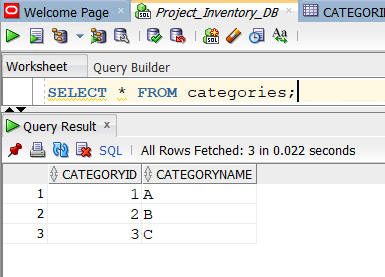
**Project 1: Database Management and Retrieval**

**1. Run the Inventory\_DB.sql Download Inventory\_DB.sqlscript to create the database tables.**

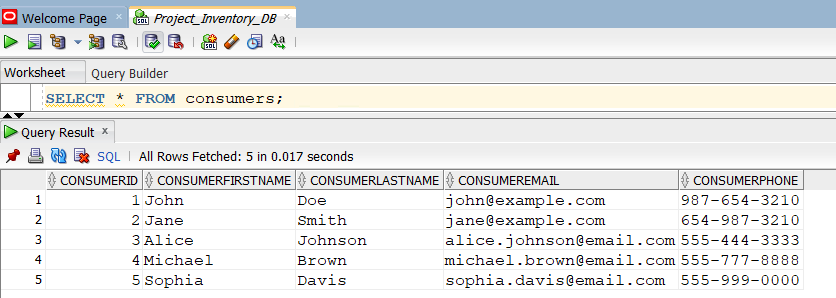
**Define and execute SQL queries that show the content of each table.**

**Categories Table:**



The above query returns all the records present in the **Categories** table and - all columns. The query returned 3 records. The **CategoryId** column serves as the Primary key in this table and there are no foreign keys in this table.

**Consumers Table:**

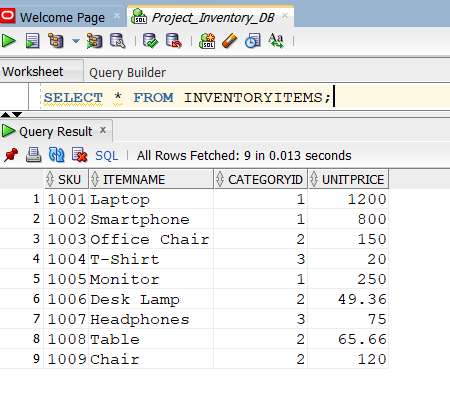


The above query returns all the records present in the **Consumers** table, including columns such as consumersId, first and last names, email and phone number. It returned 5 records.

The ConsumerId column is the as primary key of this table and this table doesn’t have any foreign keys.

This table contains most of the basic information about a consumer, starting with the first and last names, email and also the consumer’s phone number.

**INVENTORY ITEMS table:**

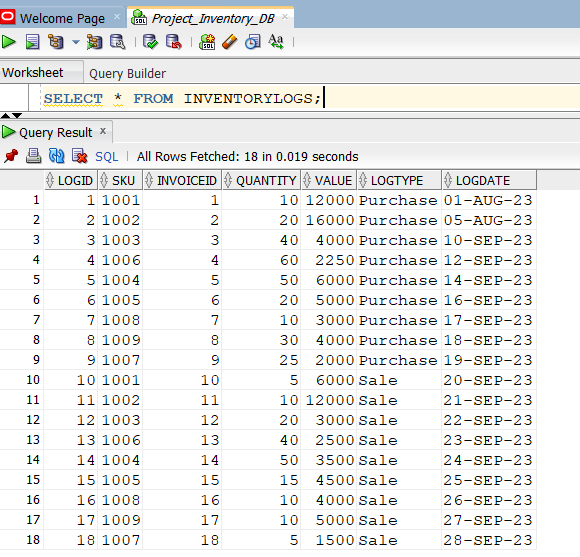


The above query returns all the records present in the InventoryItems table and - all columns. The query returned 9 records with the item’s SKU, names, CategoryId and unitprice.

The SKU column is considered as primary key of this table and CategoryId column is the foreign key in this table which is used to reference the data present in the Categories table.

This table holds the information related to each inventory item with their category and also their unit price.

**INVENTORYLOGS table:**

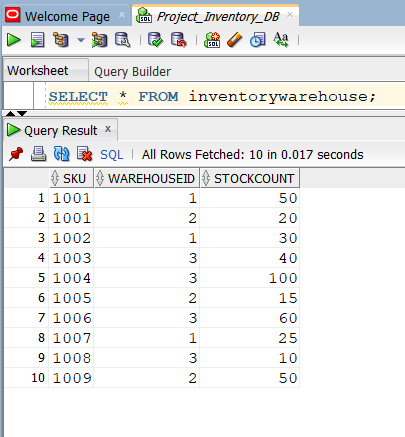


The above query returns all the records present in the **InventoryLogs** table and - all columns. The query returned 18 records with the record’s LogId, SKU, InvoiceId, Quantity, Value, Logtype and Logdate columns.

The LogId column is considered as primary key of this table, SKU and InvoiceId column are the foreign key in this table which are used to reference the data present in the InventoryItems and Invoice table.

This table holds the key information regarding the Inventory items transactions. This table tells us about the purchases or sales of inventory items along with their invoice id, logDate and their value.

**INVENTORY WAREHOUSE table:**

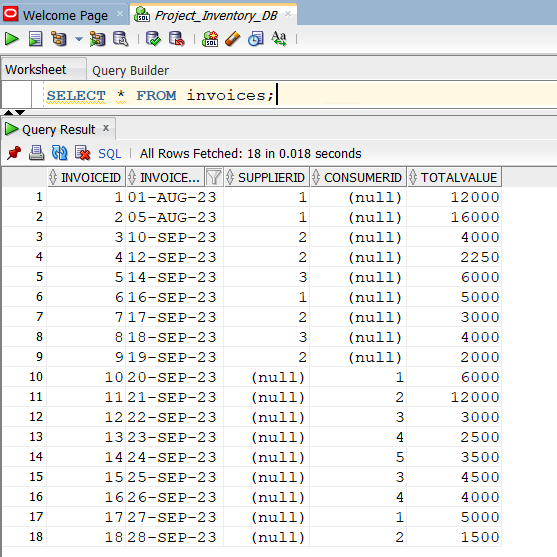


The above query returns all the records present in the InventoryWareHouse table and - all columns. The query returned 10 records with the record’s WarehouseId and StockCount columns.

The WareHouseId, SKU columns are considered as primary key of this table, SKU column is also the foreign key in this table which is used to reference the data present in the InventoryItems table.

This table also provides the information regarding the stock count of each inventory item present in different warehouses.

**Invoices table:**

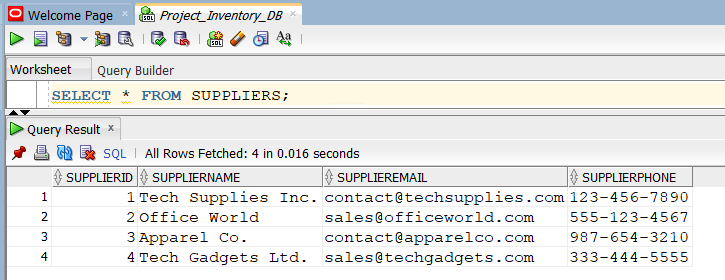


The above query returns all the records present in the **Invoices** table and - all columns. The query returned 18 records with the record’s InvoiceId, InvoiceDate, SupplierId, ConsumerId and totalvalue columns.

The InvoiceId column is considered as primary key of this table, SupplierId and ConsumerId column are the foreign key in this table which are used to reference the data present in the Supplier and Consumer tables.

This table holds the important information of Invoices. This table contains data such as invoicedate, supplier’s information and also the consumer’s information along with the total value.

**Suppliers table:**

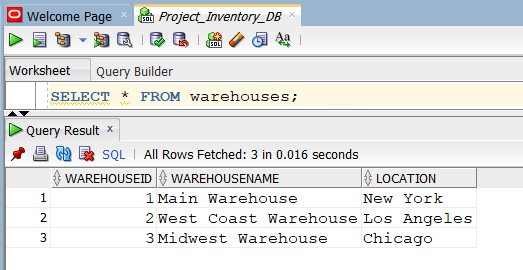


The above query returns all the records present in the Suppliers table and - all columns. The query returned 4 records with the record’s SupplierId, SupplierName, SupplierEmail and SupplierPhone columns.

The SupplierId column is considered as primary key of this table, and there are no foreign keys present in this table.

This table provides the information regarding different Suppliers along with their names, email and phone numbers.

**WARE HOUSES Table:**



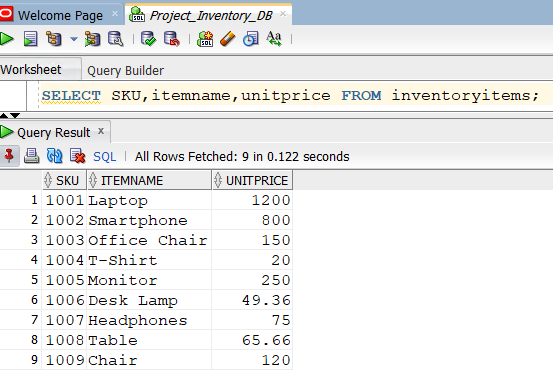
The above query returns all the records present in the Warehouse table and - all columns. The query returned 3 records with the record’s WareHouseId. WareHouseName and Location columns.

The WareHouseId column is considered as primary key of this table, and there are no foreign keys present in this table.

This table provides the information regarding different Warehouses information along with their names and location.

**2. Define and execute SQL queries to display data for one or more columns:**

**A. Write a query that displays the SKU, ItemName, and UnitPrice for all inventory items.**



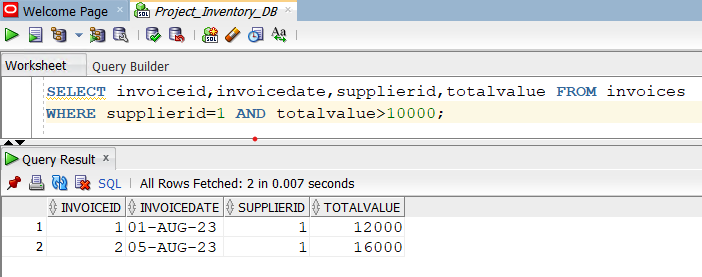
**What business question do you think this SQL query answers?**

* What are the SKUs, item names and unit prices of all inventory items?
* Give a list of all inventory Items present, along with their SKU, name and unit price information.

**How can the results of this query be beneficial or applied by the business?**

* This information provides us with the basic and most important attributes of each inventory item i.e., their unique identifier (SKU), name and unit price of that particular inventory item.
* This information can be used for managing the inventory and warehouses for keeping track of items present along with their value.
* It also helps to check the inflation/deflation of unit prices of items and also to make the pricing strategies.

**B. Write a query that displays all the invoices for supplier ID 1 with a total value greater than 10,000. Display only the Invoice ID, Invoice Date, Supplier ID, and the Invoice Total Value.**



**What business question do you think this SQL query answers?**

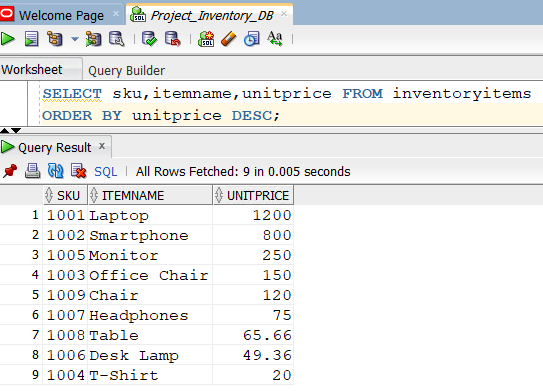
* Among all the invoices generated, what are the invoice transactions (Including the invoiceId, Invoicedate, supplierId and totalvalue) that are done with supplier whose SupplierId = 1 and the total value is greater than 10,000?
* What are the high valued transactions (total value >10,000) made with the supplier whose SupplierId=1 including the invoiceId, Invoicedate, supplierId and totalvalue?

**How can the results of this query be beneficial or applied by the business?**

* Identifying the high valued transactions from a supplier can help the business to guide the future negotiations and better the relationship with that supplier, which can be beneficial in long run where profits can be made for bulk purchases by getting discounts or better payment terms, from the supplier.
* This can be helpful for cost cutting programs, where evaluating the high valued transactions that are being made with supplier 1 and re-negotiate the quantity and prices to manage the business expenditure.

**3. Define and execute SQL queries to sort retrieved data:**

**A. Write a query that retrieves the SKU, name, and unit price of all inventory items, sorted in descending order by unit price.**



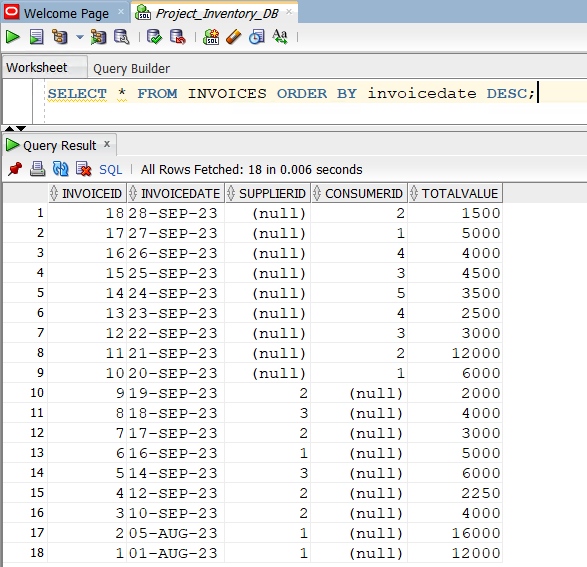
**What business question do you think this SQL query answers?**

* What are the SKUs, item names and unit prices of all inventory items, ordered from highest priced item to lowest?

**How can the results of this query be beneficial or applied by the business?**

* Helps the business understand which items are most expensive and least expensive.
* This understanding helps the business to focus on driving initiatives to increase the sale of their expensive products and increase their revenues and profit margins.
* This data along with the number of items sold can help the inventory management team to always keep a stock of most revenue generating products as per the sales.

**B. Write an SQL query to display all invoices, sorted by InvoiceDate in descending order.**



**What business question do you think this SQL query answers?**

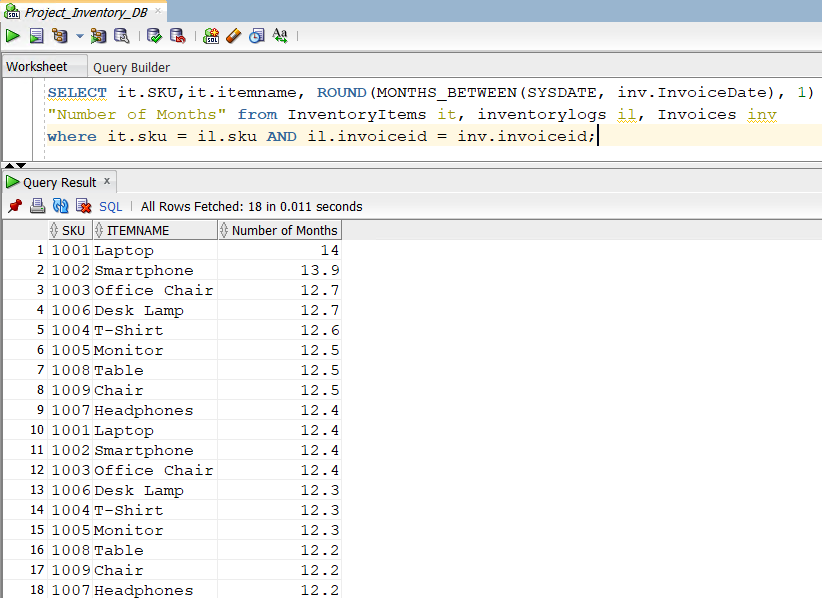
* What are all the records of invoices present in the system, ordered from the most recent to the oldest

**How can the results of this query be beneficial or applied by the business?**

* The business can look into the most recent transactions using this data.
* The results help the business understand all the invoices present in the system according to the date of the transaction aiding at the time of auditing and accounting.
* This data helps the business understand what are the costs incurred because of purchases and what are the costs incurred because of sales based on the null value present in SupplierId and ConsumerId.

**4. Define and execute SQL queries to generate data:**

**A. Write a query that displays the SKU, ItemName, and the number of months between today and the InvoiceDate for all inventory items, rounded to one decimal place**.



Since the same inventory items are present in several multiple invoices, the same inventory item have multiple entries in the result.

**What business question do you think this SQL query answers?**

* How many months has it been since the last purchase or sale of each inventory item included with the items SKU and item name?

**How can the results of this query be beneficial or applied by the business?**

* This result can help the business to calculate the age of inventory items which are purchased. The results help the business to sell the items that are close to expiration or warranty period.
* This result also helps the business owners understand the quantities of products that are to be purchased, there by evaluating demand to quantity of purchases of inventory items.
* This result also helps in preparing marketing strategies for the products are in the inventory longer than expected.
* This result also supports in forecasting the revenue that should be spent on purchases in the upcoming time frame to have a profitable margin share.
* This result can also help in separating dead stock of inventory items that use up storage space in the inventories.

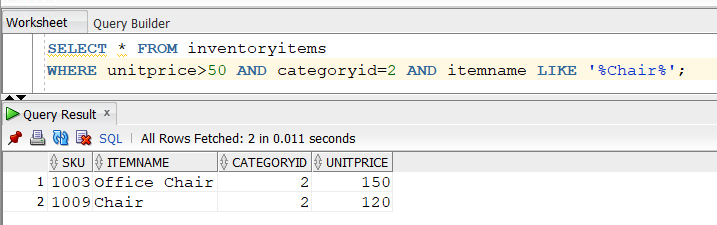
**5. Use functions to complete the following, and specify what business question you think each SQL query answers and how can the result of each of the query be beneficial or applied by the business:**

**A. Write a query that filters and displays all rows from the InventoryItems table such that the following parameters are true:**

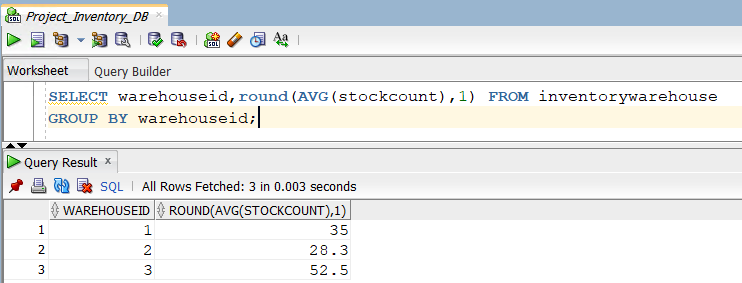
**The UnitPrice is greater than 50**

**The CategoryID is equal to 2**

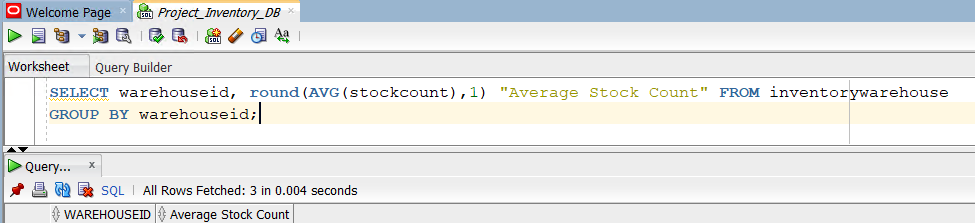
**The ItemName contains the word 'Chair'**



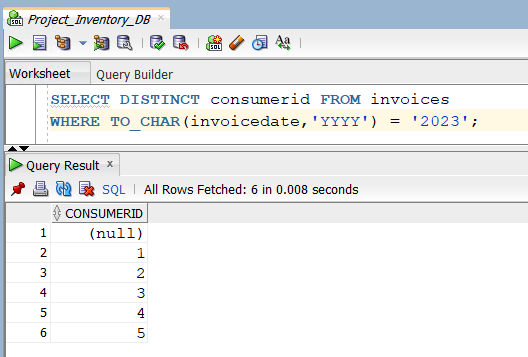
**B. Write a query to find the average stock count of inventory items in each warehouse.**



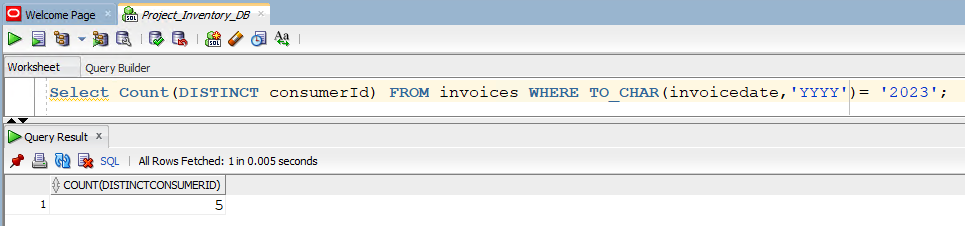
Query to get the result’s Column name as “Average Stock Count”:



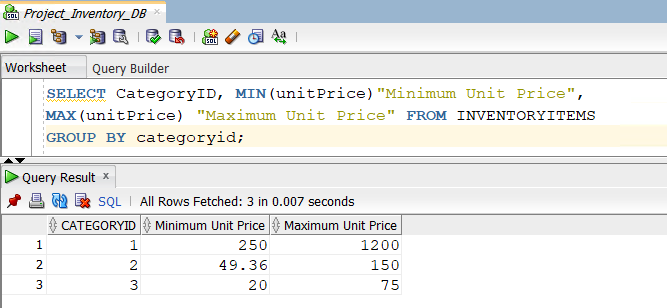
**C. Write a query to find how many unique consumers who made purchases in 2023?**



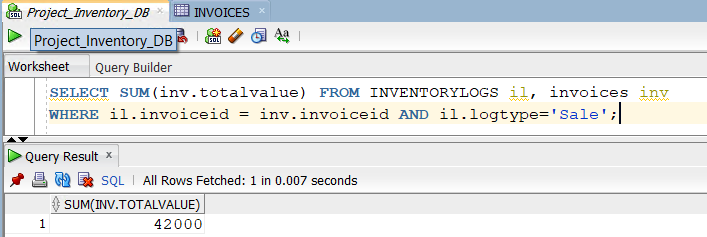
* Here the resulted 6 rows gives us the count of unique consumers including the records with consumerId’s as null.
* This can also be done using the COUNT () method. But using count will vomit the records which have null in consumerIds, hence the result is 5.



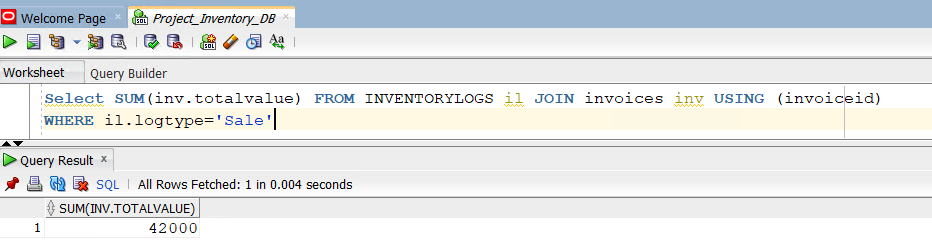
**D. Write a query to display the minimum and maximum UnitPrice of all items in each category.**



**E. Calculate the total revenue from all sales.**



JOIN..USING:



**6. Define and execute SQL commands to insert, update, and delete table records:**

**You need to complete the following:**

**A. You are required to manage the inventory database by inserting new records. Follow the steps below to ensure data integrity.**

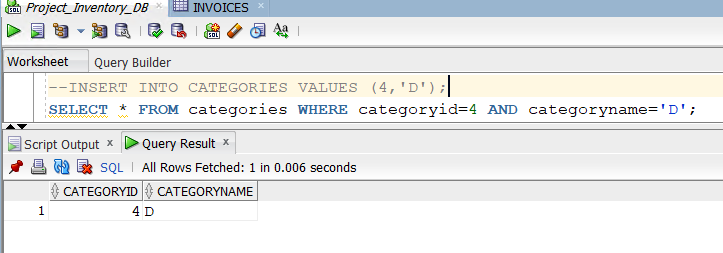
**Insert a New Category:**

**Write a SQL command to add a new category for inventory items with the following details:**

**CategoryID: 4**

**CategoryName: 'D'**

**NOTE: The lines in the comments are used to perform DML on the table, followed by a query that identifies as a proof showing that the DML has happened.**



**Insert a New Inventory Item:**

**Write a SQL command to add a new inventory item with the following details, ensuring it references the newly added category:**

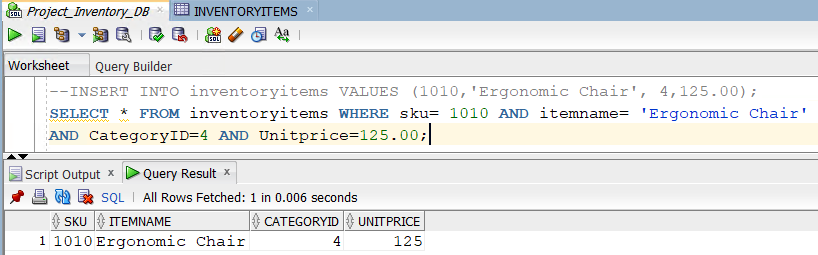
**SKU: 1010**

**ItemName: 'Ergonomic Chair'**

**CategoryID: 4**

**UnitPrice: 125.00**

**NOTE: The lines in the comments are used to perform DML on the table, followed by a query that identifies as a proof showing that the DML has happened.**



**Insert Inventory Stock Information:**

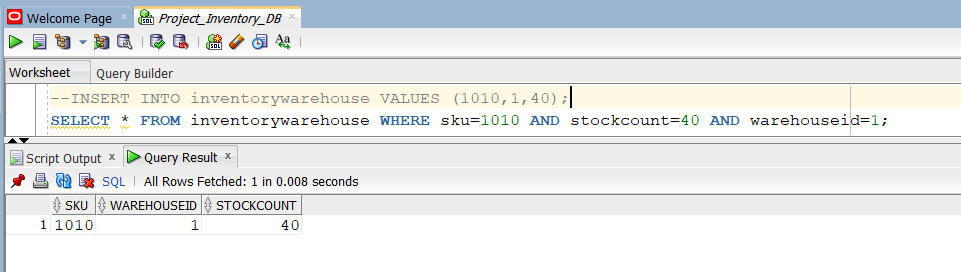
**Write SQL command to add stock information for the new inventory item in a warehouse. Use the following details:**

**SKU: 1010**

**WarehouseID: 1**

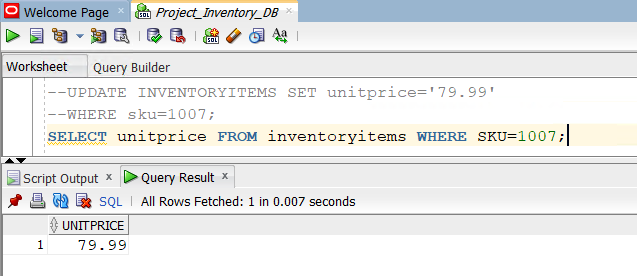
**StockCount: 40**

**NOTE: The lines in the comments are used to perform DML on the table, followed by a query that identifies as a proof showing that the DML has happened.**



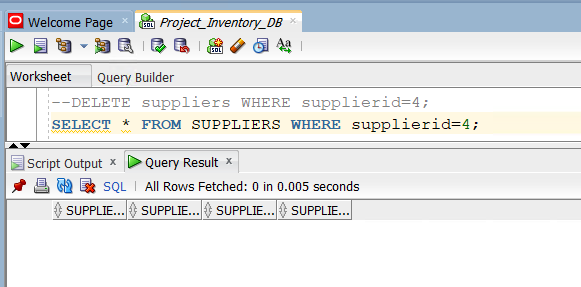
**B. Define and execute SQL command to update the UnitPrice of the item with SKU = 1007 (Headphones) in the InventoryItems table to 79.99**

**NOTE: The lines in the comments are used to perform DML on the table, followed by a query that identifies as a proof showing that the DML has happened.**

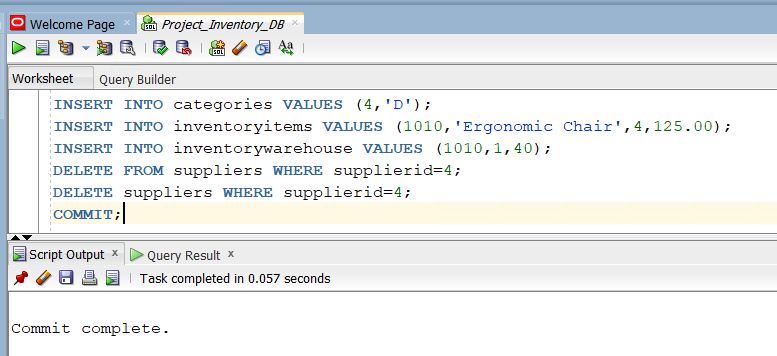


**C. Define and execute SQL command to delete the supplier with SupplierID = 4 .**

**NOTE: The lines in the comments are used to perform DML on the table, followed by a query that identifies as a proof showing that the DML has happened.**



**D. Commit the changes (Note: make sure to commit the data transactions for Section 6 before proceeding to Section 7).**



**What is the importance of these SQL commands? Why would the business apply these SQL commands?**

* Data manipulation language (DML) statements are used in SQL to modify or view the data within any table. These can be Insert/Update/Delete. These SQL commands are used to create/update/delete records in different tables such as Categories, inventory items, inventory ware house and suppliers.
* These commands hold a significant importance to any business because every inventory needs a management system where the data entries of various items along with its supplier’s information and inventory items storage information are stored. This helps any industry or company to run the business in a productive manner.

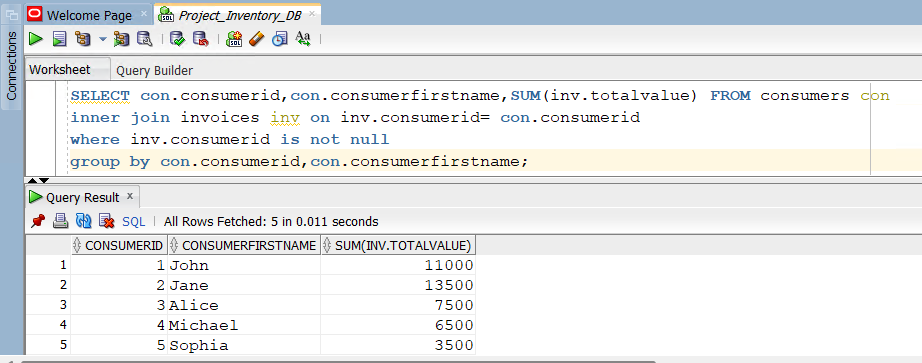
**7. Joining data from tables:**

**A. Write a query to list the ConsumerID, ConsumerFirstName, and the total value of their invoices, ensuring that you only get consumers who have made purchases.**

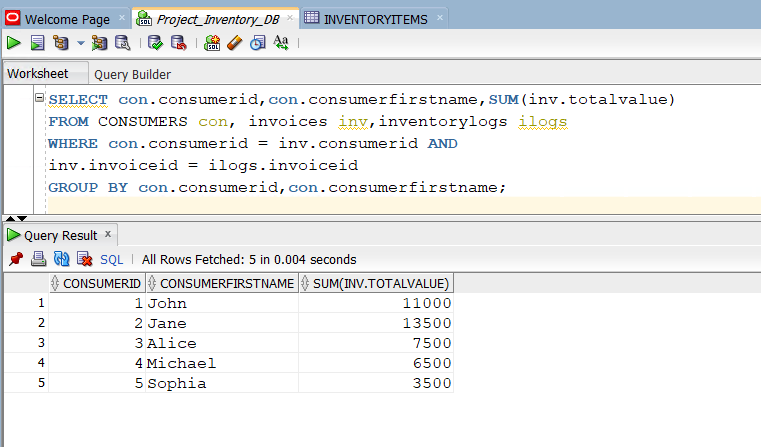
This question can be answered in two different ways, one is using the Inner join result that joins Consumers and Invoices tables based on consumerId column.

The 2nd result is using the implicit way of using multiple where conditions to join the Consumer and invoices tables.

**Using Inner Join:**



**Using Implicit Join:**



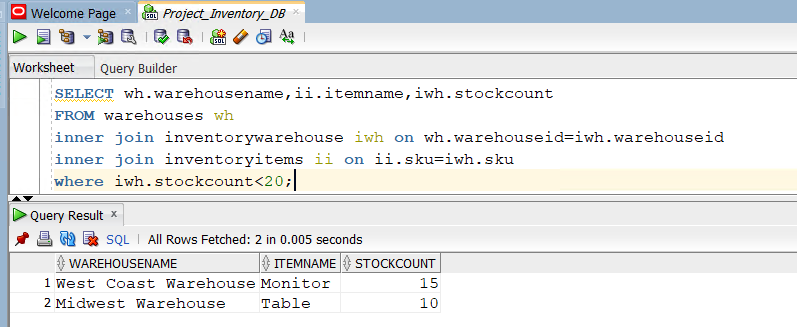
**What business question do you think this SQL query answers?**

* What are the CustomerIds, customerfirstnames and the total values of items they have purchased from the invoices?

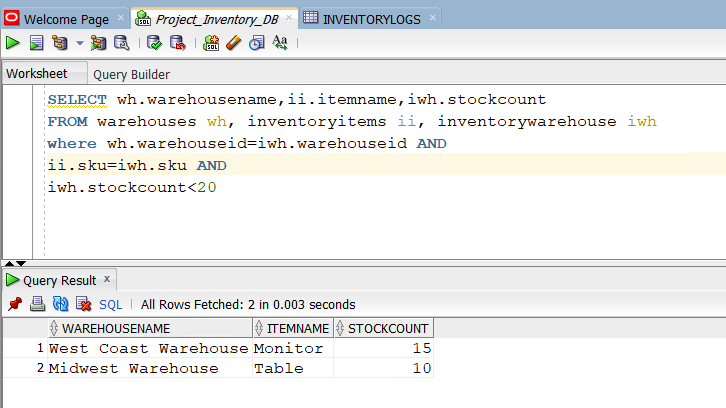
**B. Write a SQL query to display the WarehouseName, ItemName, and StockCount for all items with a stock level below 20.**

This question can be answered in two different ways, one is using the Inner joins and the other using the implicit join using multiple where conditions to join tables.

Using Inner Joins to join Warehouses, inventorywarehouse and inventoryitems:



Joining using multiple where conditions:



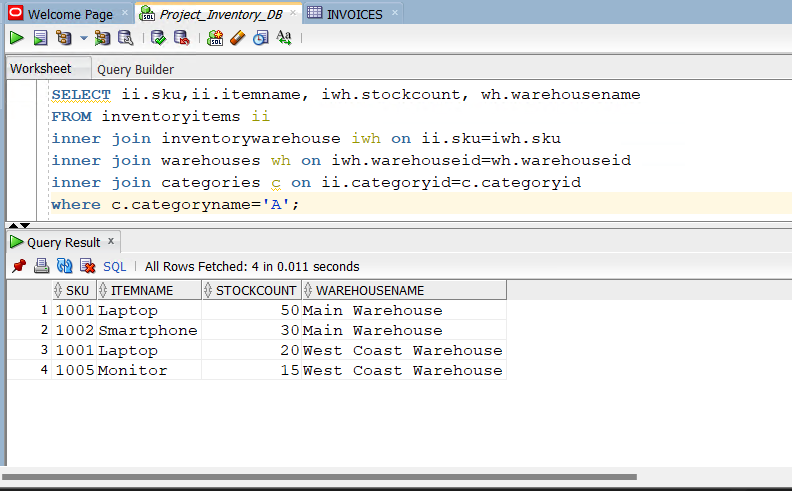
**What business question do you think this SQL query answers?**

* What are the warehouses names, item names and stock counts among all the inventory items whose stock count at a particular warehouse is less that 20?

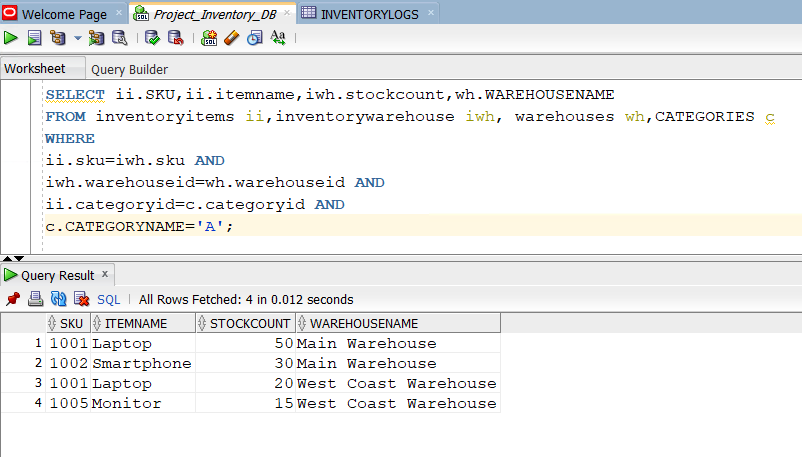
**C. Write a SQL query to display the SKU, ItemName, StockCount, and WarehouseName for all items in Category A.**

This question can be answered in two different ways, one is using the Inner joins and the other using the implicit join using multiple where conditions to join tables.

**Using Inner Joins to join Inventoryitems, inventorywarehouse, warehouses and categories tables:**

****

**Using multiple where conditions to join tables:**



**What business question do you think this SQL query answers?**

* What are the SKUs, Item names of inventory items that belong to category ‘A’ along with their stock count and warehouse name information?