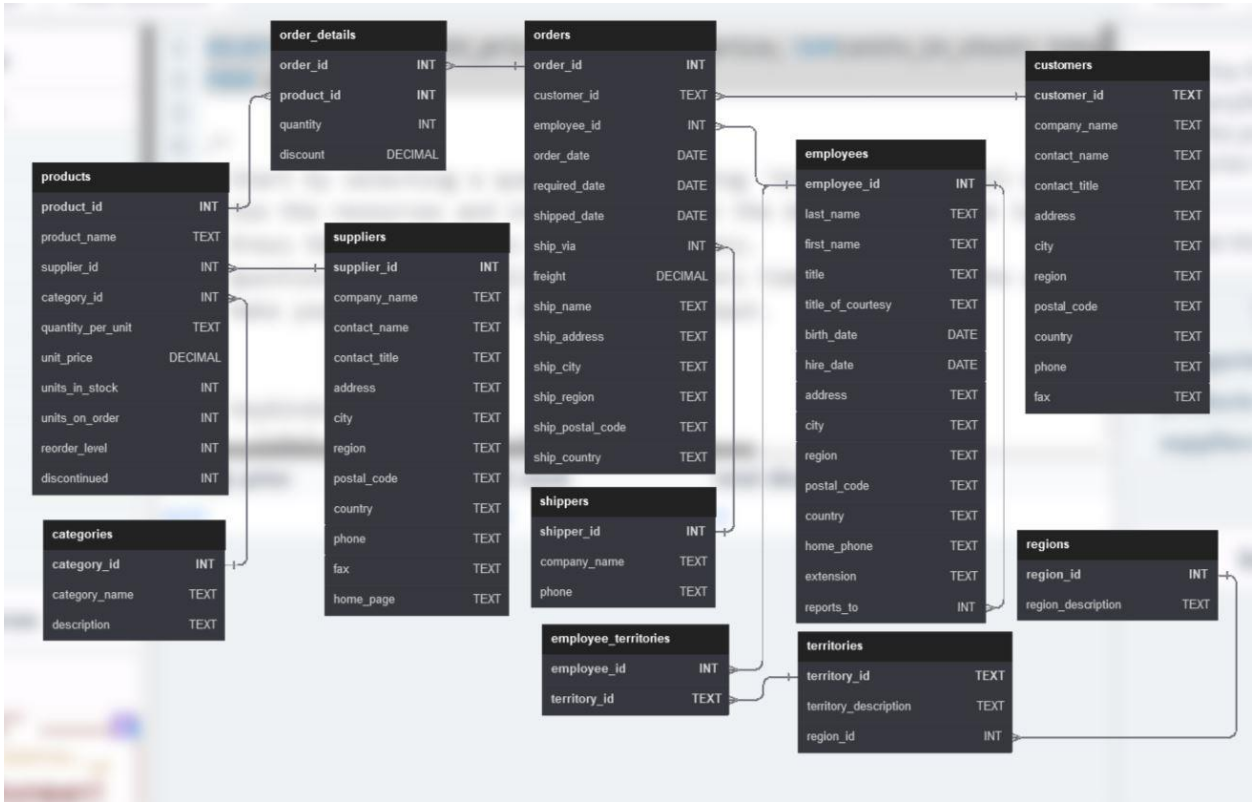

Question #1 (Level: easy)
Show the category_name and description from the categories table sorted by category_name.

```
SELECT category_name, description
FROM categories
ORDER BY category_name;
```

Question #2
Show all the contact_name, address, city of all customers which are not from 'Germany', 'Mexico', 'Spain'.

```
SELECT contact_name, address, city
FROM customers
WHERE country NOT IN ('Germany', 'Mexico', 'Spain');
```

Question #3
Show order_date, shipped_date, customer_id, Freight of all orders placed on 2018 Feb 26.

```
SELECT order_date, shipped_date, customer_id, freight
FROM orders
WHERE order_date='2018-02-26';
```

Question #4

Show the employee_id, order_id, customer_id, required_date, shipped_date from all orders shipped later than the required date.

```
SELECT employee_id, order_id, customer_id, required_date, shipped_date
FROM orders
WHERE shipped_date>required_date;
```

Question #5
Show all the even numbered Order_id from the orders table.

```
SELECT order_id
FROM orders
WHERE order_id%2=0;
```

Question #6
Show the city, company_name, contact_name of all customers from cities which contains the letter 'L' in the city name, sorted by contact_name.

```
SELECT city, company_name, contact_name
FROM customers
WHERE city LIKE '%L%'
ORDER BY contact_name;
```

Question #7
Show the company_name, contact_name, fax number of all customers that has a fax number. (not null).

```
SELECT company_name, contact_name, fax
FROM customers
WHERE fax IS NOT null;
```

Question #8
Show the first_name, last_name. hire_date of the most recently hired employee.

```
SELECT first_name, last_name, hire_date
FROM employees
ORDER BY hire_date DESC
LIMIT 1;
```

Question #9
Show the average unit price rounded to 2 decimal places, the total units in stock, total discontinued products from the products table.

```
SELECT ROUND(AVG(unit_price),2) average_price, SUM(units_in_stock) total_stock,
SUM(discontinued) total_discontinued
FROM products;
```

Question #10 (Level: medium)
Show the ProductName, CompanyName, CategoryName from the products, suppliers, and categories table.

```
SELECT product_name, company_name, category_name
FROM products p
JOIN suppliers s
USING(supplier_id)
JOIN categories c
USING(category_id);
```

Question #11
Show the category_name and the average product unit price for each category rounded to 2 decimal places.

```
SELECT category_name, ROUND(AVG(unit_price),2) unit_price
FROM products p
JOIN categories c
ON p.category_id=c.category_id
GROUP BY p.category_id;
```

Question #12
Show the city, company_name, contact_name from the customers and suppliers table merged together.
Create a column which contains 'customers' or 'suppliers' depending on the table it came from.

```
SELECT city, company_name, contact_name, 'customers' relationship
FROM customers
UNION
SELECT city, company_name, contact_name, 'suppliers' relationship
FROM suppliers;
```

Question #13 (Level: hard)
Show the employee's first_name and last_name, a "num_orders" column with a count of the orders taken, and a column called "Shipped" that displays "On Time" if the order shipped_date is less or equal to the required_date, "Late" if the order shipped late.
Order by employee last_name, then by first_name, and then descending by number of orders.

```
SELECT
e.first_name first_name,
```

```
e.last_name last_name,
COUNT(o.order_id) num_orders,
(CASE
        WHEN o.shipped_date<=o.required_date THEN 'On Time'
        ELSE 'Late'
END) AS shipped
FROM employees e
JOIN orders o
USING(employee_id)
GROUP BY e.first_name, e.last_name, shipped
ORDER BY e.last_name, e.first_name, num_orders DESC;
```

Question #14

Show how much money the company lost due to giving discounts each year, order the years from most recent to least recent. Round to 2 decimal places.

```
SELECT YEAR(o.order_date) order_year,
ROUND(SUM(p.unit_price*od.quantity*od.discount), 2) total_discount
FROM products p
JOIN order_details od
USING(product_id)
JOIN orders o
USING(order_id)
GROUP BY order_year
ORDER BY 1 DESC;
```