



BUSINESS ANALYSIS

VIDHI GANDHI

MSBA FALL 2019

DATABASE MANAGEMENT SYSTEMS

DATE: 02/16/2020

## TABLE OF CONTENTS

<b>INTRODUCTION.....</b>	<b>3</b>
SWIM LANE DIAGRAM.....	3
BUSINESS SCENARIO.....	3
<i>Parsing Scenario to identify key information.....</i>	4
BUSINESS ANALYSIS QUESTIONS .....	4
<b>SQL QUERIES.....</b>	<b>5</b>
LIST OF TABLES AND TABLE CREATION SCRIPT .....	5
VIEWS .....	8
WITHOUT JOINS .....	10
SUBQUERIES & JOINS .....	17
PROCEDURES.....	23
TRIGGERS .....	24
<b>TABLEAU.....</b>	<b>27</b>
<b>CONCLUSION .....</b>	<b>29</b>
FUTURE SCOPE .....	29

## Introduction

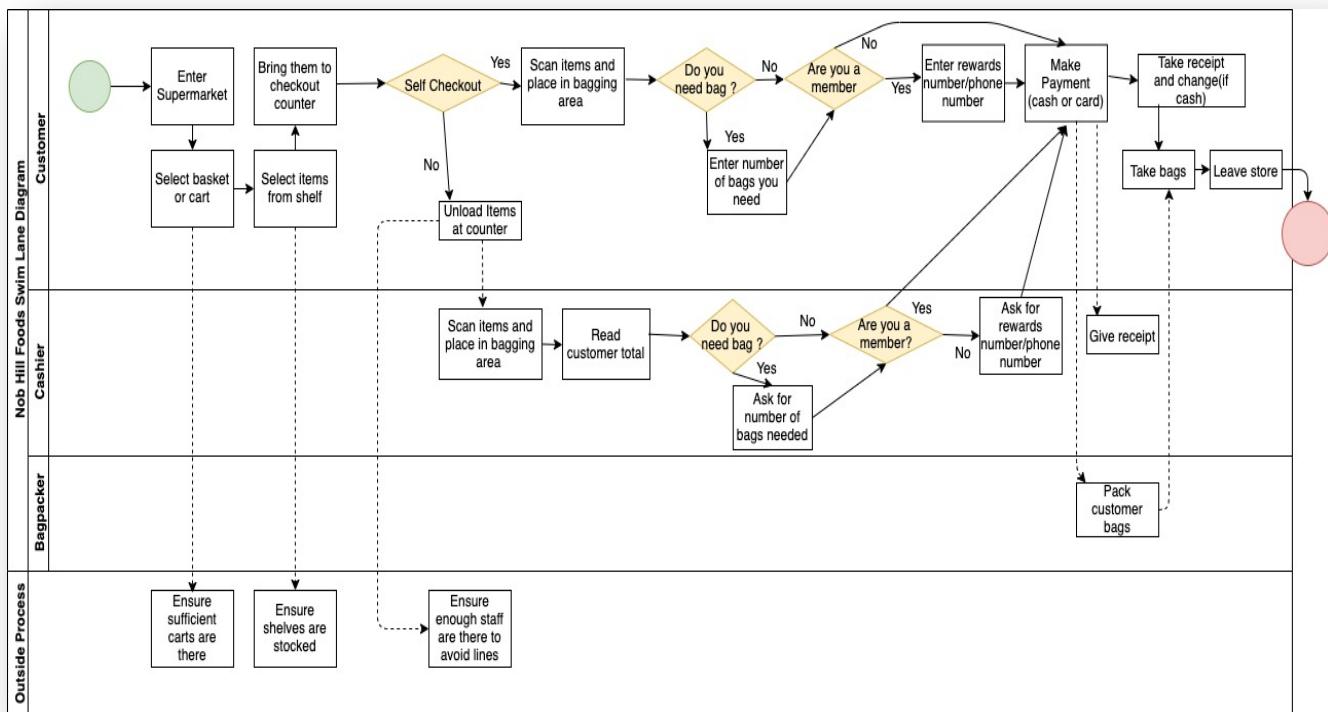
Nob Hills food is a supermarket store owned by Raley's Supermarkets. Raley's purchased Nob Hill Foods in 1997.

The grocery store industry is a multi-billion dollar industry. It touches everyone on society since we all eat. Through tough competition a grocery chain must run efficient and reduce cost overruns. This includes using updated information in its purchasing decisions, inventory control, store stocking, customer satisfaction, buying trends, and a host of other business concerns. The business of grocery stores is the ability to provide their customers with well-priced and fresh food items every day. This seemingly simple concept is very complex when put into practice. The company must source its inventory from the U.S. and foreign countries. They must buy at a competitive price then deliver the food items to the individual store quickly. They must handle food changes as the seasons change. They must provide quick checkout service to their customers and be able to keep stores stocked with high velocity items.

This project will focus on small aspect of the grocery enterprise simulating a customer buying items by selecting them on a form as if taking them from the shelves. They can then see their total and finish the transaction.

### Swim Lane Diagram

Swim lane diagrams add an extra level of clarity about who does what to process flowcharts. A swim lane diagram is a type of flowchart that delineates who does what in a process. Below is the swim lane diagram for Nob Hill food:



### **Scenario**

John and Jane walk into a Nob Hill store near Monroe Street at 5pm. They both pick up baskets and start looking for their respective items. John wants to buy milk and bread and goes to the respective shelves to buy it. Jane wants to buy spinach and banana and goes to the vegetable section to buy it. Post selecting their items, they proceed towards the checkout counter. John proceeds for normal checkout while Jane opts for self-checkout. John goes to the counter where the cashier was Amy. Amy scans both the items that John wants to buy and puts it in the baggage section. Amy then asks John if he needs a bag; John says yes and Amy charges him for the additional bag. Next Amy checks with John if he is an existing member of Nob Hill rewards; John says no. Amy then asked John how he intends to, John said card. Amy takes his card and post payment returns his card along with the receipt. Rob the bag packer packed Jack's items in the bag and handed the bag to John. John left the store at 5:15pm. In the meantime, Jane was at the self-checkout counter. She scanned the items and placed them in the baggage section. Screen prompted her if she needed any bag; she pressed no. Then she was prompted for her rewards number. She was a member, and she entered her phone number on the card reader. She then completed her payment and took her receipt. She collected her items from the baggage area and left the store at 5:10 pm.

### **Parsing Scenario to identify key information**

- Stakeholders: John, Jane (Customer), Amy (Cashier)
- Products: Milk, bread, spinach, banana
- Infrastructure: Scanner, Computer, Card Reader, Receipt Printer
- Additional Information: Location/Address
- Timeline: Entered at 5 PM. Checkout ~ 5.10/5 15 pm

### **Business Analysis Questions:**

1. What is total sales for this store (per day/month/quarter/year)?
2. What is average sale per transaction (for a given day)?
3. What is business hour of the day by number of transactions?
4. What products are sold most?
5. How many transactions are cash vs credit?
6. Which is the preferred method of checkout self or regular?
7. Are there more member transactions or non-member?
8. What is the count of customer having maximum purchase of 20\$?
9. What is the count of customer having purchase greater than 100\$?
10. What is the product which is least sold?
11. What is the product which is most sold?
12. What time and date were the most number of transactions?

## SQL QUERIES

### Tables and table creation:

Screenshots showing output for (Select \* from <table>) for each table in the schema:

- Customer Table

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for Home, Worksheet (which is selected), and Data Modeler. The left sidebar shows a Navigator with 'ADMIN' selected, and a 'Tables' section containing 'CHECKOUT', 'CUSTOMER', 'EMPLOYEES', 'INVENTORY', 'PRODUCT', and several temporary tables starting with 'SDW\$ERR\$'. The central workspace displays a SQL worksheet with the following code:

```
1 CREATE TABLE Customer
2 (
3     CustID CHAR(11) NOT NULL,
4     Cust_Type VARCHAR(20) NOT NULL,
5     CONSTRAINT PKCustID PRIMARY KEY (CustID)
6 );
7
8 select * from customer;
9
10
11
12
```

Below the code, the 'Query Result' tab is active, showing a table with two columns: 'custid' and 'cust\_type'. The data is as follows:

	custid	cust_type
1	50	M
2	51	NM
3	52	NM
4	53	M
5	58	NM

At the bottom of the worksheet, it says '2:03:20 AM - 10 rows total'.

- CustID - auto-generated integer ID.
- Cust\_Type – Details about the customer being member(M) or non-member(NM).
- Primary key: CustID
- Employees Table

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for Home, Worksheet (which is selected), Data Modeler, and Admin. The left sidebar has sections for Navigator, Worksheets, and Tables, with ADMIN selected. The main area displays a worksheet with the following SQL code:

```

10
11 CREATE TABLE Employees
12 (
13     EmpID CHAR(11) NOT NULL,
14     CONSTRAINT PKEmpID PRIMARY KEY (EmpID)
15 );
16
17 select * from employees;
18
19
20
21

```

Below the code, the Query Result tab is active, showing the results of the SELECT statement:

	empid
1	1
2	2
3	3
4	4
5	5

At the bottom of the interface, status information indicates 1 row selected, 0 rows inserted, 0 rows updated, and 10 rows total.

- EmpID - auto-generated integer ID.

➤ Primary key: EmpID

- Product Table

The screenshot shows the Oracle SQL Developer interface with a different connection named Nob\_Hill. The top navigation bar includes tabs for Home, Worksheet (selected), Data Modeler, and Admin. The left sidebar shows the Tables section with the Product table selected. The main area displays a worksheet with the following SQL code:

```

25
26 CREATE TABLE Product
27 (
28     ItemID CHAR(11) NOT NULL,
29     Item_Name VARCHAR(20) NOT NULL,
30     Price DECIMAL(5,2) NOT NULL,
31     Brand VARCHAR(20) NOT Null,
32     CONSTRAINT PKItemID PRIMARY KEY (ItemID)
33 );
34
35
36
37
38
39
39

```

Below the code, the Query Result tab is active, showing the results of the SELECT statement:

	itemid	item_name	price	brand
1	12	Cookies	2.25	Nabisco
2	658	Cigarettes	5	PhillipMorris
3	4587	Cheese	6	Kraft
4	2365	Cereal	1.99	Kellogg
5	84854	Oatmeal	2.5	Quaker
6	3521	Crackers	4	Nabisco

- ItemID - auto-generated integer ID.

- Item\_Name – Name of the item

- Price – Cost of the item (decimal values)
- Brand – Brand name of the product
- Primary key: ItemID

- Inventory

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for Home, Worksheet, Data Modeler, and Admin. The Worksheet tab is active, displaying a query window with the following SQL code:

```

53
54
55
56 CREATE TABLE Inventory
57 (
58   Inventory_Quantity CHAR(11) NOT NULL,
59   FK1ItemID CHAR(11) NOT NULL,
60   CONSTRAINT PKFK1ItemID PRIMARY KEY (FK1ItemID)
61 );
62
63 select * from inventory;
64

```

The Query Result pane below shows the data returned by the last query:

	inventory_quantity	fk1itemid
1	10	12
2	10	658
3	4	4587
4	23	2365
5	38	84854

At the bottom of the interface, status information indicates 1 row selected, 0 rows inserted, 0 rows updated, and 0 rows deleted, with a total execution time of 2:19:34 AM.

- FK1ItemID - auto-generated integer ID; FK from Items entity
- Inventory\_Quantity – Quantity of items in inventory

- Checkout

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for Home, Worksheet, Data Modeler, and Admin. The Worksheet tab is active, displaying a query window with the following SQL code:

```

36 CREATE TABLE Checkout
37 (
38   Transaction_ID CHAR(11) NOT NULL,
39   Quantity CHAR(11) NOT NULL,
40   Total DECIMAL(5,2) NOT NULL,
41   Checkout_Date VARCHAR(10) NOT NULL,
42   Checkout_Time VARCHAR(10) NOT NULL,
43   ItemID CHAR(11) NOT NULL,
44   EmpID CHAR(11) NOT NULL,
45   CustID CHAR(11) NOT NULL,
46   Checkout_Type VARCHAR(20) NOT NULL,
47   Mode_of_payment VARCHAR(20) NOT NULL,
48   CONSTRAINT PKTransaction_ID PRIMARY KEY (Transaction_ID),
49   CONSTRAINT FK1 FOREIGN KEY (ItemID) References Product (itemID),
50   CONSTRAINT FK2 FOREIGN KEY (CustID) References Customer (CustID),
51   CONSTRAINT FK3 FOREIGN KEY (EmpID) References Employees (EmpID)
52 );
53
54 Select * From Checkout;
55

```

The Query Result pane below shows the data returned by the last query:

	transaction_id	quantity	total	checkout_date	checkout_time	itemid
1	201	5	11.25	06/09/19 12:00...	2019-06-10T00:...	12
2	32	4	20	06/08/19 12:00...	2019-06-09T00:...	658
3	658	2	10	06/11/19 12:00...	2019-06-10T00:...	1507

At the bottom of the interface, status information indicates 1 row selected, 0 rows inserted, 0 rows updated, and 0 rows deleted, with a total execution time of 2:20:27 AM.

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes Home, Worksheet (which is selected), Data Modeler, and Admin. The left sidebar has sections for Navigator, Worksheets, and Tables, with ADMIN selected. Under Tables, there are several entries including CHECKOUT, CUSTOMER, EMPLOYEES, INVENTORY, PRODUCT, and several SDW\$ERRS\_\* tables. A search bar is also present.

The main workspace contains two tabs: [Worksheet]\* and Query Result. The Worksheet tab displays the following SQL code:

```

36 CREATE TABLE Checkout
37 {
38   Transaction_ID CHAR(11) NOT NULL,
39   Quantity CHAR(11) NOT NULL,
40   Total DECIMAL(5,2) NOT NULL,
41   Checkout_Date VARCHAR(10) NOT NULL,
42   Checkout_Time VARCHAR(10) NOT NULL,
43   ItemID CHAR(11) NOT NULL,
44   EmpID CHAR(11) NOT NULL,
45   CustID CHAR(11) NOT NULL,
46   Checkout_Type VARCHAR(20) NOT NULL,
47   Mode_of_payment VARCHAR(20) NOT NULL,
48   CONSTRAINT PKTransaction_ID PRIMARY KEY (Transaction_ID),
49   CONSTRAINT FK1 FOREIGN KEY (ItemID) References Product (itemID),
50   CONSTRAINT FK2 FOREIGN KEY (CustID) References Customer (CustID),
51   CONSTRAINT FK3 FOREIGN KEY (EmpID) References Employees (EmpID)
52 };
53
54 Select * From Checkout;

```

The Query Result tab shows the following data:

	_time	itemid	empid	custid	checkout_type	mode_of_payment
1	-10T00:...	12	2	50	N	Card
2	-09T00:...	658	2	51	N	Card

At the bottom of the interface, status information includes: ✘ 1 ▲ 0 ⏪ 0 | 2:20:27 AM - 10 rows total.

- EmpID - auto-generated integer ID.
- Transaction\_ID - auto-generated integer ID
- Cust\_ID - unique integer number identifying each customer – FK from CUSTOMER entity
- Emp\_ID - unique integer number identifying each store – FK from EMPLOYEE entity
- Total – double with two digit precision
- Date – Date type
- Time – Time type
- Checkout\_Type: Self-checkout(S) or Normal checkout(N).
- Mode\_of\_payment: Card or cash

### SQL Views:

1. A view that finds the employee who has the customer with the highest order of a day and prints the employee id, customer id and total of that checkout..

ORACLE SQL Developer

```

121 Select * from nameorders;
122
123
124 -- a view that finds the employee who has the customer with the highest order of a day.
125 CREATE VIEW topemployee
126 AS SELECT b.checkout_date, a.empid,b.custid,b.total
127 FROM employees a,checkout b
128 WHERE a.empid = b.empid
129 AND b.total =
130   (SELECT MAX (total)
131    FROM checkout c
132    WHERE c.checkout_date = b.checkout_date);
133 Select * from topemployee;
134
135

```

	checkout_date	empid	custid	total
7	12/24/19	7	58	2.5
8	1/2/20	1	105	7.98
9	1/10/20	3	179	8.97
10	6/9/19	2	50	20

2. A view that shows for each order the salesman id and customer id along with transaction id and total.

ORACLE SQL Developer

```

113
114
115
116
117 -- a view that shows for each order the salesman id and customer id alongwith transacion id and total.
118 CREATE VIEW ordersummary1
119 AS SELECT Transaction_ID,a.CustID, total, a.empid
120 FROM checkout a, customer c, employees e
121 WHERE a.custid = c.custid
122 AND a.empid = e.empid;
123
124 Select * from ordersummary1;
125
126
127

```

	transaction_id	custid	total	empid
1	201	50	11.25	2
2	32	51	20	2
3	6589	52	18	3
4	2147	99	7.96	4

3. A view that shows the number of orders in each day.

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for 'ORACLE SQL Developer', 'Home', 'Worksheet' (which is selected), and 'Data Modeler'. On the right, there are user and help icons. The main area has a 'Navigator' pane on the left containing a 'Tables' section with a search bar and a list of tables: CHECKOUT, CUSTOMER, EMPLOYEES, INVENTORY, MAIN, PRODUCT, and several SDW\$ERR\$\_\* tables. The central workspace displays a worksheet titled 'Nob\_Hill\*' with the following code:

```

143
144
145
146
147 -- a view that shows the number of orders in each day.
148 CREATE VIEW dateord(checkout_date, odcount)
149 AS SELECT checkout_date, COUNT (*)
150 FROM checkout
151 GROUP BY checkout_date;
152 select * from dateord;
153
154
155

```

Below the code is a 'Query Result' tab showing the output of the query:

	checkout_date	odcount
1	6/9/19	2
2	11/5/19	1
3	6/8/19	1
4	8/11/19	2
5	1/10/20	1

### SQL Queries without Joins:

#### 1. How many transactions are cash vs credit?

The screenshot shows the Oracle SQL Developer interface, similar to the previous one but with a different worksheet name. The top navigation bar includes tabs for 'ORACLE SQL Developer', 'Home', 'Worksheet' (selected), and 'Data Modeler'. The 'Navigator' pane on the left shows the same table list. The central workspace displays a worksheet titled 'Nob\_Hill\*' with the following code:

```

163
164
165
166
167 -- transaction preference= cash or card
168 Select Count(transaction_id) AS Card
169 From checkout
170 where Mode_of_payment = 'Card';
171
172
173
174

```

Below the code is a 'Query Result' tab showing the output of the query:

	card
1	7

ORACLE SQL Developer Worksheet

```

165
166
167 -- transaction preference- cash or card
168 Select Count(transaction_id) AS Card
169 From checkout
170 where Mode_of_payment = 'Card';
171
172 Select Count(transaction_id) AS Cash
173 From checkout
174 where Mode_of_payment = 'Cash';
175
176

```

	cash
1	7

2. Which is the preferred method of checkout self or regular?

ORACLE SQL Developer Worksheet

```

153
154
155 -- count of normal and self checkout
156 Select Count(transaction_id) AS NormalCheckout
157 From checkout
158 where Checkout_Type = 'N';
159
160 Select Count(transaction_id) AS SelfCheckout
161 From checkout
162 where Checkout_Type = 'S';
163
164

```

	normalcheckout
1	11

ORACLE SQL Developer Worksheet

```

153
154 -- count of normal and self checkout
155 Select Count(transaction_id) AS NormalCheckout
156 From checkout
157 where Checkout_Type = 'N';
158
159
160 Select Count(transaction_id) AS SelfCheckout
161 From checkout
162 where Checkout_Type = 'S';
163
164

```

Query Result

	selfcheckout
1	3

### 3. Are there more member transactions or non-member?

ORACLE SQL Developer Worksheet

```

175
176 -- query to check if there are more member or non-member transactions
177
178 Select customer.cust_type,Count(checkout.custid) AS Members
179 From checkout, customer
180 where checkout.custid = customer.custid
181 Group by customer.cust_type
182 having customer.Cust_Type = 'M';
183
184
185
186

```

Query Result

	cust_type	members
1	M	9

ORACLE SQL Developer

Worksheet

Nob\_Hill\*

```

177 -- query to check if there are more member or non-member transactions
178 Select customer.cust_type,Count(checkout.custid) AS Members
179 From checkout, customer
180 where checkout.custid = customer.custid
181 Group by customer.cust_type
182 having customer.Cust_Type = 'M';
183
184 Select customer.cust_type,Count(checkout.custid) AS NonMembers
185 From checkout, customer
186 where checkout.custid = customer.custid
187 Group by customer.cust_type
188 having customer.Cust_Type = 'NM';
189
190
191

```

Query Result

	cust_type	nonmembers
1	NM	5

4. Query to find highest purchase amount ordered by the each customer with their ID and highest purchase amount

ORACLE SQL Developer

Worksheet

Nob\_Hill\*

```

190
191
192
193
194 -- query to find highest purchase amount ordered by the each customer with their ID and highest purchase
amount
195 SELECT custid,MAX(total)
196 FROM checkout
197 GROUP BY custid;
198
199
200
201
202
203 |

```

Query Result

	custid	max(total)
6	50	20
7	51	20
8	105	7.98
9	99	7.96

5. What is the count of customer having maximum purchase of 20\$?

ORACLE SQL Developer

```

199 -- What is the count of customer having maximum purchase of 20$?
200 SELECT custid,MAX(total) AS TotalPurchase
201 FROM checkout
202 WHERE total >=20
203 GROUP BY custid;
204
205
206
207
208
209
210
211
212
213

```

	custid	totalpurachase
1	50	20
2	51	20

6. Query to find the total purchase amount ordered by the each customer on a particular date with their ID, order date and total purchase amount.

ORACLE SQL Developer

```

204 GROUP BY custid;
205
206
207 -- query to find the highest purchase amount ordered by the each customer on a particular date with their
208 ID, order date and highest purchase amount.
209 SELECT custid,checkout_date,sum(total)
210 FROM checkout
211 GROUP BY custid,checkout_date;
212
213
214
215
216
217
218

```

	custid	checkout_date	sum(total)
1	50	6/9/19	31.25
2	99	6/4/19	7.96
3	178	12/12/19	5.94
4	105	1/2/20	16.42

7. Query to find the number of products with a price more than or equal to 2\$.

ORACLE SQL Developer Worksheet

```

218 -- query to find the number of products with a price more than or equal to 2$.
219 SELECT COUNT(*) AS "Number of Products"
220   FROM product
221     WHERE price >= 2;
222
223
224
225
226
227
228
229

```

	number of products
1	7

8. Query to display the average price of each company's products, along with their name.

ORACLE SQL Developer Worksheet

```

227
228
229
230
231 -- query to display the average price of each company's products, along with their name.
232 SELECT AVG(price) AS "Average Price", brand AS "Brand Name"
233   FROM product
234 GROUP BY brand;
235
236
237
238

```

	average price	brand name
1	6	Kraft
2	0.99	HomeBrand
3	2.99	Kleenex
4	5	PhillipMorris
5	0.5	DelMonte

9. Which product is least sold?

ORACLE SQL Developer

Worksheet

Nob\_Hill\*

```

243
244
245
246 -- What is the product which is least sold?
247
248 Select itemid,max(quantity) AS QuantitySold
249 From checkout
250 Group by itemid
251 Order by max(quantity) ;
252
253
254

```

Query Result

	itemid	quantitysold
1	3521	1
2	84854	2
3	256	2
4	4587	3
5	145	3

### 10. What products are sold most?

ORACLE SQL Developer

Worksheet

Nob\_Hill\*

```

250 Group by itemid
251 Order by max(quantity) ;
252
253
254 -- What is the product which is most sold?
255 Select itemid,max(quantity) AS QuantitySold
256 From checkout
257 Group by itemid
258 Order by max(quantity) desc;
259
260
261

```

Query Result

	itemid	quantitysold
1	355	6
2	1566	5
3	12	5
4	658	4
5	2365	4

### 11. Which day was the highest sale?

ORACLE SQL Developer

```

259
260
261 -- which day was the highest sale
262 Select sum(total) AS TotalSale,checkout_date
263 From checkout
264 group by checkout_date
265 Order by sum(total) desc;
266
267
268
269
270

```

	totalsale	checkout_date
1	31.25	6/9/19
2	29.25	8/11/19
3	20	6/8/19
4	16.42	1/2/20
5	8.97	1/10/20

## 12. What time and date were the most number of transactions?

ORACLE SQL Developer

```

267
268 select checkout_time,checkout_date,count(transaction_id) AS NumberOfTransactions
269 from checkout
270 group by checkout_time,checkout_date
271 order by count(transaction_id) desc;
272
273
274
275

```

	checkout_time	checkout_date	numberoftransaction
1	17:40:00	1/2/20	3
2	13:10:00	8/11/19	2
3	19:38:00	6/9/19	2
4	13:20:00	11/5/19	1
5	16:10:00	12/24/19	1

⊗ 0 ▲ 0 ⚙ 0 | 9:49:06 PM - 10 rows total

## SQL SUBQUERIES & JOINS:

1. Query to display all the orders from the orders table issued by the salesman '2'.

ORACLE SQL Developer Home worksheet Data Modeler X ADMIN

Navigator Worksheets

ADMIN

Tables

Search... transaction\_id quantity total checkout\_date checkout\_time itemid

1	201	5	11.25	6/9/19	19:38:00	12
2	32	4	20	6/8/19	15:22:00	658
3	201	4	20	6/9/19	19:38:00	658

```

274 -- Write a query to display all the orders from the orders table issued by the salesman '2'.
275
276 SELECT *
277 FROM checkout
278 WHERE empid =
279   (SELECT empid
280    FROM employees
281    WHERE empid='2');
282
283
284
285

```

## 2. Query to find all the orders issued against the salesman who may works for customer whose id is 50.

ORACLE SQL Developer Home worksheet Data Modeler X ADMIN

Navigator Worksheets

ADMIN

Tables

Search... transaction\_id quantity total checkout\_date checkout\_time itemid

1	201	5	11.25	6/9/19	19:38:00	12
2	32	4	20	6/8/19	15:22:00	658
3	201	4	20	6/9/19	19:38:00	658

```

285 -- Write a query to find all the orders issued against the salesman who may works for customer whose id is
286 50.
287
288 SELECT *
289 FROM checkout
290 WHERE empid =
291   (SELECT DISTINCT empid
292    FROM checkout
293    WHERE custid =50);
294
295

```

## 3. Query to find the name and numbers of all salesmen who had more than one customer

ORACLE SQL Developer Home worksheet Data Modeler X ADMIN

Navigator Worksheets

ADMIN

Tables

Search... empid

1	1
2	2
3	3
4	6

```

311
312 -- Write a query to find the name and numbers of all salesmen who had more than one customer.
313
314 SELECT empid
315 FROM employees e
316 WHERE 1 <
317   (SELECT COUNT(*)
318    FROM checkout
319    WHERE empid=e.empid);
320
321
322

```

4. Query to find the sums of the amounts from the orders table, grouped by date, eliminating all those dates where the sum was not at least 10.00 above the maximum amount for that date.

```

-- Write a query to find the sums of the amounts from the orders table, grouped by date, eliminating all
-- those dates where the sum was not at least 10.00 above the maximum amount for that date.
338
339 SELECT checkout_date, SUM (total)
340 FROM checkout c
341 GROUP BY checkout_date
342 HAVING SUM (total) >
343   (SELECT 10.00 + MAX(total)
344    FROM checkout ch
345    WHERE c.checkout_date = ch.checkout_date);
346
347

```

	checkout_date	sum(total)
1	6/9/19	31.25
2	8/11/19	29.25

5. Query to display all the orders that had amounts that were greater than at least one of the orders from September 6th 2019.

```

-- Write a query to display all the orders that had amounts that were greater than at least one of the
-- orders from September 6th 2019.
349
350 SELECT *
351 FROM checkout
352 WHERE total > ANY
353   (SELECT total
354    FROM checkout
355    WHERE checkout_date='6/9/19');
356
357
358

```

	transaction_id	quantity	total	checkout_date	checkout_time	itemid
1	32	4	20	6/8/19	15:22:00	658
2	6589	3	18	8/11/19	13:10:00	4587
3	201	4	20	6/9/19	19:38:00	658

6. Query with transaction id, purchase amount, customer type and their type for those orders which order amount between 10 and 20.

ORACLE SQL Developer Home worksheet Data Modeler X ADMIN ▾

Navigator Worksheets

ADMIN

Tables

Search...

- ▶ CHECKOUT
- ▶ CUSTOMER
- ▶ EMPLOYEES
- ▶ INVENTORY
- ▶ MAIN
- ▶ PRODUCT
- ▶ SDW\$ERRS\$\_CHECKOUT
- ▶ SDW\$ERRS\$\_CUSTOMER
- ▶ SDW\$ERRS\$\_EMPLOYEES
- ▶ SDW\$ERRS\$\_INVENTORY
- ▶ SDW\$ERRS\$\_MAIN
- ▶ SDW\$ERRS\$\_PRODUCT

Nob\_Hill\* ▾

```

357
358
359 -- a query with transaction id, purchase amount, customer type and their type for those orders which order
360 amount between 10 and 20.
361 SELECT a.transaction_id,a.total,
362 b.cust_type,b.custid
363 FROM checkout a,customer b
364 WHERE a.custid=b.custid
365 AND a.total BETWEEN 10 AND 20;
366
367
368 -- a SQL statement to make a join on the tables salesman, customer and orders in such a form that the same

```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History

	transaction_id	total	cust_type	custid
1	201	11.25	M	50
2	201	20	M	50
3	32	20	NM	51
4	6589	18	NM	52
5	6589	11.25	NM	52

## 7. Natural join on Checkout, customer and employee tables

ORACLE SQL Developer Home worksheet Data Modeler X ADMIN ▾

Navigator Worksheets

ADMIN

Tables

Search...

- ▶ CHECKOUT
- ▶ CUSTOMER
- ▶ EMPLOYEES
- ▶ INVENTORY
- ▶ MAIN
- ▶ PRODUCT
- ▶ SDW\$ERRS\$\_CHECKOUT
- ▶ SDW\$ERRS\$\_CUSTOMER
- ▶ SDW\$ERRS\$\_EMPLOYEES
- ▶ SDW\$ERRS\$\_INVENTORY
- ▶ SDW\$ERRS\$\_MAIN
- ▶ SDW\$ERRS\$\_PRODUCT

Nob\_Hill\* ▾

```

365
366
367
368 -- a SQL statement to make a join on the tables employee, customer and checkout in such a form that the same
369 column of each table will appear once and only the relational rows will come.
370 SELECT *
371 FROM checkout
372 NATURAL JOIN customer
373 NATURAL JOIN employees;
374
375
376
377

```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History

	empid	custid	transaction_id	quantity	total	checkout_
1	2	50	201	5	11.25	6/9/19
2	2	51	32	4	20	6/8/19
3	3	52	6589	3	18	8/11/19
4	4	99	2147	4	7.96	6/4/19
5	5	53	210	2	5	11/5/19

ORACLE SQL Developer Worksheet

```

366
367
368 -- a SQL statement to make a join on the tables employee, customer and checkout in such that the same
369 SELECT *
370 FROM checkout
371 NATURAL JOIN customer
372 NATURAL JOIN employees;
373
374
375
376

```

Query Result

_date	checkout_time	itemid	checkout_type	mode_of_payment	cust_type
1	19:38:00	12	N	Card	M
2	15:22:00	658	N	Card	NM
3	13:10:00	4587	N	Card	NM
4	14:15:00	2365	N	Card	M
5	13:20:00	84854	S	Cash	M

0 0 0 0 | 1:36:04 AM - 14 rows total

8. Query to make a report with customer id, type, order number, order date, and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders.

ORACLE SQL Developer Worksheet

```

377
378 -- a SQL statement to make a report with customer id, type, order number, order date, and order amount in
379 ascending order according to the order date to find that either any of the existing customers have placed no
380 order or placed one or more orders.
381
382 SELECT a.custid,a.cust_type,b.transaction_id,
383 b.transaction_id,b.total AS "order Amount"
384 FROM customer a
385 LEFT OUTER JOIN checkout b
386 ON a.custid=b.custid
387 order by b.checkout_date;
388
389

```

Query Result

custid	cust_type	transaction_id	checkout_date	order amount
1	NM	115	1/10/20	8.97
2	M	110	1/2/20	2.5
3	M	110	1/2/20	5.94
4	M	110	1/2/20	7.98
5	M	210	11/5/19	5

0 0 0 0 | 1:37:35 AM - 15 rows total

9. Query to show for each product inventory details

ORACLE SQL Developer Worksheet

```

391
392 | -- query to show for each product inventory details
393 SELECT *
394 FROM product
395 INNER JOIN inventory
396 ON product.itemid= inventory.FK1ITEMID;
397
398
399
400
401
402

```

Query Result

itemid	item_name	price	brand	inventory_quantity	fk1itemid
1	Cookies	2.25	Nabisco	10	12
2	Cigarettes		PhillipMorris	10	658
3	Cheese	6	Kraft	4	4587
4	Cereal	1.99	Kellogg	23	2365
5	Oatmeal	2.5	Quaker	38	84854

#### 10. Query to find the minimum and maximum quantity of purchases made by customers.

ORACLE SQL Developer Worksheet

```

433 |-- minimum and maximum quantity of purchases made by customers.
434
435 select c1.custid, quantity,'Max Quantity' as Description
436 from customer c1,checkout ol
437 where c1.custid = ol.custid
438 and quantity = (select max(quantity) from checkout)
439 UNION
440 select c1.custid, quantity,'Min Quantity' as Description
441 from customer c1,checkout ol
442 where c1.custid = ol.custid
443 and quantity = (select min(quantity) from checkout)
444 order by 2;

```

Query Result

custid	quantity	description
1	105	Min Quantity
2	105	Max Quantity
3	178	Max Quantity

#### 11. Query to find the number of item IDs for all the orders that have Item\_name = 'Cookies'

ORACLE SQL Developer Worksheet

```

422
423
424 -- find the number of item IDs for all the orders that have Item_Name = 'Cookies'
425 SELECT distinct count(o.itemid) as NumOfOrderIds From checkout o
426 Where exists
427 (select * from product where itemid = o.itemid and item_name = 'Cookies');
428
429
430
431
432
433

```

Query Result

numoforderids	
1	2

### Procedure:

1. A stored procedure was created to delete a customer's orders history. This can be used when a customer is requesting a return.

The screenshot shows two sessions in Oracle SQL Developer. The top session displays the creation of a stored procedure named DeleteCheckout. The code is as follows:

```
79 -- A stored procedure was created to delete a customer's orders history. This can be used when a customer
80 -- is requesting a return.
81 CREATE OR REPLACE
82 PROCEDURE DeleteCheckout (pri_key IN NUMBER)
83 AS
84 BEGIN
85   DELETE CHECKOUT WHERE Transaction_id=pri_key;
86 END;
87 call DeleteCheckout(201);
88 select * from checkout
89 where Transaction_id = '201';
```

The bottom session shows the execution of the procedure and its results. It includes the output of the procedure compilation and the confirmation of the call completion.

```
Procedure DELETECHECKOUT compiled

Call completed.
```

The second session at the bottom shows the same code again, with the final SELECT statement highlighted in blue. The results pane indicates "No items to display".

2. Procedure to insert rows in checkout table for a new transaction

```

73 CREATE OR REPLACE
74 PROCEDURE "InsertCheckout" (Quantity CHAR,Total NUMBER,Checkout_Date VARCHAR,Checkout_Time VARCHAR,
75                               ItemID CHAR,EmpID CHAR,CustID CHAR, Checkout_Type VARCHAR,Mode_of_payment
76                               VARCHAR)
76 AS
77 BEGIN
78
79 INSERT INTO CHECKOUT VALUES ('111',QUANTITY, TOTAL, CHECKOUT_DATE, CHECKOUT_TIME, ITEMID, EMPID, CUSTID,
80                               CHECKOUT_TYPE, MODE_OF_PAYMENT);
80 END;
81
82 CALL InsertCheckout('10','35','6/2/2020','6:38:00','84854','6','201','N','Card');

```

Query Result    Script Output    DBMS Output    Explain Plan    Autotrace    SQL History    [?](#)

[Delete](#) [Download](#)

the data dictionary.)

[Copy](#) [Edit](#) [Information](#)

Procedure InsertCheckout compiled

### Trigger:

1. Trigger to update customer's total. This can be used when a customer decides against buying some goods or there is a change in the total.

```

32 CREATE TRIGGER ordersummary1_trg
33 INSTEAD OF UPDATE
34 ON ordersummary1
35 FOR EACH ROW
36 BEGIN
37   UPDATE checkout
38   SET total=:new.total
39   WHERE transaction_id=:old.transaction_id;
40 END;
41
42
43 BEGIN
44   UPDATE ordersummary1 SET total=15 WHERE transaction_id=32;
45 COMMIT;
46 END;

```

Query Result    Script Output    DBMS Output    Explain Plan    Autotrace    SQL History    [?](#)

[Delete](#) [Download](#)

Trigger ORDERSUMMARY1\_TRG compiled

[Copy](#)

PL/SQL procedure successfully completed.

1\* | | | Aa |

```

35 FOR EACH ROW
36 BEGIN
37 UPDATE checkout
38 SET total=:new.total
39 WHERE transaction_id=:old.transaction_id;
40 END;
41
42
43 BEGIN
44 UPDATE ordersummary1 SET total=15 WHERE transaction_id=32;
45 COMMIT;
46 END;
47
48 SELECT * FROM ordersummary1;
49

```

Query Result | Script Output | DBMS Output | Explain Plan | Autotrace | SQL History

Download ▾

	transaction_id	custid	total	empid
1	32	51	15	2
2	6589	52	18	3
3	2147	99	7.96	4
4	210	53	5	5

2. Trigger to keep an audit table of the checkout table. To see the new and old values. Also, to update the checkout table for change in quantity and total.

```

37
38 CREATE TABLE orders
39 ( transaction_id number(5),
40 itemid number(5),
41 quantity_before number(4),
42 quantity_after number(4),
43 total_before number(8,2),
44 total_after number(8,2)
45 );
46
47
48 CREATE OR REPLACE TRIGGER orders_after_update
49 AFTER UPDATE
50 ON Checkout
51 FOR EACH ROW
52
53 BEGIN
54
55 -- Insert record into audit table
56 INSERT INTO orders
57 ( transaction_id,
58 itemid,
59 quantity_before,
60 quantity_after,
61 total_before,
62 total_after
63 )
64 VALUES
65 ( :new.transaction_id,
66 :new.itemid,
67 :old.quantity,

```

```

67      :old.quantity,
68      :new.quantity,
69      :old.total,
70      :new.total
71      );
72
73 END;
74
75
76 Update Checkout set quantity = 8 total = 15.92 where transaction_id = 2147;

```

Query Result

Script Output

DBMS Output

Explain Plan

Autotrace

SQL History



Table ORDERS created.



Trigger ORDERS\_AFTER\_UPDATE compiled

Trigger1\*



```

67      :old.quantity,
68      :new.quantity,
69      :old.total,
70      :new.total
71      );
72
73 END;
74
75
76 Update Checkout set quantity = 8, total = 15.92 where transaction_id = 2147;
77
78 select * from orders;
79 select * from checkout;

```

Query Result

Script Output

DBMS Output

Explain Plan

Autotrace

SQL History



Download ▾

	on_id	itemid	quantity_before	quantity_after	total_before	total_after
1	2147	2365	6	8	11.94	15.92

```

// 78 select * from orders;
79 select * from checkout;

```

Query Result

Script Output

DBMS Output

Explain Plan

Autotrace

SQL History



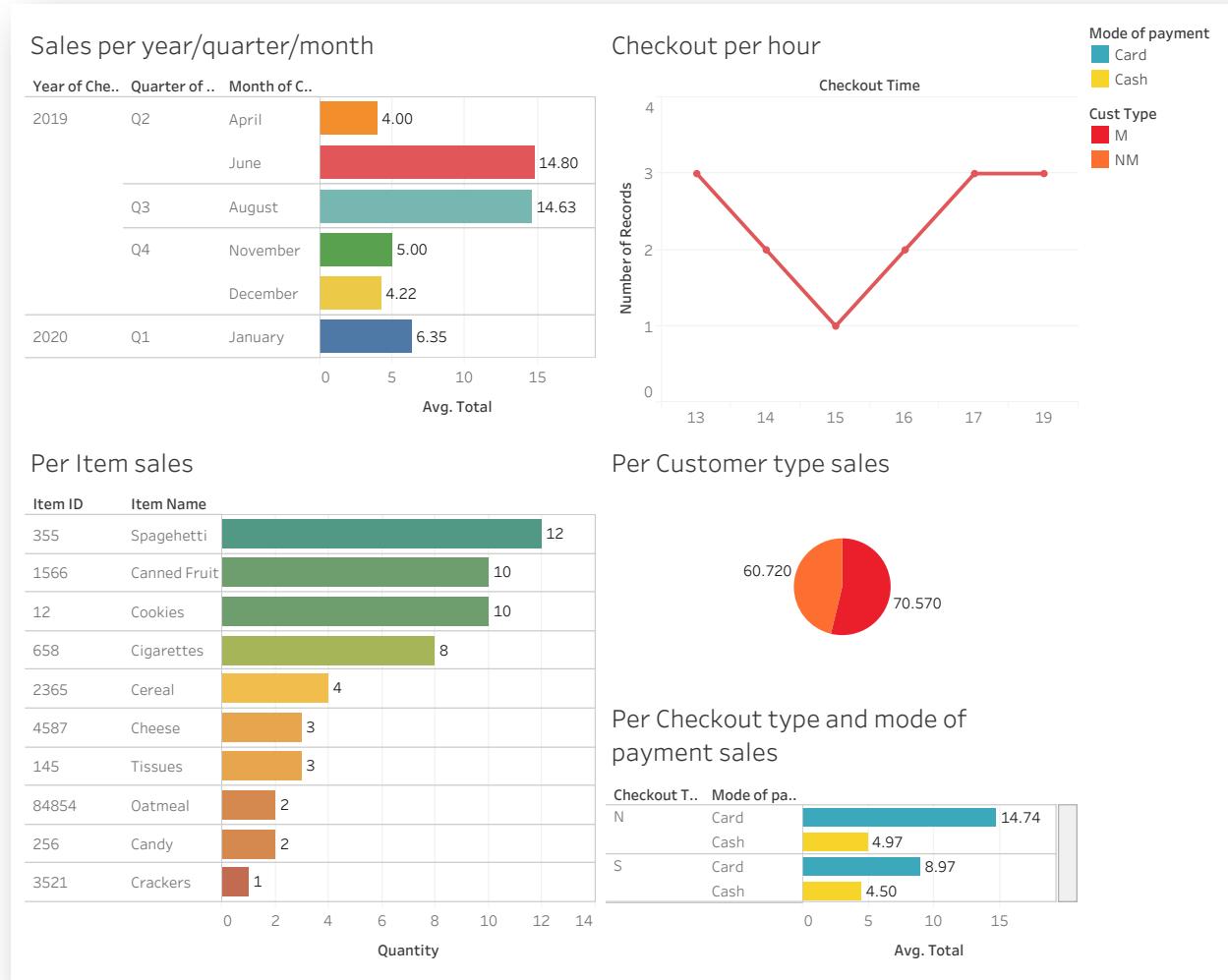
Download ▾

	transaction_id	quantity	total	checkout_date	checkout_time	itemid
3	2147	8	15.92	6/4/19	14:15:00	2365
4	210	2	5	11/5/19	13:20:00	84854
5	2141	1	4	4/4/19	14:25:00	3521

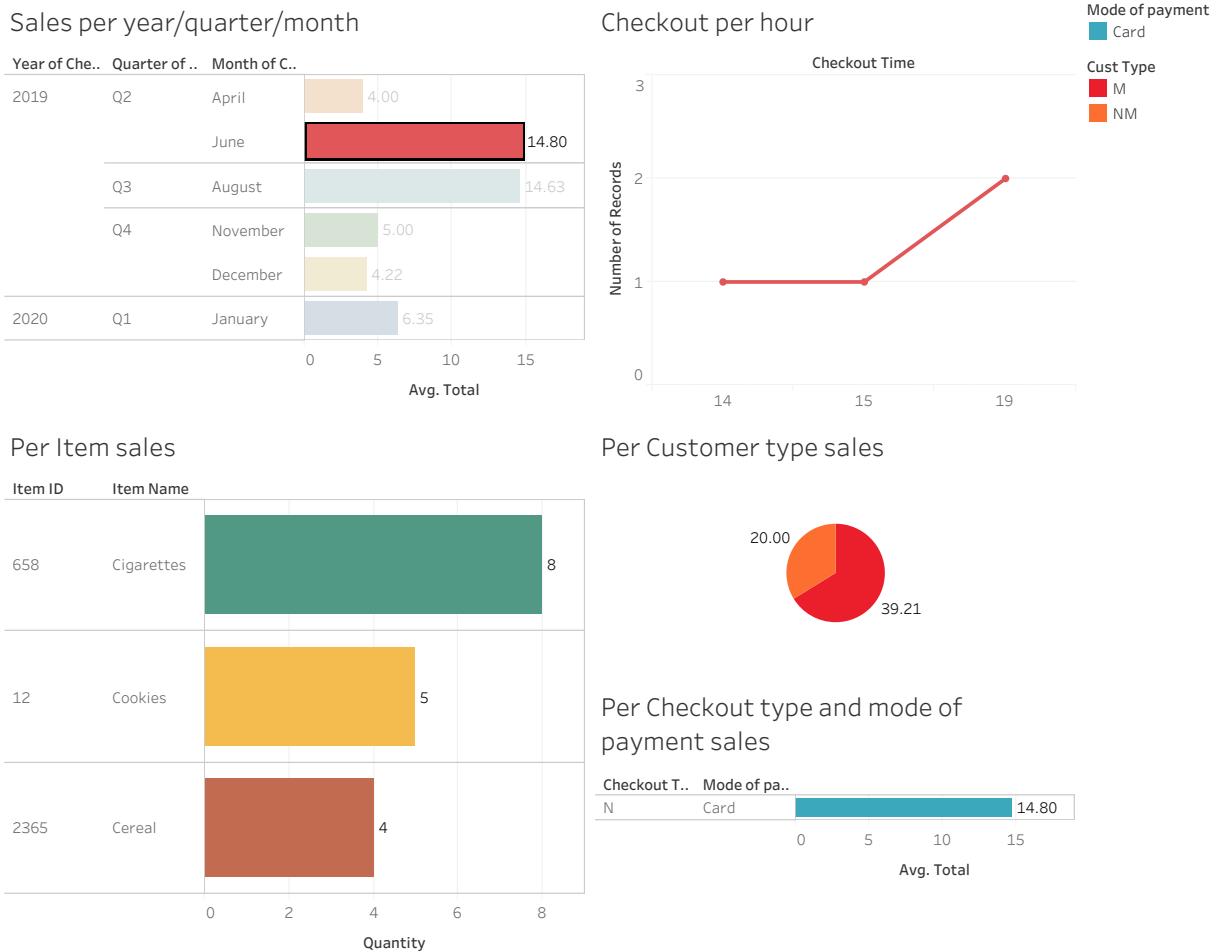
## Tableau

The tableau dashboard created below covers the different attributes which impacts the sales.

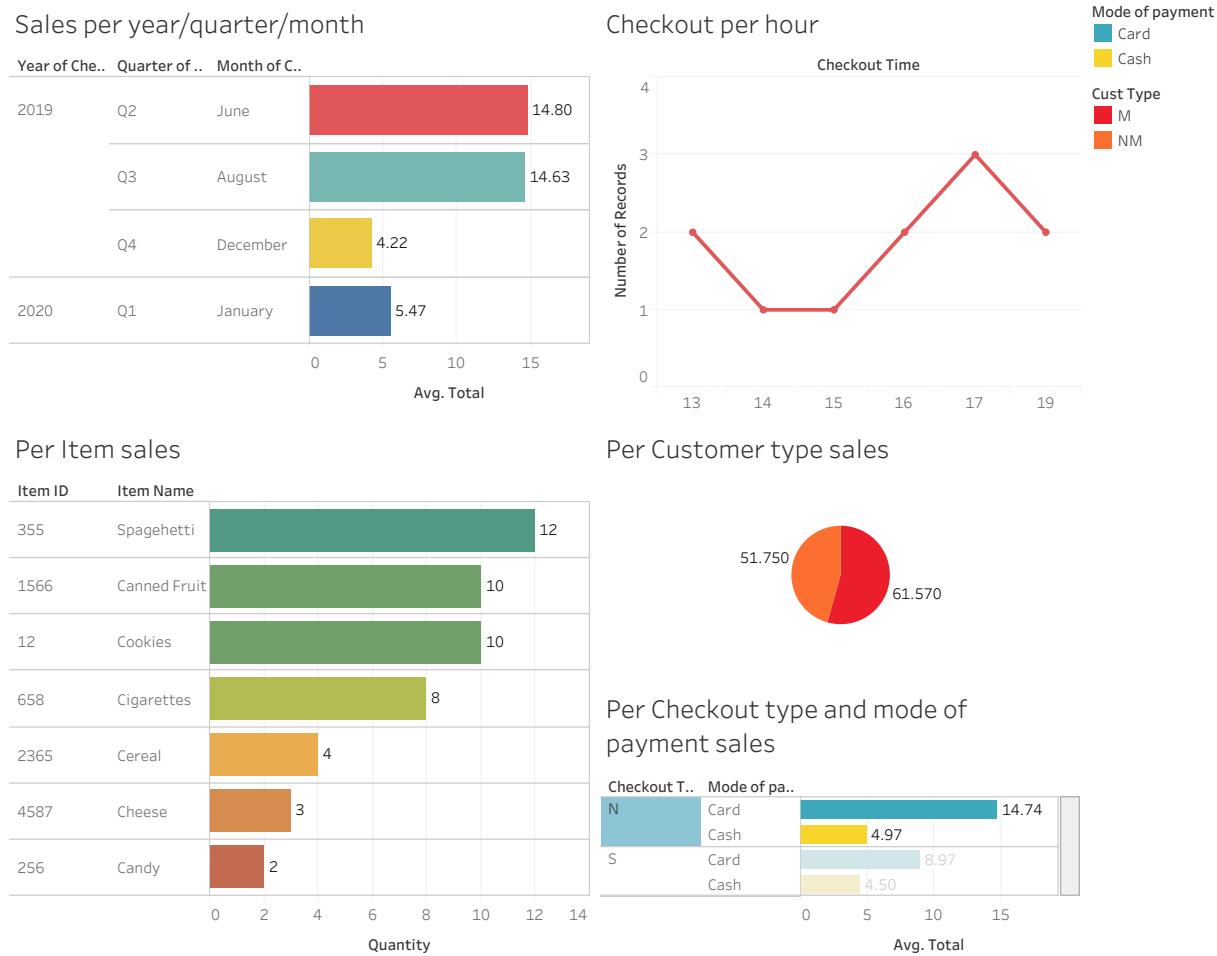
### 1. Basic Dashboard view:



### 2. Dashboard view for the month of 'June':



### 3. Dashboard view for Normal checkout customers:



### **Conclusion:**

In conclusion, grocery Management System has to do with making appropriate effort to stop the rising problem to all manual supermarket operation in order to enhance the operation of such supermarket. In this project, I have analyzed various processes of grocery management for Nob Hill and answered few business questions with the help of Oracle SQL and Tableau. The dashboard created on Tableau will help the management make informed decisions on how to make the grocery management system more efficient.

### **Future scope:**

Currently this project is limited to only the customer checkout process. This can be improved to include processes like employee management, inventory management, customer management. Also, all these processes can be integrated into a single UI for ease of use. We need more data to expand into all these processes.