

- Tables import in SQL and reading all data from table

SCHEMAS

Filter objects

cn_sql_05062025

employee

hr

orders

project1ba

Tables

churn

customers

subscriptions

transactions

Views

Stored Procedures

Functions

sys

Administration

Schemas

Information

Schema: project1ba

Limit to 50000 rows

1 • use project1ba;

2

3 • select * from churn;

4 • select * from customers;

5 • select * from subscriptions;

6 • select * from transactions;

7

8

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	TransactionID	CustomerID	TransactionDate	Amount	TransactionType
▶	1	15	2024-01-01	32	Purchase
	2	22	2024-01-02	52	Refund
	3	43	2024-01-03	165	Purchase
	4	87	2024-01-04	134	Purchase
	5	34	2024-01-05	158	Purchase
	6	56	2024-01-06	170	Refund
	7	23	2024-01-07	196	Purchase
	8	11	2024-01-08	195	Purchase
	9	76	2024-01-09	143	Purchase
	10	45	2024-01-10	69	Refund
	11	66	2024-01-11	198	Purchase
	12	54	2024-01-12	81	Refund
	13	39	2024-01-13	120	Purchase
	14	82	2024-01-14	102	Refund
	15	67	2024-01-15	141	Refund
	16	19	2024-01-16	140	Refund
	17	25	2024-01-17	122	Refund
	18	33	2024-01-18	156	Refund
	19	88	2024-01-19	61	Purchase
	20	91	2024-01-20	118	Refund
	21	17	2024-01-21	73	Refund
	22	48	2024-01-22	142	Purchase
	23	61	2024-01-23	68	Refund
	24	29	2024-01-24	200	Purchase
	25	74	2024-01-25	111	Refund
	26	37	2024-01-26	143	Purchase
	27	46	2024-01-27	99	Refund
	28	72	2024-01-28	44	Refund

churn 1 customers 2 subscriptions 3 transactions 4 ×

Query 1: Total how many customers do we have?

```
select count (*) as Total_customers, count(distinct CustomerID) as distinct_customers from customers
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Total_customers	distinct_customers
▶	300	300

Query 2: Show the first name, last name and status of all customers

```
select FirstName, LastName, Status from customers;
```

Result Grid			
Filter Rows:			
	FirstName	LastName	Status
▶	John	Doe	Active
	Jane	Smith	Inactive
	Alice	Johnson	Active
	Bob	Brown	Inactive
	Charlie	Davis	Active
	Diana	Clark	Active
	Eva	Harris	Inactive
	Frank	Garcia	Active
	Grace	Miller	Inactive
	Hank	Wilson	Active
	Ivy	Martinez	Active
	Jack	White	Inactive
	Kelly	Lewis	Active
	Liam	Walker	Inactive
	Mia	Hall	Active
	Noah	Allen	Inactive

Query 3: Retrieve all customers from the "North America" region.

```
select * from customers
where Region= "North America";
```

Result Grid								
Filter Rows:								
Export: Wrap Cell Content:								
	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region
▶	1	John	Doe	john.doe@example.com	123-456-7890	2022-01-10	Active	North America
	4	Bob	Brown	bob.brown@example.com	234-567-8901	2019-06-25	Inactive	North America
	7	Eva	Harris	eva.h@example.com	567-890-1234	2020-12-22	Inactive	North America
	10	Hank	Wilson	hank.w@example.com	890-123-4567	2020-05-27	Active	North America
	13	Kelly	Lewis	kelly.l@example.com	123-456-7890	2018-04-15	Active	North America
	16	Noah	Allen	noah.a@example.com	456-789-0123	2021-06-19	Inactive	North America
	19	Quinn	Scott	quinn.s@example.com	789-012-3456	2018-01-05	Active	North America
	22	Tina	Perez	tina.p@example.com	012-345-6789	2020-08-09	Inactive	North America
	25	Wendy	Cook	wendy.c@example.com	345-678-9012	2019-07-21	Active	North America
	28	Zane	Evans	zane.e@example.com	678-901-2345	2019-12-13	Inactive	North America
	31	Leo	Morgan	leo.m@example.com	321-654-0987	2023-01-10	Active	North America
	34	Penny	Ford	penny.f@example.com	654-321-0987	2021-10-30	Inactive	North America
	37	Tom	Wood	tom.w@example.com	987-321-6540	2023-03-03	Active	North America
	40	Will	Stone	will.s@example.com	321-654-9870	2019-07-20	Active	North America

Query 4: Retrieve all customer count who have an "Active" status.

```
select count(*) as active_customer from customers
where status="Active";
```

Result Grid		Filter Rows:
	active_customer	
▶	152	

Query 5: Find customers who joined after January 1, 2021.

```
select count(*) as customer_count from customers
where JoinDate> "01-01-2001";
```

Result Grid		Filter Rows:
	customer_count	
▶	298	

Query 6: Retrieve all active customers from the Europe region.

```
select count(*) as customer_count from customers
where Region="Europe" and Status="Active";
```

Result Grid		Filter Rows:
	customer_count	
▶	47	

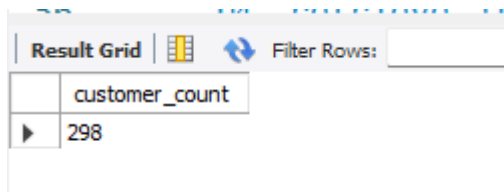
Query 7: Retrieve all customers who joined in the year 2021.

```
select count(*) as customer_count from customers
where year(JoinDate)= 2021;
```

Result Grid		Filter Rows:
	customer_count	
▶	80	

Query 8: Count customers having an email id containing example

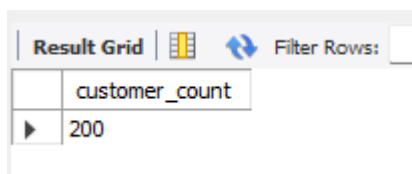
```
select count(*) as customer_count from customers
where Email like "%example%";
```



customer_count
298

Query 9: Retrieve all subscriptions with an "Annual" plan type.

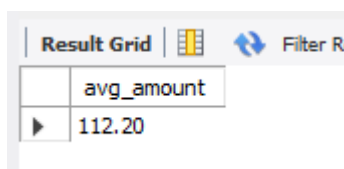
```
select count(*) as customer_count from subscriptions
where PlanType = "Annual";
```



customer_count
200

Query 10: Calculate the average amount of all transactions.

```
select round(avg(Amount),2) as avg_amount from transactions;
```



avg_amount
112.20

Query 11: Retrieve all transactions along with the first, last name, transaction id and amount of the customers who made them.

```
select c.FirstName,c.LastName,t.TransactionID,t.Amount
from transactions t join customers c
on t.CustomerID=c.CustomerID;
```

Result Grid					Filter Rows:	Ex
	FirstName	LastName	TransactionID	Amount		
▶	Ivy	Martinez	8	195		
	Mia	Hall	1	32		
	Olivia	Young	21	73		
	Quinn	Scott	16	140		
	Steve	Baker	32	32		
	Tina	Perez	2	52		
	Uma	Collins	7	196		
	Wendy	Cook	17	122		
	Xander	Cooper	50	65		
	Xander	Cooper	38	45		
	Zane	Evans	47	153		
	Abby	Flores	24	200		
	Leo	Morgan	45	66		

Query 12: Retrieve the 5 most recent transactions from the Transactions table.

```
select * from transactions
order by TransactionDate desc
limit 5;
```

Result Grid						Filter Rows:	Export:	Wrap Cell Content:	Fe
	TransactionID	CustomerID	TransactionDate	Amount	TransactionType				
▶	216	270	2024-09-08	29	Purchase				
	215	269	2024-09-07	142	Purchase				
	214	268	2024-09-06	176	Purchase				
	213	267	2024-09-05	136	Refund				
	212	266	2024-09-04	46	Purchase				

Query 13: Retrieve all the reasons for churn listed in the Churn table.

```
select distinct Reason from churn;
```

Result Grid		Filter Rows
	Reason	
▶	Poor Customer Service	
	High Prices	
	Product Quality Issues	
	Lack of Product Features	
	Competitor Offerings	
	Inconvenience	
	Lack of Engagement	
	Unmet Expectations	
	Price Increases	
	Personal Circumstances	




Query 14: Retrieve how many people left for each reason.

```
select Reason, count(*) as customer_count
from churn
group by Reason;
```

Result Grid		Filter Rows:
	Reason	customer_count
▶	Poor Customer Service	10
	High Prices	10
	Product Quality Issues	10
	Lack of Product Features	10
	Competitor Offerings	10
	Inconvenience	10
	Lack of Engagement	10
	Unmet Expectations	10
	Price Increases	10
	Personal Circumstances	10



Query 15: Retrieve customer id, full name, email, plan type.

```
select c.CustomerID as customerid, concat(c.FirstName," ",c.LastName) as Full_n
from customers c join subscriptions s
on c.CustomerID=s.CustomerID;
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap				
	customerid	Full_name	Email	PlanType
▶	1	John Doe	john.doe@example.com	Annual
	2	Jane Smith	jane.smith@example.com	Monthly
	3	Alice Johnson	alice.j@example.com	Annual
	4	Bob Brown	bob.brown@example.com	Annual
	5	Charlie Davis	charlie.d@example.com	Monthly
	6	Diana Clark	diana.c@example.com	Annual
	7	Eva Harris	eva.h@example.com	Monthly
	8	Frank Garcia	frank.g@example.com	Annual
	9	Grace Miller	grace.m@example.com	Monthly
	10	Hank Wilson	hank.w@example.com	Annual
	11	Ivy Martinez	ivy.m@example.com	Annual
	12	Jack White	jack.w@example.com	Monthly
	13	Kelly Lewis	kelly.l@example.com	Annual

Query 16: Retrieve active customer count who have an annual subscription.

```
select count(*) as customer_count
from customers c join subscriptions s
on c.CustomerID = s.CustomerID
where s.PlanType="Annual" and c.Status="Active";
```

Result Grid  	
	customer_count
▶	82

Query 17: list down churn reasons and how many north American customers left for each reason

```
select c.Reason as Reasons, count(*) as churn_customer_count
from customers cr join churn c
on cr.CustomerID=c.CustomerID
where cr.Region="North America"
group by c.Reason;
```

Result Grid			Filter Rows:
	Reasons	churn_customer_count	
▶	Poor Customer Service	6	
	Lack of Product Features	5	
	High Prices	3	
	Product Quality Issues	4	
	Personal Circumstances	5	
	Lack of Engagement	6	
	Price Increases	3	
	Competitor Offerings	3	
	Inconvenience	2	
	Unmet Expectations	3	

Query 18: Retrieve all customer count who have not made any transaction with us

```
select count(*) as customer_no_transaction
from customers c left join transactions t
on c.CustomerID=t.CustomerID
where t.CustomerID is null;=c.CustomerID
where cr.Region="North America"
group by c.Reason;
```

Result Grid			Filter Rows:
	customer_no_transaction		
▶	86		

Query 19: Retrieve customer count who have more than one subscription

```
select count (*) as customer_count from (
SELECT CustomerID FROM subscriptions
GROUP BY CustomerID
HAVING COUNT(*) > 1) as mult_sub;
```

Result Grid			Filter Rows:
	customer_count		
▶	58		

Query 20: Retrieve transaction count made in last 6 month

```
select count(*) as Transactions_Last_6_Months
from transactions where TransactionDate >= (
    select max(TransactionDate) from transactions
) - INTERVAL 6 MONTH;
```

Result Grid	Filter Rows:
Transactions_Last_6_Months	
166	

Query 21: Retrieve customer full name who had annual plan and then churned.

```
SELECT      CONCAT(c.FirstName, ' ', c.LastName)      AS
Full_Name, s.PlanType, s.EndDate AS PlanEndDate, ch.ChurnDate
FROM customers c JOIN subscriptions s ON c.CustomerID = s.CustomerID
JOIN churn ch ON c.CustomerID = ch.CustomerID
WHERE s.PlanType = 'Annual' AND ch.ChurnDate > s.EndDate;
```

Result Grid

Filter Rows:

Export:

	Full_Name	PlanType	PlanEndDate	ChurnDate
▶	Liam Walker	Annual	07-07-2022	14-01-2024
	Liam Walker	Annual	07-08-2022	14-01-2024
	Uma Collins	Annual	02-01-2022	23-01-2024
	Wendy Cook	Annual	20-07-2020	24-01-2024
	Wendy Cook	Annual	20-07-2021	24-01-2024
	Wendy Cook	Annual	20-07-2020	24-01-2024
	Wendy Cook	Annual	20-07-2021	24-01-2024
	Abby Flores	Annual	21-02-2023	24-01-2024
	Abby Flores	Annual	21-02-2023	24-01-2024
	Brian Gonzalez	Annual	11-06-2021	30-01-2024
	Brian Gonzalez	Annual	14-12-2023	30-01-2024
	Leo Morgan	Annual	09-01-2025	14-01-2024
	Leo Morgan	Annual	09-01-2025	14-02-2024

Query 22: Calculate total revenue by region

```
select c.Region, sum(t.Amount) as Total_Amount
from customers c join transactions t
on c.CustomerID=t.CustomerID
group by c.Region;
```

Result Grid			Filter Rows:
	Region	Total_Amount	
▶	Europe	9011	
	Asia	7166	
	North America	7887	
		171	

Query 23: Retrieve how many customers are there who do not have any subscription

```
select count(*) as customer_no_transaction from (
select c.CustomerID
from customers c left join subscriptions s
on c.CustomerID=s.CustomerID
where s.CustomerID is null) as cust_tran_null;
```

Result Grid			Filter Rows:
	customer_no_transaction		
▶	1		

Query 24: Retrieve customer names who made the purchase after subscription ended.

```
select FirstName,LastName from customers where CustomerID in (
select distinct t.CustomerID
from transactions t join subscriptions s
on t.CustomerID=s.CustomerID
where t.TransactionDate>s.EndDate and t.TransactionType= "Purchase");
```

Result Grid			Filter Rows
	FirstName	LastName	
▶	Ivy	Martinez	
	Mia	Hall	
	Uma	Collins	
	Xander	Cooper	
	Abby	Flores	
	Penny	Ford	
	Tom	Wood	
	Vera	Scott	
	Clara	Bennett	
	Hannah	Ward	
	Julia	Morris	
	Lily	Martinez	
	Owen	Clark	

Query 25: identify churn reason by region and rank them by count

```
select c.Region , ch.Reason,
dense_rank() over(partition by c.Region order by count(*) desc) as count_reason
from customers c join churn ch
on c.CustomerID=ch.CustomerID
group by c.Region,ch.Reason
order by c.region, count_reason;
```

Result Grid			
	Filter Rows:		Export:
Region	Reason	count_reason	
Asia	High Prices	1	
Asia	Unmet Expectations	1	
Asia	Lack of Product Features	2	
Asia	Product Quality Issues	2	
Asia	Poor Customer Service	3	
Asia	Lack of Engagement	3	
Asia	Price Increases	3	
Asia	Competitor Offerings	4	
Asia	Personal Circumstances	4	
Asia	Inconvenience	4	
Europe	Inconvenience	1	
Europe	Competitor Offerings	2	
Europe	Price Increases	3	
Europe	Personal Circumstances	4	
Europe	Product Quality Issues	5	
Europe	Lack of Engagement	6	
Europe	High Prices	6	

Result Grid			
	Filter Rows:		Export:
Region	Reason	count_reason	
Europe	Product Quality Issues	5	
Europe	Lack of Engagement	6	
Europe	High Prices	6	
Europe	Unmet Expectations	6	
Europe	Poor Customer Service	6	
Europe	Lack of Product Features	6	
North America	Lack of Engagement	1	
North America	Poor Customer Service	1	
North America	Lack of Product Features	2	
North America	Personal Circumstances	2	
North America	Product Quality Issues	3	
North America	High Prices	4	
North America	Unmet Expectations	4	
North America	Competitor Offerings	4	
North America	Price Increases	4	
North America	Inconvenience	5	

Query 26: For each month in 2024, show number of churn and total transaction value

```
SELECT  date_format(t.TransactionDate,"%Y-%m")  AS  YearMonth,  count(dist
ch.CustomerID) as churn_count, sum(t.Amount) as total_amount
from transactions t left join churn ch
on t.CustomerID=ch.CustomerID
where year(t.TransactionDate)="2024"
group by YearMonth;
```

Result Grid			
Filter Rows:			
	YearMonth	churn_count	total_amount
▶	2024-01	20	4828
	2024-02	9	2767
	2024-04	16	6639
	2024-05	17	5684
	2024-07	0	2880
	2024-08	0	3442
	2024-09	0	847



Query 27: Find top 5 customers with highest average monthly spend (based on purchase)

```
select t.CustomerID, avg(t.Amount) as avg_monthly_spend
from transactions t
where t.TransactionType="Purchase"
group by t.CustomerID
order by avg_monthly_spend desc limit 5;
```

Result Grid		
Filter Rows:		
	CustomerID	avg_monthly_spend
▶	29	200.0000
	78	200.0000
	66	198.0000
	144	196.0000
	23	196.0000

Query 28: Calculate Revenue Growth Over Time period of 1 year
Calculate the monthly revenue growth over the last year.

```
SELECT DATE_FORMAT(TransactionDate,'%Y-%m') AS Month, SUM(Amount) AS
MonthlyRevenue
FROM Transactions
WHERE TransactionDate >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
GROUP BY Month
ORDER BY Month;
```

Result Grid					Filter Rows:	
	Month	MonthlyRevenue				
▶	2024-07	2880				
	2024-08	3442				
	2024-09	847				