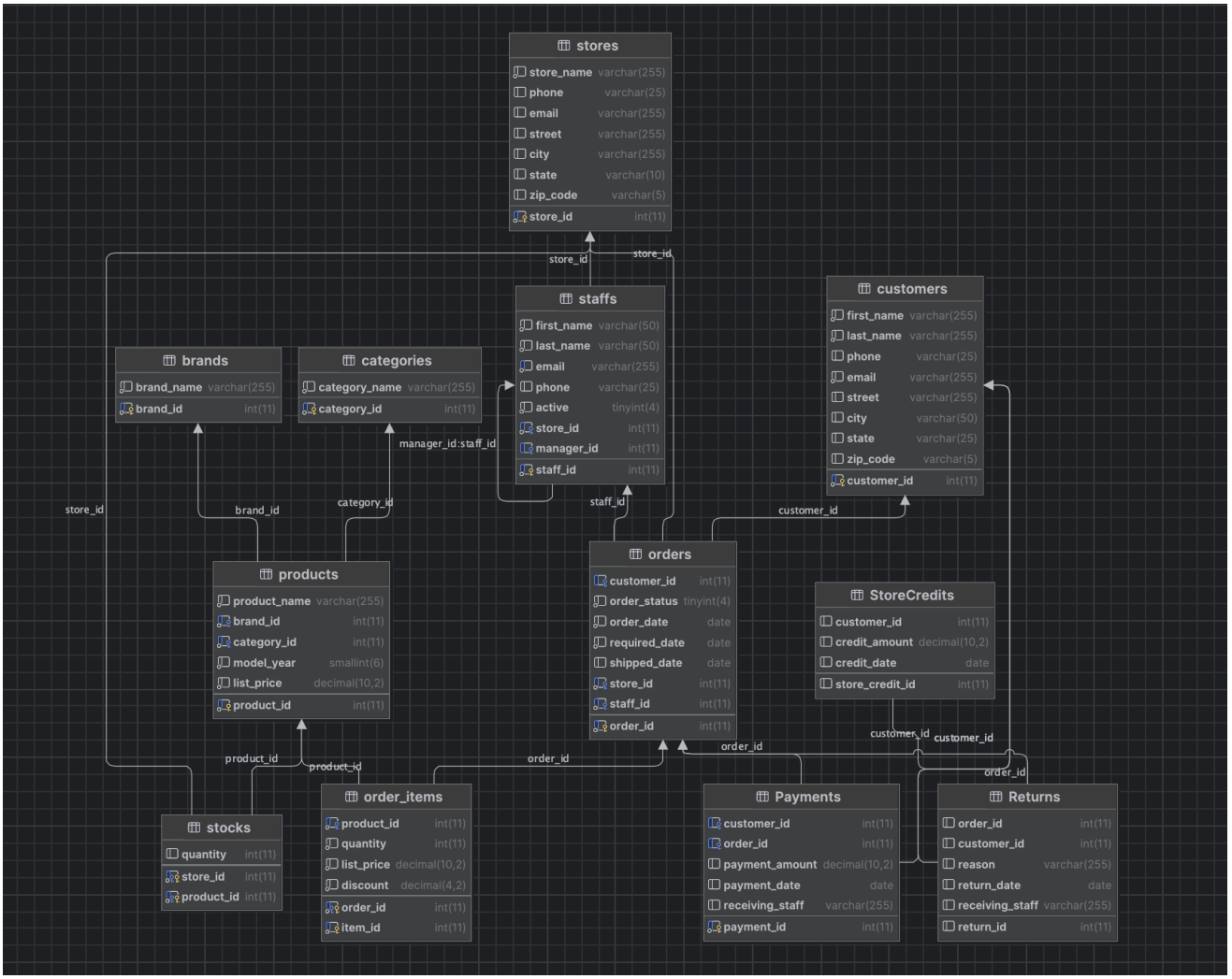
Updated ER Diagram:



SQL Queries:

Creating new Tables:

```
Create Table Payments (
                payment_id INT PRIMARY KEY AUTO_INCREMENT,
                customer_id INT,
                order_id INT,
                payment_amount DECIMAL(10, 2),
                payment_date DATE,
                receiving_staff VARCHAR(255),
                FOREIGN KEY (customer_id) REFERENCES   customers(customer_id),
                Foreign Key (order_id) REFERENCES orders(order_id));
```

```
Create Table Returns (
                    return_id INT Primary Key AUTO_INCREMENT,
                    order_id INT,
                    customer_id INT,
                    reason VARCHAR(255),
                    return_date DATE,
                    receiving_staff VARCHAR(255),
                    product_name VARCHAR(255),
                    quantity_returned INT,
                    FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
                    Foreign Key (order_id) REFERENCES orders(order_id));

CREATE TABLE StoreCredits (
                    store_credit_id INT PRIMARY KEY AUTO_INCREMENT,
                    customer_id INT,
                    credit_amount DECIMAL(10, 2),
                    credit_date DATE,
                    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

Update Query:

```
UPDATE orders SET payment_status = 1 WHERE order_id = ?
```

Insert Queries:

```
INSERT INTO StoreCredits (customer_id, credit_amount) VALUES (?, ?)

INSERT INTO Payments (customer_id, order_id, payment_amount,
payment_date, receiving_staff) VALUES (?, ?, ?, NOW(), ?)

INSERT INTO Returns (order_id, customer_id, return_date, reason,
receiving_staff, product_name, quantity_returned) VALUES (?, ?,
NOW(), ?, ?, ?, ?)
```

Outstandingpayments Query

```
select customers.customer_id,orders.order_id,
concat(customers.first_name,'', customers.last_name) as name,
      sum((order_items.quantity) * (products.list_price * (1 - discount))) as
total_order_value
from orders
```

```
        join customers on orders.customer_id = customers.customer_id
        join order_items on orders.order_id = order_items.order_id
        join products on order_items.product_id = products.product_id
where orders.payment_status!=1
group by name;
```

StoreCredit Query:

```
SELECT
   CONCAT(customers.first_name, ' ', customers.last_name) AS name,
   SUM(sc.credit_amount) AS available_credits
FROM
   latawa.customers
      JOIN
   latawa.StoreCredits sc ON customers.customer_id = sc.customer_id
GROUP BY
   customers.customer_id
HAVING
      available_credits != 0;
```

Return Window query for Avg days

```
SELECT
   CONCAT(customers.first_name, ' ', customers.last_name) AS name,
   AVG(DATEDIFF(r.return_date, o.order_date)) AS avg_days_to_return
FROM
   latawa.customers
      JOIN
   latawa.orders o ON customers.customer_id = o.customer_id
      JOIN
   latawa.Returns r ON o.order_id = r.order_id
GROUP BY
   customers.customer_id
HAVING
      COUNT(r.return_id) > 0;
```