

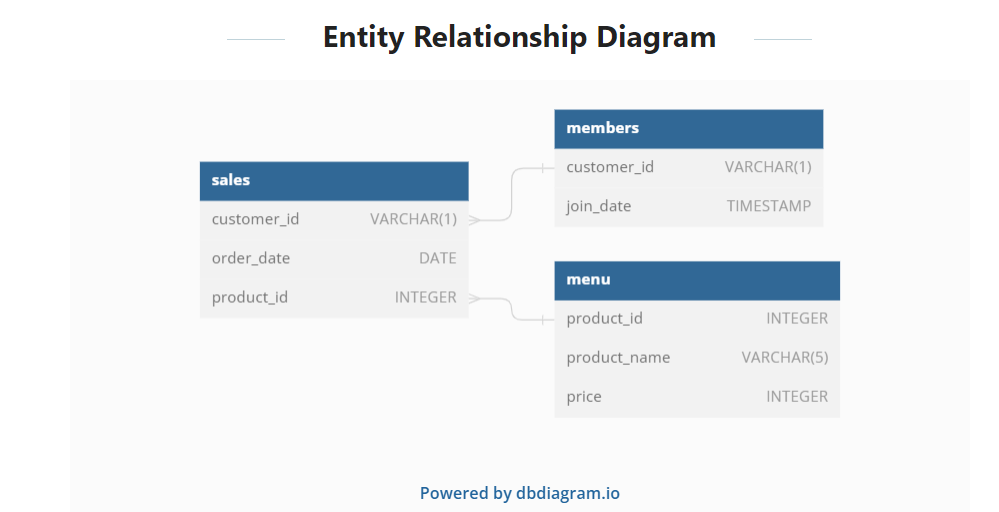
8 Week SQL Challenge: CaseStudy 1 Danny’s Diner

Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they’ve spent and also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers.

He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

Danny has shared with you 3 key datasets for this case study:



**CREATE** **SCHEMA** dannys\_diner;

**SET** search\_path = dannys\_diner;

**CREATE** **TABLE** sales (

"customerid" VARCHAR(1),

"order\_date" DATE,

"product\_id" INTEGER);

**INSERT** **INTO** sales

("customerid", "order\_date", "product\_id")

**VALUES**

('A', '2021-01-01', '1'),

('A', '2021-01-01', '2'),

('A', '2021-01-07', '2'),

('A', '2021-01-10', '3'),

('A', '2021-01-11', '3'),

('A', '2021-01-11', '3'),

('B', '2021-01-01', '2'),

('B', '2021-01-02', '2'),

('B', '2021-01-04', '1'),

('B', '2021-01-11', '1'),

('B', '2021-01-16', '3'),

('B', '2021-02-01', '3'),

('C', '2021-01-01', '3'),

('C', '2021-01-01', '3'),

('C', '2021-01-07', '3');

**CREATE** **TABLE** menu (

"product\_id" INTEGER,

"product\_name" VARCHAR(5),

"price" INTEGER

);

**INSERT** **INTO** menu

("product\_id", "product\_name", "price")

**VALUES**

('1', 'sushi', '10'),

('2', 'curry', '15'),

('3', 'ramen', '12');

**CREATE** **TABLE** members (

"customerid" VARCHAR(1),

"join\_date" DATE

);

**INSERT** **INTO** members

("customer\_id", "join\_date")

**VALUES**

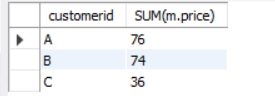
('A', '2021-01-07'),

('B', '2021-01-09');

--*Q1 What is the total amount each customer spent at the restaurant?*

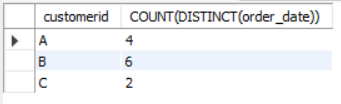
SELECT customerid, SUM(m.price) FROM sales as s

INNER JOIN menu as m ON s.product\_id = m.product\_id GROUP BY customerid;



--*Q2 How many days has each customer visited the restaurant?*

SELECT customerid, COUNT(DISTINCT(order\_date)) FROM sales GROUP BY customerid;



--Q3 *What was the first item from the menu purchased by each customer?*

WITH cte AS (

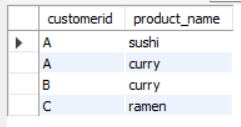
SELECT customerid, order\_date, SALES.product\_id, product\_name,

DENSE\_RANK() over (PARTITION BY customerid ORDER BY order\_date) AS DENSERANK

FROM sales INNER JOIN menu ON sales.product\_id = menu.product\_id)

SELECT customerid, product\_name FROM cte WHERE denserank = 1

GROUP BY customerid, product\_name;



--Q4 *What is the most purchased item on the menu and how many times was it purchased by all customers?*

SELECT product\_name, COUNT(sales.product\_id) AS no\_of\_times\_ordered

FROM sales JOIN menu

ON sales.product\_id = menu.product\_id

GROUP BY menu.product\_name ORDER BY no\_of\_times\_ordered DESC LIMIT 1;



--Q5 *Which item was the most popular for each customer?*

WITH food\_popularity AS (

SELECT sales.customerid, menu.product\_name,

COUNT(SALES.product\_id) AS ordcount,

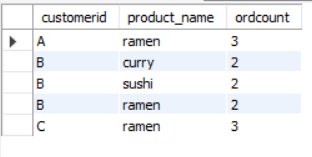
DENSE\_RANK() over (PARTITION BY sales.customerid ORDER BY COUNT(sales.product\_id) DESC) AS popularity

FROM sales JOIN menu ON sales.product\_id = menu.product\_id

GROUP BY sales.customerid, menu.product\_name)

SELECT customerid, product\_name, ordcount FROM food\_popularity

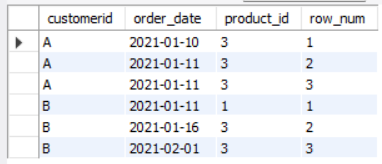
WHERE POPULARITY = 1;



--Q6 *Which item was purchased first by the customer after they became a member?*

select members.customerid, order\_date, product\_id,row\_number() over (partition by members.customerid order by order\_date) AS row\_num

from sales JOIN members ON sales.customerid = members.customerid WHERE order\_date>join\_date



--Q7 *Which item was purchased just before the customer became a member?*

WITH cte AS (

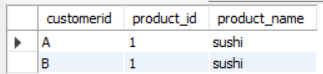
SELECT members.customerid, order\_date, product\_id,ROW\_NUMBER() OVER (PARTITION BY members.customerid ORDER BY order\_date DESC) AS row\_num

FROM sales JOIN members ON sales.customerid = members.customerid

WHERE order\_date<join\_date)

SELECT cte.customerid, cte.product\_id, product\_name FROM cte JOIN menu

ON cte.product\_id = menu.product\_id WHERE row\_num = 1 ORDER BY customerid;



--Q8 *What is the total items and amount spent for each member before they became a member?*

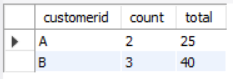
SELECT sales.customerid, COUNT(sales.product\_id) AS count ,SUM(menu.price) AS total

FROM sales JOIN members ON sales.customerid = members.customerid AND join\_date>order\_date

JOIN menu ON sales.product\_id = menu.product\_id

GROUP BY sales.customerid

ORDER BY sales.customerid



--Q9 *If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?*

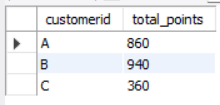
WITH points\_cte AS (

SELECT menu.product\_id, CASE WHEN product\_id = 1 THEN price \* 20 ELSE price \* 10 END AS points FROM dannys\_diner.menu)

SELECT sales.customerid, SUM(points\_cte.points) AS total\_points

FROM dannys\_diner.sales JOIN points\_cte ON sales.product\_id = points\_cte.product\_id

GROUP BY sales.customerid ORDER BY sales.customerid;



--Q10 *In the first week after a customer joins the program (including their join date) they earn 2x points on all items, not just sushi - how many points do customer A and B have at the end of January?*

SELECT s.customerid,

SUM(IF(order\_date BETWEEN join\_date AND DATE\_ADD(join\_date, INTERVAL 6 DAY), price\*10\*2, IF(product\_name = 'sushi', price\*10\*2, price\*10))) AS customer\_points

FROM dannys\_diner.menu AS m INNER JOIN dannys\_diner.sales AS s ON m.product\_id = s.product\_id

INNER JOIN dannys\_diner.members AS mem USING (customerid)

WHERE order\_date <='2021-01-31' AND order\_date >=join\_date

GROUP BY s.customerid ORDER BY s.customerid;

