

Relational Database (SQL)

- Customer table

```
CREATE TABLE Customer
(
    customer_id INT PRIMARY KEY,
    name        VARCHAR(100),
    email       VARCHAR(100),
    phone       VARCHAR(20)
);
```

- Item table

```
CREATE TABLE Item
(
    product_number VARCHAR(10) PRIMARY KEY,
    product_name    VARCHAR(100),
    price           DECIMAL(10, 2)
);
```

- Order table

```
CREATE TABLE Order
(
    order_id      INT PRIMARY KEY,
    order_date    DATE,
    customer_id   INT,
    total_price   DECIMAL(10, 2),
    FOREIGN KEY (customer_id) REFERENCES Customer (customer_id)
);
```

- Order_Item table

```
CREATE TABLE OrderItem
(
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id      INT,
    product_number VARCHAR(10),
    quantity      INT,
    FOREIGN KEY (order_id) REFERENCES Order (order_id),
    FOREIGN KEY (product_number) REFERENCES Item (product_number)
);
```

MongoDB (NoSQL)

- Customer Collection

```
{
  "customer_id": 1,
  "name": "Frank Brown",
  "email": "fbrown@gmail.com",
  "phone": "0623156543"
}
```

- Item Collection

```
{
  "product_number": "A546",
  "product_name": "iPhone 12",
  "price": 980.00
},
{
  "product_number": "S333",
  "product_name": "Samsung Galaxy 12S",
  "price": 800.00
}
```

- Order Collection

```
{
  "order_id": 122435,
  "order_date": "2021-11-09",
  "customer_id": 1,
  "total_price": 5160.00,
  "items": [
    {
      "product_number": "A546",
      "quantity": 2,
      "price": 980.00
    },
    {
      "product_number": "S333",
      "quantity": 4,
      "price": 800.00
    }
  ]
}
```

Cassandra (NoSQL Column Store)

- Customer table

```
CREATE TABLE Customer
(
  customer_id UUID PRIMARY KEY,
  name TEXT,
  email TEXT,
  phone TEXT
);
```

- Item table

```
CREATE TABLE Item
(
    product_number TEXT PRIMARY KEY,
    product_name   TEXT,
    price          DECIMAL
);
```

- Order table

```
CREATE TABLE Order
(
    order_id UUID PRIMARY KEY,
    order_date DATE,
    customer_id UUID,
    total_price DECIMAL
);
```

- Order_Item table

```
CREATE TABLE OrderItem
(
    order_item_id UUID PRIMARY KEY,
    order_id UUID,
    product_number TEXT,
    quantity       INT,
    price          DECIMAL
);
```

Neo4j (Graph Database)

```
CREATE (customer:Customer {customer_id: 1, name: 'Frank Brown', email:
'fbrown@gmail.com', phone: '0623156543'})

CREATE (item1:Item {product_number: 'A546', product_name: 'iPhone 12', price:
980.00})
CREATE (item2:Item {product_number: 'S333', product_name: 'Samsung Galaxy
12S', price: 800.00})

CREATE (order:Order {order_id: 122435, order_date: '2021-11-09', total_price:
5160.00})

CREATE (order_item1:OrderItem {quantity: 2, price: 980.00})
CREATE (order_item2:OrderItem {quantity: 4, price: 800.00})

CREATE (customer)-[:PLACED]->(order)
CREATE (order)-[:CONTAINS]->(order_item1)
CREATE (order)-[:CONTAINS]->(order_item2)
CREATE (order_item1)-[:FOR]->(item1)
CREATE (order_item2)-[:FOR]->(item2)
```