Final Assignment Report

Below are the steps needed to complete this assignment.

- I created a new database and corresponding table to insert the data into them, based on the instructions in the .sql file.
- I used a SELECT SQL query to extract the required data.

```
SELECT 'invoice_and_item_number',
  'date',
  'store_number',
  'store_name',
  'address',
  'city',
  'zip_code',
  'store_location',
  'county_number',
  'county',
  'category',
  'category_name',
  'vendor_number',
  'vendor_name',
  'item_number',
  'item_description',
  'pack',
  'bottle_volume_ml',
  'state_bottle_cost',
  'state_bottle_retail',
  'bottles_sold',
  'sale_dollars',
  'volume_sold_liters',
  'volume_sold_gallons'
   UNION ALL SELECT invoice_and_item_number,
   date,
   store_number,
   store_name,
   address,
   city,
   zip_code,
   store_location,
   county_number,
   county,
   category,
   category_name,
   vendor_number,
   vendor_name,
   item_number,
   item_description,
   pack,
   bottle_volume_ml,
    state_bottle_cost,
    state_bottle_retail,
    bottles_sold,
    sale_dollars,
    volume_sold_liters,
    volume_sold_gallons
        FROM finance_liquor_sales
        WHERE date >= '2016/01/01' and date < '2020/01/01'
        INTO OUTFILE 'filename.csv'
        FIELDS TERMINATED BY ','
        ENCLOSED BY '"'
        LINES TERMINATED BY '\n';
```

- I installed the following modules in pyCharm
  - numpu
  - pandas
  - matplotlib
- I read the data using the read_csv() method of the Pandas module.

```
filename = 'filename.csv'
csv_table = pd.read_csv(filename)
```

- I calculated the total bottles sold by grouping the data by zip code and item.

```
total_bottles_sold_per_zip_item = csv_table.groupby([ 'zip_code',
'item_description' ])[ 'bottles_sold' ].sum().reset_index()
```

- I created a scatter plot containing the total sales for each zip code and item.

```
plt.subplot(2, 1, 1)
for item in total_bottles_sold_per_zip_item[ 'item_description' ].unique():
    item_rows = total_bottles_sold_per_zip_item[
total_bottles_sold_per_zip_item[ 'item_description' ] == item ]
    plt.scatter(item_rows[ 'zip_code' ], item_rows[ 'bottles_sold' ])
plt.title("Items sold based on zip code")
plt.xlabel("zio code")
plt.ylabel("Total sales")
```

- I calculated the percentage of sold items per store.

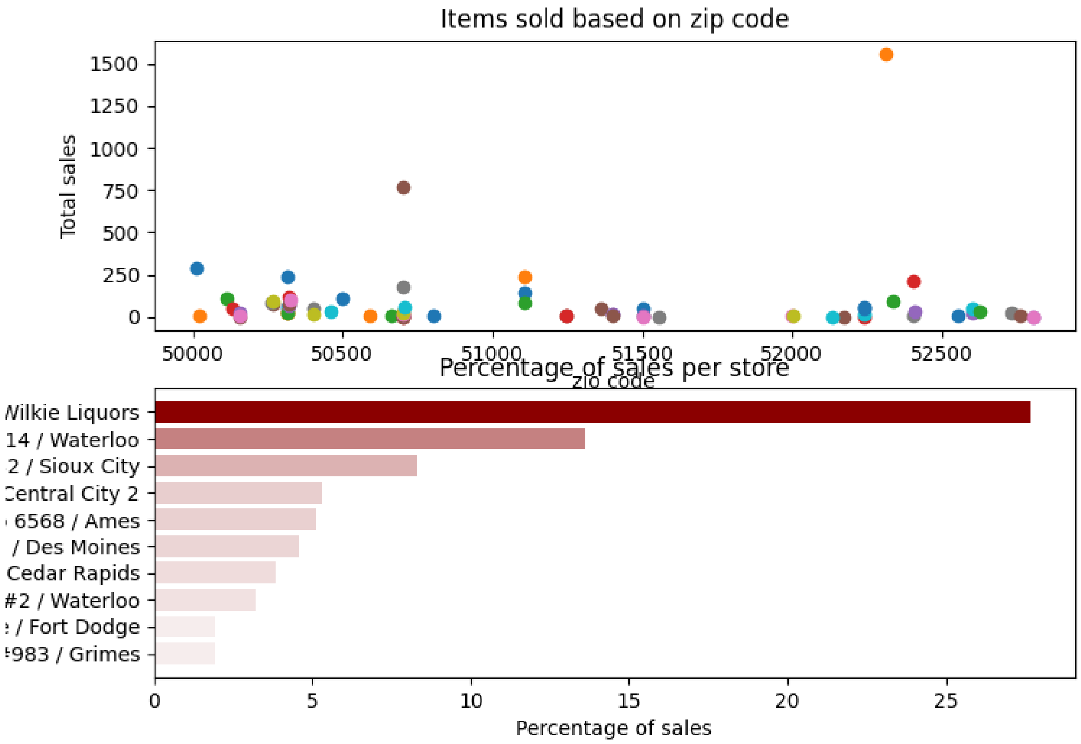
```
total_bottles_sold_per_store = csv_table.groupby([ 'store_name' ])[
'bottles_sold' ].sum()
total_sells = total_bottles_sold_per_store.sum()

sales_percentage_per_store = total_bottles_sold_per_store / total_sells *
100
max_sales_percentage_per_store = sales_percentage_per_store.max()
sales_percentage_per_store.sort_values(ascending=True, inplace=True)
```

- I created a bar plot to visualize the percentage of sold items per store (top 10).

```
sales_percentage_per_store = sales_percentage_per_store.tail(10)
plt.subplot(2, 1, 2)
plt.barh(sales_percentage_per_store.index, sales_percentage_per_store, color=[to_rgba('darkred', alpha=perc / max_sales_percentage_per_store) for perc in sales_percentage_per_store])
plt.title("Percentage of sales per store")
plt.xlabel("Percentage of sales")
plt.ylabel("Store")
plt.show()
```

Please find below the requested plots


Items sold based on zip code
Total sales
zip code
Percentage of sales per store
Wilkie Liquors
14 / Waterloo
2 / Sioux City
Central City 2
6568 / Ames
/ Des Moines
Cedar Rapids
#2 / Waterloo
/ Fort Dodge
983 / Grimes
Percentage of sales