**Question 3 (5 Points): Checking entity and referential integrity:**

Write an SQL statement to do each of the following tasks in the given order (try 1, 2, 3, 4, then 5). Explain whether and why the statement is correctly executed or not

1. Insert the following entry in CUSTOMER

10012, ‘Juan’, ‘Rodriguez’, ‘J’, 612, 7788776

GO

INSERT INTO customer VALUES

(10012, 'Juan', 'Rodriguez', 'J', 612, 7788776)

Statement will not be executed because there is already a row with cus\_code=10012 in customer table and primary key PK\_customer doesn’t allow duplicates.

1. Insert the following entry in INVOICE

1005, 10017, ’2008-11-30’

GO

INSERT INTO invoice VALUES

(1005, 10017, '2008-11-30')

Statement will not be executed because there is no row with cus\_code=10017 in customer table and foreign key PK\_invoice\_customer can not find this value.

1. Insert the following entry in PRODUCT

12322, ‘hammer’, 189, 20, 231

GO

INSERT INTO product VALUES

(12322, 'hammer', 189, 20, 231)

Statement will not be executed because there is no row with vend\_code=10017 in vendor table and foreign key PK\_product\_vendor can not find this value.

1. Insert the following entry to the VENDOR table

231,’Adam’, ‘Eric’, 615, 2158995

GO

INSERT INTO vendor VALUES

(231, 'Adam', 'Eric', 615, 2158995)

Statement will be executed because there is no conflicts with primary key PK\_vendor.

1. Insert the following entry in PRODUCT

12322, ‘coil’, 189, 20, 231

GO

INSERT INTO product VALUES

(12322, 'coil', 189, 20, 231)

Statement will be executed because there is no conflicts with primary key and foreign key.

**SQL Queries**

Write SQL statements to answer each of the following questions.

1. List the Customer Code, Last name, and First Name for all customers.

SELECT cus\_code, cus\_lname, cus\_fname FROM customer

1. List the invoice number and invoice date for all invoices of customer number 10014.

SELECT inv\_number, inv\_date FROM invoice, customer

WHERE invoice.cus\_code = customer.cus\_code AND customer.cus\_code = 10014

1. List the product code and product quantity for products with invoice number 1001.

SELECT product.prod\_code, product.prod\_quant FROM product, invoice, line

WHERE product.prod\_code = line.prod\_code AND line.inv\_number = invoice.inv\_number

AND invoice.inv\_number = 1001

1. List all product description and product price supplied by vendor whose vendor contact is ‘Nobody’.

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

WHERE product.vend\_code = vendor.vend\_code AND vendor.vend\_contact = 'Nobody'

1. Produce a list of product description, vendor name, and vendor phone for all products with quantity less than or equal to 60.

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

WHERE product.vend\_code = vendor.vend\_code AND product.prod\_quant <= 60

1. For each product bought by a customer, list product description, customer’s first name and last name.

SELECT product.prod\_desc, customer.cus\_fname, customer.cus\_lname

FROM product, line, invoice, customer

WHERE product.prod\_code = line.prod\_code AND line.inv\_number = invoice.inv\_number

AND invoice.cus\_code = customer.cus\_code

Part 2

**Group by and Aggregates:**

**Write SQL statements to answer the following questions using Assignment 4****’s schema (Customer-Invoice-Line-Product-Vendor). Make sure that your SQL script runs *without any errors*. Submit your answers in a .SQL file.**

1 (2 Points) - Find the count of distinct vendors that supplied products that are priced lower than 185?

SELECT COUNT(\*) AS count\_vend FROM (SELECT DISTINCT vendor.vend\_code FROM vendor, product

WHERE vendor.vend\_code = product.vend\_code AND product.prod\_price < 185) some\_table

2 (2 Points) - For each vendor, find their product that has the lowest product quantity. Your output should include vendor code, vendor name, product description and product quantity for each vendor.

*Hint: Use subquery to get minimum quantity*

SELECT vendor.vend\_code, vendor.vend\_name, product.prod\_desc, product.prod\_quant

FROM vendor, product

WHERE product.vend\_code = vendor.vend\_code AND product.prod\_quant =

(SELECT MIN(p.prod\_quant) FROM product p WHERE p.vend\_code = vendor.vend\_code)

3 (2 Points) - Find how many products are there in each invoice. The output should include invoice number and number of products in the invoice.

SELECT invoice.inv\_number, count(\*) FROM product, invoice, line WHERE line.prod\_code = product.prod\_code

AND line.inv\_number = invoice.inv\_number GROUP BY invoice.inv\_number

4 (2 Points) - Find how many invoices are made by each customer. The output should be a list of cus\_code and for each cus\_code, the number of invoices made by this customer.

SELECT customer.cus\_code, count(\*) AS num\_of\_invoices FROM customer, invoice

WHERE invoice.cus\_code = customer.cus\_code GROUP BY customer.cus\_code

5 (2 Points) - Find the total value for all products in the inventory. The total value in the inventory is the sum of product quantity \* product price for all products listed in the product table.

SELECT sum(product.prod\_quant\*product.prod\_price) FROM product

6 (2 Points) - Find vendor code, vendor contact, and the number of products supplied by each vendor.

SELECT vendor.vend\_code, vendor.vend\_contact, count(\*) AS num\_of\_products

FROM vendor, product

WHERE product.vend\_code = vendor.vend\_code GROUP BY vendor.vend\_code, vendor.vend\_contact

7 (2 Points) - Find product description, price, and vendor code for the cheapest (lowest price) product.

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

FROM product

WHERE product.prod\_price =

(SELECT min(p.prod\_price) FROM product p)

8 (3 Points) - For each invoice, find the total price. The total invoice price is the sum of product price\* line units for each product purchased in the invoice.

SELECT invoice.inv\_number, sum(line.line\_units\*product.prod\_price)

FROM invoice, line, product

WHERE line.inv\_number = invoice.inv\_number AND line.prod\_code = product.prod\_code

GROUP BY invoice.inv\_number

9 (3 Points) - Find how many products are bought by each customer. The output should be a list of cus\_code and for each cus\_code, the number of products purchased by this customer. A more complex query (if you want to try it), would be to list the name of the customer, along with the cus\_code.

More complex:

SELECT customer.cus\_code, customer.cus\_fname, customer.cus\_lname, count(\*) AS products\_bought

FROM invoice, line, product, customer

WHERE customer.cus\_code = invoice.cus\_code AND line.prod\_code = product.prod\_code

AND line.inv\_number = invoice.inv\_number

GROUP BY customer.cus\_code, customer.cus\_fname, customer.cus\_lname