

# Online Retail Order Management System

## Project Overview

The **Online Retail Order Management System** is a database-driven backend built using **Oracle SQL** and **PL/SQL**. It manages **products, customers, orders, order items, and payments**, automating key business processes like:

- Order placement and cancellation
- Inventory updates via triggers
- Dynamic discount and tax calculation
- Exception handling for stock unavailability
- Analytical sales and performance reporting

## Technologies Used

- **Database:** Oracle Database
- **Languages:** SQL, PL/SQL
- **Tools:** Oracle SQL Developer
- **Key Concepts:**  
DDL, DML, Joins, Aggregations, Procedures, Functions, Triggers, Exception Handling

## Database Schema

### Entity Relationship Overview



### Tables

Table	Description
Products	Product catalog with prices and stock.
Customers	Customer details and types (Regular/VIP).
Orders	Order header info with subtotal, tax, and totals.
OrderItems	Links orders with purchased products.
Payments	Tracks payments and methods per order.

## PL/SQL Components

### Package: order\_pkg

Encapsulates core business logic for managing orders.

### Procedures

- **place\_order**  
Creates a new order, adds items, applies discounts & taxes, updates totals, and processes payment.
- **cancel\_order**  
Cancels an existing order, restores stock, and records a refund entry.

Functions

- **calc\_discount(customer\_id, subtotal)** → Returns discount based on customer type or amount.
- **calc\_tax(amount)** → Calculates 18% tax on the taxable amount.

Custom Exception

- **e\_stock\_unavailable** - Raised when stock is insufficient for requested quantities.

Triggers

Trigger	Event	Description
trg_order_item_after_insert	After Insert on order_items	Automatically decreases stock on purchase.
trg_products_stock_check	Before Update on products	Prevents negative stock values.

Execution Examples

Place an Order

```
DECLARE
  v_ids order_pkg.t_num_tab := order_pkg.t_num_tab(1,2);
  v_qties order_pkg.t_num_tab := order_pkg.t_num_tab(2,1);
  v_order_id NUMBER;
BEGIN
  order_pkg.place_order(2, v_ids, v_qties, 'CARD', v_order_id);
  DBMS_OUTPUT.PUT_LINE('Order placed: ' || v_order_id);
END;
/
```

Cancel an Order

```
BEGIN
  order_pkg.cancel_order(1001);
END;
/
```

Key Features

- Automated stock management via triggers
- Discount and tax automation

- Robust exception handling and rollback safety
  - Prebuilt reporting queries for analytics
  - Scalable and modular PL/SQL design
-