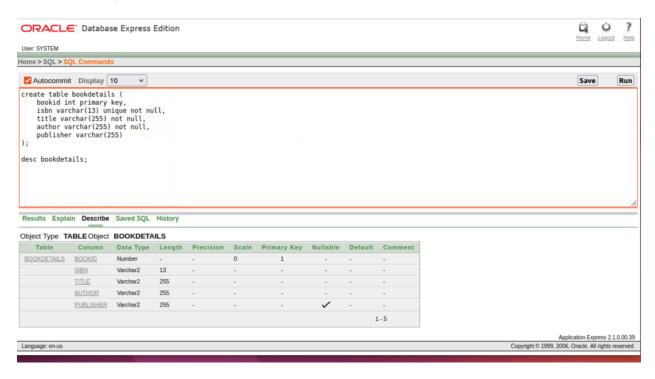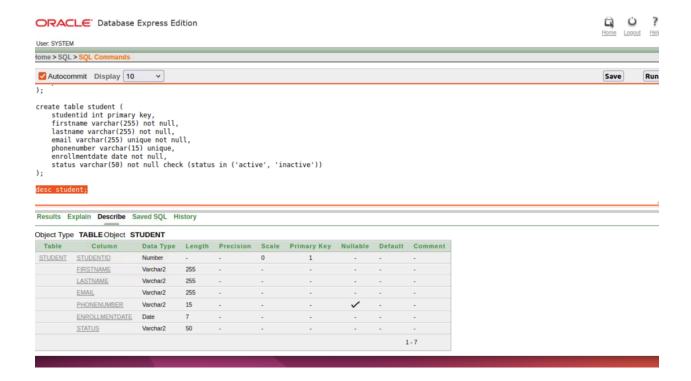# Assignment-2

Assignment 2: Design a database schema for a library system, including tables, fields, and constraints like NOT NULL, UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between tables.

```
create table bookdetails (
    bookid int primary key,
    isbn varchar(13) unique not null,
    title varchar(255) not null,
    author varchar(255) not null,
    publisher varchar(255),

);
```

desc bookdetails;



```
create table student (
    studentid int primary key,
    firstname varchar(255) not null,
    lastname varchar(255) not null,
    email varchar(255) unique not null,
    phonenumber varchar(15) unique,
    enrollmentdate date not null,
    status varchar(50) not null check (status in ('active', 'inactive'))
);
```

desc student;

User: SYSTEM

Home > SQL > SQL Commands

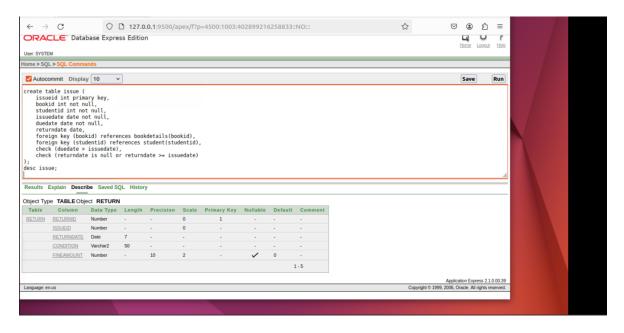☑ Autocommit  Display 10  ▾                                    Save    Run

```
);

create table student (
    studentid int primary key,
    firstname varchar(255) not null,
    lastname varchar(255) not null,
    email varchar(255) unique not null,
    phonenumber varchar(15) unique,
    enrollmentdate date not null,
    status varchar(50) not null check (status in ('active', 'inactive'))
);

desc student;
```

Results  Explain  **Describe**  Saved SQL  History

Object Type **TABLE** Object **STUDENT**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-------|--------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| STUDENT | STUDENTID | Number | - | - | 0 | 1 | - | - | - |
| | FIRSTNAME | Varchar2 | 255 | - | - | - | - | - | - |
| | LASTNAME | Varchar2 | 255 | - | - | - | - | - | - |
| | EMAIL | Varchar2 | 255 | - | - | - | - | - | - |
| | PHONENUMBER | Varchar2 | 15 | - | - | - | ✓ | - | - |
| | ENROLLMENTDATE | Date | 7 | - | - | - | - | - | - |
| | STATUS | Varchar2 | 50 | - | - | - | - | - | - |
| | | | | | | | | | 1 - 7 |

```
create table issue (
    issueid int primary key,
    bookid int not null,
    studentid int not null,
    issuedate date not null,
    duedate date not null,
    returndate date,
    foreign key (bookid) references bookdetails(bookid),
    foreign key (studentid) references student(studentid),
    check (duedate > issuedate),
    check (returndate is null or returndate >= issuedate)
);

desc issue;
```

create table return (
    returnid int primary key,
    issueid int not null,
    returndate date not null,
    condition varchar(50) not null check (condition in ('good', 'damaged', 'lost')),
    fineamount decimal(10, 2) default 0 check (fineamount >= 0),
    foreign key (issueid) references issue(issueid)
);

desc return;