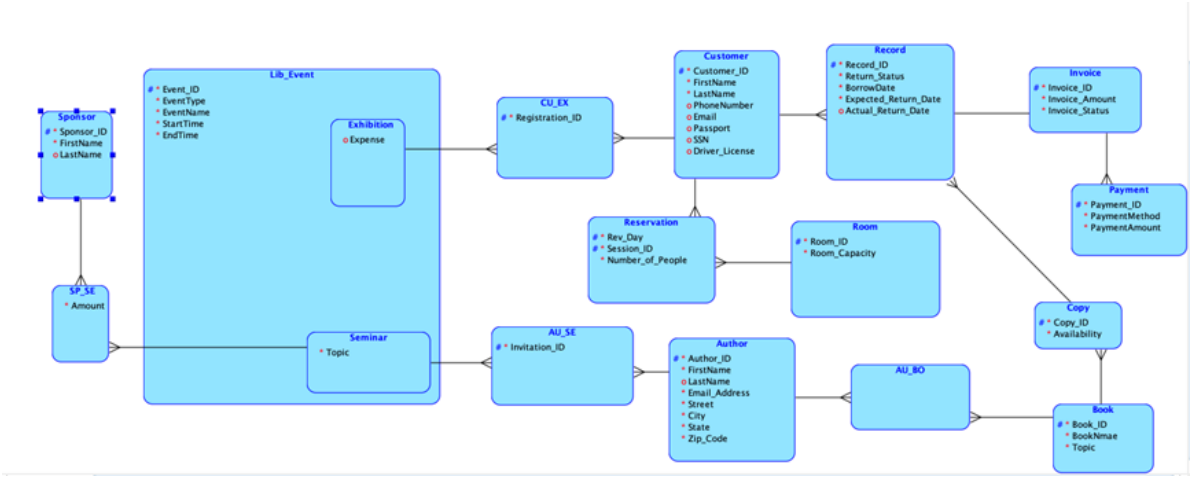
Date :2022/11/06

Siyuan Chen sc9513

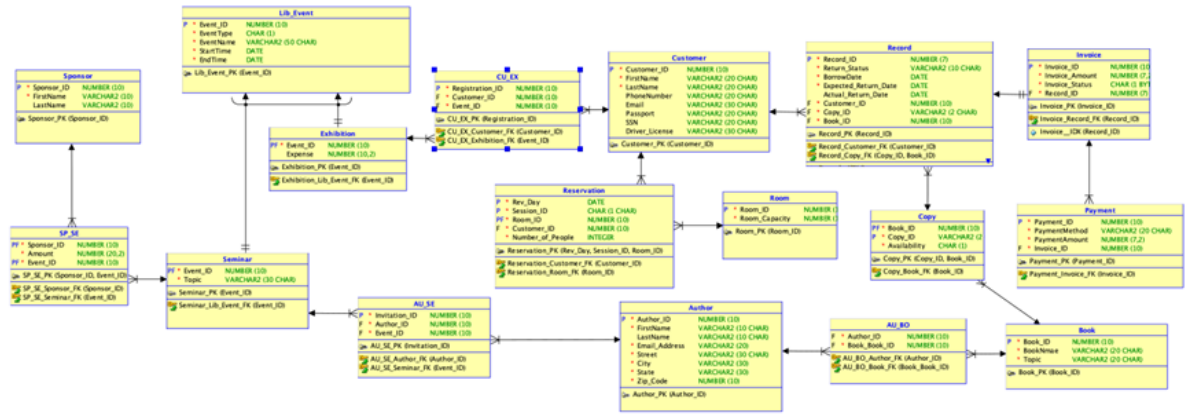
Haiyue Zhao hz3261

Haoru Wen hw3254

1. Logical Model



2. Relational Model



3. Assumptions

\1) Sponsor name can be separated into FirstName and LastName.

\2) Exhibition and seminar are the subtypes of Lib_event. Eventtype is used as the discriminator, and "E" is for exhibition, "R" is for seminar.

\3) Revday, seesion_ID, RoomID can be the primary key of reservation to resolve schedule conflict. 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 return, the actual return date and the status will be updated, and the corresponding invoice will be inserted.

\6) Return status have 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 customer and exhibition is many to many. An intersection table is added

\10) the relationship between sponsor and seminar is many to many. An intersection table is added

\11) the relationship between author and book is many to many. An intersection table is added

\12) One record should have one invoice. So, record to invoice is one to one relationship.

\13) The number of people in reservation should not exceed the room capacity.

\14) The expected return date and the actual return date cannot be ealier than borrow date.

\15) The status of Invoice is boolean, and it is to show if the invoice has been paid off.

4. DDL

```
use errorfree;
-- 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 'UNQIUE 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
    'UNQIUE 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 'UNQIUE 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
    'UNQIUE 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




-- 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 ***                        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
```

5. Check Constraints

```
ALTER table lib_event
add constraint event_ck3 check(endtime-starttime > 0);

ALTER table SP_SE
add constraint SP_SE_ck check(amount >= 0);

ALTER table payment
add constraint payment_ck check(paymentamount >= 0);

ALTER table Invoice
add constraint invoice_ck check(Invoice_Amount >= 0);

ALTER table Record
add constraint record_ck check(expected_return_date - borrowdate > 0);

ALTER table Record
add constraint record_ck2 check(actual_return_date - borrowdate >= 0);
```

We use enum instead of some check constraints.

```
ALTER table lib_event
modify  column eventtype enum('E','R');

ALTER table Record
modify  column Return_Status enum('Borrowed','Return','Late');

alter table copy
modify column availability tinyint(1);
```

6. Triggers

(1)Insertion trigger for borrowing books

When borrowing books, this trigger will check if the insertion record meets the assumption and if the book is available. Also the trigger will change the availability of the copy.

```
delimiter //
create trigger record_insert_trigger before
    insert on record
    for each row
Begin
    declare avail tinyint default 0;
    SELECT
        a.availability
    INTO avail
    FROM
        copy a
    WHERE
        a.copy_id = new.copy_id and a.book_id = new.book_id;
    IF ( new.Actual_Return_Date is not NULL) THEN
        SIGNAL SQLSTATE '10123' SET MESSAGE_TEXT = 'When you borrow the book,
the actual_return_date should be null';
    END IF;
    IF ( new.Return_Status = 'Return' OR new.Return_Status = 'Late') THEN
        SIGNAL SQLSTATE '10124' SET MESSAGE_TEXT = 'When you borrow the book,
the return_status should be borrowed';
    END IF;
    IF (avail = 0) then
    SIGNAL SQLSTATE '10127' SET MESSAGE_TEXT ='copy is not available';
    end if;
    update copy set availability = 0 where book_id = new.book_id and copy_id =
new.copy_id;
End //
```

(2) Trigger for calculating and generating invoice record

When returning books, this trigger will calculate the amount, and generate the invoice record.

```
delimiter //

CREATE TRIGGER record_update_trigger AFTER
UPDATE ON record
FOR EACH ROW
BEGIN
    DECLARE startd DATE ;
    DECLARE expectd DATE ;
    DECLARE actuald DATE ;
    declare num_invoice int default 0;
    DECLARE sum DOUBLE DEFAULT 0.0 ;
IF (new.actual_return_date is NULL) then
SIGNAL SQLSTATE '10121' SET MESSAGE_TEXT = 'book have not returned';
end if;
```

```
IF (new.expected_return_date <> old.expected_return_date OR new.borrowdate <>
old.borrowdate) then
SIGNAL SQLSTATE '10122' SET MESSAGE_TEXT = 'You should change the borrow date
and expected date after the book is borrowed';
end if;

set startd = new.borrowdate;
set expectd  = new.expected_return_date;
set actuald = new.actual_return_date;
select count(*) into num_invoice from invoice ;


IF (DATE(expectd)< actuald) THEN
IF (new.return_status <> 'Late') then
SIGNAL SQLSTATE '10125' SET MESSAGE_TEXT = 'Return Status should be late';
end if;
set sum =  timestampdiff(day, startd,
expectd)*0.2+timestampdiff(day,expectd,actuald)*0.4;
ELSEIF (DATE(expectd)> actuald) THEN
IF (new.return_status <> 'Return') then
SIGNAL SQLSTATE '10126' SET MESSAGE_TEXT = 'Return Status should be Return';
end if;
set sum =  timestampdiff(day, startd, actuald)*0.2;
END IF;
INSERT INTO `invoice` (`invoice_id`, `invoice_amount`,
`invoice_status`,`record_id`) VALUES ( num_invoice+1, sum, '0',new.record_id);
update copy set availability = 1 where book_id = new.book_id and copy_id =
new.copy_id;
END //
```

(3) Trigger to check ID of customer

Passport, ssn, and the driver_license can not be NULL at the same time.

```
delimiter //
CREATE TRIGGER chk_idnull BEFORE
    INSERT ON customer
    FOR EACH ROW
BEGIN

    IF ( new.passport IS NULL  AND new.ssn IS NULL AND new.driver_license IS
NULL) THEN
  SIGNAL SQLSTATE '20121' SET MESSAGE_TEXT = 'id required';
    END IF;


END //
```

(4) Trigger for exhibition and seminar

```
delimiter //

CREATE TRIGGER arc_fkarc_3_exhibition BEFORE
    INSERT  ON exhibition
    FOR EACH ROW
BEGIN
    declare d char(8);

    SELECT
        a.eventtype
    INTO d
    FROM
        lib_event a
    WHERE
        a.event_id = new.event_id;

    IF ( d IS NULL OR d <> 'E' ) THEN
        SIGNAL SQLSTATE '20223' SET MESSAGE_TEXT = 'FK Seminar_Event_FK in Table
Seminar violates Arc constraint on Table Event - discriminator column EventType
doesn''t have value ''E''';
    END IF;


END //
```

```
delimiter //

CREATE TRIGGER arc_fkarc_3_seminar BEFORE
    INSERT  ON seminar
    FOR EACH ROW
BEGIN
    declare d char(8);

    SELECT
        a.eventtype
    INTO d
    FROM
        lib_event a
    WHERE
        a.event_id = new.event_id;

    IF ( d IS NULL OR d <> 'R' ) THEN
        SIGNAL SQLSTATE '20224' SET MESSAGE_TEXT = 'FK Seminar_Event_FK in Table
Seminar violates Arc constraint on Table Event - discriminator column EventType
doesn''t have value ''R''';
    END IF;


END //
```

(5) Trigger for room reservation

```
delimiter //

CREATE TRIGGER chk_person_number BEFORE
    INSERT  ON reservation
    FOR EACH ROW
BEGIN
    declare d int default 0;
    SELECT
        a.room_capacity
    INTO d
    FROM
        room a
    WHERE
        a.room_id = new.room_id;

    IF ( d IS NULL OR d < new.Number_of_People ) THEN
        SIGNAL SQLSTATE '20123' SET MESSAGE_TEXT = 'Too many people';
    END IF;


END //
```

6. DML

```
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10001, 'Steven',
'King');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10023, 'Mark',
'Seamen');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10709, 'Smith',
'Charles');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10011, 'Allen',
'Black');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10231, 'Kent',
'Miller');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (13401, 'Mike',
'Ford');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10701, 'James',
'Lebron');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10091, 'Devin',
'Griffin');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10801, 'Nicholas',
'Jones');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (11001, 'Jiayin',
'Liu');
Insert into Sponsor (Sponsor_ID,FirstName,LastName) values (10441, 'Chris',
null);
commit;
```

```
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1001,'R','History', STR_TO_DATE('03-13-2017 11,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('04-13-2017 11,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1002,'R','Children', STR_TO_DATE('03-22-2018 09,30,08', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('03-27-2018 09,30,08', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1003,'E','Travel', STR_TO_DATE('06-20-2018 14,15,48', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('07-02-2018 14,15,48', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1004,'R','Arts', STR_TO_DATE('06-19-2018 16,15,48', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('07-19-2018 16,15,48', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1005,'E','Drama', STR_TO_DATE('05-10-2018 15,15,47', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('05-20-2018 15,15,47', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1006,'E','Adventures', STR_TO_DATE('02-09-2018 15,16,48', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('02-12-2018 15,16,48', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1007,'E','Arts', STR_TO_DATE('04-05-2018 14,08,52', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('06-24-2018 14,08,52', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1008,'R','Travel', STR_TO_DATE('12-16-2017 14,10,57', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('03-26-2018 14,10,57', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1009,'R','Science', STR_TO_DATE('03-16-2018 14,12,33', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('05-25-2018 14,12,33', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1010,'E','Adventures', STR_TO_DATE('05-05-2018 14,14,18', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('07-24-2018 14,14,18', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1011,'R','Arts', STR_TO_DATE('06-09-2018 14,15,48', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('09-02-2018 14,15,48', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1012,'R','Children', STR_TO_DATE('03-10-2017 11,20,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('04-10-2017 11,20,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1013,'R','Science', STR_TO_DATE('02-11-2017 08,15,00', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('02-17-2017 08,15,00', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1014,'R','Arts', STR_TO_DATE('05-22-2017 15,11,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('05-29-2017 15,11,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1015,'R','Drama', STR_TO_DATE('06-13-2017 14,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('06-15-2017 11,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1016,'R','History', STR_TO_DATE('07-07-2017 11,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('07-08-2017 11,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1017,'R','Drama', STR_TO_DATE('08-09-2017 11,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('08-09-2017 11,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1018,'R','Drama', STR_TO_DATE('08-07-2017 11,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('08-07-2017 11,15,40', '%m-%d-%Y %H,%i,%s'));
```

```
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1019,'E','History', STR_TO_DATE('08-11-2017 10,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('08-13-2017 10,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1020,'E','Science', STR_TO_DATE('11-13-2017 11,11,11', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('11-13-2017 11,11,11', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1021,'E','Arts', STR_TO_DATE('03-30-2017 17,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('04-13-2017 17,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1022,'E','Drama', STR_TO_DATE('03-11-2017 19,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('04-13-2017 19,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1023,'E','Arts', STR_TO_DATE('04-13-2017 10,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('04-15-2017 11,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1024,'E','Children', STR_TO_DATE('09-10-2017 20,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('09-13-2017 20,15,40', '%m-%d-%Y %H,%i,%s'));
Insert into Lib_Event (Event_ID, EventType, EventName, StartTime, EndTime) values
(1025,'E','History', STR_TO_DATE('10-17-2017 11,15,40', '%m-%d-%Y %H,%i,%s'),
STR_TO_DATE('10-17-2017 11,15,40', '%m-%d-%Y %H,%i,%s'));


Insert into Exhibition (event_id,Expense) values(1003,null);
Insert into Exhibition (event_id,Expense) values(1005,3100);
Insert into Exhibition (event_id,Expense) values(1006,3100);
Insert into Exhibition (event_id,Expense) values(1007,3100);
Insert into Exhibition (event_id,Expense) values(1010,3100);
Insert into Exhibition (event_id,Expense) values(1019,3100);
Insert into Exhibition (event_id,Expense) values(1020,3100);
Insert into Exhibition (event_id,Expense) values(1021,3100);
Insert into Exhibition (event_id,Expense) values(1022,3100);
Insert into Exhibition (event_id,Expense) values(1023,3100);
commit;
Insert into seminar (event_id,topic) values (1001,'Macb');
Insert into seminar (event_id,topic) values (1002,'Shre');
Insert into Seminar (event_id,topic) values (1004,'Poem');
Insert into Seminar (event_id,topic) values (1008,'Chnce');
Insert into Seminar (event_id,topic) values (1009,'Poem');
Insert into Seminar (event_id,topic) values (1012,'iefm');
Insert into Seminar (event_id,topic) values (1013,'efefem');
Insert into Seminar (event_id,topic) values (1014,'fefm');
Insert into Seminar (event_id,topic) values (1015,'kkfem');
Insert into Seminar (event_id,topic) values (1016,'fsefm');
Insert into Seminar (event_id,topic) values (1017,'fesfm');
Insert into Seminar (event_id,topic) values (1018,'efsem');

Insert into SP_SE (sponsor_id,Amount,event_id) values (10001,2400,1001);
Insert into SP_SE (sponsor_id,Amount,event_id) values (10023,1400,1013);
Insert into SP_SE (sponsor_id,Amount,event_id) values (10709,400,1014);
Insert into SP_SE (sponsor_id,Amount,event_id) values (10011,1890,1015);
Insert into SP_SE (sponsor_id,Amount,event_id) values (10231,2567,1016);
Insert into SP_SE (sponsor_id,Amount,event_id) values (13401,3830,1017);
Insert into SP_SE (sponsor_id,Amount,event_id) values (10701,4380,1018);
Insert into SP_SE (sponsor_id,Amount,event_id) values (10091,2500,1001);
```

```
Insert into SP_SE (sponsor_id,Amount,event_id) values (10801,3490,1002);
Insert into SP_SE (sponsor_id,Amount,event_id) values (11001,4410,1004);
commit;
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (33001,'Clara','Vishney','clara001@gmail.com','33 Bond
St','Brooklyn','New York',11201);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (33021,'Jonathon','Taylor','taylor11@gmail.com','145
Chestnut St','Santa Cruz','California',95060);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (33056,'Jack','Livingston','jackliv@gmail.com','5th
Avenue','New York','New York',10004);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (33099,'Kimberely','Grant','grant2367@gmail.com','Jay
St','New York','New York',10001);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (33991,'Martha','Sullivan','martha912gmail.com','31 Lee
St','Santa Cruz','California',95062);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (33213,'Amit','Banda','banda2789@nyu.edu','583 Anges
St','Atlasburg','Pennsylvania',15004);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (33088,'Louise','Doran','louisedoran7@gmail.com','99
St','Bairdford','Pennsylvania',15006);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (34680,'Peter','Hall','petetrhhhal@gmail.com','23 Co
St','New York','New York',11003);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values
(37890,'Nanette','Cambrault','nanettecam7@gmail.com','West Wind St','East
Orange','New Jersey',07019);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (32440,'Karen','Partners','karen16@gmail.com','Tom
Avenue','Orange','New Jersey',07051);
Insert into Author (Author_ID, FirstName, LastName, Email_Address, Street, City,
State, Zip_Code) values (31999,'Lisa','Ozer','lissssa@gmail.com','11th
Avenue','Lima','Pennsylvania',15011);

Insert into AU_SE (Invitation_ID, author_id, event_id) values (7771,
33021,1001);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7981,33056,
1002);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7561,
33099,1004);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7777,33991,1008);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7890,33213,
1009);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7546,
34680,1012);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7345,
37890,1013);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7689,32440,
1014);
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7699, 31999,
1015);
```

```
Insert into AU_SE (Invitation_ID, author_id, event_id) values (7700, 31999,
1018);
commit;
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22220,'Tom','Walker',null,'twalker@gmail.com','C0300496','123-45-
6789','Y7608211');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22221,'Adam','Vanloo','515.127.4564','avanloo@gmail.com','C0300711','234-
77-5690','Y567111');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22222,'Allen','Smith','650.124.5234','asmith@nyu.edu',null,'901-34-
7840','Y890156');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22223,'Sam','Liu','515.124.4567','samliu@gmail.com','C0300506','211-38-
9000','I783129');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22224,'Matthew','Ward','650.124.1214','ward33@gmail.com',null,'332-44-
1234','Y749345');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values
(22323,'Chris','Martin','515.123.8181','chrisss@gmail.com','C0310477','902-99-
3478','I679256');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22280,'Anna','Rakowski',null,'annaRak@gamil.com','C0300444','810-34-
3894','I689256');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22996,'Nichol','Kim','650.507.9811','nicholK@gmail.com','C0322492','678-
33-3241','P890458');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values
(22277,'Amber','Grayson','650.501.4876','Grayson@gmail.com','C0312456','978-40-
3578','Y278009');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22444,'Sam','Jones','650.501.2876','Jones2333@nyu.edu','C0345678','992-
34-7810','I900234');
Insert into Customer
(Customer_ID,FirstName,LastName,PhoneNumber,Email,Passport,SSN,Driver_License)
values (22235,'Jane','Smith','650.501.1876','SmithJane@gmail.com',null,'678-40-
3476','K678009');
commit;
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6344,22220,1003);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6322,22221,1005);
```

```
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6345,22222,1006);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6789,22223,1007);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6666,22224,1010);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6689,22323,1019);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6677,22280,1020);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6547,22996,1021);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6548,22277,1022);
Insert into CU_EX (Registration_ID, customer_id, event_id) values
(6549,22444,1023);

commit;
Insert into Room (Room_ID, Room_Capacity) values (320,30);
Insert into Room (Room_ID, Room_Capacity) values (321,30);
Insert into Room (Room_ID, Room_Capacity) values (323,35);
Insert into Room (Room_ID, Room_Capacity) values (324,30);
Insert into Room (Room_ID, Room_Capacity) values (345,40);
Insert into Room (Room_ID, Room_Capacity) values (326,50);
Insert into Room (Room_ID, Room_Capacity) values (327,30);
Insert into Room (Room_ID, Room_Capacity) values (328,30);
Insert into Room (Room_ID, Room_Capacity) values (379,30);
Insert into Room (Room_ID, Room_Capacity) values (380,35);
Insert into Room (Room_ID, Room_Capacity) values (381,30);


Insert into reservation (Rev_Day, Session_ID, room_id,
Customer_ID,Number_of_People) values (STR_TO_DATE('05-10-2020 08,00,00', '%m-%d-
%Y %H,%i,%s'),'A',320,22220,20);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('05-11-2020 11,00,00',
'%m-%d-%Y %H,%i,%s'),'B',321,22221,9);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('07-09-2020 13,00,00',
'%m-%d-%Y %H,%i,%s'),'A',323,22222,8);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('04-29-2020 16,00,00',
'%m-%d-%Y %H,%i,%s'),'C',324,22223,20);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('04-10-2020 11,00,00',
'%m-%d-%Y %H,%i,%s'),'D',345,22224,9);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('04-12-2020 11,00,00',
'%m-%d-%Y %H,%i,%s'),'A',326,22323,16);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('12-29-2019 08,00,00',
'%m-%d-%Y %H,%i,%s'),'B',327,22280,27);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('05-06-2020 16,00,00',
'%m-%d-%Y %H,%i,%s'),'B',328,22996,23);
```

```
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('05-23-2020 16,00,00',
'%m-%d-%Y %H,%i,%s'),'D',328,22277,12);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('05-05-2020 13,00,00',
'%m-%d-%Y %H,%i,%s'),'D',379,22444,1);
Insert into reservation (Rev_Day, Session_ID,
room_id,Customer_ID,Number_of_People) values (STR_TO_DATE('05-09-2020 13,00,00',
'%m-%d-%Y %H,%i,%s'),'C',380,22235,5);
commit;
Insert into Copy (Book_id,Copy_ID, Availability) values (41100,5,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41100,1,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41100,2,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41100,3,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41100,4,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41101,1,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41101,2,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41101,3,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41102,1,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41102,2,1);
Insert into Copy (Book_id,Copy_ID, Availability) values (41102,3,1);
commit;
Insert into Book (Book_ID, BookName, Topic) values (41100,'Macbeth','Drama');
Insert into Book (Book_ID, BookName, Topic) values (41101,'The Guns of
August','History');
Insert into Book (Book_ID, BookName, Topic) values (41102,'Heart of
Darkness','Adventures');
Insert into Book (Book_ID, BookName, Topic) values (41103,'The Visual
Arts','Arts');
Insert into Book (Book_ID, BookName, Topic) values (41104,'Treasure
Island','Adventures');
Insert into Book (Book_ID, BookName, Topic) values (41105,'The
Merchant','Drama');
Insert into Book (Book_ID, BookName, Topic) values (41106,'Intro to
C++','Science');
Insert into Book (Book_ID, BookName, Topic) values (41107,'The
Prince','Children');
Insert into Book (Book_ID, BookName, Topic) values (41108,'R and J','Drama');
Insert into Book (Book_ID, BookName, Topic) values (41109,'Janpan','Travel');
Insert into Book (Book_ID, BookName, Topic) values (41110,'1776','History');
commit;
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53310,22220,'Borrowed',STR_TO_DATE('04-08-2020 13,00,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('05-20-2020 13,00,00', '%m-%d-%Y
%H,%i,%s'),null,41100,5);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53311,22220,'Borrowed',STR_TO_DATE('04-09-2020 11,00,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('04-29-2020 11,00,00', '%m-%d-%Y
%H,%i,%s'),null,41100,1);
```

```
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53312,22220,'Borrowed',STR_TO_DATE('04-11-2020 15,00,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('05-11-2020 15,00,00', '%m-%d-%Y
%H,%i,%s'),null,41100,2);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53313,22221,'Borrowed',STR_TO_DATE('05-19-2020 14,05,05', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('08-19-2020 14,05,05', '%m-%d-%Y
%H,%i,%s'),null,41100,3);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53314,22221,'Borrowed',STR_TO_DATE('03-22-2020 01,00,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('04-22-2020 01,00,00', '%m-%d-%Y
%H,%i,%s'),null,41100,4);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53315,22221,'Borrowed',STR_TO_DATE('08-12-2021 09,00,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('10-12-2021 11,00,00', '%m-%d-%Y
%H,%i,%s'),null,41101,1);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53316,22221,'Borrowed',STR_TO_DATE('01-23-2020 15,30,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('03-12-2020 15,30,00', '%m-%d-%Y
%H,%i,%s'),null,41101,2);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53317,22221,'Borrowed',STR_TO_DATE('06-02-2020 14,30,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('07-02-2020 14,30,00', '%m-%d-%Y
%H,%i,%s'),null,41101,3);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53318,22221,'Borrowed',STR_TO_DATE('04-12-2020 11,00,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('05-12-2020 11,00,00', '%m-%d-%Y
%H,%i,%s'),null,41102,1);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53319,22222,'Borrowed',STR_TO_DATE('11-12-2020 20,07,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('12-12-2020 20,07,00', '%m-%d-%Y
%H,%i,%s'),null,41102,2);
Insert into Record (Record_ID, customer_id,Return_Status,BorrowDate,
Expected_Return_Date,Actual_Return_Date,Book_id,Copy_ID) values
(53320,22222,'Borrowed',STR_TO_DATE('07-09-2020 11,00,00', '%m-%d-%Y
%H,%i,%s'),STR_TO_DATE('09-13-2020 11,00,00', '%m-%d-%Y
%H,%i,%s'),null,41102,3);
commit;



Insert into au_bo (author_id, book_id) values (33001,41100);
Insert into au_bo (author_id, book_id) values (33021,41101);
Insert into au_bo (author_id, book_id) values (33056,41102);
Insert into au_bo (author_id, book_id) values (33099,41103);
Insert into au_bo (author_id, book_id) values (33991,41104);
```

```
Insert into au_bo (author_id, book_id) values (33213,41105);
Insert into au_bo (author_id, book_id) values (33088,41106);
Insert into au_bo (author_id, book_id) values (34680,41107);
Insert into au_bo (author_id, book_id) values (37890,41108);
Insert into au_bo (author_id, book_id) values (32440,41109);
```

record_id	Return_Status	borrowdate	expected_return_d...	actual_return_d...	customer_id	copy_id	book_id
53310	Borrowed	2020-04-08 13:00:00	2020-05-20 13:00:00	NULL	22220	5	41100
53311	Borrowed	2020-04-09 11:00:00	2020-04-29 11:00:00	NULL	22220	1	41100
53312	Borrowed	2020-04-11 15:00:00	2020-05-11 15:00:00	NULL	22220	2	41100
53313	Borrowed	2020-05-19 14:05:05	2020-08-19 14:05:05	NULL	22221	3	41100
53314	Borrowed	2020-03-22 01:00:00	2020-04-22 01:00:00	NULL	22221	4	41100
53315	Borrowed	2021-08-12 09:00:00	2021-10-12 11:00:00	NULL	22221	1	41101
53316	Borrowed	2020-01-23 15:30:00	2020-03-12 15:30:00	NULL	22221	2	41101
53317	Borrowed	2020-06-02 14:30:00	2020-07-02 14:30:00	NULL	22221	3	41101
53318	Borrowed	2020-04-12 11:00:00	2020-05-12 11:00:00	NULL	22221	1	41102
53319	Borrowed	2020-11-12 20:07:00	2020-12-12 20:07:00	NULL	22222	2	41102
53320	Borrowed	2020-07-09 11:00:00	2020-09-13 11:00:00	NULL	22222	3	41102
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

After update the record, the invoices will be generated

```
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Late',`actual_return_date` =
'2022-01-01 00:00:00' WHERE (`record_id` = '53319');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Late',`actual_return_date` =
'2021-02-01 00:00:00' WHERE (`record_id` = '53320');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Return', `actual_return_date`
= '2020-04-14' WHERE (`record_id` = '53318');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Return', `actual_return_date`
= '2020-07-01' WHERE (`record_id` = '53317');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Return', `actual_return_date`
= '2020-01-24' WHERE (`record_id` = '53316');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Return', `actual_return_date`
= '2021-10-11' WHERE (`record_id` = '53315');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Late', `actual_return_date` =
'2021-05-10' WHERE (`record_id` = '53314');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Late', `actual_return_date` =
'2020-08-30' WHERE (`record_id` = '53313');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Return', `actual_return_date`
= '2020-05-10' WHERE (`record_id` = '53312');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Late', `actual_return_date` =
'2021-05-04' WHERE (`record_id` = '53311');
UPDATE `errorfree`.`Record` SET `Return_Status` = 'Late', `actual_return_date` =
'2022-04-10' WHERE (`record_id` = '53310');
```

record_id	Return_Status	borrowdate	expected_return_d...	actual_return_date	customer_id	copy_id	book_id
53310	Late	2020-04-08 13:00:00	2020-05-20 13:00:00	2022-04-10 00:00:00	22220	5	41100
53311	Late	2020-04-09 11:00:00	2020-04-29 11:00:00	2021-05-04 00:00:00	22220	1	41100
53312	Return	2020-04-11 15:00:00	2020-05-11 15:00:00	2020-05-10 00:00:00	22220	2	41100
53313	Late	2020-05-19 14:05:05	2020-08-19 14:05:05	2020-08-30 00:00:00	22221	3	41100
53314	Late	2020-03-22 01:00:00	2020-04-22 01:00:00	2021-05-10 00:00:00	22221	4	41100
53315	Return	2021-08-12 09:00:00	2021-10-12 11:00:00	2021-10-11 00:00:00	22221	1	41101
53316	Return	2020-01-23 15:30:00	2020-03-12 15:30:00	2020-01-24 00:00:00	22221	2	41101
53317	Return	2020-06-02 14:30:00	2020-07-02 14:30:00	2020-07-01 00:00:00	22221	3	41101
53318	Return	2020-04-12 11:00:00	2020-05-12 11:00:00	2020-04-14 00:00:00	22221	1	41102
53319	Late	2020-11-12 20:07:00	2020-12-12 20:07:00	2022-01-01 00:00:00	22222	2	41102
53320	Late	2020-07-09 11:00:00	2020-09-13 11:00:00	2021-02-01 00:00:00	22222	3	41102
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

invoice_id	invoice_amount	invoice_stat...	record_id
1	160.00	0	53319
2	69.60	0	53320
3	0.40	0	53318
4	5.80	0	53317
5	0.20	0	53316
6	12.00	0	53315
7	159.40	0	53314
8	22.80	0	53313
9	5.80	0	53312
10	152.00	0	53311
11	284.40	0	53310

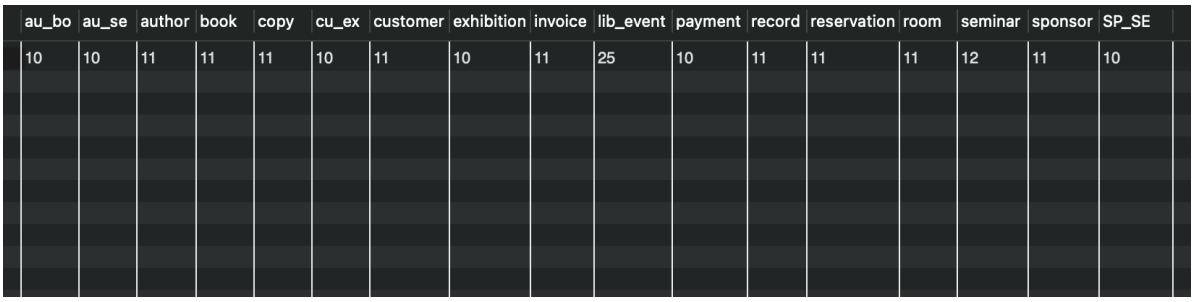
Then we can insert payment

```
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('1', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('2', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('3', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('4', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('5', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('6', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('7', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('8', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('9', 'cash', '10', '1');
INSERT INTO `errorfree`.`payment` (`payment_id`, `paymentmethod`,
`paymentamount`, `invoice_id`) VALUES ('10', 'cash', '10', '1');
```

7. Count query

```
SELECT
(SELECT COUNT(*) FROM au_bo) AS au_bo,
(SELECT COUNT(*) FROM au_se) AS au_se,
(SELECT COUNT(*) FROM author) AS author,
(SELECT COUNT(*) FROM book) AS book,
```

```
(SELECT COUNT(*) FROM copy) AS copy,
(SELECT COUNT(*) FROM cu_ex) AS cu_ex,
(SELECT COUNT(*) FROM customer) AS customer,
(SELECT COUNT(*) FROM exhibition) AS exhibition,
(SELECT COUNT(*) FROM invoice) AS invoice,
(SELECT COUNT(*) FROM lib_event) AS lib_event,
(SELECT COUNT(*) FROM payment) AS payment,
(SELECT COUNT(*) FROM record) AS record,
(SELECT COUNT(*) FROM reservation) AS reservation,
(SELECT COUNT(*) FROM room) AS room,
(SELECT COUNT(*) FROM seminar) AS seminar,
(SELECT COUNT(*) FROM sponsor) AS sponsor,
(SELECT COUNT(*) FROM SP_SE) AS SP_SE
```



au_bo	au_se	author	book	copy	cu_ex	customer	exhibition	invoice	lib_event	payment	record	reservation	room	seminar	sponsor	SP_SE
10	10	11	11	11	10	11	10	11	25	10	11	11	11	12	11	10

8. Data Dictionary Query

```
SELECT a.TABLE_NAME "table",
a.COLUMN_NAME "column",
a.COLUMN_TYPE "type",
a.COLUMN_DEFAULT "default_value",
a.IS_NULLABLE "is_optional",
a.CHARACTER_SET_NAME "CharsetName",
a.COLLATION_NAME "CollationName",
CONCAT_WS(',',a.COLUMN_COMMENT,a.COLUMN_KEY, a.EXTRA) "column_comments",

b.TABLE_COMMENT "table_comments"
FROM information_schema.COLUMNS a,information_schema.TABLES b
WHERE a.TABLE_SCHEMA=b.TABLE_SCHEMA
AND a.TABLE_SCHEMA='errorfree'
AND a.TABLE_NAME=b.TABLE_NAME
order by a.ORDINAL_POSITION;
```

table	column	type	default_value	is_optional	CharsetName	CollationName	column_comments
au_bo	author_id	bigint	NULL	NO	NULL	NULL	,MUL,
Invoice	invoice_id	bigint	NULL	NO	NULL	NULL	UNIQUE INVOICE ID,PRI,
au_se	invitation_id	bigint	NULL	NO	NULL	NULL	INVITATION ID OF AUTHOR,PRI,
exhibition	event_id	bigint	NULL	NO	NULL	NULL	UNIQUE ID OF THE EVENT,PRI,
lib_event	event_id	bigint	NULL	NO	NULL	NULL	UNIQUE ID OF THE EVENT,PRI,
author	author_id	bigint	NULL	NO	NULL	NULL	UNQIUE AUTHOR ID,PRI,
payment	payment_id	bigint	NULL	NO	NULL	NULL	UNIQUE PAYMENT ID,PRI,
sponsor	sponsor_id	bigint	NULL	NO	NULL	NULL	UNIQUE SPONSOR ID,PRI,
Record	record_id	int	NULL	NO	NULL	NULL	UNIQUE RECORD ID,PRI,
SP_SE	sponsor_id	bigint	NULL	NO	NULL	NULL	,PRI,
customer	customer_id	bigint	NULL	NO	NULL	NULL	UNQIUE CUSTOMER ID,PRI,
seminar	event_id	bigint	NULL	NO	NULL	NULL	UNIQUE ID OF THE EVENT,PRI,
reservation	rev_day	datetime	NULL	NO	NULL	NULL	DATE OF THE RESERVATION,PRI,
book	book_id	bigint	NULL	NO	NULL	NULL	UNIQUE BOOK ID,PRI,
cu_ex	registration_id	bigint	NULL	NO	NULL	NULL	UNIQUE REGISTRATION ID,PRI,
room	room_id	bigint	NULL	NO	NULL	NULL	UNIQUE ROOM ID,PRI,
copy	book_id	bigint	NULL	NO	NULL	NULL	,PRI,
copy	copy_id	varchar(2)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	UNIQUE COPY ID,PRI,
reservation	session_id	char(1)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	TIME SLOT OF THE EVENT,PRI,
book	bookname	char(20)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	,,
cu_ex	customer_id	bigint	NULL	NO	NULL	NULL	,MUL,
room	room_capacity	bigint	NULL	NO	NULL	NULL	CAPACITY OF ROOM,,
seminar	topic	varchar(30)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	TOPIC OF SEMINAR,,
customer	firstname	varchar(20)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	FIRST NAME OF CUSTOMER,,
Record	Return_Status	enum('Borrowed','Return','Late')	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	,,
SP_SE	amount	decimal(20,2)	NULL	NO	NULL	NULL	AMOUNT SPONSOR SUPPORT,,
payment	paymentmethod	varchar(20)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	PAYMENT METHOD: CASH, CREDIT, DEBIT, PAYPAL,,
author	firstname	varchar(10)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	FIRST NAME OF AUTHOR,,
lib_event	eventtype	enum('E','R')	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	,,
sponsor	firstname	varchar(10)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	FIRST NAME OF SPONSOR,,
au_se	author_id	bigint	NULL	NO	NULL	NULL	,MUL,
exhibition	expense	decimal(10,2)	NULL	YES	NULL	NULL	EXPENSE OF THE EXHIBITION,,
au_bo	book_id	bigint	NULL	NO	NULL	NULL	,MUL,
Invoice	invoice_amount	decimal(7,2)	NULL	NO	NULL	NULL	AMOUNT OF THE INVOICE,,
Invoice	invoice_status	char(1)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	STATUS OF THE INVOCIE,,
sponsor	lastname	varchar(10)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	LAST NAME OF SPONSOR,,
au_se	event_id	bigint	NULL	NO	NULL	NULL	,MUL,
author	lastname	varchar(10)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	LAST NAME OF AUTHOR,,
lib_event	eventname	varchar(50)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	NAME OF THE EVENT,,
SP_SE	event_id	bigint	NULL	NO	NULL	NULL	,PRI,
book	topic	varchar(20)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	TOPIC OF THE BOOK,,
reservation	room_id	bigint	NULL	NO	NULL	NULL	,PRI,
copy	availability	tinyint(1)	NULL	YES	NULL	NULL	,,
payment	paymentamount	decimal(7,2)	NULL	NO	NULL	NULL	AMOUNT OF THE PAYMENT,,
cu_ex	event_id	bigint	NULL	NO	NULL	NULL	,MUL,
Record	borrowdate	datetime	NULL	NO	NULL	NULL	BORROWED DATE OF COPY,,
customer	lastname	varchar(20)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	LAST NAME OF CUSTOMER,,
customer	phonenumber	varchar(20)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	PHONE NAME OF CUSTOMER,,
Record	expected_return_date	datetime	NULL	NO	NULL	NULL	EXPECTED RETURN DATE OF THE COPY,,
Invoice	record_id	int	NULL	NO	NULL	NULL	,UNI,
payment	invoice_id	bigint	NULL	NO	NULL	NULL	,MUL,
reservation	customer_id	bigint	NULL	NO	NULL	NULL	,MUL,
author	email_address	varchar(20)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	EMAIL ADDRESS OF THE AUTHOR,,
lib_event	starttime	datetime	NULL	NO	NULL	NULL	START TIME OF THE EVENT,,
Record	actual_return_date	datetime	NULL	YES	NULL	NULL	ACUTUAL RETURN DATE OF THE COPY,,
customer	email	varchar(30)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	EMAIL OF CUSTOMER,,
reservation	number_of_people	int	NULL	NO	NULL	NULL	NUMBER OF PEOPLE FOR THE RESERVATION,,
lib_event	endtime	datetime	NULL	NO	NULL	NULL	END TIME OF THE EVENT,,
author	street	varchar(30)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	STREET ADDRESS,,
Record	customer_id	bigint	NULL	NO	NULL	NULL	,MUL,
customer	passport	varchar(20)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	PASSPORT OF CUSTOMER,,
author	city	varchar(30)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	CITY ADDRESS,,
author	state	varchar(30)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	STATE,,
customer	ssn	varchar(20)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	SSN OF CUSTOMER,,
Record	copy_id	varchar(2)	NULL	NO	utf8mb4	utf8mb4_0900_ai_ci	,MUL,

table	column	type	default_value	is_optional	CharsetName	CollationName	column_comments
Record	book_id	bigint	NULL	NO	NULL	NULL	,,
author	zip_code	bigint	NULL	NO	NULL	NULL	ZIP CODE OF ADDRESS,,
customer	driver_license	varchar(30)	NULL	YES	utf8mb4	utf8mb4_0900_ai_ci	DRIVER'S LICENSE OF CUSTOMER,,

```
SELECT TABLE_NAME,CONSTRAINT_NAME,CONSTRAINT_TYPE
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE CONSTRAINT_SCHEMA="errorfree"
```

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	
au_bo	au_bo_author_fk	FOREIGN KEY	
au_bo	au_bo_book_fk	FOREIGN KEY	
au_se	PRIMARY	PRIMARY KEY	
au_se	au_se_author_fk	FOREIGN KEY	

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	
au_se	au_se_seminar_fk	FOREIGN KEY	
author	PRIMARY	PRIMARY KEY	
book	PRIMARY	PRIMARY KEY	
copy	PRIMARY	PRIMARY KEY	
copy	copy_book_fk	FOREIGN KEY	
cu_ex	PRIMARY	PRIMARY KEY	
cu_ex	cu_ex_customer_fk	FOREIGN KEY	
cu_ex	cu_ex_exhibition_fk	FOREIGN KEY	
customer	PRIMARY	PRIMARY KEY	
exhibition	PRIMARY	PRIMARY KEY	
exhibition	exhibition_lib_event_fk	FOREIGN KEY	
Invoice	invoice__idx	UNIQUE	
Invoice	PRIMARY	PRIMARY KEY	
Invoice	invoice_record_fk	FOREIGN KEY	
Invoice	invoice_ck	CHECK	
lib_event	PRIMARY	PRIMARY KEY	
lib_event	ch_inh_lib_event	CHECK	
lib_event	event_ck3	CHECK	
payment	PRIMARY	PRIMARY KEY	
payment	payment_invoice_fk	FOREIGN KEY	
payment	payment_ck	CHECK	
Record	PRIMARY	PRIMARY KEY	
Record	record_copy_fk	FOREIGN KEY	
Record	record_customer_fk	FOREIGN KEY	
Record	record_ck	CHECK	
Record	record_ck2	CHECK	
reservation	PRIMARY	PRIMARY KEY	
reservation	reservation_customer_fk	FOREIGN KEY	
reservation	reservation_room_fk	FOREIGN KEY	

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	
room	PRIMARY	PRIMARY KEY	
seminar	PRIMARY	PRIMARY KEY	
seminar	seminar_lib_event_fk	FOREIGN KEY	
SP_SE	PRIMARY	PRIMARY KEY	
SP_SE	sp_se_seminar_fk	FOREIGN KEY	
SP_SE	sp_se_sponsor_fk	FOREIGN KEY	
SP_SE	SP_SE_ck	CHECK	
sponsor	PRIMARY	PRIMARY KEY	