SQL Case Study

SQL CASE STUDY

FOODIE-FI

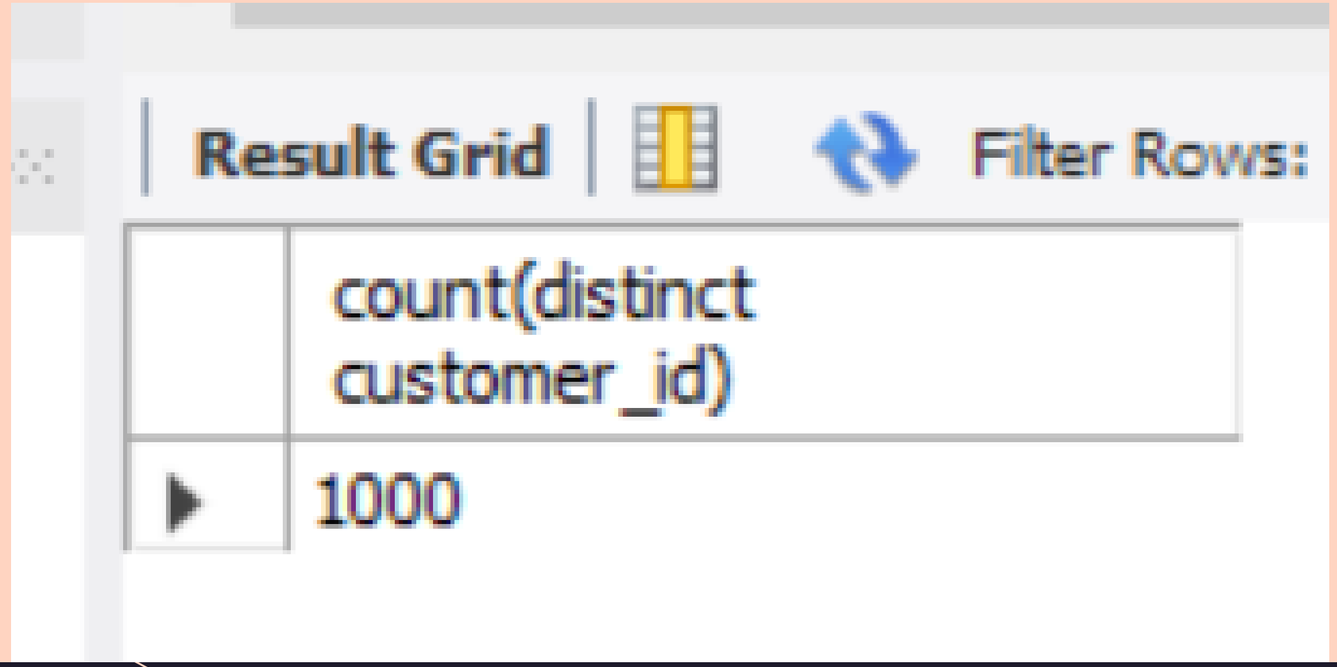
INTRODUCTION:

Foodie-Fi is a cool website where you can sign up to watch lots of cooking shows from all over the world. Danny, who started Foodie-Fi, knows that to make the website even better, he has to look at what shows people like and how they watch them. This study is all about how Danny uses numbers to make good choices for Foodie-Fi, making it a favorite place for people who love food.

We're going to look closely at two big pieces of information (tables) from the Foodie-Fi database and even try to make a totally new piece of information (table).

1. How many customers has Foodie-Fi ever had?

```
select count(distinct customer_id) from subscriptions;
```



2. What is the monthly distribution of trial plan start_date values for our dataset - use the start of the month as the group by value?

```
select month(start_date) as months, count(customer_id) as num_customers
from subscriptions
group by months
order by months;
```

	months	num_customers
▶	1	236
	2	195
	3	245
	4	217
	5	214
	6	204
	7	221
	8	235
	9	225
	10	230
	11	208
	12	220

3. What plan start_date values occur after the year 2020 for our dataset? Show the breakdown by count of events for each plan_name?

```sql
select p.plan_name, p.plan_id, count(*) as total_count
from plans p
join subscriptions s
on s.plan_id = p.plan_id
where s.start_date >= "2021-01-01"
group by p.plan_id, p.plan_name
order by p.plan_id;
```

plan_name	plan_id	total_count
basic monthly	1	8
pro monthly	2	60
pro annual	3	63
churn	4	71

4. What is the customer count and percentage of customers who have churned rounded to 1 decimal place?

```
select count(*) as total_churn,
round(count(*) * 100 / (select count(distinct customer_id) from subscriptions),1) as percentage
from subscriptions
where plan_id = 4;
```

total_churn	percentage
307	30.7

5. How many customers have churned straight after their initial free trial - what percentage is this rounded to the nearest whole number?

```sql
with chrn_cte as (
    select *,
    LAG(plan_id, 1) over(partition by customer_id order by plan_id) as prev_plan
    from subscriptions)
    select count(prev_plan) as count_chrn,
    round(count(*) * 100 / (select count(distinct customer_id) from subscriptions),0) as perc_chrn
    from chrn_cte
    where plan_id = 4 and
    prev_plan = 0;
```

count_chrn	perc_chrn
92	9

6. What is the number and percentage of customer plans after their initial free trial?

```sql
with next_plane_cte as (
    select *, lead(plan_id,1) over(partition by customer_id order by plan_id) as next_plane
    from subscriptions)
    select next_plane,
    count(*) as num_cust,
    round(count(*) * 100 / (select count(distinct customer_id) from subscriptions),0) as perc_next_plane
    from next_plane_cte
    where next_plane is not null and plan_id = 0
    group by next_plane
    order by next_plane;
```

next_plane	num_cust	perc_next_plane
1	546	55
2	325	33
3	37	4
4	92	9

7. What is the customer count and percentage breakdown of all 5 plan_name values at 2020-12-31?

```sql
select
    plan_name,
    COUNT(distinct customer_id) as customer_count,
    ROUND(COUNT(distinct customer_id) / (select COUNT(distinct customer_id) from subscriptions) * 100, 1) as percentage
from subscriptions
join plans on subscriptions.plan_id = plans.plan_id
where start_date <= '2020-12-31'
group by plan_name;
```

plan_name	customer_count	percentage
basic monthly	538	53.8
churn	236	23.6
pro annual	195	19.5
pro monthly	479	47.9
trial	1000	100.0

8. How many customers have upgraded to an annual plan in 2020?

```sql
select COUNT(distinct customer_id) as annual_plan_upgrades
from subscriptions
where plan_id = 3
and year(start_date) = 2020;
```

annual_plan_upgrades
195

9. How many days on average does it take for a customer to an annual plan from the day they join Foodie-Fi?

```sql
select
    round(avg(DATEDIFF(s2.start_date, s1.start_date)),0) as avg_days_to_annual_plan
from subscriptions s1
join subscriptions s2 on s1.customer_id = s2.customer_id
where s1.plan_id = 0
and s2.plan_id = 3
and s2.start_date > s1.start_date;
```

avg_days_to_annual_plan
105

10. Can you further breakdown this average value into 30 day periods (i.e. 0-30 days, 31-60 days etc)?

```sql
select
    case
        when avg_days_to_annual_plan between 0 and 30 then '0-30 days'
        when avg_days_to_annual_plan between 31 and 60 then '31-60 days'
        when avg_days_to_annual_plan between 61 and 90 then '61-90 days'
        else 'More than 90 days'
    end as period,
    COUNT(*) as customers_count
from (
    select
        DATEDIFF(s2.start_date, s1.start_date) as avg_days_to_annual_plan
    from subscriptions s1
    join subscriptions s2 on s1.customer_id = s2.customer_id
    where s1.plan_id = 0
    and s2.plan_id = 3
    and s2.start_date > s1.start_date
) as sub
group by period
order by period;
```

period	customers_count
0-30 days	49
31-60 days	24
61-90 days	34
More than 90 days	151

11. How many customers downgraded from a pro monthly to a basic monthly plan in 2020?

```sql
with next_plan as (
    select *, lead(plan_id,1) over(partition by customer_id order by start_date, plan_id) as plan
    from subscriptions
)
    select count(distinct customer_id) as downgrade
    from next_plan np
    left join plans p on p.plan_id = np.plan_id
    where p.plan_name = "pro monthly" and np.plan = 1 and start_date < "2020-12-31";
```

	downgrade
▶	0