BUAN 6320.502
Project 1
IOWA State Liquor Sales data

Submitted by
Raghothama Rao Pranesha – rxr220019

Afsaruddin Mohammed - axm210415

Koti Reddy Gangasani - kxg220019

Sri Mahalakshmi Pendyala – sxp220096

Vikrant Sagar Remoddula - vxr220005

Dataset Requirements

Link to dataset: https://www.kaggle.com/datasets/alexsueppel/iowa-liquor-sales-eda

- Dataset is of size 5.41 GB and structured format with over 20 million records and 24 columns of varied datatypes (Integer, Float, Varchar and Date).
- Dataset does not contain a significant amount of missing data (only 0.05%).
- This file contains liquor sale data from January 2012 – September 2021 from the IOWA State.

Business Understanding

This dataset contains every wholesale purchase of liquor in the State of Iowa by retailers for sale to individuals since January 1, 2012. The State of Iowa controls the wholesale distribution of liquor intended for retail sale, which means this dataset offers a complete view of retail liquor sales in the entire state. This dataset contains information on the name, kind, price, quantity, and location of sale of sales of individual containers or packages of containers of alcoholic beverages.

- The data has been gathered to explore the dataset to collect as many insights as possible that can be used to improve the performance in the following year. For example, collecting information regarding which brand is the most popular, that allows the business to strategize, or plan based on that insight for the following year.

- The following inferences can be made using the data:
  1. Most popular brands and types of alcohol
  2. Price variance between same-city stores and different-city stores
  3. Retail Revenue by County
  4. Revenue per store and city
  5. Profit by Type of Liquor
  6. Top Categories and Top selling stores.
  7. How much revenue does Iowa make from the business each month? Which product drives the most sales?
  8. Are there any variations among regions in terms of the most popular product?
  9. How many bottles will Iowa sell in the next three months?
- By manipulating the data using various data analysis techniques and tools, you can begin to find trends, correlations, outliers, and variations that tell a story. The purpose of this project is to provide actionable business insights to both distilleries and liquor stores through retail sales analytics.
- Analysing the data, recommendations can be made around pricing, procurement, stocking, and production strategies, and can be broken down on a month-by-month basis in order to maximize sales and profits by meeting variable consumer demand, identifying gaps and saturation points in the market, minimizing overstocking costs, and identifying optimal price points for different liquor categories.
- Total liquor sales tend to go in descending order by day of the week, with Sunday having the lowest number of sales. This could be due to city/county restrictions placed on

alcohol sales for weekends. We can optimize the business by analysing the data and calculating the moving average to identify the trends and seasonality cut-offs.

Data Understanding

The details of the data we are using for the analysis is presented below:

Column	Description	Data type
Invoice/Item locator	alphanumeric value unique for every item purchased. Ex: S30190300003	VARCHAR
Date	Date on which an item is purchased Ex: 01/14/2016	DATE
Store Number	Contains the designated store numbers Ex: 3731	INTEGER
Store Name	contains the name of the stores of retail Ex: Wal-Mart 1241/Davenport	VARCHAR
Address	addresses of each of the stores Ex:5811 Elmore Ave	VARCHAR
City	cities to which a store belong Ex: Davenport	VARCHAR
Zip Code	zip codes of the store location Ex: 52807	INTEGER
Store Location	Co ordinates of the store Ex: POINT (-90.525525 41.580212)	VARCHAR
County Number	County where that store is located. Ex: 82	INTEGER
County	Name of the county. Ex: Scott	VARCHAR
Category	Refer to the category to which an item belongs. Ex: 1011100	INTEGER
Category Name	Refer to the category to which an item belongs. Ex: BLENDED WHISKIES	VARCHAR
Vendor Number	Number of the vendors issuing the supply to the stores. Ex: 297	INTEGER

Vendor Name	Name of the vendors issuing the supply to the stores. Ex: Laird And Company	VARCHAR
Item Number	Number to the items in specific as in what the item. Ex: 82	INTEGER
Item Description	Name of the item. Ex: Five Star	VARCHAR
Pack	No. of bottles that are packed. Ex: 6	INTEGER
Bottle Volume (ml)	volume of liquor filled in the bottle. Ex: 1750	INTEGER
State Bottle Retail	price at which the bottle sold in the market. Ex: 11.19	FLOAT
Bottles sold	Number of bottles that are being sold. Ex: 6	INTEGER
Sale (Dollars)	The amount which a pack of bottles sold. Ex: 67.14	FLOAT
Volume Solid (liters)	Volume of total bottles sold in a pack. Ex: 10.5	FLOAT
Volume Solid (Gallons)	Volume of total bottles sold in a pack. Ex: 2.77	FLOAT
Sate Bottles Cost	Price at which the state bought the bottle. Ex: 7.46	FLOAT

- Missing values constitute about 0.5% of the overall data values. Their information is presented in the table below. Excluding missing values, we have 20,053,130 values with no duplicate records to analyse.
- Additionally, some of the column names are changed to follow the SQL convention and for ease of use.
- The table below shows the number and percentage of missing values along with the updated column names.

Columns	No. of Missing Values	% of Missing Values	Updated Column names
Invoice/Item locator	**0**	**0**	Invoice_id

Date	**0**	**0**	Date
Store Number	**0**	**0**	Store_id
Store Name	**0**	**0**	Store_name
Address	**79992**	**0.358951**	Street_address
City	**79991**	**0.358955**	City
Zip Code	**80036**	**0.359153**	Zip_code
Store Location	**2138686**	**9.597116**	Location
County Number	**156796**	**0.703605**	County_id
County	**156794**	**0.703596**	County_name
Category	**16974**	**0.076169**	Category_id
Category Name	**25040**	**0.112364**	Category_name
Vendor Number	**9**	**0.00004**	Vendor_id
Vendor Name	**7**	**0.000031**	Vendor_name
Item Number	**0**	**0**	Item_Id
Item Description	**0**	**0**	Item_description
Pack	**0**	**0**	Pack
Bottle Volume (ml)	**0**	**0**	**Bottle_volume**
State Bottle Retail	**10**	**0.000045**	Bottle_retail_value
Bottles sold	**0**	**0**	**Bottle_sold**
Sale (Dollars)	**10**	**0.000045**	**Amount**
Volume Solid (liters)	**0**	**0**	***removed***
Volume Solid (Gallons)	**0**	**0**	***removed***
Sate Bottles Cost	**10**	**0.000045**	**State_bottle_cost**

Column	Maximum	Minimum	Mean	Std Dev	Range
Retail in $	39.75	1.7	13.74	7.59	

Cost in $	26.5	1.13	9.14	5.06	
Volume in ml	1750	200	1292.81	504.79	
Revenue(Sales) in $	279557	0	138.85	488.66	

Some of the following information are inferred from the dataset:

- The city with the highest revenue is Western Union. Western Union has the maximum amount of sales with revenue amounting to 2575987.67$ followed by Earling with an incoming revenue of 183144.41$. The difference between the top two performing cities is 2.39 Million $. Leclaire, Denison, and Fonda are the trailing cities. [Query - Select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id = s.Store_id group by st.City order by s.amount desc;]

- It is observed that American Sloe Gins are popular among the masses of Iowa State Thus, it brings in the highest revenue of 4126432.99$, after gins Imported Vodka brings in the second most revenue it is also observed that vodka as an alcoholic beverage performed better than Dry Gins and Whiskies. [Query - Select c.Category_name as Category, sum(s.amount) as Revenue from sales s Ajoin items i on s.ITEM_id = i.Item_id join category c on i.Category_Category_id = c.Category_id group by c.Category_name order by s.amount desc limit 10;]

- Store_id - 3814 on the date of 2015-11-09 had the maximum amount of sales, The difference between the top-performing store and least performing store (Store id - 3420) is 485470$ with a percentage difference of 31.88% .The Median amount of sales was found to be 1048810.5$. [Query - Select s.date, sum(s.amount) s.Store_id from sales s inner join store st on st.Store_id = s.Store_ID group by Date order by amount desc;]

- The maximum number of sales happened on 2nd of October 2018, The Maximum amount spent, in a single transaction is 279557$, The transaction occurred in Store_id 2663. The top seven transactions occurred in a single store, bearing the Store_id 2633. Store 2633 brought in a total revenue of 1638963$ on respective 7 dates in a period of 9 years. [Query - select s.date, s.amount, s.Store_id from Sales s inner join store st on st.Store_id=s.Store_id order by amount desc;]

- Fields like Invoice_id, County_id, Category_id, Store_id and Vendor_id are independent and can uniquely identify other records. They are called Primary Keys.

Designing the Database

Entity	Column Name	Data Type
Item	**Item_id**	int
	Item_description	varchar
	Pack	int
	Bottle_retail_value	float
	Bottle_sold	int
	State_bottle_cost	float
	Bottle_volume	int
Store	**Store_id**	int
	Location	varchar
	Store_name	varchar
	Street_address	varchar
	City	varchar
	Zip_code	int

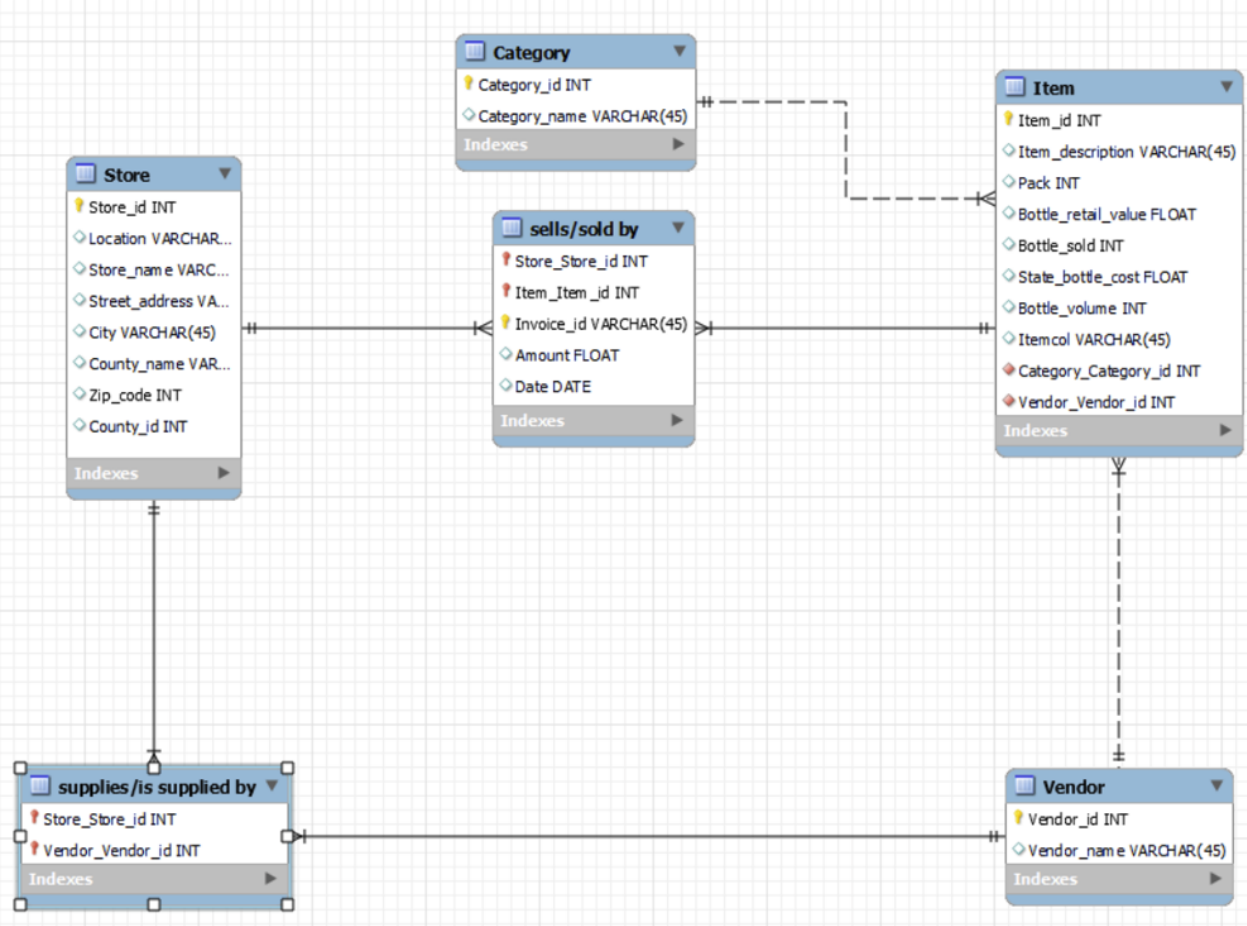
Entity	Column Name	Data Type
Vendor	**Vendor_id**	int
	Vendor_name	varchar
County	**County_id**	int
	County_name	varchar
Category	**Category_id**	int
	Category_name	varchar
Sales	**Invoice_id**	int
	Item_id	varchar
	Store_id	int
	Amount	float
	Date	int
Supply	**Store_id**	int
	Vendor_id	varchar

This data represents the sales of alcohol in the state of Iowa, United States. It can be used to answer many questions like how much alcohol is sold and consumed in the Iowa, what kind, what are the most popular brands and labels, what is the distribution of prices paid in-store, category of the item, vendor and store transactions, and sales of particular item in an particular store and so on.

According to the business requirements, the data has been divided into **seven** separate entities and their respective attributes were assigned to those entities as shown above. The following data represents the Entities and their foreign keys along with the relationships:

Entity 1	Item	Category	County	Store	Supply	Store	Sale
Entity 2	Vendor	Item	Store	Supply	Vendor	Sales	Item
Foreign Keys	Vendor_id	Category_id	County_id	Store_id	Vendor_id	Store_id	Item_id
Relationship	Many to One	One to Many	One to Many	One to Many	Many to One	One to Many	Many to One

ER Diagram before Normalization:



Schema Normalization:

- Functional Dependencies:

{Vendor_id}->{Vendor_name}
{category_id}->{category_name}
{Store_id}-> {Location, Store_name, Street_address, city, Zip_code, County_id}
{item_id}-> {item_description, pack, bottle_retail_value, bottle_sold, state_bottle_cost, Bottle_volume}
{invoice_id}-> {amount,date}

{county _id}->{county_name}
{Store_id, Vendor_id}->{Vendor_id, Vendor_name, Location, Store_name, Street_address, City, Zip_code, County_id}

For example,
For the item entity A={item, description, pack, bottle_retail_value, Bottles_sold, State_bottle_cost, Bottle_volume} and the FD
{Item_id}-> {Item_description, Pack, Bottle_retail_value, Bottle_sold, State_bottle_cost, Bottle_volume} and no other attribute can be added to the respective FD.

Similarly {Vendor_id},{Category_id},{Store_id},{item_id},{invoice_id},{county_id} for their respective entities.

- The tables are in 1st Normal form as all the columns are atomic, and all the values stored in a particular column are of the single valued.
- The table also satisfies the 2nd Normal Form, as there is no Partial Dependency.
- In Store table present in the above ERD, store_id determines county_id, and county_id determines county_name. Therefore, store_id determines county_name via county_id. This implies that the table possesses a transitive functional dependency, and it does not fulfil the third normal form criteria.
- Now to change the table to the third normal form, you need to decompose the table as shown below:

Store_id	Location	Store_name	Street_address	City	Zip_ code	County_id	County_name

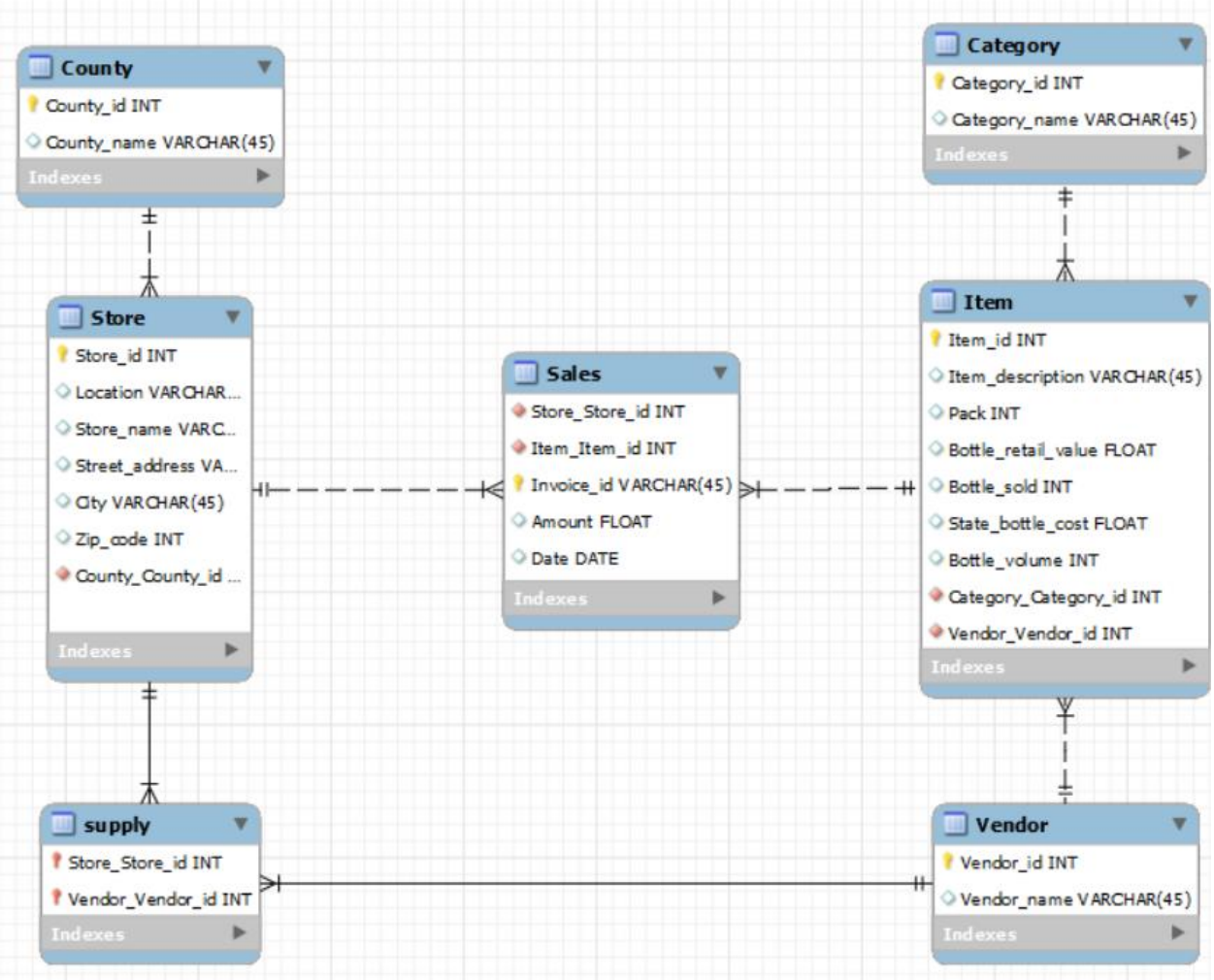
Store_id	Location	Store_name	Street_address	City	Zip_code	County_id

County_id	County_name

As you can see in both the tables, all the non-key attributes are now fully functional, dependent only on the primary key satisfying the third normal form. For a table to satisfy the Boyce-Codd Normal Form, it should satisfy the following two conditions:

- It should be in the Third Normal Form.
- For any dependency A → B, A should be a super key.

Our schema holds the above condition. So, our schema is in **BCNF**. The final ERD after normalization and entity name updates for Sales and Supply is shown below:



Data Import:
We have imported the data using Table Data Import Wizard for small tables and the query below for larger tables:

LOAD DATA INFILE '--filepath'
ignore INTO TABLE tablename
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS; (to ignore column names)

Error while importing the database:

1. Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement
   Reason : The variable secure_file_priv is used to limit the effect of data import and export operations and these operations are allowed only to users who have the FILE privilege.

Fix : We may use SHOW VARIABLES LIKE "secure_file_priv"; to see the directory that has been configured. We can fix this by moving the file to the directory specified by secure-file-priv.

2. Error Code: 2013. Lost connection while loading data to MySQL server during query

Reason : This error appears when the connection between your MySQL client and database server times out. Essentially, it took too long for the query to return data so the connection gets dropped

Fix : We can increase your MySQL client's timeout values by editing the SQL Editor preferences in MySQL Workbench:

1. In the application menu, select Edit > Preferences > SQL Editor.
2. Look for the MySQL Session section and increase the DBMS connection read time out value.
3. Save the settings, quite MySQL Workbench and reopen the connection.

We have changed it to **6000 seconds.**

Data Cleaning and Database Testing

Data Cleaning is done in Jupyter Notebook using Python as shown in the following steps:

1. Importing the necessary libraries: os, pandas, numpy
2. Reading the csv file(dataset) and creating a dataframe object

```
df=pd.read_csv("Iowa_Liquor_Sales.csv")
```

```
C:\Users\14696\AppData\Local\Temp\ipykernel_27628\3175427799.py:1: DtypeWarning: Columns (6,14) have mixed types. Specify dt
ype option on import or set low_memory=False.
  df=pd.read_csv("Iowa_Liquor_Sales.csv")
```

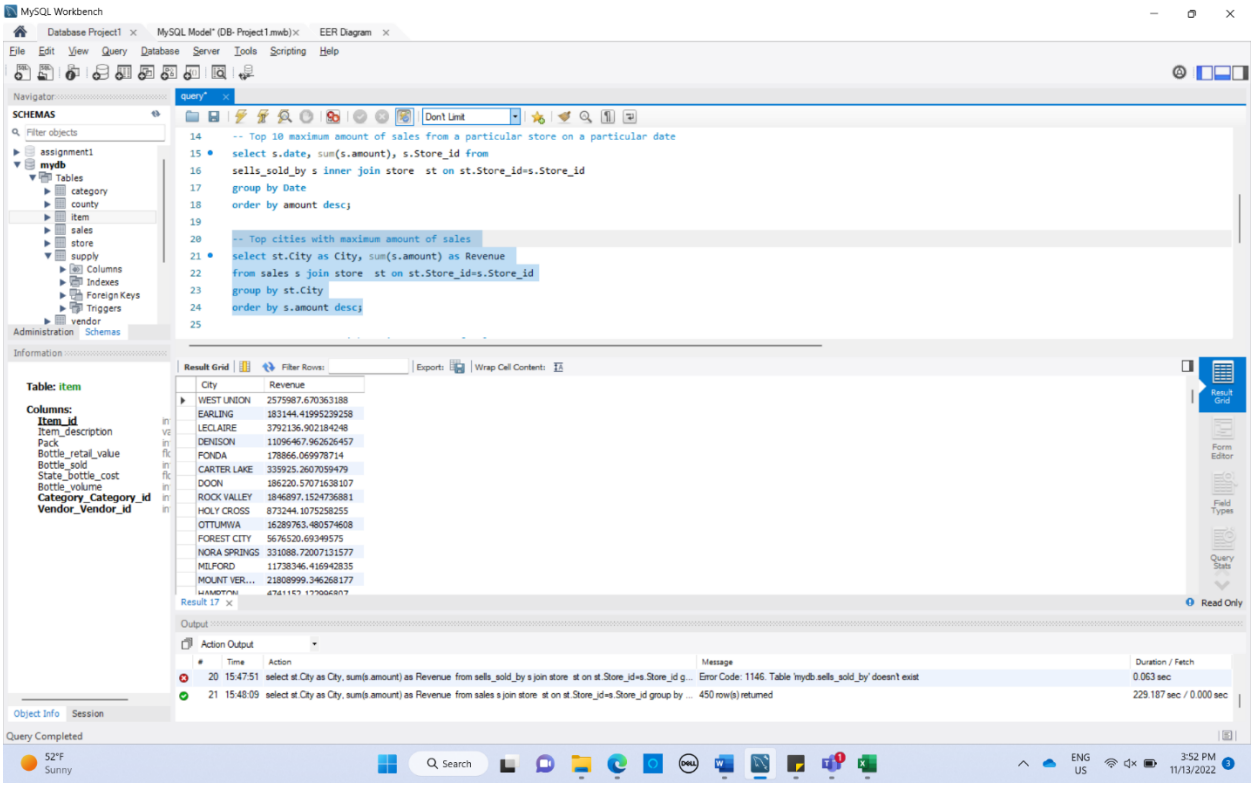
3. Counting the number of NULL values in each column using pandas function on the dataframe 'df' that we created. *'df.isna().sum()'*
4. Dropping the records(rows) which has null values using '*df.dropna(inplace=True)*' since it constitutes only 0.5% of the data resulting in 2005310 records to analyse.
5. Checking the count of null values in each column after dropping null values will result in 0 null values
6. Changing the data type of Category(number),Vendor number, County Number and Item Number columns from float to int using:
   df['Category']=df['Category'].astype('int64',copy=False)
   df['Vendor Number']=df['Vendor Number'].astype('int64',copy=False)
   df['County Number']=df['County Number'].astype('int64',copy=False)
   df['Item Number']=df['County Number'].astype('int64',copy=False)
7. Changing the date format of Sale table(from mm-dd-yyyy to yyyy-mm-dd)
   df['Date']=pd.to_datetime(df['Date'])
8. Splitting the data set into the seven required tables and removing duplicates from it based on the primary key. Finally, exporting it as a csv file. Ex:
   vendor=df.filter(['Vendor Number','Vendor Name'],axis=1)

vendor.drop_duplicates(subset=['Vendor Number'],inplace=True)
vendor.shape this will result in ->(381, 2)
vendor.to_csv("vendor.csv",index=False)

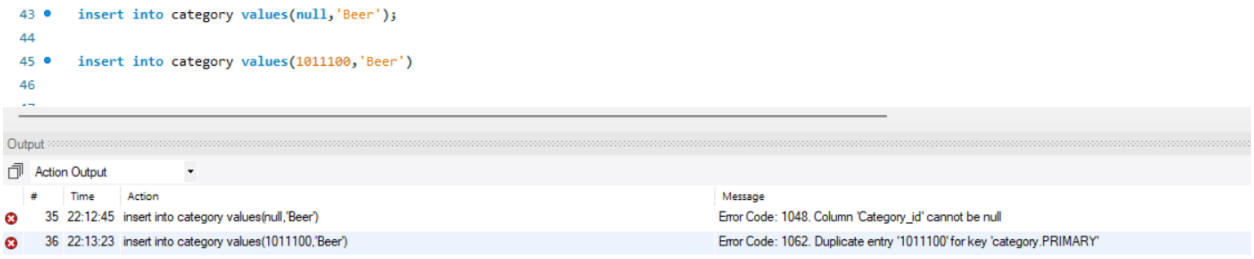
- For each table in your database, check all the columns and the values they contain – Done in above steps.
- For numeric columns, we have checked for the statistics in above steps

A few statistics above and below are insights inferred from the data:

Top 10 alcohol selling cities in Iowa state along with their revenues



Primary Key constraint: Primary key must be unique and not null.



Foreign key constraint:

```
47 •  insert into supply values (100,100);
48
```

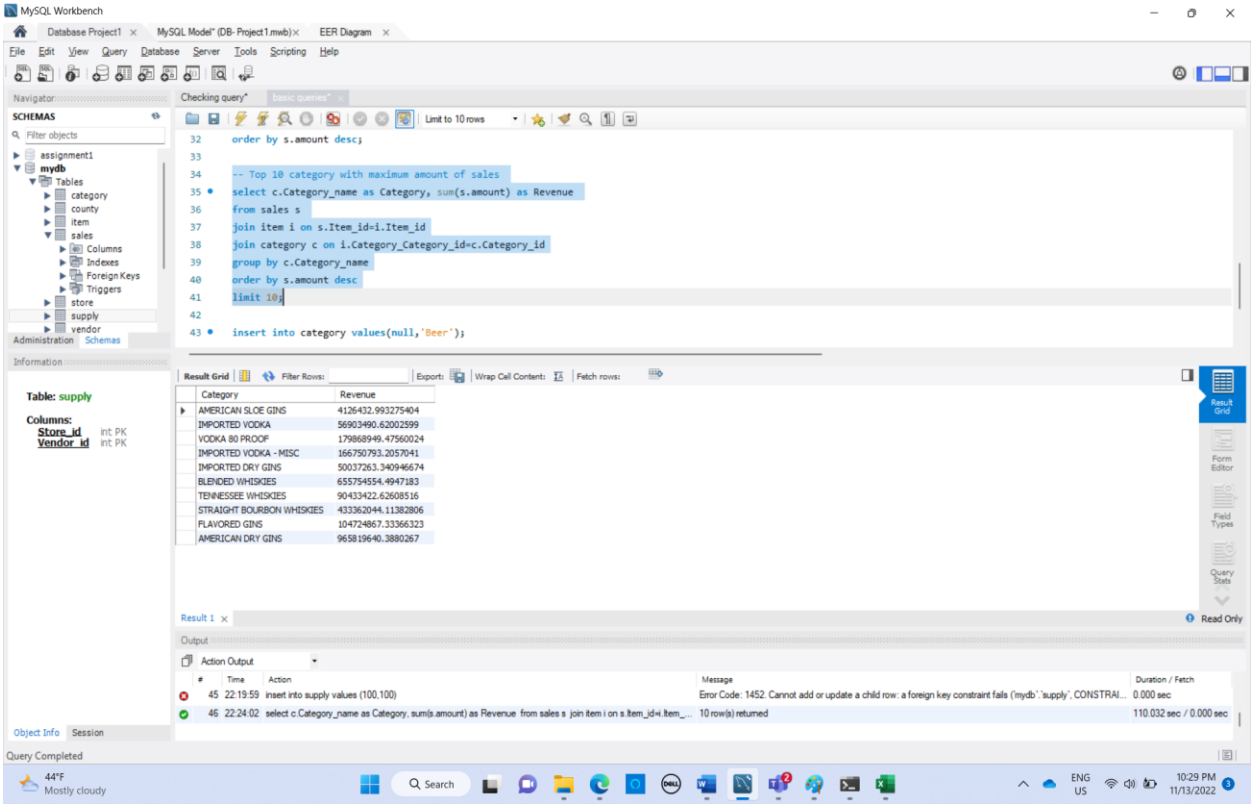
Output

Action Output

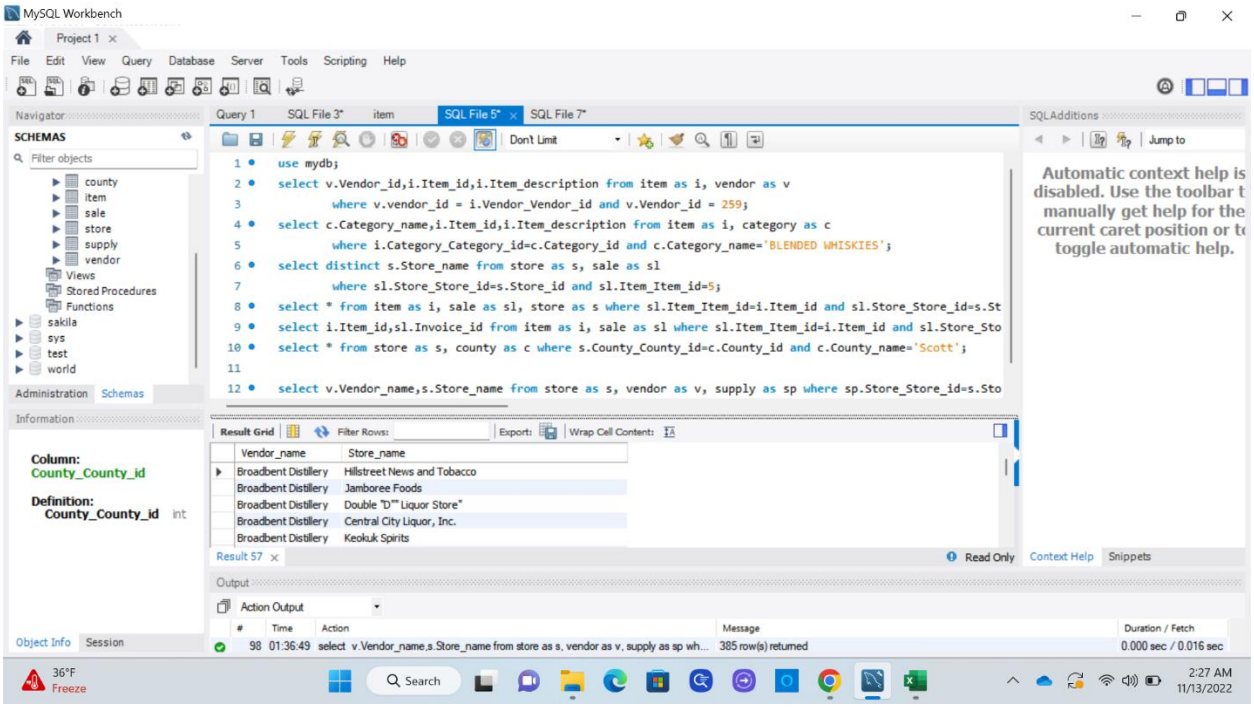
#	Time	Action	Message	Duration / Fe
44	22:19:55	insert into supply values (100,100)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`supply`, CONSTRAI...	0.000 sec
45	22:19:59	insert into supply values (100,100)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`supply`, CONSTRAI...	0.000 sec

Foreign key relationships involve a parent table that holds the central data values, and a child table with identical values pointing back to its parent. The FOREIGN KEY clause is specified in the child table.
It will reject any INSERT or UPDATE operation that attempts to create a foreign key value in a child table if there is no a matching candidate key value in the parent table.

Testing joins:



To get the Category_name , we are joining three tables:
Table sales joined to Table item on Item_id
Table item joined to Table category on Category_id to finally get the Category_name.



Testing all the joints between the connected entities