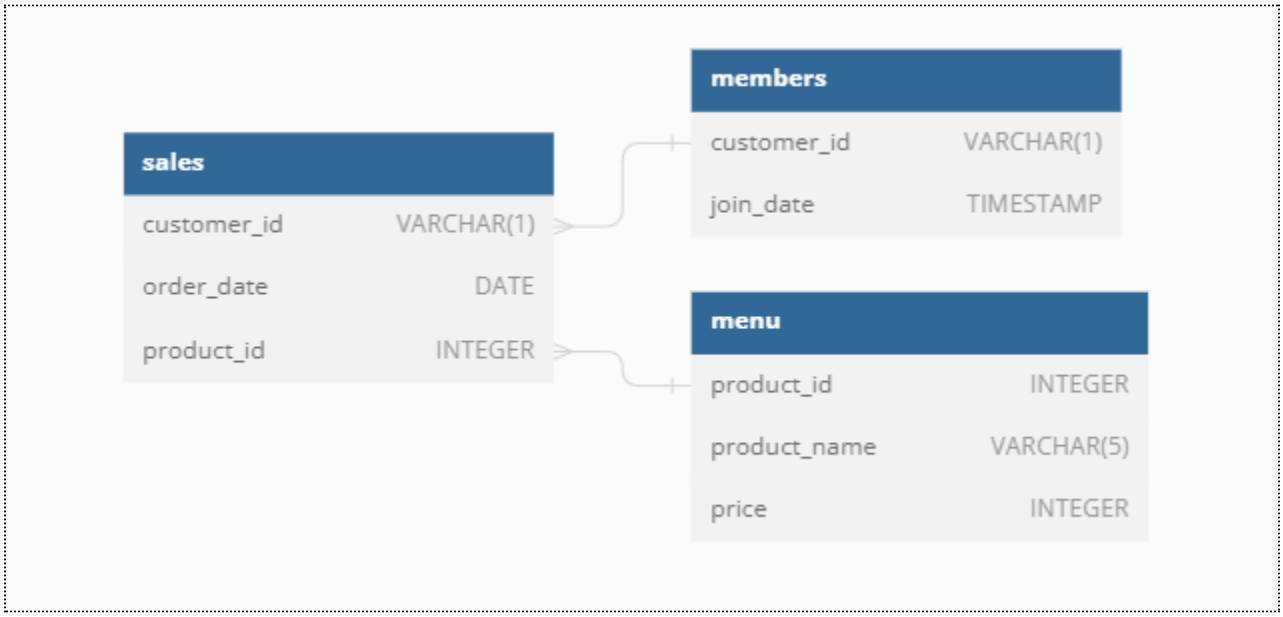
Case Study #1 - Danny's Diner

User 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 favorite.

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

- sales
- menu
- members



Case Study Questions

```
select * from dbo.members;
select * from dbo.menu;
select * from dbo.sales;


-- 1. What is the total amount each customer spent at the restaurant?

-- First find which products was brought how many times by each customer
-- calculate the cost of product from Menu table
-- Finally Aggregate cost for each customer

select customer_id, sum(cost) 'Total Spent' from
(select x.*, (tyms*price) as cost from
(select customer_id, product_id, count(*) as tyms from dbo.sales group by customer_id, product_id)x
join dbo.menu m on x.product_id=m.product_id)y group by customer_id
```

	customer_id	Total Spent
1	A	76
2	B	74
3	C	36

-- 2. How many days has each customer visited the restaurant?

-- Count unique dates each customers has visited

```
select customer_id, count(distinct order_date) as Visits from dbo.sales group by customer_id
```

	customer_id	Visits
1	A	4
2	B	6
3	C	2

-- 3. What was the first item from the menu purchased by each customer?

-- First filter the sales table by the first date each customer visited
-- Show the product name by joining the Menu table

```
select customer_id, order_date, product_name from
(select * from dbo.sales where order_date in
(select min(order_date) from dbo.sales group by customer_id))x join dbo.menu m on
x.product_id=m.product_id;
```

	customer_id	order_date	product_name
1	A	2021-01-01	sushi
2	A	2021-01-01	curry
3	B	2021-01-01	curry
4	C	2021-01-01	ramen
5	C	2021-01-01	ramen

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

-- Group by product_id and find how many times each product has been ordered
-- Join Menu table to get the product name
-- Order the data by times ordered in descending and show the top most record

```
select top 1 product_name, Tyms from
(select product_id, count(*) as Tyms from dbo.sales group by product_id)x join dbo.menu m on
x.product_id=m.product_id
order by x.Tyms desc
```

	product_name	Tyms
1	ramen	8

-- 5. Which item was the most popular for each customer?

-- Means how many times each customer has brought each product
-- First count the number of times each customer has brought each product

-- Perform a rank function to rank the products purchased most by each customer
-- Filter based on rank 1 and join Menu table to get the product name

```
select customer_id, product_name, tyms from
(select *, rank() over(partition by customer_id order by tyms desc) as rnk from
(select customer_id, product_id, count(*) tyms from dbo.sales group by customer_id, product_id)x)y join
dbo.menu m on y.product_id=m.product_id
where y.rnk=1
```

	customer_id	product_name	tyms
1	A	ramen	3
2	B	sushi	2
3	B	curry	2
4	B	ramen	2
5	C	ramen	3

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

```
select customer_id, order_date, product_name from
(select *, ROW_NUMBER() over(partition by customer_id order by order_date) as rw from
(select s.customer_id, order_date, product_id from dbo.sales s join
dbo.members mb on s.customer_id=mb.customer_id and s.order_date>=mb.join_date)x)y join dbo.menu m on
y.product_id=m.product_id
where rw=1
```

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

-- 7. Which item was purchased just before the customer became a member?

```
select customer_id, order_date, product_name from
(select *, Rank() over(partition by customer_id order by order_date desc) as rw from
(select s.customer_id, order_date, product_id from dbo.sales s join
dbo.members mb on s.customer_id=mb.customer_id and s.order_date<mb.join_date)x)y join dbo.menu m on
y.product_id=m.product_id
where rw=1
```

	customer_id	order_date	product_name
1	A	2021-01-01	sushi
2	A	2021-01-01	curry
3	B	2021-01-04	sushi

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

```
select customer_id, count(Brought) as 'Total Items', sum(brought*price) as 'Amt Spent' from
(select customer_id, product_id, count(*) Brought from (
select s.customer_id, order_date, product_id from dbo.sales s join dbo.members m on
s.customer_id=m.customer_id and s.order_date<m.join_date)x
group by customer_id, product_id)y join dbo.menu mn on y.product_id=mn.product_id
group by customer_id
```

	customer_id	Total Items	Amt Spent
1	A	2	25
2	B	2	40

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

```
select customer_id, sum(points) as Tot_points from
(select customer_id, s.product_id, price,
case when s.product_id=1 then price*20 else price*10 end as points
from dbo.sales s join dbo.menu m on m.product_id=s.product_id)x
group by customer_id;
```

	customer_id	Tot_points
1	A	860
2	B	940
3	C	360

-- 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. How many points do customer A and B have at the end of January?

```
select customer_id, sum(points) as tot_points from
(select s.*, price, price*20 as points
from dbo.sales s  join dbo.members mb on  mb.customer_id=s.customer_id and mb.join_date<=s.order_date
and DATEADD(day,7,mb.join_date)>=s.order_date join dbo.menu m on m.product_id=s.product_id)x group by
customer_id;
```

	customer_id	tot_points
1	A	1020
2	B	440

BONUS Ques Below.

Recreate the following table output using the available data:

customer_id	order_date	product_name	price	member
A	2021-01-01	curry	15	N
A	2021-01-01	sushi	10	N
A	2021-01-07	curry	15	Y
A	2021-01-10	ramen	12	Y
A	2021-01-11	ramen	12	Y
A	2021-01-11	ramen	12	Y
B	2021-01-01	curry	15	N
B	2021-01-02	curry	15	N
B	2021-01-04	sushi	10	N
B	2021-01-11	sushi	10	Y
B	2021-01-16	ramen	12	Y
B	2021-02-01	ramen	12	Y
C	2021-01-01	ramen	12	N
C	2021-01-01	ramen	12	N
C	2021-01-07	ramen	12	N

```
select customer_id, order_date, product_name, price,
(case when (order_date>=join_date) then 'Y' when order_date is null then 'N' else 'N' end) as member from
(select s.customer_id, s.order_date, m.product_name, m.price, join_date from dbo.sales s join dbo.menu m on
s.product_id=m.product_id
left join dbo.members mb on s.customer_id=mb.customer_id)x
```

Bonus Question Below.

-- User also requires further information about the ranking of customer products, but he purposely does not need the ranking for non-member purchases so he expects null ranking values for the records when customers are not yet part of the loyalty program.

customer_id	order_date	product_name	price	member	ranking
A	2021-01-01	curry	15	N	null
A	2021-01-01	sushi	10	N	null
A	2021-01-07	curry	15	Y	1
A	2021-01-10	ramen	12	Y	2
A	2021-01-11	ramen	12	Y	3
A	2021-01-11	ramen	12	Y	3
B	2021-01-01	curry	15	N	null
B	2021-01-02	curry	15	N	null
B	2021-01-04	sushi	10	N	null
B	2021-01-11	sushi	10	Y	1
B	2021-01-16	ramen	12	Y	2
B	2021-02-01	ramen	12	Y	3
C	2021-01-01	ramen	12	N	null
C	2021-01-01	ramen	12	N	null
C	2021-01-07	ramen	12	N	null

```
select *, (case when member like 'N%' then null else (DENSE_RANK() over(partition by customer_id, member order by order_date)) end)as ranking from
(select customer_id, order_date, product_name, price,
(case when order_date>=join_date then 'Y' else 'N' end) as member from
(select s.customer_id, order_date, product_name, price, join_date from dbo.sales s join dbo.menu m on s.product_id=m.product_id
left join dbo.members mb on s.customer_id=mb.customer_id)x)y
```
