Smart Home Products Online Store Database

CIS – 5430 SPRING 2023

Group -8 Dhwani Vaishnav, Phue Thant, Pratiksha Yadav

Contents

Introduction

In today's rapidly evolving world of smart home technology, efficient and organized data management is crucial for businesses that manufacture or sell smart home products. Our database design system provides a robust solution for managing data related to customers, orders, products, order items, and employees, with a focus on handling both in-house manufactured products and products supplied by suppliers.

Our system is designed to streamline the process of managing smart home product data, enabling businesses to easily track customer information, process orders, and monitor employee activities. The system includes various tables, such as customer table for storing customer details, order table for capturing order information, product table for managing product details, order item table for recording order items, and employee table for tracking employee information.

One unique feature of our system is the ability to handle both in-house manufactured products and products supplied by suppliers. This allows businesses to efficiently manage products from different sources, keeping track of product specifications, availability, pricing, and supplier information. With this system, we can easily track the entire supply chain of smart home products, from manufacturing to distribution, ensuring smooth operations and timely deliveries.

Our user-friendly interface and comprehensive functionality make it easy for businesses to input, retrieve, and analyze data related to smart home products. As manufacturer, distributor, or retailer of smart home products, our database design system provides the tools that are needed to effectively manage and organize our data, helping us to make informed decisions and optimize our business processes.

Purpose

The purpose of the Smart Home online store database documentation is to provide information about the structure, components, and functionality of the database. It serves as a reference guide for developers, administrators, and other stakeholders to understand how the database is designed, implemented, and used, facilitating effective management and maintenance of the online store's data.

Functionalities

The functionality of the Smart Home products online store database documentation includes:

Describing entity types, relationships, and attributes: The documentation outlines the entity types in the database, their relationships, and associated attributes, providing an understanding of how data is organized and stored.

Detailing data validation rules and constraints: The documentation specifies validation rules and constraints for data in the database, ensuring data accuracy and consistency by defining data types, allowed values, and business rules.

Providing DDL & DML statements for operations: The documentation includes instructions for performing Create, Read, Update & Delete operations on the database, guiding users on how to interact with the database effectively and safely.

Providing ORDBMS related support: The documentation utilizes PL/SQL blocks for data processing, improving performance and reducing data transfer, and Object Types for encapsulation of data and logic, enhancing code organization, reusability, and security while integrating with database capabilities.

Users

The users of the Smart Home products online store database documentation may include the Database designers & developers, DBAs, Business analysts, End users & Maintenance and support team.

Roles

All team members have fulfilled specific roles in the implementation of both Project 1 and Project 2, including responsibilities such as team leadership, data modeling, database design, and database development.

Below are the defined roles in the team:

Role	Team Member
Team Lead	Dhwani Vaishnav
Data Modeler	Pratiksha Yadav
Database Designer	Dhwani Vaishnav, Pratiksha Yadav, Phue Thant
Database Developer	Dhwani Vaishnav, Pratiksha Yadav, Phue Thant
Quality Assurance Engineer	Dhwani Vaishnav, Pratiksha Yadav, Phue Thant

Group Project 1

Conceptual and Logical Design

The online store selling Smart home products needs a system to keep track of orders and inventory. This includes recording customer orders, updating available stock, and confirming stock availability when orders are processed. The system should also maintain records of customers, employees, suppliers, and manage manufacturing and supplier-related details such as quantity, locations, brands, distributors, etc.

Let's examine the key entity types in our database in detail:

- Customer: This entity stores information about our customers, including a unique customer number, the employee managing the customer, shipping, and billing addresses, contact details, and demographic information such as gender and age.
- Product: This entity type captures data related to the products sold by our company. Each product has a unique product number and can be categorized as either manufactured by our own brand or purchased from another brand's distributor. We also track product details, warranty information, and pricing.
- Employee: This entity type contains comprehensive information about our company's employees, including their names, addresses, sex, and date of birth (DOB).
- Orders: This table allows us to track each order placed, which helps manage inventory. We store the order by its unique order number, order date, order price, warranty details, and order status, providing valuable information for inventory management and order processing.

Business Rules

One customer may or may not place many orders.
One order must be placed by one and only one customer.

One order must contain one or more product.
One product may or may not be in many orders.

One employee may process one or more orders.
One order must be processed by one and only one employee.

One product must be either manufactured or purchased.

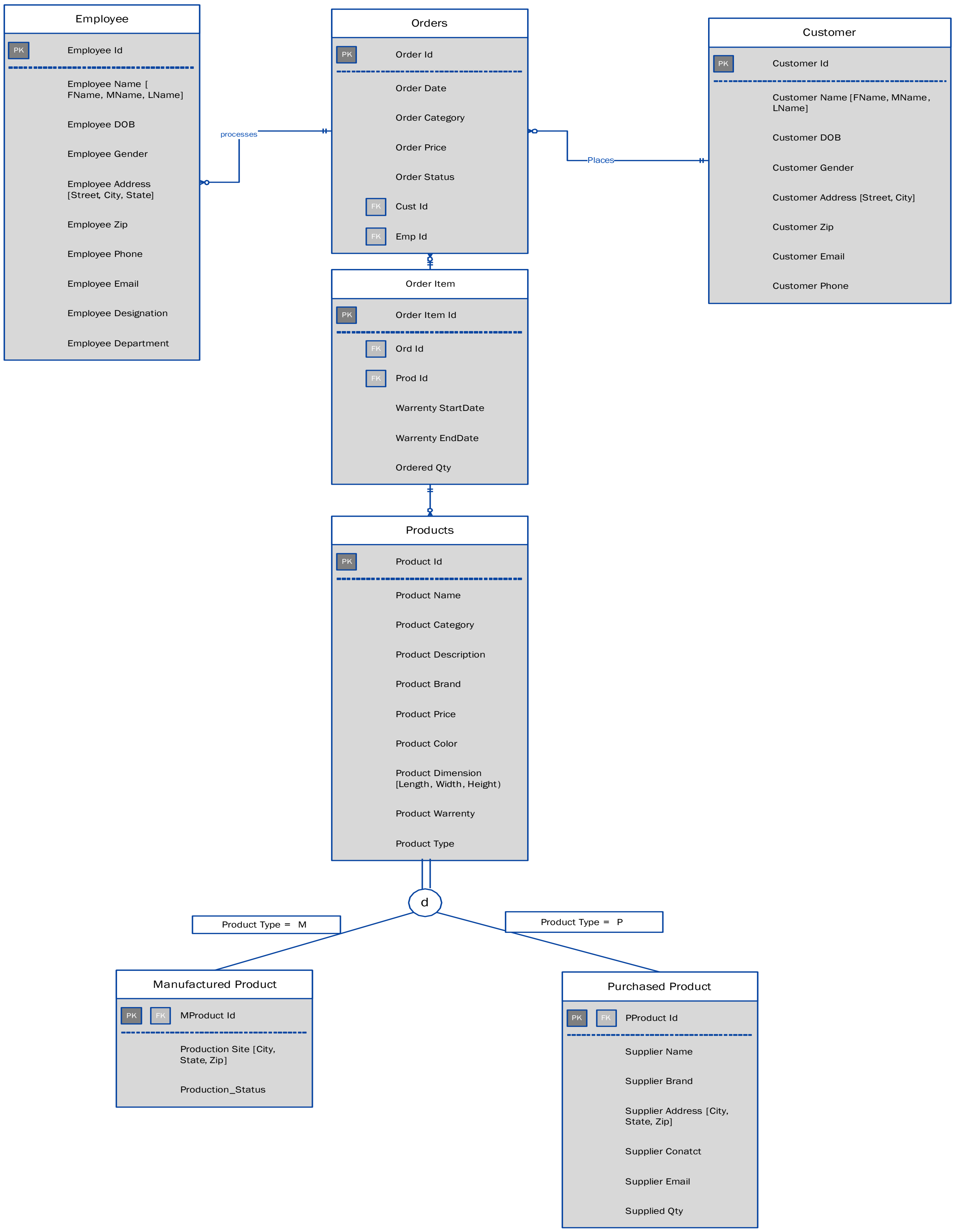
Identify entity types and relationship types. Fill out the following relationship matrix.

.

	Customer	***Orders***	***Products***	***Employee***
Customer	--	Places	--	--
Orders	Is placed	--	Contains	Is processed
Products	--	Has	Manufactured/Purchased	--
Employee	--	Processes	--	--

Draw an ER/EER diagram using software tools

includes 1) entity types, 2) relationship types, 3) keys, 4) attributes, and cardinality constraints


Employee
PK Employee Id
Employee Name [FName, MName, LName]
Employee DOB
Employee Gender
Employee Address [Street, City, State]
Employee Zip
Employee Phone
Employee Email
Employee Designation
Employee Department
processes
Orders
PK Order Id
Order Date
Order Category
Order Price
Order Status
FK Cust Id
FK Emp Id
Places
Customer
PK Customer Id
Customer Name [FName, MName, LName]
Customer DOB
Customer Gender
Customer Address [Street, City]
Customer Zip
Customer Email
Customer Phone
Order Item
PK Order Item Id
FK Ord Id
FK Prod Id
Warrenty StartDate
Warrenty EndDate
Ordered Qty
Products
PK Product Id
Product Name
Product Category
Product Description
Product Brand
Product Price
Product Color
Product Dimension [Length, Width, Height)
Product Warrenty
Product Type
d
Product Type = M
Product Type = P
Manufactured Product
PK FK MProduct Id
Production Site [City, State, Zip]
Production_Status
Purchased Product
PK FK PProduct Id
Supplier Name
Supplier Brand
Supplier Address [City, State, Zip]
Supplier Conatct
Supplier Email
Supplied Qty

Database Logical Design

Map the ER diagram to a relational database schema indicating the relation's name, primary key and foreign key. Add appropriate additional attributes by yourself.

Customer:

Customer Id	Customer FName	Customer LName	Customer DOB	Customer Gender	Customer Street	Customer City	Customer State	Customer Zip	Customer Phone	Customer Email

Orders:

Order Id	Order Date	Order Category	Order Price	Order Status	Cust Id	Emp Id

Order Item:

Order Item Id	Ord Id	Prod Id	Ordered Qty	Warranty StartDate	Warranty EndDate

Product:

Product Id	Product Name	Product Category	Description	Brand	Product Price	Product Length	Product Width	Product Height	Product Color	Warranty Period	Product Type

Manufactured Product:

MProduct Id	Production Site City	Production Site State	Production Site Zip	Production Status

Purchased Product:

PProduct Id	Supplier Name	Supplier City	Supplier State	Supplier Zip	Supplied Brand	Supplier Phone	Supplier Email	Supplied Qty

Employee:

Employee Id	Employee FName	Employee LName	Employee DOB	Employee Gender	Employee Street	Employee City	Employee State	Employee Zip	Employee Phone	Employee Email	Designation	Department

Establish join paths for the above relational database using the referential integrity Indicate all the foreign keys (FK).

F.K. Orders.Cust Id -> P.K. Customer.Customer Id

F.K. Orders.Emp Id -> P.K Employee.Employee Id

F.K. Order Item.Ord Id -> P.K Orders.Order Id

F.K. Order Item.Prod P.K -> Products.Product Id

F.K. Manufactured Product.MProduct Id -> P.K Products.Product Id

F.K. Purchased Product.PProduct Id -> P.K Products.Product Id

Do function analysis for each of your tables

Attribute A -> Attribute B (Determinant attribute(s) Determines Dependent Attribute(s))

Full Dependencies:

- Customer Id -> Customer Name, Customer DOB, Customer Gender, Customer Street, Customer City, Customer State, Customer Zip, Customer Phone, Customer Email
- Order Id -> Order Date, Order Category, Order Price, Order Status
- Order Item Id -> Order Qty, Warranty StartDate, Warranty EndDate
- Product Id -> Product Name, Product Category, Description, Brand, Product Price, Product Length, Product Width, Product Height, Product Color, Warranty Period, Product Type
- MProduct Id -> Production Site City, Production Site State, Production Site Zip, Production Status
- PProduct Id -> Supplier Name, Brand, Supplier City, Supplier State, Supplier Zip, Supplier Contact, Supplied Qty
- Employee Id -> Employee Name, Employee DOB, Employee Gender, Employee Street, Employee City, Employee State, Employee Zip, Employee Phone, Employee Email, Designation, Department

Transitive Dependencies:

- Customer Zip -> Customer Street, Customer City, Customer State
- Production Site Zip -> Production Site City, Production Site State
- Supplier Zip -> Supplier City, Supplier State
- Employee Zip -> Employee Street, Employee City, Employee State

NOTE: We are not normalizing tables with above transitive dependencies since it's not a good idea to separate address fields in different tables to prevent more JOINS than required for querying the data

Show all the normalized tables and indicate their normalization form

Table Name	1NF	2NF	3NF
Customer	✓	✓	
Order	✓	✓	✓
Order Item	✓	✓	✓
Product	✓	✓	✓
Manufactured Product	✓	✓	
Purchased Product	✓	✓	
Employee	✓	✓	

Tables in 2 NF and 3 NF:

Customer (2 NF)

Customer Id	Customer Name	Customer DOB	Customer Gender	Customer Street	Customer City	Customer State	Customer Zip	Customer Phone	Customer Email

Order (3NF)

Order Id	Order Date	Order Category	Order Price	Order Status	Cust Id	Emp Id

Order Item (3NF)

Order Item Id	Ord Id	Prod Id	Ordered Qty	Warranty StartDate	Warranty EndDate

Product (3NF)

Product Id	Product Name	Product Category	Description	Brand	Product Price	Product Length	Product Width	Product Height	Warranty Period	Product Color	Product Type

Manufactured Product (2NF)

MProduct Id	Production Site City	Production Site State	Production Site Zip	Production Status

Purchased Product (2 NF)

PProduct Id	Supplier Name	Supplier City	Supplier State	Supplier Zip	Supplied Brand	Supplier Contact	Supplied Qty

Employee (2NF)

Employee Id	Employee Name	Employee DOB	Employee Gender	Employee Street	Employee City	Employee State	Employee Zip	Employee Phone	Employee Email	Designation	Department

Group Project 2

Database Creation Script (Tables, Constraints & Inserting Data)

Table Name: Customer

```
DROP TABLE Customer CASCADE CONSTRAINTS;

 CREATE TABLE Customer
(
Customer_Id VARCHAR2(20) NOT NULL,
Customer_FName VARCHAR2(25) NOT NULL,
Customer_LName VARCHAR2(25) NOT NULL,
Customer_DOB CHAR(30),
Customer_Gender CHAR(20),
Customer_Address VARCHAR(100),
Customer_City VARCHAR(50),
Customer_State CHAR(2),
Customer_Zip VARCHAR(9),
Customer_Phone CHAR(10),
Customer_Email VARCHAR(256) NOT NULL,
CONSTRAINT CustomerPK PRIMARY KEY(Customer_Id),
CONSTRAINT Customer_UK_CustomerEmail UNIQUE (Customer_Email),
CONSTRAINT Customer_NN_Customer_FName CHECK (Customer_FName IS NOT NULL),
CONSTRAINT Customer_NN_Customer_LName CHECK (Customer_LName IS NOT NULL)
);
```

Inserting values into Customer Table:

```
INSERT INTO Customer VALUES('CUST_01','John','Will','01-Jan-1885','M','50
Crossstreet','Tucson','AZ',99040,'9689526365','johnwill@gmail.com');

INSERT INTO Customer VALUES('CUST_02','Smith','Jonas','02-Feb-1889','M','6000 Walkway
Hwy','Tempa','FL',90945,'9605257863','smijoh@gmail.com');

INSERT INTO Customer VALUES('CUST_03','Sky','Sharma','08-Mar-1990','F','Hollywood
Walk','LA','CA',99040,'9689980765','skysharma@gmail.com');

INSERT INTO Customer VALUES('CUST_04','Hanna','Williams','26-OCt-1976','F','90
Manhattan Hwy','Brooklyn','NY',98346,'9045095670','hannawilliam@gmail.com');
```

INSERT INTO Customer VALUES('CUST_05','Jwoo','Woo','15-April-1879','M','8901 Norway Fwy','Philadelphia','PA',95110,'9636926365','jwoowoo@gmail.com');

Table Name:Employee

```
DROP TABLE Employee CASCADE CONSTRAINTS;
CREATE TABLE Employee
(
Employee_Id VARCHAR(20) NOT NULL,
Employee_FName VARCHAR2(25) NOT NULL,
Employee_LName VARCHAR2(25) NOT NULL,
Employee_DOB CHAR(30),
Employee_Gender CHAR(20),
Employee_Address VARCHAR2(100),
Employee_City VARCHAR2(50),
Employee_State CHAR(2),
Employee_Zip VARCHAR2(9),
Employee_Phone CHAR(10),
Employee_Email VARCHAR2(256) NOT NULL,
Designation VARCHAR2(15),
Department VARCHAR2(15) NOT NULL,
CONSTRAINT EmployeePK PRIMARY KEY (Employee_Id),
CONSTRAINT Employee_UK_Employee_Email UNIQUE (Employee_Email),
CONSTRAINT Employee_UK_Department UNIQUE (Department),
CONSTRAINT Employee_NN_Employee_FName CHECK (Employee_FName IS NOT NULL),
CONSTRAINT Employee_NN_Employee_LName CHECK (Employee_LName IS NOT NULL)
);
```

Inserting Values into Employee Table:

INSERT INTO Employee VALUES('EMP_01','Pratiksha','Yadav','02-Dec-1992','F','100 Imp Hwy','Norwalk','CA',90243,'5625526365','prat02@gmail.com','Engineer','IT');

INSERT INTO Employee VALUES('EMP_02','Phue','Thant','19-Dec-1991','F','324 Spark Street','Fullerton','CA',92123,'5625567805','phue@gmail.com','Analyst','Finance');

INSERT INTO Employee VALUES('EMP_03','Dhwani','Vaishnav','01-Dec-1990','F','100 Burbank Street','Los Angeles','CA',90283,'5622251111','dhwani@gmail.com','Team Leader','Sales');

INSERT INTO Employee VALUES('EMP_04','Ruta','Antaliya','27-Sep-1995','F','3000 Bear Street','Malibu','CA',98843,'5520000085','ruta@gmail.com','HR','HRM');

INSERT INTO Employee VALUES('EMP_05','Ash','Parhad','05-Nov-1991','M','Adam Street','Long Beach','CA',90443,'5629329705','ash@gmail.com','Manager','Marketing');

Table Name: Orders

DROP TABLE Orders CASCADE CONSTRAINTS;

```
CREATE TABLE Orders
(
Order_ID VARCHAR(10) NOT NULL,
Order_Date DATE,
Order_Category VARCHAR(50) NOT NULL,
Order_Price FLOAT,
Order_Status VARCHAR(20) NOT NULL,
Cust_ID VARCHAR(10) NOT NULL,
Emp_ID VARCHAR(10) NOT NULL,
CONSTRAINT Orders_pk PRIMARY KEY (Order_ID),
CONSTRAINT Orders_Cust_fk FOREIGN KEY (Cust_ID) REFERENCES Customer (Customer_ID),
CONSTRAINT Orders_Emp_fk FOREIGN KEY (Emp_ID) REFERENCES Employee (Employee_ID));
```

Insert Data into the table : Orders

INSERT INTO Orders VALUES ('ORD_01','01-JAN-2','Standard',200.99,'Completed','CUST_01','EMP_05');

INSERT INTO Orders VALUES ('ORD_02', '09-JAN-23', 'Pre-Order', 450.50, 'Delivered', 'CUST_03', 'EMP_01');

INSERT INTO Orders VALUES ('ORD_03', '27-MAR-23', 'Custom Order', 150.99, 'Shipped', 'CUST_02','EMP_02');

INSERT INTO Orders VALUES ('ORD_04', '19-DEC-22', 'Subscription', 150.99, 'Completed', 'CUST_04','EMP_03');

INSERT INTO Orders VALUES ('ORD_05', '27-NOV-22', 'Standard', 480.00, 'Completed', 'CUST_05', 'EMP_04');

Table Name: Product

```
DROP TABLE Product CASCADE CONSTRAINTS;

CREATE TABLE Product
(
Product_ID VARCHAR(10) NOT NULL,
Product_Name VARCHAR(30) NOT NULL,
Product_Category VARCHAR(30) NOT NULL,
Description VARCHAR(50) NOT NULL,
Brand VARCHAR(30) NOT NULL,
Product_Price FLOAT,
Product_Length DECIMAL,
Product_Width DECIMAL,
Product_Height DECIMAL,
Product_Color VARCHAR(15),
Warranty_Period NUMBER,
Product_Type VARCHAR(30),
CONSTRAINT Product_ID_pk PRIMARY KEY (Product_ID) );
```

Insert Data into the table: Product

```
INSERT INTO Product VALUES ('Prod_01', 'PH Smart Lights', 'Smart Lighting', ' Color
Changing LED Lamps', 'Philips Hue', 24.47, 6, 3, 2, 'White', 6, 'Color and Turnable');

INSERT INTO Product VALUES ('Prod_02', 'Google Thermostat', 'Smart Wifi Thermostat', ' Wifi
Thermostat system', 'Nest', 129.99, 3, 3, 1, 'Silver', 12, 'Nest Learning');

INSERT INTO Product VALUES ('Prod_03', 'Echo Show', 'Entertainment', ' Smart Display with
FireTV Built-in', 'Amazon Echo', 279.99, 21, 18, 2, 'Black',24, 'HD Smart Display');

INSERT INTO Product VALUES ('Prod_04', 'Ring Video Bell', 'Home Security', ' Easy
Installation security', 'Ring', 149.99, 2, 4, 1, 'Silver', 12, 'Doorbell');

INSERT INTO Product VALUES ('Prod_05', 'Google Nest Hub', 'Entertainment', ' Your Home at
a glance', 'Google', 249.98, 12, 8, 1, 'Black', 6, 'Nest Hub 2nd Gen');

INSERT INTO Product VALUES ('Prod_06', 'iRobot Roomba i7+ Robot Vacuum', 'Smart Home
Appliances', 'self-emptying robot vacuum', 'iRobot',799.99, 13.34, 13.34, 3.36, 'Charcoal', 12,
'Robot Vacuum');
```

```
INSERT INTO Product VALUES ('Prod_07', 'Arlo Pro 4 Spotlight Camera', 'Smart Security',
'wire-free security camera ', 'Arlo', 199.99, 3.5, 2 , 3, 'Black', 12, 'Security Camera');
```

Table Name: Manufactured_Product

```
DROP TABLE Manufactured_Product;
CREATE TABLE Manufactured_Product
(MProduct_ID VARCHAR(10) NOT NULL,
Production_Site_City VARCHAR(50),
Production_Site_State VARCHAR(2),
Production_Site_ZIP VARCHAR(9) NOT NULL,
Production_Status VARCHAR(30) NOT NULL,
Product_ID VARCHAR(10) NOT NULL,
CONSTRAINT Manufactured_Product_PK PRIMARY KEY (MProduct_ID),
CONSTRAINT Manufactured_Product_FK FOREIGN KEY (Product_ID) REFERENCES
Product (Product_ID));
```

Insert Data into the table: Manufactured_Product

```
INSERT INTO Manufactured_Product
VALUES('MProd_1','California','CA','91505','Shipped','Prod_02');

INSERT INTO Manufactured_Product VALUES('MProd_2','North Carolina','NC','27513','In
Production','Prod_04');
```

Table Name: Purchased_Product

```
DROP TABLE Purchased_Product;

CREATE TABLE Purchased_Product
( PProduct_ID VARCHAR(10) NOT NULL,
Supplier_Name VARCHAR(20) NOT NULL,
Supplier_City VARCHAR(50),
Supplier_State VARCHAR(2),
Supplier_ZIP VARCHAR(9) NOT NULL,
Supplied_Brand VARCHAR(50) NOT NULL,
Supplier_Phone VARCHAR(15),
Supplier_Email VARCHAR(256) NOT NULL,
Supplied_Qty INT NOT NULL,
```

Product_ID VARCHAR(10) NOT NULL,
CONSTRAINT Purchased_Product_PK PRIMARY KEY (PProduct_ID),
CONSTRAINT Purchased_Product_FK FOREIGN KEY (Product_ID) REFERENCES Product (Product_ID));

<u>Insert Data into the table: Purchased_Product:</u>

INSERT INTO Purchased_Product VALUES('PProd_1','Signify N.V.','Beaverton','OR','97005','Phillips','+19998889990','david_johns@phillips.com',100000,'Prod_01');
INSERT INTO Purchased_Product VALUES('PProd_2','Amazon Inc','Hengyang','','411225','Amazon','+868585854545','anna_williams@amazon.cn',50000,'Prod_03');
INSERT INTO Purchased_Product VALUES('PProd_3','Google Nest','Palo Alto','CA','94304','Google','+15554448822','shoun_brown_support@google.com',75000,'Prod_05');

<u>Table Name: OrderItem</u>

DROP TABLE OrderItem CASCADE CONSTRAINTS;

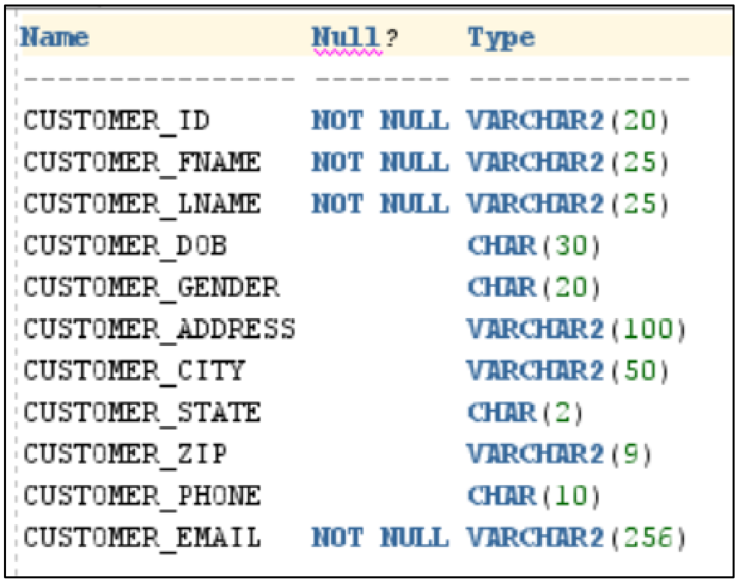
CREATE TABLE OrderItem
(
OrderItem_ID VARCHAR(10) NOT NULL,
Ord_ID VARCHAR(10) NOT NULL,
Prod_ID VARCHAR(10) NOT NULL,
Ordered_Qty NUMBER,
CONSTRAINT OrderItem_pk PRIMARY KEY (OrderItem_ID),
CONSTRAINT OrderItem_Order_fk FOREIGN KEY (Ord_ID) REFERENCES Orders (Order_ID),
CONSTRAINT Orders_ProdID_fk FOREIGN KEY (Prod_ID) REFERENCES Product (Product_ID));

<u>Insert Data into the table : OrderItem</u>

INSERT INTO OrderItem VALUES ('OItem_01','ORD_03','Prod_04',3,'27-MAR-23', NULL);
INSERT INTO OrderItem VALUES ('OItem_02','ORD_05','Prod_05',8,'27-NOV-22', NULL);
INSERT INTO OrderItem VALUES ('OItem_03','ORD_04','Prod_03',5,'19-DEC-22', NULL);
INSERT INTO OrderItem VALUES ('OItem_04','ORD_01','Prod_02',9,'01-JAN-23', NULL);
INSERT INTO OrderItem VALUES ('OItem_05','ORD_02','Prod_01',12,'09-JAN-23', NULL);

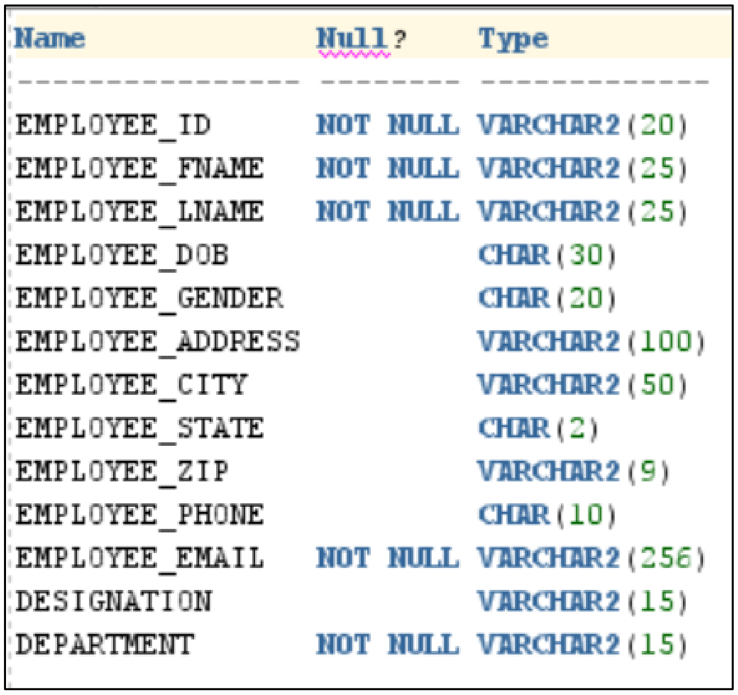
Describing Tables

DESC CUSTOMER;



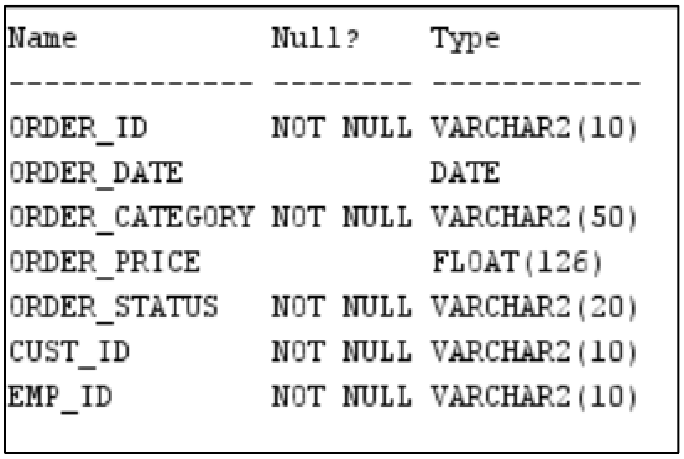
```
Name              Null?    Type
----------------- -------- -------------
CUSTOMER_ID       NOT NULL VARCHAR2(20)
CUSTOMER_FNAME    NOT NULL VARCHAR2(25)
CUSTOMER_LNAME    NOT NULL VARCHAR2(25)
CUSTOMER_DOB               CHAR(30)
CUSTOMER_GENDER            CHAR(20)
CUSTOMER_ADDRESS           VARCHAR2(100)
CUSTOMER_CITY              VARCHAR2(50)
CUSTOMER_STATE             CHAR(2)
CUSTOMER_ZIP               VARCHAR2(9)
CUSTOMER_PHONE             CHAR(10)
CUSTOMER_EMAIL    NOT NULL VARCHAR2(256)
```

DESC EMPLOYEE;



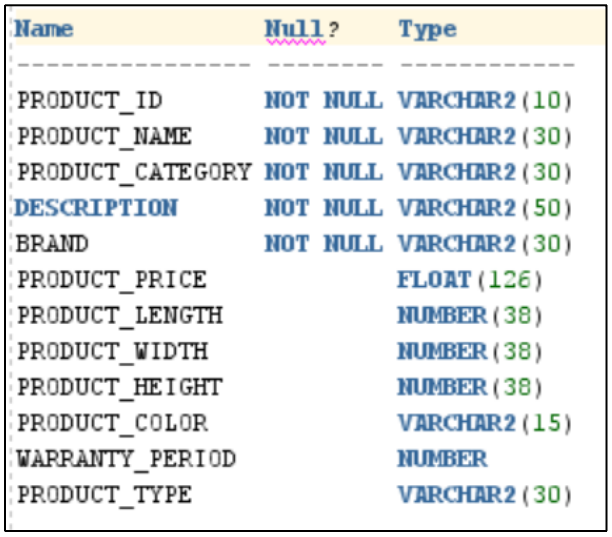
```
Name              Null?    Type
----------------- -------- -------------
EMPLOYEE_ID       NOT NULL VARCHAR2(20)
EMPLOYEE_FNAME    NOT NULL VARCHAR2(25)
EMPLOYEE_LNAME    NOT NULL VARCHAR2(25)
EMPLOYEE_DOB               CHAR(30)
EMPLOYEE_GENDER            CHAR(20)
EMPLOYEE_ADDRESS           VARCHAR2(100)
EMPLOYEE_CITY              VARCHAR2(50)
EMPLOYEE_STATE             CHAR(2)
EMPLOYEE_ZIP               VARCHAR2(9)
EMPLOYEE_PHONE             CHAR(10)
EMPLOYEE_EMAIL    NOT NULL VARCHAR2(256)
DESIGNATION                VARCHAR2(15)
DEPARTMENT        NOT NULL VARCHAR2(15)
```

DESC ORDERS;



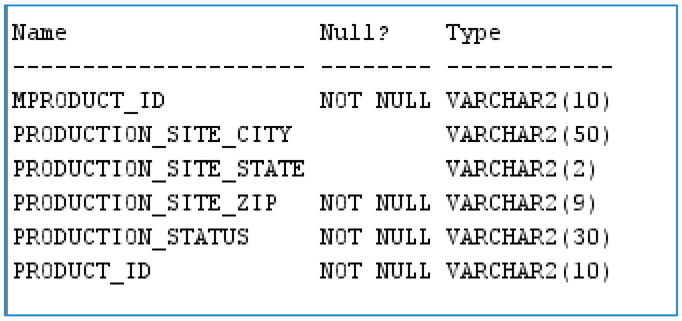
```
Name           Null?    Type
-------------- -------- ------------
ORDER_ID       NOT NULL VARCHAR2(10)
ORDER_DATE              DATE
ORDER_CATEGORY NOT NULL VARCHAR2(50)
ORDER_PRICE             FLOAT(126)
ORDER_STATUS   NOT NULL VARCHAR2(20)
CUST_ID        NOT NULL VARCHAR2(10)
EMP_ID         NOT NULL VARCHAR2(10)
```

DESC PRODUCT;



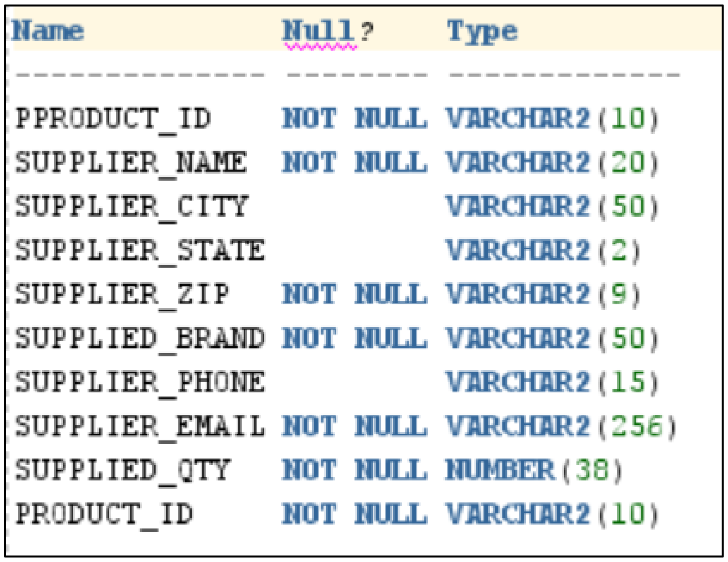
```
Name             Null?    Type
---------------- -------- ------------
PRODUCT_ID       NOT NULL VARCHAR2(10)
PRODUCT_NAME     NOT NULL VARCHAR2(30)
PRODUCT_CATEGORY NOT NULL VARCHAR2(30)
DESCRIPTION      NOT NULL VARCHAR2(50)
BRAND            NOT NULL VARCHAR2(30)
PRODUCT_PRICE             FLOAT(126)
PRODUCT_LENGTH            NUMBER(38)
PRODUCT_WIDTH             NUMBER(38)
PRODUCT_HEIGHT            NUMBER(38)
PRODUCT_COLOR             VARCHAR2(15)
WARRANTY_PERIOD           NUMBER
PRODUCT_TYPE              VARCHAR2(30)
```

DESC MANUFACTURED_PRODUCT;



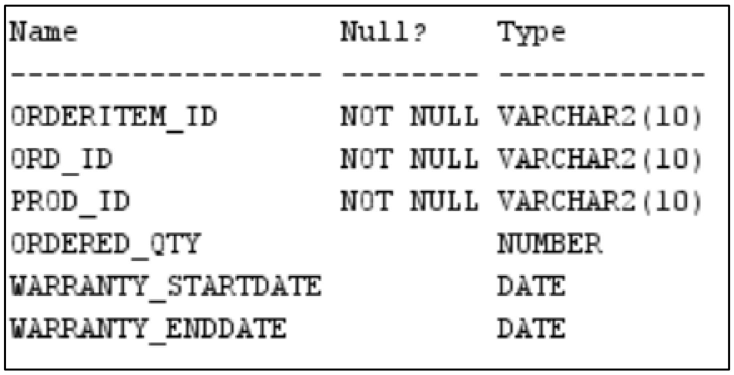
```
Name                  Null?    Type
--------------------- -------- ------------
MPRODUCT_ID           NOT NULL VARCHAR2(10)
PRODUCTION_SITE_CITY           VARCHAR2(50)
PRODUCTION_SITE_STATE          VARCHAR2(2)
PRODUCTION_SITE_ZIP   NOT NULL VARCHAR2(9)
PRODUCTION_STATUS     NOT NULL VARCHAR2(30)
PRODUCT_ID            NOT NULL VARCHAR2(10)
```

DESC PURCHASED_PRODUCT;



```
Name           Null?    Type
-------------- -------- -------------
PPRODUCT_ID    NOT NULL VARCHAR2(10)
SUPPLIER_NAME  NOT NULL VARCHAR2(20)
SUPPLIER_CITY           VARCHAR2(50)
SUPPLIER_STATE          VARCHAR2(2)
SUPPLIER_ZIP   NOT NULL VARCHAR2(9)
SUPPLIED_BRAND NOT NULL VARCHAR2(50)
SUPPLIER_PHONE          VARCHAR2(15)
SUPPLIER_EMAIL NOT NULL VARCHAR2(256)
SUPPLIED_QTY   NOT NULL NUMBER(38)
PRODUCT_ID     NOT NULL VARCHAR2(10)
```

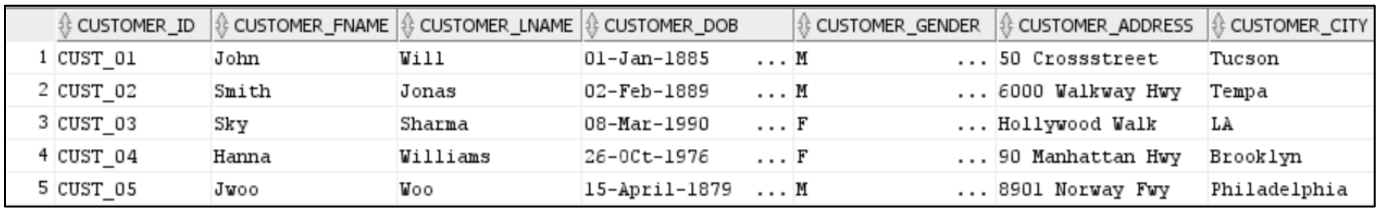
DESC ORDERITEM;



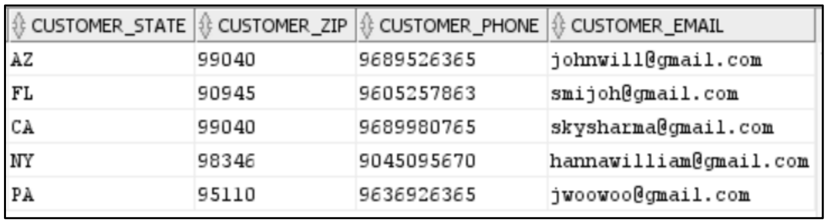
```
Name                  Null?    Type
--------------------- -------- ------------
ORDERITEM_ID          NOT NULL VARCHAR2(10)
ORD_ID                NOT NULL VARCHAR2(10)
PROD_ID               NOT NULL VARCHAR2(10)
ORDERED_QTY                    NUMBER
WARRANTY_STARTDATE             DATE
WARRANTY_ENDDATE               DATE
```

Selecting All from Tables:

SELECT * FROM CUSTOMER;

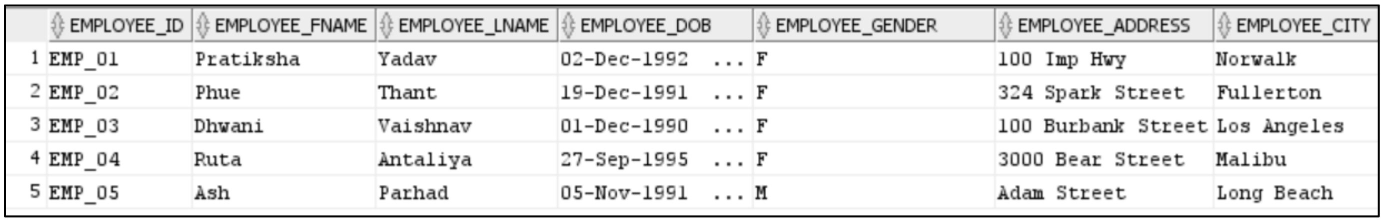


	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_DOB	CUSTOMER_GENDER	CUSTOMER_ADDRESS	CUSTOMER_CITY
1	CUST_01	John	Will	01-Jan-1885 ...	M ...	50 Crossstreet	Tucson
2	CUST_02	Smith	Jonas	02-Feb-1889 ...	M ...	6000 Walkway Hwy	Tempa
3	CUST_03	Sky	Sharma	08-Mar-1990 ...	F ...	Hollywood Walk	LA
4	CUST_04	Hanna	Williams	26-OCt-1976 ...	F ...	90 Manhattan Hwy	Brooklyn
5	CUST_05	Jwoo	Woo	15-April-1879 ...	M ...	8901 Norway Fwy	Philadelphia

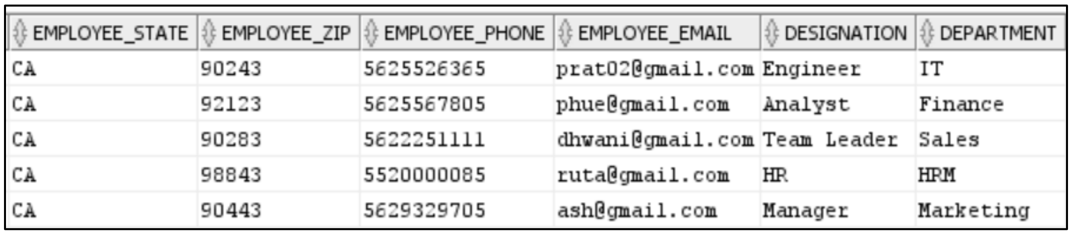


CUSTOMER_STATE	CUSTOMER_ZIP	CUSTOMER_PHONE	CUSTOMER_EMAIL
AZ	99040	9689526365	johnwill@gmail.com
FL	90945	9605257863	smijoh@gmail.com
CA	99040	9689980765	skysharma@gmail.com
NY	98346	9045095670	hannawilliam@gmail.com
PA	95110	9636926365	jwoowoo@gmail.com

SELECT * FROM EMPLOYEE;

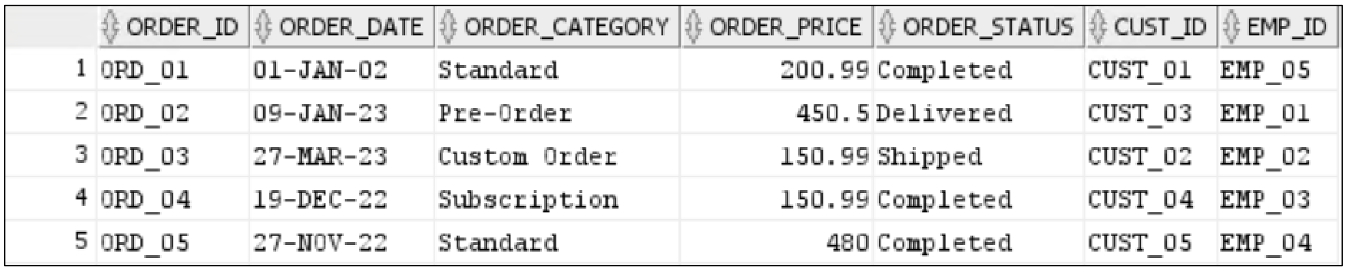


	EMPLOYEE_ID	EMPLOYEE_FNAME	EMPLOYEE_LNAME	EMPLOYEE_DOB	EMPLOYEE_GENDER	EMPLOYEE_ADDRESS	EMPLOYEE_CITY
1	EMP_01	Pratiksha	Yadav	02-Dec-1992 ...	F	100 Imp Hwy	Norwalk
2	EMP_02	Phue	Thant	19-Dec-1991 ...	F	324 Spark Street	Fullerton
3	EMP_03	Dhwani	Vaishnav	01-Dec-1990 ...	F	100 Burbank Street	Los Angeles
4	EMP_04	Ruta	Antaliya	27-Sep-1995 ...	F	3000 Bear Street	Malibu
5	EMP_05	Ash	Parhad	05-Nov-1991 ...	M	Adam Street	Long Beach



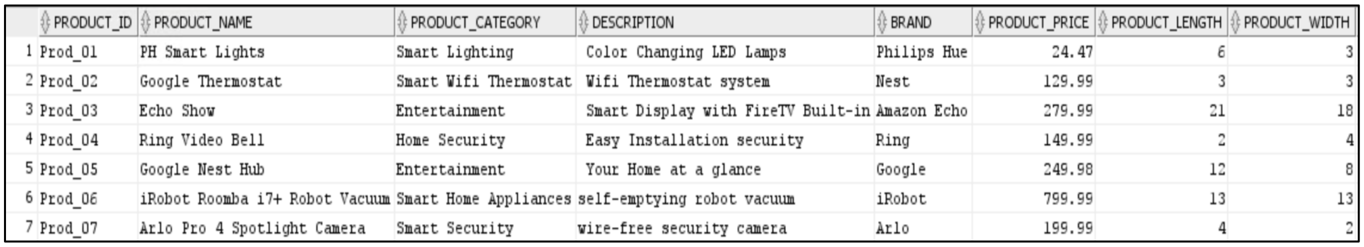
EMPLOYEE_STATE	EMPLOYEE_ZIP	EMPLOYEE_PHONE	EMPLOYEE_EMAIL	DESIGNATION	DEPARTMENT
CA	90243	5625526365	prat02@gmail.com	Engineer	IT
CA	92123	5625567805	phue@gmail.com	Analyst	Finance
CA	90283	5622251111	dhwani@gmail.com	Team Leader	Sales
CA	98843	5520000085	ruta@gmail.com	HR	HRM
CA	90443	5629329705	ash@gmail.com	Manager	Marketing

SELECT * FROM ORDERS;

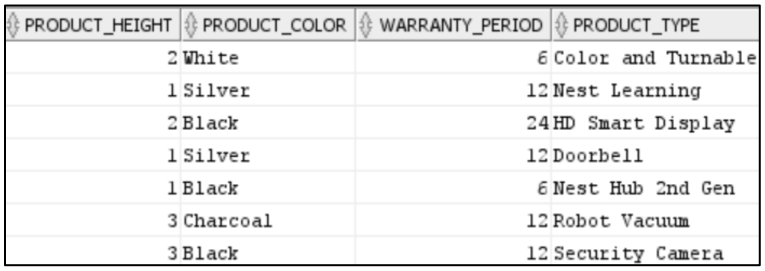


	ORDER_ID	ORDER_DATE	ORDER_CATEGORY	ORDER_PRICE	ORDER_STATUS	CUST_ID	EMP_ID
1	ORD_01	01-JAN-02	Standard	200.99	Completed	CUST_01	EMP_05
2	ORD_02	09-JAN-23	Pre-Order	450.5	Delivered	CUST_03	EMP_01
3	ORD_03	27-MAR-23	Custom Order	150.99	Shipped	CUST_02	EMP_02
4	ORD_04	19-DEC-22	Subscription	150.99	Completed	CUST_04	EMP_03
5	ORD_05	27-NOV-22	Standard	480	Completed	CUST_05	EMP_04

SELECT * FROM PRODUCT;

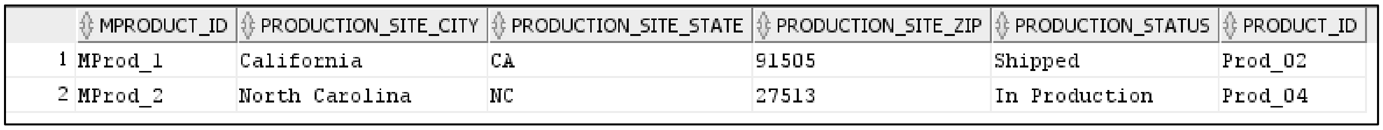


	PRODUCT_ID	PRODUCT_NAME	PRODUCT_CATEGORY	DESCRIPTION	BRAND	PRODUCT_PRICE	PRODUCT_LENGTH	PRODUCT_WIDTH
1	Prod_01	PH Smart Lights	Smart Lighting	Color Changing LED Lamps	Philips Hue	24.47	6	3
2	Prod_02	Google Thermostat	Smart Wifi Thermostat	Wifi Thermostat system	Nest	129.99	3	3
3	Prod_03	Echo Show	Entertainment	Smart Display with FireTV Built-in	Amazon Echo	279.99	21	18
4	Prod_04	Ring Video Bell	Home Security	Easy Installation security	Ring	149.99	2	4
5	Prod_05	Google Nest Hub	Entertainment	Your Home at a glance	Google	249.98	12	8
6	Prod_06	iRobot Roomba i7+ Robot Vacuum	Smart Home Appliances	self-emptying robot vacuum	iRobot	799.99	13	13
7	Prod_07	Arlo Pro 4 Spotlight Camera	Smart Security	wire-free security camera	Arlo	199.99	4	2



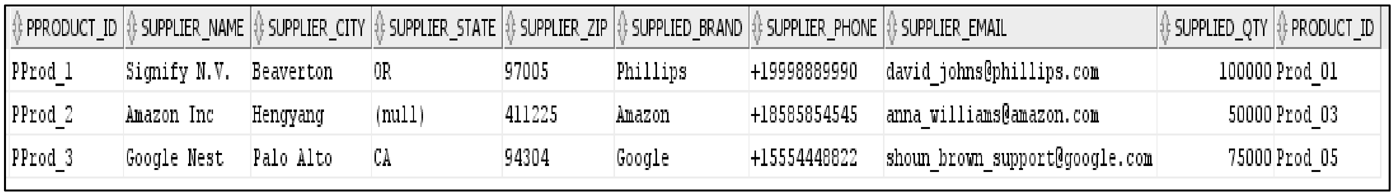
PRODUCT_HEIGHT	PRODUCT_COLOR	WARRANTY_PERIOD	PRODUCT_TYPE
2	White	6	Color and Turnable
1	Silver	12	Nest Learning
2	Black	24	HD Smart Display
1	Silver	12	Doorbell
1	Black	6	Nest Hub 2nd Gen
3	Charcoal	12	Robot Vacuum
3	Black	12	Security Camera

SELECT * FROM MANUFACTURED_PRODUCT;



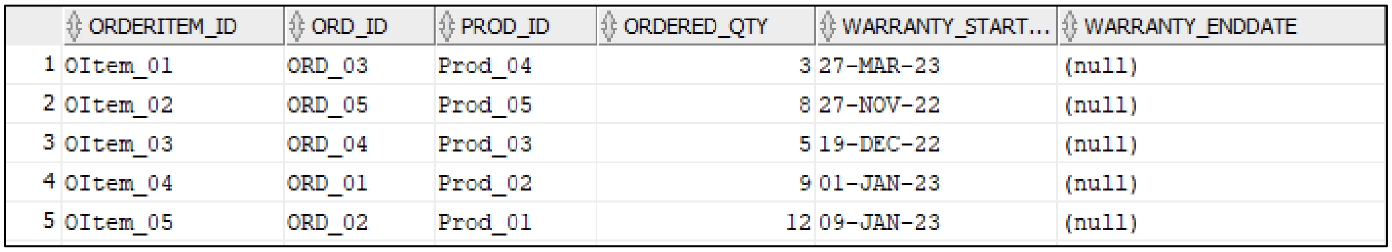
	MPRODUCT_ID	PRODUCTION_SITE_CITY	PRODUCTION_SITE_STATE	PRODUCTION_SITE_ZIP	PRODUCTION_STATUS	PRODUCT_ID
1	MProd_1	California	CA	91505	Shipped	Prod_02
2	MProd_2	North Carolina	NC	27513	In Production	Prod_04

SELECT * FROM PURCHASED_PRODUCT;



PPRODUCT_ID	SUPPLIER_NAME	SUPPLIER_CITY	SUPPLIER_STATE	SUPPLIER_ZIP	SUPPLIED_BRAND	SUPPLIER_PHONE	SUPPLIER_EMAIL	SUPPLIED_QTY	PRODUCT_ID
PProd_1	Signify N.V.	Beaverton	OR	97005	Phillips	+19998889990	david_johns@phillips.com	100000	Prod_01
PProd_2	Amazon Inc	Hengyang	(null)	411225	Amazon	+18585854545	anna_williams@amazon.com	50000	Prod_03
PProd_3	Google Nest	Palo Alto	CA	94304	Google	+15554448822	shoun_brown_support@google.com	75000	Prod_05

SELECT * FROM ORDERITEM;



	ORDERITEM_ID	ORD_ID	PROD_ID	ORDERED_QTY	WARRANTY_START...	WARRANTY_ENDDATE
1	OItem_01	ORD_03	Prod_04	3	27-MAR-23	(null)
2	OItem_02	ORD_05	Prod_05	8	27-NOV-22	(null)
3	OItem_03	ORD_04	Prod_03	5	19-DEC-22	(null)
4	OItem_04	ORD_01	Prod_02	9	01-JAN-23	(null)
5	OItem_05	ORD_02	Prod_01	12	09-JAN-23	(null)

Performing Insert, Update, Delete, Create Views

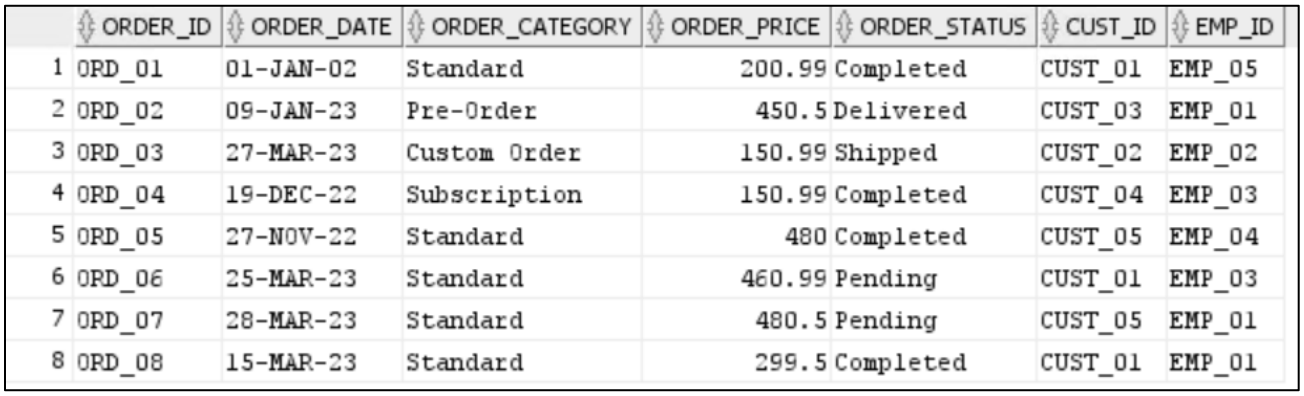
INSERT Values

Inserting records to Orders table

INSERT INTO Orders VALUES ('ORD_06','25-Mar-23','Standard
',460.99,'Pending','CUST_01','EMP_03');

INSERT INTO Orders VALUES ('ORD_07', '28-MAR-23', 'Standard', 480.50,
'Pending','CUST_05', 'EMP_01');

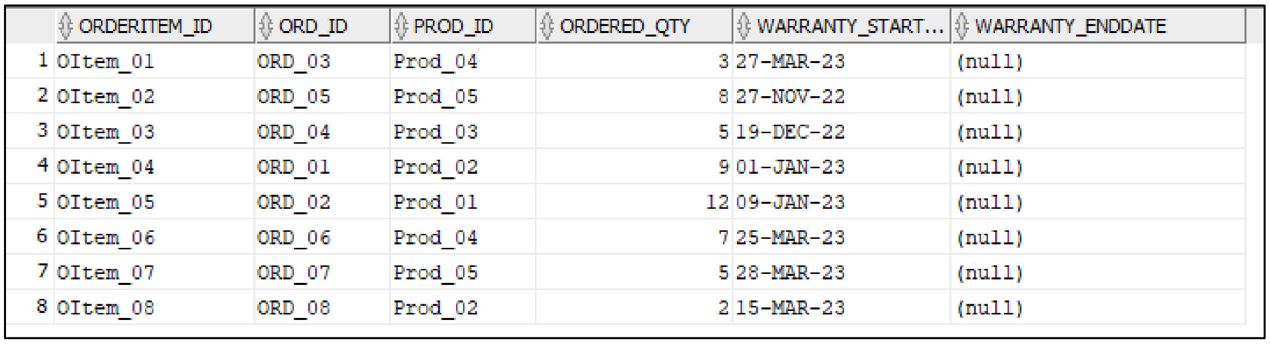
INSERT INTO Orders VALUES ('ORD_08', '15-MAR-23', 'Standard', 299.50, 'Completed','CUST_01', 'EMP_01');



	ORDER_ID	ORDER_DATE	ORDER_CATEGORY	ORDER_PRICE	ORDER_STATUS	CUST_ID	EMP_ID
1	ORD_01	01-JAN-02	Standard	200.99	Completed	CUST_01	EMP_05
2	ORD_02	09-JAN-23	Pre-Order	450.5	Delivered	CUST_03	EMP_01
3	ORD_03	27-MAR-23	Custom Order	150.99	Shipped	CUST_02	EMP_02
4	ORD_04	19-DEC-22	Subscription	150.99	Completed	CUST_04	EMP_03
5	ORD_05	27-NOV-22	Standard	480	Completed	CUST_05	EMP_04
6	ORD_06	25-MAR-23	Standard	460.99	Pending	CUST_01	EMP_03
7	ORD_07	28-MAR-23	Standard	480.5	Pending	CUST_05	EMP_01
8	ORD_08	15-MAR-23	Standard	299.5	Completed	CUST_01	EMP_01

Inserting records to Orderitem table

INSERT INTO OrderItem VALUES ('OItem_06','ORD_06','Prod_04',7,'25-MAR-23', NULL);
INSERT INTO OrderItem VALUES ('OItem_07','ORD_07','Prod_05',5, '28-MAR-23', NULL);
INSERT INTO OrderItem VALUES ('OItem_08','ORD_08','Prod_02',2, '15-MAR-23', NULL);



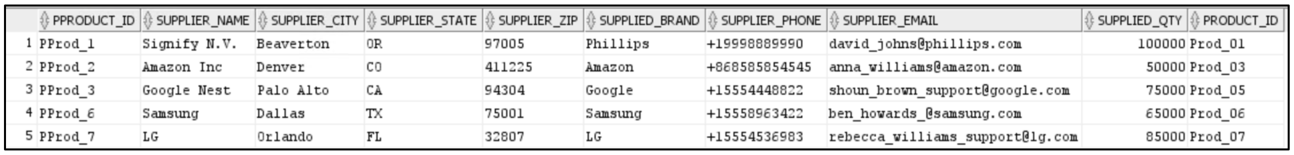
	ORDERITEM_ID	ORD_ID	PROD_ID	ORDERED_QTY	WARRANTY_START...	WARRANTY_ENDDATE
1	OItem_01	ORD_03	Prod_04	3	27-MAR-23	(null)
2	OItem_02	ORD_05	Prod_05	8	27-NOV-22	(null)
3	OItem_03	ORD_04	Prod_03	5	19-DEC-22	(null)
4	OItem_04	ORD_01	Prod_02	9	01-JAN-23	(null)
5	OItem_05	ORD_02	Prod_01	12	09-JAN-23	(null)
6	OItem_06	ORD_06	Prod_04	7	25-MAR-23	(null)
7	OItem_07	ORD_07	Prod_05	5	28-MAR-23	(null)
8	OItem_08	ORD_08	Prod_02	2	15-MAR-23	(null)

Inserting records to purchased_product table

INSERT INTO Purchased_Product VALUES('PProd_6','Samsung','Dallas','TX','75001','Samsung','+15558963422','ben_howards_@samsung.com',65000,'Prod_06');

INSERT INTO Purchased_Product VALUES('PProd_7','LG','Orlando','FL','32807','LG','+15554536983','rebecca_williams_support@lg.com',85000,'Prod_07');

SELECT * FROM purchased_product;

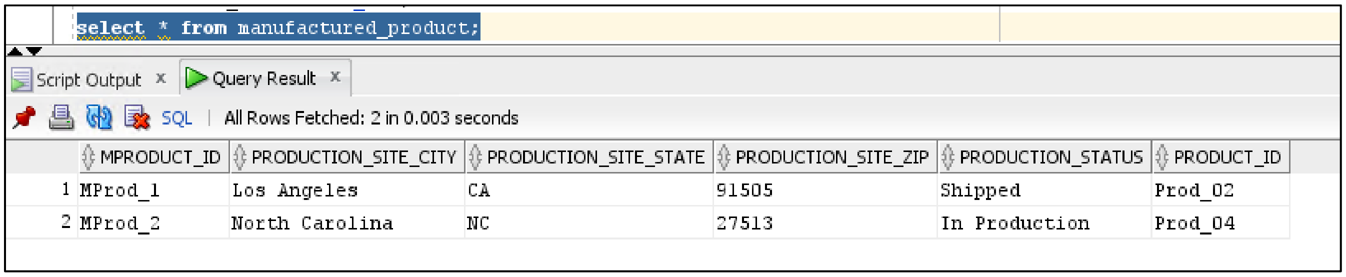


	PPRODUCT_ID	SUPPLIER_NAME	SUPPLIER_CITY	SUPPLIER_STATE	SUPPLIER_ZIP	SUPPLIED_BRAND	SUPPLIER_PHONE	SUPPLIER_EMAIL	SUPPLIED_QTY	PRODUCT_ID
1	PProd_1	Signify N.V.	Beaverton	OR	97005	Phillips	+19998889990	david_johns@phillips.com	100000	Prod_01
2	PProd_2	Amazon Inc	Denver	CO	411225	Amazon	+868585854545	anna_williams@amazon.com	50000	Prod_03
3	PProd_3	Google Nest	Palo Alto	CA	94304	Google	+15554448822	shoun_brown_support@google.com	75000	Prod_05
4	PProd_6	Samsung	Dallas	TX	75001	Samsung	+15558963422	ben_howards_@samsung.com	65000	Prod_06
5	PProd_7	LG	Orlando	FL	32807	LG	+15554536983	rebecca_williams_support@lg.com	85000	Prod_07

UPDATE values

Table: MANUFACTURED_PRODUCT; updating Manufacturing city to “Los Angeles” for the product having product id = Prod_02

UPDATE MANUFACTURED_PRODUCT SET production_site_city = 'Los Angeles'
WHERE PRODUCT_ID = 'Prod_02';



```
select * from manufactured_product;
```

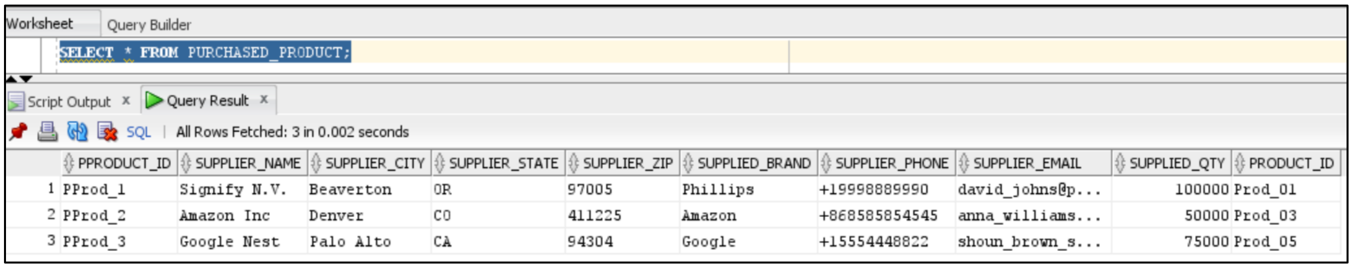
Script Output × Query Result ×

SQL | All Rows Fetched: 2 in 0.003 seconds

	MPRODUCT_ID	PRODUCTION_SITE_CITY	PRODUCTION_SITE_STATE	PRODUCTION_SITE_ZIP	PRODUCTION_STATUS	PRODUCT_ID
1	MProd_1	Los Angeles	CA	91505	Shipped	Prod_02
2	MProd_2	North Carolina	NC	27513	In Production	Prod_04

Table: PURCHASED_PRODUCT; updating Supplier city & State to Denver, CO for the product having pproduct_id = PProd_2

UPDATE PURCHASED_PRODUCT SET Supplier_City = 'Denver', Supplier_State = 'CO'
WHERE PProduct_ID = 'PProd_2';



Worksheet Query Builder

```
SELECT * FROM PURCHASED_PRODUCT;
```

Script Output × Query Result ×

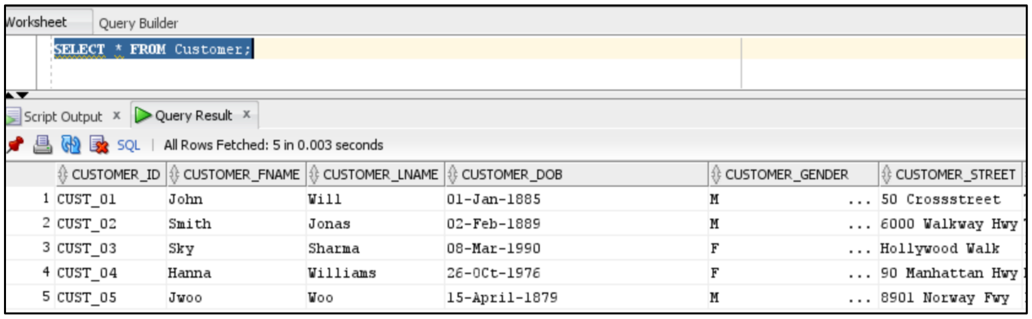
SQL | All Rows Fetched: 3 in 0.002 seconds

	PPRODUCT_ID	SUPPLIER_NAME	SUPPLIER_CITY	SUPPLIER_STATE	SUPPLIER_ZIP	SUPPLIED_BRAND	SUPPLIER_PHONE	SUPPLIER_EMAIL	SUPPLIED_QTY	PRODUCT_ID
1	PProd_1	Signify N.V.	Beaverton	OR	97005	Phillips	+19998889990	david_johns@p...	100000	Prod_01
2	PProd_2	Amazon Inc	Denver	CO	411225	Amazon	+868585854545	anna_williams...	50000	Prod_03
3	PProd_3	Google Nest	Palo Alto	CA	94304	Google	+15554448822	shoun_brown_s...	75000	Prod_05

ALTER Table

Table: Customer; altering table for column name from Customer_Address to Customer_Street:

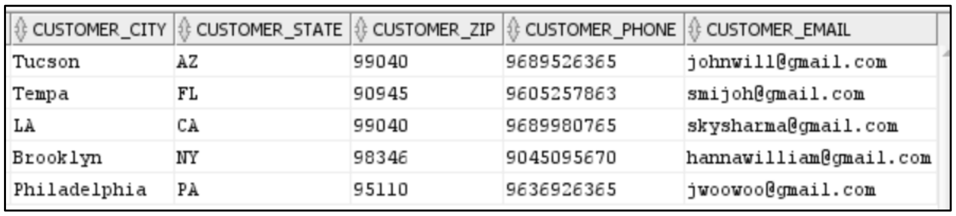
ALTER TABLE Customer
RENAME COLUMN CUSTOMER_Address TO Customer_Street;



```
SELECT * FROM Customer;
```

All Rows Fetched: 5 in 0.003 seconds

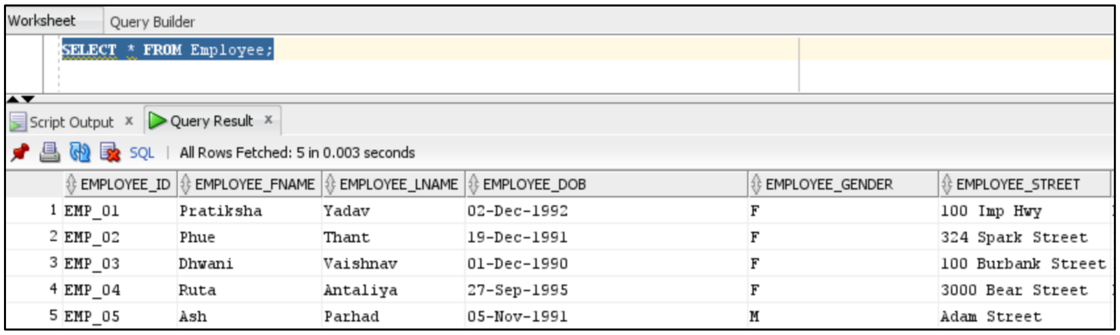
	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_DOB	CUSTOMER_GENDER	CUSTOMER_STREET
1	CUST_01	John	Will	01-Jan-1885	M	... 50 Crossstreet
2	CUST_02	Smith	Jonas	02-Feb-1889	M	... 6000 Walkway Hwy
3	CUST_03	Sky	Sharma	08-Mar-1990	F	... Hollywood Walk
4	CUST_04	Hanna	Williams	26-OCt-1976	F	... 90 Manhattan Hwy
5	CUST_05	Jwoo	Woo	15-April-1879	M	... 8901 Norway Fwy



CUSTOMER_CITY	CUSTOMER_STATE	CUSTOMER_ZIP	CUSTOMER_PHONE	CUSTOMER_EMAIL
Tucson	AZ	99040	9689526365	johnwill@gmail.com
Tempa	FL	90945	9605257863	smijoh@gmail.com
LA	CA	99040	9689980765	skysharma@gmail.com
Brooklyn	NY	98346	9045095670	hannawilliam@gmail.com
Philadelphia	PA	95110	9636926365	jwoowoo@gmail.com

Table: Employee; altering table for column name from Employee_Address to Employee_Street:

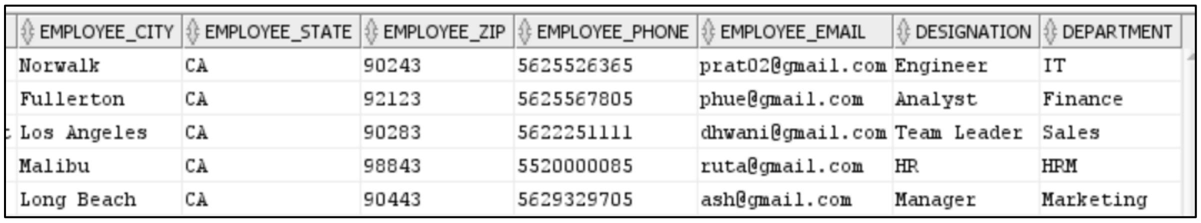
ALTER TABLE Employee
RENAME COLUMN Employee_Address TO Employee_Street;



```
SELECT * FROM Employee;
```

All Rows Fetched: 5 in 0.003 seconds

	EMPLOYEE_ID	EMPLOYEE_FNAME	EMPLOYEE_LNAME	EMPLOYEE_DOB	EMPLOYEE_GENDER	EMPLOYEE_STREET
1	EMP_01	Pratiksha	Yadav	02-Dec-1992	F	100 Imp Hwy
2	EMP_02	Phue	Thant	19-Dec-1991	F	324 Spark Street
3	EMP_03	Dhwani	Vaishnav	01-Dec-1990	F	100 Burbank Street
4	EMP_04	Ruta	Antaliya	27-Sep-1995	F	3000 Bear Street
5	EMP_05	Ash	Parhad	05-Nov-1991	M	Adam Street

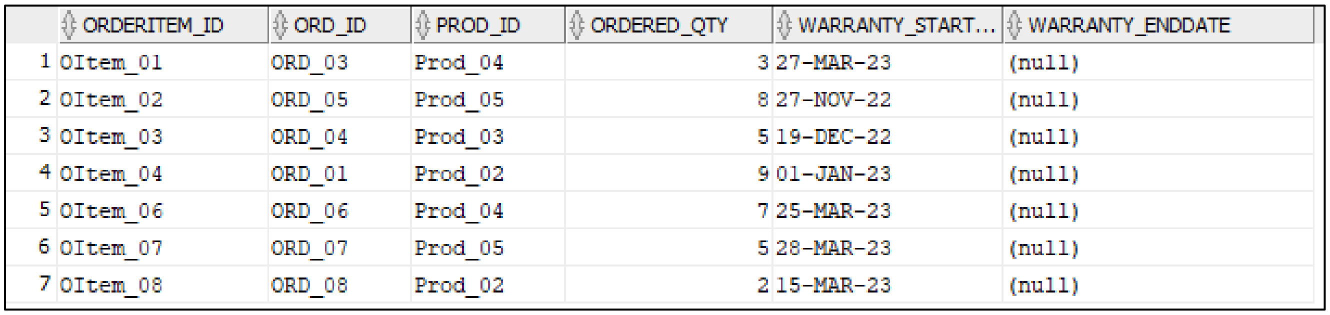


EMPLOYEE_CITY	EMPLOYEE_STATE	EMPLOYEE_ZIP	EMPLOYEE_PHONE	EMPLOYEE_EMAIL	DESIGNATION	DEPARTMENT
Norwalk	CA	90243	5625526365	prat02@gmail.com	Engineer	IT
Fullerton	CA	92123	5625567805	phue@gmail.com	Analyst	Finance
Los Angeles	CA	90283	5622251111	dhwani@gmail.com	Team Leader	Sales
Malibu	CA	98843	5520000085	ruta@gmail.com	HR	HRM
Long Beach	CA	90443	5629329705	ash@gmail.com	Manager	Marketing

DELETE Values

Table: OrderItem; deleting order item having order item id 'OItem_05'

DELETE FROM OrderItem WHERE OrderItem_ID = 'OItem_05';



	ORDERITEM_ID	ORD_ID	PROD_ID	ORDERED_QTY	WARRANTY_START...	WARRANTY_ENDDATE
1	OItem_01	ORD_03	Prod_04	3	27-MAR-23	(null)
2	OItem_02	ORD_05	Prod_05	8	27-NOV-22	(null)
3	OItem_03	ORD_04	Prod_03	5	19-DEC-22	(null)
4	OItem_04	ORD_01	Prod_02	9	01-JAN-23	(null)
5	OItem_06	ORD_06	Prod_04	7	25-MAR-23	(null)
6	OItem_07	ORD_07	Prod_05	5	28-MAR-23	(null)
7	OItem_08	ORD_08	Prod_02	2	15-MAR-23	(null)

Create View

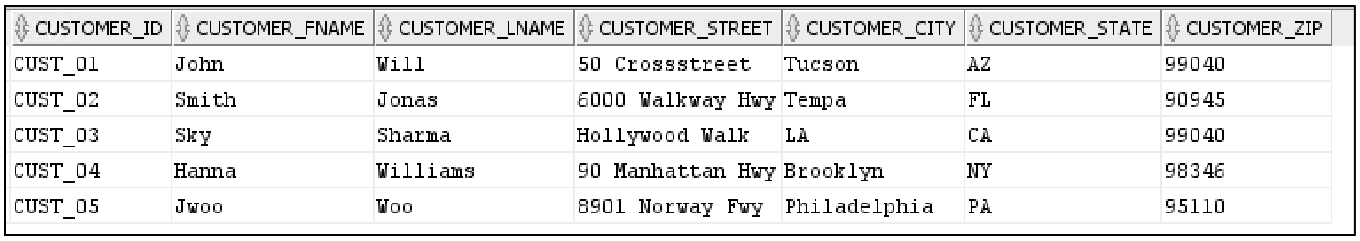
Creating a view for Customer Address

CREATE VIEW Customer_Address_View(CUSTOMER_ID, CUSTOMER_FNAME, CUSTOMER_LNAME, CUSTOMER_STREET, CUSTOMER_CITY, CUSTOMER_STATE, CUSTOMER_ZIP)

AS

SELECT CUSTOMER_ID, CUSTOMER_FNAME, CUSTOMER_LNAME, CUSTOMER_STREET, CUSTOMER_CITY, CUSTOMER_STATE, CUSTOMER_ZIP

FROM CUSTOMER;



CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_STREET	CUSTOMER_CITY	CUSTOMER_STATE	CUSTOMER_ZIP
CUST_01	John	Will	50 Crossstreet	Tucson	AZ	99040
CUST_02	Smith	Jonas	6000 Walkway Hwy	Tempa	FL	90945
CUST_03	Sky	Sharma	Hollywood Walk	LA	CA	99040
CUST_04	Hanna	Williams	90 Manhattan Hwy	Brooklyn	NY	98346
CUST_05	Jwoo	Woo	8901 Norway Fwy	Philadelphia	PA	95110

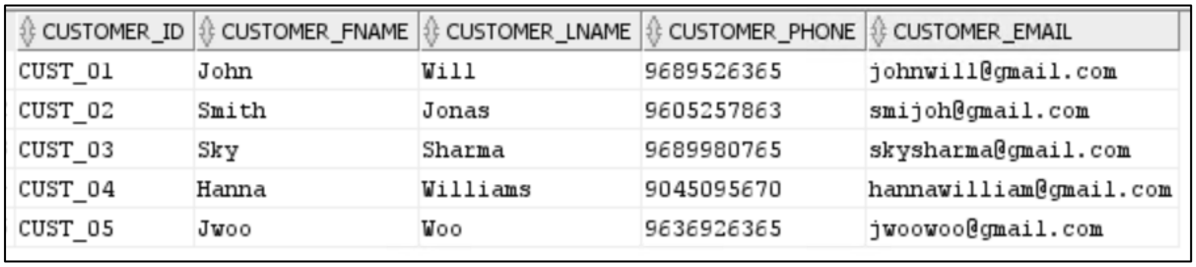
Creating a view for Customer Contact

CREATE VIEW Customer_Contact_View(CUSTOMER_ID, CUSTOMER_FNAME, CUSTOMER_LNAME, CUSTOMER_PHONE, CUSTOMER_EMAIL)

AS

SELECT CUSTOMER_ID, CUSTOMER_FNAME, CUSTOMER_LNAME, CUSTOMER_PHONE, CUSTOMER_EMAIL

FROM CUSTOMER;



CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_PHONE	CUSTOMER_EMAIL
CUST_01	John	Will	9689526365	johnwill@gmail.com
CUST_02	Smith	Jonas	9605257863	smijoh@gmail.com
CUST_03	Sky	Sharma	9689980765	skysharma@gmail.com
CUST_04	Hanna	Williams	9045095670	hannawilliam@gmail.com
CUST_05	Jwoo	Woo	9636926365	jwoowoo@gmail.com

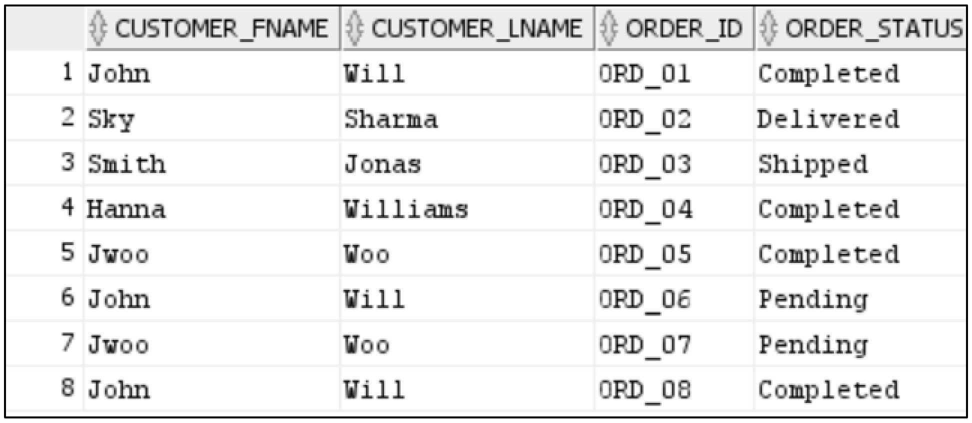
Creating a View of Customer Order Status

CREATE VIEW Order_Status_By_Customer_View AS

SELECT Customer.Customer_Fname, Customer_Lname, Orders.Order_id, Orders.Order_Status FROM Customer, Orders

WHERE Customer.Customer_ID = Orders.Cust_ID;

Select * FROM Order_Status_By_Customer_View;

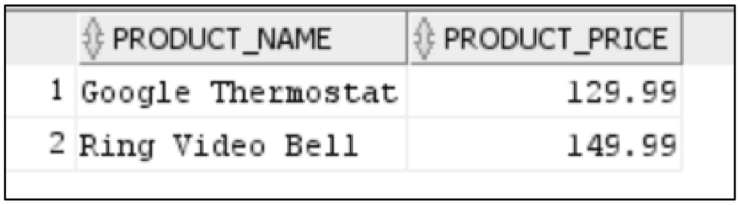


	CUSTOMER_FNAME	CUSTOMER_LNAME	ORDER_ID	ORDER_STATUS
1	John	Will	ORD_01	Completed
2	Sky	Sharma	ORD_02	Delivered
3	Smith	Jonas	ORD_03	Shipped
4	Hanna	Williams	ORD_04	Completed
5	Jwoo	Woo	ORD_05	Completed
6	John	Will	ORD_06	Pending
7	Jwoo	Woo	ORD_07	Pending
8	John	Will	ORD_08	Completed

Testing Database with (Select, join, where, group by, having) Queries

SELECT
List all the Products of the company that have a price between $100 and $200.

SELECT Product_Name, Product_Price
FROM Product
WHERE Product_Price BETWEEN 100 AND 200;



	PRODUCT_NAME	PRODUCT_PRICE
1	Google Thermostat	129.99
2	Ring Video Bell	149.99

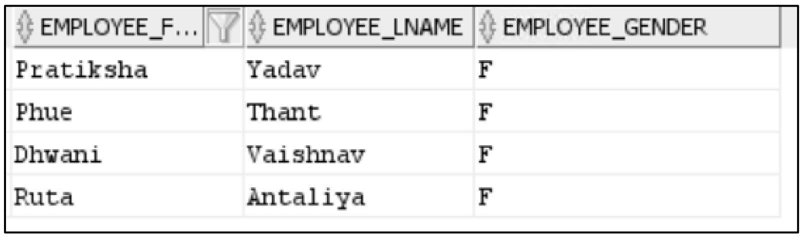
WHERE

Retrieve the First Name & Last Name of all Female Employees:

SELECT Employee_FName, Employee_LName, Employee_Gender

FROM Employee

WHERE Employee_Gender = 'F';



EMPLOYEE_F...	EMPLOYEE_LNAME	EMPLOYEE_GENDER
Pratiksha	Yadav	F
Phue	Thant	F
Dhwani	Vaishnav	F
Ruta	Antaliya	F

Retrieve Product Id, brand, supplier name, supplier's city & states and supplied quantity for the product supplied from the state 'California'

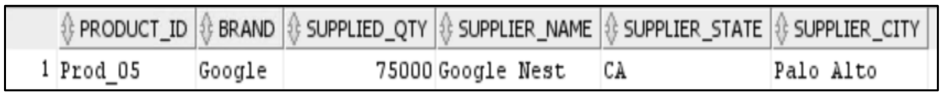
Select Product.Product_ID, Product.Brand, Purchased_Product.Supplied_Qty, Purchased_Product.Supplier_Name, Purchased_Product.Supplier_State, Purchased_Product.Supplier_City

From Product

LEFT JOIN Purchased_Product

ON Product.Product_ID = Purchased_Product.Product_ID

WHERE Supplier_State = 'CA';

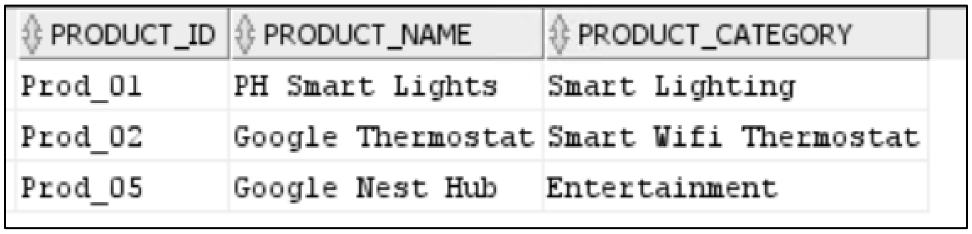


	PRODUCT_ID	BRAND	SUPPLIED_QTY	SUPPLIER_NAME	SUPPLIER_STATE	SUPPLIER_CITY
1	Prod_05	Google	75000	Google Nest	CA	Palo Alto

SUBQUERY

Retrieve the names and categories of products that have been ordered in quantities greater than 5

```
SELECT Product_id, Product_name, Product_category
FROM Product
WHERE Product_ID IN
(SELECT Prod_ID
FROM Orderitem
WHERE Orderitem.Ordered_Qty > 5);
```

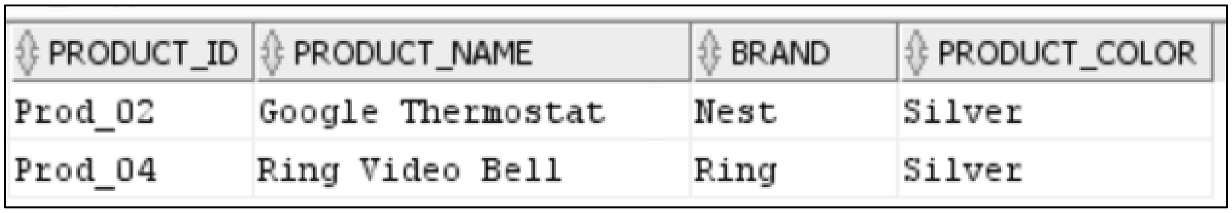


PRODUCT_ID	PRODUCT_NAME	PRODUCT_CATEGORY
Prod_01	PH Smart Lights	Smart Lighting
Prod_02	Google Thermostat	Smart Wifi Thermostat
Prod_05	Google Nest Hub	Entertainment

JOIN

List the Manufactured Products which have Silver Color.

```
SELECT Product.Product_ID, Product.Product_Name, Product.Brand,Product.Product_Color
FROM Product
INNER JOIN Manufactured_Product ON
Product.Product_ID = Manufactured_Product.Product_ID
WHERE Product_Color = 'Silver';
```

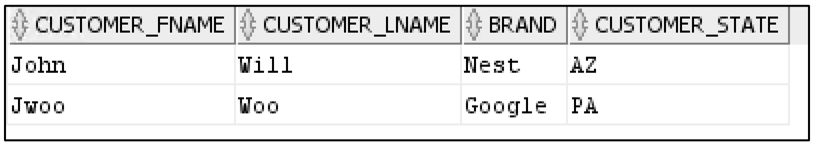


PRODUCT_ID	PRODUCT_NAME	BRAND	PRODUCT_COLOR
Prod_02	Google Thermostat	Nest	Silver
Prod_04	Ring Video Bell	Ring	Silver

Retrieve customers details (first name, last name & state) who have ordered google & its sub-brand.

```
SELECT c.Customer_FName, c.Customer_LName, p.Brand, c.Customer_State
FROM Customer c JOIN Orders o
```

ON c.Customer_ID = o.Cust_id
JOIN Orderitem oi
ON oi.Ord_id = o.Order_id
JOIN Product p
ON p.Product_id = oi.Prod_id
WHERE p.Brand IN ('Google', 'Nest')
ORDER BY p.Brand DESC;



CUSTOMER_FNAME	CUSTOMER_LNAME	BRAND	CUSTOMER_STATE
John	Will	Nest	AZ
Jwoo	Woo	Google	PA

Retrieve Product Details and Supplier Information for Products with Supplied Quantity of 50,000 or More

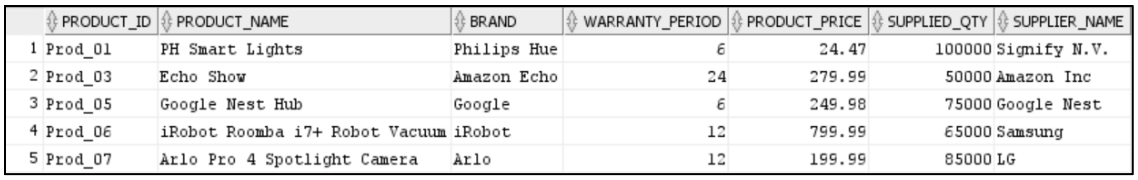
Select Product.Product_ID, Product.Product_Name, Product.Brand, Product.Warranty_Period, Product.Product_Price, Purchased_Product.Supplied_Qty, Purchased_Product.Supplier_Name

From Product

LEFT JOIN Purchased_Product

ON Product.Product_ID = Purchased_Product.Product_ID

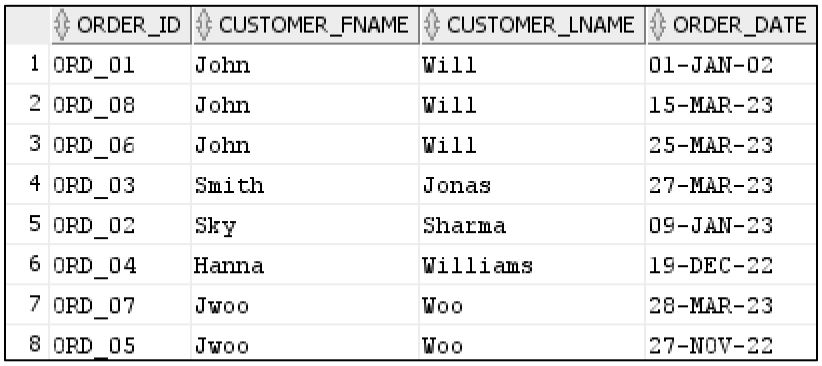
WHERE Supplied_Qty >= 50000;



	PRODUCT_ID	PRODUCT_NAME	BRAND	WARRANTY_PERIOD	PRODUCT_PRICE	SUPPLIED_QTY	SUPPLIER_NAME
1	Prod_01	PH Smart Lights	Philips Hue	6	24.47	100000	Signify N.V.
2	Prod_03	Echo Show	Amazon Echo	24	279.99	50000	Amazon Inc
3	Prod_05	Google Nest Hub	Google	6	249.98	75000	Google Nest
4	Prod_06	iRobot Roomba i7+ Robot Vacuum	iRobot	12	799.99	65000	Samsung
5	Prod_07	Arlo Pro 4 Spotlight Camera	Arlo	12	199.99	85000	LG

Retrieve Order Details and Customer Information for all Orders Placed

Select Orders.Order_ID, Customer.Customer_Fname, Customer_Lname, Orders.Order_Date
FROM Orders
INNER JOIN Customer ON Orders.Cust_ID=Customer.Customer_ID;

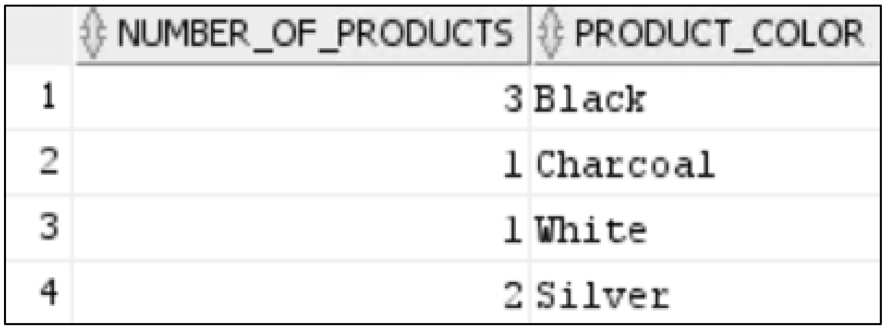


	ORDER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	ORDER_DATE
1	ORD_01	John	Will	01-JAN-02
2	ORD_08	John	Will	15-MAR-23
3	ORD_06	John	Will	25-MAR-23
4	ORD_03	Smith	Jonas	27-MAR-23
5	ORD_02	Sky	Sharma	09-JAN-23
6	ORD_04	Hanna	Williams	19-DEC-22
7	ORD_07	Jwoo	Woo	28-MAR-23
8	ORD_05	Jwoo	Woo	27-NOV-22

GROUP BY, HAVING

Retrieve number of products with same colors

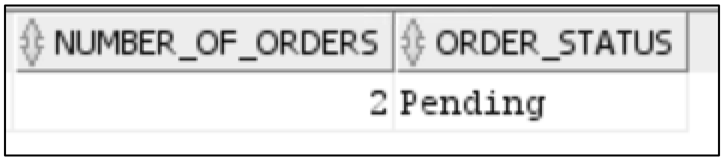
SELECT COUNT(Product_ID) As Number_of_Products, Product_Color
FROM Product
GROUP BY Product_Color;



	NUMBER_OF_PRODUCTS	PRODUCT_COLOR
1	3	Black
2	1	Charcoal
3	1	White
4	2	Silver

How many Orders have the Pending Order Status?

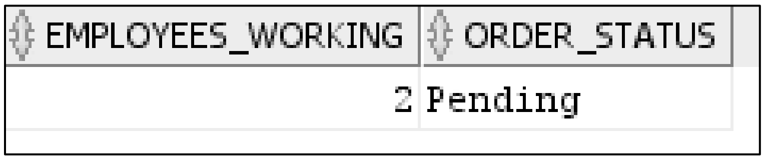
SELECT COUNT(Order_ID) AS Number_of_Orders , Order_Status
FROM Orders
GROUP BY Order_Status
HAVING Order_Status = 'Pending';



NUMBER_OF_ORDERS	ORDER_STATUS
2	Pending

How many employees are currently working on the "Pending" orders?

```
SELECT DISTINCT COUNT(Emp_ID) as Employees_Working, order_status
FROM orders
GROUP BY order_status
HAVING order_status = 'Pending';
```



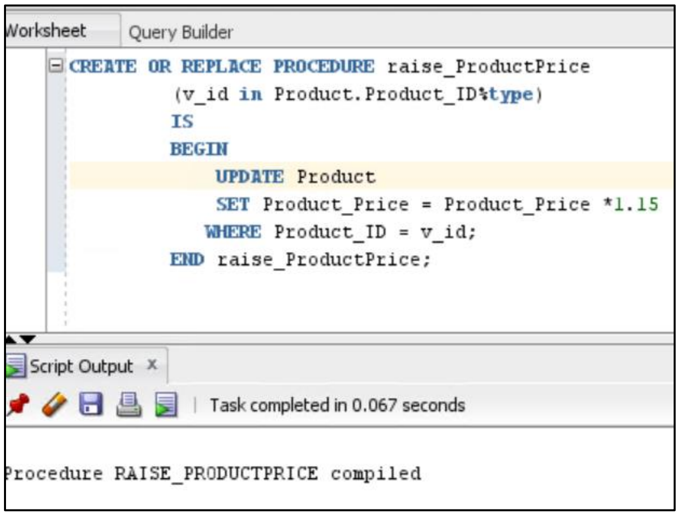
EMPLOYEES_WORKING	ORDER_STATUS
2	Pending

PL/SQL Procedures & Functions

PRODECURES

Raise the product's price by 15%.

```
CREATE OR REPLACE PROCEDURE raise_ProductPrice
        (v_id in Product.Product_ID%type)
        IS
        BEGIN
           UPDATE Product
           SET Product_Price = Product_Price *1.15
           WHERE Product_ID = v_id;
        END raise_ProductPrice;
```



Worksheet Query Builder

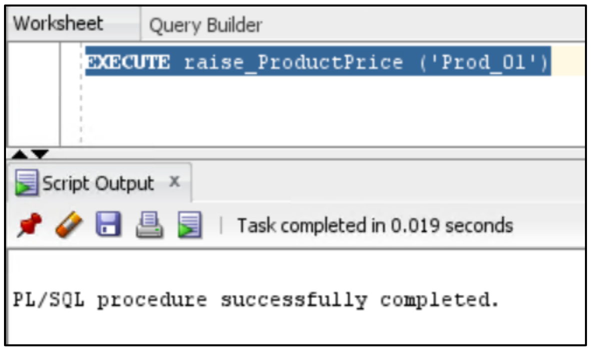
```
CREATE OR REPLACE PROCEDURE raise_ProductPrice
        (v_id in Product.Product_ID%type)
        IS
        BEGIN
            UPDATE Product
            SET Product_Price = Product_Price *1.15
           WHERE Product_ID = v_id;
        END raise_ProductPrice;
```

Script Output

Task completed in 0.067 seconds

```
Procedure RAISE_PRODUCTPRICE compiled
```

EXECUTE raise_ProductPrice ('Prod_01')



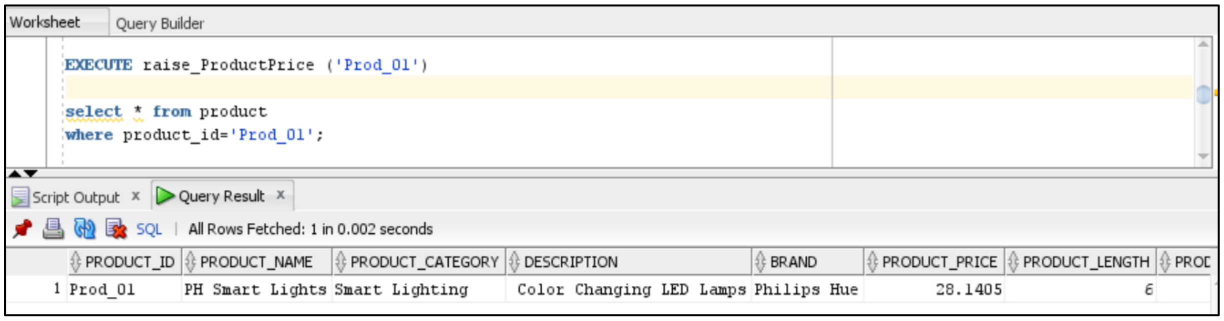
Worksheet Query Builder

```
EXECUTE raise_ProductPrice ('Prod_01')
```

Script Output

Task completed in 0.019 seconds

```
PL/SQL procedure successfully completed.
```



Worksheet Query Builder

```
EXECUTE raise_ProductPrice ('Prod_01')

select * from product
where product_id='Prod_01';
```

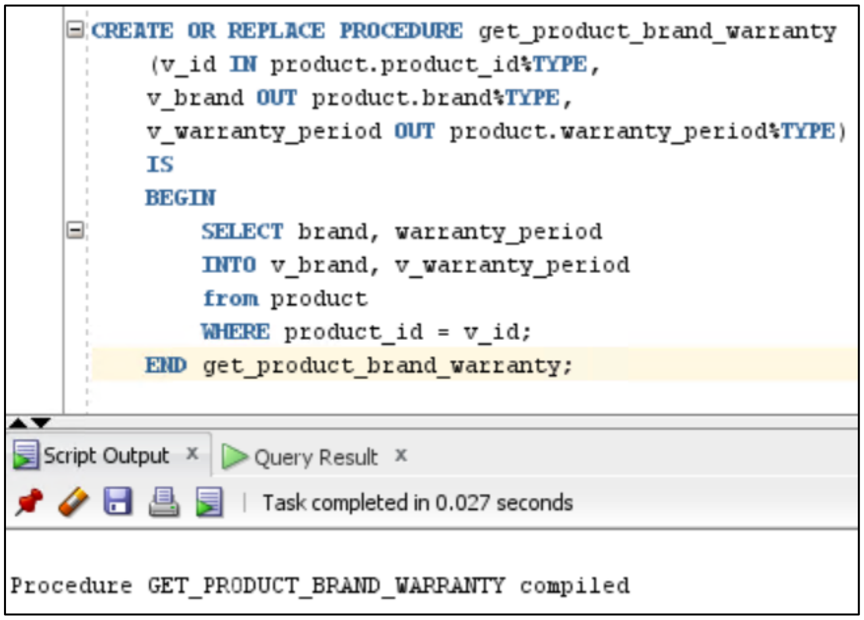
Script Output Query Result

All Rows Fetched: 1 in 0.002 seconds

	PRODUCT_ID	PRODUCT_NAME	PRODUCT_CATEGORY	DESCRIPTION	BRAND	PRODUCT_PRICE	PRODUCT_LENGTH	PROD
1	Prod_01	PH Smart Lights	Smart Lighting	Color Changing LED Lamps	Philips Hue	28.1405	6	

Creating a Procedure with IN and OUT parameters to Retrieve the brand and warranty information for the product Id and store them in the product_brand and product_warranty variables, respectively.

```
CREATE OR REPLACE PROCEDURE get_product_brand_warranty
 (v_id IN product.product_id%TYPE,
 v_brand OUT product.brand%TYPE,
 v_warranty_period OUT product.warranty_period%TYPE)
 IS
 BEGIN
     SELECT brand, warranty_period
     INTO v_brand, v_warranty_period
     from product
     WHERE product_id = v_id;
 END get_product_brand_warranty;
```



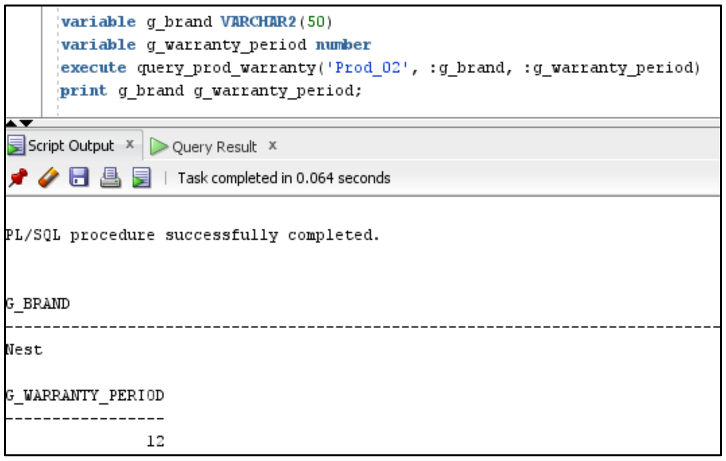
```
CREATE OR REPLACE PROCEDURE get_product_brand_warranty
    (v_id IN product.product_id%TYPE,
    v_brand OUT product.brand%TYPE,
    v_warranty_period OUT product.warranty_period%TYPE)
    IS
    BEGIN
        SELECT brand, warranty_period
        INTO v_brand, v_warranty_period
        from product
        WHERE product_id = v_id;
    END get_product_brand_warranty;
```

Task completed in 0.027 seconds

```
Procedure GET_PRODUCT_BRAND_WARRANTY compiled
```

Calling the above procedure:

```
variable g_brand VARCHAR2(50)
variable g_warranty_period number
execute query_prod_warranty('Prod_02', :g_brand, :g_warranty_period)
print g_brand g_warranty_period
```



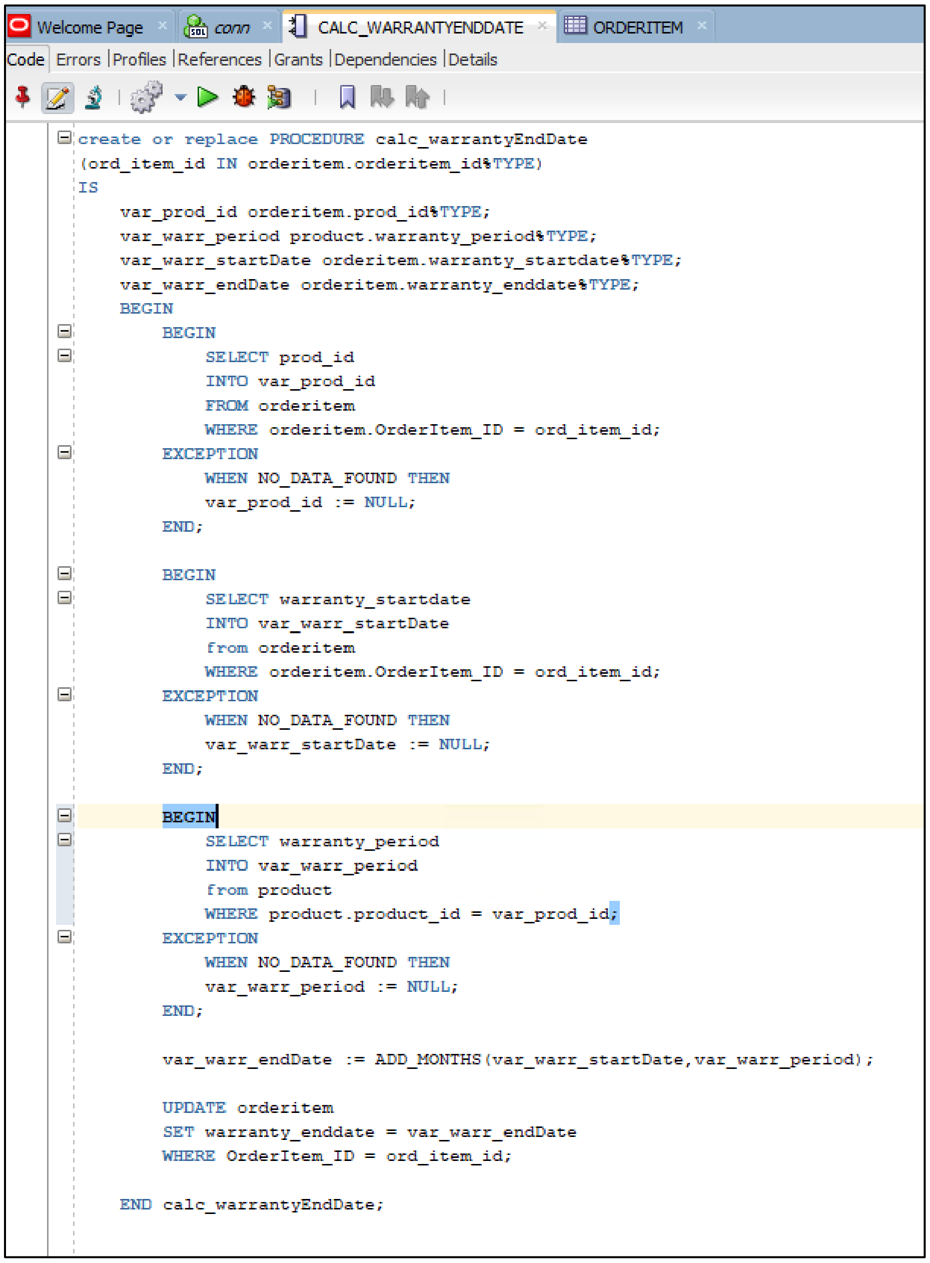
```
variable g_brand VARCHAR2(50)
variable g_warranty_period number
execute query_prod_warranty('Prod_02', :g_brand, :g_warranty_period)
print g_brand g_warranty_period;
```

```
PL/SQL procedure successfully completed.


G_BRAND
--------------------------------------------------------------------------------
Nest

G_WARRANTY_PERIOD
-----------------
               12
```

A Procedure to update Warranty_Enddate in Orderitem Table

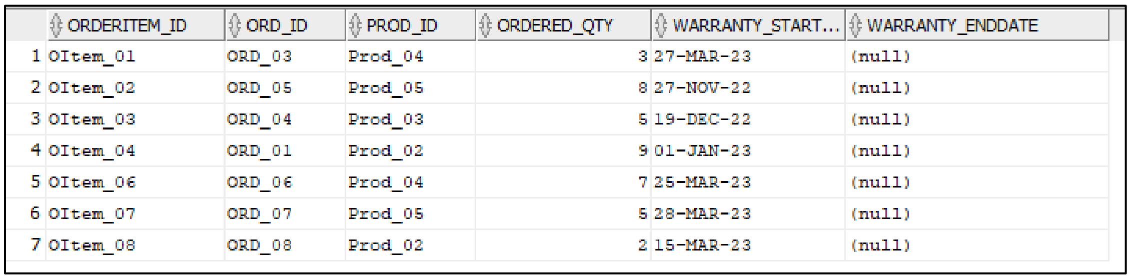
```
CREATE OR REPLACE PROCEDURE calc_warrantyEndDate
(ord_item_id IN orderitem.orderitem_id%TYPE)
IS
   var_prod_id orderitem.prod_id%TYPE;
   var_warr_period product.warranty_period%TYPE;
   var_warr_startDate orderitem.warranty_startdate%TYPE;
   var_warr_endDate orderitem.warranty_enddate%TYPE;
   BEGIN
      BEGIN
         SELECT prod_id
         INTO var_prod_id
         FROM orderitem
         WHERE orderitem.OrderItem_ID = ord_item_id;
      EXCEPTION
         WHEN NO_DATA_FOUND THEN
         var_prod_id := NULL;
      END;

      BEGIN
         SELECT warranty_startdate
         INTO var_warr_startDate
         from orderitem
         WHERE orderitem.OrderItem_ID = ord_item_id;
      EXCEPTION
         WHEN NO_DATA_FOUND THEN
         var_warr_startDate := NULL;
```

```
    END;

    BEGIN
        SELECT warranty_period
        INTO var_warr_period
        from product
        WHERE product.product_id = var_prod_id;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
        var_warr_period := NULL;
    END;

    var_warr_endDate := ADD_MONTHS(var_warr_startDate,var_warr_period);

    UPDATE orderitem
    SET warranty_enddate = var_warr_endDate
    WHERE OrderItem_ID = ord_item_id;

END calc_warrantyEndDate;
```


Welcome Page
conn
CALC_WARRANTYENDDATE
ORDERITEM
Code | Errors | Profiles | References | Grants | Dependencies | Details

```
create or replace PROCEDURE calc_warrantyEndDate
(ord_item_id IN orderitem.orderitem_id%TYPE)
IS
    var_prod_id orderitem.prod_id%TYPE;
    var_warr_period product.warranty_period%TYPE;
    var_warr_startDate orderitem.warranty_startdate%TYPE;
    var_warr_endDate orderitem.warranty_enddate%TYPE;
    BEGIN
        BEGIN
            SELECT prod_id
            INTO var_prod_id
            FROM orderitem
            WHERE orderitem.OrderItem_ID = ord_item_id;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
            var_prod_id := NULL;
        END;

        BEGIN
            SELECT warranty_startdate
            INTO var_warr_startDate
            from orderitem
            WHERE orderitem.OrderItem_ID = ord_item_id;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
            var_warr_startDate := NULL;
        END;

        BEGIN
            SELECT warranty_period
            INTO var_warr_period
            from product
            WHERE product.product_id = var_prod_id;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
            var_warr_period := NULL;
        END;

        var_warr_endDate := ADD_MONTHS(var_warr_startDate,var_warr_period);

        UPDATE orderitem
        SET warranty_enddate = var_warr_endDate
        WHERE OrderItem_ID = ord_item_id;

    END calc_warrantyEndDate;
```

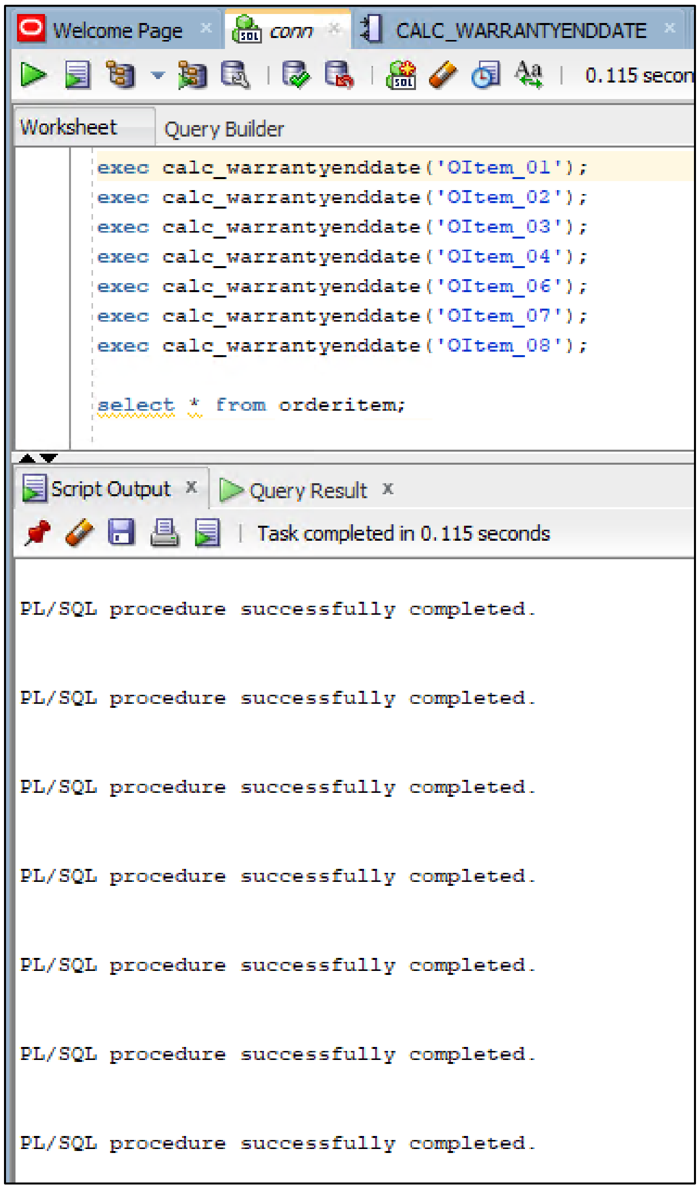
Table before executing the procedure:



	ORDERITEM_ID	ORD_ID	PROD_ID	ORDERED_QTY	WARRANTY_START...	WARRANTY_ENDDATE
1	OItem_01	ORD_03	Prod_04	3	27-MAR-23	(null)
2	OItem_02	ORD_05	Prod_05	8	27-NOV-22	(null)
3	OItem_03	ORD_04	Prod_03	5	19-DEC-22	(null)
4	OItem_04	ORD_01	Prod_02	9	01-JAN-23	(null)
5	OItem_06	ORD_06	Prod_04	7	25-MAR-23	(null)
6	OItem_07	ORD_07	Prod_05	5	28-MAR-23	(null)
7	OItem_08	ORD_08	Prod_02	2	15-MAR-23	(null)

Execution of the procedure:

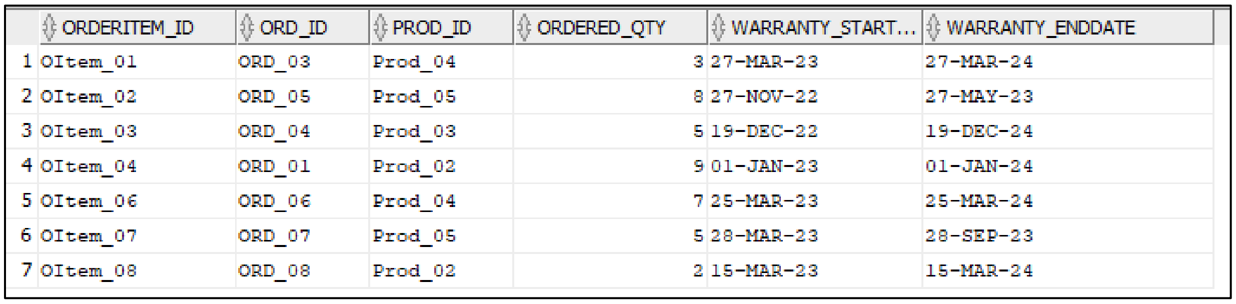
exec calc_warrantyenddate('OItem_01');
exec calc_warrantyenddate('OItem_02');
exec calc_warrantyenddate('OItem_03');
exec calc_warrantyenddate('OItem_04');
exec calc_warrantyenddate('OItem_06');
exec calc_warrantyenddate('OItem_07');
exec calc_warrantyenddate('OItem_08');



```
exec calc_warrantyenddate('OItem_01');
exec calc_warrantyenddate('OItem_02');
exec calc_warrantyenddate('OItem_03');
exec calc_warrantyenddate('OItem_04');
exec calc_warrantyenddate('OItem_06');
exec calc_warrantyenddate('OItem_07');
exec calc_warrantyenddate('OItem_08');

select * from orderitem;
```

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

Output table after execution:

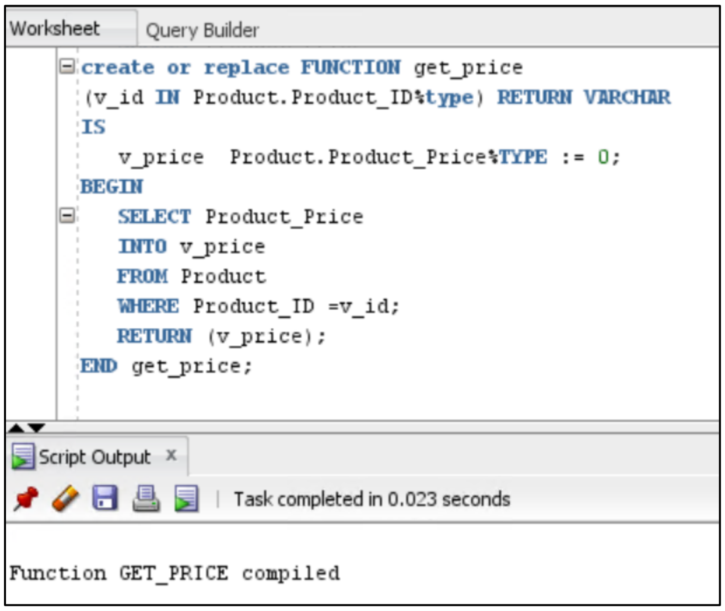


	ORDERITEM_ID	ORD_ID	PROD_ID	ORDERED_QTY	WARRANTY_START...	WARRANTY_ENDDATE
1	OItem_01	ORD_03	Prod_04	3	27-MAR-23	27-MAR-24
2	OItem_02	ORD_05	Prod_05	8	27-NOV-22	27-MAY-23
3	OItem_03	ORD_04	Prod_03	5	19-DEC-22	19-DEC-24
4	OItem_04	ORD_01	Prod_02	9	01-JAN-23	01-JAN-24
5	OItem_06	ORD_06	Prod_04	7	25-MAR-23	25-MAR-24
6	OItem_07	ORD_07	Prod_05	5	28-MAR-23	28-SEP-23
7	OItem_08	ORD_08	Prod_02	2	15-MAR-23	15-MAR-24

<u>FUNCTIONS</u>

Get Product PRICE by a given PRODUCT_ID

```
create or replace FUNCTION get_price
(v_id IN Product.Product_ID%type) RETURN VARCHAR
IS
  v_price  Product.Product_Price%TYPE := 0;
BEGIN
  SELECT Product_Price
  INTO v_price
  FROM Product
  WHERE Product_ID =v_id;
  RETURN (v_price);
END get_price;
```

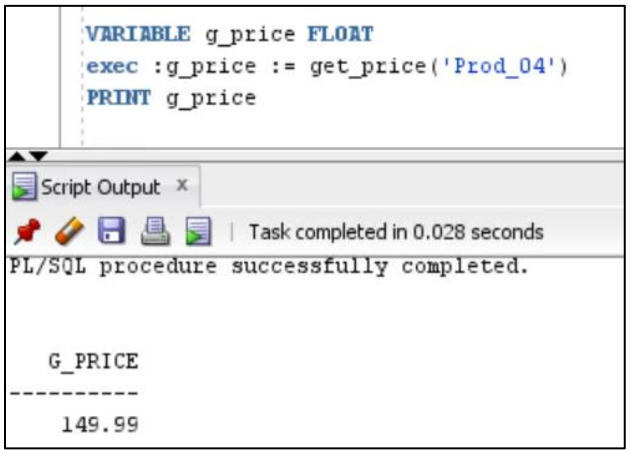


```
create or replace FUNCTION get_price
(v_id IN Product.Product_ID%type) RETURN VARCHAR
IS
   v_price  Product.Product_Price%TYPE := 0;
BEGIN
   SELECT Product_Price
   INTO v_price
   FROM Product
   WHERE Product_ID =v_id;
   RETURN (v_price);
END get_price;
```

Task completed in 0.023 seconds

```
Function GET_PRICE compiled
```

```
VARIABLE g_price FLOAT
exec :g_price := get_price('Prod_04')

PRINT g_price
```

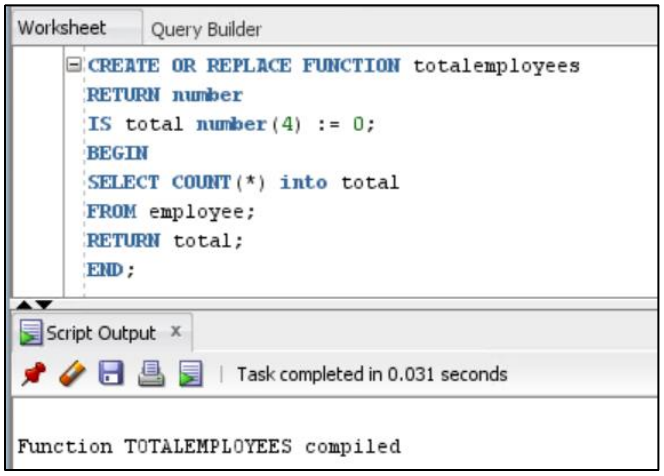


```
VARIABLE g_price FLOAT
exec :g_price := get_price('Prod_04')
PRINT g_price
```

```
PL/SQL procedure successfully completed.


   G_PRICE
----------
    149.99
```

Creating a Function to Count total number of employees:

```
CREATE OR REPLACE FUNCTION totalemployees
RETURN number
IS total number(4) := 0;
BEGIN
SELECT COUNT(*) into total
FROM employee;
RETURN total;
END;
```

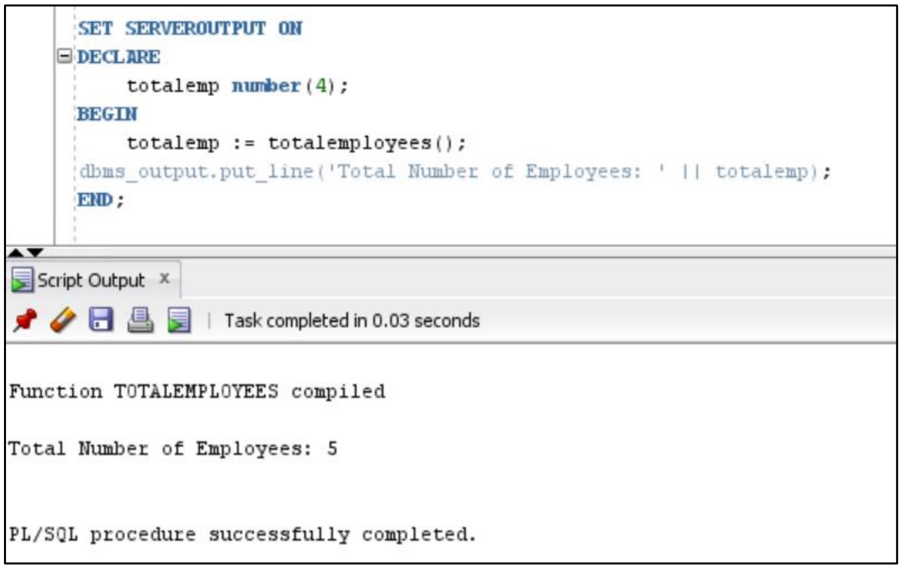


```
CREATE OR REPLACE FUNCTION totalemployees
RETURN number
IS total number(4) := 0;
BEGIN
SELECT COUNT(*) into total
FROM employee;
RETURN total;
END;
```

```
Function TOTALEMPLOYEES compiled
```

Calling the above Function

```
SET SERVEROUTPUT ON
DECLARE
   totalemp number(4);
BEGIN
   totalemp := totalemployees();
dbms_output.put_line('Total Number of Employees: ' || totalemp);
END;
```



```
SET SERVEROUTPUT ON
DECLARE
    totalemp number(4);
BEGIN
    totalemp := totalemployees();
dbms_output.put_line('Total Number of Employees: ' || totalemp);
END;
```

```
Function TOTALEMPLOYEES compiled

Total Number of Employees: 5


PL/SQL procedure successfully completed.
```

ORDBMS

Address Type Object:

```
CREATE OR REPLACE TYPE Address_ty_new AS Object

(
   street  Varchar2(50),
   City    varchar2(25),
   State   char(2),
   Zip     number
);
```

Contact Type Object:

```
CREATE OR REPLACE TYPE Contact_ty_new AS Object

(
   Email   Varchar2(256),
   Phone   Char(12)
);
```

Full Name Type Object:

```
CREATE OR REPLACE TYPE Name_ty_new AS Object

(
   FName   CHAR(20),
   LName   CHAR(20)
);
```

Demographic details Object with member function:

```
CREATE OR REPLACE TYPE Demographic_ty AS OBJECT
(
   Gender  CHAR(10),
   DOB DATE,
   MEMBER FUNCTION age(DOB DATE) RETURN Number
);

CREATE OR REPLACE TYPE BODY demographic_ty AS
MEMBER FUNCTION age(DOB DATE)
RETURN Number IS
Begin
    RETURN ROUND(ABS(MONTHS_BETWEEN(SYSDATE, DOB)/12),0);
End age;
```

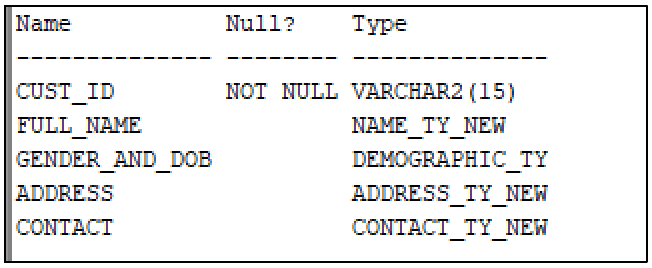
End;

Creating new table using Address_ty_new & Contact_ty_new: Customer_New

```
CREATE TABLE Customer_New
(
   Cust_ID VARCHAR(15),
   Full_Name Name_ty_new,
   Gender_and_DOB Demographic_ty,
   Address Address_ty_new,
   Contact Contact_ty_new,
   CONSTRAINT Cust_id_pk PRIMARY KEY(Cust_ID)
);
```

Describing the table: Customer_New

DESCRIBE Customer_New;



```
Name           Null?    Type
-------------- -------- --------------
CUST_ID        NOT NULL VARCHAR2(15)
FULL_NAME               NAME_TY_NEW
GENDER_AND_DOB          DEMOGRAPHIC_TY
ADDRESS                 ADDRESS_TY_NEW
CONTACT                 CONTACT_TY_NEW
```

Inserting data into the table: Customer _New

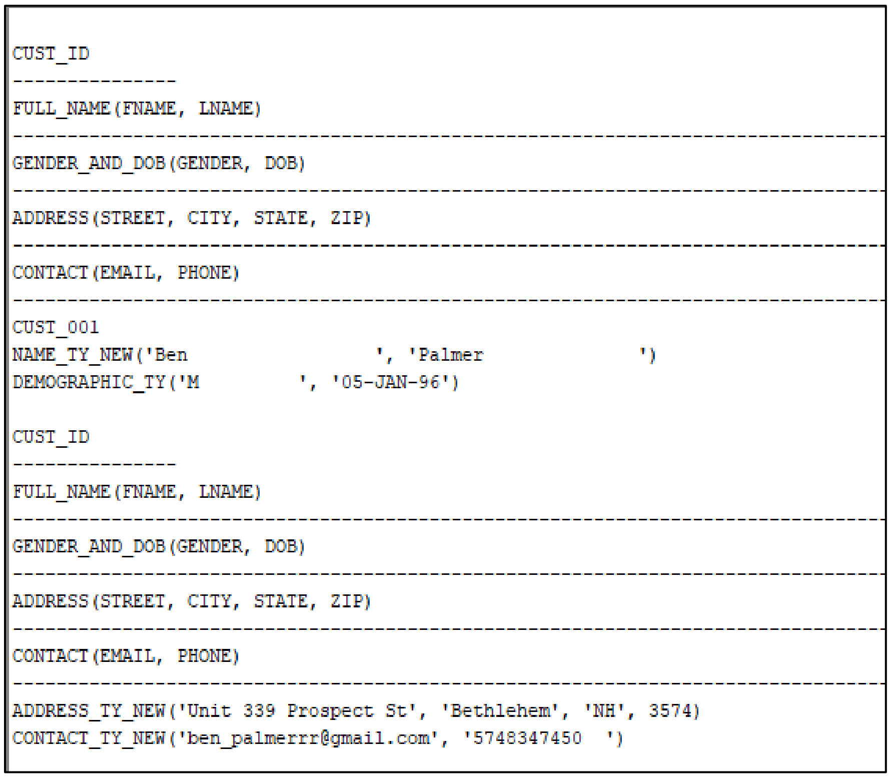
INSERT INTO Customer_New VALUES('CUST_001', Name_ty_new('Ben', 'Palmer'), Demographic_ty('M','05-JAN-96'), Address_ty_new('Unit 339 Prospect St','Bethlehem','NH', 03574), Contact_ty_new('ben_palmerrr@gmail.com', '5748347450'));

INSERT INTO Customer_New VALUES('CUST_002', Name_ty_new('Leon', 'Day'), Demographic_ty('F','08-OCT-89'), Address_ty_new('11110 Mary Ball Rd','Lancaster','VA', 03574), Contact_ty_new('dayleon22@gmail.com', '6017988825'));

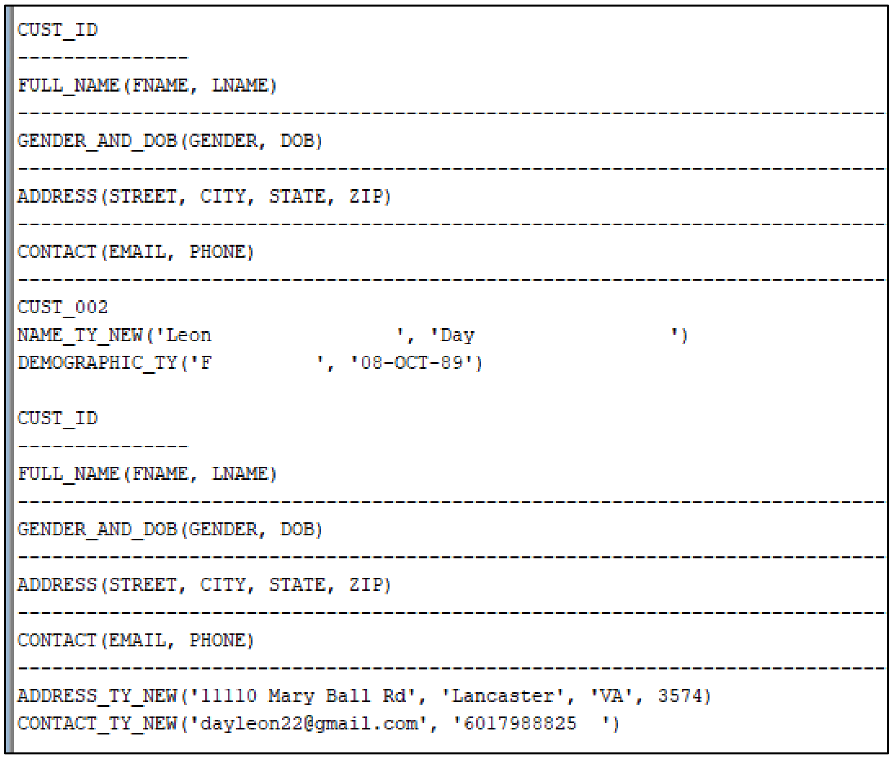
INSERT INTO Customer_New VALUES('CUST_003', Name_ty_new('Tristen', 'Rush'), Demographic_ty('F','17-JUL-65'), Address_ty_new('1203 N Expressway #77 305','Harlingen','TX', 78552), Contact_ty_new('trishrush123@gmail.com', '5075244696'));

Selecting the data from table: Customer _New

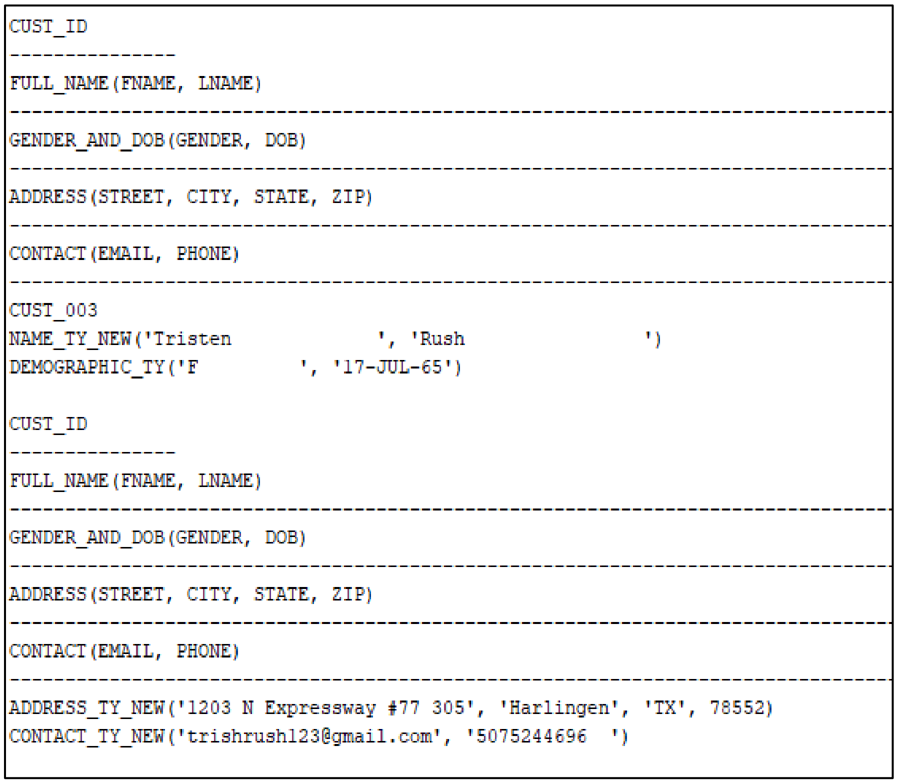
SELECT Cust_ID, o.Full_Name, o. Gender_and_DOB, o.Address, o.Contact
FROM Customer_New o;



```
CUST_ID
---------------
FULL_NAME(FNAME, LNAME)
--------------------------------------------------------------------------------
GENDER_AND_DOB(GENDER, DOB)
--------------------------------------------------------------------------------
ADDRESS(STREET, CITY, STATE, ZIP)
--------------------------------------------------------------------------------
CONTACT(EMAIL, PHONE)
--------------------------------------------------------------------------------
CUST_001
NAME_TY_NEW('Ben                 ', 'Palmer              ')
DEMOGRAPHIC_TY('M         ', '05-JAN-96')

CUST_ID
---------------
FULL_NAME(FNAME, LNAME)
--------------------------------------------------------------------------------
GENDER_AND_DOB(GENDER, DOB)
--------------------------------------------------------------------------------
ADDRESS(STREET, CITY, STATE, ZIP)
--------------------------------------------------------------------------------
CONTACT(EMAIL, PHONE)
--------------------------------------------------------------------------------
ADDRESS_TY_NEW('Unit 339 Prospect St', 'Bethlehem', 'NH', 3574)
CONTACT_TY_NEW('ben_palmerrr@gmail.com', '5748347450  ')
```



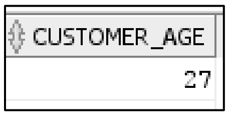
```
CUST_ID
---------------
FULL_NAME(FNAME, LNAME)
--------------------------------------------------------------------------------
GENDER_AND_DOB(GENDER, DOB)
--------------------------------------------------------------------------------
ADDRESS(STREET, CITY, STATE, ZIP)
--------------------------------------------------------------------------------
CONTACT(EMAIL, PHONE)
--------------------------------------------------------------------------------
CUST_002
NAME_TY_NEW('Leon                ', 'Day                 ')
DEMOGRAPHIC_TY('F         ', '08-OCT-89')

CUST_ID
---------------
FULL_NAME(FNAME, LNAME)
--------------------------------------------------------------------------------
GENDER_AND_DOB(GENDER, DOB)
--------------------------------------------------------------------------------
ADDRESS(STREET, CITY, STATE, ZIP)
--------------------------------------------------------------------------------
CONTACT(EMAIL, PHONE)
--------------------------------------------------------------------------------
ADDRESS_TY_NEW('11110 Mary Ball Rd', 'Lancaster', 'VA', 3574)
CONTACT_TY_NEW('dayleon22@gmail.com', '6017988825  ')
```



```
CUST_ID
---------------
FULL_NAME(FNAME, LNAME)
--------------------------------------------------------------------------------
GENDER_AND_DOB(GENDER, DOB)
--------------------------------------------------------------------------------
ADDRESS(STREET, CITY, STATE, ZIP)
--------------------------------------------------------------------------------
CONTACT(EMAIL, PHONE)
--------------------------------------------------------------------------------
CUST_003
NAME_TY_NEW('Tristen             ', 'Rush                ')
DEMOGRAPHIC_TY('F          ', '17-JUL-65')

CUST_ID
---------------
FULL_NAME(FNAME, LNAME)
--------------------------------------------------------------------------------
GENDER_AND_DOB(GENDER, DOB)
--------------------------------------------------------------------------------
ADDRESS(STREET, CITY, STATE, ZIP)
--------------------------------------------------------------------------------
CONTACT(EMAIL, PHONE)
--------------------------------------------------------------------------------
ADDRESS_TY_NEW('1203 N Expressway #77 305', 'Harlingen', 'TX', 78552)
CONTACT_TY_NEW('trishrush123@gmail.com', '5075244696  ')
```

The following SELECT statement calls the method defined in the Customer_New table to get the age of the customer based on DOB:

select c.GENDER_AND_DOB.age('05-JAN-96') as customer_age From Customer_new c;



CUSTOMER_AGE
27