Name – Shrutika Singodia

Date – 11/27/2019

In prior iterations, you designed your database schema with both conceptual and logical ERDs, and began implementing the design using SQL. Great work! Now you have a chance to further implement your database with SQL by implementing Aspects 2 and 3.

This work is summarized in the steps below.

**STRUCTURAL DATABASE RULES:**

Structural database rules are the ones which are written attentively to make sure that they define almost all the components and constraints of a particular database. These rules states the entities, relationship between those entities, participation on both sides of the relationship, plurality constraints on both sides of the relationship as well.

The nature of the database is relational and will have different table to maintain information of its respective entities. In a relational database, a table, which is also called relation represents an entity. Every row represents the instance and the columns represents the attributes.

The tables incorporated with the Amazon database is:

**Entities:**

* Products
* Transaction
* Customer
* Listing
* Shipments
* Account
* Seller
* Vendors

Now, here we have structural database business rules with relational entities and their relation between them.

1. Products information entity relationship:

The products table has the following fields.

* Product\_Id ,is the Primary key
* Name
* Product\_Description
* Product\_category
* Price
* Quantity
* Condition

2. Customer information entity relationship:

The customer table has the following fields.

* Customer\_Id is the primary key
* Firstname
* Lastname
* Address
* Phone\_number
* Email

3. Transaction information entity relationship:

The Transaction table has the following fields.

* Transaction\_Id is Primary key
* Product\_Id is foreign key
* Customer\_Id is foreign key
* Transaction\_Date
* Transaction\_Quantity
* Transaction\_Amount
* Shipping\_Speed

4. Account information entity relationship:

The Account table has the following field.

* Account\_Id is primary key
* Account\_Open\_Date
* Customer\_Id is Foreign Key

5. Shipment information entity relationship:

The Shipment table has the following field.

* Shipment\_Id primary key
* Transaction\_Id foreign key
* Shipment\_Date
* Shipment\_Address

6. Listing information entity relationship:

The listing Table has following field.

* Product\_Id primary key, foreign key
* Seller\_Id primary key, foreign key
* Listing\_Date

7. Seller information entity relationship:

The Seller Table has following field.

* Seller\_Id primary key
* Rating

8. Vendors information entity relationship:

The Vendors Table has following field.

* Vendor\_Id
* Vendor\_Address
* Company\_name
* Vendor\_Location

1). Make any enhancements and corrections to your business rules and provide the updated list

Full DBMS physical ERD with the description of the business rules.

|  |  |  |
| --- | --- | --- |
| Business rule | Relational entities | Relation |
| A seller/retailer can add only 1 new product in the database | seller to new product | 1:1 |
| A new product can have many retailers (as added) | New product to retailers | 1:M |
| A retailer can sell maximum of 3 different products | seller to product | 1:M |
| Different products can have different sellers | product to seller | 1:M |
| Product category can have up to 10 products | product category to products | 1:M |
| Each product can only be placed in one product category | product to products category | 1:1 |
| Product can have only 1 product description | Product to product description | 1:1 |
| Same description can be used for different products | product description to Product | 1:M |
| A product can be sold in different type (old, new, refurbished) | product to product type | 1:M |
| There can be more than one product in one product type | product type to product | 1:M |
| Each product can have many customers | product to customers | 1:M |
| Each customer can buy many products | customer buy many products | 1:M |
| There can be only one warehouse per city | warehouse to city | 1:1 |
| One city can only have one warehouse | city to warehouse | 1:1 |
| Each customer can have only 1 mailing address | customer to address | 1:1 |
| One mailing address can have many customers | address to customers | 1:M |
| One customer can have only one primary phone | customer to phone | 1:1 |
| One phone number can only link to one customer | phone to customer | 1:1 |
| One customer can have only one email id | customer to email | 1:1 |
| One email id can only link to one customer | email to customer | 1:1 |
| One customer can have more than one orders | Customer to order | 1:M |
| One order is linked to only one customer | Order to customer | 1:1 |
| One order can have more than one product | Order to product | 1:M |
| One product can be linked with only one order | Product to order | 1:1 |
| Each product can only have one shipping speed (super saver shipping, standard shipping, two‐day, one‐day) | Product to shipping speed | 1:1 |
| One shipping speed may have many products | Shipping speed to product | 1:M |
| Each order can have one or many tracking ids | order to tracking ids | 1:M |
| Each tracking id is linked to only one order | tracking id to order | 1:1 |

Vendor table is in relation with Product table:

Business rule: Vendor Table is in one to many relation with Product Table.

Each vendor can have many Products, But each Product order refers to only one vendor, This represents one to many relationship between the Product table and vendor table.

Product is in one to many relation with listing table.

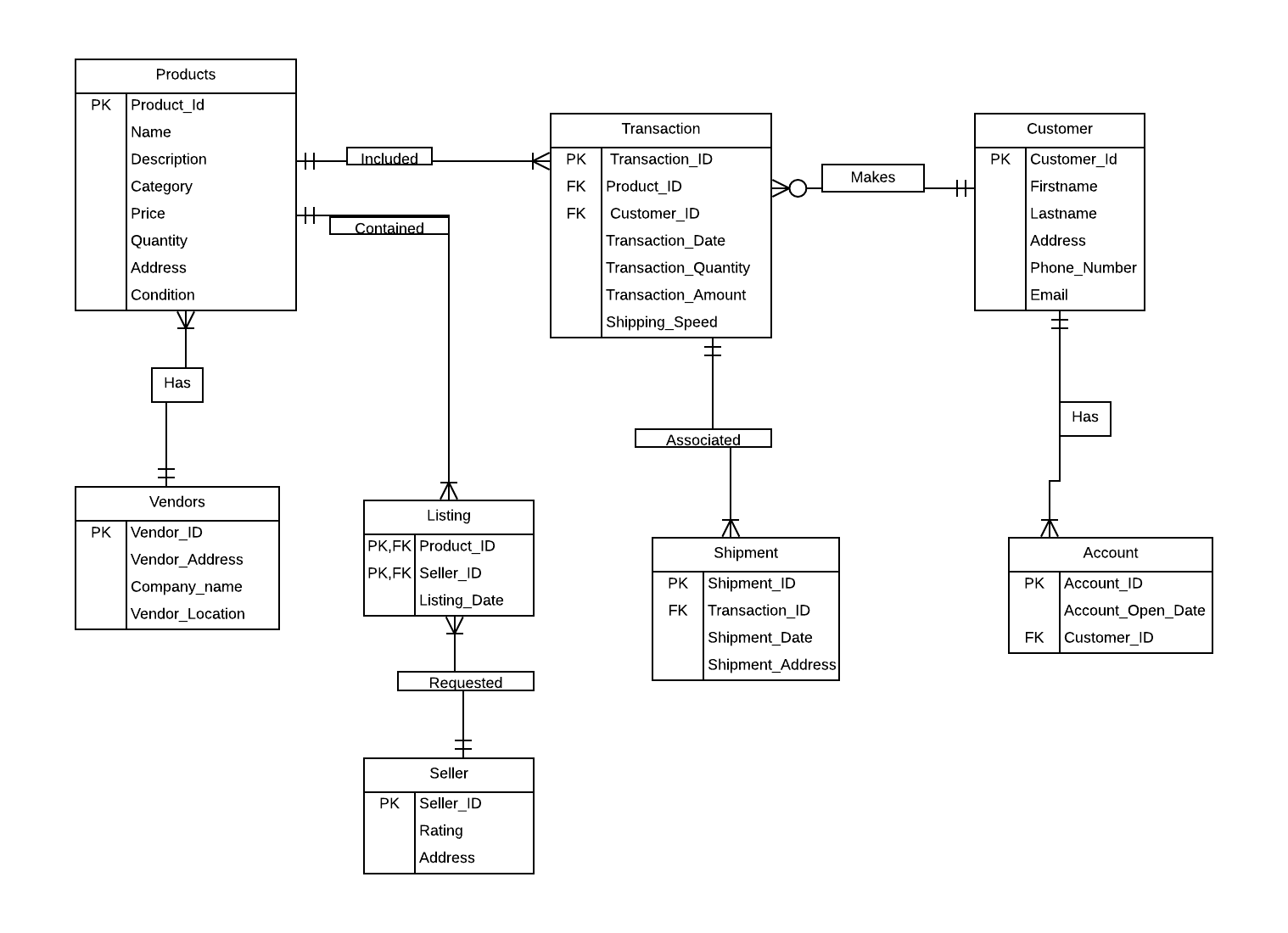
Each Product contained many listing and each product can be purchased by many sellers which let us to build one to many relationship with product to listing and one to many relationship with Product to listing table.

Each Customer has many accounts but each particular customer has only one account\_id which let us to build one to many relationship with Customer to Account Table. While, each customer can have many transaction corresponding to their Id’s which let us build one to many relationship.

Each Customer can buy many products, which let us to build many to many relationship between the tables customer and products.

Each transaction has many shipment but each particular transaction has only one shipment which let us to build one to many relationship between the tables transaction and shipment.

2. Make any enhancements and corrections to your conceptual ERD and provide the updated ERD below.



**Entities:**

* Products
* Transaction
* Customer
* Listing
* Shipments
* Account
* Seller
* Vendors

Now, here we have structural database business rules with relational entities and their relation between them.

1. Products information entity relationship:

The products table has the following fields.

* Product\_Id ,is the Primary key
* Name
* Product\_Description
* Product\_category
* Price
* Quantity
* Condition

2. Customer information entity relationship:

The customer table has the following fields.

* Customer\_Id is the primary key
* Firstname
* Lastname
* Address
* Phone\_number
* Email

3. Transaction information entity relationship:

The Transaction table has the following fields.

* Transaction\_Id is Primary key
* Product\_Id is foreign key
* Customer\_Id is foreign key
* Transaction\_Date
* Transaction\_Quantity
* Transaction\_Amount
* Shipping\_Speed

4. Account information entity relationship:

The Account table has the following field.

* Account\_Id is primary key
* Account\_Open\_Date
* Customer\_Id is Foreign Key

5. Shipment information entity relationship:

The Shipment table has the following field.

* Shipment\_Id primary key
* Transaction\_Id foreign key
* Shipment\_Date
* Shipment\_Address

6. Listing information entity relationship:

The listing Table has following field.

* Product\_Id primary key, foreign key
* Seller\_Id primary key, foreign key
* Listing\_Date

7. Seller information entity relationship:

The Seller Table has following field.

* Seller\_Id primary key
* Rating

8. Vendors information entity relationship:

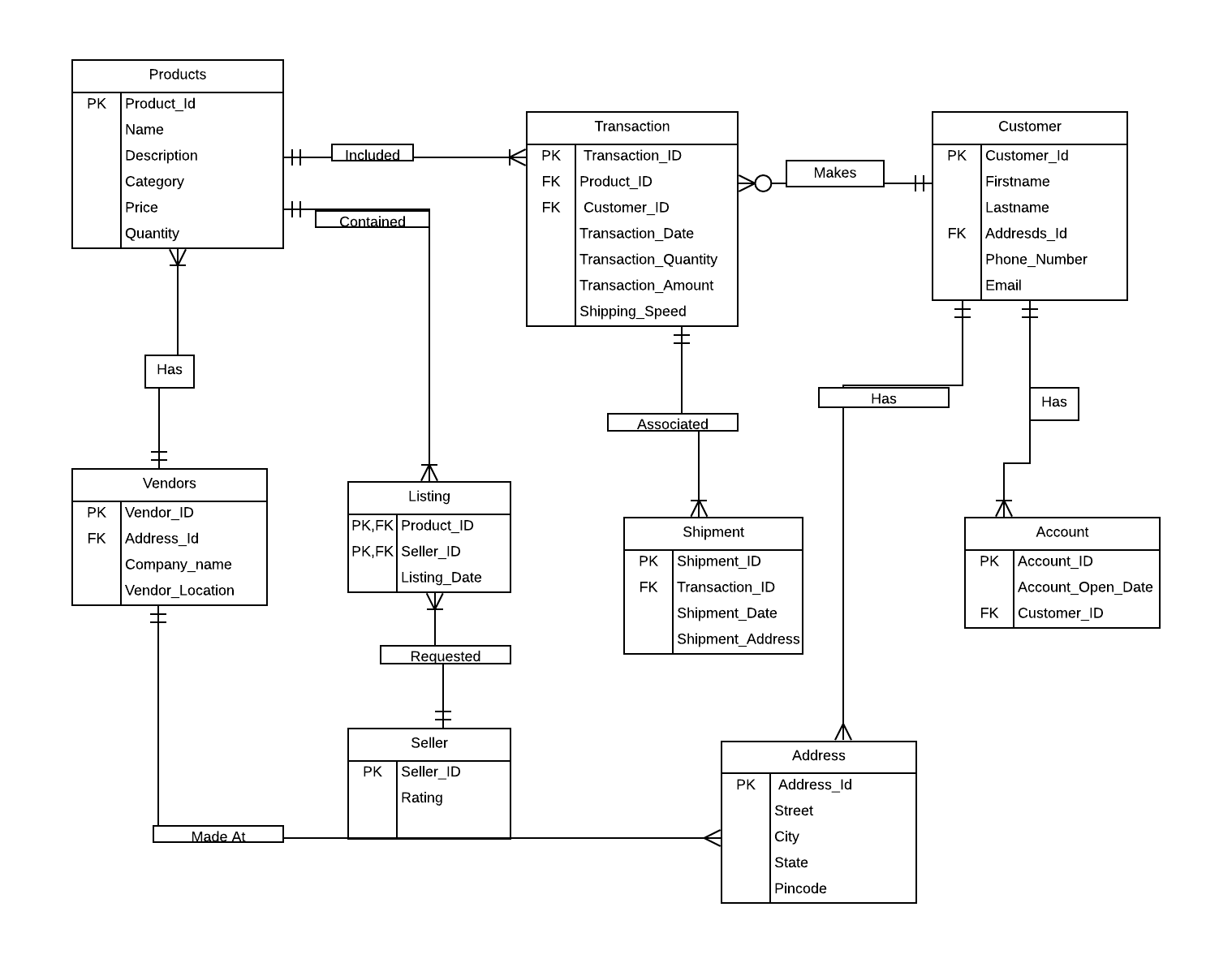
The Vendors Table has following field.

* Vendor\_Id
* Vendor\_Address
* Company\_name
* Vendor\_Location

3. Make any enhancements and corrections to your logical ERD and provide the updated ERD below.

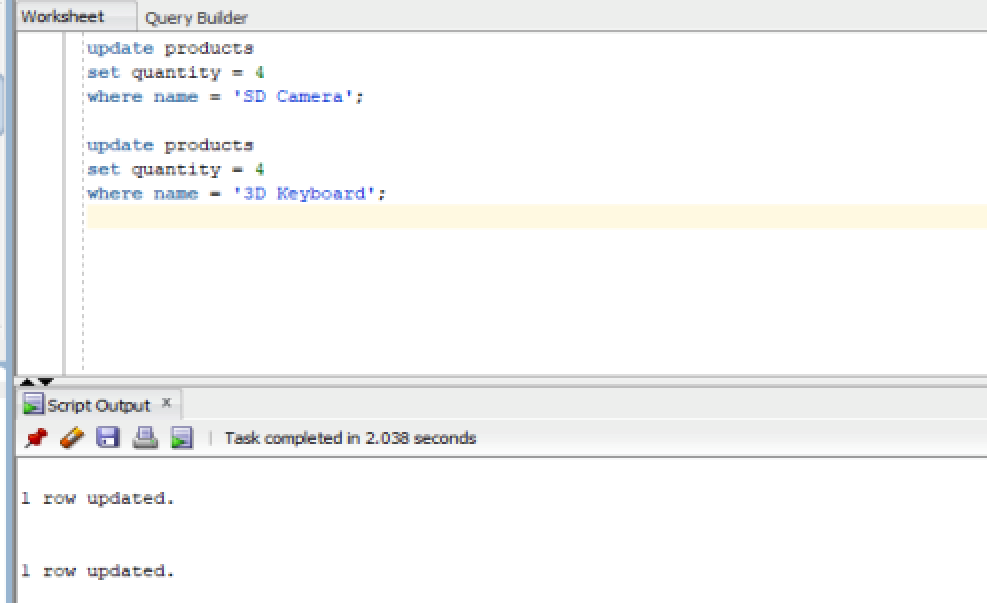
Normalization of the physical ERD to eliminate redundancy.

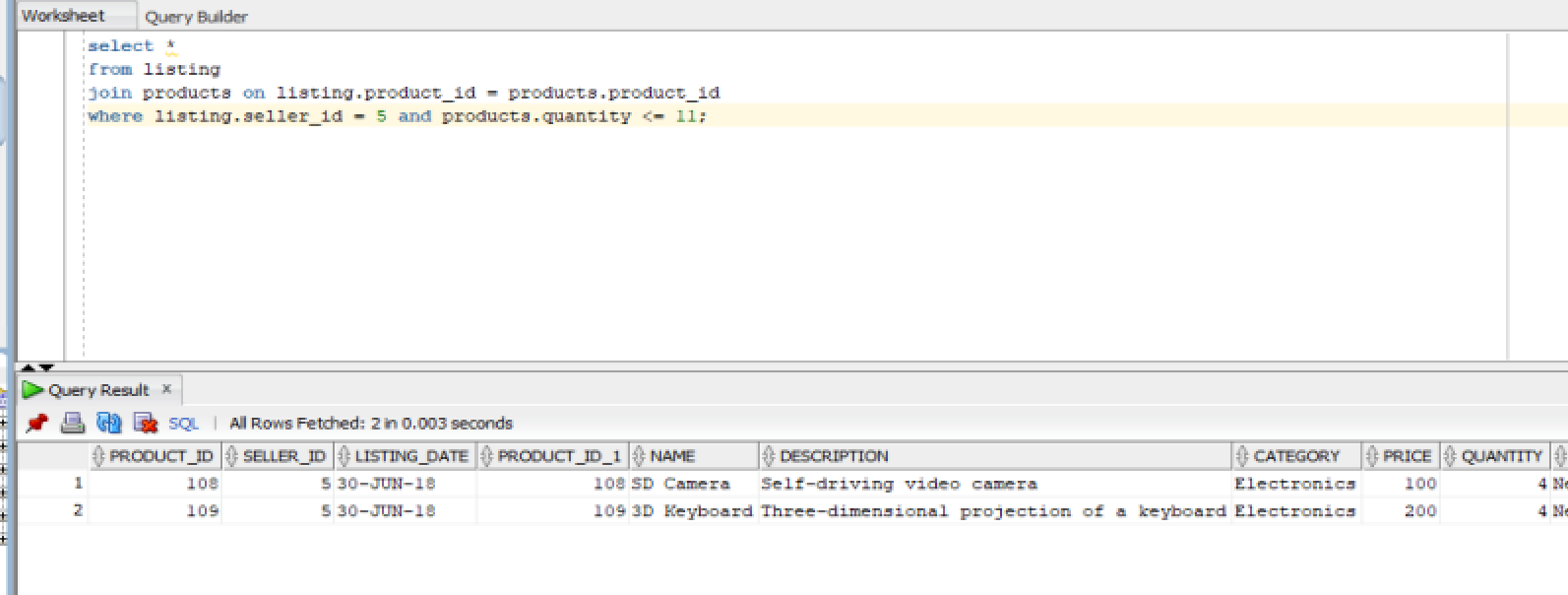
As address comes out to be the dependencies for several entities, so the physical ERD can be normalized into by adding address table to the ERD below.

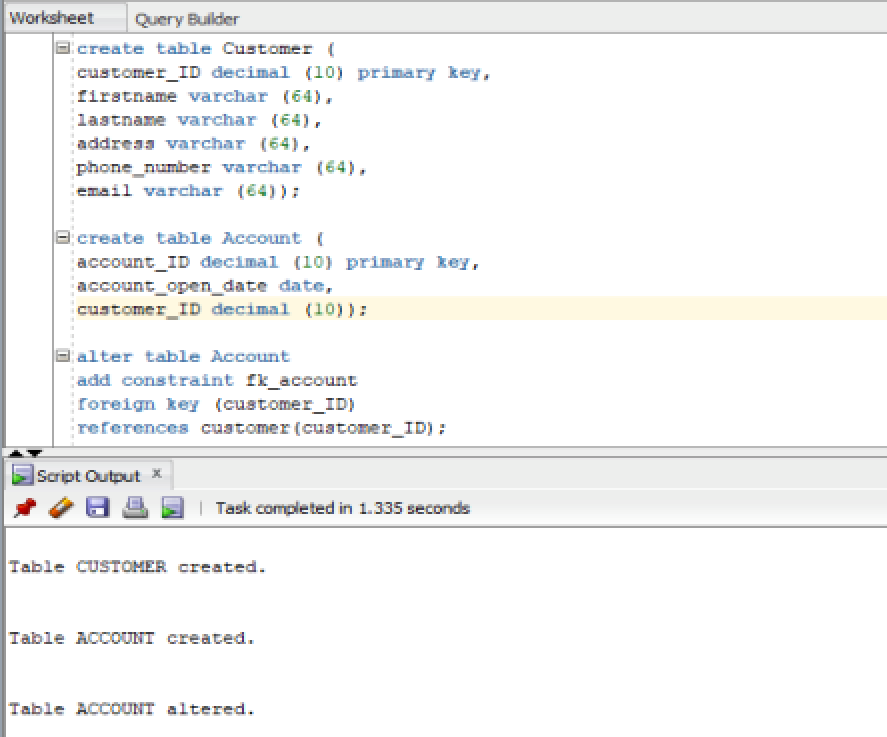


4. Create and execute the SQL for Aspect 2. Provide screenshots.

1. Tables and constraints.
2. Reusable stored procedure.
3. Use of the stored procedure.
4. SQL query.

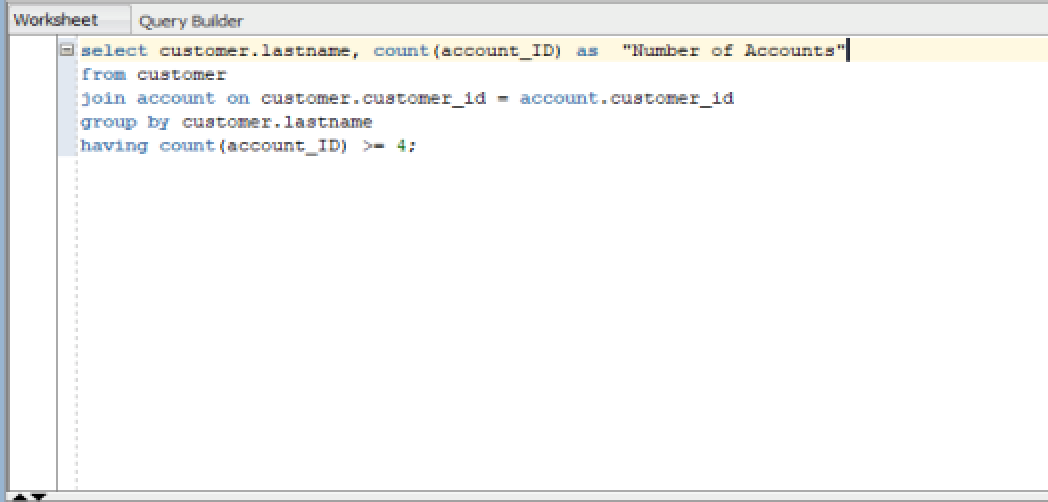






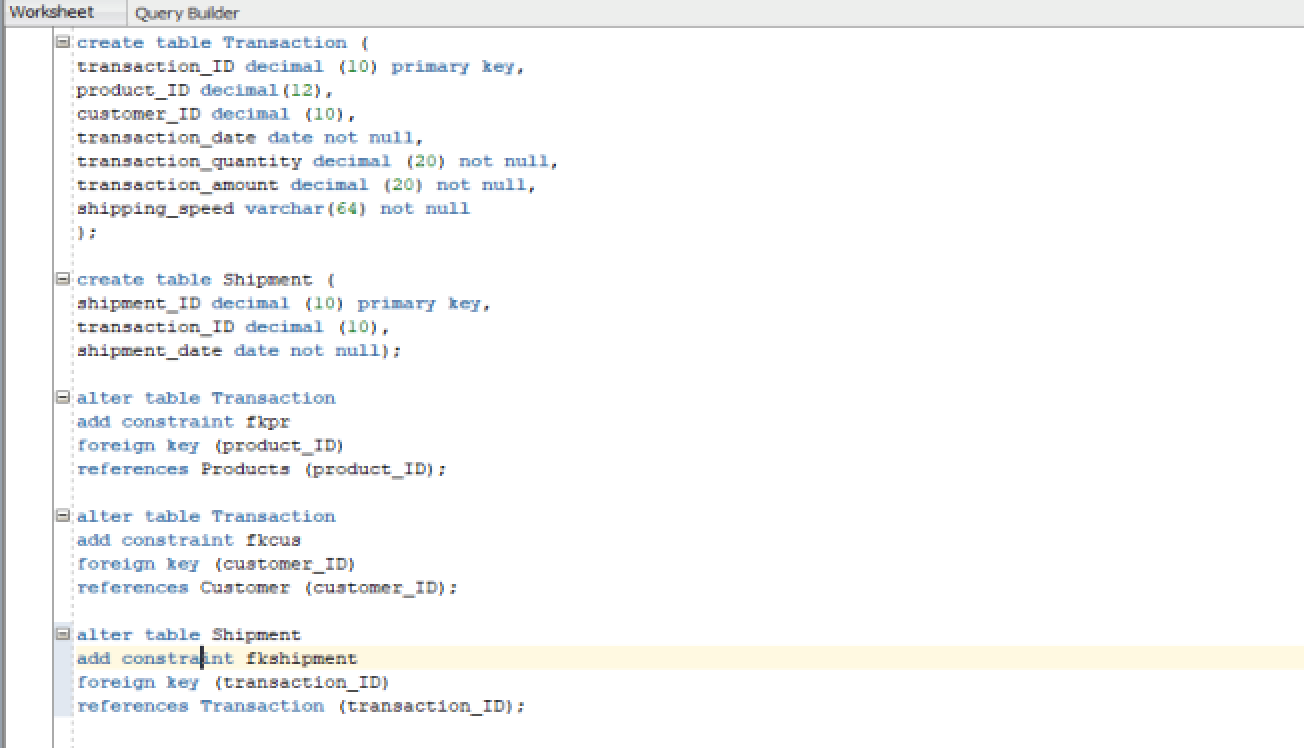


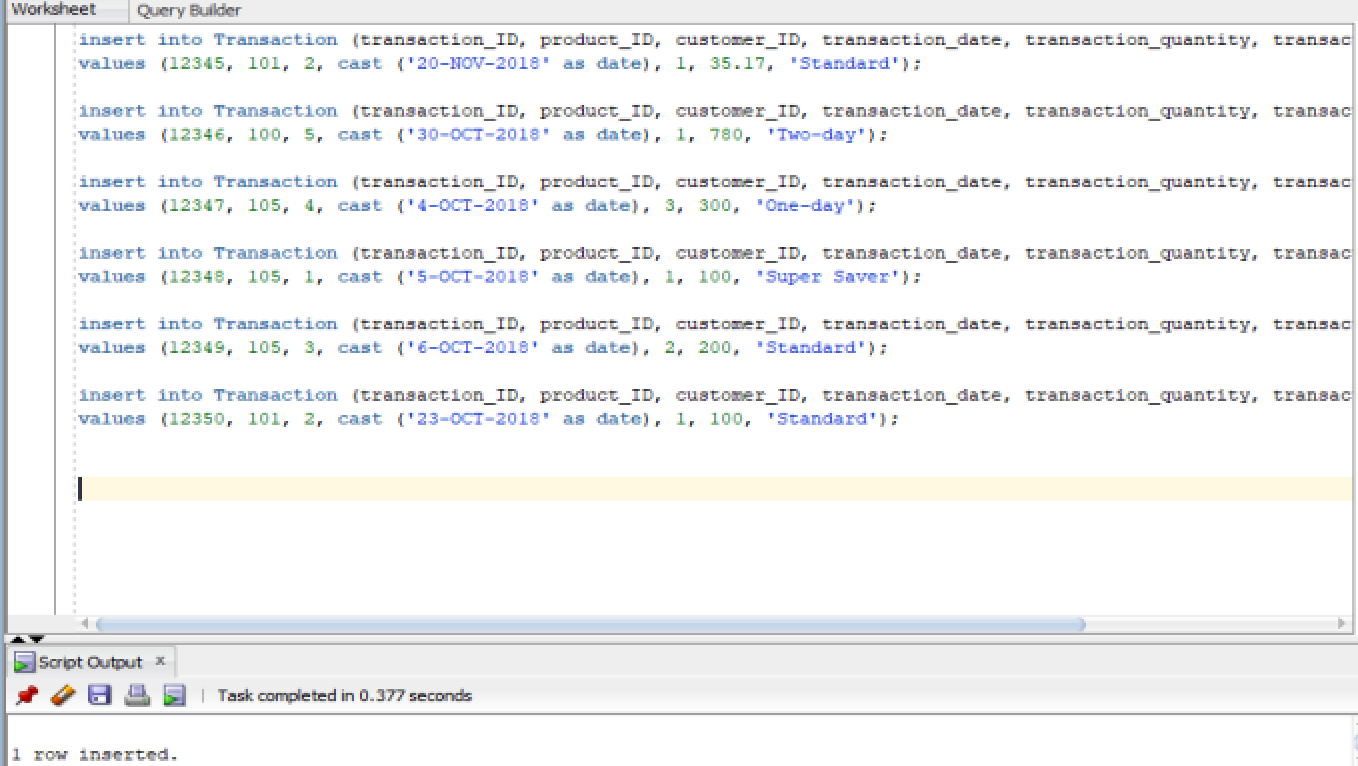


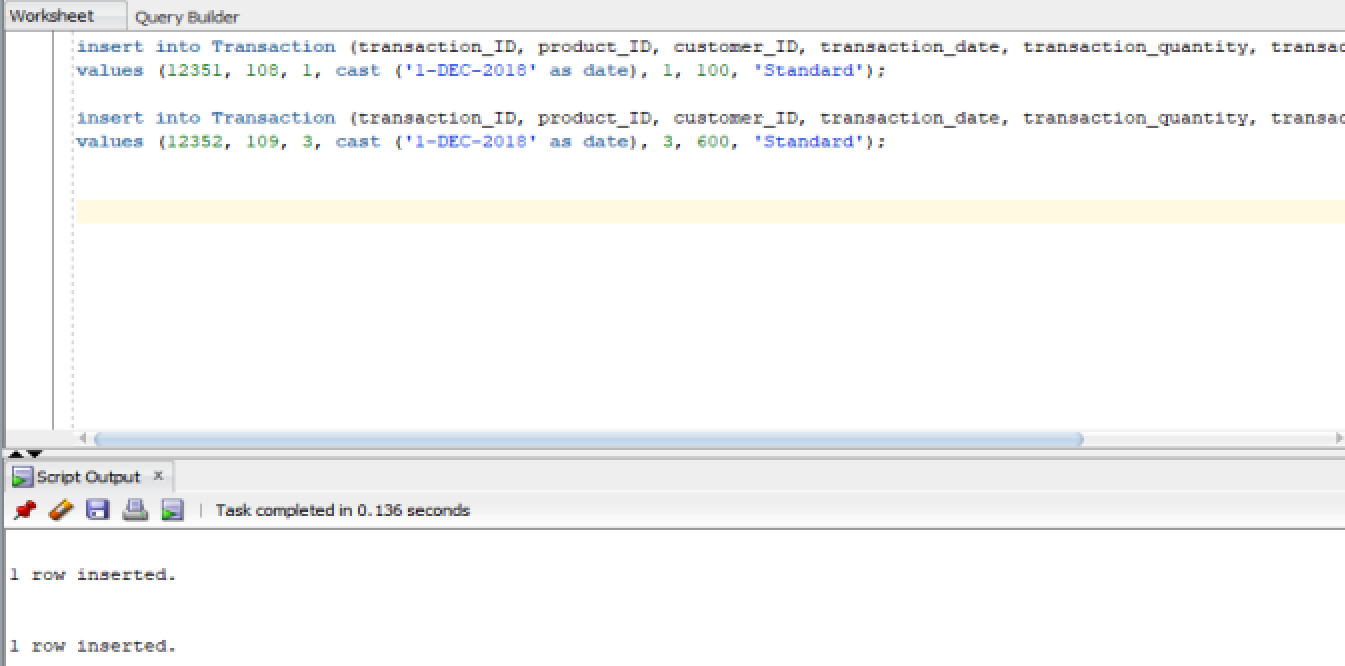


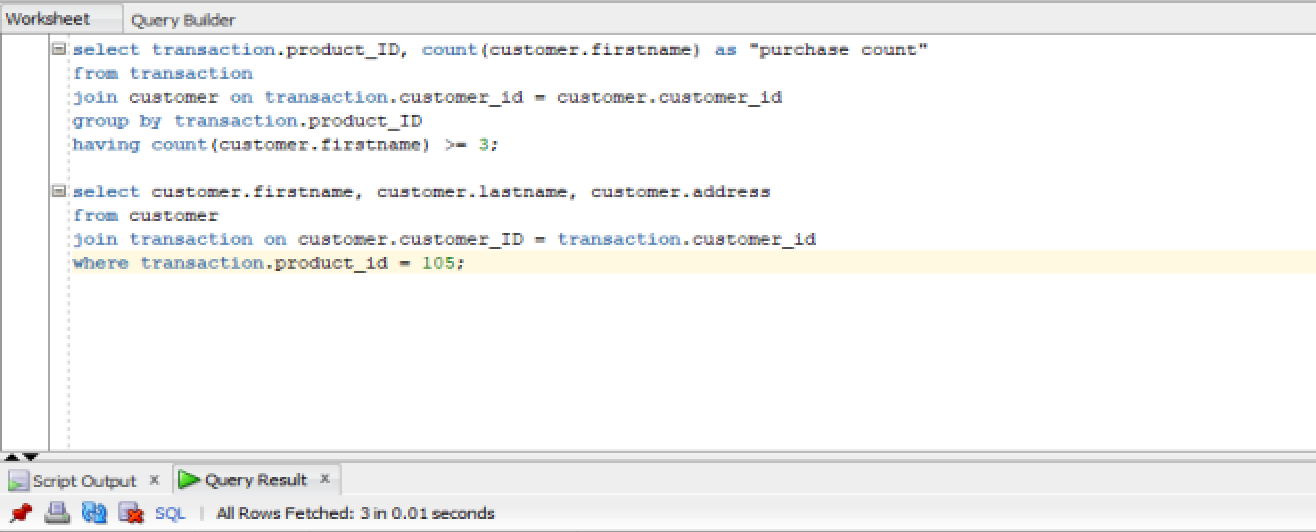
5. Create and execute the SQL for Aspect 3. Provide screenshots.

1. Tables and constraints.
2. Reusable stored procedure.
3. Use of the stored procedure.
4. SQL query.

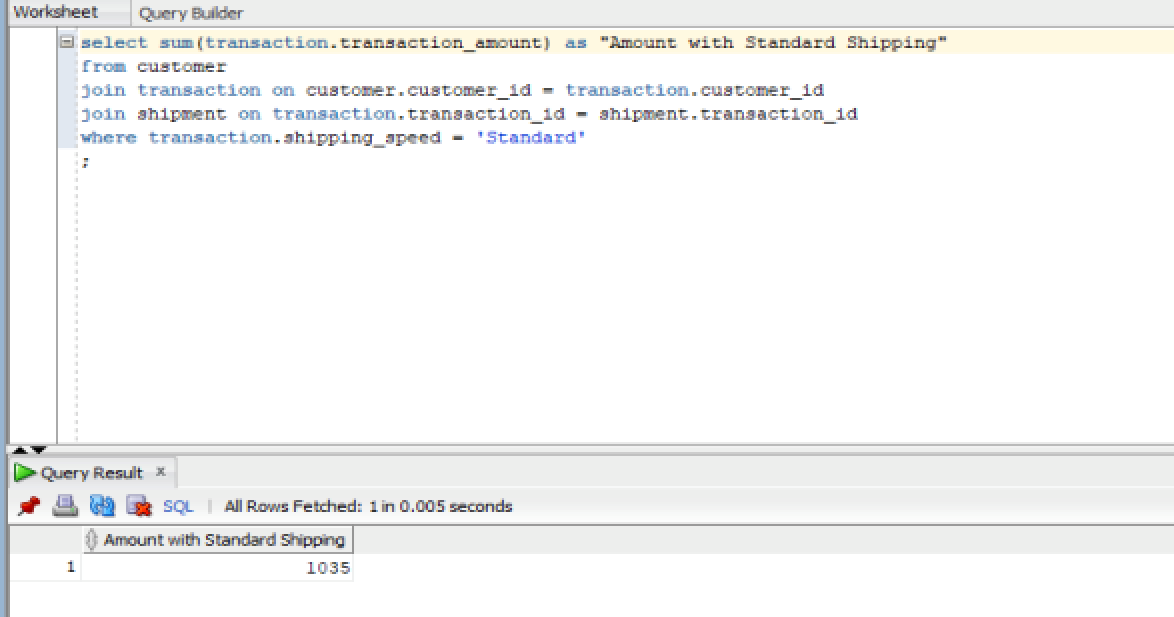












**SUMMARY:**

My database is for an e-commerce shopping website and app named Amazon which sells millions of the product online through its website and application. This contains the following tables with respective fields.

Product

Customer

Transaction

Seller

Listing

Shipment

Account

Vendors

Address

This iteration is not your final term project submission. You have the opportunity to modify all sections of your term project before the final submission at the end of the course.

Use the **Ask your Facilitator Discussion Board** if you have any questions regarding the how to approach this assignment.

Save your assignment as ***lastnameFirstname\_iteration5.doc*** and submit it in the *Assignments* section of the course.

For help uploading files please refer to the *Technical Support* page in the syllabus.