kaggle
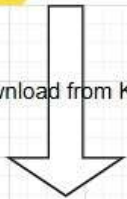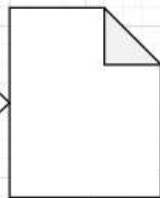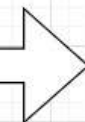
Download from Kaggle

Python

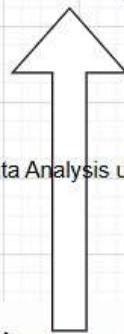Data Cleaning

Python Pandas

Data Loading

Microsoft
SQL Server

Data Analysis using SQL

```
[1]: !pip install kaggle
     import kaggle
     !kaggle datasets download ankitbansal06/retail-orders -f orders.csv
```

Requirement already satisfied: kaggle in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (1.6.17)
Requirement already satisfied: six>=1.10 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi>=2023.7.22 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (2024.8.30)
Requirement already satisfied: python-dateutil in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (2.9.0.post0)
Requirement already satisfied: requests in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (2.32.3)
Requirement already satisfied: tqdm in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (4.67.1)
Requirement already satisfied: python-slugify in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (2.2.3)
Requirement already satisfied: bleach in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from kaggle) (6.2.0)
Requirement already satisfied: webencodings in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from python-slugify->ka
ggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from requests->kag
gle) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from requests->kaggle) (3.10)
Requirement already satisfied: colorama in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from tqdm->kaggle) (0.4.6)
Dataset URL: https://www.kaggle.com/datasets/ankitbansal06/retail-orders
License(s): CC0-1.0
orders.csv.zip: Skipping, found more recently modified local copy (use --force to force download)

```
[2]: !pip install pandas
     import pandas as pd
```

Requirement already satisfied: pandas in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.1.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.9.0.
post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in c:\users\noors\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.8.2->panda
s) (1.16.0)

```python
#uploading dataset in pandas
df = pd.read_csv(r"C:\Users\noors\OneDrive\Documents\sql+python\orders.csv.csv",na_values=['Not Available', 'unknown'])
```

```python
[4]: df.head(5)
```

[4]:

	Order Id	Order Date	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub Category	Product Id	cost price	List Price	Quantity	Discount Percent
0	1	2023-03-01	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	FUR-BO-10001798	240	260	2	2
1	2	2023-08-15	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	FUR-CH-10000454	600	730	3	3
2	3	2023-01-10	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	OFF-LA-10000240	10	10	2	5
3	4	2022-06-18	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	FUR-TA-10000577	780	960	5	2
4	5	2022-07-13	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	OFF-ST-10000760	20	20	2	5

```python
[5]: df["Ship Mode"].unique()
```

```python
[5]: array(['Second Class', 'Standard Class', nan, 'First Class', 'Same Day'],
      dtype=object)
```

```python
[10]: #df.rename(columns={'Order Id':'order_id', 'Order Date':'order_date'})
      #df.column=df.columns.str.lower()
      #df.columns=df.column.str.replace(" ","_")
      df.head(5)
```

```
[10]:    #df.rename(columns={'Order Id':'order_id', 'Order Date':'order_date'})
         #df.column=df.columns.str.lower()
         #df.columns=df.column.str.replace(" ","_")
         df.head(5)
```

[10]:

	order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	cost_price	list_price	quantit
0	1	2023-03-01	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	FUR-BO-10001798	240	260	
1	2	2023-08-15	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	FUR-CH-10000454	600	730	
2	3	2023-01-10	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	OFF-LA-10000240	10	10	
3	4	2022-06-18	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	FUR-TA-10000577	780	960	
4	5	2022-07-13	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	OFF-ST-10000760	20	20	

```
#derive new columns discount , sale price and profit
#df['discount']=df['list_price']*df['discount_percent']*0.01
#df['sales_price']=df['list_price']-df['discount']
#df['profit']=df['sales_price']-df['cost_price']
#df['profit']
df
```

	order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	cost_price	list_price	c
0	1	2023-03-01	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	FUR-BO-10001798	240	260	
1	2	2023-08-15	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	FUR-CH-10000454	600	730	
2	3	2023-01-10	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	OFF-LA-10000240	10	10	
3	4	2022-06-18	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	FUR-TA-10000577	780	960	
4	5	2022-07-13	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	OFF-ST-10000760	20	20	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9989	9990	2023-02-18	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	FUR-FU-10001889	30	30	
9990	9991	2023-03-17	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	FUR-FU-10000747	70	90	
9991	9992	2022-08-07	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	TEC-PH-10003645	220	260	
9992	9993	2022-11-	Standard	Consumer	United	Costa Mesa	California	92627	West	Office	Paper	OFF-PA-	30	30	

```
[18]:  #convert order date from object data type to datetime
       df.dtypes
       df['order_date']=pd.to_datetime(df['order_date'],format=("%Y-%m-%d"))
       df['order_date']
```

```
[18]:  0        2023-03-01
       1        2023-08-15
       2        2023-01-10
       3        2022-06-18
       4        2022-07-13
                   ...
       9989     2023-02-18
       9990     2023-03-17
       9991     2022-08-07
       9992     2022-11-19
       9993     2022-07-17
       Name: order_date, Length: 9994, dtype: datetime64[ns]
```

```
[20]:  #drop cost price List price and discount percent columns
       #df.drop(columns=['cost_price','List_price','discount_percent'],inplace=True)
       df.head()
```

[20]:

	order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	quantity	discount	sales_pric
0	1	2023-03-01	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	FUR-BO-10001798	2	5.2	254.
1	2	2023-08-15	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	FUR-CH-10000454	3	21.9	708.
2	3	2023-01-10	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	OFF-LA-10000240	2	0.5	9.
3	4	2022-06-18	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	FUR-TA-10000577	5	19.2	940.

```python
[25]: #load the data into sql server using replace option
      #!pip install sqlalchemy
      #!pip install pyodbc
      #import sqlalchemy as sal
      engine = sal.create_engine(r'mssql://NANI2208\SQLEXPRESS/master?driver=ODBC+DRIVER+17+FOR+SQL+SERVER')
      conn=engine.connect()
```

```python
[28]: #load the data into sql server using append option
      df.to_sql('df_orders', con=conn , index=False, if_exists = 'append')
```

```
[28]: 38
```

File   Edit   View   Query   Project   Tools   Window   Help

```sql
select * from df_orders;

--find top 10 highest reveue generating products

select top 10 Product_id,sum(sales_price) as revenue
from df_orders
group by product_id
order by sum(sales_price) desc ;

--find top 5 highest selling products in each region

with cts as (
select product_id,sum(sales) as sales,region,ROW_NUMBER() over(partition by region  order by sum(sales) desc) as row_no
from df_orders
group by product_id,region)
select * from cts
where row_no<=5;


--find month over month growth comparison for 2022 and 2023 sales eg : jan 2022 vs jan 2023

select * from df_orders;


with cte as (
select year(order_date) as order_year,month(order_date) as order_month,
sum(sales_price) as sales
from df_orders
```

Object Explorer

Connect

es
n Databases
ster
Tables
- System Tables
- External Tables
- Graph Tables
- dbo.df_orders
  - Columns
    - order_id (bigint, null)
    - order_date (datetime, null)
    - ship_mode (varchar(max), null)
    - segment (varchar(max), null)
    - country (varchar(max), null)
    - city (varchar(max), null)
    - state (varchar(max), null)
    - postal_code (bigint, null)
    - region (varchar(max), null)
    - category (varchar(max), null)
    - sub_category (varchar(max), null)
    - product_id (varchar(max), null)
    - quantity (bigint, null)
    - discount (float, null)
    - sales_price (float, null)
    - profit (float, null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.EmployeeDemographics
- dbo.EmployeeErrors
- dbo.EmployeeSalary
- dbo.trips_details4
- dbo.WareHouseEmployeeDemographics

Results   Messages

	sub_category	year_2022	year_2023	highest_salary
1	Machines	73723.2	109178.5	35455.3

Query executed successfully.   NANI2208\SQLEXPRESS (16.0 RTM) | NANI2208\noors (57) | master | 00:00:00 | 1 rows

File   Edit   View   Query   Project   Tools   Window   Help

```sql
--find month over month growth comparison for 2022 and 2023 sales eg : jan 2022 vs jan 2023

select * from df_orders;


with cte as (
select year(order_date) as order_year,month(order_date) as order_month,
sum(sales_price) as sales
from df_orders
group by year(order_date),month(order_date)
--order by year(order_date),month(order_date)
    )
select order_month
, sum(case when order_year=2022 then sales else 0 end) as sales_2022
, sum(case when order_year=2023 then sales else 0 end) as sales_2023
from cte
group by order_month
order by order_month

--for each category which month had highest sales
with cte as (
select category,format(order_date,'yyyyMM') as order_year_month, ROW_NUMBER() over(partition by category order by sum(sales)  desc) as row_no
, sum(sales) as sales
from df_orders
group by category,format(order_date,'yyyyMM') )
select category, order_year_month ,sales from
cte
where row_no=1;
```

Results | Messages

	sub_category	year_2022	year_2023	highest_salary
1	Machines	73723.2	109178.5	35455.3

Object Explorer

Connect

- Databases
- ster
- Tables
  - System Tables
  - External Tables
  - Graph Tables
  - dbo.df_orders
    - Columns
      - order_id (bigint, null)
      - order_date (datetime, null)
      - ship_mode (varchar(max), null)
      - segment (varchar(max), null)
      - country (varchar(max), null)
      - city (varchar(max), null)
      - state (varchar(max), null)
      - postal_code (bigint, null)
      - region (varchar(max), null)
      - category (varchar(max), null)
      - sub_category (varchar(max), null)
      - product_id (varchar(max), null)
      - quantity (bigint, null)
      - discount (float, null)
      - sales_price (float, null)
      - profit (float, null)
    - Keys
    - Constraints
    - Triggers
    - Indexes
    - Statistics
  - dbo.EmployeeDemographics
  - dbo.EmployeeErrors
  - dbo.EmployeeSalary
  - dbo.trips_details4

```sql
--which sub category had highest growth by profit in 2023 compare to 2022
select * from df_orders;
with cts as(
select sub_category,sum(sales) as sales,year(order_date) as year from
df_orders
group by sub_category,year(order_date)),
cts2 as(
select sub_category,
sum(case when year=2022 then sales else 0 end) as year_2022,
sum(case when year=2023 then sales else 0 end) as year_2023
from cts
group by sub_category)
select top 1 *,
(year_2023-year_2022) as highest_salary
from cts2
order by (year_2023-year_2022) desc
```

Results

sub_category	year_2022	year_2023	highest_salary
Machines	73723.2	109178.5	35455.3

Query executed successfully.