Homework Assignment #2 Solutions

1.

a) CQL Query:

```cql
DESCRIBE "Hoofers";
```

b) Result:

```cql
CREATE KEYSPACE "Hoofers" WITH replication = {'class': 'NetworkTopologyStrategy', 'us-west-2': '3'} AND durable_writes = true;
CREATE TABLE "Hoofers".boats (    bid int PRIMARY KEY,    bname text,    color text ) WITH additional_write_policy = '99PERCENTILE'    AND bloom_filter_fp_chance = 0.01    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}    AND comment = ''    AND compaction = {'class': 'org.apache.cassandra.db.compaction.UnifiedCompactionStrategy', 'log_all': 'true', 'num_shards': '128'}    AND compression = {'chunk_length_in_kb': 64, 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}    AND crc_check_chance = 1.0    AND default_time_to_live = 0    AND gc_grace_seconds = 864000    AND max_index_interval = 2048    AND memtable_flush_period_in_ms = 0    AND min_index_interval = 128    AND read_repair = 'BLOCKING'    AND speculative_retry = '99PERCENTILE';
```

c) Answers:

3 copies. us-west-2 region. 2 for writes and 2 for reads.

2.

a) CQL CREATE Statements:
USE "ShopALot";

CREATE TABLE "Customers" (  
user_id text,  
PRIMARY KEY (user_id)  
);

CREATE TABLE "Orders" (  
order_id text,  
total_price decimal,  
time_placed timestamp,  
pickup_time timestamp,  
customer_id text,  
shopper_id text,  
vehicle_state text,  
vehicle_license_plate text,  
store_id text,  
time_fulfilled timestamp,  
PRIMARY KEY (order_id)  
);

CREATE TABLE "OrderItems" (  
item_id text,  
qty int,  
selling_price decimal,  
order_id text,  
product_id text,  
PRIMARY KEY (item_id, order_id),  
);

CREATE TABLE "Products" (  
product_id text,  
category text,  
name text,  
description text,  
list_price decimal,  
PRIMARY KEY (product_id)  
);
a) PostgreSQL COPY commands:

```sql
COPY "ShopALot"."Products" TO 'PATH/Products.csv' DELIMITER ',' CSV HEADER;
COPY "ShopALot"."Customers" TO 'PATH/Customers.csv' DELIMITER ',' CSV HEADER;
COPY "ShopALot"."Orders" TO 'PATH/Orders.csv' DELIMITER ',' CSV HEADER;
COPY "ShopALot"."OrderItems" TO 'PATH/OrderItems.csv' DELIMITER ',' CSV HEADER;
```

4.

a) First CQL Query:

```sql
SELECT name, list_price
FROM "Products"
WHERE category = 'Meat & Seafood'
LIMIT 10;
```

b) Result:

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
```

c) Modified CQL Query:

```sql
SELECT name, list_price
FROM "Products"
WHERE category = 'Meat & Seafood'
LIMIT 10
ALLOW FILTERING;
```

b) Result:

```
name                  | list_price
------------------------------------------+-------------
USDA Choice Beef Top Loin New York Strip Steak Bone In Value Pack - 3.5 Lbs. (approx. weight) | 34.97
waterfront BISTRO Salmon Fillets Wild Alaskan Pink Boneless & Skin On - 32 Oz | null
Signature Farms Boneless Skinless Chicken Thighs Value Pack - 3 Lbs. | 13.47
Aidells Smoked Chicken Sausage Links Chicken & Apple 4 Count - 12 Oz | 6.49
```
5.
a) CQL Create Statement:
```sql
CREATE TABLE "ProductsCategory" (
    product_id text,
    category text,
    name text,
    description text,
    list_price decimal,
    PRIMARY KEY (category, product_id)
);
```

b) CQL Query:
```sql
SELECT name, list_price
FROM "ProductsCategory"
WHERE category = 'Meat & Seafood'
LIMIT 10;
```

c) Result:
```
<table>
<thead>
<tr>
<th>name</th>
<th>list_price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Beef 80% Lean 20% Fat - 1.25 Lbs</td>
<td>4.99</td>
</tr>
<tr>
<td>USDA Choice Beef Filet Mignon Steak Tenderloin - 1 Lb.</td>
<td>19.99</td>
</tr>
<tr>
<td>Jimmy Dean Fully Cooked Turkey Sausage Links 12 Count - 9.6 Oz</td>
<td>5.99</td>
</tr>
<tr>
<td>Seafood Counter Fish Salmon Fresh Atlantic Salmon Fillet Color Added - 1.00 LB</td>
<td>9.99</td>
</tr>
</tbody>
</table>
```

(10 rows)
A) CQL Query:

```
SELECT name, list_price
FROM "Products"
WHERE category = 'Meat & Seafood'
ORDER BY list_price DESC
LIMIT 10
ALLOW FILTERING;
```

b) CQL CREATE Statement:

```
CREATE TABLE "ProductsCategory" (  
  product_id text,
  category text,
  name text,
  description text,
  list_price decimal,
  PRIMARY KEY (category, list_price, product_id)
)
WITH CLUSTERING ORDER BY (list_price DESC, product_id DESC);
```
c) Results:

<table>
<thead>
<tr>
<th>name</th>
<th>list_price</th>
</tr>
</thead>
<tbody>
<tr>
<td>USDA Choice Beef Ribeye Roast Bone In - 6 Lbs. (approx. weight)</td>
<td>59.94</td>
</tr>
<tr>
<td>USDA Choice Beef Top Loin New York Strip Steak Bone In Value Pack - 3.5 Lbs. (approx. weight)</td>
<td>34.97</td>
</tr>
<tr>
<td>Signature Farms Beef Corned Beef Brisket Flat Cut - 3.50 LB</td>
<td>22.72</td>
</tr>
<tr>
<td>USDA Choice Beef Filet Mignon Steak Tenderloin - 1 Lb.</td>
<td>19.99</td>
</tr>
<tr>
<td>USDA Choice Beef Boneless Chuck Roast - 3 Lbs.</td>
<td>17.97</td>
</tr>
<tr>
<td>Signature Farms Boneless Skinless Chicken Thighs Value Pack - 3 Lbs.</td>
<td>13.47</td>
</tr>
<tr>
<td>Signature Farms Boneless Skinless Chicken Breasts Value Pack - 3 Lbs.</td>
<td>13.47</td>
</tr>
<tr>
<td>Open Nature Frozen Boneless Skinless Chicken Breasts - 36 Oz.</td>
<td>11.99</td>
</tr>
<tr>
<td>Seafood Counter Fish Salmon Fresh Atlantic Salmon Fillet Color Added - 1.00 LB</td>
<td>9.99</td>
</tr>
<tr>
<td>Hormel Black Label Bacon Original - 16 Oz</td>
<td>8.99</td>
</tr>
</tbody>
</table>

(10 rows)

d) Explanation:

Data in Cassandra’s tables is sorted based on the clustering key. Cassandra does not support resorting once the table is created, because sorting a large table is very costly. Once we created the clustering key on list_price and product_id, the table is sorted based on the descending order of list price and product id. We can retrieve the data without additional sorting.

7.

a) CQL Create Statement:

```sql
CREATE TABLE "ordersq7a"
(customer_id text,
order_id text,
total_price decimal,
PRIMARY KEY(customer_id, total_price, order_id)
)
WITH CLUSTERING ORDER BY (total_price DESC, order_id DESC);
```
b) CQL Create Statement:

```sql
CREATE TABLE "ordersitemsq7b"
(
shopper_id text,
order_id text,
item_id text,
PRIMARY KEY(shopper_id, order_id, item_id)
);
```

c) CQL Create Statement:

```sql
CREATE TABLE "ordersproductsq7c"
(
order_id text,
list_price decimal,
item_id text,
category text,
item_name text,
PRIMARY KEY(order_id, list_price, item_id)
);
```

d) CQL Create Statement:

```sql
CREATE TABLE "ordersq7d"
(
customer_id text,
time_placed timestamp,
order_id text,
total_price decimal,
PRIMARY KEY(customer_id, time_placed, order_id)
);
```

8. a) CQL Query:

```sql
SELECT order_id FROM "ordersq7a" WHERE customer_id = '24590' ORDER BY total_price DESC;
```

- Result:

  order_id
b) CQL Query:

```sql
SELECT order_id, COUNT(item_id) AS num_items FROM "ordersitemsq7b" WHERE shopper_id = '0JKLY' GROUP BY order_id;
```

- Result:

<table>
<thead>
<tr>
<th>order_id</th>
<th>num_items</th>
</tr>
</thead>
<tbody>
<tr>
<td>07DYO</td>
<td>1</td>
</tr>
<tr>
<td>9MLF9</td>
<td>2</td>
</tr>
<tr>
<td>A3BRA</td>
<td>1</td>
</tr>
<tr>
<td>NHOKA</td>
<td>1</td>
</tr>
<tr>
<td>SJ097</td>
<td>1</td>
</tr>
</tbody>
</table>

c) CQL Query:

```sql
SELECT item_name, category FROM "ordersproductsq7c" WHERE order_id = '005SN' ORDER BY list_price ASC;
```

- Result:

<table>
<thead>
<tr>
<th>item_name</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Organic Milk Reduced Fat 2% - Half Gallon</td>
<td>Dairy &amp; Eggs &amp; Cheese</td>
</tr>
<tr>
<td>Larabar Food Bar Apple Pie - 5-1.6 Oz</td>
<td>Personal Care &amp; Health</td>
</tr>
</tbody>
</table>
d)  

- CQL Query:

```sql
SELECT sum(total_price) as total_amount FROM "ordersq7d" WHERE customer_id = '32976' AND time_placed > '2020-03-01' AND time_placed < '2020-09-01';
```

- Result:

```
   total_amount
------------------
     614.47
```

9.  

a) CQL INSERT statements:

```sql
--Inserting into the Orders base table
INSERT INTO "Orders"(order_id, total_price, time_placed, pickup_time, customer_id, shopper_id, vehicle_state, vehicle_license_plate, store_id, time_fulfilled) VALUES('12MDAE', 7.14, '2021-04-10T19:30:24.000Z', '2021-04-10T21:01:45.000Z', '24590', 'MQD30', 'CA', 'AKM 554', 'A72NF', '2021-04-10T23:20:56.000Z');
--Inserting into the OrderItems base table
INSERT INTO "OrderItems"(item_id, order_id, product_id, qty, selling_price) VALUES('9B317', '12MDAE', 'GMGO5', 3, 2.38);
--Inserting into the 7a table
INSERT INTO "ordersq7a"(customer_id, total_price, order_id) VALUES('24590', 7.14, '12MDAE');
--Inserting into the 7b table
INSERT INTO "ordersitemsq7b"(shopper_id, order_id, item_id) VALUES('MQD30', '12MDAE', '9B317');
--Inserting into the 7c table
INSERT INTO "ordersproductsq7c"(order_id, list_price, item_id, category, item_name) VALUES('12MDAE', 1.59, '9B317', 'Canned Good & Soups', 'Green Giant Corn Whole Kernel Sweet - 15.25 Oz');
--Inserting into the 7d table
INSERT INTO "ordersq7d"(customer_id, time_placed, order_id, total_price) VALUES('24590', '2021-04-10T19:30:24.000Z', '12MDAE', 7.14);
```