Assignment 1

Pragati Koladiya
FALL A 2020
ALY6030
Assignment 1
Last Updated: 09/25/2020

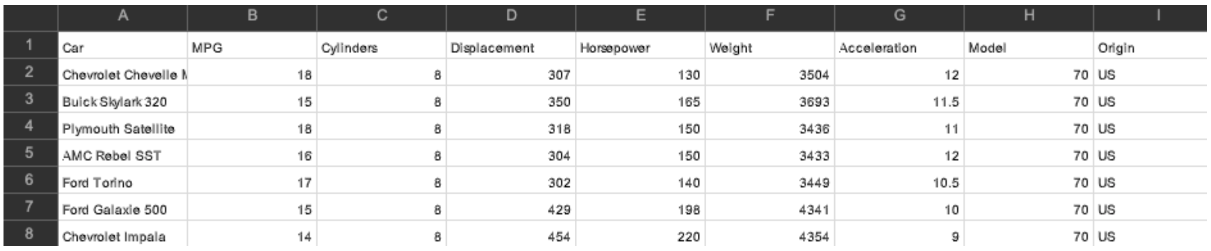
Problem 1: Data modeling and SQL

For the problem 1, I have considered 'car.csv' as my dataset.

1. Columns with it's appropriate datatype is as follows,

Columns(attributes)	Datatype
Car	varchar
MPG	int
Cylinders	int
Displacement	int
Horsepower	int
Weight	int
Acceleration	decimal
Model	int
Origin	varchar

- Head of the dataset:



	A	B	C	D	E	F	G	H	I
1	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Origin
2	Chevrolet Chevelle M	18	8	307	130	3504	12	70	US
3	Buick Skylark 320	15	8	350	165	3693	11.5	70	US
4	Plymouth Satellite	18	8	318	150	3436	11	70	US
5	AMC Rebel SST	16	8	304	150	3433	12	70	US
6	Ford Torino	17	8	302	140	3449	10.5	70	US
7	Ford Galaxie 500	15	8	429	198	4341	10	70	US
8	Chevrolet Impala	14	8	454	220	4354	9	70	US

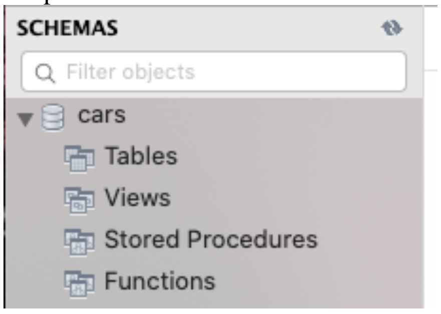
- I have chosen datatypes by observing the table values as there are multiple options for single column for instance, 'Car' column may have varchar, char or text likewise all the column fields have multiple options to choose from.

2. Creating schema in MYSQL workbench

Input Query:

```
/* Creating schema 'cars' */
CREATE SCHEMA `cars`;
```

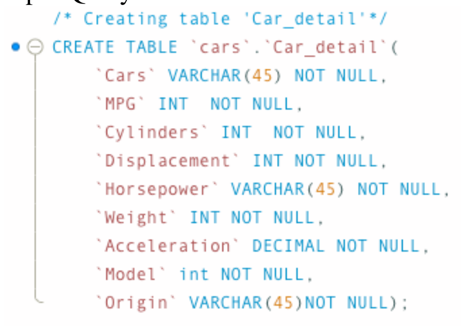
Output:



- Schema is created after query execution.

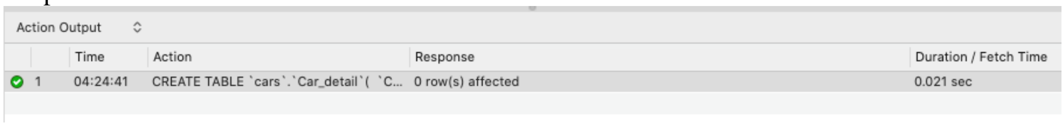
a. Creating the table ‘cars’

Input Query:



```
/* Creating table 'Car_detail'*/
CREATE TABLE `cars`.`Car_detail`(
    `Cars` VARCHAR(45) NOT NULL,
    `MPG` INT  NOT NULL,
    `Cylinders` INT  NOT NULL,
    `Displacement` INT NOT NULL,
    `Horsepower` VARCHAR(45) NOT NULL,
    `Weight` INT NOT NULL,
    `Acceleration` DECIMAL NOT NULL,
    `Model` int NOT NULL,
    `Origin` VARCHAR(45)NOT NULL);
```

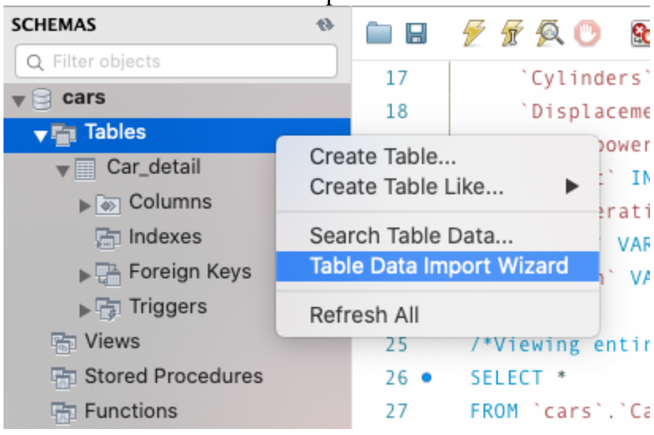
Output:



```
Action Output
     Time      Action                                   Response            Duration / Fetch Time
✓ 1  04:24:41  CREATE TABLE `cars`.`Car_detail`( `C...  0 row(s) affected   0.021 sec
```

b. Importing data from ‘Cars.csv’ using following steps,

- Select Table Data Import Wizard



- Browse the csv file

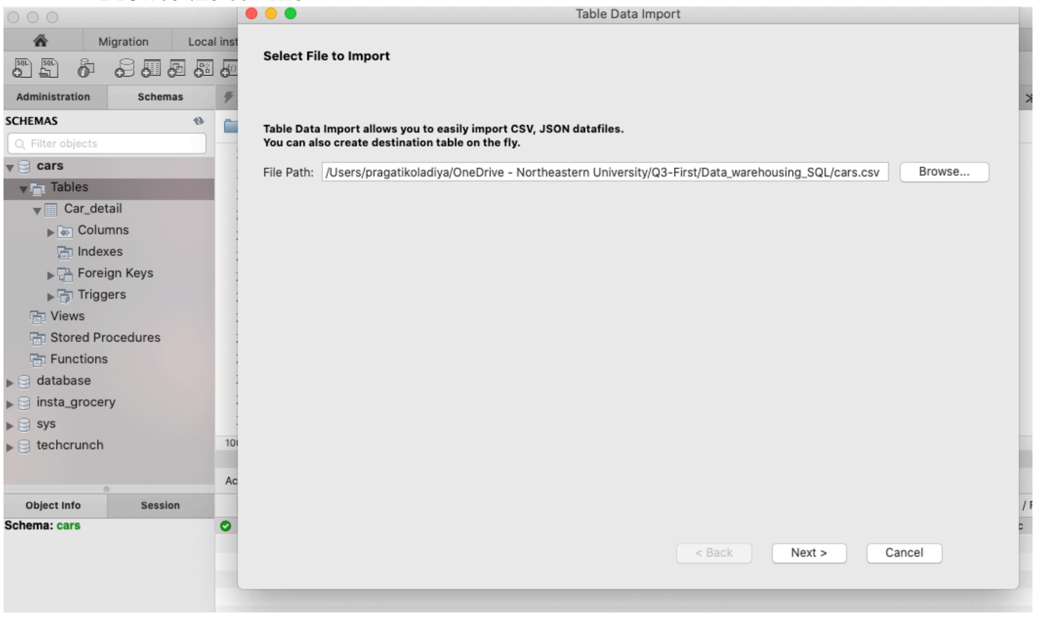


Table Data Import

Select File to Import

Table Data Import allows you to easily import CSV, JSON datafiles.
You can also create destination table on the fly.

File Path: /Users/pragatikoladiya/OneDrive - Northeastern University/Q3-First/Data_warehousing_SQL/cars.csv

- Use existing table which we have created earlier name 'Car_detail'

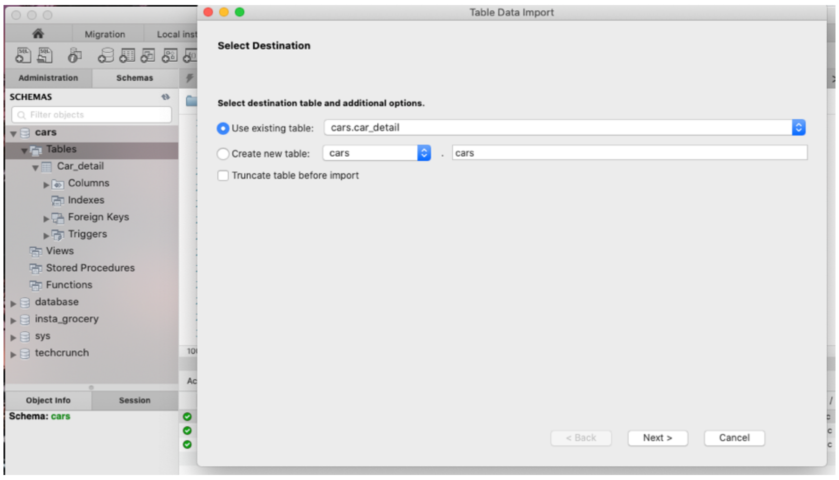


Table Data Import

Select Destination

Select destination table and additional options.

[✓ Use existing table:] cars.car_detail

Create new table: cars . cars

Truncate table before import

Schema: cars

- View the configure import settings

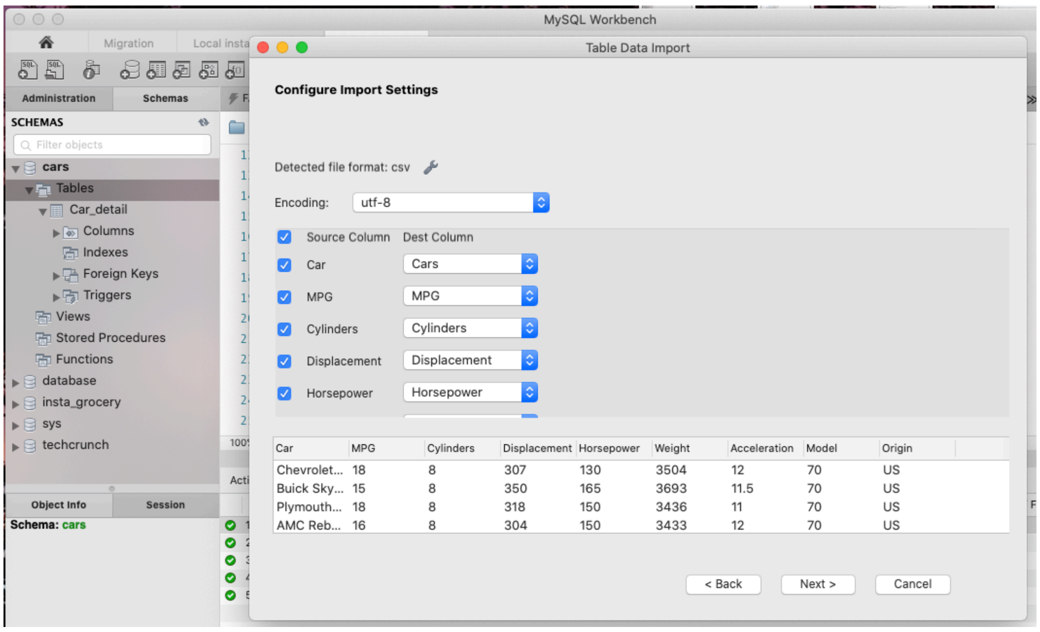


Table Data Import

Configure Import Settings

Detected file format: csv

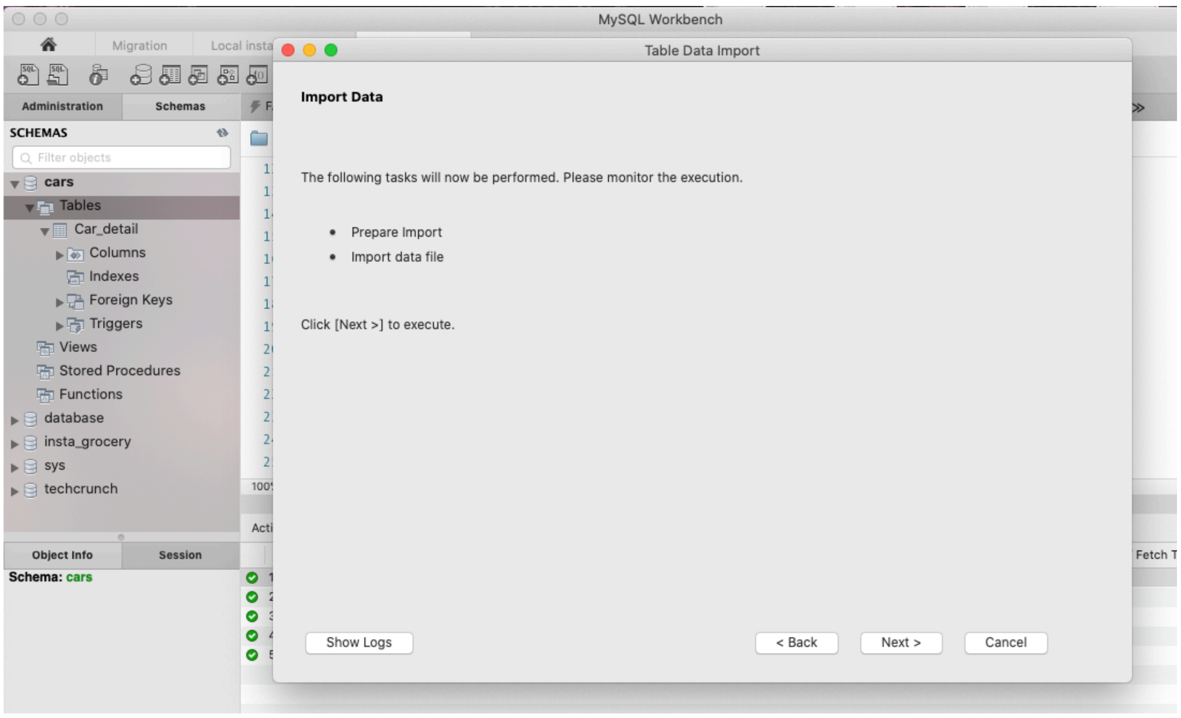
Encoding: utf-8

Source Column	Dest Column
✓ Car	Cars
✓ MPG	MPG
✓ Cylinders	Cylinders
✓ Displacement	Displacement
✓ Horsepower	Horsepower

Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Origin
Chevrolet...	18	8	307	130	3504	12	70	US
Buick Sky...	15	8	350	165	3693	11.5	70	US
Plymouth...	18	8	318	150	3436	11	70	US
AMC Reb...	16	8	304	150	3433	12	70	US

Schema: cars

- Click next to import



Import Data

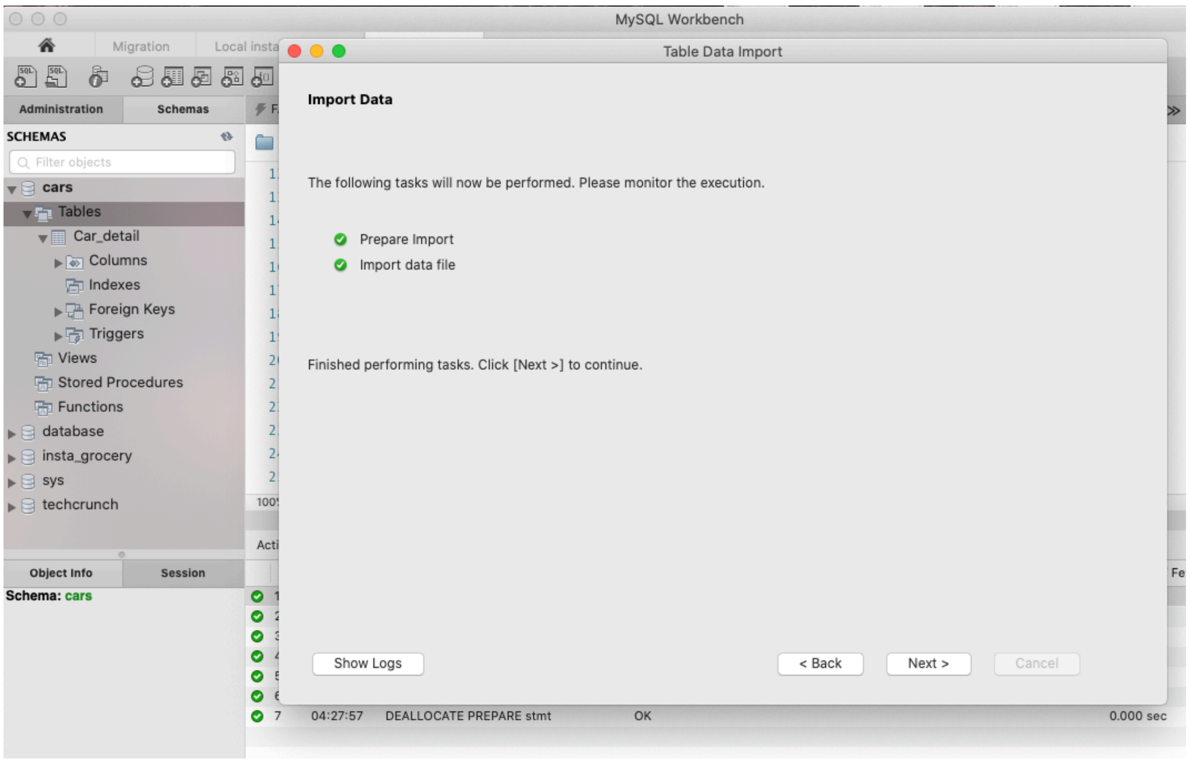
The following tasks will now be performed. Please monitor the execution.

- Prepare Import
- Import data file

Click [Next >] to execute.

Schema: cars

- Import data file loaded



Import Data

The following tasks will now be performed. Please monitor the execution.

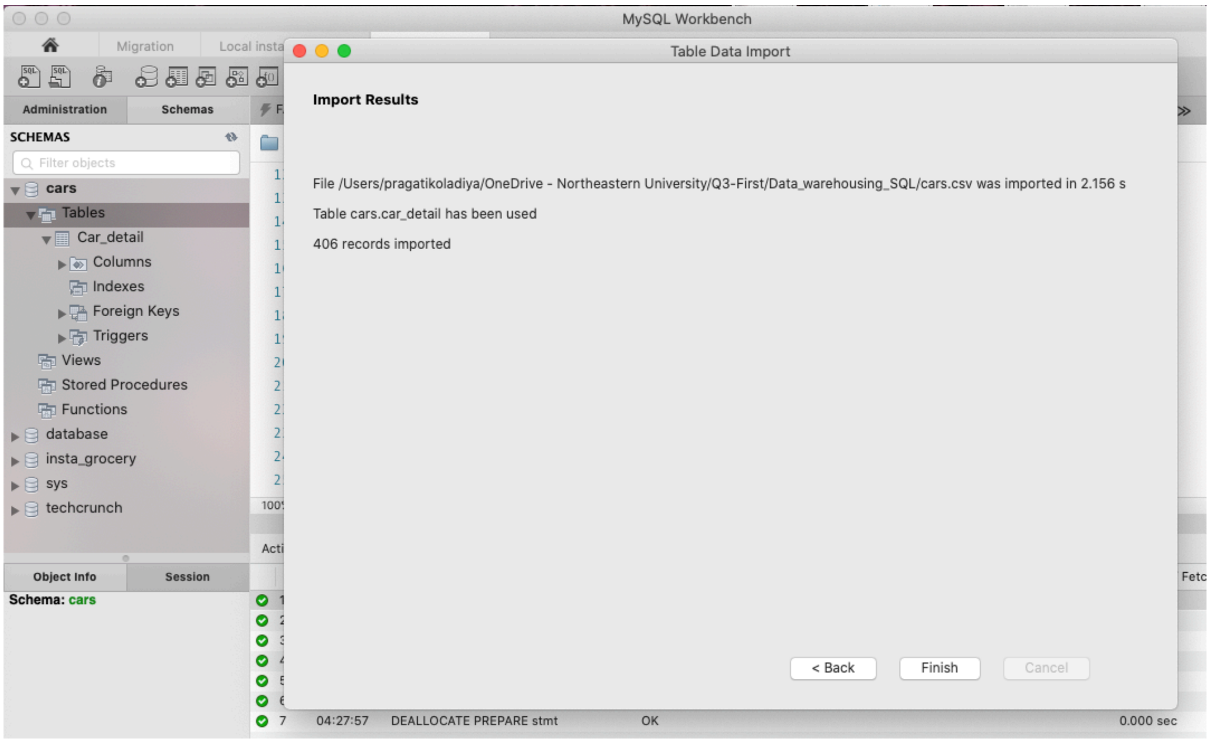
- ✓ Prepare Import
- ✓ Import data file

Finished performing tasks. Click [Next >] to continue.

Schema: cars

```
7    04:27:57    DEALLOCATE PREPARE stmt    OK    0.000 sec
```

- Import results shows 406 records are imported, click finish to continue with MySQL Workbench



Import Results

File /Users/pragatikoladiya/OneDrive - Northeastern University/Q3-First/Data_warehousing_SQL/cars.csv was imported in 2.156 s

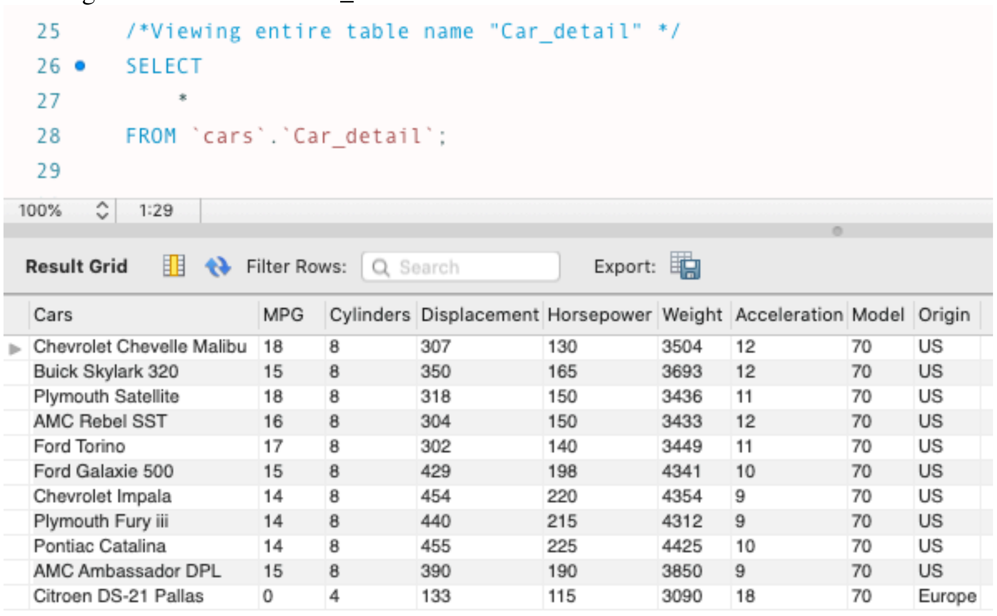
Table cars.car_detail has been used

406 records imported

```
7    04:27:57    DEALLOCATE PREPARE stmt    OK    0.000 sec
```

c. Questions asked based on the dataset provided are as follows,

- Viewing entire table name 'Car_detail'



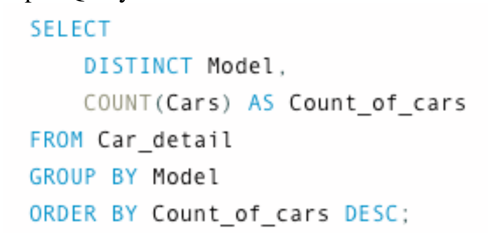
```
/*Viewing entire table name "Car_detail" */
SELECT
    *
FROM `cars`.`Car_detail`;
```

Cars	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Origin
Chevrolet Chevelle Malibu	18	8	307	130	3504	12	70	US
Buick Skylark 320	15	8	350	165	3693	12	70	US
Plymouth Satellite	18	8	318	150	3436	11	70	US
AMC Rebel SST	16	8	304	150	3433	12	70	US
Ford Torino	17	8	302	140	3449	11	70	US
Ford Galaxie 500	15	8	429	198	4341	10	70	US
Chevrolet Impala	14	8	454	220	4354	9	70	US
Plymouth Fury iii	14	8	440	215	4312	9	70	US
Pontiac Catalina	14	8	455	225	4425	10	70	US
AMC Ambassador DPL	15	8	390	190	3850	9	70	US
Citroen DS-21 Pallas	0	4	133	115	3090	18	70	Europe

- Glimpse of full table before executing queries.

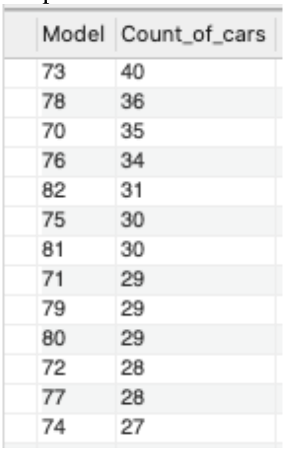
Q1. What is the count of cars by distinct models?

Input Query:



```
SELECT
    DISTINCT Model,
    COUNT(Cars) AS Count_of_cars
FROM Car_detail
GROUP BY Model
ORDER BY Count_of_cars DESC;
```

Output:

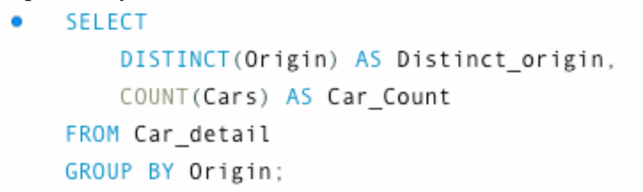


Model	Count_of_cars
73	40
78	36
70	35
76	34
82	31
75	30
81	30
71	29
79	29
80	29
72	28
77	28
74	27

There are 13 unique car model among those the maximum number of cars belongs to model 73 with cars count 40.

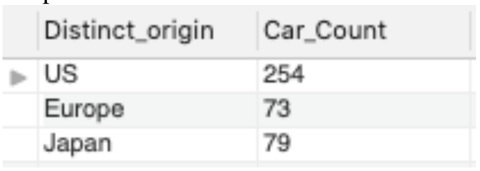
Q2. List count of cars by distinct origin.

Input Query:

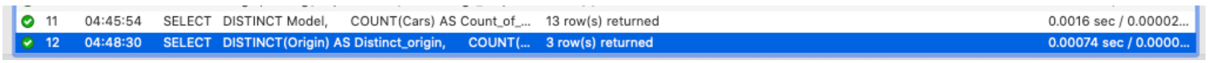


```
SELECT
    DISTINCT(Origin) AS Distinct_origin,
    COUNT(Cars) AS Car_Count
FROM Car_detail
GROUP BY Origin;
```

Output:



Distinct_origin	Car_Count
US	254
Europe	73
Japan	79

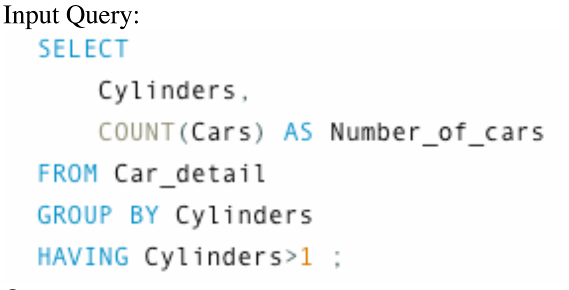


```
11  04:45:54  SELECT  DISTINCT Model,      COUNT(Cars) AS Count_of_...  13 row(s) returned  0.0016 sec / 0.00002...
12  04:48:30  SELECT  DISTINCT(Origin) AS Distinct_origin,    COUNT(...  3 row(s) returned  0.00074 sec / 0.0000...
```

There are mainly three origin US, Europe and Japan. US has the maximum number of cars among all the origin which is 254 in total.

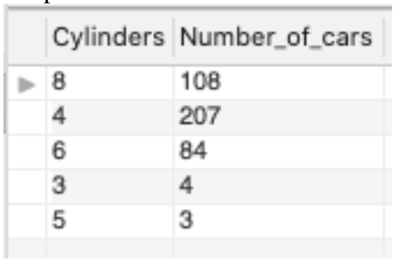
Q3. What is the count of car which has more than one cylinder?

Input Query:

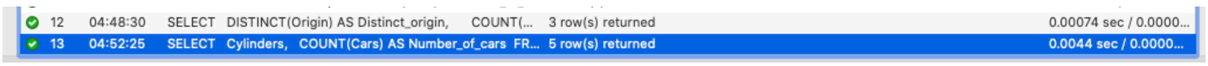


```
SELECT
    Cylinders,
    COUNT(Cars) AS Number_of_cars
FROM Car_detail
GROUP BY Cylinders
HAVING Cylinders>1 ;
```

Output:



Cylinders	Number_of_cars
8	108
4	207
6	84
3	4
5	3

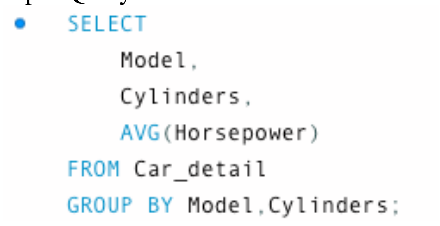


```
12  04:48:30  SELECT  DISTINCT(Origin) AS Distinct_origin,    COUNT(...  3 row(s) returned   0.00074 sec / 0.0000...
13  04:52:25  SELECT  Cylinders,  COUNT(Cars) AS Number_of_cars  FR...  5 row(s) returned   0.0044 sec / 0.0000...
```

The maximum number of cars has 4 cylinder and the second most is with 8-cylinder car whereas 3 and 5 cylinder cars the list popular.

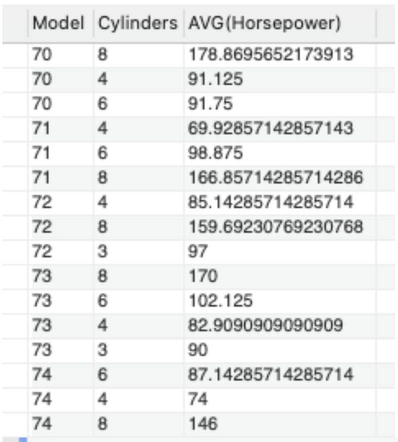
Q4. Give a list of models with maximum cylinders and an engine with highest horsepower along with high MPG.

Input Query:

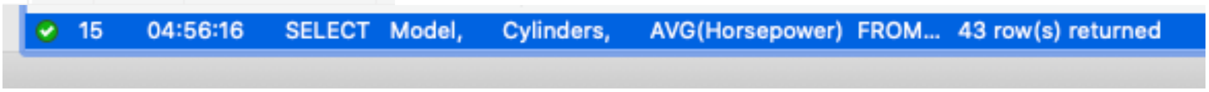


```
SELECT
    Model,
    Cylinders,
    AVG(Horsepower)
FROM Car_detail
GROUP BY Model,Cylinders;
```

Output:



Model	Cylinders	AVG(Horsepower)
70	8	178.8695652173913
70	4	91.125
70	6	91.75
71	4	69.92857142857143
71	6	98.875
71	8	166.85714285714286
72	4	85.14285714285714
72	8	159.69230769230768
72	3	97
73	8	170
73	6	102.125
73	4	82.9090909090909
73	3	90
74	6	87.14285714285714
74	4	74
74	8	146

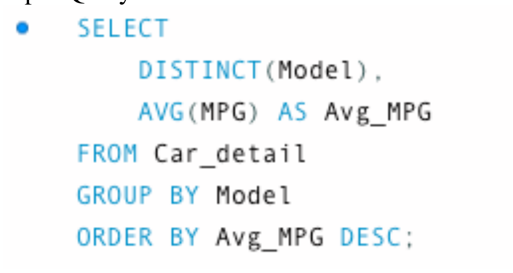


```
15  04:56:16  SELECT  Model,  Cylinders,  AVG(Horsepower)  FROM...  43 row(s) returned
```

The model 70 with cylinders 8 has the highest average horsepower among 13 different models.

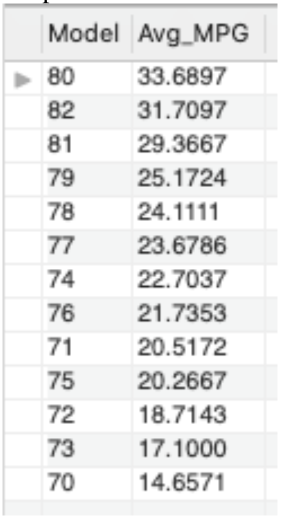
Q5. Provide a list of unique models with average MPG (Mileage Per Gallon).

Input Query:

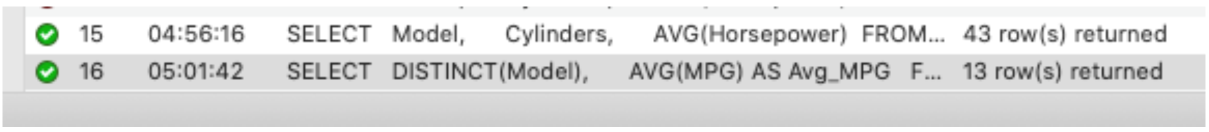


```
SELECT
    DISTINCT(Model),
    AVG(MPG) AS Avg_MPG
FROM Car_detail
GROUP BY Model
ORDER BY Avg_MPG DESC;
```

Output:



Model	Avg_MPG
80	33.6897
82	31.7097
81	29.3667
79	25.1724
78	24.1111
77	23.6786
74	22.7037
76	21.7353
71	20.5172
75	20.2667
72	18.7143
73	17.1000
70	14.6571

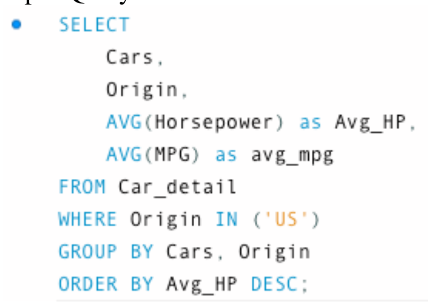


```
15  04:56:16  SELECT  Model,  Cylinders,  AVG(Horsepower)  FROM...  43 row(s) returned
16  05:01:42  SELECT  DISTINCT(Model),  AVG(MPG) AS Avg_MPG  F...  13 row(s) returned
```

The model 80 has the highest average mileage per gallon,33.69.

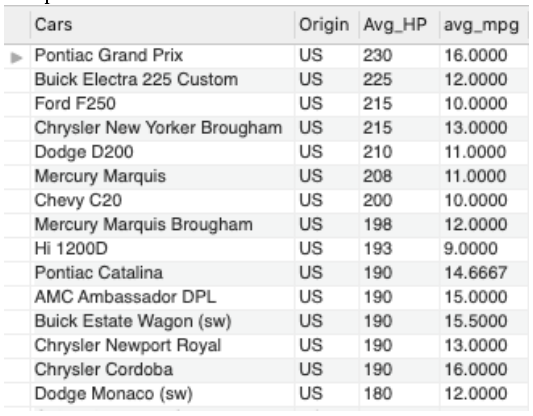
Q6. Provide a list of cars which has origin 'US', average horsepower and MPG from highest to lowest.

Input Query:

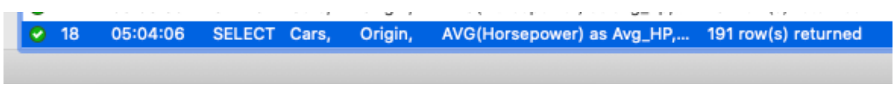


```
SELECT
    Cars,
    Origin,
    AVG(Horsepower) as Avg_HP,
    AVG(MPG) as avg_mpg
FROM Car_detail
WHERE Origin IN ('US')
GROUP BY Cars, Origin
ORDER BY Avg_HP DESC;
```

Output:



Cars	Origin	Avg_HP	avg_mpg
Pontiac Grand Prix	US	230	16.0000
Buick Electra 225 Custom	US	225	12.0000
Ford F250	US	215	10.0000
Chrysler New Yorker Brougham	US	215	13.0000
Dodge D200	US	210	11.0000
Mercury Marquis	US	208	11.0000
Chevy C20	US	200	10.0000
Mercury Marquis Brougham	US	198	12.0000
Hi 1200D	US	193	9.0000
Pontiac Catalina	US	190	14.6667
AMC Ambassador DPL	US	190	15.0000
Buick Estate Wagon (sw)	US	190	15.5000
Chrysler Newport Royal	US	190	13.0000
Chrysler Cordoba	US	190	16.0000
Dodge Monaco (sw)	US	180	12.0000

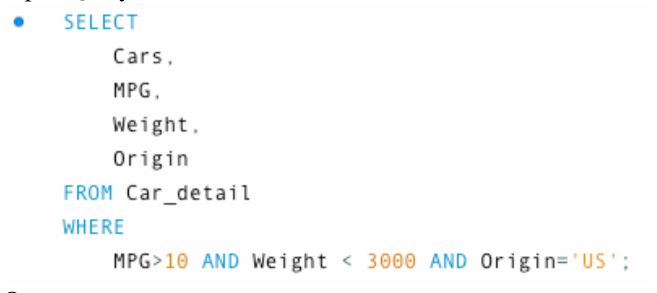


```
18    05:04:06    SELECT  Cars,   Origin,    AVG(Horsepower) as Avg_HP,...    191 row(s) returned
```

The above table shows the cars from US. They have highest to lowest order of average MPG.

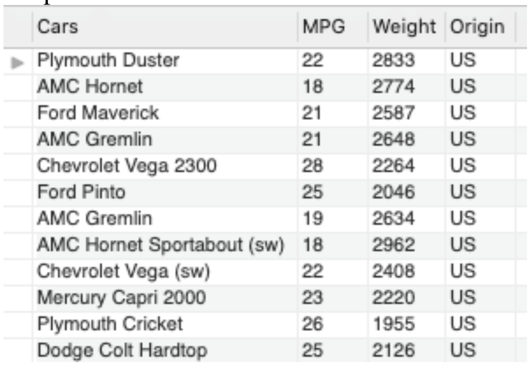
Q7. List the name of cars which has origin US and their MPG should be more than 10 with weight less than 3000

Input Query:

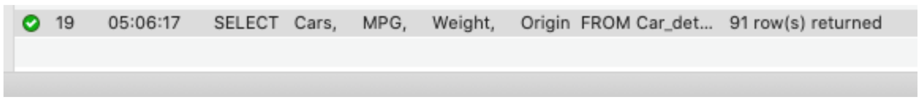


```
SELECT
    Cars,
    MPG,
    Weight,
    Origin
FROM Car_detail
WHERE
    MPG>10 AND Weight < 3000 AND Origin='US';
```

Output:



Cars	MPG	Weight	Origin
Plymouth Duster	22	2833	US
AMC Hornet	18	2774	US
Ford Maverick	21	2587	US
AMC Gremlin	21	2648	US
Chevrolet Vega 2300	28	2264	US
Ford Pinto	25	2046	US
AMC Gremlin	19	2634	US
AMC Hornet Sportabout (sw)	18	2962	US
Chevrolet Vega (sw)	22	2408	US
Mercury Capri 2000	23	2220	US
Plymouth Cricket	26	1955	US
Dodge Colt Hardtop	25	2126	US



```
19    05:06:17    SELECT  Cars,   MPG,   Weight,   Origin  FROM Car_det...   91 row(s) returned
```

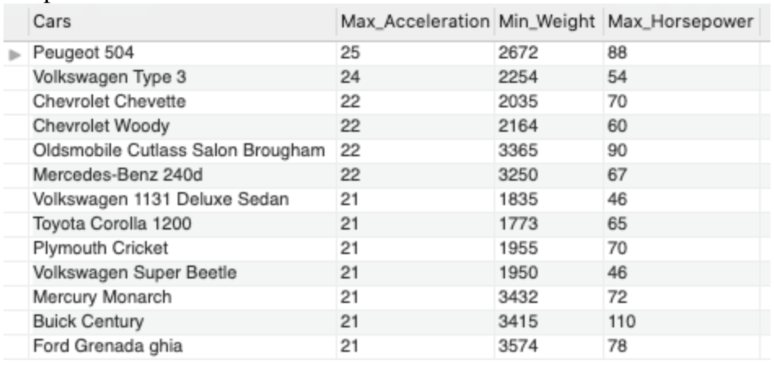
There are 91 records which show cars from the US. Their MPG is more than 10 and weight is less than 3000. Most cars have MPG above 10.

Q8. List the cars with maximum acceleration and horsepower but minimum weight.

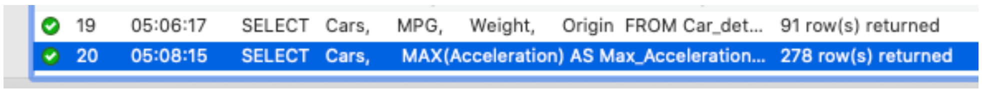
Input Query:

```
•   SELECT
        Cars,
        MAX(Acceleration) AS Max_Acceleration,
        MIN(Weight)AS Min_Weight,
        MAX(Horsepower) AS Max_Horsepower
    FROM Car_detail
    WHERE
        MPG BETWEEN 0 AND 35
    GROUP BY Cars
    ORDER BY Max_Acceleration DESC;
```

Output:



Cars	Max_Acceleration	Min_Weight	Max_Horsepower
Peugeot 504	25	2672	88
Volkswagen Type 3	24	2254	54
Chevrolet Chevette	22	2035	70
Chevrolet Woody	22	2164	60
Oldsmobile Cutlass Salon Brougham	22	3365	90
Mercedes-Benz 240d	22	3250	67
Volkswagen 1131 Deluxe Sedan	21	1835	46
Toyota Corolla 1200	21	1773	65
Plymouth Cricket	21	1955	70
Volkswagen Super Beetle	21	1950	46
Mercury Monarch	21	3432	72
Buick Century	21	3415	110
Ford Grenada ghia	21	3574	78



```
19    05:06:17    SELECT  Cars,   MPG,   Weight,   Origin  FROM Car_det...   91 row(s) returned
20    05:08:15    SELECT  Cars,   MAX(Acceleration) AS Max_Acceleration...   278 row(s) returned
```

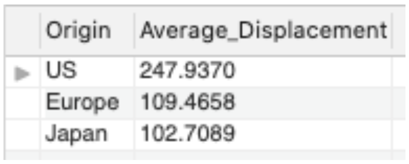
The speed of a car is high if acceleration and horsepower are high. But the weight is inversely proportional to its velocity. Lower the weight of the car better the speed. According to the given data Peugeot 504 is a high-speed car.

Q9. What is the average displacement by origin?

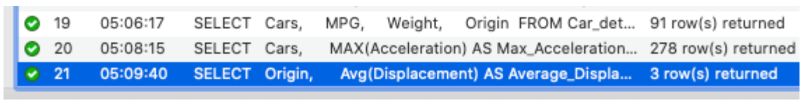
Input Query:

```
•   SELECT
        Origin,
        Avg(Displacement) AS Average_Displacement
    FROM Car_detail
    GROUP BY Origin;
```

Output:



Origin	Average_Displacement
US	247.9370
Europe	109.4658
Japan	102.7089



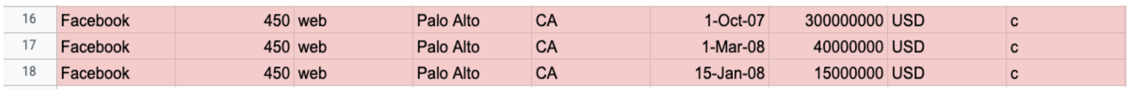
✓	19	05:06:17	SELECT Cars, MPG, Weight, Origin FROM Car_det...	91 row(s) returned
✓	20	05:08:15	SELECT Cars, MAX(Acceleration) AS Max_Acceleration...	278 row(s) returned
✓	21	05:09:40	SELECT Origin, Avg(Displacement) AS Average_Displa...	3 row(s) returned

The average displacement is highest in the US cars whereas Japan has the lowest displacement of 102.71.

Problem 2: Normalization

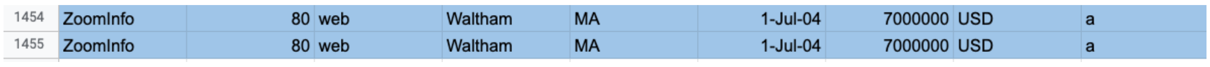
For this problem we need to consider dataset 'techchrunch.csv'. We assume that the composite key is {company, round} where round denotes the given round for funding.

1. Why is {company, round} the key? Why doesn't {company} work?
- Company cannot be the primary key. The purpose of the primary key is to identify data uniquely. Company column is repeated. Even on considering company with column round both create composite key bit using that, the data is not being uniquely identified. For instance, consider rows 16 to 17, even if we combine company and last column round it will not display unique data. It is observed that all the values are identical for each column but the fundedDate and raisedAmt field is different.



16	Facebook	450	web	Palo Alto	CA	1-Oct-07	300000000	USD	c
17	Facebook	450	web	Palo Alto	CA	1-Mar-08	40000000	USD	c
18	Facebook	450	web	Palo Alto	CA	15-Jan-08	15000000	USD	c

1. Does the data satisfy 1NF? Why or Why not?
- First normal form rules are,
    - o Each table should contain a single value
    - o Each record needs to be unique
- Here, the table satisfies the first condition as each column contains single value but it does not satisfy the second constraint. Reason being, some records are redundant in the table for example row number 1454 and 1455 this records are completely identical.
- Hence, the table is not in 1NF.



1454	ZoomInfo	80	web	Waltham	MA	1-Jul-04	7000000	USD	a
1455	ZoomInfo	80	web	Waltham	MA	1-Jul-04	7000000	USD	a

2. Does the data satisfy 2NF? Why or Why not?
- Second Normal Form rules are,
    - o Be in 1NF
    - o Single column Primary Key
- As we proved earlier the table is not in 1NF then it cannot be in second normal form(2NF).

3. Does the data satisfy 3NF? Why or Why not?
- Third Normal Form rules,
    - o Be in 2NF
    - o It should not have transitive dependencies
- The reason is same as previous table is not in 2NF it cannot be in 3NF

The answer is "no" this given table cannot be in any of the normal forms, to convert given table into normal form here are the steps,
- Solving problem becomes easy when issues were identified earlier. In order to convert given table in to 1NF.
- We need to filter the data meaning need to clean the records which are redundant.
    - o First thing to do is store the given dataset into new table assuming table name as "New_Techcrunch"
    - o Secondly, remove all the records which are duplicate
    - o Thirdly, Store that unique records into assuming table "Unique_Data_Techrunch"
    - o Finally, if we create relationship between "Unique_Data_Techrunch" and "techcrunch" it will be One to Many

Entity Relationship Diagram to bring dataset in to 3NF:

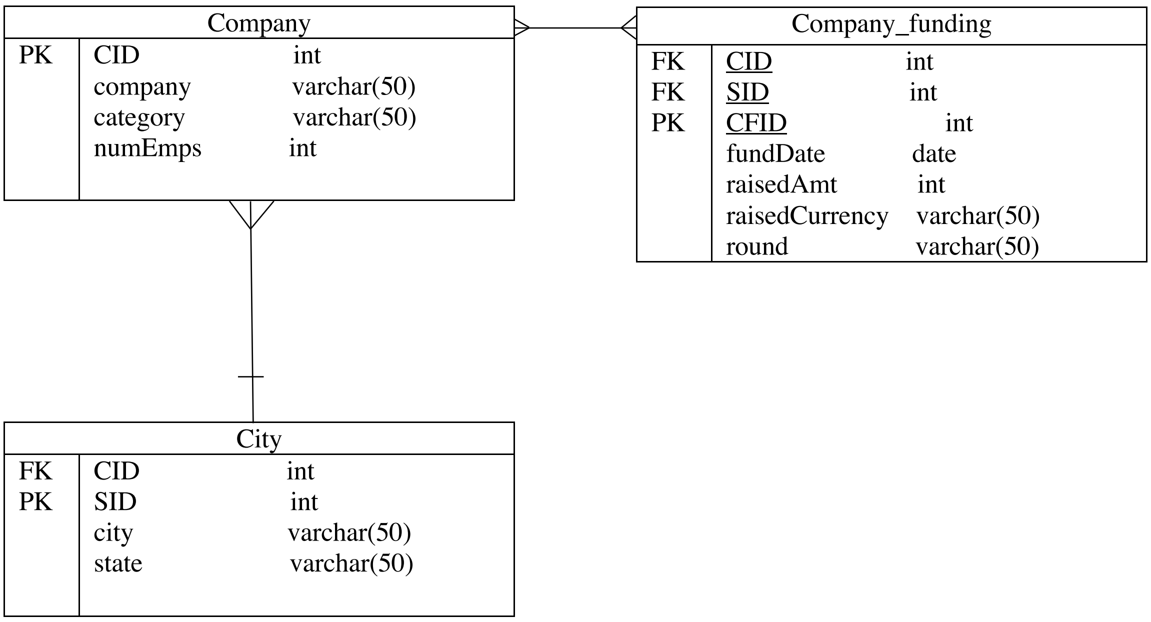
1NF:

- After converting dataset in to 1NF the table columns will look like as follows

Unique_Data_Techrunch		
	company	varchar(50)
	numEmps	int
	category	varchar(50)
	state	varchar(50)
	fundedDate	date
	raisedAmt	int
	raisedCurrency	varchar(50)
	round	varchar(50)

2NF:

- As we discussed earlier, the table should be in 1NF and it should have single column primary key which will be denoted by PK in the ER diagram.
- Foreign key will be used to create a relationship which will be denoted by FK.



Company		
PK	CID	int
	company	varchar(50)
	category	varchar(50)
	numEmps	int

Company_funding		
FK	CID	int
FK	SID	int
PK	CFID	int
	fundDate	date
	raisedAmt	int
	raisedCurrency	varchar(50)
	round	varchar(50)

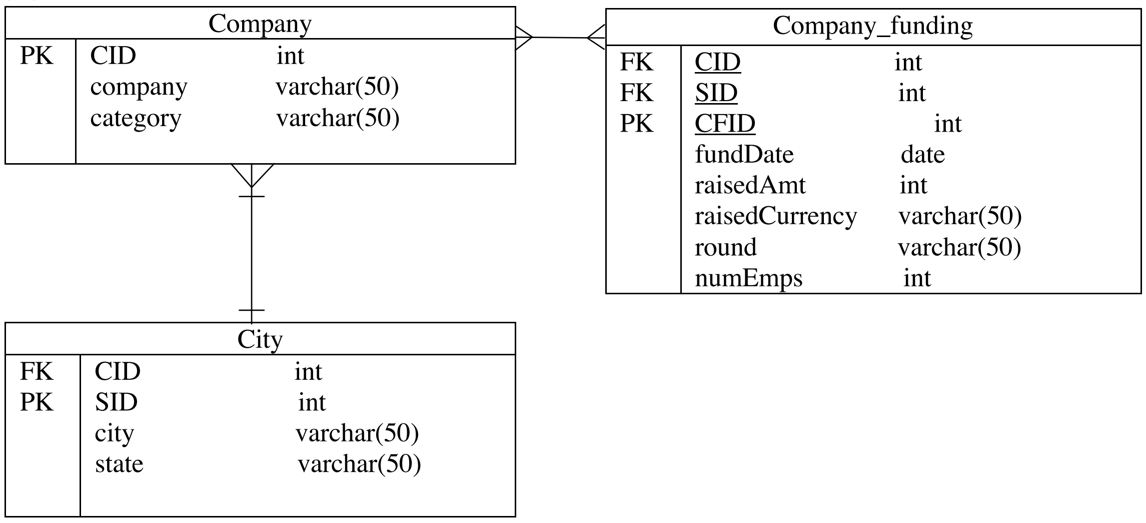
City		
FK	CID	int
PK	SID	int
	city	varchar(50)
	state	varchar(50)

- Company and Company_funding has Many to Many relationships. It can be interpreted that one company can make multiple rounds of funding's so the relation should be One to Many. Even though according to the records many cities have many funds records, so their relation is Many to Many.
- Company to City has Many to One relationship as multiple company can be in same city.
- City and company have One to Many relationships as one city has many companies and those companies have many rounds of findings.

3NF:

- In third normal form table should be in 2NF and that should not have any functional dependency.
- After analyzing 2NF it is apparent that company is dependent upon the number of employees so there should not be any transitive dependency so during this step we will remove it.

3NF



Company

PK	CID	int
	company	varchar(50)
	category	varchar(50)

Company_funding

FK	CID	int
FK	SID	int
PK	CFID	int
	fundDate	date
	raisedAmt	int
	raisedCurrency	varchar(50)
	round	varchar(50)
	numEmps	int

City

FK	CID	int
PK	SID	int
	city	varchar(50)
	state	varchar(50)

- As we can observe from the ‘Company’ table there is no record of employee and there is no transitive dependency now we can say that the assumed table “Unique_Data_Techrunch”(which has unique records) is 3NF.

Problem 3- Case study

Problem statement:

Make an application which works as intermediator between grocery store and customer. The application must allow the customer to choose any three out of five stores. The items should display before the customer and can order them. Payment should be done beforehand.

If the manager wants to contact grocery store, then he should have enough details to contact the store.
On the other end if manager want's to contact grocery store then he/she would be able to contact with enough details.

List of tables being used:
customers, checkout, checkout_action, items, employee, store, inventory, and contact_manager

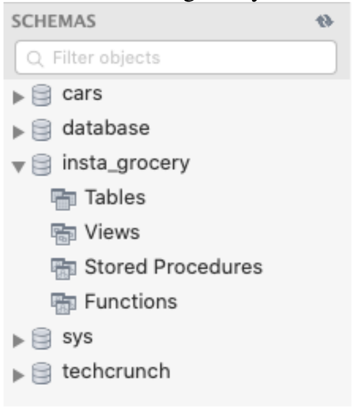
Few assumptions being made which are as follows,

1. All the tables data are not real data
2. Customer has checked out items randomly
3. Approximately 9 stores were assumed that has different items in inventory
4. Each product price, cost subtotal, weight, brand and many more are being assumed.
5. In employee table column 'Pay_E_M' fields have boolean values 0 represents employee and 1 represents manager.
6. During purchase there is no restriction on count of item.

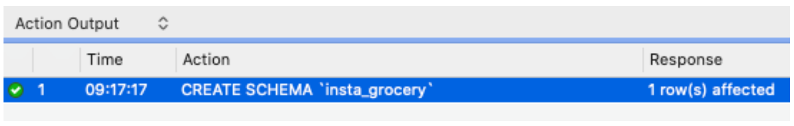
Input query:

```
/* Creating schema `insta_grocery` */
CREATE SCHEMA `insta_grocery`;
```

Output:
- Schema `insta_grocery` is created



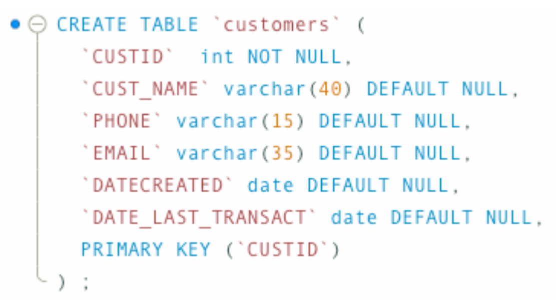
```
SCHEMAS
Filter objects
▶ cars
▶ database
▼ insta_grocery
    Tables
    Views
    Stored Procedures
    Functions
▶ sys
▶ techcrunch
```



```
Action Output
     Time      Action                                  Response
✓ 1  09:17:17  CREATE SCHEMA `insta_grocery`           1 row(s) affected
```

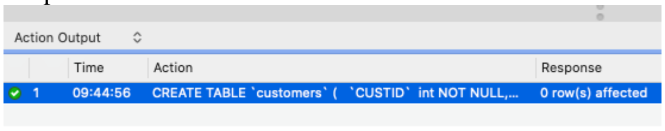
Creating ‘customers’ table

Input query:



```
CREATE TABLE `customers` (
  `CUSTID`  int NOT NULL,
  `CUST_NAME` varchar(40) DEFAULT NULL,
  `PHONE` varchar(15) DEFAULT NULL,
  `EMAIL` varchar(35) DEFAULT NULL,
  `DATECREATED` date DEFAULT NULL,
  `DATE_LAST_TRANSACT` date DEFAULT NULL,
  PRIMARY KEY (`CUSTID`)
) ;
```

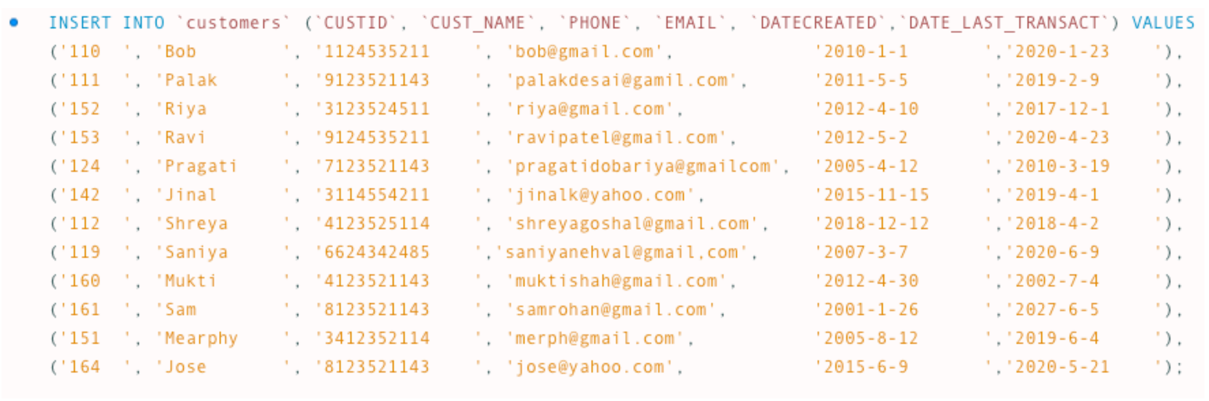
Output:



```
Action Output
   Time      Action                                                Response
✓ 1  09:44:56  CREATE TABLE `customers` (  `CUSTID` int NOT NULL,...   0 row(s) affected
```

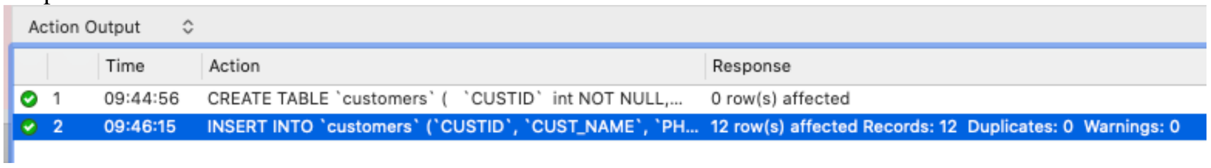
Insertion into `customers`

Input query:



```
INSERT INTO `customers` (`CUSTID`, `CUST_NAME`, `PHONE`, `EMAIL`, `DATECREATED`,`DATE_LAST_TRANSACT`) VALUES
('110  ', 'Bob       ', '1124535211    ', 'bob@gmail.com',            '2010-1-1       ','2020-1-23    '),
('111  ', 'Palak     ', '9123521143    ', 'palakdesai@gamil.com',     '2011-5-5       ','2019-2-9     '),
('152  ', 'Riya      ', '3123524511    ', 'riya@gmail.com',           '2012-4-10      ','2017-12-1    '),
('153  ', 'Ravi      ', '9124535211    ', 'ravipatel@gmail.com',      '2012-5-2       ','2020-4-23    '),
('124  ', 'Pragati   ', '7123521143    ', 'pragatidobariya@gmailcom', '2005-4-12      ','2010-3-19    '),
('142  ', 'Jinal     ', '3114554211    ', 'jinalk@yahoo.com',         '2015-11-15     ','2019-4-1     '),
('112  ', 'Shreya    ', '4123525114    ', 'shreyagoshal@gmail.com',   '2018-12-12     ','2018-4-2     '),
('119  ', 'Saniya    ', '6624342485    ','saniyanehval@gmail.com',    '2007-3-7       ','2020-6-9     '),
('160  ', 'Mukti     ', '4123521143    ', 'muktishah@gmail.com',      '2012-4-30      ','2002-7-4     '),
('161  ', 'Sam       ', '8123521143    ', 'samrohan@gmail.com',       '2001-1-26      ','2027-6-5     '),
('151  ', 'Mearphy   ', '3412352114    ', 'merph@gmail.com',          '2005-8-12      ','2019-6-4     '),
('164  ', 'Jose      ', '8123521143    ', 'jose@yahoo.com',           '2015-6-9       ','2020-5-21    ');
```

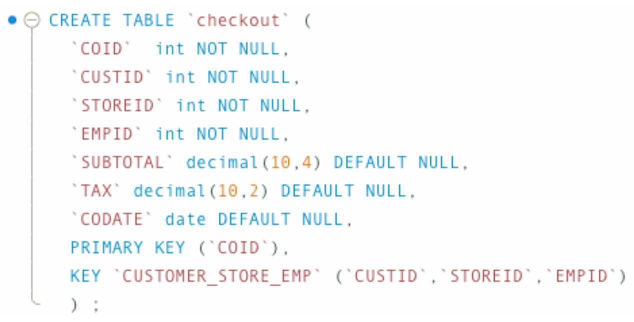
Output:



```
Action Output
   Time      Action                                                  Response
✓ 1  09:44:56  CREATE TABLE `customers` (  `CUSTID` int NOT NULL,...     0 row(s) affected
✓ 2  09:46:15  INSERT INTO `customers` (`CUSTID`, `CUST_NAME`, `PH...   12 row(s) affected Records: 12  Duplicates: 0  Warnings: 0
```

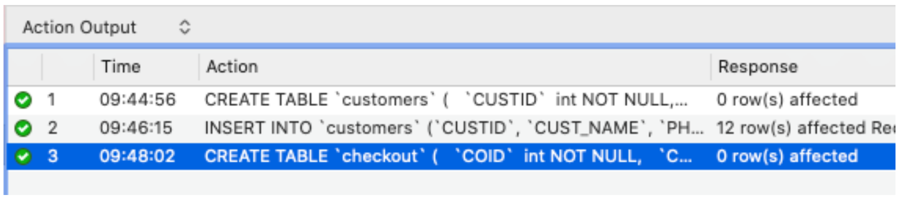
Creating table `checkout`

Input query:



```
CREATE TABLE `checkout` (
  `COID`  int NOT NULL,
  `CUSTID` int NOT NULL,
  `STOREID` int NOT NULL,
  `EMPID` int NOT NULL,
  `SUBTOTAL` decimal(10,4) DEFAULT NULL,
  `TAX` decimal(10,2) DEFAULT NULL,
  `CODATE` date DEFAULT NULL,
  PRIMARY KEY (`COID`),
  KEY `CUSTOMER_STORE_EMP` (`CUSTID`,`STOREID`,`EMPID`)
  ) ;
```

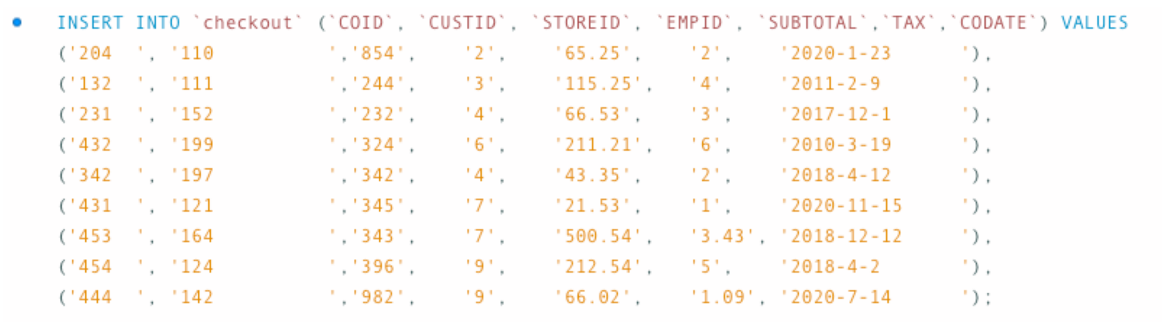
Output:



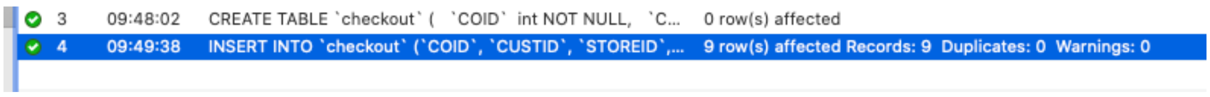
```
Action Output
     Time      Action                                                      Response
1    09:44:56  CREATE TABLE `customers` (  `CUSTID` int NOT NULL,...      0 row(s) affected
2    09:46:15  INSERT INTO `customers` (`CUSTID`, `CUST_NAME`, `PH...    12 row(s) affected Re
3    09:48:02  CREATE TABLE `checkout` (  `COID` int NOT NULL,  `C...     0 row(s) affected
```

Insertion into `checkout`

Input query:



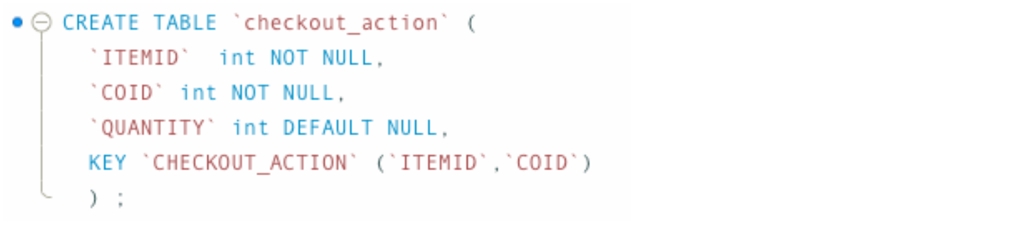
```
INSERT INTO `checkout` (`COID`, `CUSTID`, `STOREID`, `EMPID`, `SUBTOTAL`,`TAX`,`CODATE`) VALUES
('204  ', '110          ','854',  '2',   '65.25',   '2',   '2020-1-23     '),
('132  ', '111          ','244',  '3',   '115.25',  '4',   '2011-2-9      '),
('231  ', '152          ','232',  '4',   '66.53',   '3',   '2017-12-1     '),
('432  ', '199          ','324',  '6',   '211.21',  '6',   '2010-3-19     '),
('342  ', '197          ','342',  '4',   '43.35',   '2',   '2018-4-12     '),
('431  ', '121          ','345',  '7',   '21.53',   '1',   '2020-11-15    '),
('453  ', '164          ','343',  '7',   '500.54',  '3.43', '2018-12-12    '),
('454  ', '124          ','396',  '9',   '212.54',  '5',   '2018-4-2      '),
('444  ', '142          ','982',  '9',   '66.02',   '1.09', '2020-7-14     ');
```



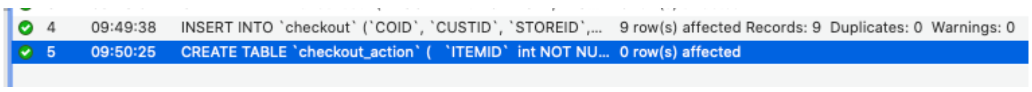
```
3    09:48:02  CREATE TABLE `checkout` (  `COID` int NOT NULL,  `C...     0 row(s) affected
4    09:49:38  INSERT INTO `checkout` (`COID`, `CUSTID`, `STOREID`,...    9 row(s) affected Records: 9  Duplicates: 0  Warnings: 0
```

Creating table `checkout_action`

Input query:



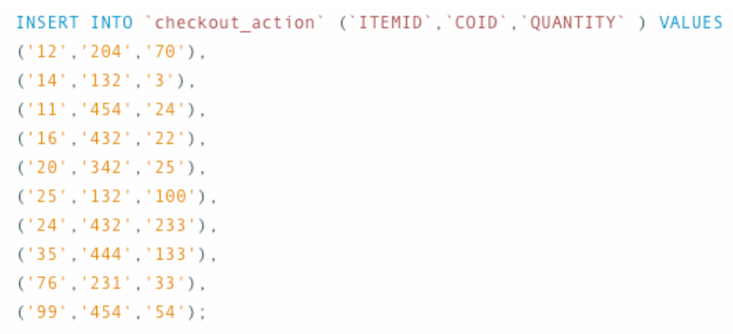
```
CREATE TABLE `checkout_action` (
  `ITEMID`  int NOT NULL,
  `COID` int NOT NULL,
  `QUANTITY` int DEFAULT NULL,
  KEY `CHECKOUT_ACTION` (`ITEMID`,`COID`)
  ) ;
```



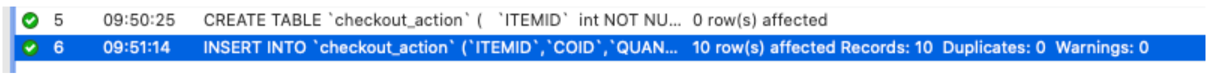
```
4    09:49:38  INSERT INTO `checkout` (`COID`, `CUSTID`, `STOREID`,...    9 row(s) affected Records: 9  Duplicates: 0  Warnings: 0
5    09:50:25  CREATE TABLE `checkout_action` (  `ITEMID` int NOT NU...   0 row(s) affected
```

Insertion into `checkout_action`

Input query:



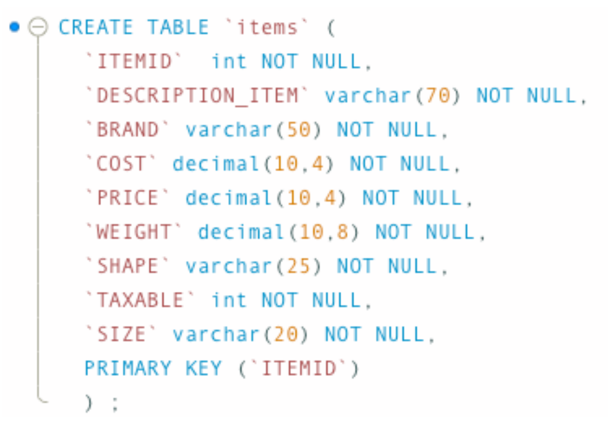
```
INSERT INTO `checkout_action` (`ITEMID`,`COID`,`QUANTITY` ) VALUES
('12','204','70'),
('14','132','3'),
('11','454','24'),
('16','432','22'),
('20','342','25'),
('25','132','100'),
('24','432','233'),
('35','444','133'),
('76','231','33'),
('99','454','54');
```



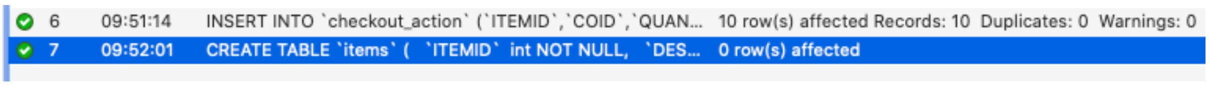
```
5    09:50:25  CREATE TABLE `checkout_action` (  `ITEMID` int NOT NU...   0 row(s) affected
6    09:51:14  INSERT INTO `checkout_action` (`ITEMID`,`COID`,`QUAN...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
```

Creating table `items`

Input query:



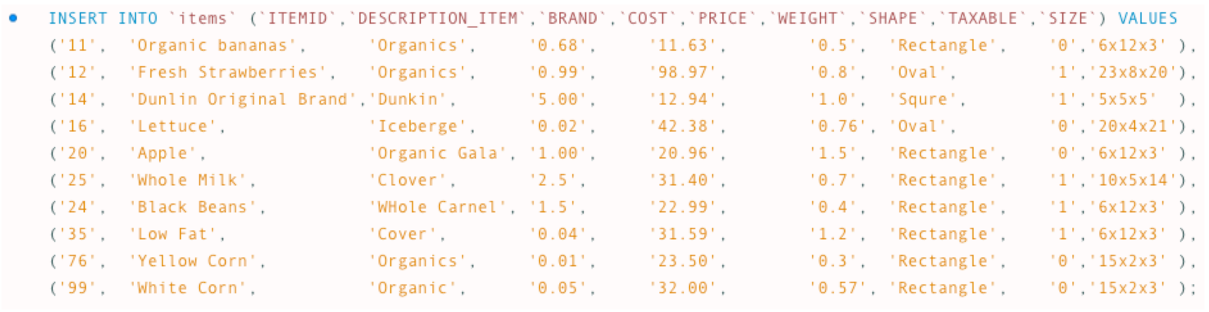
```
CREATE TABLE `items` (
  `ITEMID`  int NOT NULL,
  `DESCRIPTION_ITEM` varchar(70) NOT NULL,
  `BRAND` varchar(50) NOT NULL,
  `COST` decimal(10,4) NOT NULL,
  `PRICE` decimal(10,4) NOT NULL,
  `WEIGHT` decimal(10,8) NOT NULL,
  `SHAPE` varchar(25) NOT NULL,
  `TAXABLE` int NOT NULL,
  `SIZE` varchar(20) NOT NULL,
  PRIMARY KEY (`ITEMID`)
  ) ;
```



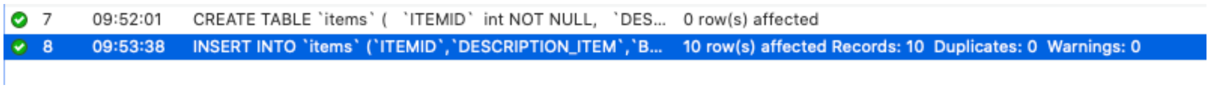
```
6   09:51:14   INSERT INTO `checkout_action` (`ITEMID`,`COID`,`QUAN...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
7   09:52:01   CREATE TABLE `items` (  `ITEMID` int NOT NULL,  `DES...   0 row(s) affected
```

Insertion into `items`

Input query:



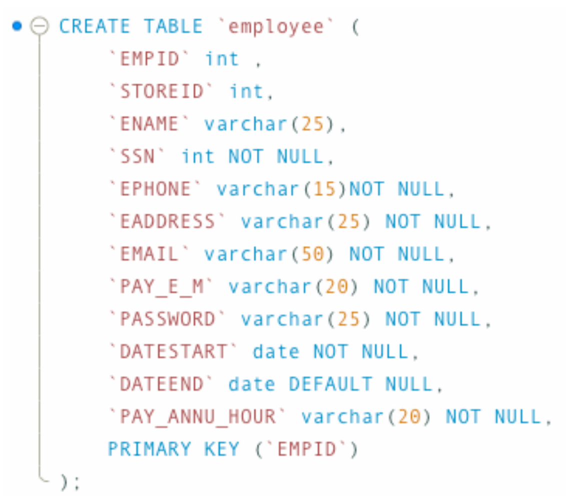
```
INSERT INTO `items` (`ITEMID`,`DESCRIPTION_ITEM`,`BRAND`,`COST`,`PRICE`,`WEIGHT`,`SHAPE`,`TAXABLE`,`SIZE`) VALUES
('11',  'Organic bananas',       'Organics',     '0.68',    '11.63',        '0.5',  'Rectangle',    '0','6x12x3' ),
('12',  'Fresh Strawberries',    'Organics',     '0.99',    '98.97',        '0.8',  'Oval',         '1','23x8x20'),
('14',  'Dunlin Original Brand','Dunkin',        '5.00',    '12.94',        '1.0',  'Squre',        '1','5x5x5'  ),
('16',  'Lettuce',               'Iceberge',     '0.02',    '42.38',        '0.76', 'Oval',         '0','20x4x21'),
('20',  'Apple',                 'Organic Gala', '1.00',    '20.96',        '1.5',  'Rectangle',    '0','6x12x3' ),
('25',  'Whole Milk',            'Clover',       '2.5',     '31.40',        '0.7',  'Rectangle',    '1','10x5x14'),
('24',  'Black Beans',           'WHole Carnel', '1.5',     '22.99',        '0.4',  'Rectangle',    '1','6x12x3' ),
('35',  'Low Fat',               'Cover',        '0.04',    '31.59',        '1.2',  'Rectangle',    '1','6x12x3' ),
('76',  'Yellow Corn',           'Organics',     '0.01',    '23.50',        '0.3',  'Rectangle',    '0','15x2x3' ),
('99',  'White Corn',            'Organic',      '0.05',    '32.00',        '0.57', 'Rectangle',    '0','15x2x3' );
```



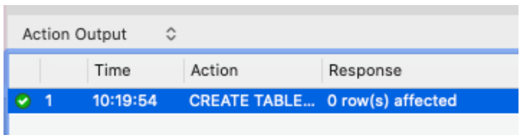
```
7   09:52:01   CREATE TABLE `items` (  `ITEMID` int NOT NULL,  `DES...   0 row(s) affected
8   09:53:38   INSERT INTO `items` (`ITEMID`,`DESCRIPTION_ITEM`,`B...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
```

Creating table `employee`

Input query:



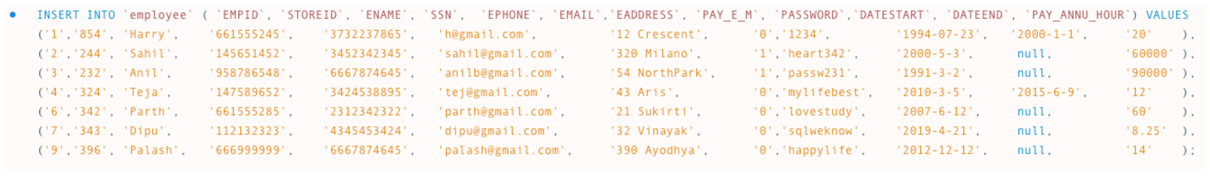
```
CREATE TABLE `employee` (
    `EMPID` int ,
    `STOREID` int,
    `ENAME` varchar(25),
    `SSN` int NOT NULL,
    `EPHONE` varchar(15)NOT NULL,
    `EADDRESS` varchar(25) NOT NULL,
    `EMAIL` varchar(50) NOT NULL,
    `PAY_E_M` varchar(20) NOT NULL,
    `PASSWORD` varchar(25) NOT NULL,
    `DATESTART` date NOT NULL,
    `DATEEND` date DEFAULT NULL,
    `PAY_ANNU_HOUR` varchar(20) NOT NULL,
    PRIMARY KEY (`EMPID`)
);
```



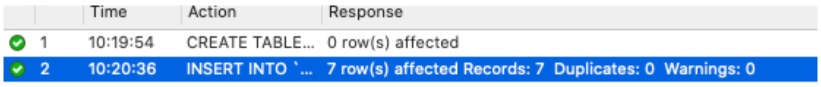
```
Action Output

   Time       Action           Response
1  10:19:54   CREATE TABLE...  0 row(s) affected
```

Insertion into `employee`

Input query:



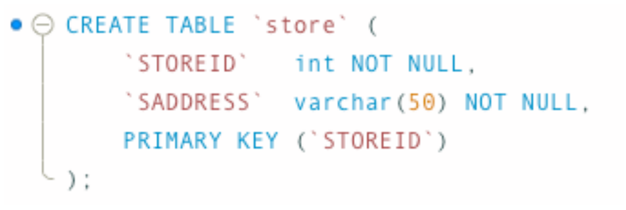
```
INSERT INTO `employee` ( `EMPID`, `STOREID`, `ENAME`, `SSN`,  `EPHONE`, `EMAIL`,`EADDRESS`, `PAY_E_M`, `PASSWORD`,`DATESTART`, `DATEEND`, `PAY_ANNU_HOUR`) VALUES
('1','854', 'Harry',    '661555245',    '3732237865',   'h@gmail.com',          '12 Crescent',     '0','1234',          '1994-07-23',   '2000-1-1',     '20'    ),
('2','244', 'Sahil',    '145651452',    '3452342345',   'sahil@gmail.com',      '320 Milano',      '1','heart342',      '2000-5-3',      null,          '60000' ),
('3','232', 'Anil',     '958786548',    '6667874645',   'anilb@gmail.com',      '54 NorthPark',    '1','passw231',      '1991-3-2',      null,          '90000' ),
('4','324', 'Teja',     '147589652',    '3424538895',   'tej@gmail.com',        '43 Aris',         '0','mylifebest',    '2010-3-5',     '2015-6-9',     '12'    ),
('6','342', 'Parth',    '661555285',    '2312342322',   'parth@gmail.com',      '21 Sukirti',      '0','lovestudy',     '2007-6-12',     null,          '60'    ),
('7','343', 'Dipu',     '112132323',    '4345453424',   'dipu@gmail.com',       '32 Vinayak',      '0','sqlweknow',     '2019-4-21',     null,          '8.25'  ),
('9','396', 'Palash',   '666999999',    '6667874645',   'palash@gmail.com',     '390 Ayodhya',     '0','happylife',     '2012-12-12',    null,          '14'    );
```



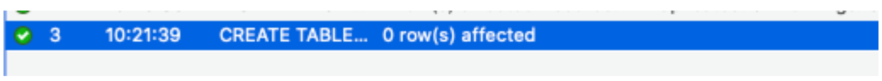
```
   Time       Action           Response
1  10:19:54   CREATE TABLE...  0 row(s) affected
2  10:20:36   INSERT INTO `... 7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0
```

Creating table `store `

Input query:



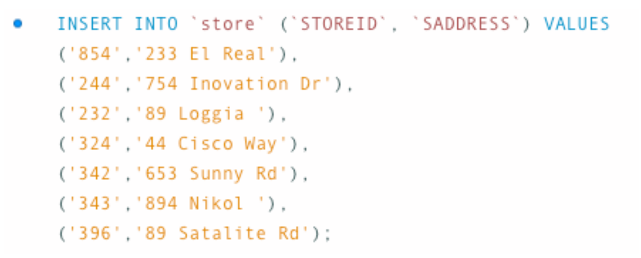
```
CREATE TABLE `store` (
    `STOREID`   int NOT NULL,
    `SADDRESS`  varchar(50) NOT NULL,
    PRIMARY KEY (`STOREID`)
);
```



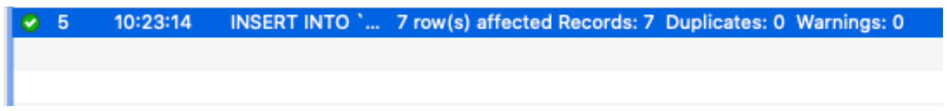
```
3  10:21:39   CREATE TABLE...  0 row(s) affected
```

Insertion into `store`

Input query:



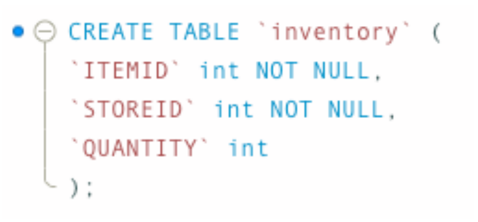
```
INSERT INTO `store` (`STOREID`, `SADDRESS`) VALUES
('854','233 El Real'),
('244','754 Inovation Dr'),
('232','89 Loggia '),
('324','44 Cisco Way'),
('342','653 Sunny Rd'),
('343','894 Nikol '),
('396','89 Satalite Rd');
```



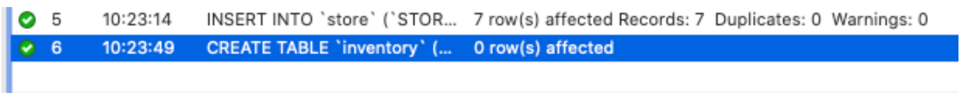
```
5  10:23:14   INSERT INTO `... 7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0
```

Creating table `inventory`

Input query:



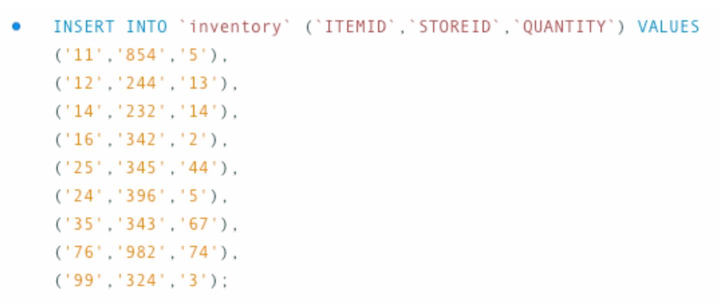
```
CREATE TABLE `inventory` (
`ITEMID` int NOT NULL,
`STOREID` int NOT NULL,
`QUANTITY` int
);
```



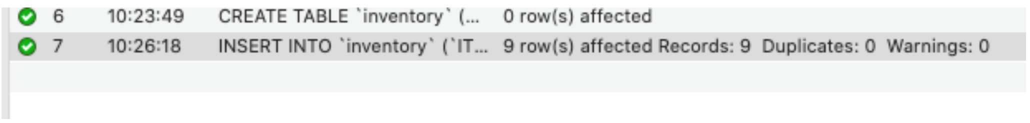
```
5    10:23:14    INSERT INTO `store` (`STOR...    7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0
6    10:23:49    CREATE TABLE `inventory` (...    0 row(s) affected
```

Insertion into `inventory`

Input query:



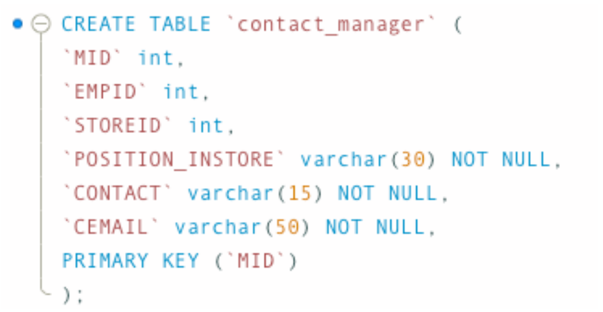
```
INSERT INTO `inventory` (`ITEMID`,`STOREID`,`QUANTITY`) VALUES
('11','854','5'),
('12','244','13'),
('14','232','14'),
('16','342','2'),
('25','345','44'),
('24','396','5'),
('35','343','67'),
('76','982','74'),
('99','324','3');
```



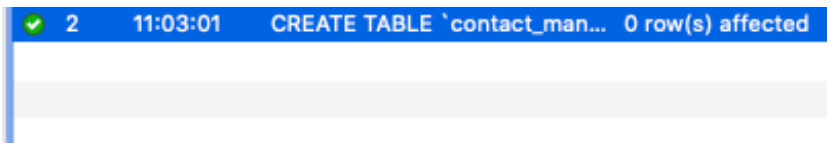
```
6    10:23:49    CREATE TABLE `inventory` (...    0 row(s) affected
7    10:26:18    INSERT INTO `inventory` (`IT...   9 row(s) affected Records: 9  Duplicates: 0  Warnings: 0
```

Creating table `contact_manager`

Input query:



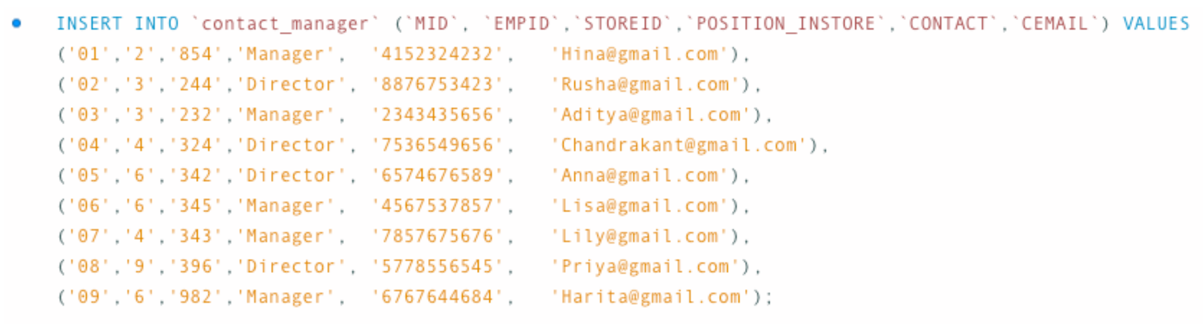
```
CREATE TABLE `contact_manager` (
`MID` int,
`EMPID` int,
`STOREID` int,
`POSITION_INSTORE` varchar(30) NOT NULL,
`CONTACT` varchar(15) NOT NULL,
`CEMAIL` varchar(50) NOT NULL,
PRIMARY KEY (`MID`)
);
```



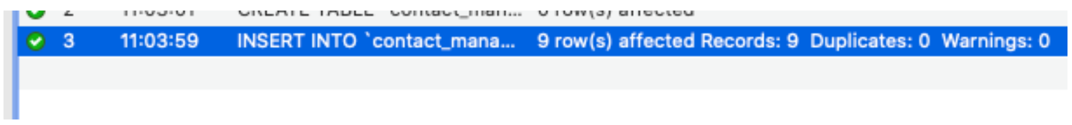
```
2    11:03:01    CREATE TABLE `contact_man...    0 row(s) affected
```

Insertion into `customer_manager`

Input query:



```
• INSERT INTO `contact_manager` (`MID`, `EMPID`,`STOREID`,`POSITION_INSTORE`,`CONTACT`,`CEMAIL`) VALUES
  ('01','2','854','Manager',  '4152324232',   'Hina@gmail.com'),
  ('02','3','244','Director', '8876753423',   'Rusha@gmail.com'),
  ('03','3','232','Manager',  '2343435656',   'Aditya@gmail.com'),
  ('04','4','324','Director', '7536549656',   'Chandrakant@gmail.com'),
  ('05','6','342','Director', '6574676589',   'Anna@gmail.com'),
  ('06','6','345','Manager',  '4567537857',   'Lisa@gmail.com'),
  ('07','4','343','Manager',  '7857675676',   'Lily@gmail.com'),
  ('08','9','396','Director', '5778556545',   'Priya@gmail.com'),
  ('09','6','982','Manager',  '6767644684',   'Harita@gmail.com');
```

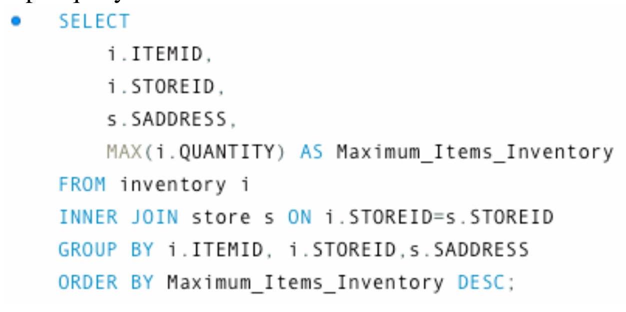


```
2   11:03:01   CREATE TABLE `contact_man...   0 row(s) affected
3   11:03:59   INSERT INTO `contact_mana...   9 row(s) affected Records: 9  Duplicates: 0  Warnings: 0
```

Generating 10 different report that grocery management and customers to have.

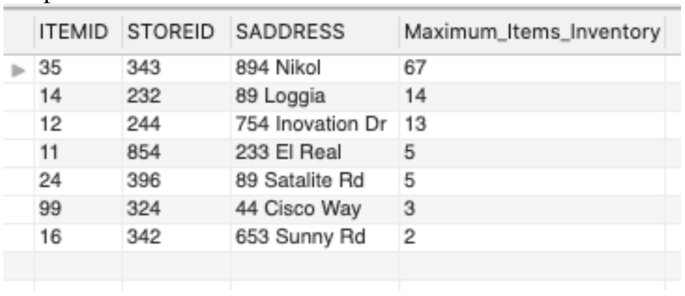
Q1. Which grocery store has the largest quantity in inventory

Input query:

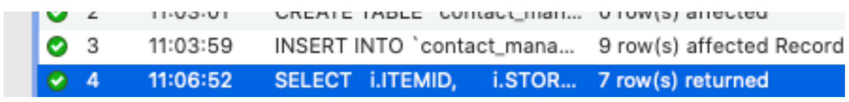


```
• SELECT
      i.ITEMID,
      i.STOREID,
      s.SADDRESS,
      MAX(i.QUANTITY) AS Maximum_Items_Inventory
  FROM inventory i
  INNER JOIN store s ON i.STOREID=s.STOREID
  GROUP BY i.ITEMID, i.STOREID,s.SADDRESS
  ORDER BY Maximum_Items_Inventory DESC;
```

Output:



ITEMID	STOREID	SADDRESS	Maximum_Items_Inventory
35	343	894 Nikol	67
14	232	89 Loggia	14
12	244	754 Inovation Dr	13
11	854	233 El Real	5
24	396	89 Satalite Rd	5
99	324	44 Cisco Way	3
16	342	653 Sunny Rd	2



```
2   11:03:01   CREATE TABLE `contact_man...   0 row(s) affected
3   11:03:59   INSERT INTO `contact_mana...   9 row(s) affected Record
4   11:06:52   SELECT  i.ITEMID,   i.STOR...   7 row(s) returned
```

343 Store has the maximum number of inventories. 342 store has the list, using this information the manager can request to update their store inventory.

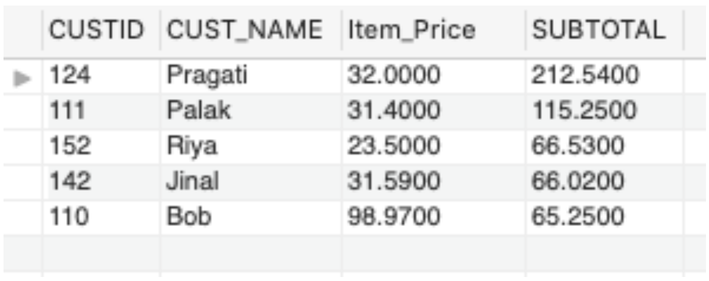
It can also be interpreted that most of the customers are going to 894 Nikol store may that store is closer to customer to pick up their order.

Q2. How many items are more than $20 and customer has checked out?

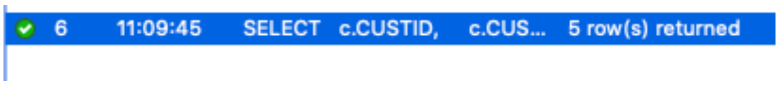
Input query:

```
SELECT
    c.CUSTID,
    c.CUST_NAME,
    i.PRICE AS Item_Price,
    co.SUBTOTAL
FROM items i
INNER JOIN checkout_action ca ON i.ITEMID=ca.ITEMID
INNER JOIN checkout co ON co.COID=ca.COID
INNER JOIN customers c ON c.CUSTID=co.CUSTID
WHERE
    i.PRICE > 20
GROUP BY c.CUSTID,c.CUST_NAME, i.PRICE,co.SUBTOTAL
ORDER BY co.SUBTOTAL DESC;
```

Output:



CUSTID	CUST_NAME	Item_Price	SUBTOTAL
124	Pragati	32.0000	212.5400
111	Palak	31.4000	115.2500
152	Riya	23.5000	66.5300
142	Jinal	31.5900	66.0200
110	Bob	98.9700	65.2500



```
6    11:09:45    SELECT  c.CUSTID,    c.CUS...   5 row(s) returned
```

There are five customers who have checked-out the products which cost more than $20. The maximum purchase is done by customer named pragati with item cost 32$.

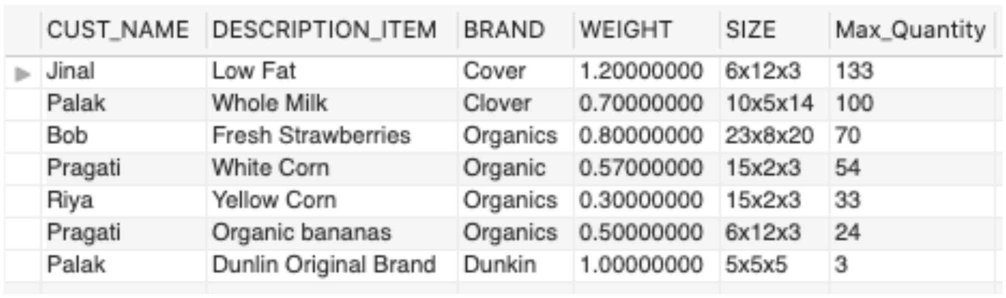
From this information business insider may know what is the maximum sub total with item price. This will help him decide whether there should be an increment or decrement in the price.

Q3. Which item is ordered by customer with maximum quantity?

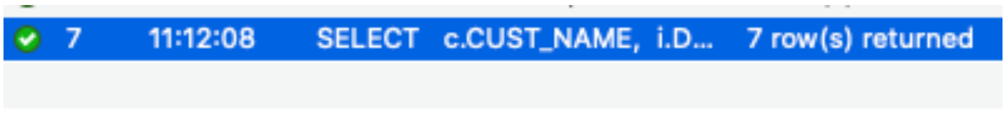
Input query:

```
SELECT
    c.CUST_NAME,
    i.DESCRIPTION_ITEM,
    i.BRAND,
    i.WEIGHT,
    i.SIZE,
    MAX(ca.QUANTITY) as Max_Quantity
FROM items i
INNER JOIN checkout_action ca ON i.ITEMID=ca.ITEMID
INNER JOIN checkout co ON co.COID=ca.COID
INNER JOIN customers c ON c.CUSTID=co.CUSTID
GROUP BY  c.CUST_NAME, i.DESCRIPTION_ITEM, i.BRAND, i.WEIGHT, i.SIZE
ORDER BY Max_Quantity DESC;
```

Output:



	CUST_NAME	DESCRIPTION_ITEM	BRAND	WEIGHT	SIZE	Max_Quantity
▶	Jinal	Low Fat	Cover	1.20000000	6x12x3	133
	Palak	Whole Milk	Clover	0.70000000	10x5x14	100
	Bob	Fresh Strawberries	Organics	0.80000000	23x8x20	70
	Pragati	White Corn	Organic	0.57000000	15x2x3	54
	Riya	Yellow Corn	Organics	0.30000000	15x2x3	33
	Pragati	Organic bananas	Organics	0.50000000	6x12x3	24
	Palak	Dunlin Original Brand	Dunkin	1.00000000	5x5x5	3



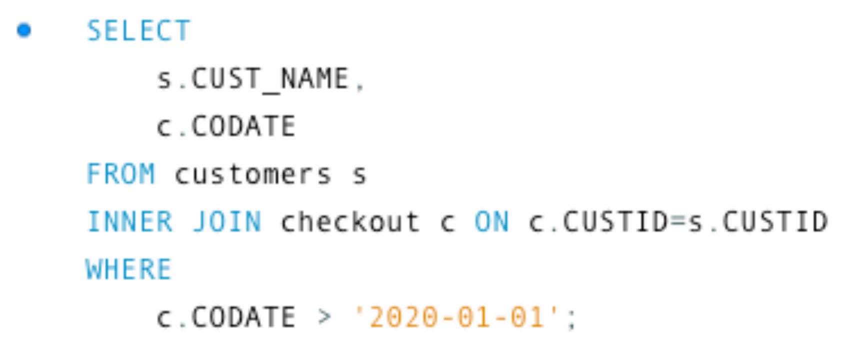
```
✓ 7    11:12:08    SELECT  c.CUST_NAME,  i.D...   7 row(s) returned
```

The above result shows the details of the item. Which customer has ordered a particular product the maximum time. Here the customer 'Jinal' has ordered 'Low Fat' 133 times which is the max.

Out of all purchases most of people has ordered product which is of brand organics

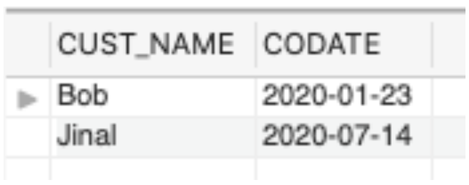
Q4. How many customers has check out in 2020-JAN

Input query:

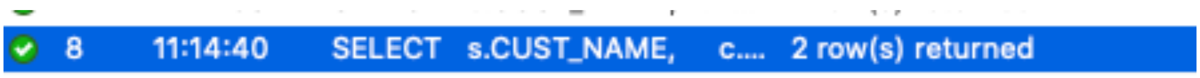


```
SELECT
    s.CUST_NAME,
    c.CODATE
FROM customers s
INNER JOIN checkout c ON c.CUSTID=s.CUSTID
WHERE
    c.CODATE > '2020-01-01';
```

Output:



	CUST_NAME	CODATE
▶	Bob	2020-01-23
	Jinal	2020-07-14



```
✓ 8    11:14:40    SELECT  s.CUST_NAME,    c....   2 row(s) returned
```

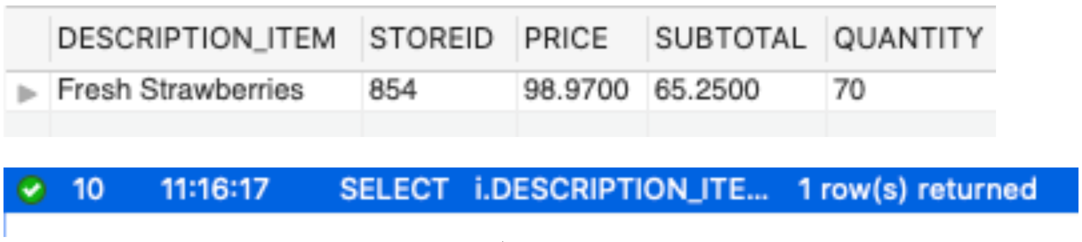
There are only two customers who has checkout after 2020-Jan-01. It shows there must be something to look into.

Q5. List the items that has price more than 70$ and customer bought it

Input query:

```
•   SELECT
        i.DESCRIPTION_ITEM,
        c.STOREID,
        i.PRICE AS PRICE,
        c.SUBTOTAL,
        co.QUANTITY
    FROM items i
    INNER JOIN checkout_action co ON i.ITEMID=co.ITEMID
    INNER JOIN checkout c ON c.COID=co.COID
    WHERE
        i.PRICE > 70.00
    GROUP BY i.DESCRIPTION_ITEM,c.STOREID,i.PRICE,c.SUBTOTAL,co.QUANTITY
    ORDER BY PRICE DESC;
```

Output:



DESCRIPTION_ITEM	STOREID	PRICE	SUBTOTAL	QUANTITY
Fresh Strawberries	854	98.9700	65.2500	70

```
10    11:16:17    SELECT  i.DESCRIPTION_ITE...   1 row(s) returned
```

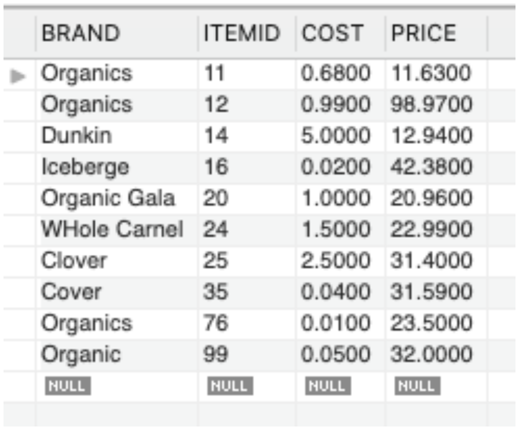
There is only one item which price more than $70 and customer has ordered Fresh Strawberries

Q6. Which item has more than 5% markup price.

Input query:

```
•   SELECT
        BRAND,
        ITEMID,
        COST,
        PRICE
    FROM items
    WHERE
        (COST*1.05) < PRICE;
```

Output:



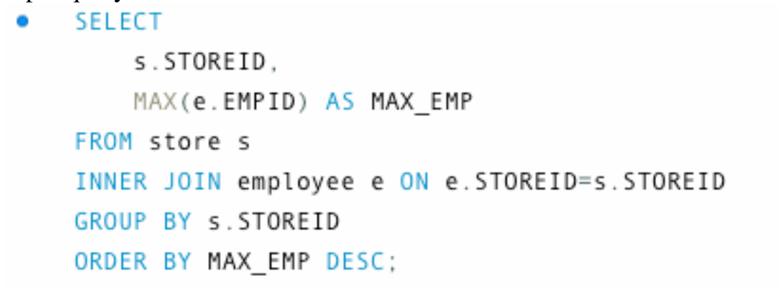
BRAND	ITEMID	COST	PRICE
Organics	11	0.6800	11.6300
Organics	12	0.9900	98.9700
Dunkin	14	5.0000	12.9400
Iceberge	16	0.0200	42.3800
Organic Gala	20	1.0000	20.9600
WHole Camel	24	1.5000	22.9900
Clover	25	2.5000	31.4000
Cover	35	0.0400	31.5900
Organics	76	0.0100	23.5000
Organic	99	0.0500	32.0000
NULL	NULL	NULL	NULL

```
12    11:21:46    SELECT  BRAND,    ITEMID,...    10 row(s) returned
```

The price set for the customer is higher than the 5%markup given to the manufacturing price. Using this key column the owner can manage his profit margin.

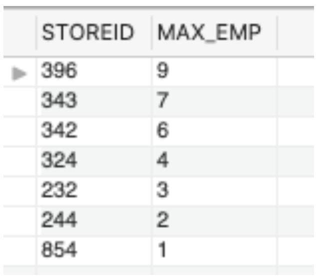
Q7. Which store has maximum number of employees?

Input query:

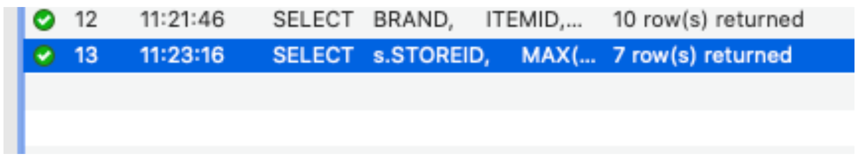


```
•   SELECT
        s.STOREID,
        MAX(e.EMPID) AS MAX_EMP
    FROM store s
    INNER JOIN employee e ON e.STOREID=s.STOREID
    GROUP BY s.STOREID
    ORDER BY MAX_EMP DESC;
```

Output:



STOREID	MAX_EMP
396	9
343	7
342	6
324	4
232	3
244	2
854	1



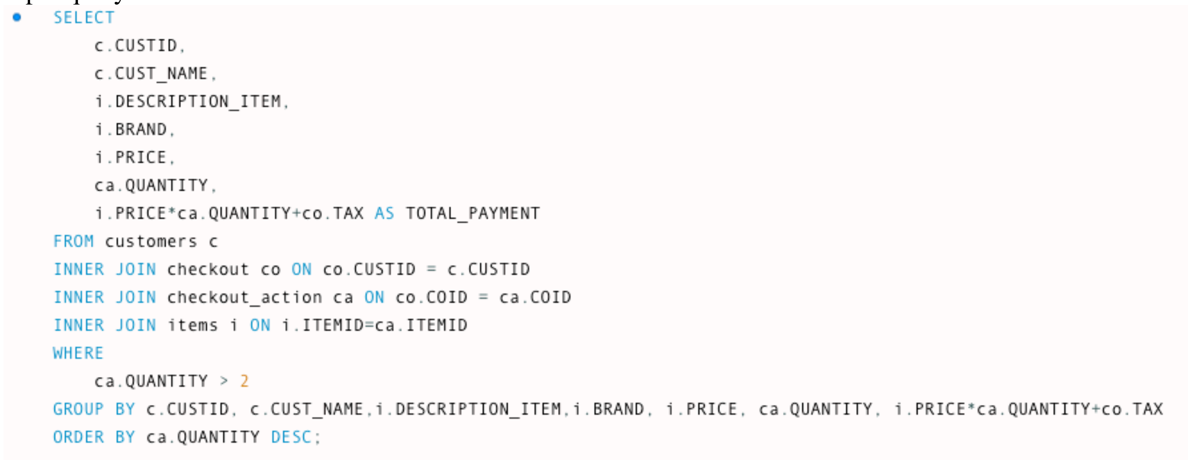
```
12    11:21:46    SELECT  BRAND,     ITEMID,...    10 row(s) returned
13    11:23:16    SELECT  s.STOREID,    MAX(...    7 row(s) returned
```

Maximum number of employees are working for store 396.

This table would be help full to understand the relationship between buyers and store employees meaning that if the store has more number of employees may help to reduce waiting time for customer pick up orders.

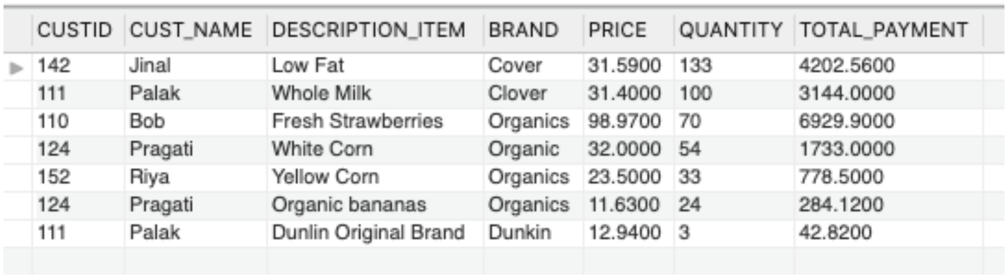
Q8. Provide a list of customers who bought more than 2 items in a single transaction.

Input query:

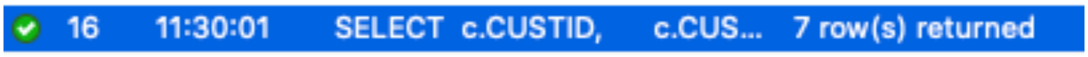


```
•   SELECT
        c.CUSTID,
        c.CUST_NAME,
        i.DESCRIPTION_ITEM,
        i.BRAND,
        i.PRICE,
        ca.QUANTITY,
        i.PRICE*ca.QUANTITY+co.TAX AS TOTAL_PAYMENT
    FROM customers c
    INNER JOIN checkout co ON co.CUSTID = c.CUSTID
    INNER JOIN checkout_action ca ON co.COID = ca.COID
    INNER JOIN items i ON i.ITEMID=ca.ITEMID
    WHERE
        ca.QUANTITY > 2
    GROUP BY c.CUSTID, c.CUST_NAME,i.DESCRIPTION_ITEM,i.BRAND, i.PRICE, ca.QUANTITY, i.PRICE*ca.QUANTITY+co.TAX
    ORDER BY ca.QUANTITY DESC;
```

Output:



CUSTID	CUST_NAME	DESCRIPTION_ITEM	BRAND	PRICE	QUANTITY	TOTAL_PAYMENT
142	Jinal	Low Fat	Cover	31.5900	133	4202.5600
111	Palak	Whole Milk	Clover	31.4000	100	3144.0000
110	Bob	Fresh Strawberries	Organics	98.9700	70	6929.9000
124	Pragati	White Corn	Organic	32.0000	54	1733.0000
152	Riya	Yellow Corn	Organics	23.5000	33	778.5000
124	Pragati	Organic bananas	Organics	11.6300	24	284.1200
111	Palak	Dunlin Original Brand	Dunkin	12.9400	3	42.8200



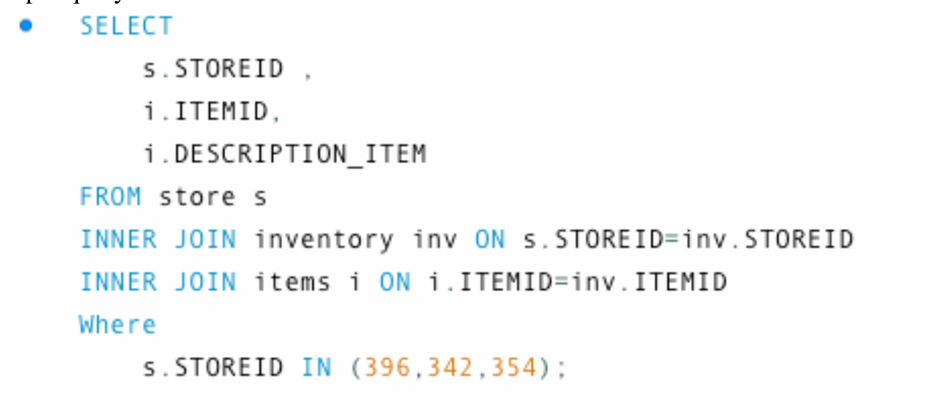
```
16    11:30:01    SELECT  c.CUSTID,    c.CUS...   7 row(s) returned
```

There are 7 more customers who have ordered more than two items in a single transaction.
Total payment can be calculated using the formula (price*quantity)+tax.

This shows the customer the rate at which they are spending and the quantity they are getting.

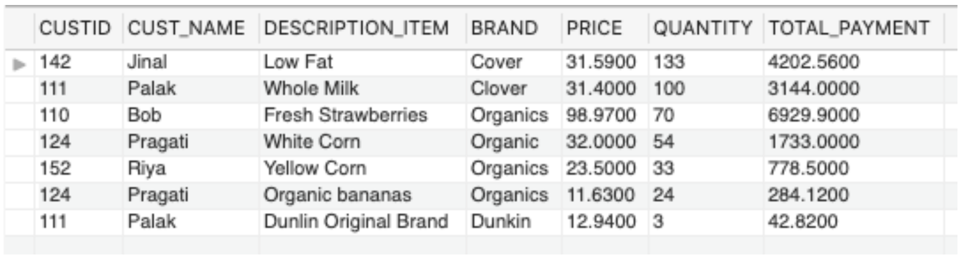
Q9. Provide the list of items from only three stores selected by customer.

Input query:

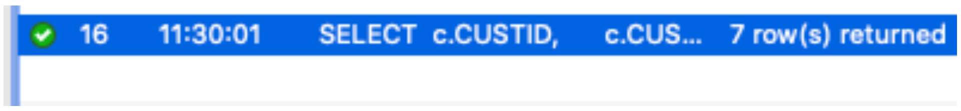


```
SELECT
    s.STOREID ,
    i.ITEMID,
    i.DESCRIPTION_ITEM
FROM store s
INNER JOIN inventory inv ON s.STOREID=inv.STOREID
INNER JOIN items i ON i.ITEMID=inv.ITEMID
Where
    s.STOREID IN (396,342,354);
```

Output:



CUSTID	CUST_NAME	DESCRIPTION_ITEM	BRAND	PRICE	QUANTITY	TOTAL_PAYMENT
142	Jinal	Low Fat	Cover	31.5900	133	4202.5600
111	Palak	Whole Milk	Clover	31.4000	100	3144.0000
110	Bob	Fresh Strawberries	Organics	98.9700	70	6929.9000
124	Pragati	White Corn	Organic	32.0000	54	1733.0000
152	Riya	Yellow Corn	Organics	23.5000	33	778.5000
124	Pragati	Organic bananas	Organics	11.6300	24	284.1200
111	Palak	Dunlin Original Brand	Dunkin	12.9400	3	42.8200

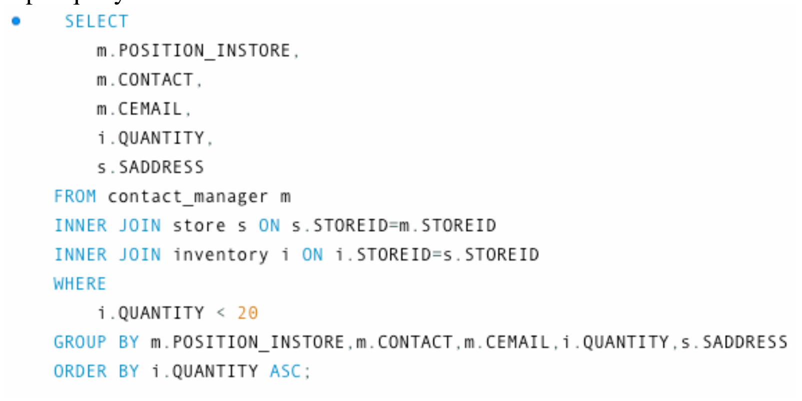


```
16    11:30:01    SELECT  c.CUSTID,    c.CUS...   7 row(s) returned
```

The customer has the liberty to choose his grocery stores. Assuming the customer has selected store number 396,342 and 354. The above table shows only those items of the chosen three store inventories.

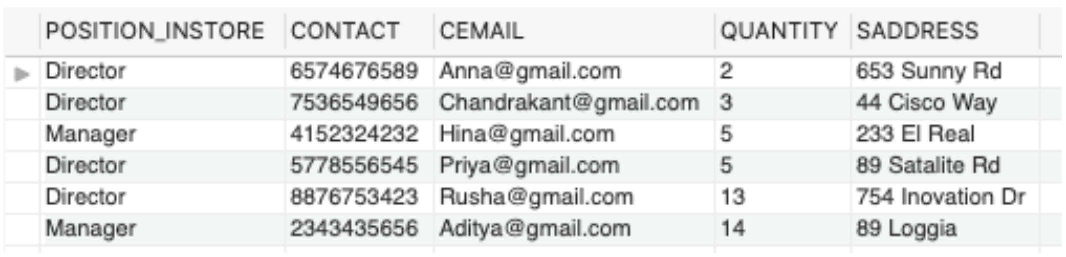
Q10. Provide a list to contact the store which has low items quantity in inventory.

Input query

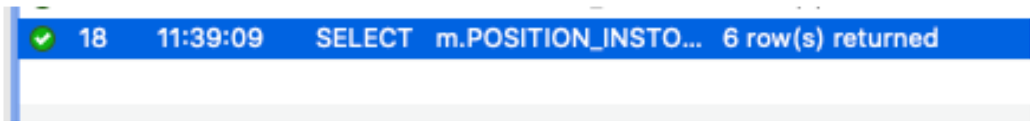


```
•   SELECT
        m.POSITION_INSTORE,
        m.CONTACT,
        m.CEMAIL,
        i.QUANTITY,
        s.SADDRESS
    FROM contact_manager m
    INNER JOIN store s ON s.STOREID=m.STOREID
    INNER JOIN inventory i ON i.STOREID=s.STOREID
    WHERE
        i.QUANTITY < 20
    GROUP BY m.POSITION_INSTORE,m.CONTACT,m.CEMAIL,i.QUANTITY,s.SADDRESS
    ORDER BY i.QUANTITY ASC;
```

Output:



POSITION_INSTORE	CONTACT	CEMAIL	QUANTITY	SADDRESS
Director	6574676589	Anna@gmail.com	2	653 Sunny Rd
Director	7536549656	Chandrakant@gmail.com	3	44 Cisco Way
Manager	4152324232	Hina@gmail.com	5	233 El Real
Director	5778556545	Priya@gmail.com	5	89 Satalite Rd
Director	8876753423	Rusha@gmail.com	13	754 Inovation Dr
Manager	2343435656	Aditya@gmail.com	14	89 Loggia



```
18    11:39:09    SELECT  m.POSITION_INSTO...  6 row(s) returned
```

- This table will helpful to manager who need to handle the inventory. If `insta_grocery` app manager wants to see which store has the lowest inventory, then he can access this table and easily reach out to the store's manager in order to warn them about current status of their inventory.
- This information is also helpful when app manger wants to contact all stores manager for any heads up.

Conclusion:

From the given problems, one could understand the gravity of brainstorming and creating questions on our own. The applications depend on data and their assemblance.

The minimum and maximum order grouping helps to understand the requirements of the market or a particular locality.

References:

1] https://www.w3schools.com/sql/sql_where.asp
2] https://practice.geeksforgeeks.org/problems/design-database-schema-for-supermarket
3] Database Systems- Design, Implementation and Management 12e, Carlos Cornel | Steven Morris