```python
#import libraries
!pip install kaggle

import kaggle

!kaggle datasets download ankitbansal06/retail-orders -f orders.csv
```

```
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)
```

```python
#extract file from zip file
import zipfile
zip_ref = zipfile.ZipFile('orders.csv.zip')
zip_ref.extractall() # extract file to dir
zip_ref.close() # close file

#read data from the file and handle null values
import pandas as pd
df = pd.read_csv('orders.csv',na_values=['Not Available','unknown'])
df['Ship Mode'].unique()
```

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

```python
#rename columns names ..make them lower case and replace space with
underscore
df.rename(columns={'Order Id':'order_id', 'City':'city'})
df.columns=df.columns.str.lower()
df.columns=df.columns.str.replace(' ','_')
df.head(5)
```

```
   order_id  order_date        ship_mode    segment        country  \
0         1  2023-03-01    Second Class   Consumer  United States
1         2  2023-08-15    Second Class   Consumer  United States
2         3  2023-01-10    Second Class  Corporate  United States
3         4  2022-06-18  Standard Class   Consumer  United States
4         5  2022-07-13  Standard Class   Consumer  United States

              city       state  postal_code region         category  \
0        Henderson    Kentucky        42420  South        Furniture
1        Henderson    Kentucky        42420  South        Furniture
2      Los Angeles  California        90036   West  Office Supplies
3  Fort Lauderdale     Florida        33311  South        Furniture
4  Fort Lauderdale     Florida        33311  South  Office Supplies

  sub_category       product_id  cost_price  list_price  quantity  \
0    Bookcases  FUR-BO-10001798         240         260         2
```

```
1       Chairs  FUR-CH-10000454         600         730         3
2       Labels  OFF-LA-10000240          10          10         2
3       Tables  FUR-TA-10000577         780         960         5
4      Storage  OFF-ST-10000760          20          20         2

   discount_percent
0                 2
1                 3
2                 5
3                 2
4                 5
```

```
#derive new columns discount , sale price and profit
df['discount']=df['list_price']*df['discount_percent']*.01
df['sale_price']= df['list_price']-df['discount']
df['profit']=df['sale_price']-df['cost_price']
df
```

```
      order_id  order_date       ship_mode    segment
country  \
0            1  2023-03-01    Second Class   Consumer  United States

1            2  2023-08-15    Second Class   Consumer  United States

2            3  2023-01-10    Second Class  Corporate  United States

3            4  2022-06-18  Standard Class   Consumer  United States

4            5  2022-07-13  Standard Class   Consumer  United States

...        ...         ...             ...        ...            ...

9989      9990  2023-02-18    Second Class   Consumer  United States

9990      9991  2023-03-17  Standard Class   Consumer  United States

9991      9992  2022-08-07  Standard Class   Consumer  United States

9992      9993  2022-11-19  Standard Class   Consumer  United States

9993      9994  2022-07-17    Second Class   Consumer  United States


                 city       state  postal_code region         category
\
0           Henderson    Kentucky        42420  South        Furniture

1           Henderson    Kentucky        42420  South        Furniture

2         Los Angeles  California        90036   West  Office Supplies
```

```
3     Fort Lauderdale     Florida        33311  South        Furniture

4     Fort Lauderdale     Florida        33311  South  Office Supplies

...               ...         ...          ...    ...              ...

9989            Miami     Florida        33180  South        Furniture

9990       Costa Mesa  California        92627   West        Furniture

9991       Costa Mesa  California        92627   West       Technology

9992       Costa Mesa  California        92627   West  Office Supplies

9993      Westminster  California        92683   West  Office Supplies


     sub_category       product_id  cost_price  list_price
quantity  \
0       Bookcases  FUR-BO-10001798         240         260         2

1          Chairs  FUR-CH-10000454         600         730         3

2          Labels  OFF-LA-10000240          10          10         2

3          Tables  FUR-TA-10000577         780         960         5

4         Storage  OFF-ST-10000760          20          20         2

...           ...              ...         ...         ...       ...

9989  Furnishings  FUR-FU-10001889          30          30         3

9990  Furnishings  FUR-FU-10000747          70          90         2

9991       Phones  TEC-PH-10003645         220         260         2

9992        Paper  OFF-PA-10004041          30          30         4

9993   Appliances  OFF-AP-10002684         210         240         2


      discount_percent  discount  sale_price  profit
0                    2       5.2       254.8    14.8
1                    3      21.9       708.1   108.1
2                    5       0.5         9.5    -0.5
3                    2      19.2       940.8   160.8
4                    5       1.0        19.0    -1.0
...                ...       ...         ...     ...
9989                 4       1.2        28.8    -1.2
9990                 4       3.6        86.4    16.4
```

```
9991                    2       5.2         254.8    34.8
9992                    3       0.9          29.1    -0.9
9993                    3       7.2         232.8    22.8

[9994 rows x 19 columns]

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

#drop cost price list price and discount percent columns
df.drop(columns=['list_price','cost_price','discount_percent'],inplace
=True)

!pip install mysqlclient

#load the data into sql server using replace option
import sqlalchemy as sal
engine =
sal.create_engine("mysql+mysqldb://root:123456789@localhost:3306/order
_data")
conn=engine.connect()

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

9994

df.columns

Index(['order_id', 'order_date', 'ship_mode', 'segment', 'country',
'city',
       'state', 'postal_code', 'region', 'category', 'sub_category',
       'product_id', 'quantity', 'discount', 'sale_price', 'profit'],
      dtype='object')
```

```sql
-- Using the order_data database
USE order_data;

-- find top 10 highest revenue generating products
select product_id, sum(sale_price) as sales
from df_orders
group by product_id
order by sales desc
limit 10;

-- Find top 5 highest selling products in each region
with cte as (
    select region, product_id, sum(sale_price) as sales
    from df_orders
    group by region, product_id
)
select *
from (
    select *, row_number() over (partition by region order by sales desc)
as rn
    from cte
) A
where rn <= 5;

-- Find month-over-month growth comparison for 2022 and 2023 sales (e.g.,
Jan 2022 vs Jan 2023)
with cte as (
    select year(order_date) as order_year, month(order_date) as
order_month,
           sum(sale_price) as sales
    from df_orders
    group 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, find which month had the highest sales
with cte as (
    select category, date_format(order_date, '%Y%m') as order_year_month,
           sum(sale_price) as sales
    from df_orders
    group by category, date_format(order_date, '%Y%m')
)
select * from (
    select *,
           row_number() over (partition by category order by sales desc)
as rn
```

```
    from cte
) a
where rn = 1;

-- Which sub-category had highest growth by profit in 2023 compared to
2022
with cte as (
    select sub_category, year(order_date) as order_year,
           sum(sale_price) as sales
    from df_orders
    group by sub_category, year(order_date)
)
, cte2 as (
    select sub_category,
           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 sub_category
)
select *,
       (sales_2023 - sales_2022) as sales_growth
from cte2
order by sales_growth desc
limit 1;
```