1. Create a Database with the name “university” and create all the required tables and insert data in to them.

Consider the following requirements list:

• The university has many Departments.

• The Departments offer one or more programs.

• A program is made up of one or more courses.

• A student must enroll in a program.

• A student takes the courses that are part of her program.

• A program has a name, a program identifier, department id, the total credit points required to graduate, and

the year it commenced.

• A course has a name, a course identifier, program id, a credit point value, and the year it commenced.

• Students have name, a surname, a student identifier, a date of birth, and the year they first enrolled.

• When a student takes a course, the year and semester he attempted it are recorded.

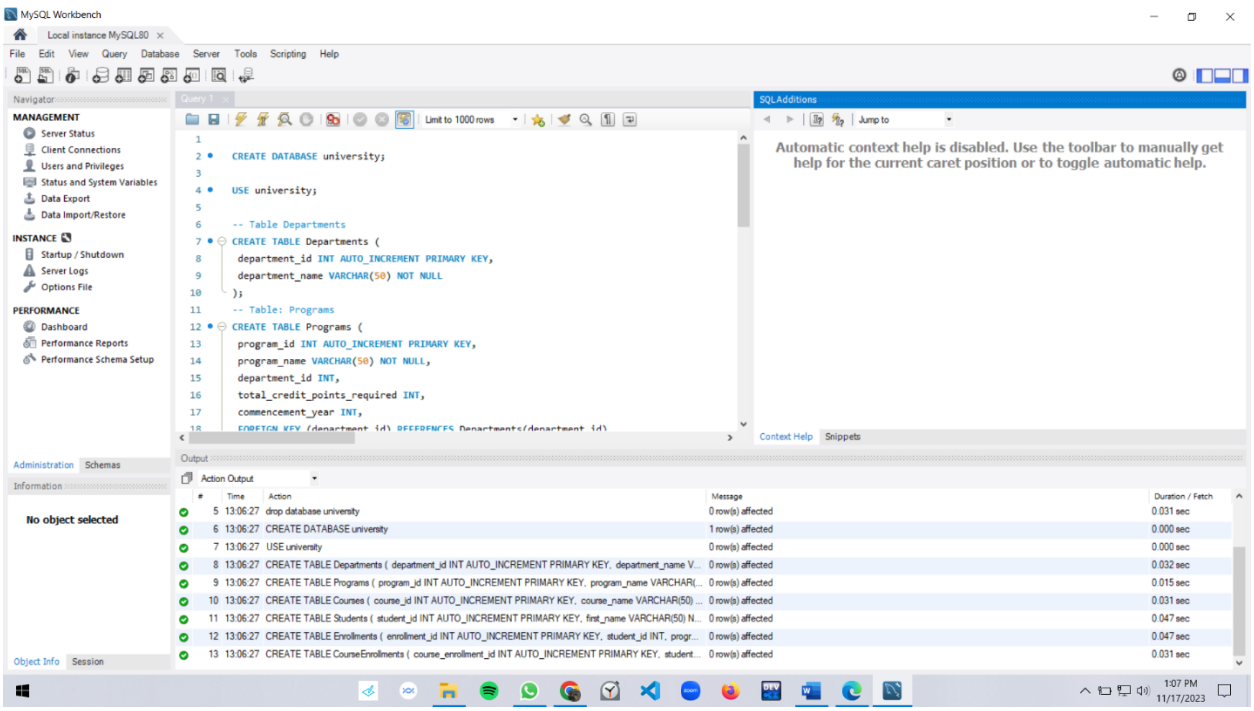
When he finishes the course, a grade (such as A or B) and a mark (such as 60 percent) are recorded.

• Each course in a program is sequenced into a year (for example, year 1) and a semester (for example, semester 1).

```
CREATE DATABASE university;

USE university;


//Table Departments

CREATE TABLE Departments (

   department_id INT AUTO_INCREMENT PRIMARY KEY,

   department_name VARCHAR(50) NOT NULL

);


// Table: Programs

CREATE TABLE Programs (
```

```
    program_id INT AUTO_INCREMENT PRIMARY KEY,
    program_name VARCHAR(50) NOT NULL,
    department_id INT,
    total_credit_points_required INT,
    commencement_year INT,
    FOREIGN KEY (department_id) REFERENCES Departments(department_id)
);

//  Table: Courses
CREATE TABLE Courses (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    course_name VARCHAR(50) NOT NULL,
    program_id INT,
    credit_points INT,
    commencement_year INT,
    FOREIGN KEY (program_id) REFERENCES Programs(program_id)
);

// Table: Students
CREATE TABLE Students (
    student_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    date_of_birth DATE,
    year_enrolled INT
);

// Table: Enrollments
CREATE TABLE Enrollments (
```

```
    enrollment_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    program_id INT,
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (program_id) REFERENCES Programs(program_id)
);

// Table: CourseEnrollments
CREATE TABLE CourseEnrollments (
    course_enrollment_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    course_id INT,
    year_attempted INT,
    semester_attempted INT,
    grade VARCHAR(2),
    mark DECIMAL(5, 2),
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```



2. Alter the above tables with Foreign Keys as required for the scenario. And perform the following activities.

a. List all students in a specific department (e.g., Computer Science)

b. List all courses a specific student (e.g., with Student ID 123) is enrolled in

c. Retrieve the total number of students enrolled in each course

d. Find the course with the highest enrolment

e. List all Programs of a particular department

f. List all courses under a specific program.

ALTER TABLE Programs ADD FOREIGN KEY (department_id) REFERENCES Departments(department_id);

ALTER TABLE Courses ADD FOREIGN KEY (program_id) REFERENCES Programs(program_id);

ALTER TABLE Enrollments ADD FOREIGN KEY (student_id) REFERENCES Students(student_id);

ALTER TABLE Enrollments ADD FOREIGN KEY (program_id) REFERENCES Programs(program_id);

ALTER TABLE CourseEnrollments ADD FOREIGN KEY (student_id) REFERENCES Students(student_id);

ALTER TABLE CourseEnrollments ADD FOREIGN KEY (course_id) REFERENCES Courses(course_id);

A)

```
SELECT * FROM Students

WHERE student_id IN (SELECT student_id FROM Enrollments WHERE program_id IN (SELECT program_id FROM Programs WHERE department_id = (SELECT department_id FROM Departments WHERE department_name = 'Computer Science')));
```

B)

```
SELECT Courses.* 0FROM Courses

JOIN CourseEnrollments ON Courses.course_id = CourseEnrollments.course_id

WHERE CourseEnrollments.student_id = 123;
```

C)

```
SELECT course_id, COUNT(student_id) AS total_students_enrolled

FROM CourseEnrollments GROUP BY course_id;
```

D)

```
SELECT course_id, COUNT(student_id) AS total_students_enrolled

FROM CourseEnrollments

GROUP BY course_id

ORDER BY total_students_enrolled DESC

LIMIT 1;
```

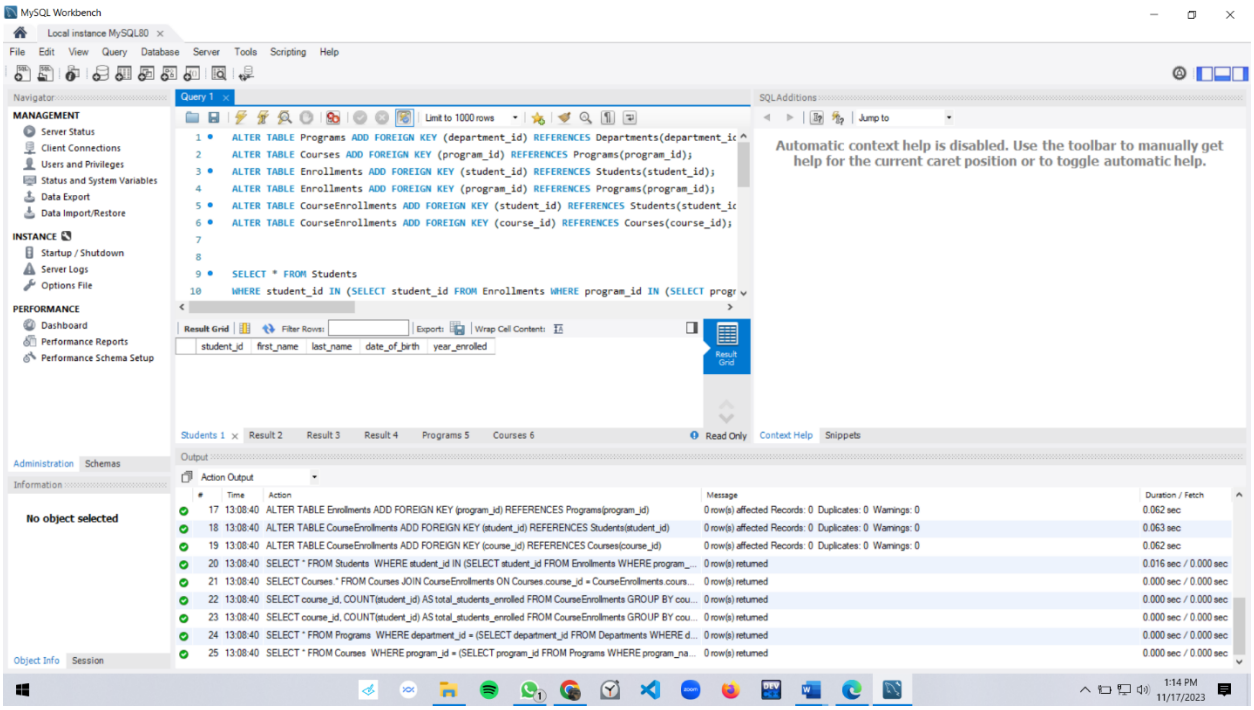
E)

```
SELECT * FROM Programs

WHERE department_id = (SELECT department_id FROM Departments WHERE department_name = 'Computer Science');
```

F)

```
SELECT * FROM Courses
```

WHERE program_id = (SELECT program_id FROM Programs WHERE program_name = 'Your Program Name');



3. The data to be recorded according to the requirements are as follows.

Staff Id, Name, Designation, staff address, staff email, staff phone No, salary, branch Id, branch Description,

Branch Address, branch phone No

a. Normalize the data and form the tables accordingly.

b. Create the normalized tables, with required relationships

c. Perform all CRUD operations on these tables

//Staff table

```
CREATE TABLE Staff (
    staff_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    designation VARCHAR(50),
    staff_address VARCHAR(100),
    staff_email VARCHAR(50),
```

```
    staff_phone_no VARCHAR(20),
    salary DECIMAL(10, 2),
    branch_id INT,
    FOREIGN KEY (branch_id) REFERENCES Branch(branch_id)
);

//Branch table
CREATE TABLE Branch (
    branch_id INT AUTO_INCREMENT PRIMARY KEY,
    branch_description VARCHAR(100),
    branch_address VARCHAR(100),
    branch_phone_no VARCHAR(20)
);
```

A)

```
INSERT INTO Branch (branch_description, branch_address, branch_phone_no)
VALUES ('SBI Branch', Pune',’123456789’);
INSERT INTO Staff (name, designation, staff_address, staff_email, staff_phone_no, salary, branch_id)
VALUES ('Amardip', 'Manager', 'Pune', 'Amardip@gmail.com', '9876543210', 50000.00, 1);
```

B)

```
SELECT * FROM Staff;
SELECT * FROM Branch;
SELECT * FROM Staff WHERE staff_id = 1;
SELECT * FROM Branch WHERE branch_id = 1;
```

C)

```
-- Update staff information (e.g., Update Staff ID 1)
UPDATE Staff
```

SET salary = 55000.00

WHERE staff_id = 1;

-- Update branch information (e.g., Update Branch ID 1)

UPDATE Branch

SET branch_description = 'Updated Branch Description'

WHERE branch_id = 1;

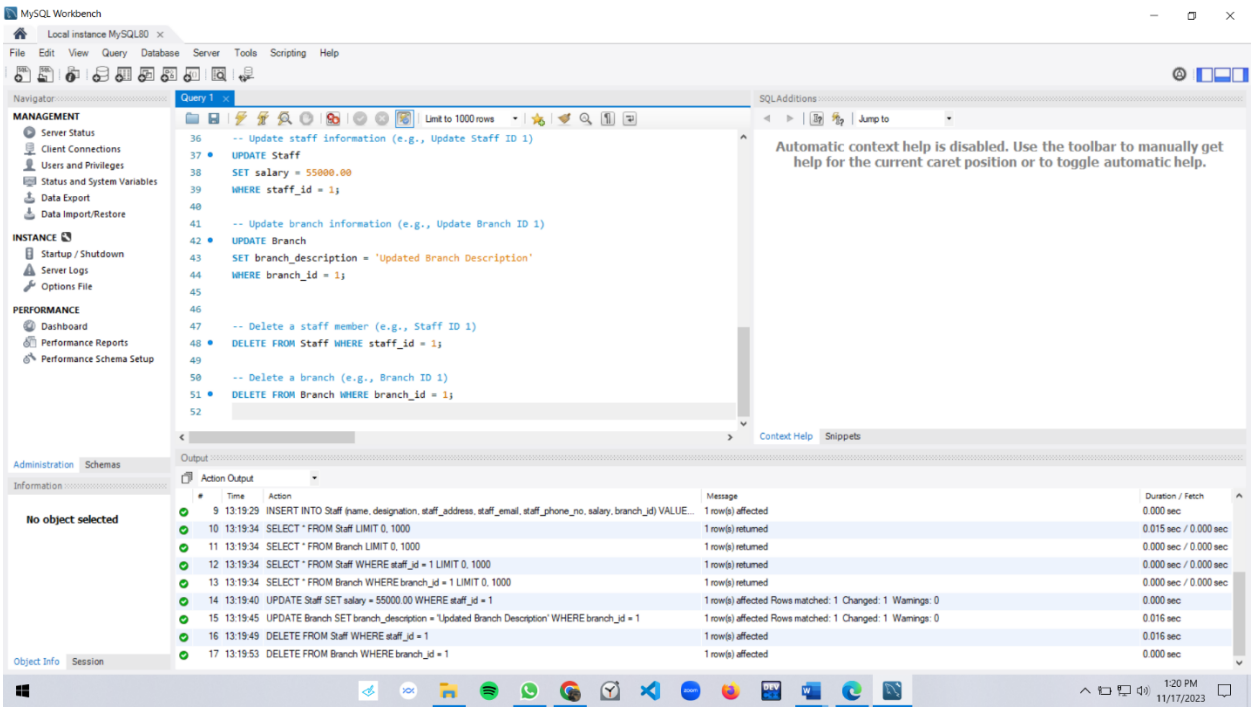
D)

-- Delete a staff member (e.g., Staff ID 1)

DELETE FROM Staff WHERE staff_id = 1;

-- Delete a branch (e.g., Branch ID 1)

DELETE FROM Branch WHERE branch_id = 1;



Q. 4

a. Insert data in all the tables.

b. Display the details of a Product, including the following data

a. Product Id, Name, desc, Category name, quantity, discount percentage (simple join)

c. Display the details about an order to include the following data

a. order ID, first name, product name, quantity. (inner join)

d. Display the following details

a. First Name, product name(use left join)

e. Display the following data

a. First name, product name (use right join)

f. Display the following data

a. First name, product name (use full outer join)

```
create database eShopping;
use eShopping;

create table user (
id int primary key not null,
username varchar(50),
password text(50),
first_name varchar(50),
last_name varchar(50),
telephone int,
created_at timestamp,
modified_at timestamp);
select * from user;
create table user_address (
user_id int,
address_line1 varchar(50),
```

```
address_line2 varchar(50),

city varchar(50),

postal_code varchar(50),

country varchar(50),

telephone int,

mobile varchar(50));

alter table user_address add column id int;

alter table user_address add primary key(id);

desc user_address;

select * from user_address;

alter table user_address add foreign key(user_id) references user(id);

insert into
user(id,username,password,first_name,last_name,telephone,created_at,modified_at)values(1,'amardip
07',1234,'Amardip','PAwar','12','2013-06-02','2013-07-01');

select * from user;

insert into
user(id,username,password,first_name,last_name,telephone,created_at,modified_at)values(2,'abbhi03',
1235,'Abhishekh','Kadam','19','2014-06-02','2014-07-01');

insert into user(id,username,password,first_name,last_name,telephone,created_at,modified_at)

values(3,'Atharv',223456,'Atharv','ABC','325','2014-02-01','2014-02-10'),

(4,'saurabh12',3234,'Saurabh','Bhau','425','2012-02-01','2012-02-20'),

(5,'Temp',3234,'Rushikesh','ABC','12251','2015-02-01','2015-02-10');

select * from user_address;

desc user_address;

insert into user_address values(11,'awer','pune','Delhi','PUN','India','7452','7896523041','1');

update user_address set user_id=11 where id = 1;

delete from user_address where user_id=1;

insert into user_address values(22,'awer','pune','Delhi','PUN','India','7452','7896523041','2');

create table user_address (

id int auto_increment primary key,
```

```
user_id int,
address_line1 varchar(50),
address_line2 varchar(50),
city varchar(50),
postal_code varchar(50),
country varchar(40),
telephone varchar(40),
mobile varchar(50)
);
select * from user;
select * from user_address;
alter table user_address add foreign key(user_id) references user(id);
desc user_address;
insert into
user_address(user_id,address_line1,address_line2,city,postal_code,country,telephone,mobile)values(1,'
Pune','ASD','Bvp','DHAN','IND','12','7845129620');
select * from user_address;
insert into
user_address(user_id,address_line1,address_line2,city,postal_code,country,telephone,mobile)
values(2,'Beng','WER','AQW','INF','IND','13','8845129620'),
(3,'Hyd','BHU','POI','KAT','IND','325','8843029620'),
(4,'GOA','WERT','VQW','SAHK','IND','3254','9845129620'),
(5,'CHE','OPI','NJI','MAN','IND','201','9685417230');
create table user_payment (
id int auto_increment primary key,
user_id int,
payment_type varchar(50),
provider varchar(50),
account_no int,
expiry date,
```

```
foreign key(user_id) references user(id));

desc user_payment;

insert into user_payment(user_id,payment_type,provider,account_no,expiry)

values(1,'online','amz','3621','2027-03-05'),(2,'card','flip','4152','2028-02-03'),(3,'card','axis','7451','2026-03-04');

select * from user_payment;

create table shopping_session (

id int auto_increment primary key,

user_id int,

total float,

created_at timestamp,

modified_at timestamp,

foreign key(user_id) references user(id));

desc shopping_session;

insert into shopping_session(user_id,total,created_at,modified_at)values(1,2.3,'2014-12-21 10:30:00','2014-12-29 11:30:00'),(2,1.2,'2013-02-14 11:40:00','2013-02-21 10:30:00'),(3,3.2,'2014-10-21 01:30:00','2014-11-21 10:45:00');

select * from shopping_session;

create table cart_item (

id int auto_increment primary key,

session_id int,

product_id int,

quantity int,

created_at timestamp,

modified_at timestamp,

foreign key(session_id) references shopping_session(id));

alter table  cart_item add foreign key(product_id) references product(id);

desc cart_item;

select * from cart_item;
```

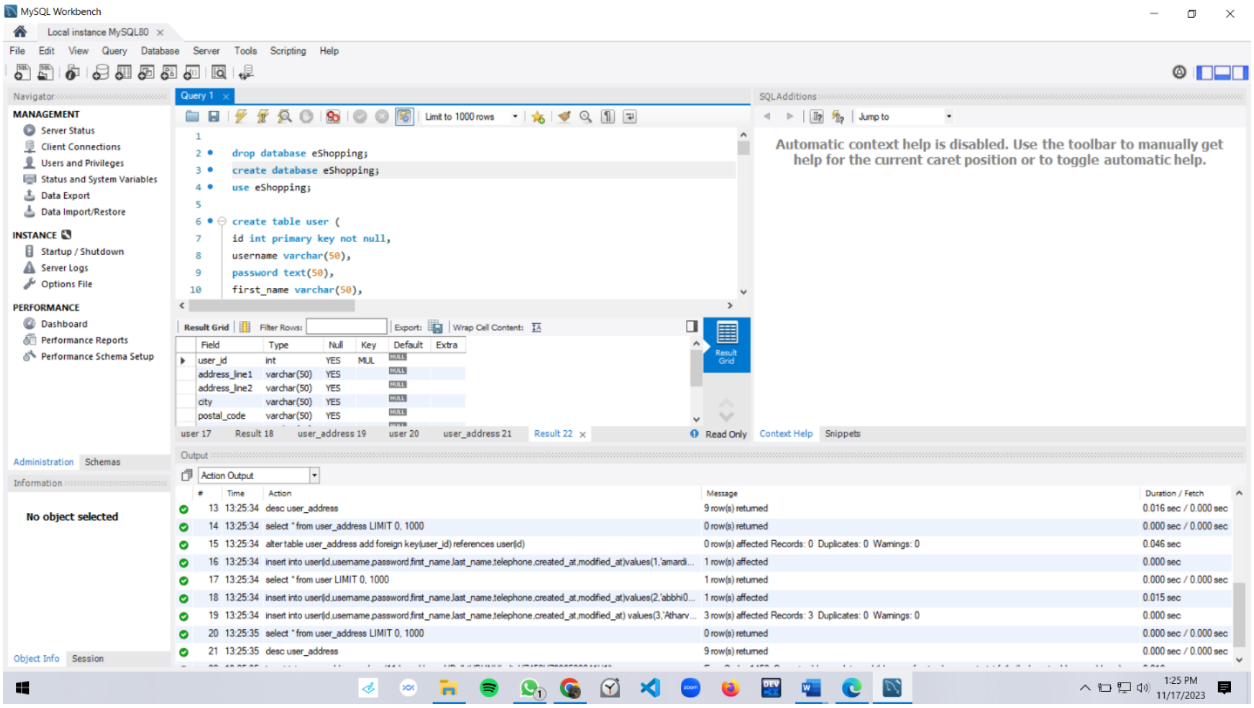
```
insert into cart_item(session_id,product_id,quantity,created_at,modified_at)

values(1,2001,1,'2014-12-21 10:45:00','2014-12-29 11:00:00'),(2,2002,2,'2013-12-11 11:45:00','2014-12-29 12:20:00'),(3,2003,1,'2015-01-21 09:45:00','2015-02-02 11:00:00');

select * from cart_item;

create table payment_details (

id int auto_increment primary key,

order_id int,

amount int,

provider varchar(50),

status varchar(50),

created_at timestamp,

modified_at timestamp);

desc payment_details;

insert into payment_details(order_id,amount,provider,status,created_at,modified_at)

values(121,1000,'gpay','received','2015-12-21 10:45:00','2015-12-29 11:00:00'),(122,1500,'gpay','received','2016-02-21 11:45:00','2016-02-22 11:00:00'),(123,2000,'phonepay','not received','2014-10-21 11:45:00','2014-11-01 11:00:00');

select * from payment_details;

desc order_details;




desc order_details;

select * from user;

desc user;


create table order_details (

id int primary key,

user_id int,

total float,
```

```
payment_id int,
created_at timestamp,
modified_at timestamp,
foreign key(user_id) references user(id));
alter table order_details add foreign key(payment_id) references payment_details(id);
desc order_details;
desc payment_details;
select * from payment_details;
insert into order_details values(1,1,20.3,1,'2014-12-02 10:30:00','2014-12-21 11:30:00');
insert into order_details
values(2,2,21.3,2,'2015-02-02 11:30:00','2015-03-21 09:30:00'),(3,3,21.3,3,'2015-11-02 01:30:00','2015-11-06 09:30:00');
select * from order_details;
create table order_items (
id int auto_increment primary key,
order_id int,
product_id int,
quantity int,
created_at timestamp,
modified_at timestamp,
foreign key(order_id) references order_details(id));
desc order_items;
create table product (
id int auto_increment primary key,
name varchar(100),
Descr varchar(100),
SKU varchar(100),
category_id int,
inventory_id int,
```

```
price float,
discount_id int,
created_at timestamp,
modified_at timestamp,
deleted_at timestamp);
alter table order_items add foreign key(product_id) references product(id);
create table cart_item (
id int auto_increment primary key,
session_id int,
product_id int,
quantity int,
created_at timestamp,
modified_at timestamp,
foreign key(session_id) references shopping_session(id));
desc cart_item;
alter table cart_item add foreign key(product_id) references product(id);
create table product_category (
id int auto_increment primary key,
name varchar(100),
descr text,
created_at timestamp,
modified_at timestamp,
deleted_at timestamp);
alter table product add foreign key(category_id) references product_category(id);
create table product_inventory (
id int auto_increment primary key,
quantity int,
created_at timestamp,
modified_at timestamp,
```

```
deleted_at timestamp);

alter table product add foreign key(inventory_id) references product_inventory(id);

create table discount (

id int auto_increment primary key,

name varchar(100),

descr text,

discount_percent float,

active boolean,

created_at timestamp,

modified_at timestamp,

deleted_at timestamp);

alter table product add foreign key(discount_id) references discount(id);

desc product_inventory;

select * from product_inventory;

insert into product_category(name,descr,created_at,modified_at,deleted_at)

values('iphone','asd','2016-02-02 10:30:00','2016-02-07 09:30:00','2016-02-11
09:30:00'),('sams','esd','2017-10-02 11:30:00','2017-10-07 10:30:00','2017-10-11
09:30:00'),('mic','asc','2016-08-02 10:30:00','2016-08-07 09:30:00','2016-08-11 09:30:00');

insert into product_inventory(quantity,created_at,modified_at,deleted_at)

values(2,'2016-02-02 10:30:00','2016-02-07 09:30:00','2016-02-11 09:30:00'),(2,'2015-02-02
10:30:00','2015-02-07 09:30:00','2015-02-11 09:30:00'),(3,'2018-02-02 10:30:00','2018-02-07
09:30:00','2018-02-11 09:30:00');

insert into discount(name,descr,discount_percent,active,created_at,modified_at,deleted_at)

values('axc','qwer',2.3,true,'2016-02-02 10:30:00','2016-02-07 09:30:00','2016-02-11 09:30:00'),

('qsd','wsde',4.2,true,'2016-03-03 10:30:00','2016-03-07 09:30:00','2016-03-11 09:30:00'),

('vcb','rfgh',4.3,true,'2016-04-02 10:30:00','2016-04-07 09:30:00','2016-04-11 09:30:00');

select * from product;

desc product;

insert into product(name, Descr, SKU, category_id, inventory_id, price, discount_id, created_at,
modified_at, deleted_at)
```

```
values('asd','wsdf','oiuy',1,1,23.3,1,'2016-04-02 10:30:00','2016-04-07 09:30:00','2016-04-11 09:30:00');

insert into product(name, Descr, SKU, category_id, inventory_id, price, discount_id, created_at,
modified_at, deleted_at)

values('sdf','polk','mkij',2,2,24.3,2,'2016-07-02 10:30:00','2016-07-07 09:30:00','2016-07-11 09:30:00'),

('qaws','edcv','wszx',3,3,14.3,3,'2016-08-02 10:30:00','2016-08-07 09:30:00','2016-08-11 09:30:00');

select * from product;

desc order_items;

insert into order_items(order_id, product_id, quantity, created_at, modified_at)

values(1,1,2,'2016-10-02 11:30:00','2016-10-07 12:30:00'),(2,2,4,'2016-11-02 11:30:00','2016-11-07
12:30:00'),(3,3,2,'2016-11-02 05:30:00','2016-11-07 12:30:00');

select * from shopping_session;

desc cart_item;

insert into cart_item(session_id, product_id, quantity, created_at, modified_at)

values(1,1,3,'2019-01-02 10:30:00','2019-01-07 12:30:00'),(2,2,4,'2019-02-12 10:30:00','2019-02-17
12:30:00'),(3,3,5,'2019-03-12 12:30:00','2019-03-17 12:45:00');

select * from discount;

select p.id,p.name,p.Descr,pc.name as category_name,oi.quantity,d.discount_percent from product p

join product_category pc on p.id = pc.id

join order_items oi on p.id = oi.id

join discount d on p.id = d.id;

select oi.id,u.first_name,p.name,oi.quantity from order_items oi

inner join user u on oi.id = u.id

inner join product p on oi.id = p.id;

select u.first_name,p.name as product_name from user u

left join order_items oi on u.id = oi.id

left join product p on oi.id = p.id;

select u.first_name,p.name as product_name from user u

right join order_items oi on u.id = oi.id

right join product p on oi.id = p.id;

select * from discount;
```


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
1
2 drop database eShopping;
3 create database eShopping;
4 use eShopping;
5
6 create table user (
7 id int primary key not null,
8 username varchar(50),
9 password text(50),
10 first_name varchar(50),
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Result Grid Filter Rows: Export: Wrap Cell Content:
Field Type Null Key Default Extra
user_id int YES MUL NULL
address_line1 varchar(50) YES NULL
address_line2 varchar(50) YES NULL
city varchar(50) YES NULL
postal_code varchar(50) YES NULL
user 17 Result 18 user_address 19 user 20 user_address 21 Result 22
Read Only Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
13 13:25:34 desc user_address 9 row(s) returned 0.016 sec / 0.000 sec
14 13:25:34 select * from user_address LIMIT 0, 1000 0 row(s) returned 0.000 sec / 0.000 sec
15 13:25:34 alter table user_address add foreign key(user_id) references user(id) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.046 sec
16 13:25:34 insert into user(id,username,password,first_name,last_name,telephone,created_at,modified_at)values(1,'amardi... 1 row(s) affected 0.000 sec
17 13:25:34 select * from user LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
18 13:25:34 insert into user(id,username,password,first_name,last_name,telephone,created_at,modified_at)values(2,'abbhi0... 1 row(s) affected 0.015 sec
19 13:25:34 insert into user(id,username,password,first_name,last_name,telephone,created_at,modified_at) values(3,'Atharv... 3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0 0.000 sec
20 13:25:35 select * from user_address LIMIT 0, 1000 0 row(s) returned 0.000 sec / 0.000 sec
21 13:25:35 desc user_address 9 row(s) returned 0.000 sec / 0.000 sec
1:25 PM
11/17/2023