Name: K.M. Tahlil Mahfuz Faruk

Student ID: 200042158

Department: CSE(SWE)

Course: CSE 4410

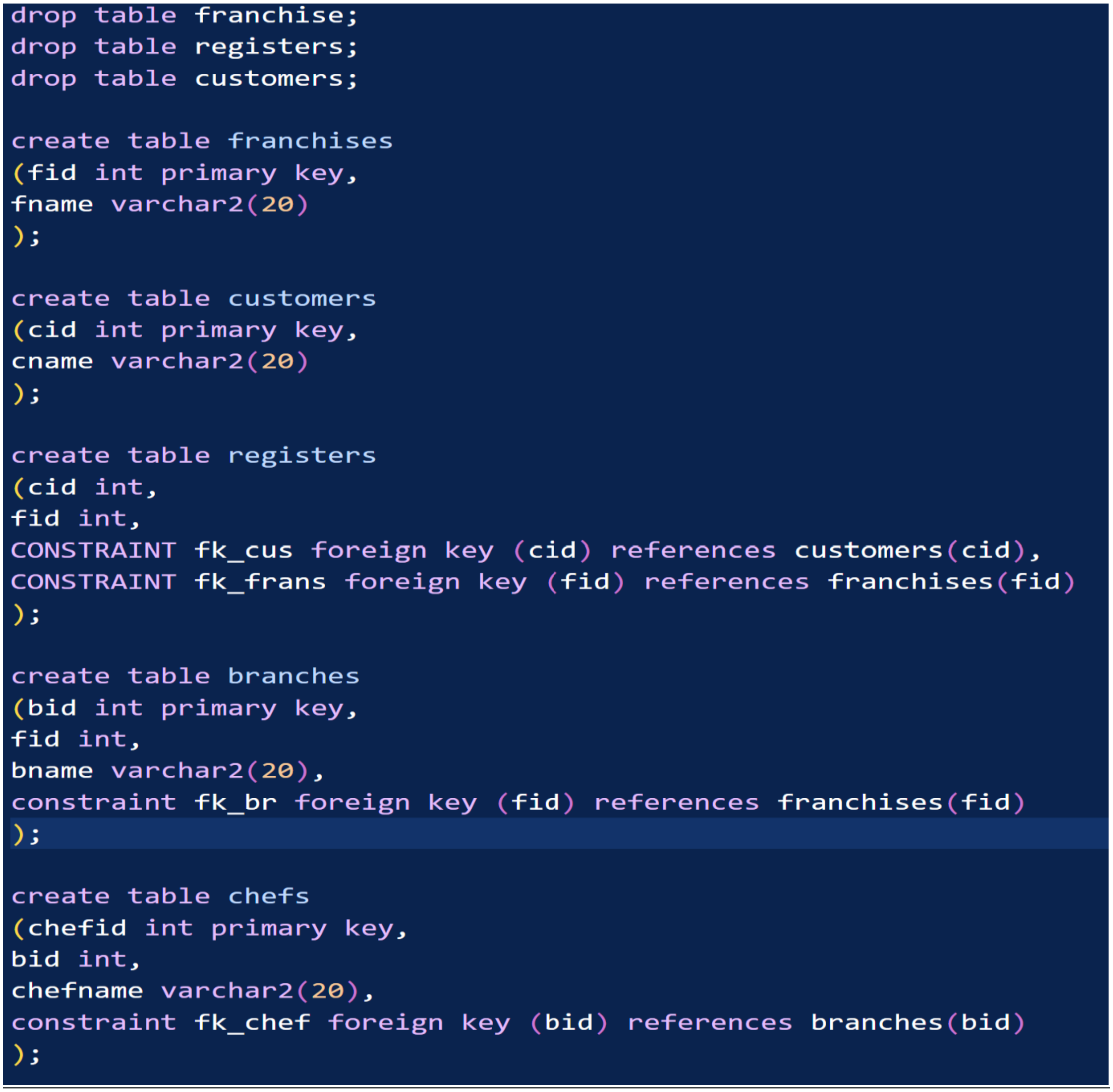
Database Management System II LAB 1

K.M. Tahlil Mahfuz Faruk
Islamic University of Technology

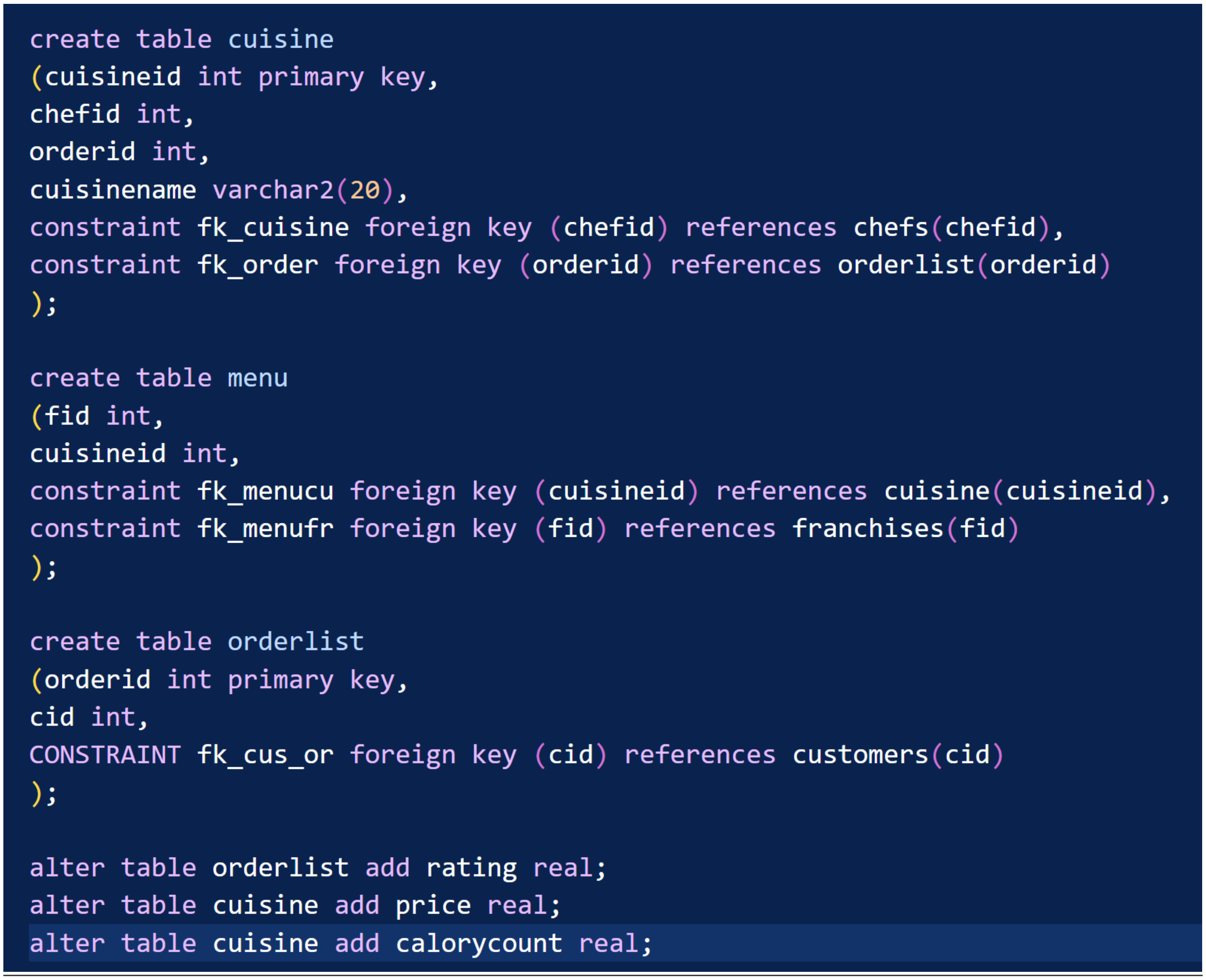
Table Creation:

The table and their cardinality is given below-

- Franchises - many to many – customers
- Branches - many to one - franchises
- Branches - one to many – chefs
- Chefs – one to many – cuisine
- Cuisine – many to many – franchise
- Cuisine – many to one – orderlist
- Customers – one to many – orderlist



```
drop table franchise;
drop table registers;
drop table customers;

create table franchises
(fid int primary key,
fname varchar2(20)
);

create table customers
(cid int primary key,
cname varchar2(20)
);

create table registers
(cid int,
fid int,
CONSTRAINT fk_cus foreign key (cid) references customers(cid),
CONSTRAINT fk_frans foreign key (fid) references franchises(fid)
);

create table branches
(bid int primary key,
fid int,
bname varchar2(20),
constraint fk_br foreign key (fid) references franchises(fid)
);

create table chefs
(chefid int primary key,
bid int,
chefname varchar2(20),
constraint fk_chef foreign key (bid) references branches(bid)
);
```



```sql
create table cuisine
(cuisineid int primary key,
chefid int,
orderid int,
cuisinename varchar2(20),
constraint fk_cuisine foreign key (chefid) references chefs(chefid),
constraint fk_order foreign key (orderid) references orderlist(orderid)
);

create table menu
(fid int,
cuisineid int,
constraint fk_menucu foreign key (cuisineid) references cuisine(cuisineid),
constraint fk_menufr foreign key (fid) references franchises(fid)
);

create table orderlist
(orderid int primary key,
cid int,
CONSTRAINT fk_cus_or foreign key (cid) references customers(cid)
);

alter table orderlist add rating real;
alter table cuisine add price real;
alter table cuisine add calorycount real;
```

Tasks:

In task 1,

Explanation:

- Natural join registers, customers, franchises
- Count(cid)
- Group them using their franchise name and id

In task 2,

Explanation:

- Natural join cuisine and orderlist
- Take the average rating
- Group them using cuisine name and id

In task 3,

Explanation:

- Natural join cuisine and orderlist
- Take the top 5 cuisines using rownum
- Group them using the number of cuisines

In task 4,

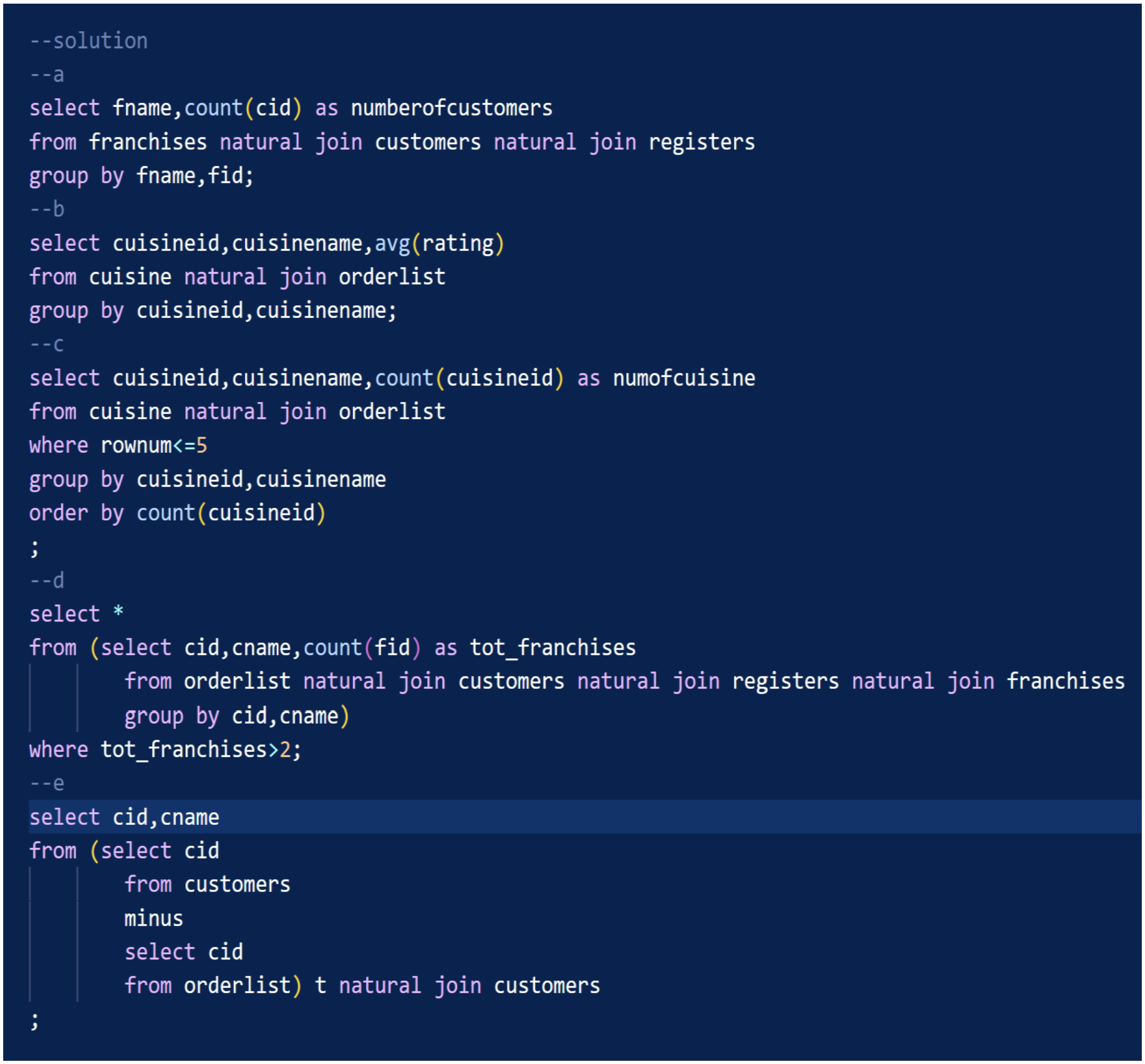
Explanation:

- Make a subquery to find out the number of franchises a customer is registered
- Then using where clause apply a condition on the total franchises number greater than 2.

In task 5,

Explanation:

- Subtract the customers of orderlist table from the customers table.
- That will provide us the customers id who has not ordered an at all.
- Natural join the subquery table (t) with customers table
  Selecting cid and cname will provide their name



```
--solution
--a
select fname,count(cid) as numberofcustomers
from franchises natural join customers natural join registers
group by fname,fid;
--b
select cuisineid,cuisinename,avg(rating)
from cuisine natural join orderlist
group by cuisineid,cuisinename;
--c
select cuisineid,cuisinename,count(cuisineid) as numofcuisine
from cuisine natural join orderlist
where rownum<=5
group by cuisineid,cuisinename
order by count(cuisineid)
;
--d
select *
from (select cid,cname,count(fid) as tot_franchises
        from orderlist natural join customers natural join registers natural join franchises
        group by cid,cname)
where tot_franchises>2;
--e
select cid,cname
from (select cid
        from customers
        minus
        select cid
        from orderlist) t natural join customers
;
```

Problems:

During the lab confusions between menu and cuisine appeared.

Solution:

If we take menu a particular table and create a relation between franchise and cuisine using many to many cardinality that will solve the problem. Now the design have multiple menus for multiple franchises. Though they might have the same name but their menu is different in this design.

Other than that no significant problem caused during the lab task.