SQL-Driven Analysis of Retail Stores

```sql
create table stores
(
store_id varchar(50) primary key,
store_category varchar(100)
);

insert into stores(store_id,store_category)
values('a','Food'),
('b','Accessories'),
('c','Cloth'),
('d','Electronics'),
('e','Furniture'),
('f','Groceries'),
('g','Jewelry'),
('h','Mobile'),
('i','Watch'),
('j','Minimarket');

select * from stores;
```

Results | Messages

	store_id	store_category
1	a	Food
2	b	Accessories
3	c	Cloth
4	d	Electronics
5	e	Furniture
6	f	Groceries
7	g	Jewelry
8	h	Mobile
9	i	Watch
10	j	Minimarket

```sql
create table items
(
store_id varchar(50),
item_id varchar(50) primary key,
item_category varchar(100),
item_name varchar(100)
FOREIGN KEY (store_id) REFERENCES stores(store_id),
);
insert into items(store_id,item_id,item_category,item_name)
values ('c','c1','pants','denim pants'),
('c','c2','tops','blouse'),
('e','e1','table','coffee table'),
('e','e2','chair','lounge chair'),
('e','e3','chair','armchair'),
('g','g7','jewelry','bracelet'),
('d','d4','TV','LED'),
('f','f6','rice','Basamati rice'),
('i','i8','watch','Maxima watch'),
('d','d5','earphone','airpods');

select * from items;
```

Prepared By- Vishal Yadav

Results | Messages

	store_id	item_id	item_category	item_name
1	c	c1	pants	denim pants
2	c	c2	tops	blouse
3	d	d4	TV	LED
4	d	d5	earphone	airpods
5	e	e1	table	coffee table
6	e	e2	chair	lounge chair
7	e	e3	chair	armchair
8	f	f6	rice	Basamati rice
9	g	g7	jewelry	bracelet
10	i	i8	watch	Maxima watch

```
create table transactions
(
buyer_id int NOT NULL,
purchase_time datetimeoffset,
refund_item datetimeoffset,
store_id varchar(50),
item_id varchar(50),
gross_transaction_value_in_Rupee numeric,
FOREIGN KEY (item_id) REFERENCES items(item_id),
FOREIGN KEY (store_id) REFERENCES stores(store_id)
);
insert into transactions(buyer_id,purchase_time,store_id,item_id,gross_transaction_value_in_Rupee)
values(6,'2021-09-25 18:18:06.644','c','c1',800);
insert into
transactions(buyer_id,purchase_time,refund_item,store_id,item_id,gross_transaction_value_in_Rupee)
values(11,'2021-12-11 20:10:14.324','2021-12-15 22:20:06.504','c','c2',575);
insert into
transactions(buyer_id,purchase_time,refund_item,store_id,item_id,gross_transaction_value_in_Rupee)
values(6,'2022-03-02 23:59:44.551','2022-03-03 21:23:06.231','e','e1',2700);
insert into transactions(buyer_id,purchase_time,store_id,item_id,gross_transaction_value_in_Rupee)
values(3,'2022-03-28 20:18:06.444','e','e2',3982);
insert into transactions(buyer_id,purchase_time,store_id,item_id,gross_transaction_value_in_Rupee)
values(2,'2022-03-12 23:20:06.331','e','e3',4530);
insert into transactions(buyer_id,purchase_time,store_id,item_id,gross_transaction_value_in_Rupee)
values(5,'2022-04-12 22:10:22.234','g','g7',47562);
insert into
transactions(buyer_id,purchase_time,refund_item,store_id,item_id,gross_transaction_value_in_Rupee)
values(7,'2021-09-22 12:08:35.532','2021-09-26 01:55:02.124','d','d4',24850);

insert into
transactions(buyer_id,purchase_time,refund_item,store_id,item_id,gross_transaction_value_in_Rupee)
values(7,'2022-04-04 21:59:46.566','2022-04-06 22:21:06.231','f','f6',386);
insert into
transactions(buyer_id,purchase_time,refund_item,store_id,item_id,gross_transaction_value_in_Rupee)
values(6,'2021-09-02 22:59:45.566','2021-09-03 22:21:06.345','i','i8',3337);
insert into transactions(buyer_id,purchase_time,store_id,item_id,gross_transaction_value_in_Rupee)
values(15,'2021-04-15 20:11:23.216','d','d5',2430);

select * from transactions;
```

Results | Messages

	buyer_id	purchase_time	refund_item	store_id	item_id	gross_transaction_value_in_Rupee
1	6	2021-09-25 18:18:06.6440000 +00:00	NULL	c	c1	800
2	11	2021-12-11 20:10:14.3240000 +00:00	2021-12-15 22:20:06.5040000 +00:00	c	c2	575
3	6	2022-03-02 23:59:44.5510000 +00:00	2022-03-03 21:23:06.2310000 +00:00	e	e1	2700
4	3	2022-03-28 20:18:06.4440000 +00:00	NULL	e	e2	3982
5	2	2022-03-12 23:20:06.3310000 +00:00	NULL	e	e3	4530
6	5	2022-04-12 22:10:22.2340000 +00:00	NULL	g	g7	47562
7	7	2021-09-22 12:08:35.5320000 +00:00	2021-09-26 01:55:02.1240000 +00:00	d	d4	24850
8	7	2022-04-04 21:59:46.5660000 +00:00	2022-04-06 22:21:06.2310000 +00:00	f	f6	386
9	6	2021-09-02 22:59:45.5660000 +00:00	2021-09-03 22:21:06.3450000 +00:00	i	i8	3337
10	15	2021-04-15 20:11:23.2160000 +00:00	NULL	d	d5	2430

Q 1) what is the count of purchases per month excluding refunded purchases order the months in descending.

Query:-

```
select month(purchase_time) as Purchase_Month,
       count(purchase_time) as Count_of_Purchase_per_month
from transactions
where refund_item is NULL
group by month(purchase_time)
order by month(purchase_time) DESC;
```

Output:-

Results | Messages

	Purchase_Month	Count_of_Purchase_per_month
1	9	1
2	4	2
3	3	2

Q 2) how many stores receive at least three orders in March 2022.

Query:-

```
select count(store_id) as Number_of_stores
from transactions
where month(purchase_time) = '03'
      and year(purchase_time) = '2022'
group by store_id
having count(store_id) >= 3;
```

Output:-

Results | Messages

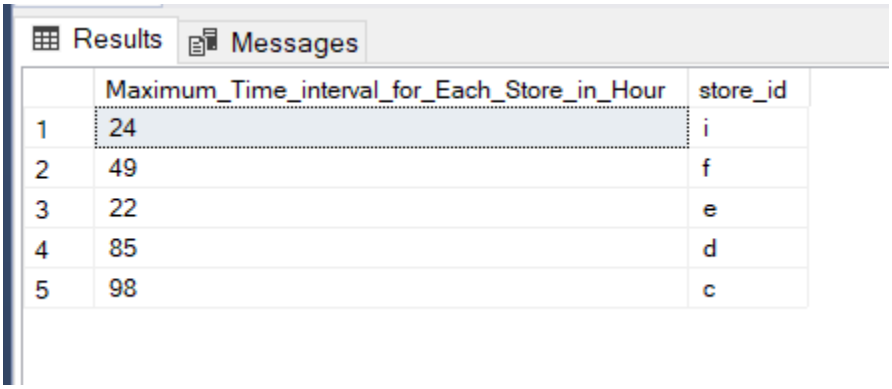
	Number_of_stores
1	3

Q 3) for each store, what is the maximum time interval (in Hour) from purchase to refund time, arrange the store in descending order.

Query:-

```
select MAX(DATEDIFF(HOUR, purchase_time, refund_item)) as Maximum_Time_interval_for_Each_Store_in_Hour,
       store_id
from transactions
where refund_item is not null
group by store_id
order by store_id DESC;
```

Output:-



Results | Messages

	Maximum_Time_interval_for_Each_Store_in_Hour	store_id
1	24	i
2	49	f
3	22	e
4	85	d
5	98	c

Q 4) Create a flag indicating weather refund is processed, too late or not requested. The condition for a refund to be processed is that it has to happen within 72 hours of purchase time.

Query:-

```
select buyer_id,
       purchase_time,
       refund_item,
       DATEDIFF(HOUR, purchase_time, refund_item) as Date_Difference,
       (CASE
            WHEN refund_item is NULL THEN
                'Not requested'
            WHEN DATEDIFF(HOUR, purchase_time, refund_item) > 72 THEN
                'Too late'
            ELSE
                'Processed'
        END
       ) as FLAG
from transactions;
```

Output:-

Results | Messages

	buyer_id	purchase_time	refund_item	Date_Difference	FLAG
1	6	2021-09-25 18:18:06.6440000 +00:00	NULL	NULL	Not requested
2	11	2021-12-11 20:10:14.3240000 +00:00	2021-12-15 22:20:06.5040000 +00:00	98	Too late
3	6	2022-03-02 23:59:44.5510000 +00:00	2022-03-03 21:23:06.2310000 +00:00	22	Processed
4	3	2022-03-28 20:18:06.4440000 +00:00	NULL	NULL	Not requested
5	2	2022-03-12 23:20:06.3310000 +00:00	NULL	NULL	Not requested
6	5	2022-04-12 22:10:22.2340000 +00:00	NULL	NULL	Not requested
7	7	2021-09-22 12:08:35.5320000 +00:00	2021-09-26 01:55:02.1240000 +00:00	85	Too late
8	7	2022-04-04 21:59:46.5660000 +00:00	2022-04-06 22:21:06.2310000 +00:00	49	Processed
9	6	2021-09-02 22:59:45.5660000 +00:00	2021-09-03 22:21:06.3450000 +00:00	24	Processed
10	15	2021-04-15 20:11:23.2160000 +00:00	NULL	NULL	Not requested

Q 5) create a rank by buyer Id in the transaction table & filter for only the third purchase per buyer.

Query:-

```
select buyer_id,
       purchase_time
from
(
    select buyer_id,
           purchase_time,
           rank() OVER (PARTITION BY buyer_id ORDER BY purchase_time ASC) AS rownumber
    from transactions
) t
where rownumber = 3;
```

Output:-

Results | Messages

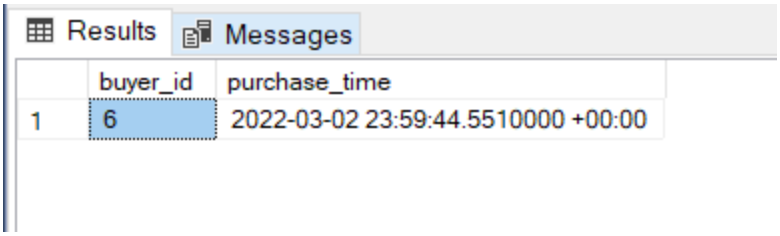
	buyer_id	purchase_time
1	6	2022-03-02 23:59:44.5510000 +00:00

Q 6) Find third transaction time per buyer.

Query:-

```
select buyer_id,
       purchase_time
from
(
    select buyer_id,
           purchase_time,
           row_number() OVER (PARTITION BY buyer_id ORDER BY purchase_time ASC) AS rn
    from transactions
) sub
where rn = 3;
```

Output:-



Results Messages

	buyer_id	purchase_time
1	6	2022-03-02 23:59:44.5510000 +00:00