***Part 1:***

create database homework3;

use homework3;

create table customer(

cus\_code int,

cus\_lname varchar(20),

cus\_fname varchar(20),

cus\_initial char,

cus\_areacode int,

cus\_phone int,

primary key(cus\_code));

create table invoice(

inv\_number int,

cus\_code int,

inv\_date datetime,

primary key(inv\_number),

foreign key(cus\_code) references customer(cus\_code));

create table vendor(

vend\_code int,

vend\_name varchar(30),

vend\_contact varchar(30),

vend\_areacode int,

vend\_phone int,

primary key(vend\_code));

create table product(

prod\_code int,

prod\_desc varchar(50),

prod\_price int,

prod\_quant int,

vend\_code int,

primary key(prod\_code),

foreign key(vend\_code) references vendor(vend\_code));

create table line(

inv\_number int,

prod\_code int,

line\_units int,

primary key(inv\_number, prod\_code),

foreign key(inv\_number) references invoice(inv\_number),

foreign key(prod\_code) references product(prod\_code));

***Part 2:***

/\* cus\_code, cus\_lname, cus\_fname, cus\_initial, cus\_areacode, cus\_phone \*/

insert into customer values (10010, "Ramas", "Alfred", 'A', 615, 8442573);

insert into customer values (10011, "Dunne", "Leona", 'K', 713, 8941238);

insert into customer values (10012, "Smith", "Kathy", 'W', 615, 8942285);

insert into customer values (10013, "Olowski", "Paul", 'F', 615, 2221672);

insert into customer values (10014, "Orlando", "Myron", NULL, 615, 2971228);

/\* inv\_number, cus\_code, inv\_date \*/

insert into invoice values (1001, 10011, '2008-08-03');

insert into invoice values (1002, 10014, '2008-08-04');

insert into invoice values (1003, 10012, '2008-03-20');

insert into invoice values (1004, 10011, '2008-09-23');

/\* vend\_code, vend\_name, vend\_contact, vend\_areacode, vend\_phone \*/

insert into vendor values (232, "Bryson", "Smith", 615, 2233234);

insert into vendor values (235, "SuperLoo", "Anderson", 615, 2158995);

/\* prod\_code, prod\_desc, prod\_price, prod\_quant, vend\_code \*/

insert into product values (12321, "hammer", 189 ,20, 232);

insert into product values (65781, "chain", 12, 45, 235);

insert into product values (34256, "tape", 35, 60, 235);

insert into product values (12333, "hanger", 200 ,10, 232);

/\* inv\_number, prod\_code, line\_units \*/

insert into line values (1001, 12321, 1);

insert into line values (1001, 65781, 3);

insert into line values (1002, 34256, 6);

insert into line values (1003, 12321, 5);

insert into line values (1002, 12333, 6);

/\* 1 List the First name, middle initial, last name, and area code for all customers.

\*/

SELECT customer.cus\_fname, customer.cus\_lname, customer.cus\_areacode

FROM customer

ORDER BY customer.cus\_areacode;

/\* 2 List the invoice number and invoice date for all invoices of customer number 10011.

\*/

SELECT invoice.inv\_number, invoice.inv\_date

FROM invoice

WHERE invoice.cus\_code = 10011

ORDER BY invoice.inv\_number;

/\* 3 List the product number and product quantity for products in invoice number 1001

\*/

SELECT product.prod\_code, product.prod\_quant

FROM product

JOIN line ON line.prod\_code = product.prod\_code

JOIN invoice ON line.inv\_number = invoice.inv\_number

WHERE invoice.inv\_number = 1001

ORDER BY prod\_code;

/\* 4 List all product description and product price supplied by vendor whose vendor

contact name is Smith

\*/

SELECT product.prod\_desc, product.prod\_price

FROM product

JOIN vendor ON product.vend\_code = vendor.vend\_code

WHERE vendor.vend\_contact = "Smith"

ORDER BY prod\_price;

/\* 5 Produce a list of product description, vendor name, and vendor phone for all

products with price less than 50.

\*/

SELECT product.prod\_desc, vendor.vend\_name, vendor.vend\_phone

FROM product

JOIN vendor ON vendor.vend\_code = product.vend\_code

WHERE product.prod\_price < 50

ORDER BY vend\_name;

/\* 6 For each product bought by a customer, list product description and

customer’s first name and last name.

\*/

SELECT prod\_desc, cus\_fname, cus\_lname

FROM customer

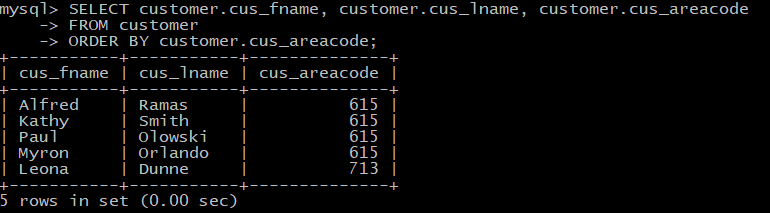
JOIN invoice ON customer.cus\_code = invoice.cus\_code

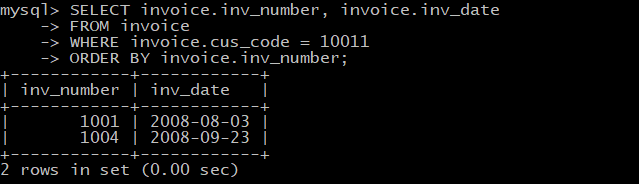
JOIN line ON line.inv\_number = invoice.inv\_number

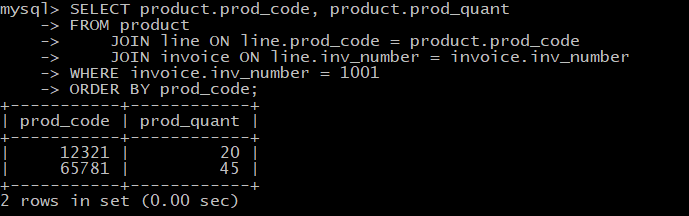
JOIN product ON product.prod\_code = line.prod\_code

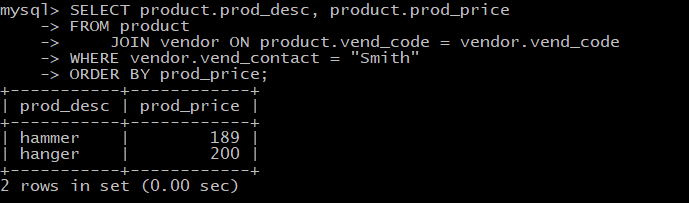
ORDER BY cus\_lname;

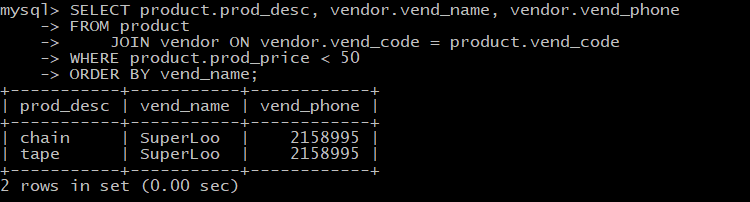
***Part 3:***

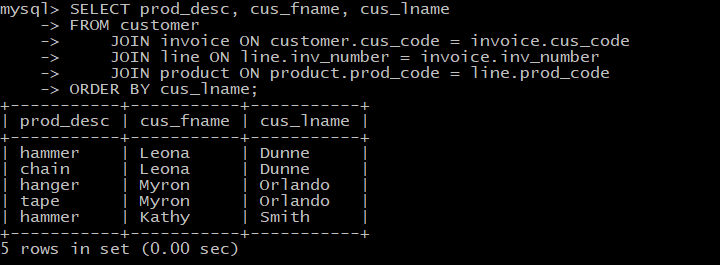












***Part 4:***

1. The statement did not execute. The cus\_code entry for this is a duplicate meaning someone in the database already has this cus\_code(PK). Primary keys must be unique; therefore, this cannot work.
2. The statement did not execute. The fact that the table INVOICE uses the foreign key cus\_code which is the primary key for the table CUSTOMER, means that for us to put the value “10017” into invoice would mean that it must first exist in the table CUSTOMER as a cus\_code. You cannot update or add to a child row information that does not exist in its parent. We could fix this error by adding an entry for a customer in table CUSTOMER using the cus\_code(PK) 10017.
3. The statement did not execute. Again, this query tries to update and add to a child row. The foreign key vend\_code in table PRODUCT is also the primary key in table VENDOR. Therefore, the value “231” must exist in table VENDOR as a vend\_code(PK) before it can be added to table PRODUCT as vend\_code(FK). We could fix this issue by adding a new vendor entry in table VENDOR with vend\_code(PK) as 231.
4. This query executed with no problems. It added a new entry into vendor with the vend\_code = 231 which should now allow us to assign products using the newly added vendor entry.
5. This query executed with no problems. We have addressed and fixed the issue faced in problem #3. Since we added a new entry to table VENDOR using the vend\_code 231, we can now assign products in table PRODUCT using vend\_code 231 since it now exists in the parent.