

**Clothing Store Database Guidebook**

**A Comprehensive Guide to Managing Retail Data Efficiently**

*By Manveer Kaur*

**Data Analyst, St. Clair College, Toronto**

## Table of Contents

1. Introduction.....	2
2. Database Structure .....	2
3. Table Descriptions.....	3
4. Key SQL Queries .....	5
A Comprehensive Guide to the Design, Creation, and Management of SQL Tables .....	11

**1. Introduction:** The **Clothing Store Database Guidebook** provides a structured overview of the database designed to manage various aspects of a clothing retail business. This database is optimized for handling **customers, products, suppliers, orders, inventory, and categories** efficiently.

The guidebook includes:

- A **detailed explanation** of the database structure
- **Descriptions of all tables** and their relationships
- **SQL scripts** for creating tables, inserting data, and retrieving information
- **Best practices** for database management

This document is useful for **database administrators, developers, and business managers** who need to understand and interact with the system effectively. Whether you're managing inventory, tracking sales, or analyzing customer behavior, this guide will help you navigate and utilize the database efficiently.

## **2. Database Structure**

The database consists of seven primary tables:

- **Customers:** Stores customer details.
- **Categories:** Defines product categories.
- **Suppliers:** Manages supplier information.
- **Products:** Contains details of clothing items.
- **Orders:** Stores customer order data.
- **Order\_Items:** Links orders with products.
- **Inventory:** Tracks stock levels.

### **3. Table Descriptions**

- **Customers Table:**

- customer\_id: Unique ID for each customer.
- first\_name, last\_name: Customer name.
- email: Unique email.
- phone, address: Contact details.
- created\_at: Timestamp of account creation.

- **Categories Table:**

- category\_id: Unique category ID.
- category\_name: Name of category.
- description: Category details.

- **Suppliers Table:**

- supplier\_id: Unique supplier ID.
- supplier\_name: Name of supplier.
- contact\_name, email, phone, address: Contact details.

- **Products Table:**

- product\_id: Unique product ID.
- product\_name: Name of product.
- category\_id: Foreign key linking to Categories.

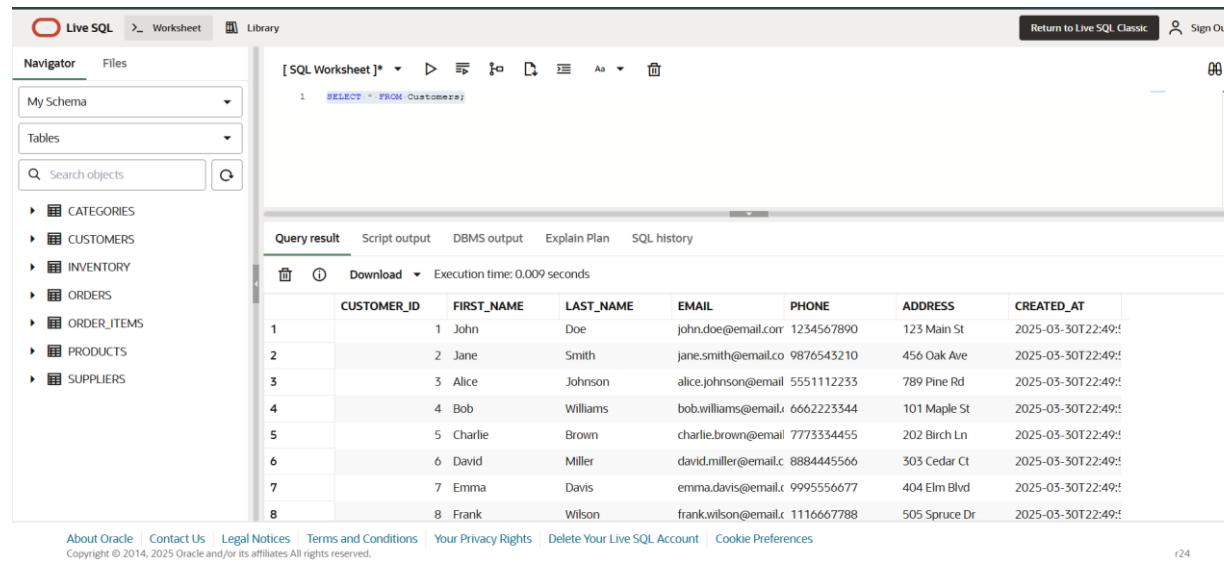
- price: Product price.
  - product\_size, color: Attributes of the product.
  - stock\_quantity: Available stock.
  - supplier\_id: Foreign key linking to Suppliers.
- **Orders Table:**
    - order\_id: Unique order ID.
    - customer\_id: Foreign key linking to Customers.
    - order\_date: Date of order.
    - total\_amount: Total order cost.
    - status: Order status (Pending, Shipped, etc.).
  - **Order\_Items Table:**
    - order\_item\_id: Unique ID.
    - order\_id: Foreign key linking to Orders.
    - product\_id: Foreign key linking to Products.
    - quantity: Number of items ordered.
    - subtotal: Price for the quantity ordered.
  - **Inventory Table:**
    - inventory\_id: Unique ID.
    - product\_id: Foreign key linking to Products.

- quantity: Stock level.
- last\_updated: Timestamp of last update.

## 4. Key SQL Queries

### View all Customers:

SELECT \* FROM Customers;



The screenshot shows the Oracle Live SQL interface. On the left, the Navigator pane lists several schema objects: CATEGORIES, CUSTOMERS, INVENTORY, ORDERS, ORDER\_ITEMS, PRODUCTS, and SUPPLIERS. The main workspace contains a SQL Worksheet with the query: "SELECT \* FROM Customers;". Below the query, the "Query result" tab is selected, displaying the following data:

CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	ADDRESS	CREATED_AT
1	John	Doe	john.doe@email.com	1234567890	123 Main St	2025-03-30T22:49:1
2	Jane	Smith	jane.smith@email.co	9876543210	456 Oak Ave	2025-03-30T22:49:1
3	Alice	Johnson	alice.johnson@email	5551112233	789 Pine Rd	2025-03-30T22:49:1
4	Bob	Williams	bob.williams@email.c	6662223344	101 Maple St	2025-03-30T22:49:1
5	Charlie	Brown	charlie.brown@email	7773334455	202 Birch Ln	2025-03-30T22:49:1
6	David	Miller	david.miller@email.c	8884445566	303 Cedar Ct	2025-03-30T22:49:1
7	Emma	Davis	emma.davis@email.c	9995556677	404 Elm Blvd	2025-03-30T22:49:1
8	Frank	Wilson	frank.wilson@email.c	1116667788	505 Spruce Dr	2025-03-30T22:49:1

At the bottom of the interface, there are links for About Oracle, Contact Us, Legal Notices, Terms and Conditions, Your Privacy Rights, Delete Your Live SQL Account, and Cookie Preferences. The page also includes a copyright notice: "Copyright © 2014, 2025 Oracle and/or its affiliates. All rights reserved." and a page number "r24".

### Retrieve all Orders with Customer Details:

Live SQL > Worksheet Library

Return to Live SQL Classic Sign Out

**Navigator**

My Schema

Tables

Search objects

CATEGORIES

CUSTOMERS

INVENTORY

ORDERS

ORDER\_ITEMS

PRODUCTS

SUPPLIERS

[ SQL Worksheet ]\*

```
1  SELECT o.order_id, o.order_date, o.total_amount, o.status,
2      c.first_name, c.last_name, c.email, c.phone
3  FROM Orders o
4  JOIN Customers c ON o.customer_id = c.customer_id;
```

Query result Script output DBMS output Explain Plan SQL history

Download Execution time: 0.006 seconds

	ORDER_ID	ORDER_DATE	TOTAL_AMOUNT	STATUS	FIRST_NAME	LAST_NAME	EMAIL	PHONE
1		1 2025-03-30T22:49:00Z	79.98	Pending	John	Doe	john.doe@email.com	1234567890
2		2 2025-03-30T22:49:00Z	49.99	Shipped	Jane	Smith	jane.smith@email.co	9876543210
3		3 2025-03-30T22:49:00Z	199.99	Delivered	Alice	Johnson	alice.johnson@email	5551112233
4		4 2025-03-30T22:49:00Z	59.99	Pending	Bob	Williams	bob.williams@email.c	6662223344
5		5 2025-03-30T22:49:00Z	29.99	Cancelled	Charlie	Brown	charlie.brown@email	7773334455
6		6 2025-03-30T22:49:00Z	99.99	Pending	David	Miller	david.miller@email.c	8884445566
7		7 2025-03-30T22:49:00Z	39.99	Shipped	Emma	Davis	emma.davis@email.c	9995556677

[About Oracle](#) | [Contact Us](#) | [Legal Notices](#) | [Terms and Conditions](#) | [Your Privacy Rights](#) | [Delete Your Live SQL Account](#) | [Cookie Preferences](#)

Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.

r24

## View Inventory Levels:

The screenshot shows the Oracle Live SQL interface. The top navigation bar includes 'Live SQL' (highlighted in red), 'Worksheet' (selected), and 'Library'. On the right, there are links to 'Return to Live SQL Classic' and 'Sign Out'. The left sidebar is titled 'Navigator' and contains a tree view of schema objects: 'My Schema' (selected), 'Tables', and a search bar. Below the tree are categories: 'CATEGORIES', 'CUSTOMERS', 'INVENTORY', 'ORDERS', 'ORDER\_ITEMS', 'PRODUCTS', and 'SUPPLIERS'. The main workspace is titled '[SQL Worksheet]'. It displays the following SQL query:

```
1  SELECT p.product_name, i.quantity, i.last_updated
2  FROM Inventory i
3  JOIN Products p ON i.product_id = p.product_id;
```

The 'Query result' tab is selected, showing the execution time as 0.039 seconds. The results are presented in a table:

	PRODUCT_NAME	QUANTITY	LAST_UPDATED
1	Formal Shirt	50	2025-03-30T22:49:1
2	Casual Jeans	40	2025-03-30T22:49:1
3	Winter Jacket	30	2025-03-30T22:49:1
4	Sneakers	60	2025-03-30T22:49:1
5	Leather Belt	70	2025-03-30T22:49:1
6	Summer Dress	20	2025-03-30T22:49:1
7	Graphic T-Shirt	45	2025-03-30T22:49:1
8	Business Suit	15	2025-03-30T22:49:1

At the bottom, there are links for 'About Oracle', 'Contact Us', 'Legal Notices', 'Terms and Conditions', 'Your Privacy Rights', 'Delete Your Live SQL Account', and 'Cookie Preferences'. A copyright notice states 'Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.' On the far right, it says 'r24'.

## Find Orders for a Specific Customer (Example: John Doe)

The screenshot shows the Oracle Live SQL interface. The top navigation bar includes 'Live SQL' (highlighted), 'Worksheet', 'Library', 'Return to Live SQL Classic', and 'Sign Out'. The left sidebar has a 'Navigator' tab selected, showing 'My Schema' (Tables) and a search bar. Below the sidebar is a tree view with nodes: CATEGORIES, CUSTOMERS, INVENTORY, ORDERS, ORDER\_ITEMS, PRODUCTS, and SUPPLIERS. The main workspace is titled '[ SQL Worksheet ]\*' and contains the following SQL code:

```
1  SELECT o.order_id, o.order_date, o.total_amount, o.status
2  FROM Orders o
3  JOIN Customers c ON o.customer_id = c.customer_id
4  WHERE c.first_name = 'John' AND c.last_name = 'Doe';
```

The 'Query result' tab is selected, showing the execution time: 0.007 seconds. The results are displayed in a table:

	ORDER_ID	ORDER_DATE	TOTAL_AMOUNT	STATUS
1		2025-03-30T22:49:11	79.98	Pending

At the bottom of the page, there are links: About Oracle, Contact Us, Legal Notices, Terms and Conditions, Your Privacy Rights, Delete Your Live SQL Account, and Cookie Preferences. A copyright notice states: Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.

## Alternative: View Data in a More Readable Format

If you want a more formatted display, use column alignment in SQL\*Plus or SQLcl:

```
SET LINESIZE 200;
```

```
SET PAGESIZE 50;
COLUMN first_name FORMAT A15;
COLUMN last_name FORMAT A15;
COLUMN email FORMAT A25;
COLUMN phone FORMAT A15;
COLUMN product_name FORMAT A20;
COLUMN category_name FORMAT A20;
COLUMN supplier_name FORMAT A20;
```

-- Now run SELECT commands again

```
SELECT * FROM Customers;
SELECT * FROM Categories;
SELECT * FROM Suppliers;
SELECT * FROM Products;
SELECT * FROM Orders;
SELECT * FROM Order_Items;
SELECT * FROM Inventory;
```

The screenshot shows the Oracle Live SQL interface. The top navigation bar includes 'Live SQL' (highlighted), 'Worksheet' (selected), 'Library', 'Return to Live SQL Classic', and 'Sign Out'. The left sidebar has 'Navigator' selected, showing 'My Schema' (Tables) and a search bar. Below this is a tree view of tables: CATEGORIES, CUSTOMERS, INVENTORY, ORDERS, ORDER\_ITEMS, PRODUCTS, and SUPPLIERS. The main workspace is titled '[SQL Worksheet]'. It contains the following SQL script:

```
1 SET LINESIZE 200;
2 SET PAGESIZE 50;
3 COLUMN first_name FORMAT A15;
4 COLUMN last_name FORMAT A15;
5 COLUMN email FORMAT A25;
6 COLUMN phone FORMAT A15;
7 COLUMN product_name FORMAT A20;
8 COLUMN category_name FORMAT A20;
9 COLUMN supplier_name FORMAT A20;
10
11 -- Now run SELECT commands again
12 SELECT * FROM Customers;
13 SELECT * FROM Categories;
14 SELECT * FROM Suppliers;
```

The 'Query result' tab is selected, showing the following data for the INVENTORY table:

	INVENTORY_ID	PRODUCT_ID	QUANTITY	LAST_UPDATED
1		1	1	50 2025-03-30T22:49:f
2		2	2	40 2025-03-30T22:49:f
3		3	3	30 2025-03-30T22:49:f
4		4	4	60 2025-03-30T22:49:f
5		5	5	70 2025-03-30T22:49:f

[About Oracle](#) | [Contact Us](#) | [Legal Notices](#) | [Terms and Conditions](#) | [Your Privacy Rights](#) | [Delete Your Live SQL Account](#) | [Cookie Preferences](#)  
Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.

r24

## 5. Database Management Best Practices

- Regularly update stock levels in the Inventory table.
- Ensure email addresses are unique in the Customers table.
- Use foreign keys to maintain referential integrity between tables.
- Backup the database periodically to prevent data loss.

## A Comprehensive Guide to the Design, Creation, and Management of SQL Tables

```
-- Drop tables if they already exist

BEGIN

    EXECUTE IMMEDIATE 'DROP TABLE Order_Items CASCADE CONSTRAINTS';
    EXECUTE IMMEDIATE 'DROP TABLE Orders CASCADE CONSTRAINTS';
    EXECUTE IMMEDIATE 'DROP TABLE Inventory CASCADE CONSTRAINTS';
    EXECUTE IMMEDIATE 'DROP TABLE Products CASCADE CONSTRAINTS';
    EXECUTE IMMEDIATE 'DROP TABLE Suppliers CASCADE CONSTRAINTS';
    EXECUTE IMMEDIATE 'DROP TABLE Categories CASCADE CONSTRAINTS';
    EXECUTE IMMEDIATE 'DROP TABLE Customers CASCADE CONSTRAINTS';

EXCEPTION

    WHEN OTHERS THEN NULL;

END;
/


-- Customers Table

CREATE TABLE Customers (
    customer_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    first_name VARCHAR2(50) NOT NULL,
```

```
last_name VARCHAR2(50) NOT NULL,  
email VARCHAR2(100) UNIQUE NOT NULL,  
phone VARCHAR2(20),  
address CLOB,  
created_at TIMESTAMP DEFAULT SYSTIMESTAMP  
);
```

-- Categories Table

```
CREATE TABLE Categories (  
    category_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    category_name VARCHAR2(50) UNIQUE NOT NULL,  
    description CLOB  
);
```

-- Suppliers Table

```
CREATE TABLE Suppliers (  
    supplier_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    supplier_name VARCHAR2(100) NOT NULL,  
    contact_name VARCHAR2(50),
```

```
email VARCHAR2(100) UNIQUE,  
phone VARCHAR2(20),  
address CLOB  
);
```

-- Products Table

```
CREATE TABLE Products (  
    product_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    product_name VARCHAR2(100) NOT NULL,  
    category_id NUMBER,  
    price NUMBER(10,2) NOT NULL,  
    product_size VARCHAR2(10), -- Renamed from SIZE to product_size  
    color VARCHAR2(50),  
    stock_quantity NUMBER DEFAULT 0,  
    supplier_id NUMBER,  
    FOREIGN KEY (category_id) REFERENCES Categories(category_id) ON DELETE CASCADE,  
    FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id) ON DELETE SET NULL  
);
```

```
-- Orders Table
```

```
CREATE TABLE Orders (
    order_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    customer_id NUMBER,
    order_date TIMESTAMP DEFAULT SYSTIMESTAMP,
    total_amount NUMBER(10,2),
    status VARCHAR2(20) DEFAULT 'Pending' CHECK (status IN ('Pending', 'Shipped', 'Delivered', 'Cancelled')),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id) ON DELETE CASCADE
);
```

```
-- Order_Items Table
```

```
CREATE TABLE Order_Items (
    order_item_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    order_id NUMBER,
    product_id NUMBER,
    quantity NUMBER NOT NULL,
    subtotal NUMBER(10,2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES Products(product_id) ON DELETE CASCADE
)
```

```
);
```

```
-- Inventory Table
```

```
CREATE TABLE Inventory (
    inventory_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    product_id NUMBER,
    quantity NUMBER NOT NULL,
    last_updated TIMESTAMP DEFAULT SYSTIMESTAMP,
    FOREIGN KEY (product_id) REFERENCES Products(product_id) ON DELETE CASCADE
);
```

```
-- Insert into Customers
```

```
INSERT INTO Customers (first_name, last_name, email, phone, address) VALUES
('John', 'Doe', 'john.doe@email.com', '1234567890', '123 Main St'),
('Jane', 'Smith', 'jane.smith@email.com', '9876543210', '456 Oak Ave'),
('Alice', 'Johnson', 'alice.johnson@email.com', '5551112233', '789 Pine Rd'),
('Bob', 'Williams', 'bob.williams@email.com', '6662223344', '101 Maple St'),
('Charlie', 'Brown', 'charlie.brown@email.com', '7773334455', '202 Birch Ln'),
('David', 'Miller', 'david.miller@email.com', '8884445566', '303 Cedar Ct'),
```

```
('Emma', 'Davis', 'emma.davis@email.com', '9995556677', '404 Elm Blvd'),  
('Frank', 'Wilson', 'frank.wilson@email.com', '1116667788', '505 Spruce Dr'),  
('Grace', 'Moore', 'grace.moore@email.com', '2227778899', '606 Walnut Pl'),  
('Henry', 'Taylor', 'henry.taylor@email.com', '3338889900', '707 Chestnut Ln');
```

-- Insert into Categories

```
INSERT INTO Categories (category_name, description) VALUES
```

```
('Shirts', 'Casual and formal shirts'),  
(Pants', 'Denim, chinos, and joggers'),  
(Jackets', 'Winter and casual jackets'),  
(Shoes', 'Casual and formal footwear'),  
(Accessories', 'Belts, wallets, and more'),  
(Dresses', 'Casual and party wear dresses'),  
(T-Shirts', 'Graphic and plain t-shirts'),  
(Suits', 'Formal and business suits'),  
(Hoodies', 'Winter and casual hoodies'),  
(Shorts', 'Casual summer shorts');
```

-- Insert into Suppliers

```
INSERT INTO Suppliers (supplier_name, contact_name, email, phone, address) VALUES
('ABC Textiles', 'Mike Johnson', 'mike@abc.com', '4445556666', 'New York, USA'),
('XYZ Clothing', 'Sara Lee', 'sara@xyz.com', '7778889999', 'Los Angeles, USA'),
('Fashion Hub', 'Tom Clark', 'tom@fashionhub.com', '1231231234', 'Paris, France'),
('Trend Wear', 'Lily Adams', 'lily@trendwear.com', '4564564567', 'London, UK'),
('Urban Styles', 'Robert Brown', 'robert@urban.com', '7897897890', 'Milan, Italy'),
('Classic Attire', 'Emma White', 'emma@classic.com', '3213213210', 'Berlin, Germany'),
('Elite Fashion', 'William Black', 'william@elite.com', '6546546543', 'Tokyo, Japan'),
('Luxury Designs', 'Sophia Green', 'sophia@luxury.com', '9879879876', 'Seoul, South Korea'),
('Budget Clothes', 'David Carter', 'david@budget.com', '1471471470', 'Bangkok, Thailand'),
('Street Wear Co.', 'Olivia Scott', 'olivia@streetwear.com', '2582582581', 'Sydney, Australia');
```

-- Insert into Products

```
INSERT INTO Products (product_name, category_id, price, product_size, color, stock_quantity, supplier_id) VALUES
('Formal Shirt', 1, 29.99, 'M', 'Blue', 50, 1),
('Casual Jeans', 2, 49.99, 'L', 'Black', 40, 2),
('Winter Jacket', 3, 99.99, 'XL', 'Red', 30, 3),
('Sneakers', 4, 59.99, '10', 'White', 60, 4),
('Leather Belt', 5, 19.99, 'L', 'Brown', 70, 5),
```

```
('Summer Dress', 6, 39.99, 'S', 'Pink', 20, 6),  
(('Graphic T-Shirt', 7, 25.99, 'M', 'Green', 45, 7),  
(('Business Suit', 8, 199.99, 'L', 'Navy', 15, 8),  
(('Hooded Sweatshirt', 9, 45.99, 'XL', 'Gray', 35, 9),  
(('Denim Shorts', 10, 35.99, 'M', 'Blue', 25, 10);
```

-- Insert into Orders

```
INSERT INTO Orders (customer_id, total_amount, status) VALUES  
(1, 79.98, 'Pending'),  
(2, 49.99, 'Shipped'),  
(3, 199.99, 'Delivered'),  
(4, 59.99, 'Pending'),  
(5, 29.99, 'Cancelled'),  
(6, 99.99, 'Pending'),  
(7, 39.99, 'Shipped'),  
(8, 19.99, 'Delivered'),  
(9, 45.99, 'Pending'),  
(10, 35.99, 'Shipped');
```

```
-- Insert into Order_Items
```

```
INSERT INTO Order_Items (order_id, product_id, quantity, subtotal) VALUES  
(1, 1, 2, 59.98),  
(2, 2, 1, 49.99),  
(3, 8, 1, 199.99),  
(4, 4, 1, 59.99),  
(5, 1, 1, 29.99),  
(6, 3, 1, 99.99),  
(7, 6, 1, 39.99),  
(8, 5, 1, 19.99),  
(9, 9, 1, 45.99),  
(10, 10, 1, 35.99);
```

```
-- Insert into Inventory
```

```
INSERT INTO Inventory (product_id, quantity) VALUES  
(1, 50),  
(2, 40),  
(3, 30),  
(4, 60),
```

```
(5, 70),  
(6, 20),  
(7, 45),  
(8, 15),  
(9, 35),  
(10, 25);
```

```
COMMIT;
```

## Conclusion & Benefits

### Conclusion

The **Clothing Store Database** provides a structured and efficient way to manage essential retail operations, including **customer management, product inventory, order processing, and supplier coordination**. By implementing this system, businesses can streamline workflows, reduce manual errors, and gain valuable insights into sales and inventory trends. This guidebook serves as a blueprint for setting up and optimizing the database for maximum efficiency.

### Benefits of This Database System

- ◊ **Efficient Data Management** – Organizes customer, product, and order details systematically.
- ◊ **Improved Inventory Tracking** – Helps maintain accurate stock levels, reducing shortages or overstocking.
- ◊ **Enhanced Customer Insights** – Tracks customer purchases, enabling personalized marketing and better service.

- ◊ **Seamless Order Processing** – Automates order management, reducing delays and errors.
- ◊ **Supplier Coordination** – Maintains supplier records for easier procurement and restocking.
- ◊ **Scalability & Performance** – Designed to handle business growth and increased data volume efficiently.
- ◊ **Data Security & Integrity** – Ensures accurate and secure storage of business-critical information.

## Future Enhancements & Use Cases

- ◊ **Integration with E-commerce Platforms** – Connect the database with online stores for real-time inventory updates.
- ◊ **Advanced Analytics & Reporting** – Implement business intelligence tools to generate sales and customer insights.
- ◊ **Automation & AI-driven Insights** – Use AI to predict sales trends and optimize stock levels.
- ◊ **Mobile Accessibility** – Develop a mobile-friendly dashboard for easy data access on the go.
- ◊ **Loyalty Programs & Customer Engagement** – Enhance customer retention with loyalty points and targeted promotions.
- ◊ **Multi-Store Support** – Expand the database to manage multiple retail locations efficiently.

## Platform Used

This database was designed and implemented using **Oracle Live SQL**, a cloud-based SQL development environment. Oracle Live SQL allows users to write, test, and execute SQL queries in a web-based interface without requiring local database installation.

🔗 Platform Link: [Oracle Live SQL](#)

1

---

<sup>1</sup> This guidebook serves as a foundational resource for database management in a clothing retail environment. It is intended for developers, business owners, and IT professionals seeking to implement and optimize a structured database system. Future adaptations may be required to align with emerging retail trends and technological advancements.