So this is my case investigation ☺
But I've done my own data structure and I've answered to below questions with my way.
First I have read carefully all questions.
I marked what I should pay attention to, when I will be writing queries in SQL language:

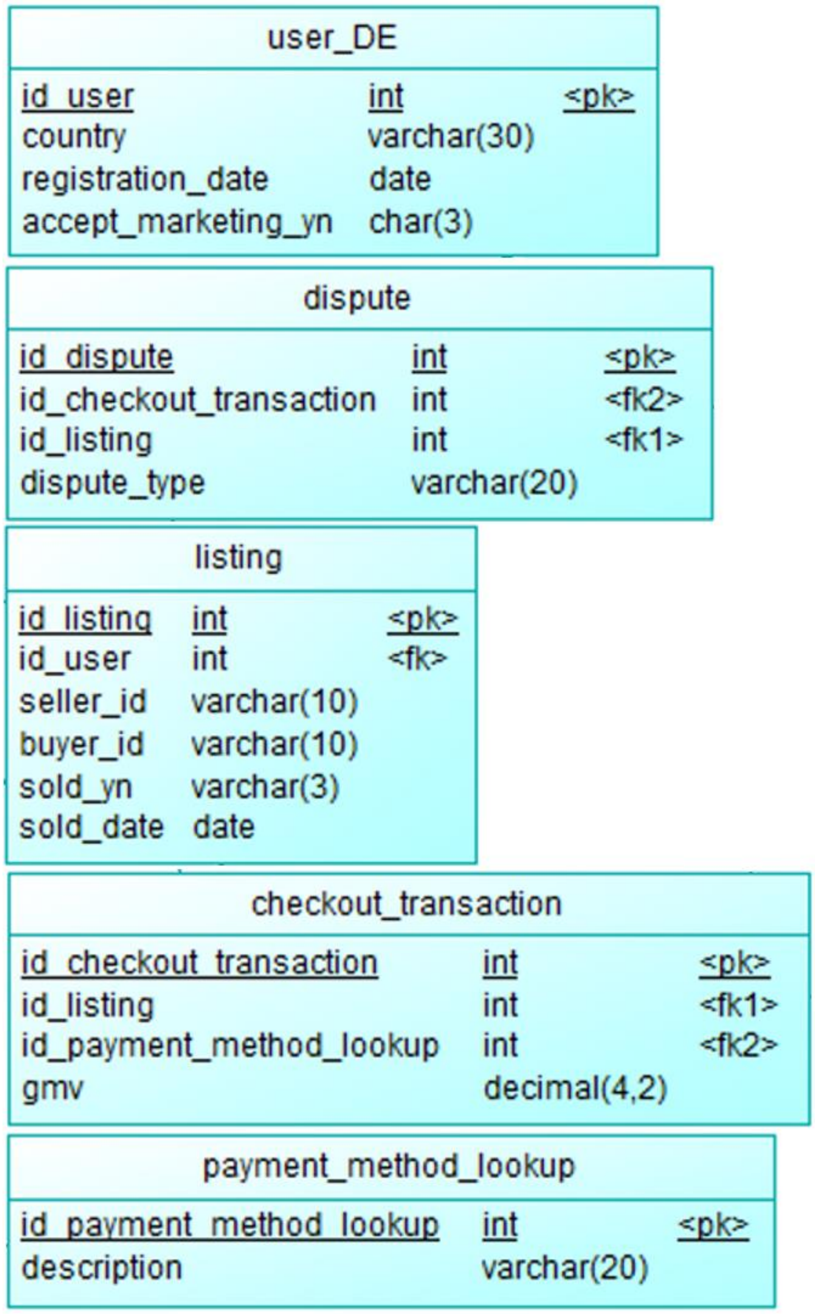
1. Total GMV (= gross merchandise volume ($)) from items on DE site sold in October 2018 from users accepting marketing.
2. List of DE users with a column for how many items they have sold, and a column for how many they have bought in October 2018, including those who have sold or bought 0 items.
3. List the 5 users spending the most GMV in October 2018.
4. Number of DE users who bought an item in October 2014 and paid via credit card split by whether they opened any dispute and by those that haven't.
5. List of sellers who have increased their GMV per month by at least 25% for the 3 months since registration. For example, Month 2 GMV must be more than a 25% increase on Month 1, and Month 3 must be more than a 25% increase on Month 2.

I've tried to analyze which attribute will be the primary key and which one is a foreign key:

Table	Columns	Comment
Listing	• Item_ID • Country (id_user from user_DE) • Seller_ID • Buyer_ID • Sold_YN • GMV • Sold_date	• Contains a row for every item listed on the site. • Buyer_ID is NULL if item didn't sell. • The table Listing joins to User by Seller_ID/ Buyer_ID and User_ID.
User (my sample table name: user_DE)	• User_ID • Country • Registration_date • Accept_marketing_YN	• Contains row for every registered user.
Checkout (my sample table name: checkout_transaction)	• Item_ID (id_listing) • Transaction_ID • Payment_method_ID (id_payment_method_lookup) • GMV	• Contains a row for every item sold, assume all sold items are paid for. • Joins to Payment_method_lookup by Payment_method_ID.
Payment_method_lookup	• Payment_method_ID • Description	• The payment_mthd_desc for credit card = 'CC'.
Dispute	• Item_ID (id_listing) • Transaction_ID (id_transaction_checkout) • Dispute_type	• Contains a row for every transaction which had a dispute.

I've also noticed that comments are important.

Structure for all created tables (my idea to solve this case):



user_DE		
id_user	int	<pk>
country	varchar(30)	
registration_date	date	
accept_marketing_yn	char(3)	

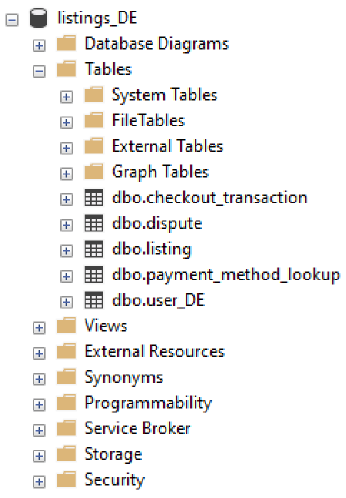
dispute		
id_dispute	int	<pk>
id_checkout_transaction	int	<fk2>
id_listing	int	<fk1>
dispute_type	varchar(20)	

listing		
id_listing	int	<pk>
id_user	int	<fk>
seller_id	varchar(10)	
buyer_id	varchar(10)	
sold_yn	varchar(3)	
sold_date	date	

checkout_transaction		
id_checkout_transaction	int	<pk>
id_listing	int	<fk1>
id_payment_method_lookup	int	<fk2>
gmv	decimal(4,2)	

payment_method_lookup		
id_payment_method_lookup	int	<pk>
description	varchar(20)	

Database (listings_DE) created in Microsoft SQL Server Management Studio 18:



```
⊟ listings_DE
    ⊞ Database Diagrams
    ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.checkout_transaction
        ⊞ dbo.dispute
        ⊞ dbo.listing
        ⊞ dbo.payment_method_lookup
        ⊞ dbo.user_DE
    ⊞ Views
    ⊞ External Resources
    ⊞ Synonyms
    ⊞ Programmability
    ⊞ Service Broker
    ⊞ Storage
    ⊞ Security
```

Insert sample data to analyze better (in **MS SQL**):

'user_DE' table:

```
insert into user_DE values ('Germany', '2014-10-01', 'Yes');
insert into user_DE values ('Germany', '2014-10-02', 'Yes');
insert into user_DE values ('Germany', '2014-10-10', 'Yes');
insert into user_DE values ('Germany', '2018-10-05', 'No');
insert into user_DE values ('Germany', '2018-10-07', 'No');
insert into user_DE values ('Germany', '2018-10-17', 'No');
insert into user_DE values ('Germany', '2018-10-20', 'Yes');
insert into user_DE values ('Germany', '2018-10-12', 'Yes');
insert into user_DE values ('Germany', '2018-10-18', 'Yes');
```

'listing' table:

```
insert into listing values(
(select id_user from user_DE where registration_date='2014-10-01'),
'1', '231', 'Yes', '2014-10-02');
insert into listing values(
(select id_user from user_DE where registration_date='2014-10-02'),
'2', '342', 'Yes', '2014-10-03');
insert into listing values(
(select id_user from user_DE where registration_date='2014-10-10'),
'3', '562', 'Yes', '2014-10-11');
insert into listing values(
(select id_user from user_DE where registration_date='2018-10-05'),
'4', 'Null', 'No', Null);
insert into listing values(
(select id_user from user_DE where registration_date='2018-10-07'),
'5', 'Null', 'No', Null);
insert into listing values(
(select id_user from user_DE where registration_date='2018-10-17'),
'6', 'Null', 'No', Null);
insert into listing values(
(select id_user from user_DE where registration_date='2018-10-20'),
'7', '945', 'Yes', '2018-10-21');
insert into listing values(
(select id_user from user_DE where registration_date='2018-10-12'),
'8', '620', 'Yes', '2018-10-13');
insert into listing values(
(select id_user from user_DE where registration_date='2018-10-18'),
'9', '728', 'Yes', '2018-10-19');
```

'payment_method_lookup' table:

```
insert into payment_method_lookup values ('CC')
insert into payment_method_lookup values ('cash')
```

'checkout_transaction' table:

```
insert into checkout_transaction values(
(select id_listing from listing where seller_id='1'),
(select id_payment_method_lookup from payment_method_lookup where description='CC'),
'22.55');
insert into checkout_transaction values(
(select id_listing from listing where seller_id ='2'),
(select id_payment_method_lookup from payment_method_lookup where description='CC'),
'45.25');
insert into checkout_transaction values(
(select id_listing from listing where seller_id ='3'),
(select id_payment_method_lookup from payment_method_lookup where description='CC'),
'32.62');
insert into checkout_transaction values(
(select id_listing from listing where seller_id ='7'),
(select id_payment_method_lookup from payment_method_lookup where description='cash'),
'23.41');
insert into checkout_transaction values(
(select id_listing from listing where seller_id ='8'),
(select id_payment_method_lookup from payment_method_lookup where description='cash'),
'19.55');
insert into checkout_transaction values(
(select id_listing from listing where seller_id ='9'),
(select id_payment_method_lookup from payment_method_lookup where description='cash'),
'77.30');
```

'dispute' table:

```
insert into dispute values(
(select id_listing from listing where seller_id='1'),
(select id_payment_method_lookup from payment_method_lookup where description='CC'),
'damaged product');
```

Ans. 1.

First I've checked my all 'checkout_transaction', 'gmv' inserts (all sold dates not only October 2018 and not only users which accepted marketing):

```
select avg (c.gmv) as "gross merchandise volume"
from checkout_transaction c
```

Then I've wrote the proper query with including the conditions (Total GMV is 'gross merchandise volume ($)'):

```
SELECT avg (c.gmv) as "gross merchandise volume",
l.sold_date as "sold date", u.accept_marketing_yn
as "accept marketing only Yes"
FROM checkout_transaction c
inner JOIN
listing l ON
c.id_listing = l.id_listing
inner JOIN
user_DE u ON
u.id_user = l.id_user
where u.accept_marketing_yn='Yes'
and l.sold_date like '2018-10%'
group by l.sold_date, u.accept_marketing_yn
```

Ans. 2 (in **SQL**).

A query to see how many users sold or bought 0 items:

```
select l.sold_yn, l.sold_date
```

```
from listing l
where l.sold_yn = 'No'
and l.sold_date is Null
```

After my sample inserts I have
3 'No' in my database, where ‘No’ means users those sold or bought 0 items.

So the proper query from question should be:

```
select u.id_user as "user", l.sold_yn as "sold items",
l.sold_date as "sold date"
from user_DE u
inner join
listing l on
l.id_user = u.id_user
WHERE YEAR(sold_date)=2018
and MONTH(sold_date)=10
or l.sold_yn='No'
group by u.id_user, l.sold_yn, l.sold_date
```

And just for comparison, a query with all sold dates (not only October 2018) in my sample database:

```
select u.id_user as "user", l.sold_yn as "sold items",
l.sold_date as "sold date"
from user_DE u
inner join
listing l on
l.id_user = u.id_user
group by u.id_user, l.sold_yn, l.sold_date
```

<u>Ans. 2 (**Pseudocode Python**).</u>

# Create a database instance, and connect to it:

from databases import Database

database = Database('sqlite:///listing_DE.db')

await database.connect()

# Create a table:

query = """CREATE TABLE Listing (id INTEGER PRIMARY KEY, Seller_id VARCHAR(10), Buyer_id VARCHAR(10),

Sold_YN VARCHAR(3), GMV DECIMAL(4,2), Sold_date DATE"""

await database.execute(query=query)

# Insert sample data:

```
query = "INSERT INTO Listing(Seller_id, Buyer_id, Sold_YN, GMV, Sold_date) " \
        "VALUES (:Seller_id, :Buyer_id, :Sold_YN, :GMV, :Sold_date)"

values = [
        {"Seller_id": "1", "Buyer_id": "231", "Sold_YN": "Yes", "GMV": "22.55", "Sold_date": "2014-10-01"},
        {"Seller_id": "2", "Buyer_id": "342", "Sold_YN": "Yes", "GMV": "45.25", "Sold_date": "2014-10-02"},
        {"Seller_id": "3", "Buyer_id": "562", "Sold_YN": "Yes", "GMV": "32.62", "Sold_date": "2014-10-10"},
        {"Seller_id": "4", "Buyer_id": "Null", "Sold_YN": "No", "GMV": "Null", "Sold_date": "2018-10-05"},
        {"Seller_id": "5", "Buyer_id": "Null", "Sold_YN": "No", "GMV": "Null", "Sold_date": "2018-10-07"},
        {"Seller_id": "6", "Buyer_id": "Null", "Sold_YN": "No", "GMV": "Null", "Sold_date": "2018-10-17"},
        {"Seller_id": "7", "Buyer_id": "945", "Sold_YN": "Yes", "GMV": "23.41", "Sold_date": "2018-10-20"},
        {"Seller_id": "8", "Buyer_id": "620", "Sold_YN": "Yes", "GMV": "19.55", "Sold_date": "2018-10-12"},
        {"Seller_id": "9", "Buyer_id": "728", "Sold_YN": "Yes", "GMV": "77.30", "Sold_date": "2018-10-18"},
    ]

await database.execute_many(query=query, values=values)

# Run a database query:
query = "SELECT * FROM Listing WHERE YEAR(Sold_date)=2018"
rows = await database.fetch_all(query=query)
print('Listing:', rows)
```

<u>Ans. 3.</u>

TOP() in MS SQL:

```
SELECT top (5) u.id_user as "user",
c.gmv as "GMV", l.sold_date as "sold date"
FROM user_DE u
inner join
listing l on
```

```
l.id_user = u.id_user
inner join
checkout_transaction c on
c.id_listing = l.id_listing
WHERE YEAR(sold_date)=2018
and MONTH(sold_date)=10
group by  c.gmv, u.id_user, l.sold_date
order by c.gmv desc
```

or LIMIT in MySQL:

```
SELECT u.id_user as "user",
c.gmv as "GMV", l.sold_date as "sold date"
FROM user_DE u
inner join
listing l on
l.id_user = u.id_user
inner join
checkout_transaction c on
c.id_listing = l.id_listing
WHERE YEAR(sold_date)=2018
and MONTH(sold_date)=10
group by  c.gmv, u.id_user, l.sold_date
order by c.gmv desc

LIMIT 5;
```

Ans. 4.

First I've typed all 'disputes' to check how many I have:

```
select * from dispute;
```

and in my case is only one 'dispute_type': 'damaged product', so the proper query will be:

```
SELECT u.id_user as "DE users",
l.sold_date as "sold date",
p.description as "payment method",
d.dispute_type as "dispute type"
FROM user_DE u
left join
listing l on
l.id_user = u.id_user
left join
checkout_transaction c on
c.id_listing = l.id_listing
left join
payment_method_lookup p on
p.id_payment_method_lookup = c.id_payment_method_lookup
left join
dispute d on
d.id_checkout_transaction = c.id_checkout_transaction
and
```

```
d.id_listing = l.id_listing
WHERE YEAR(sold_date)=2014
and MONTH(sold_date)=10
and p.description like 'CC'
group by  u.id_user, l.sold_date, p.description, d.dispute_type
order by p.description;
```

Ans. 5.

This query it is not what it should to be from the question but I think proper one should be similar to this one:

```
SELECT
  c.gmv, sold_date as "sold date",
  CAST(100 * sum(c.gmv) OVER (ORDER BY c.gmv)
           / sum(c.gmv) OVER () AS numeric(10, 2)) percentage
FROM checkout_transaction c
join
listing l on
l.id_listing = c.id_listing
ORDER BY c.gmv asc;
```

Of course, the 25 percent increase and months should be taken into account.

Thank You!