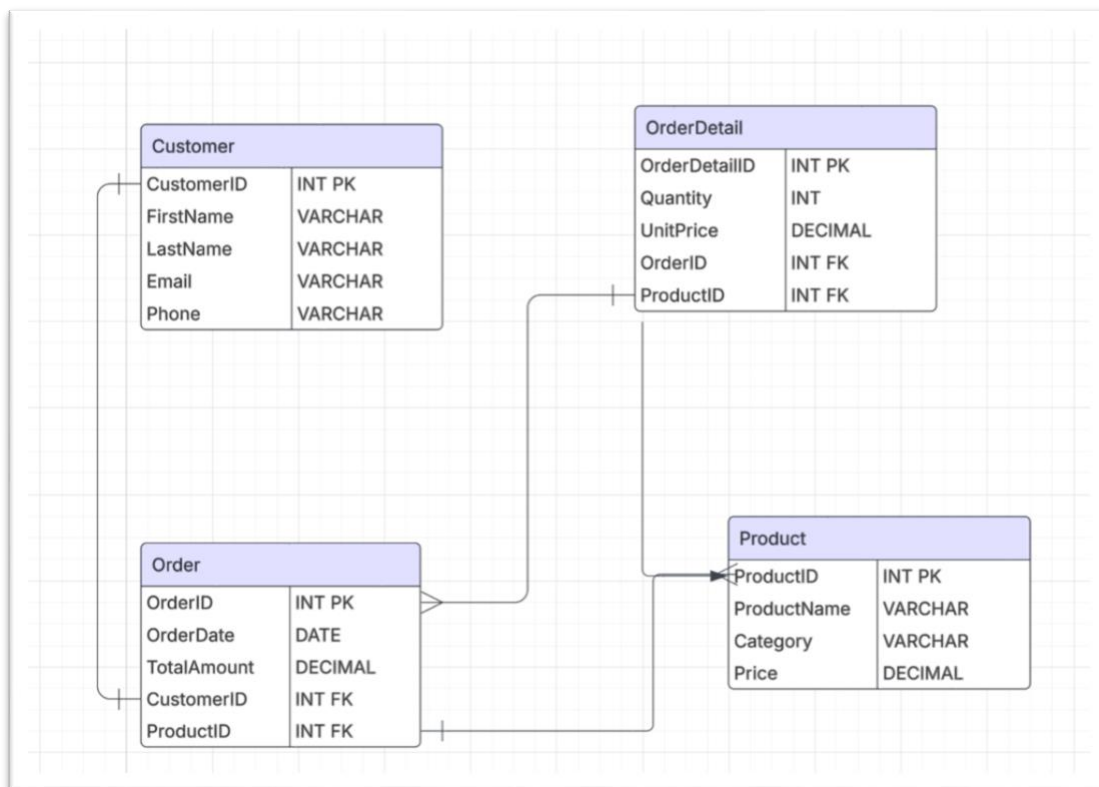DESIGN: Online Technology Marketplace
Entities: Customer, Product, Order, OrderDetail

ERD (Visio):

Private Keys: CustomerID (Customer), OrderID (Order), OrderDetailID (OrderDetail), ProductID (Product and One to many)

Order FK: CustomerID (easily able to track which customer made the order) ProductID (easy to see which product the customer ordered)

OrderDetail FK: OrderID(able to see each order's quantity and unit price) I also accidentally linked it on the ERD model to product (ignore that I meant to link it to the orderID column this mistake is only on the ERD model and not on the SQL table so don't take off points because it was a small mistake and I already screenshotted it) and product ID FK to make it so you can easily check the product information this isnt nessecary but it makes it slightly easier.

Customer	
CustomerID	INT PK
FirstName	VARCHAR
LastName	VARCHAR
Email	VARCHAR
Phone	VARCHAR

OrderDetail	
OrderDetailID	INT PK
Quantity	INT
UnitPrice	DECIMAL
OrderID	INT FK
ProductID	INT FK

Order	
OrderID	INT PK
OrderDate	DATE
TotalAmount	DECIMAL
CustomerID	INT FK
ProductID	INT FK

Product	
ProductID	INT PK
ProductName	VARCHAR
Category	VARCHAR
Price	DECIMAL

Here are my SQL tables:

FirstName	LastName	Email	Phone	CustomerID
Sam	Johnson	sjohnson@outlook.com	904-000-002	2
Greg	Williams	gwilliams@outlook.com	904-000-003	3
Peter	Griffin	pgriff@outlook.com	904-000-004	4
Lebron	James	lebron@outlook.com	904-000-005	5
NULL	NULL	NULL	NULL	NULL

OrderID	OrderDate	TotalAmount	CustomerID	ProductID
2	2025-01-02	80.00	2	2
3	2025-01-03	40.00	3	3
4	2025-01-04	60.00	4	4
5	2025-01-05	50.00	5	5
NULL	NULL	NULL	NULL	NULL

OrderDetailID	Quantity	UnitPrice	OrderID	ProductID
2	1	80.00	2	2
3	1	40.00	3	3
4	1	60.00	4	4
5	1	50.00	5	5
NULL	NULL	NULL	NULL	NULL

ProductID	ProductName	Category	Price
1	Mouse	Accessories	100.00
2	Keyboard	Accessories	80.00
3	Laptop	Hardware	40.00
4	Headphones	Accessories	60.00
5	Televison	Hardware	50.00
NULL	NULL	NULL	NULL

EACH TABLE HAD 5 entries, HOWEVER THE TASK THAT ASKED ME TO REMOVE ONE OF THE CUSTOMERS BROUGHT IT DOWN TO 4 CUSTOMERS. I DIDN'T KNOW THAT I NEEDED TO SCREENSHOT IT UNTIL AFTER THAT TASK

HERE ARE THE QUERIES: The first one is just deleting the first entry

```
1 •   DELETE FROM OrderDetail WHERE OrderDetailID = 1;
2 •   DELETE FROM first.Order WHERE OrderID = 1;
3 •   DELETE FROM Customer WHERE CustomerID = 1;
4 |
```

```
1
2 •   SELECT *
3     FROM Customer;
4 •   SELECT *
5     FROM first.Order;
6 |
```

1:6

Result Grid | Filter Rows: Q Search | Edit: | Export/Import: | Result Grid

OrderID	OrderDate	TotalAmount	CustomerID	ProductID
2	2025-01-02	80.00	2	2
3	2025-01-03	40.00	3	3
4	2025-01-04	60.00	4	4
5	2025-01-05	50.00	5	5
NULL	NULL	NULL	NULL	NULL

Form Editor

Field Types

Customer 27   Order 28   Apply   Revert

```
1
2 •  SELECT *
3    FROM Product
4    ORDER BY Price Desc;
5
```

21:4

Result Grid | Filter Rows: Search | Edit: | Export/Import:

ProductID	ProductName	Category	Price
1	Mouse	Accessories	100.00
2	Keyboard	Accessories	80.00
4	Headphones	Accessories	60.00
5	Televison	Hardware	50.00
3	Laptop	Hardware	40.00
NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

```
1 •  SELECT CustomerID, OrderDate
2    FROM first.Order
3    ORDER BY OrderDate DESC;
```

25:3

Result Grid | Filter Rows: Search | Export:

CustomerID	OrderDate
5	2025-01-05
4	2025-01-04
3	2025-01-03
2	2025-01-02

Result Grid

Form Editor

Field Types

```
1 •   SELECT CustomerID, COUNT(*) AS OrderCount
2     FROM first.Order
3     GROUP BY CustomerID;
4
```

21:3

Result Grid | Filter Rows: Q Search | Export:

CustomerID	OrderCount
2	1
3	1
4	1
5	1

Result 38

Read Only

```
1 •   SELECT *
2     FROM OrderDetail
3     ORDER BY Quantity Desc;
```

24:3

Result Grid | Filter Rows: Q Search | Edit: | Export/Import:

OrderDetailID	Quantity	UnitPrice	OrderID	ProductID
2	1	80.00	2	2
3	1	40.00	3	3
4	1	60.00	4	4
5	1	50.00	5	5
NULL	NULL	NULL	NULL	NULL

```sql
SELECT c.CustomerID, c.FirstName
FROM Customer c
LEFT JOIN first.Order o ON c.CustomerID = o.CustomerID
WHERE o.OrderID IS NULL;
```

33:1

Result Grid | Filter Rows: Search | Export:

CustomerID	FirstName

Limit to 1000 rows

```sql
SELECT *
FROM first.Order
WHERE TotalAmount > 1000;
```

1:4

Result Grid | Filter Rows: Search | Edit: | Export/Import:

OrderID	OrderDate	TotalAmount	CustomerID	ProductID
NULL	NULL	NULL	NULL	NULL

```sql
1 •  SELECT CustomerID, CategoryCount
2    FROM (
3        SELECT o.CustomerID, COUNT(DISTINCT p.Category) AS CategoryCount
4        FROM first.Order o
5        JOIN Product p ON o.ProductID = p.ProductID
6        GROUP BY o.CustomerID
7    ) AS CustomerCategories
8    WHERE CategoryCount > 1;
9
```

21:4

Result Grid | Filter Rows: Search | Export:

CustomerID	CategoryCount

Result Grid

Form Editor

```sql
1 •  Execute the selected portion of the script or
2    everything, if there is no selection
     CustomerID,
3        AVG(TotalAmount) AS AvgOrderValue,
4        RANK() OVER (ORDER BY AVG(TotalAmount) DESC) AS OrderValueRank
5    FROM first.Order
6    GROUP BY CustomerID;
7
```

67:4

Result Grid | Filter Rows: Search | Export:

CustomerID	AvgOrderValue	OrderValueRank
2	80.000000	1
4	60.000000	2
5	50.000000	3
3	40.000000	4

Result Grid

Form Editor

```sql
SELECT c.CustomerID, c.FirstName, c.LastName
FROM Customer c
LEFT JOIN first.Order o
  ON c.CustomerID = o.CustomerID AND o.OrderDate >= CURDATE() - INTERVAL 3 MONTH
WHERE o.OrderID IS NULL;
```

37:1

Result Grid | Filter Rows: Search | Export:

CustomerID	FirstName	LastName
2	Sam	Johnson
3	Greg	Williams
4	Peter	Griffin
5	Lebron	James

Limit to 1000 rows

```sql
SELECT c.CustomerID, c.Firstname,c.LastName, SUM(o.TotalAmount) AS TotalSpent
FROM Customer c
JOIN first.Order o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID, c.FirstName, c.LastName
ORDER BY TotalSpent DESC
LIMIT 3;
```

45:1

Result Grid | Filter Rows: Search | Export: | Fetch rows:

CustomerID	Firstname	LastName	TotalSpent
2	Sam	Johnson	80.00
4	Peter	Griffin	60.00
5	Lebron	James	50.00

```sql
1  ●  SELECT
2          CustomerID,
3          OrderID,
4          OrderDate,
5          TotalAmount,
6  ⊖      SUM(TotalAmount) OVER (
7              PARTITION BY CustomerID
8              ORDER BY OrderDate
9              ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
10          ) AS CumulativeTotal
11     FROM first.Order
12     ORDER BY CustomerID, OrderDate;
13
```

17:11

Result Grid | Filter Rows: | Search | Export: | | Result Grid / Form Editor

CustomerID	OrderID	OrderDate	TotalAmount	CumulativeTotal
2	2	2025-01-02	80.00	80.00
3	3	2025-01-03	40.00	40.00
4	4	2025-01-04	60.00	60.00
5	5	2025-01-05	50.00	50.00