Capstone 2

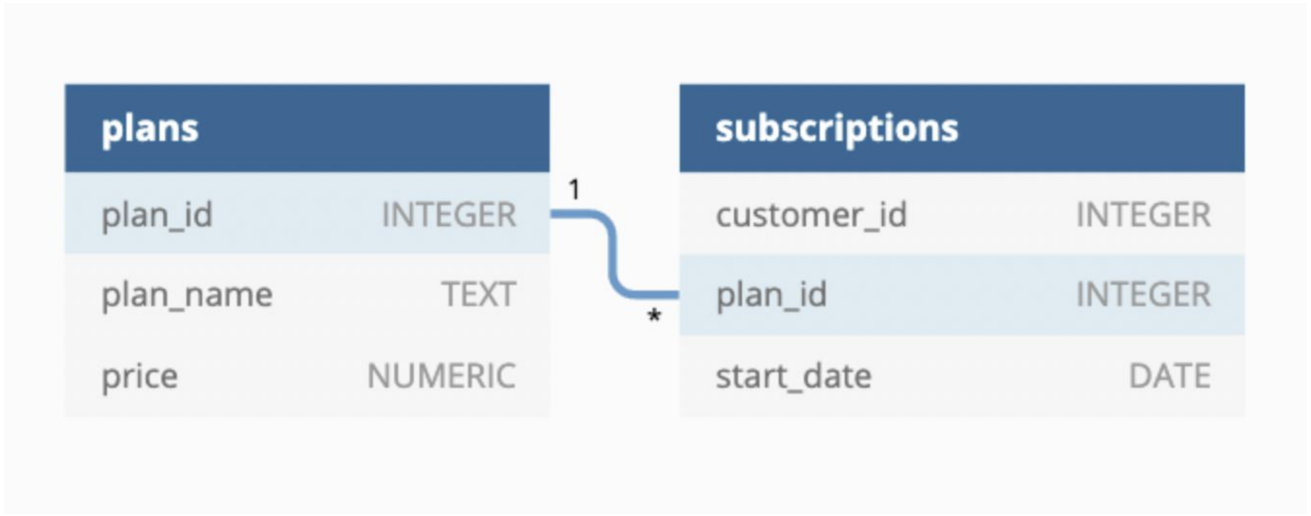
Relevel
by Unacademy

Introduction to Case

Ron and his buddies founded **Foodie-Fi** 🥑 and began **selling monthly and annual subscriptions**, providing clients with unrestricted on-demand **access to exclusive cuisine videos** from around the world.

This case study focuses on the use of subscription-style **digital data** to answer critical business questions about the **customer journey, payments, and business performance.**

Table Relationship



plans

plan_id	INTEGER
plan_name	TEXT
price	NUMERIC

subscriptions

customer_id	INTEGER
plan_id	INTEGER
start_date	DATE

Table ‘plans’ Description

There are 5 customer plans:

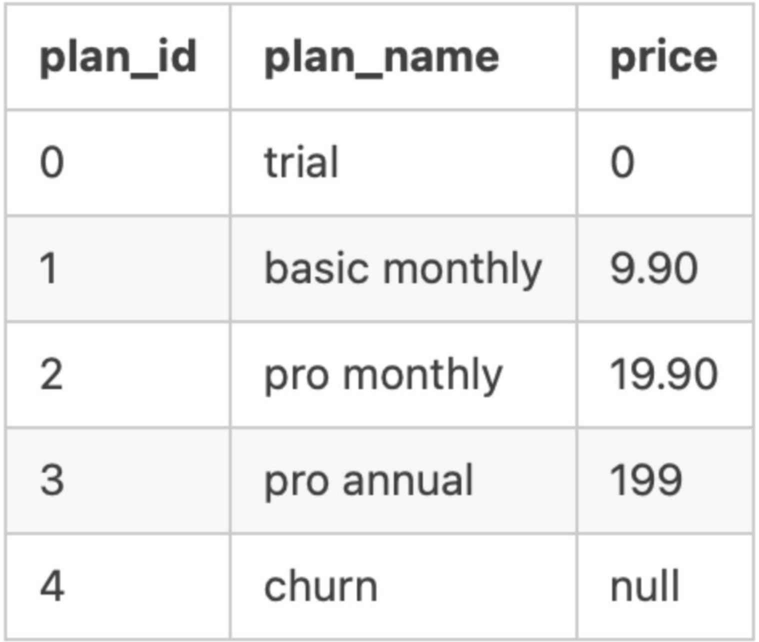
Trial— Customers sign up for a 7-day free trial and will be automatically enrolled in the **pro monthly** subscription plan unless they unsubscribe, downgrade to basic, or upgrade to an annual pro plan during the trial.

Basic plan — Customers have limited access and can only stream their videos with the basic package, which is only available monthly for $9.90.

Pro plan — Customers on the Pro plan have no watch time limits and can download videos for offline viewing. Pro plans begin at $19.90 per month or $199 for a yearly subscription.

When clients cancel their Foodie-Fi service, a Churn plan record with a null pricing is created, but their plan continues until the end of the billing cycle.

Table ‘plans’ Description (contd.)



plan_id	plan_name	price
0	trial	0
1	basic monthly	9.90
2	pro monthly	19.90
3	pro annual	199
4	churn	null

Table 1: plans

Table ‘subscriptions’ Description

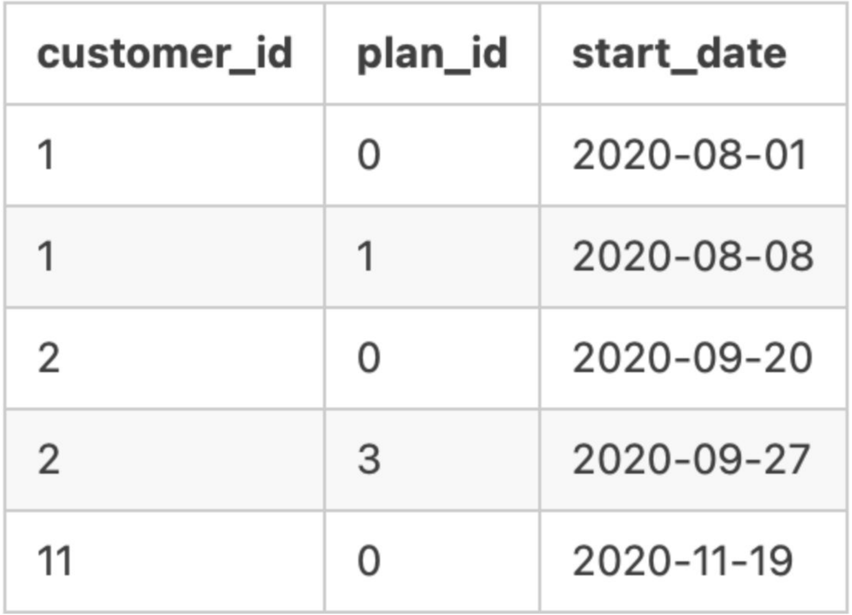
Customer subscriptions display the precise date on which their specific plan id begins.

If a customer downgrades from a pro plan or cancels their subscription — the higher program will remain in place until the period expires — the start date in the subscriptions table will reflect the date the actual plan changes.

When clients upgrade their account from a basic plan to a pro or annual pro plan, the higher plan becomes active immediately.

When customers cancel their subscription, they will retain access until the end of their current billing cycle, but the start date will be the day they opted to quit their service.

Table ‘subscriptions’ Description (contd.)



customer_id	plan_id	start_date
1	0	2020-08-01
1	1	2020-08-08
2	0	2020-09-20
2	3	2020-09-27
11	0	2020-11-19

Database

- The database can be accessed here: https://www.db-fiddle.com/f/jbahqhW5AQwqV1RZ2xExEz/0

- This tool will also be used to query

Problem Statement - 1

How many customers has Foodie-Fi ever had?

Solution - 1

```
SELECT

  COUNT(DISTINCT customer_id) AS unique_customer

FROM dbo.subscriptions;
```

Problem Statement - 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.

Solution - 2

```
SELECT
        DATE_PART('month',start_date) AS month_date,
        TO_CHAR(start_date, 'Month') AS month_name,
        COUNT(*) AS trial_subscriptions
FROM dbo.subscriptions s
JOIN dbo.plans p
  ON s.plan_id = p.plan_id
WHERE s.plan_id = 0
GROUP BY DATE_PART('month',start_date),
  TO_CHAR(start_date, 'Month')
ORDER BY month_date ASC;
```

Problem Statement - 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.

Solution - 3

```
SELECT
  p.plan_id,
  p.plan_name,
  COUNT(*) AS events
FROM dbo.subscriptions s
JOIN dbo.plans p
  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;
```

Problem Statement - 4

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

Solution - 4

```
SELECT

  COUNT(*) AS churn_count,

  ROUND(100 * COUNT(*)::NUMERIC / (

    SELECT COUNT(DISTINCT customer_id)

    FROM dbo.subscriptions),1) AS churn_percentage

FROM dbo.subscriptions s

JOIN dbo.plans p

  ON s.plan_id = p.plan_id

WHERE s.plan_id = 4;
```

Problem Statement - 5

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

Solution - 5

```
WITH ranking AS (

SELECT

  s.customer_id,

  s.plan_id,

  p.plan_name,

  ROW_NUMBER() OVER (

    PARTITION BY s.customer_id

    ORDER BY s.plan_id) AS plan_rank

FROM dbo.subscriptions s
```

Solution - 5

```
JOIN dbo.plans p

  ON s.plan_id = p.plan_id)



SELECT

  COUNT(*) AS churn_count,

  ROUND(100 * COUNT(*) / (

    SELECT COUNT(DISTINCT customer_id)

    FROM dbo.subscriptions),0) AS churn_percentage

FROM ranking

WHERE plan_id = 4 -- Filter to churn plan

  AND plan_rank = 2
```

Problem Statement - 6

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

Solution - 6

```
WITH next_plan_cte AS (

SELECT

  customer_id,

  plan_id,

  LEAD(plan_id, 1) OVER(

    PARTITION BY customer_id

    ORDER BY plan_id) as next_plan

FROM dbo.subscriptions)
```

Solution - 6

```
SELECT
  next_plan,
  COUNT(*) AS conversions,
  ROUND(100 * COUNT(*)/ (
    SELECT COUNT(DISTINCT customer_id)
    FROM dbo.subscriptions),1) AS conversion_percentage
FROM next_plan_cte
WHERE next_plan IS NOT NULL
  AND plan_id = 0
GROUP BY next_plan
ORDER BY next_plan;
```

Problem Statement - 7

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

Solution - 7

```
WITH next_plan AS(

SELECT

  customer_id,

  plan_id,

  start_date,

  LEAD(start_date, 1) OVER(PARTITION BY customer_id ORDER BY start_date) as next_date

FROM dbo.subscriptions

WHERE start_date <= '2020-12-31'

),
```

Solution - 7

```
customer_breakdown AS (
 SELECT
  plan_id,
  COUNT(DISTINCT customer_id) AS customers
 FROM next_plan
 WHERE
  (next_date IS NOT NULL AND (start_date < '2020-12-31'
   AND next_date > '2020-12-31'))
  OR (next_date IS NULL AND start_date < '2020-12-31')
 GROUP BY plan_id)
```

Solution - 7

```
SELECT plan_id, customers,

  ROUND(100 * customers / (

    SELECT COUNT(DISTINCT customer_id)

    FROM dbo.subscriptions),1) AS percentage

FROM customer_breakdown

GROUP BY plan_id, customers

ORDER BY plan_id;
```

Problem Statement - 8

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

Solution - 8

```
SELECT
  COUNT(DISTINCT customer_id) AS unique_customer
FROM foodie_fi.subscriptions
WHERE plan_id = 3
  AND start_date <= '2020-12-31';
```

Problem Statement - 9

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

Solution - 9

```
-- Filter results to customers at trial plan = 0

WITH trial_plan AS

  (SELECT

    customer_id,

    start_date AS trial_date

  FROM dbo.subscriptions

  WHERE plan_id = 0

),

-- Filter results to customers at pro annual plan = 3

annual_plan AS
```

Solution - 9

```
 (SELECT
   customer_id,
   start_date AS annual_date
 FROM dbo.subscriptions
 WHERE plan_id = 3
)
SELECT
 ROUND(AVG(annual_date - trial_date),0) AS avg_days_to_upgrade
FROM trial_plan tp
JOIN annual_plan ap
 ON tp.customer_id = ap.customer_id;
```

Problem Statement - 10

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

Solution - 10

```
-- Filter results to customers at trial plan = 0
WITH trial_plan AS
  (SELECT
    customer_id,
    start_date AS trial_date
  FROM dbo.subscriptions
  WHERE plan_id = 0
),
-- Filter results to customers at pro annual plan = 3
annual_plan AS
```

Solution - 10

```
  (SELECT
    customer_id,
    start_date AS annual_date
  FROM dbo.subscriptions
  WHERE plan_id = 3
),
-- Sort values above in buckets of 12 with range of 30 days each
bins AS
  (SELECT
    WIDTH_BUCKET(ap.annual_date - tp.trial_date, 0, 360, 12) AS     avg_days_to_upgrade
  FROM trial_plan tp
```

Solution - 10

```
  JOIN annual_plan ap
    ON tp.customer_id = ap.customer_id)


SELECT
  ((avg_days_to_upgrade - 1) * 30 || ' - ' ||   (avg_days_to_upgrade) * 30) || ' days' AS breakdown,
  COUNT(*) AS customers
FROM bins
GROUP BY avg_days_to_upgrade
ORDER BY avg_days_to_upgrade
```

Problem Statement - 11

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

Solution - 11

```
WITH next_plan_cte AS (
 SELECT
  customer_id,
  plan_id,
  start_date,
  LEAD(plan_id, 1) OVER(
   PARTITION BY customer_id
   ORDER BY plan_id) as next_plan
 FROM dbo.subscriptions)
```

Solution - 11

```
SELECT
  COUNT(*) AS downgraded
FROM next_plan_cte
WHERE start_date <= '2020-12-31'
  AND plan_id = 2
  AND next_plan = 1;
```

Conclusion