Step One.

Develop a database to hold shoppers' wishlists for a shopping website (most of them have a feature like this, holding things you wish you had but aren't buying). Your database should have at least two tables and a foreign key/primary key relationship between them. You can decide on appropriate columns in each table; you need to store customer information and wish list items. At a minimum, include customer name, customer email address, wishlist item name, and quantity (how many they want of a wishlist item).

For maximum points your database must be in third normal form.

Submit the CREATE TABLE statements, and if needed, ALTER TABLE statements that would create the database.

If you wish to add some text about the design of your database, that additional information would be welcome.

***Table shopper\_litem***

***(Shopper id references shopper\_info shopper\_id references items item\_id, item id )pk, quantity***

CREATE TABLE shopper\_wl(

shopper\_id NUMBER(5) REFERENCES shopper\_info(shopper\_id),

item\_id NUMBER(5) REFERENCES store\_items(item\_id),

quantity NUMBER(3),

PRIMARY KEY (shopper\_id, item\_id))

***Table items***

***(item id)pk, description***

CREATE TABLE store\_items(

item\_id NUMBER(5) PRIMARY KEY,

description VARCHAR(50) NOT NULL)

ALTER TABLE store\_items

ADD cost NUMBER(7,2)

***Talbe shopper\_info***

***(Shopper id)pk, first name, last name, age, gender, email***

CREATE TABLE shopper\_info (

shopper\_id NUMBER(5) PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

age NUMBER(3),

gender VARCHAR(50),

email VARCHAR(50) NOT NULL)

Step Two.

Populate the database with at least 3 customers and 2 items on each customers' wishlist. It is fine to use "nonsense" text for the wishlist items, but each wishlist item should contain different text.

Submit the INSERT, and if needed, UPDATE, statements that would populate the database.

If you wish to add some text about the design of your statements, that additional information would be welcome.

INSERT INTO store\_items VALUES(1,'hand soap')

UPDATE store\_items

SET cost = 2.5

WHERE item\_id = 1

INSERT INTO store\_items VALUES(2,'desk chair')

UPDATE store\_items

SET cost = 200

WHERE item\_id = 2

INSERT INTO store\_items VALUES(3, 'beach towel')

UPDATE store\_items

SET cost = 45.99

WHERE item\_id = 3

INSERT INTO shopper\_info VALUES(1, 'drew', 'olson', 25, 'male', 'drew@oc.edu')

INSERT INTO shopper\_info VALUES(2, 'sam', 'wise', 30, 'male', 'samw@oc.edu')

INSERT INTO shopper\_info VALUES(3, 'karen', 'mulch', 18, 'female', 'karenm@oc.edu')

INSERT INTO shopper\_wl VALUES(1, 1, 2)

INSERT INTO shopper\_wl VALUES(2, 3, 5)

INSERT INTO shopper\_wl VALUES(2, 1, 1)

INSERT INTO shopper\_wl VALUES(3, 1, 10)

INSERT INTO shopper\_wl VALUES(3, 3, 1)

INSERT INTO shopper\_wl VALUES(3, 2, 2)

Step Three.

Pose at least 3 questions that your wishlist database could be asked, and write queries that would provide those answers.

The questions should be about customers & wishlists from a business/data perspective, not about SQL (don't say, for example "use a LIKE in a query" as your question - make it something someone would want to find out about that was in a wishlist database).

The queries should focus on answering the question regardless of the contents of the database, they should not rely on your current data from question 2, but should work even if the data changed. The questions should be business questions, that might be asked by casual users of the database.

For maximum points, these queries need to show a grasp of SQL features such as aggregates, joins, compound conditions, and complex expressions.

For each question, provide:

(a) the question written in English

(b) the SQL SELECT statement that when run would produce the answer to the question as its result

If you wish to add some text about the design of your questions and statements, that additional information would be welcome.

Provide the first name, email, and cart subtotal so you can send a reminder email using the first name as an introduction to have people return to the site and checkout

SELECT first\_name, email, SUM(cost) AS cart\_total

FROM store\_items NATURAL JOIN shopper\_wl

NATURAL JOIN shopper\_info

GROUP BY shopper\_id, first\_name, email

Provide first name and email of people over 21 who have beach related items in their card to send advertisements for 21+ beach parties

SELECT first\_name, email

FROM store\_items NATURAL JOIN shopper\_wl

NATURAL JOIN shopper\_info

WHERE description LIKE '%beach%' AND age >21

Determine which carts have an unacceptable items to cost ratio where the shipping cost could cut into profits

SELECT shopper\_id, SUM(quantity) AS number\_in\_cart, SUM(cost) AS cart\_total, ROUND(SUM(cost)/SUM(quantity),1) AS ratio

FROM store\_items NATURAL JOIN shopper\_wl

GROUP BY shopper\_id

HAVING ROUND(SUM(cost)/SUM(quantity),1)<10.0