

## Mini Project Description

### Inventory and Sales System

**Project Objective:** As per the requirement from the client you are required to create a console based application using Java as frontend and Oracle as backend for their Inventory and Sales maintenance. Already the design team have completed the requirement design and you are expected to complete the assigned module.

#### Project Design:

**Database Design:** you are required to get the Database ready using Oracle SQLPlus.

#### Task 1:

Create a table called TBL\_STOCK with the given specification:

Column Name	Type	Description
Product_ID	Varchar length 6	Primary Key
Product_Name	Varchar length 20	Unique
Quantity_On_Hand	Number	Should not be < 0
Product_Unit_Price	Number	Should not be < 0
Reorder_Level	Number	Should not be < 0

Create a table called TBL\_SALES with the given specification:

Column Name	Type	Description
Sales_ID	Varchar Length 6	Primary Key
Sales_Date	Date	
Product_ID	Varchar length 6	Foreign Key from TBL_STOCK table
Quantity_Sold	Number	Should not be < 0
Sales_Price_Per_Unit	Number	Should not be < 0

#### Task 2:

Enter sample records into TBL\_STOCK table

Product_ID	Product_Name	Quantity_On_Hand	Product_Unit_Price	Reorder_Level
RE1001	REDMI Note 3	20	12000	5
ip1002	Iphone 5S	10	21000	2
PA1003	Panasonic P55	50	5500	5

Task 3:

Create the following sequences:

Sequence Name	Start value	Incremental Value
SEQ_SALES_ID	1000	1
SEQ_PRODUCT_ID	1004	1

Task 4:

Create a view named V\_SALES\_REPORT using TBL\_SALES table joined with TBL\_STOCK table based on ProductID order the result based on Profit\_Amount in descending and Sales\_ID in Ascending.

Column Name	Description
Sales_ID	
Sales_Date	
Product_ID	
Product_Name	
Quantity_Sold	
Product_Unit_Price	
Sales_Price_Per_Unit	
Profit_Amount	returns the difference between the Sales_Price_Per_Unit and Product_Unit_Price

**Application Design:** Create Java Application to manage the Sales:

Create a Java Application to manage the Sales and control the inventory. Create a new Java project under eclipse.

Task 5:

Create the following packages and the specified classes below.

Name of the package	Usage
com.wipro.sales.util	Contains the class that establishes the database connection
com.wipro.sales.bean	Contains all the bean classes
com.wipro.sales.dao	Contains the DAO classes that performs the real JDBC operations
com.wipro.sales.service	Contains the administrator class that receives input from Servlets and that invokes the respective DAO class methods
com.wipro.sales.main	Contains executable class with the main method

Under the package com.wipro.sales.util create the following classes.

Class	Method and Variables	Description
<b>DBUtil</b>		DB connection class
	public static Connection <b>getDBConnection()</b>	Establish a connection to the database and return the java.sql.Connection reference

Under the package com.wipro.sales.bean create the following classes.

Class	Method and Variables	Description
<b>Product</b>		<b>Bean Class</b>
String	productID	
String	productName	
int	quantityOnHand	
double	productUnitPrice	
int	reorderLevel	
	Setters and Getters for all properties	Using Eclipse, create getters and setters for all the properties

Class	Method and Variables	Description
<b>Sales</b>		<b>Bean Class</b>
String	salesID	
Java.util.Date	salesDate	
String	productID	
int	quantitySold	
double	salesPricePerUnit	
	Setters and Getters for all properties	Using Eclipse, create getters and setters for all the properties

Class	Method and Variables	Description
<b>SalesReport</b>		<b>Bean Class</b>
String	salesID	
Java.util.Date	salesDate	
String	productID	
String	productName	
int	quantitySold	
double	productUnitPrice	
double	salesPricePerUnit	
double	profitAmount	
	Setters and Getters for all properties	Using Eclipse, create getters and setters for all the properties

Under the package com.wipro.sales.dao create the following classes.

Class	Method and Variables	Description
<b>SalesDao</b>		<b>Dao Class</b>
	Int insertSales(Sales sales)	This method is used to insert the given sales obj into TBL_SALES table
	String generateSalesID(java.util.Date salesDate)	This method is used to generate Sales ID using the last 2 digit of the year part of the given date concatenated with the SEQ_SALES_ID sequence generated number.
	ArrayList<SalesReport> getSalesReport()	This method runs the V_SALES_REPORT view and stores every record in SalesREport Bean adding them to an arraylist. Which is return back to the user.

Class	Method and Variables	Description
<b>StockDao</b>		<b>Dao Class</b>
	Int insertStock(Product prdt)	This method is used to insert the given stock obj into TBL_STOCK table
	String generateProductID(String productName)	This method is used to generate Stock ID using the First 2 letters of the given product name concatenated with the SEQ_PRODUCT_ID sequence generated number.
	Int updateStock(String productID, int soldQty)	This method is used to update the Stock table by subtracting the current Quantity_On_Hand by the given soldQty of the given productID.
	Product getStock(String productID)	This method is used to fetch a specific record details from the Stock table for the given productID, store the information to a Product bean object the return the same.
	Int deleteStock(String productID)	This method is used to delete the stock record of the given ProductID

Under the package com.wipro.sales.service create the following classes.

Class	Method and Variables	Description
Administrator		<b>Service Class</b>
	String insertStock(Product prdt)	<p>This method is used to insert the given Product obj into the TBL_STOCK table using StockDao class insertStock method if the below conditions are successful.</p> <ol style="list-style-type: none"> <li>1. Stockobj should not be null</li> <li>2. ProductName should be of minimum 2 letters in length</li> <li>3. If above 2 are valid generate Product Id using StockDao class generateProductId method and store the same in the ProductID member of the given StockObject</li> </ol> <p>If any of the above conditions fail return "Data not Valid for insertion"</p> <p>Else Return the generated ProductId</p>
	String deleteStock(String ProductID)	<p>Delete the record of the given Productid using StockDao class deleteStock method, if delete is successful return "deleted" else return "record cannot be deleted"</p>
	String insertSales(Sales salesobj)	<p>This method is used to insert the given salesobj into the TBL_SALES table using SalesDao class insertSales method if the below conditions are successful.</p> <ol style="list-style-type: none"> <li>1. Salesobj should not be null else return "Object not valid for insertion"</li> <li>2. ProductID should be present in the TBL_STOCK table else return "Unknown Product for sales"</li> <li>3. Products current QuatityOnHand value should be more than the QuantitySold value else return "Not enough stock on hand for sales"</li> </ol>

		<p>4. SalesDate should be currentdate or earlier date and not future date, else return "Invalid date"</p> <p>5. If above 4 are valid generate Sales Id using SalesDao class generateSalesId method and store the same in the SalesID member of the given Sales Object</p> <p>Call the <b>insertSales</b> method of <b>SalesDao</b> and insert the record. If insertion is successful call the <b>updateStock</b> method of the <b>StockDao</b> and update the sold quantity to the stock.</p> <p>On successful completion of both the transaction return "Sales Completed" else "Error".</p>
	ArrayList<SalesReport> getSalesReport()	This method calls the getSalesReport of the SalesDao and returns the ArrayList

Under the package com.wipro.sales.main create the following classes.

Class	Method and Variables	Description
SalesApplication		<b>Executable Class</b>
	public static void main(String args[])	<p>This method has to display a main menu with following Options:</p> <ol style="list-style-type: none"> <li>1. Insert Stock</li> <li>2. Delete Stock</li> <li>3. Insert Sales</li> <li>4. View Sales Report</li> </ol> <p>Enter your Choice:</p> <p>On selecting the choice It should accept the required data from the user create appropriate object and call the valid method from the Administrator class.</p> <p>Eg: if the selected option is 1. Then create Stock bean object and get all Stock bean data from user and set it to the object and call insertStock method from the Administrator class.</p>