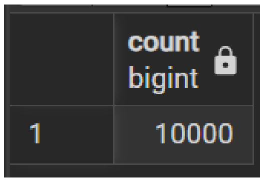
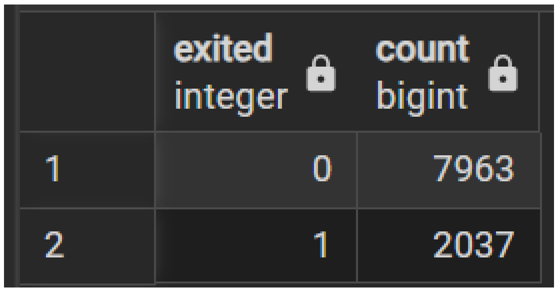
# ***SQL QUERIES AND THEIR OUTPUTS FOR CUSTOMER CHURN MODELS***

```
DROP TABLE IF EXISTS customer_churn;

CREATE TABLE customer_churn (
   customer_id BIGINT,
   geography TEXT,
   gender TEXT,
   age INT,
   credit_score INT,
   tenure INT,
   balance NUMERIC,
   num_of_products INT,
   has_cr_card INT,
   is_active_member INT,
   estimated_salary NUMERIC,
   exited INT
);

select count(*) from customer_churn;
```



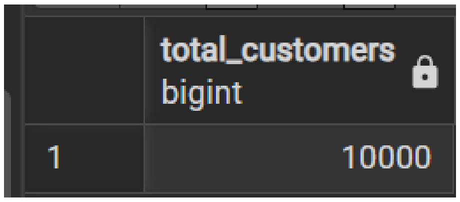
| | count bigint |
|---|---|
| 1 | 10000 |

```
select exited, count(*) from customer_churn group by exited;
```



| | exited integer | count bigint |
|---|---|---|
| 1 | 0 | 7963 |
| 2 | 1 | 2037 |

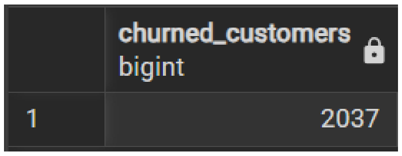
```
-- KPI 1: Total Customers --

select count(customer_id) as Total_Customers from customer_churn;
```



| | total_customers bigint |
|---|---|
| 1 | 10000 |

-- KPI 2: Churned Customers --

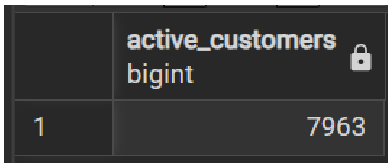
```
select count(customer_id) as Churned_Customers from customer_churn where exited = 1;
```



|  | churned_customers bigint |
|---|---|
| 1 | 2037 |

-- KPI 3: Active Customers (Not Churned) --

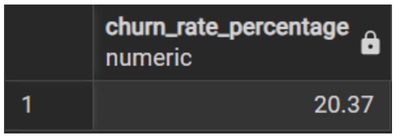
```
select count(customer_id) as Active_Customers from customer_churn where exited = 0;
```



|  | active_customers bigint |
|---|---|
| 1 | 7963 |

-- KPI 4: Churn Rate (%) --

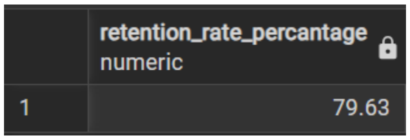
```
select round(count(case when exited = 1 then 1 end) * 100.0 / count(customer_id), 2) as
Churn_rate_percentage from customer_churn;
```



|  | churn_rate_percentage numeric |
|---|---|
| 1 | 20.37 |

-- KPI 5: Retention Rate (%) --

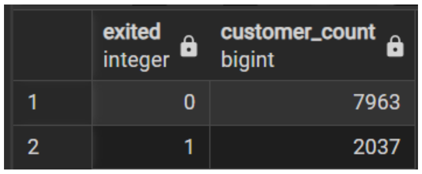
```
select round(count(case when exited = 0 then 1 end) * 100.0 / count(customer_id), 2) as
Retention_rate_percantage from customer_churn;
```



|  | retention_rate_percantage numeric |
|---|---|
| 1 | 79.63 |

-- KPI 6: Churn vs Retained Distribution --

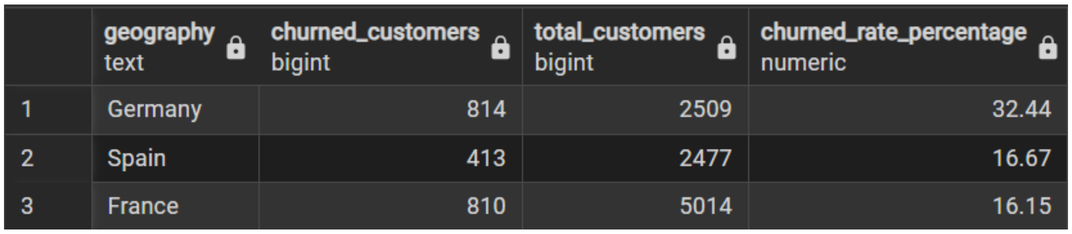
```
select exited, count(customer_id) as Customer_count from customer_churn group by exited;
```



|  | exited integer | customer_count bigint |
|---|---|---|
| 1 | 0 | 7963 |
| 2 | 1 | 2037 |

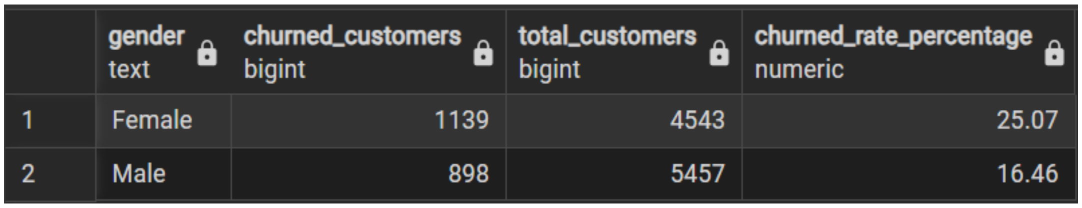
-- KPI 7: Churn Rate by Geography (IMPORTANT) --

```
select geography, count(case when exited = 1 then 1 end) as Churned_customers,
                    count(customer_id) as Total_customers,
                    round(count(case when exited = 1 then 1 end ) * 100.0 / count(customer_id), 2)
                    as Churned_rate_percentage
from customer_churn
group by geography
order by Churned_rate_percentage desc;
```



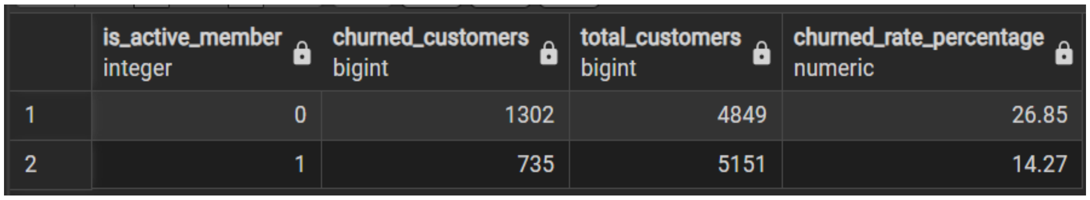
| | geography text | churned_customers bigint | total_customers bigint | churned_rate_percentage numeric |
|---|---|---|---|---|
| 1 | Germany | 814 | 2509 | 32.44 |
| 2 | Spain | 413 | 2477 | 16.67 |
| 3 | France | 810 | 5014 | 16.15 |

```
-- KPI 8: Churn Rate by Gender --

select gender, count(case when exited = 1 then 1 end) as Churned_customers,
                    count(customer_id) as Total_customers,
                    round(count(case when exited = 1 then 1 end ) * 100.0 / count(customer_id), 2)
                    as Churned_rate_percentage
from customer_churn
group by gender;
```



| | gender text | churned_customers bigint | total_customers bigint | churned_rate_percentage numeric |
|---|---|---|---|---|
| 1 | Female | 1139 | 4543 | 25.07 |
| 2 | Male | 898 | 5457 | 16.46 |

```
-- KPI 9: Churn Rate by IsActiveMember --

select is_active_member, count(case when exited = 1 then 1 end) as Churned_customers,
                    count(customer_id) as Total_customers,
                    round(count(case when exited = 1 then 1 end ) * 100.0 / count(customer_id), 2)
                    as Churned_rate_percentage
from customer_churn
group by is_active_member;
```

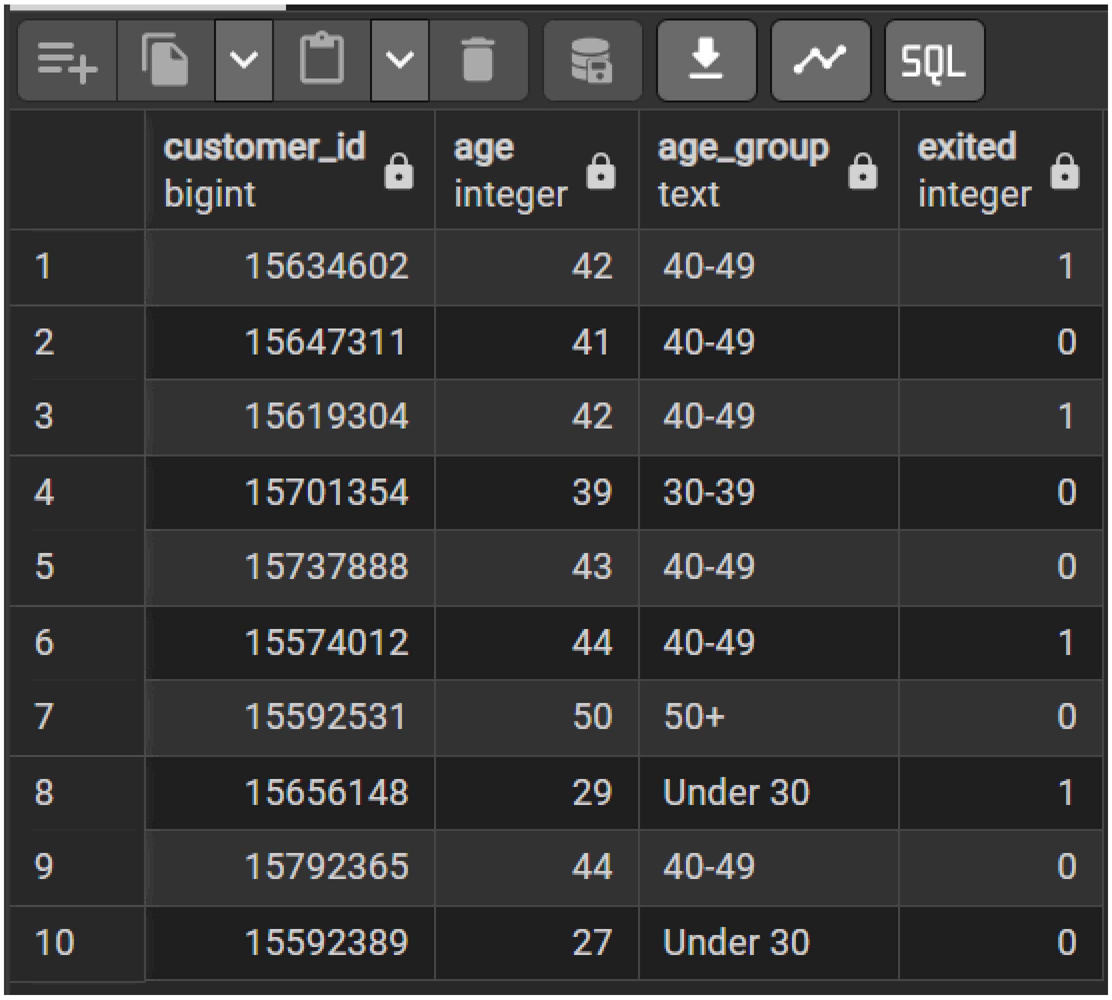


| | is_active_member integer | churned_customers bigint | total_customers bigint | churned_rate_percentage numeric |
|---|---|---|---|---|
| 1 | 0 | 1302 | 4849 | 26.85 |
| 2 | 1 | 735 | 5151 | 14.27 |

-- I calculated churn KPIs in SQL including total customers, churn rate, retention rate, and churn segmentation by geography, gender, and activity status. --

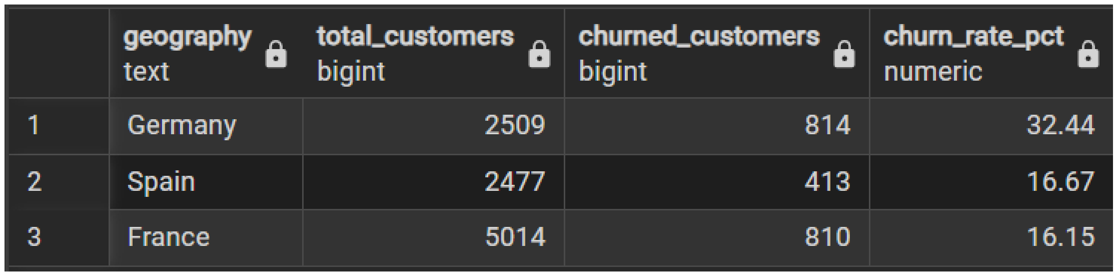
Advanced SQL
-- CASE WHEN – CUSTOMER SEGMENTATION --
-- Dividing customers based on age groups --
-- CASE WHEN allows us to create derived columns for analysis.--

```
select
    customer_id,
    age,
    case
        when age < 30 then 'Under 30'
        when age between 30 and 39 then '30-39'
        when age between 40 and 49 then '40-49'
        else '50+'
    end as age_group,
    exited
from customer_churn
limit 10;
```



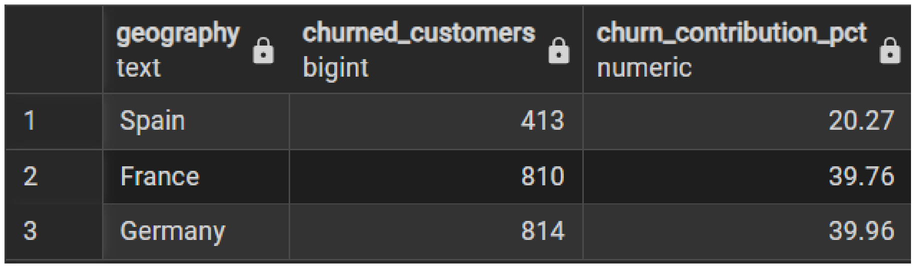
| | customer_id bigint | age integer | age_group text | exited integer |
|---|---|---|---|---|
| 1 | 15634602 | 42 | 40-49 | 1 |
| 2 | 15647311 | 41 | 40-49 | 0 |
| 3 | 15619304 | 42 | 40-49 | 1 |
| 4 | 15701354 | 39 | 30-39 | 0 |
| 5 | 15737888 | 43 | 40-49 | 0 |
| 6 | 15574012 | 44 | 40-49 | 1 |
| 7 | 15592531 | 50 | 50+ | 0 |
| 8 | 15656148 | 29 | Under 30 | 1 |
| 9 | 15792365 | 44 | 40-49 | 0 |
| 10 | 15592389 | 27 | Under 30 | 0 |

```
-- CTE (WITH) – CLEAN & READABLE SQL --
-- Churn rate by geography using CTE --
-- CTEs improve query readability and allow step-by-step transformations --

with churn_geo as (
   select
      geography,
      count(*) as total_customers,
      count(case when exited = 1 then 1 end) as churned_customers
   from customer_churn
   group by geography
)
select
   geography,
   total_customers,
   churned_customers,
   ROUND(churned_customers * 100.0 / total_customers, 2) as churn_rate_pct
from churn_geo
order by churn_rate_pct desc;
```



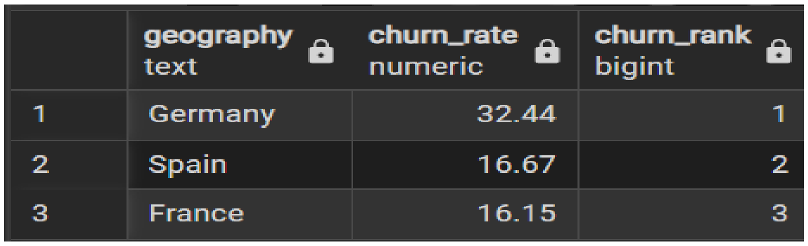
|  | geography text | total_customers bigint | churned_customers bigint | churn_rate_pct numeric |
|---|---|---|---|---|
| 1 | Germany | 2509 | 814 | 32.44 |
| 2 | Spain | 2477 | 413 | 16.67 |
| 3 | France | 5014 | 810 | 16.15 |

```
-- WINDOW FUNCTION – OVER() --
-- Compare each geography’s churn with overall churn --
-- Window functions allow calculations across result sets without collapsing rows --

select
   geography,
   count(case when exited = 1 then 1 end) as churned_customers,
   round(
      count(case when exited = 1 then 1 end) * 100.0 /
      sum(count(case when exited = 1 then 1 end)) over (),
      2
   ) as churn_contribution_pct
from customer_churn
group by geography;
```



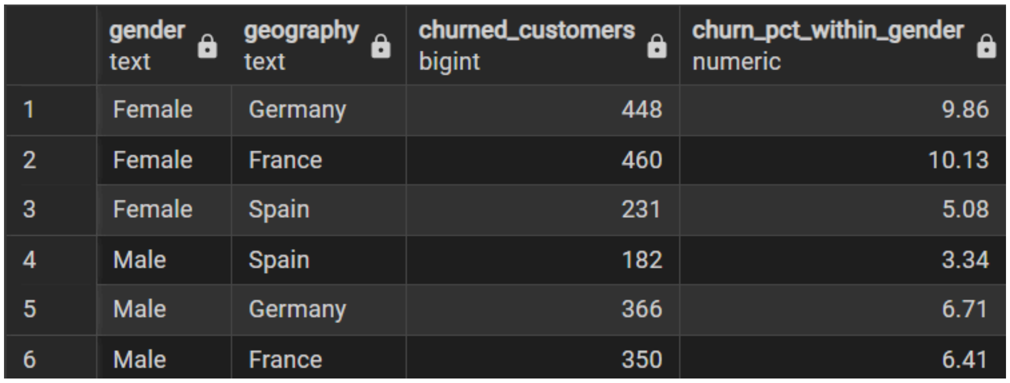
|  | geography text | churned_customers bigint | churn_contribution_pct numeric |
|---|---|---|---|
| 1 | Spain | 413 | 20.27 |
| 2 | France | 810 | 39.76 |
| 3 | Germany | 814 | 39.96 |

```
-- RANKING – Top Churn Regions --
-- Rank regions by churn rate --
-- RANK() helps identify top churn-prone regions --

WITH churn_geo AS (
   SELECT
      geography,
      COUNT(*) AS total_customers,
      COUNT(CASE WHEN exited = 1 THEN 1 END) AS churned_customers
   FROM customer_churn
   GROUP BY geography
)
SELECT
   geography,
   ROUND(churned_customers * 100.0 / total_customers, 2) AS churn_rate,
   RANK() OVER (
      ORDER BY churned_customers * 1.0 / total_customers DESC
   ) AS churn_rank
FROM churn_geo;
```



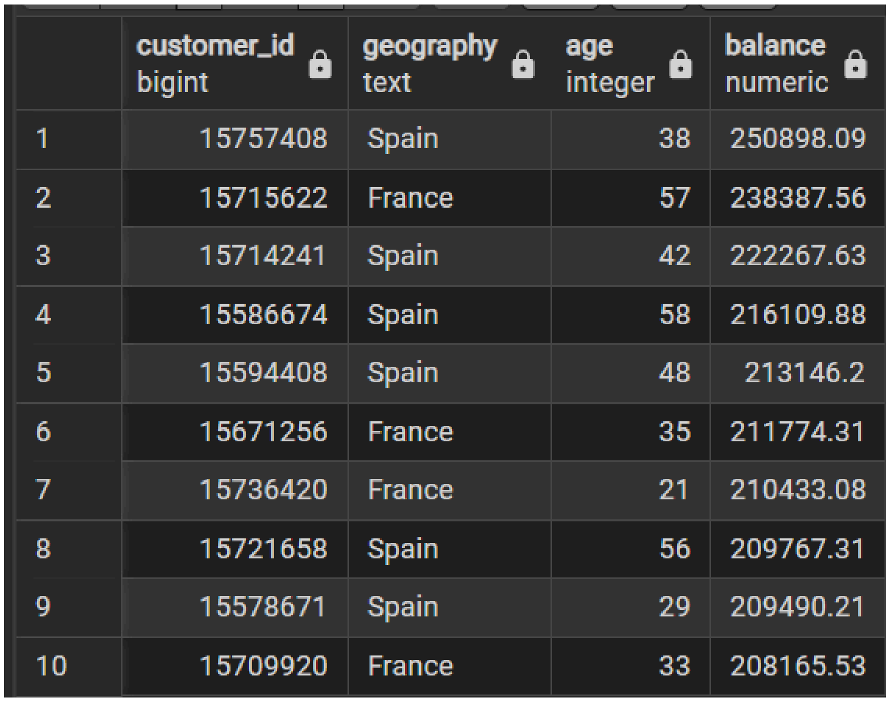
|  | geography text | churn_rate numeric | churn_rank bigint |
|---|---|---|---|
| 1 | Germany | 32.44 | 1 |
| 2 | Spain | 16.67 | 2 |
| 3 | France | 16.15 | 3 |

```
-- WINDOW FUNCTION – PARTITION BY --
-- Churn rate within each gender --
-- PARTITION BY allows group-level calculations without GROUP BY collapse --

SELECT
   gender,
   geography,
   COUNT(CASE WHEN exited = 1 THEN 1 END) AS churned_customers,
   ROUND(
      COUNT(CASE WHEN exited = 1 THEN 1 END) * 100.0 /
      SUM(COUNT(*)) OVER (PARTITION BY gender),
      2
   ) AS churn_pct_within_gender
FROM customer_churn
GROUP BY gender, geography
ORDER BY gender;
```



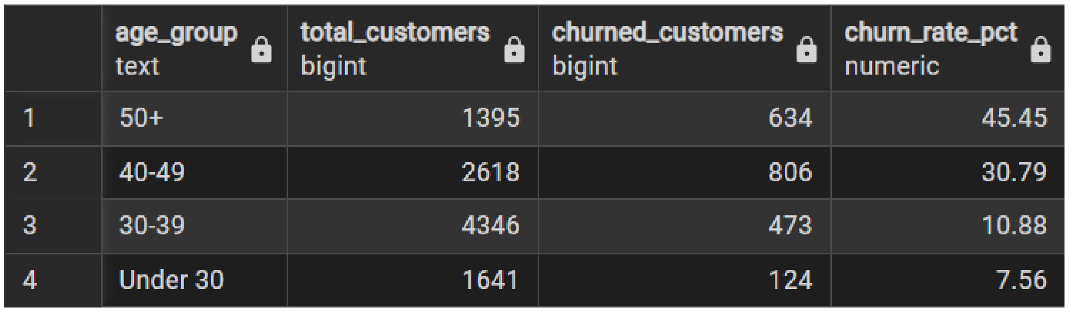
| | gender text | geography text | churned_customers bigint | churn_pct_within_gender numeric |
|---|---|---|---|---|
| 1 | Female | Germany | 448 | 9.86 |
| 2 | Female | France | 460 | 10.13 |
| 3 | Female | Spain | 231 | 5.08 |
| 4 | Male | Spain | 182 | 3.34 |
| 5 | Male | Germany | 366 | 6.71 |
| 6 | Male | France | 350 | 6.41 |

```
-- TOP-N HIGH-RISK CUSTOMERS --
-- Identify top 10 high-balance churned customers --
-- This helps retention teams target high-value churned customers --

SELECT
    customer_id,
    geography,
    age,
    balance
FROM customer_churn
WHERE exited = 1
ORDER BY balance DESC
LIMIT 10;
```



| | customer_id bigint | geography text | age integer | balance numeric |
|---|---|---|---|---|
| 1 | 15757408 | Spain | 38 | 250898.09 |
| 2 | 15715622 | France | 57 | 238387.56 |
| 3 | 15714241 | Spain | 42 | 222267.63 |
| 4 | 15586674 | Spain | 58 | 216109.88 |
| 5 | 15594408 | Spain | 48 | 213146.2 |
| 6 | 15671256 | France | 35 | 211774.31 |
| 7 | 15736420 | France | 21 | 210433.08 |
| 8 | 15721658 | Spain | 56 | 209767.31 |
| 9 | 15578671 | Spain | 29 | 209490.21 |
| 10 | 15709920 | France | 33 | 208165.53 |

```
-- ADVANCED KPI – CHURN RATE BY AGE GROUP --
-- CTEs + CASE WHEN simplify complex segmentation logic --

WITH age_segments AS (
   SELECT
      CASE
         WHEN age < 30 THEN 'Under 30'
         WHEN age BETWEEN 30 AND 39 THEN '30-39'
         WHEN age BETWEEN 40 AND 49 THEN '40-49'
         ELSE '50+'
      END AS age_group,
      exited
   FROM customer_churn
)
SELECT
   age_group,
   COUNT(*) AS total_customers,
   COUNT(CASE WHEN exited = 1 THEN 1 END) AS churned_customers,
   ROUND(
      COUNT(CASE WHEN exited = 1 THEN 1 END) * 100.0 / COUNT(*),
      2
   ) AS churn_rate_pct
FROM age_segments
GROUP BY age_group
ORDER BY churn_rate_pct DESC;
```



| | age_group text | total_customers bigint | churned_customers bigint | churn_rate_pct numeric |
|---|---|---|---|---|
| 1 | 50+ | 1395 | 634 | 45.45 |
| 2 | 40-49 | 2618 | 806 | 30.79 |
| 3 | 30-39 | 4346 | 473 | 10.88 |
| 4 | Under 30 | 1641 | 124 | 7.56 |

```
-- I used advanced SQL techniques such as
-- CTEs, CASE statements, and window functions to analyze customer churn patterns,
-- rank high-risk segments, and compute churn contributions across regions and demographics
```