




```
1 • create database danny_diner;  
2 • show databases;  
3 • use danny_diner;  
4  
5 • - create table sales(  
6     customer_id varchar(10),  
7     order_date date,  
8     product_id integer);  
9
```

```
10 • INSERT INTO sales
11     (customer_id, order_date, product_id)
12 VALUES
13     ('A', '2021-01-01', '1'),
14     ('A', '2021-01-01', '2'),
15     ('A', '2021-01-07', '2'),
16     ('A', '2021-01-10', '3'),
17     ('A', '2021-01-11', '3'),
18     ('A', '2021-01-11', '3'),
19     ('B', '2021-01-01', '2'),
20     ('B', '2021-01-02', '2'),
21     ('B', '2021-01-04', '1'),
22     ('B', '2021-01-11', '1'),
23     ('B', '2021-01-16', '3'),
24     ('B', '2021-02-01', '3'),
25     ('C', '2021-01-01', '3'),
26     ('C', '2021-01-01', '3'),
27     ('C', '2021-01-07', '3');
```

```
--
29 ●      select * from sales;
30
31
32 ● ○      CREATE TABLE menu (
33           product_id INTEGER,
34           product_name VARCHAR(5),
35           price INTEGER
36       );
37
38 ●      INSERT INTO menu
39           (product_id, product_name, price)
40       VALUES
41           ('1', 'sushi', '10'),
42           ('2', 'curry', '15'),
43           ('3', 'ramen', '12');
44
```

-  `CREATE TABLE members (  
    customer_id VARCHAR(1),  
    join_date DATE  
);`

- `INSERT INTO members  
    (customer_id, join_date)  
VALUES  
    ('A', '2021-01-07'),  
    ('B', '2021-01-09');`

```
--  
58 -- 1. What is the total amount each customer spent at the restaurant?  
59  
60 • select customer_id,  
61    sum(m.price) as total_amount  
62    from sales s  
63    join menu m  
64    on  
65    s.product_id = m.product_id  
66    group by s.customer_id;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	total_amount
▶	A	76
	B	74
	C	36

```
68
69 -- 2. How many days has each customer visited the restaurant?
70
71 • select customer_id,
72    count(order_date) as count
73 from sales
74 group by customer_id;
75
```

**Result Grid**



**Filter Rows:**

	customer_id	count
▶	A	6
	B	6
	C	3



```
77      -- 3. What was the first item from the menu purchased by each customer?
78
79 •   select m.product_name,
80      s.product_id,
81      s.customer_id
82   from menu m
83  join sales s on
84 m.product_id= s.product_id
85  group by s.product_id,
86      s.customer_id
87  order by order_date;
88
```

```
76
77 -- 3. What was the first item from the menu purchased by each customer?
78
79 • SELECT customer_id,
80         product_name
81 FROM
82     (SELECT customer_id,
83            order_date,
84            product_name,
85            RANK() OVER(PARTITION BY s.customer_id
86                       ORDER BY s.order_date) AS item_rank
87     FROM sales AS s
88     JOIN menu AS m ON s.product_id = m.product_id) AS first_item
89 WHERE item_rank=1
90 GROUP BY customer_id,
91         product_name;
92
```

**Result Grid**



Filter Rows:

	customer_id	product_name
▶	A	sushi
	A	curry
	B	curry
	C	ramen

```
94      -- 4. What is the most purchased item on the menu and how many times was it purchased by all customers?
95
96 •  SELECT
97      menu.product_name,
98      COUNT(sales.product_id) AS most_purchased_item
99  FROM menu
100 INNER JOIN sales
101      ON sales.product_id = menu.product_id
102 GROUP BY menu.product_name
103 ORDER BY most_purchased_item DESC
104 LIMIT 1;
```

Result Grid



Filter Rows:

Export:



	product_name	most_purchased_item
▶	ramen	8

```
106      -- 5. Which item was the most popular for each customer?
107
108 • WITH order_info AS
109     (SELECT product_name,
110            customer_id,
111            count(product_name) AS order_count,
112            rank() over(PARTITION BY customer_id
113                        ORDER BY count(product_name) DESC) AS rank_num
114     FROM menu
115     INNER JOIN sales ON menu.product_id = sales.product_id
116     GROUP BY customer_id,
117            product_name)
118 SELECT customer_id,
119        product_name,
120        order_count
121 FROM order_info
122 WHERE rank_num = 1;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	product_name	order_count
▶	A	ramen	3
	B	curry	2
	B	sushi	2
	B	ramen	2
	C	ramen	3

-- 6. Which item was purchased first by the customer after they became a member?

WITH diner\_info AS

(SELECT product\_name,  
s.customer\_id,  
order\_date,  
join\_date,  
m.product\_id,  
RANK() OVER(PARTITION BY s.customer\_id  
ORDER BY s.order\_date) AS first\_item

FROM menu AS m

INNER JOIN sales AS s ON m.product\_id = s.product\_id

INNER JOIN members AS mem ON mem.customer\_id = s.customer\_id

WHERE order\_date >= join\_date )

SELECT customer\_id,  
product\_name,  
order\_date

FROM diner\_info

WHERE first\_item=1;



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	product_name	order_date
▶	A	curry	2021-01-07
	B	sushi	2021-01-11

```
146 -- 8. What is the total items and amount spent for each member before they became a member?
147
148 • SELECT s.customer_id,
149         count(product_name) as total_item,
150         sum(price) as total_amount
151 FROM menu AS m
152 INNER JOIN sales AS s ON m.product_id = s.product_id
153 INNER JOIN members AS mem ON mem.customer_id = s.customer_id
154 WHERE order_date < join_date
155 group by s.customer_id
156 order by customer_id;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	total_item	total_amount
▶	A	2	25
	B	3	40

```

158 -- 9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
159 • SELECT customer_id,
160      SUM(CASE
161          WHEN product_name = 'sushi' THEN price*20
162          ELSE price*10
163      END) AS customer_points
164 FROM menu AS m
165 INNER JOIN sales AS s ON m.product_id = s.product_id
166 GROUP BY customer_id
167 ORDER BY customer_id;
168
169 -- Total points that each customer has accrued after taking a membership
170 • SELECT s.customer_id,
171      SUM(CASE
172          WHEN product_name = 'sushi' THEN price*20
173          ELSE price*10
174      END) AS customer_points
175 FROM menu AS m
176 INNER JOIN sales AS s ON m.product_id = s.product_id
177 INNER JOIN members AS mem ON mem.customer_id = s.customer_id
178 WHERE order_date >= join_date
179 GROUP BY s.customer_id
180 ORDER BY s.customer_id;

```

Result Grid



Filter Rows:

Export:



V

customer\_id

customer\_points



A

860

B

940

C

360



Result Grid



Filter Rows:

Export:



Wrap Cell

	customer_id	customer_points
▶	A	510
	B	440

```

181
182 -- 10. In the first week after a customer joins the program (including their join date) they earn 2x points on all items, not just sushi -
183 -- how many points do customer A and B have at the end of January?
184
185 • WITH program_last_day_cte AS
186     (SELECT join_date,
187          DATE_ADD(join_date, INTERVAL 7 DAY) AS program_last_date,
188          customer_id
189     FROM members)
190 SELECT s.customer_id,
191        SUM(CASE
192            WHEN order_date BETWEEN join_date AND program_last_date THEN price*10*2
193            WHEN order_date NOT BETWEEN join_date AND program_last_date
194              AND product_name = 'sushi' THEN price*10*2
195            WHEN order_date NOT BETWEEN join_date AND program_last_date
196              AND product_name != 'sushi' THEN price*10
197            END) AS customer_points
198 FROM menu AS m
199 INNER JOIN sales AS s ON m.product_id = s.product_id
200 INNER JOIN program_last_day_cte AS mem ON mem.customer_id = s.customer_id
201 AND order_date <='2021-01-31'
202 AND order_date >=join_date
203 GROUP BY s.customer_id
204 ORDER BY s.customer_id;

```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	customer_points
▶	A	1020
	B	440