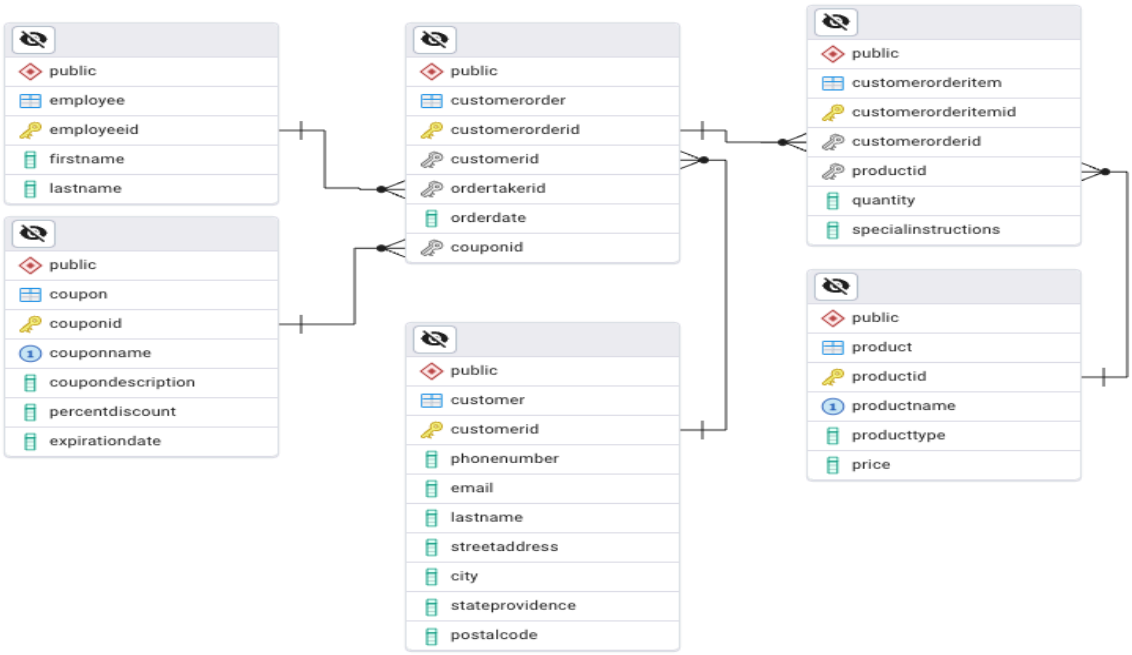
SQL Sample

DB Schema for a Pizza Store. The database contains 92 orders with 282 order items placed between March 1st, 2022 and March 15th, 2022



-- Get the order detail for order placed by customer on 4th March, 2022 by customer 'Duffy'
-- Sample of multi table join query

```
SELECT c.lastname, co.orderdate, coi.quantity, p.productname, (coi.quantity * p.price) AS
prod_price,
round((coi.quantity * p.price) - ((coi.quantity * p.price) * cp.percentdiscount/100),2) AS
discounted_price,  cp.coupondescription
FROM customer c
LEFT JOIN customerorder co ON c.customerid = co.customerid
LEFT JOIN customerorderitem coi ON co.customerorderid = coi.customerorderid
LEFT JOIN product p ON coi.productid = p.productid
LEFT JOIN coupon cp ON cp.couponid = co.couponid
WHERE c.lastname = 'Duffy' AND (co.orderdate) = TO_DATE('2022-03-04', 'YYYY-MM-DD');
```

Output:

	lastname character varying (40)	orderdate date	quantity integer	productname character varying (20)	prod_price numeric	discounted_price numeric	coupondescription character varying (80)
1	Duffy	2022-03-04	1	Delivery	5.00	4.50	10% off order
2	Duffy	2022-03-04	1	Extra Pepperoni	1.50	1.35	10% off order
3	Duffy	2022-03-04	1	2 Liter Diet Coke	4.50	4.05	10% off order
4	Duffy	2022-03-04	3	Medium Pizza	35.97	32.37	10% off order

-- List total quantity of pizzas ordered by customers from most to least quantity
-- Sample of GroupBy, Having and OrderBy query

```
SELECT p.productname, SUM(coi.quantity) AS total_qty_ordered
FROM product p
LEFT JOIN  customerorderitem coi ON coi.productid = p.productid
GROUP BY 1
HAVING p.productname LIKE '%Pizza'
ORDER BY 2 DESC;
```

Output:

	productname character varying (20)	total_qty_ordered bigint
1	Large Pizza	45
2	Pan Pizza	33
3	Small Pizza	22
4	Medium Pizza	20
5	Smelt Pizza	11

-- Running total of revenue for first 15 days of March'2022 using window function
-- Sample of Window Function Over a period

```
SELECT DISTINCT
   co.orderdate ,
   SUM(round((coi.quantity * p.price) - ((coi.quantity * p.price) * cp.percentdiscount/100),2))
         OVER (ORDER BY orderdate) AS running_total
FROM
   customerorder co
LEFT JOIN customerorderitem coi ON co.customerorderid = coi.customerorderid
LEFT JOIN product p ON coi.productid = p.productid
LEFT JOIN coupon cp ON cp.couponid = co.couponid
ORDER BY orderdate;
```

Output:

	orderdate date	running_total numeric
1	2022-03-01	73.07
2	2022-03-03	128.25
3	2022-03-04	283.95
4	2022-03-05	345.61
5	2022-03-10	382.40
6	2022-03-13	558.15
7	2022-03-14	591.70
8	2022-03-15	672.91