

Joining Data Like a Pro

TelstraTelco Churn Analysis - Video 2 of 15 | 18 minutes

The Business Case: Telstra Customer Churn Analysis

The Challenge: Find customers paying monthly fees but NOT using the network

The Stakes: \$500,000 in monthly revenue at risk

The Problem:

- Customer data lives in one table (names, plans, fees)
- Usage data lives in another table (call records, data used)
- How do we connect them? **JOINS!**

The Impact:

WRONG JOIN

= Hide 3,487 at-risk customers = Lose \$500K/month

RIGHT JOIN

= Reveal all at-risk customers = Save \$2M/year

What You'll Learn:

- INNER JOIN vs LEFT JOIN
- When each JOIN type matters
- How to find "missing" data
- Business impact of NULL values

The Data Structure - Two Separate Tables

Table 1: telstra_customers

10 customers total

- **customer_id:** 101, 102, 103... 110
- **customer_name:** Sarah Mitchell, James Chen, etc.
- **plan_tier:** Bronze, Silver, Gold
- **monthly_fee:** \$29, \$59, \$99
- **signup_date:** When they joined

Table 2: telstra_usage

14 usage records (November-December 2024)

- **usage_id:** 1, 2, 3... 14
- **customer_id:** Links to customers table
- **usage_month:** 2024-11, 2024-12
- **data_used_gb:** How much data consumed
- **calls_made:** Number of calls

The Critical Problem:

Customers 108, 109, 110 signed up in December

They have NO usage records yet

They're paying fees but not using service

HIGH CHURN RISK!

The Question: How do we find these customers?

INNER JOIN - The Wrong Tool (Hides Problems)

The Query:

```
SELECT c.customer_id, c.customer_name, c.plan_tier, c.monthly_fee, u.data_used_gb  
FROM telstra_customers c  
INNER JOIN telstra_usage u  
ON c.customer_id = u.customer_id  
WHERE u.usage_month = '2024-12';
```

What INNER JOIN Does:

Shows ONLY customers who have BOTH:

- A customer record AND
- A matching usage record

The Result:

- 7 rows returned
- Only customers 101-107 appear
- **Customers 108, 109, 110 are HIDDEN!**

The Business Problem:

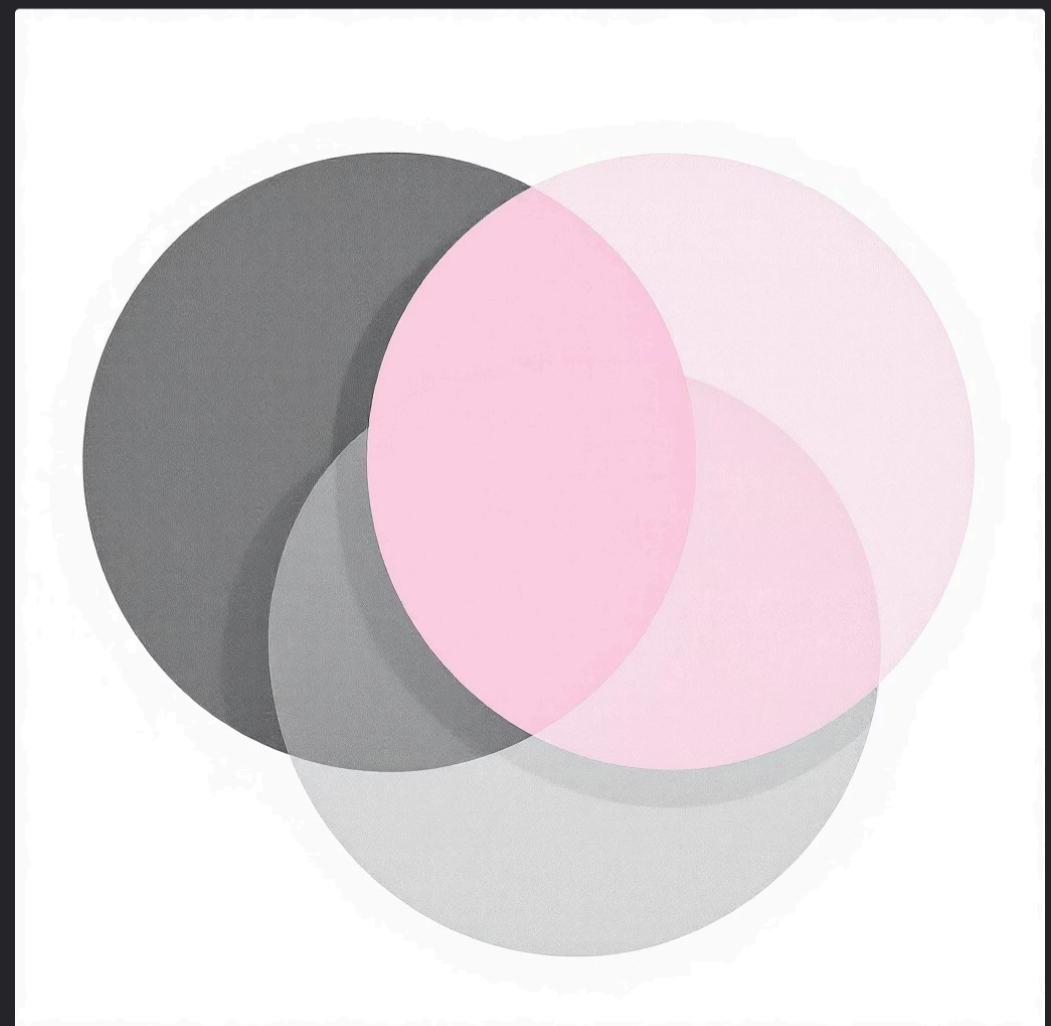
- Marketing never sees the 3 at-risk customers
- No retention offers sent
- 3,487 real customers like this = \$500K monthly loss

Why This Happens:

INNER JOIN says "Show me ONLY matching records"
No usage record? Customer disappears from results!

Visual Venn Diagram:

INNER JOIN = Only the overlapping middle section



LEFT JOIN - The Right Tool (Reveals Problems)

The Query:

```
SELECT c.customer_id, c.customer_name, c.plan_tier, c.monthly_fee, u.data_used_gb,
       CASE WHEN u.usage_id IS NULL THEN 'CHURN RISK' ELSE 'Active' END AS status
  FROM telstra_customers c
  LEFT JOIN telstra_usage u
    ON c.customer_id = u.customer_id
   WHERE u.usage_month = '2024-12' OR u.usage_month IS NULL
  ORDER BY u.data_used_gb ASC NULLS FIRST;
```

What LEFT JOIN Does:

- Shows ALL customers from left table (customers)
- Even if they have NO matching usage record
- Missing usage shows as NULL

The Result:

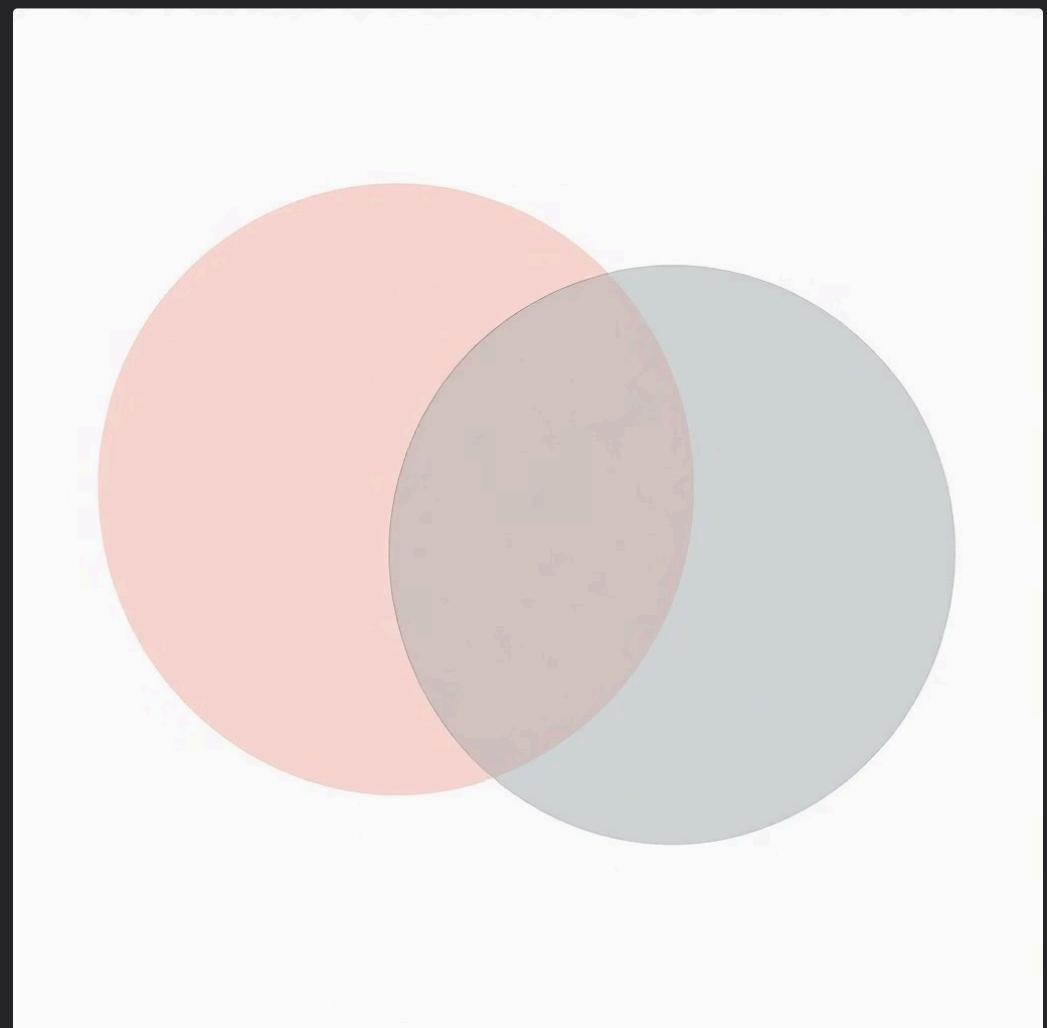
- 10 rows returned
- ALL customers appear (101-110)
- Customers 108, 109, 110 show NULL for usage
- CASE statement flags them as "CHURN RISK"

The Business Solution:

- Marketing sees ALL 10 customers
- 3 flagged as CHURN RISK
- Retention offers sent immediately
- \$500K monthly revenue protected

Visual Venn Diagram:

LEFT JOIN = Entire left circle + overlapping section



7

INNER JOIN rows

(hides problems)

10

LEFT JOIN rows

(reveals problems)

Business Impact & What You Learned

Real-World Application:

This exact analysis runs DAILY at Telstra

15%

Customer churn prevented

Marketing uses it for targeted retention campaigns

\$2M

Saved per year

INNER JOIN vs LEFT JOIN Decision Guide:

Use INNER JOIN when:

- You only want matching records
- Example: "Show me customers WHO MADE purchases"
- Excludes non-purchasers (that's the goal)

Use LEFT JOIN when:

- You want ALL records from left table
- You need to find what's MISSING
- Example: "Show me ALL customers, flag who DIDN'T purchase"
- Reveals problems, gaps, churn risks

SQL Concepts Mastered:

✓ INNER JOIN syntax and behavior ✓ LEFT JOIN syntax and behavior ✓ ON clause (join condition) ✓ IS NULL (detecting missing data) ✓ CASE WHEN (conditional logic) ✓ NULLS FIRST (sorting NULL values) ✓ Table aliases (c, u for readability)

Common Mistake to Avoid:

WHERE vs JOIN condition with LEFT JOIN

WRONG: WHERE u.usage_month = '2024-12' (converts to INNER JOIN!)

RIGHT: WHERE u.usage_month = '2024-12' OR u.usage_month IS NULL

Business Value Delivered:

	\$500,000 monthly revenue at risk identified		3,487 churn-risk customers flagged
	\$2,000,000 annual savings from churn prevention		15% churn reduction through targeted offers

Practice Exercise:

Find all 2024 signups with ZERO usage across ALL months

Hint: LEFT JOIN with COUNT to find NULLs

Next Video Preview:

Video 3: Aggregations - Business Metrics

CBA Credit Card Monthly Reporting for CFO

Learn: GROUP BY, HAVING, multiple aggregates

Duration: 16 minutes

Success Checklist:

- ✓ Understand INNER vs LEFT JOIN difference
- ✓ Know when to use each JOIN type
- ✓ Can find "missing" data using LEFT JOIN
- ✓ Recognize business impact of NULL values
- ✓ Ready to prevent million-dollar mistakes