ORDERS ANALYSIS PROJECT

Python + Sql Project

Python Code

```
# download dataset and unzip file

import kaggle

kaggle.api.authenticate()

kaggle.api.dataset_download_files('ankitbansal06/retail-orders', path='D:\csv file\python_sql_file', unzip=True)
```

```
#read data from the file and handle null values

df = pd.read_csv('D:\\csv file\\python_sql_file\\orders.csv',na_values=['Not Available', 'unknown'])
df.head(20)
df['Ship Mode'].unique()
```

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

```
#rename columns names ..make them lower case and replace space with underscore
df.columns=df.columns.str.lower()
df.columns=df.columns.str.replace(' ','_')
```

```
#derive new columns discount

df['discount']=df['list_price']/100*df['discount_percent']
```

```
#derive new columns  sale price
df['sale_price']=df['list_price']-df['discount']
```

```
#derive new columns profit
df['profit']=df['sale_price']-df['cost_price']
```

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

]:
```
df.dtypes
```

]:
```
order_id                    int64
order_date          datetime64[ns]
ship_mode                   object
segment                     object
country                     object
city                        object
state                       object
postal_code                  int64
region                      object
category                    object
sub_category                object
product_id                  object
cost_price                   int64
list_price                   int64
quantity                     int64
discount_percent             int64
discount                   float64
sale_price                 float64
profit                     float64
dtype: object
```

[28]:
```
df
```

[28]:

	order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	cost_price	list_price	qu
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	2022-11-19	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	OFF-PA-10004041	30	30	
9993	9994	2022-07-17	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	OFF-AP-10002684	210	240	

9994 rows × 19 columns

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

```
#connect python with sql server

import sqlalchemy as sal
engine=sal.create_engine('mssql://DESKTOP-DGONMI4/Orders?driver=ODBC+DRIVER+17+FOR+SQL+SERVER')
conn=engine.connect()
```

```
#frist create a database with same name and create table with same name columns
#load the data into sql server using append option

df.to_sql('df_orders', con=conn,index=False,if_exists ='append')
```

```
38
```

SQL Query

```
select * from df_orders
```

	order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	quantity	discount	sale_price	profit
16	16	2022-06-18	Standard Class	Home O...	United States	Fort Worth	Texas	76106	Cen...	Office Supplies	Binders	OFF-BI-10000756	3	0.00	0.00	0.00
17	17	2022-02-04	Standard Class	Consumer	United States	Madison	Wiscon...	53711	Cen...	Office Supplies	Storage	OFF-ST-10004186	6	20.10	649.90	39.90
18	18	2023-08-04	Second Class	Consumer	United States	West Jordan	Utah	84084	West	Office Supplies	Storage	OFF-ST-10000107	2	2.40	57.60	-2.40
19	19	2022-01-23	Second Class	Consumer	United States	San Francisco	California	94109	West	Office Supplies	Art	OFF-AR-10003056	2	0.40	9.60	-0.40
20	20	2022-01-11	Second Class	Consumer	United States	San Francisco	California	94109	West	Technology	Phones	TEC-PH-10001949	3	6.30	203.70	33.70
21	21	2022-10-05	Second Class	Consumer	United States	San Francisco	California	94109	West	Office Supplies	Binders	OFF-BI-10002215	4	0.40	19.60	-0.40
22	22	2023-07-16	Standard Class	Corporate	United States	Fremont	Nebraska	68025	Cen...	Office Supplies	Art	OFF-AR-10000246	7	0.80	19.20	-0.80
23	23	2023-05-06	Standard Class	Corporate	United States	Fremont	Nebraska	68025	Cen...	Office Supplies	Appliances	OFF-AP-10001492	7	1.80	58.20	8.20
24	24	2023-05-21	Second Class	Consumer	United States	Philadelphia	Pennsyl...	19140	East	Furniture	Chairs	FUR-CH-10002774	2	2.80	67.20	7.20
25	25	2023-02-24	Standard Class	Consumer	United States	Orem	Utah	84057	West	Furniture	Tables	FUR-TA-10000577	3	52.00	988.00	88.00
26	26	2022-06-20	Second Class	Consumer	United States	Los Angeles	California	90049	West	Office Supplies	Binders	OFF-BI-10001634	2	0.50	9.50	-0.50
27	27	2022-02-08	Second Class	Consumer	United States	Los Angeles	California	90049	West	Technology	Accessories	TEC-AC-10003027	3	2.70	87.30	7.30

Query executed successfully.

```
--find top 10 highest revenue generating products

select top 10 product_id,sum(profit) as revenue from df_orders
group by product_id
order by revenue desc
```

Results Messages

	product_id	revenue
1	TEC-CO-10004722	5644.00
2	TEC-MA-10002412	3624.40
3	OFF-BI-10003527	3435.30
4	TEC-CO-10001449	2631.20
5	FUR-CH-10002024	2246.20
6	OFF-BI-10001359	2080.20
7	OFF-BI-10000545	1959.00
8	OFF-BI-10001120	1695.70
9	OFF-BI-10004995	1654.80
10	FUR-BO-10004834	1614.10

```sql
--find top 10 highest sales generating products
select top 10 product_id,sum(sale_price) as topsale from df_orders
group by product_id
order by topsale desc
```

Results | Messages

	product_id	topsale
1	TEC-CO-10004722	59514.00
2	OFF-BI-10003527	26525.30
3	TEC-MA-10002412	21734.40
4	FUR-CH-10002024	21096.20
5	OFF-BI-10001359	19090.20
6	OFF-BI-10000545	18249.00
7	TEC-CO-10001449	18151.20
8	TEC-MA-10001127	17906.40
9	OFF-BI-10004995	17354.80
10	OFF-SU-10000151	16325.80

```sql
--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 *
,rank() over(partition by region order by sales desc) as rn from cte) AS A
where rn<=5
```

Results | Messages

	region	product_id	sales	m
1	Central	TEC-CO-10004722	16975.00	1
2	Central	TEC-MA-10000822	13770.00	2
3	Central	OFF-BI-10001120	11056.50	3
4	Central	OFF-BI-10000545	10132.70	4
5	Central	OFF-BI-10004995	8416.10	5
6	East	TEC-CO-10004722	29099.00	1
7	East	TEC-MA-10001047	13767.00	2
8	East	FUR-BO-10004834	11274.10	3
9	East	OFF-BI-10001359	8463.60	4
10	East	TEC-CO-10001449	8316.00	5
11	South	TEC-MA-10002412	21734.40	1
12	South	TEC-MA-10001127	11116.40	2
13	South	OFF-BI-10001359	8053.20	3
14	South	TEC-MA-10004125	7840.00	4
15	South	OFF-BI-10003527	7391.40	5
16	West	TEC-CO-10004722	13440.00	1
17	West	OFF-SU-10000151	12592.30	2
18	West	FUR-CH-10001215	9604.00	3
19	West	OFF-BI-10003527	7804.80	4
20	West	TEC-AC-10003832	7722.70	5

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

with cte as(
select year(order_date) as years, month(order_date) as months,sum(sale_price) as sales
from df_orders
group by year(order_date),month(order_date))
select months
,sum(case when years=2022 then sales else 0 end) as sale_2022
,sum(case when years=2023 then sales else 0 end) as sale_2023
from cte
group by months
order by months
```

Results | Messages

	months	sale_2022	sale_2023
1	1	94712.50	88632.60
2	2	90091.00	128124.20
3	3	80106.00	82512.30
4	4	95451.60	111568.60
5	5	79448.30	86447.90
6	6	94170.50	68976.50
7	7	78652.20	90563.80
8	8	104808.00	87733.60
9	9	79142.20	76658.60
10	10	118912.70	121061.50
11	11	84225.30	75432.80
12	12	95869.90	102556.10

```sql
--for each category which month had highest sales

with cte as (
select category,format(order_date,'yyyy-MM') as order_year_month,sum(sale_price) as sales
from df_orders
group by category,format(order_date,'yyyy-MM'))
select * from (
select *
,rank() over(partition by category order by sales desc) as ranked
from cte) a
where ranked=1;
```

100 %

Results | Messages

	category	order_year_month	sales	ranked
1	Furniture	2022-10	42888.90	1
2	Office Supplies	2023-02	44118.50	1
3	Technology	2023-10	53000.10	1

```sql
--which sub category had highest growth by profit in 2023 compare to 2022

with cte as(
select sub_category,year(order_date) as years, sum(sale_price) as sales
from df_orders
group by sub_category,year(order_date))
, cte2 as(
select sub_category
,sum(case when years=2022 then sales else 0 end) as sale_2022
,sum(case when years=2023 then sales else 0 end) as sale_2023
from cte
group by sub_category)
select *,(sale_2023-sale_2022)*100/sale_2022 as year_prof_psnt from cte2
order by (sale_2023-sale_2022)*100/sale_2022 desc
```

Results | Messages

	sub_category	sale_2022	sale_2023	year_prof_psnt
1	Supplies	16140.70	28917.40	79.158276
2	Machines	73723.20	109178.50	48.092459
3	Binders	87675.50	108363.10	23.595645
4	Storage	102907.40	113000.60	9.808041
5	Chairs	151395.30	165429.80	9.270102
6	Accessories	77627.20	83977.40	8.180380
7	Bookcases	53469.50	57346.60	7.251049
8	Fasteners	1430.10	1508.80	5.503111
9	Phones	157334.70	160673.60	2.122163
10	Paper	38898.90	36932.40	-5.055412
11	Art	13644.10	12615.70	-7.537323
12	Labels	6329.60	5665.30	-10.495133
13	Furnishings	47816.20	40522.10	-15.254453
14	Envelopes	8767.20	7150.00	-18.446026
15	Tables	111305.90	88055.30	-20.888919
16	Copiers	82090.30	62268.60	-24.146214
17	Appliances	65034.40	38663.30	-40.549463