1.1 SQL test

1. List all unique customer id that have completed Box order with cash as their payment method

```
Select distinct customer_no
from `bi-dwhdev-01.source.daily_order`
where lower(order_type) = 'box'
and lower(order_payment) = 'cash'
and lower(order_status) = 'completed';
```

2. Show number of completed booking transactions per service type, and sort it by number of completed booking from biggest to smallest

```
with transaction_service as (
Select distinct customer_no, order_time, order_type
FROM `bi-dwhdev-01.source.daily_order`
where lower(order_status) = 'completed'
)
Select order_type, count(*) as completed_booking
from transaction_service
group by 1
order by 2 desc;
```

3. Show customer's transaction distribution for completed RIDE orders between 1st - 10th of April 2018 (Distribution of customers that have done 1 transaction, 2, 3,4,etc)

```
with transaction_service as (
Select customer_no, floor(transactions/5) as txn_bin
from
(Select customer_no, count(*) as transactions
from
(Select distinct customer_no, order_time
FROM `bi-dwhdev-01.source.daily_order`
where lower(order_status) = 'completed' and lower(order_type) = 'ride'
and date(order_time) between '2018-04-01' and '2018-04-10')
group by 1)
)
Select concat(txn_bin *5,'-',txn_bin*5+5) as txn, count(distinct customer_no) as
customers
from transaction_service
group by 1
```

4. List all customer id that have completed more than 1 gojek service type in April 2018. (Note: Please consider Ride and/or Car as 1 service, as we consider them as mobility

```
With customer_service as (
SELECT distinct customer_no,
case when lower(order_type) in ('car','ride') then 'ride'
    else order_type end as order_type
FROM `bi-dwhdev-01.source.daily_order`
where lower(order_status) = 'completed' and date(order_time) between '2018-04-01'
and '2018-04-30'
)
Select customer_no
from customer_service
```

```
group by 1
having count(distinct order_type) > 1
```

5. List all of customer id that had their FIRST 3 GOJEK orders EXACTLY IN THE following sequence: First order is RIDE, second is CAR and third is FOOD

```
With customer_service as (
Select customer_no, lower(order_type) as order_type, rn
from
(SELECT customer_no,order_type, row_number() over(partition by customer_no order by
order_time) as rn
FROM `bi-dwhdev-01.source.daily_order`)
where rn <= 3
)
Select customer_no
from
(Select customer_no, case when rn = 1 and order_type = 'ride' then 1 else 0 end as
rn1, case when rn = 2 and order_type = 'car' then 1 else 0 end as rn2,
case when rn = 3 and order_type = 'food' then 1 else 0 end as rn3
from customer_service)
where rn1 = 1 and rn2 = 1 and rn3 = 1
```

Question section 2

Create a summary table to show how many customers use different GO-JEK services in a daily basis, along with the combination of services used

```
with data_customer as (
Select distinct date(order_time) as order_date, customer_no, order_type,
order_payment
from `bi-dwhdev-01.source.daily_order`
where lower(order_status) = 'completed'
),
customer_order_type as (
Select order_date, no_of_service, order_payment, count(distinct customer_no) as
total_customer
from
(Select order_date, customer_no, count(distinct order_type) as no_of_service,
string_agg(distinct order_payment,',' order by order_payment) as order_payment
from data_customer
group by 1,2)
group by 1,2,3
),
order_type_customer as (
Select order_date, order_type, order_payment, no_of_service, count(distinct
customer_no) as total_customer_per_order_type
from
```

```sql
(Select order_date, customer_no, count(distinct order_type) as no_of_service,
string_agg(distinct order_type,',' order by order_type) as order_type,
string_agg(distinct order_payment,',' order by order_payment) as order_payment
from data_customer
group by 1,2)
group by 1,2,3,4
)
Select a.order_date, a.no_of_service, total_customer, order_type,
total_customer_per_order_type, a.order_payment
from customer_order_type as a
left join order_type_customer as b
on a.order_date = b.order_date and a.order_payment = b.order_payment and
a.no_of_service = b.no_of_service
order by 1;
```