Project 1 Relational Database for an Online Beauty Products Store

Project-1| Group 6

Team Members

Bhumika Suvagia (Team Leader)

Ankita Savaliya (Database Developer)

Akanksha Khaire (Database Developer)

MSIS, California State University, Los Angeles

5430: Databases and Data Warehousing, Spring 2023

Professor Dr. Ming Wang

April 30th, 2023

Table of Content

Project 1(Database Design):

Project 2 (Database Development):

Conceptual and Logical Design

The online Beauty store database needs to keep track of orders for its inventory. When a customer places orders, the system must record the order and order items. The system must update the available quantity on hand to reflect that the by product(s) has been sold. When an employee processes orders, the system must confirm that the ordered items are in stock. The online store needs to keep track of customers and employees, too. The system must update the available quantity on hand to reflect that the by product(s) has been sold. Each team creates your store, database and sell your own products.

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.

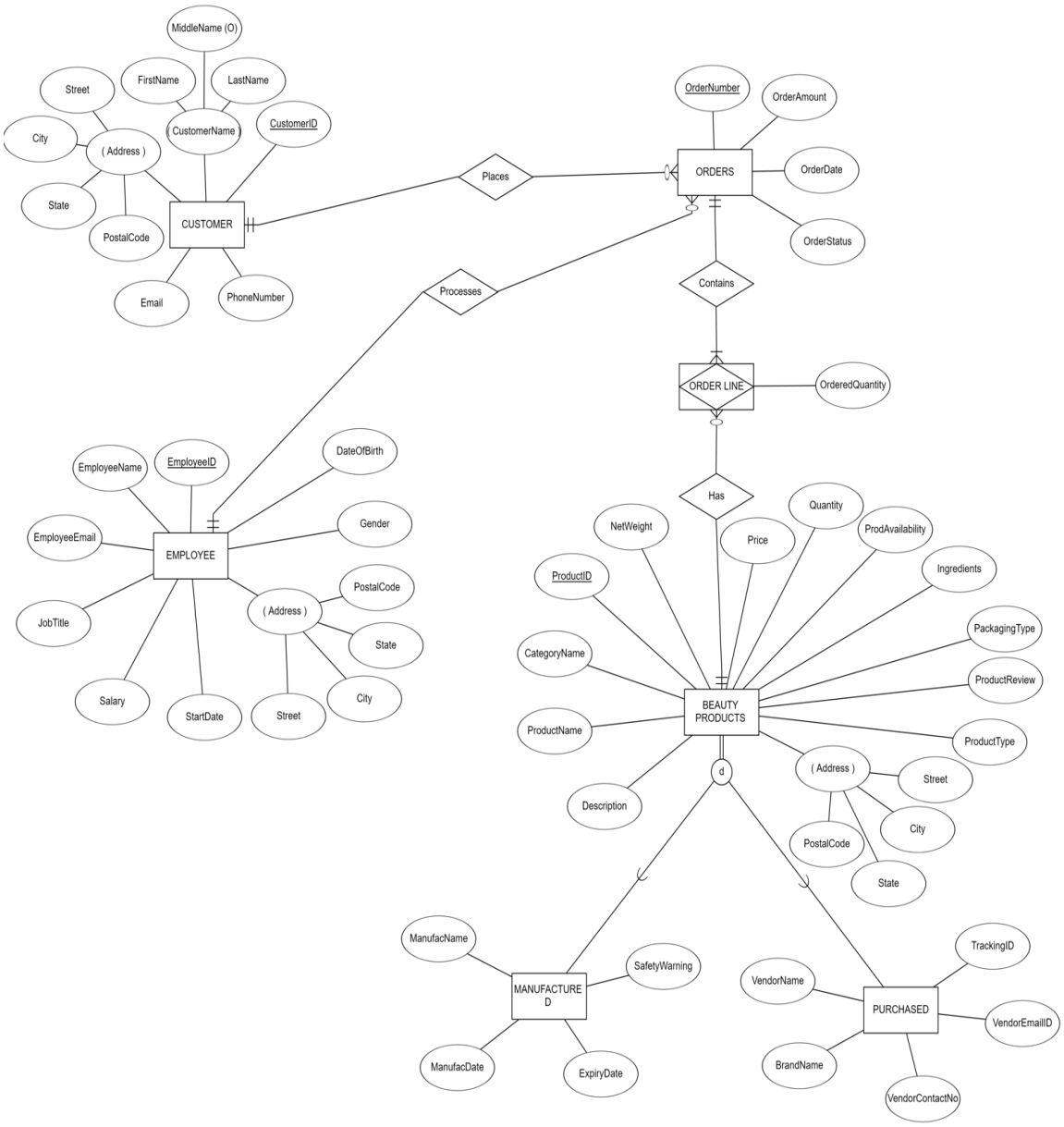
One product must be either manufactured or purchased.

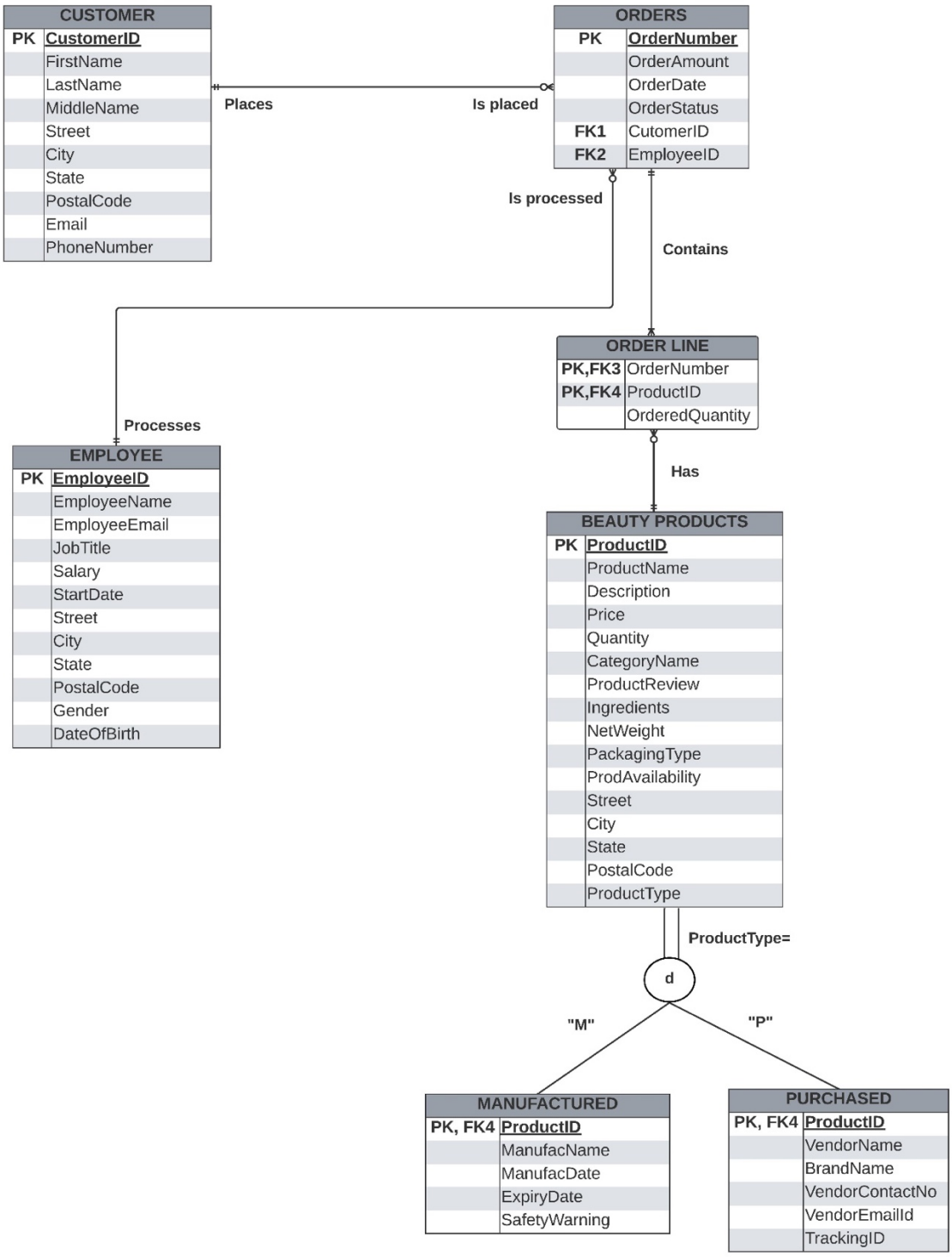
1. Identify entity types and relationship types. Fill out the following relationship matrix.

	Customer	**Orders**	**Product**	**Manufactured**	**Purchased**	**Employee**
Customer	---	Is placed	---	---	---	---
Orders	Places	---	Has	---	---	Processes
Product	---	Contains	---	---	---	---
Manufactured	---	---	Must be	---	---	---
Purchased	---	---	Must be	---	---	---
Employee	---	Is processed	---	---	---	---

2. Draw an ER/EER diagram using software tools includes 1) entity types, 2) relationship types, 3) keys, 4) attributes, and cardinality constraints (must show participation).

ER Diagram




CUSTOMER
PK CustomerID
FirstName
LastName
MiddleName
Street
City
State
PostalCode
Email
PhoneNumber
Places
Is placed
ORDERS
PK OrderNumber
OrderAmount
OrderDate
OrderStatus
FK1 CutomerID
FK2 EmployeeID
Is processed
Contains
ORDER LINE
PK,FK3 OrderNumber
PK,FK4 ProductID
OrderedQuantity
Processes
EMPLOYEE
PK EmployeeID
EmployeeName
EmployeeEmail
JobTitle
Salary
StartDate
Street
City
State
PostalCode
Gender
DateOfBirth
Has
BEAUTY PRODUCTS
PK ProductID
ProductName
Description
Price
Quantity
CategoryName
ProductReview
Ingredients
NetWeight
PackagingType
ProdAvailability
Street
City
State
PostalCode
ProductType
ProductType=
d
"M"
"P"
MANUFACTURED
PK, FK4 ProductID
ManufacName
ManufacDate
ExpiryDate
SafetyWarning
PURCHASED
PK, FK4 ProductID
VendorName
BrandName
VendorContactNo
VendorEmailId
TrackingID

3. 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.

Table Name: **CUSTOMER**

CustomerID(PK)	FirstName	MiddleName	LastName	Street
City	State	PostalCode	Email	PhoneNumber

Table Name: **ORDERS**

OrderNumber(PK)	OrderDate	OrderAmount	OrderStatus
CustomerID(FK1)	EmployeeID(FK2)		

Table Name: **ORDER LINE**

OrderedQuantity	OrderNumber(FK3)	ProductID(FK4)		

Table Name: **BEAUTY PRODUCTS**

ProductID(PK)	CategoryName	ProductName	Description	Price
Quantity	ProdAvailability	ProductReview	Ingredients	NetWeight
PackagingType	Street	City	State	PostalCode
ProductType				

Table Name: **MANUFACTURED**

MProductID(FK4) (PK)	ManufacName	ManufacDate	ExpiryDate	SafetyWarning

Table Name: **PURCHASED**

PProductID (FK4) (PK)	VendorName	BrandName	VendorContactNo	VendorEmailID
TrackingID				

Table Name: **EMPLOYEE**

EmployeeID(PK)	EmployeeName	EmployeeEmail	DateOfBirth	Gender
JobTitle	StartDate	Salary	Street	City
State	PostalCode			

4. Establish join paths for the above relational database using the referential integrity by drawing arrow lines between the above tables. Indicate all the foreign keys (FK).

F.K. -> P.K. (Foreign Key refers to Primary Key)

Database Logical Design:

Referential Integrity:

F.K. →P.K.

ORDERS.CustomerID →CUSTOMER.CustomerID

ORDERS.EmployeeID → EMPLOYEE.EmployeeID

ORDERLINE.OrderNumber → ORDERS.OrderNumber

ORDERLINE.ProductID → BEAUTYPRODUCTS.ProductID

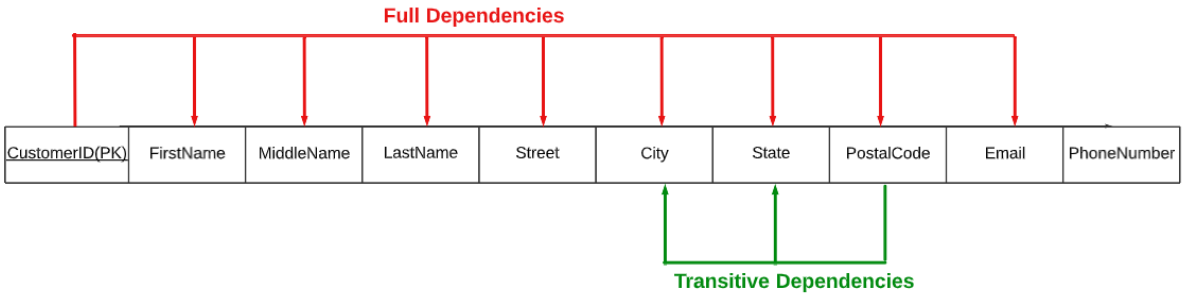
MANUFACTURED. MProductID → BEAUTYPRODUCTS.ProductID

PURCHASED.PProductID → BEAUTYPRODUCTS.ProductID

Function Analysis:

5. Do function analysis for each of your tables Attribute A -> Attribute B (Determinant attribute(s) Determines Dependent Attribute(s))

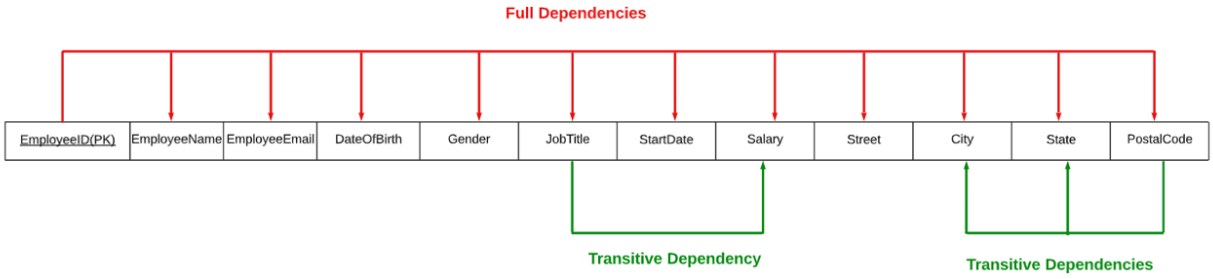
CUSTOMER:



CustomerID → FirstName, MiddleName, LastName, Street, City, State, PostalCode, Email, PhoneNumber (Full Dependencies)

PostalCode → City, State (Transitive Dependencies)

EMPLOYEE:

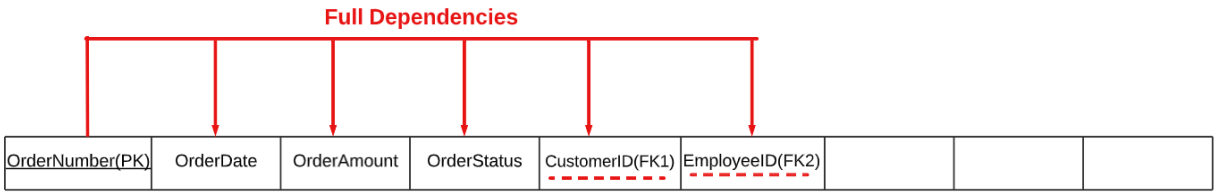


EmployeeID → EmployeeName, EmployeeEmail, DateOfBirth, Gender, JobTitle, StartDate, Salary, Street, City, State, PostalCode (Full Dependencies)

JobTitle → Salary (Transitive Dependency)

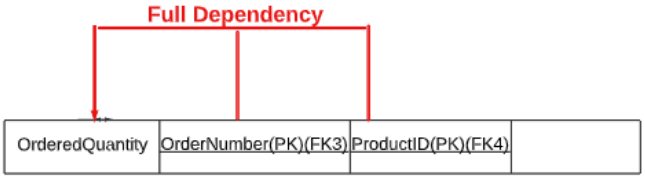
PostalCode → City, State (Transitive Dependencies)

ORDERS:



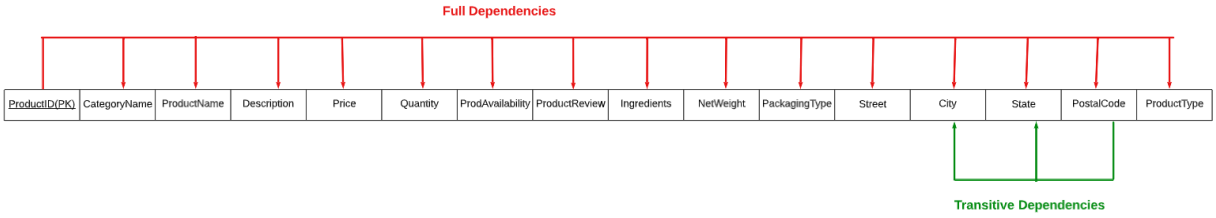
OrderNumber → OrderDate ,OrderAmount, OrderStatus, CustomerID, EmployeeID (Full Dependencies)

ORDERLINE:



(OrderNumber, ProductID) → OrderedQuantity (Full Dependencies)

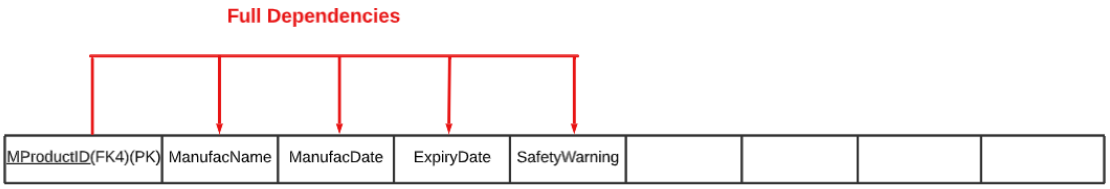
BEAUTY PRODUCTS:



ProductID → ProductName, CategoryName, Description, Price, Quantity, ProductType, ProductReview, ProdAvailability, Ingredients, NetWeight, PackagingType ,ProductType(Full Dependencies)

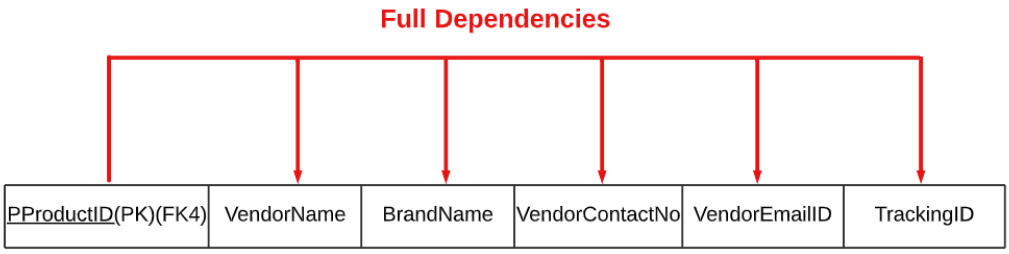
PostalCode → City, State (Transitive Dependencies)

MANUFACTURED:



MProductID → ManufacName,ManufacDate, ExpiryDate, SafetyWarning (Full Dependencies)

PURCHASED:



PProductID -> VendorName, BrandName, VendorContactNo, VendorEmailID, TrackingID(Full Dependencies)

Normalization Forms:

6. Show all the normalized tables and indicate the normalization form for each of your tables.

Table Name	1NF	2NF	3NF
CUSTOMER	✅	✅	
EMPLOYEE	✅	✅	
BEAUTY PRODUCTS	✅	✅	
MANUFACTURED	✅	✅	✅
PURCHASED	✅	✅	✅
ORDER	✅	✅	✅
ORDERLINE	✅	✅	✅

Beauty Products e-store Database

Project-2|Group 6

Name of the Team Members and their Roles and Responsibilities:

Team Leader- Bhumika Suvagia

As a team leader for our e-store beauty product database design and development project, the primary responsibility is to oversee the project from start to finish. Communicating within the team to gain a deep understanding and requirements of the project to ensure that the project is completed on time.

Database Developers- Ankita Savaliya and Akanksha Khaire

We all shared a similar responsibility to make the project work. As a team, we finalized the requirements and created the necessary infrastructure to support the project. This includes creating tables, functions, objects, and stored procedures to ensure that the project has a solid foundation. Also create documentation to ensure that the project is well-documented.

By sharing similar responsibilities, we worked together to achieve the common goal of delivering a successful project.

Introduction (Akanksha Khaire)

Our database design model is developed for an e-store that sells beauty products to customers around the world. The purpose of this database is to keep track of the employees, beauty products inventory, orders, customers, and suppliers of the beauty products. We aim to improve the efficiency of the e-store's operations by providing an organized and secure platform to manage the flow of information.

The e-store is divided into categories of beauty products, each with a distinct name and number. Each category has its own set of products, including their name, description, price, and image. We keep records of the number of products available in stock, the number of the items sold and the number of products on order from suppliers. The e-store has multiple suppliers from whom it sources its products. Some of the products are manufactured and some products are purchased from the vendors. We keep track of the name, tracking information, and products supplied by each supplier.

Customers can create accounts and place orders for the products they want to purchase. We keep track of each customer's name, contact information, shipping and billing addresses, and order history. Each order has a unique order number, date, and status. We also keep track of the amount for each order.

Functionalities:

The database also includes functionalities for the e-store's administrators. They can manage the inventory, update product information, process orders, and track the status of orders. They can also manage the suppliers, update supplier information, and track the status of supplier orders. Additionally, the database has functionalities for generating reports on sales, inventory, and customer data.

Users:

Users of the database include end-users, such as customers who shop on the e-store, and administrators who manage the operations of the e-store. The database is also used by the e-store's developers and designers, who oversee the design and implementation of the e-store's website and backend systems.

Purpose:

Our database design model for the beauty products e-store aims to improve the efficiency and organization of the e-store's operations by providing a secure and organized platform for managing inventory, orders, customers, and suppliers. The database includes functionalities for managing data, generating reports, and ensuring data integrity. It is used by end-users, administrators, developers, and designers to support the e-store's operations

SQL DDL (Database Definition Language):

CUSTOMER TABLE: (Bhumika Suvagia)

```
CREATE TABLE Customer (
 CustomerID INT NOT NULL,
 FirstName VARCHAR2(50),
 LastName VARCHAR2(50),
 MiddleName VARCHAR2(50),
 Street VARCHAR2(100),
 City VARCHAR2(50),
 State VARCHAR2(50),
 PostalCode VARCHAR2(10),
 Email VARCHAR2(100),
 PhoneNumber VARCHAR2(20),
CONSTRAINT CustomerPK PRIMARY KEY(CustomerID)
);

ALTER TABLE Customer
ADD CONSTRAINT Customer_email CHECK (Email LIKE '%_@__%.%');
ALTER TABLE Customer
ADD CONSTRAINT Customer_details UNIQUE (CustomerID, LastName);

INSERT INTO Customer VALUES (1, 'Bhumika', 'Suvagia', 'M', '5022 S Power RD', 'Mesa',
'AZ', '85212', 'johndoe@email.com', '555-1234');
INSERT INTO Customer VALUES (2, 'Ankita', 'Savaliya', NULL, '3935 Grand Ave', 'Chino
Hills', 'CA', '91710', 'janesmith@email.com', '555-5678');
INSERT INTO Customer VALUES (3, 'Akanksha', 'Khair', 'T', '2219 Park Ave', 'Tustin', 'CA',
'92782', 'bob.johnson@email.com', '555-9876');
INSERT INTO Customer VALUES (4, 'Sara', 'Lee', 'J', '200 Towne Center Blvd', 'Sanford', 'FL',
'32771', 'sara.lee@email.com', '555-5555');
```

INSERT INTO Customer VALUES (5, 'David', 'Nguyen', 'H', '1201 F St', 'Washington', 'DC', '20004', 'david.nguyen@email.com', '555-4321');

ORDER TABLE: (Bhumika Suvagia)

```
CREATE TABLE Orders
(
OrderNumber INT NOT NULL,
OrderAmount DECIMAL (10,2),
OrderDate DATE,
OrderStatus VARCHAR2(50),
CustomerID INT,
EmployeeID INT,
CONSTRAINT OrderNumberPK PRIMARY KEY(OrderNumber),
CONSTRAINT OrderNumberFK1 FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID),
CONSTRAINT OrderNumberFK2 FOREIGN KEY (EmployeeID) REFERENCES AKHAIRE3.employee(EmployeeID)
);

ALTER TABLE Orders
MODIFY OrderDate DEFAULT SYSDATE.

INSERT INTO Orders VALUES (1001, 500.00, TO_DATE ('03-22-2015', 'MM-DD-YYYY'), 'Pending', 1, 1);
INSERT INTO Orders VALUES (1002, 750.00, TO_DATE('04-01-2017','MM-DD-YYYY'), 'Completed', 2, 2);
INSERT INTO Orders VALUES (1003, 125.00, TO_DATE('05-22-2015','MM-DD-YYYY'), 'Cancelled', 3, 1);
INSERT INTO Orders VALUES (1004, 1000.00, TO_DATE('02-15-2020','MM-DD-YYYY'), 'Pending', 1, 3);
```

```
INSERT INTO Orders VALUES (1005, 250.00, TO_DATE('05-03-2019','MM-DD-YYYY'),
'Completed', 4, 2);
```

ORDERLINE TABLE: (Akanksha Khaire)

```
CREATE TABLE OrderLine (
OrderNumber INT NOT NULL,
ProductID INT NOT NULL,
OrderedQuantity INT,
CONSTRAINT OrderLinePK PRIMARY KEY (OrderNumber, ProductID),
CONSTRAINT OrderLine_FK3 FOREIGN KEY (OrderNumber) REFERENCES
bsuvagi.Orders (OrderNumber),
CONSTRAINT OrderLine_FK4 FOREIGN KEY(ProductID) REFERENCES
asavali2.Beauty_Products(ProductID)
);

INSERT INTO OrderLine VALUES (1001, 1, 2);
INSERT INTO OrderLine VALUES (1001, 2, 1);
INSERT INTO OrderLine VALUES (1002, 3, 3);
INSERT INTO OrderLine VALUES (1002, 4, 2);
INSERT INTO OrderLine VALUES (1003, 5, 1);
```

BEAUTY PRODUCTS TABLE: (Ankita Savaliya)

```
DROP TABLE Beauty_Products;
CREATE TABLE Beauty_Products
(
   ProductID INT NOT NULL,
   ProductName VARCHAR2(50),
   Description VARCHAR2(50),
   Price DECIMAL(10,2),
   Quantity INT,
```

```
    CategoryName VARCHAR2(30),
    ProductReview VARCHAR2(100),
    Ingredients VARCHAR2(50),
    NetWeight VARCHAR2(10),
    PackagingType VARCHAR2(20),
    ProdAvailability VARCHAR2(15),
    Street VARCHAR2(50),
    City VARCHAR2(25),
    State CHAR(2),
    PostalCode VARCHAR2(5),
    ProductType VARCHAR2(5),
    CONSTRAINT ProductIDPK PRIMARY KEY(ProductID)
);

ALTER TABLE Beauty_Products
ADD CONSTRAINT ProdAvalibility_Check CHECK (ProdAvailability IN ('Available','Not
Available'));
ALTER TABLE Beauty_Products
ADD CONSTRAINT ProductType_ManufacturedOrPurchased CHECK (ProductType
IN('M','P'));

INSERT INTO Beauty_Products (ProductID, ProductName, Description, Price, Quantity,
CategoryName, ProductReview, Ingredients, NetWeight, PackagingType, ProdAvailability,
Street, City, State, PostalCode, ProductType)
VALUES (1, 'Lipstick', 'Long-lasting lipstick in a bold red shade', 9.99, 50, 'Makeup', 'Love this
lipstick! Goes on smooth and lasts for hours', 'Castor oil', '12 oz', 'Tube', 'Available', '356 OLD
STEESE HWY', 'FAIRBANKS', 'AK', '99701', 'M');
INSERT INTO Beauty_Products VALUES (2, 'Mascara', 'Lengthening mascara in black', 14.99,
30, 'Eye', 'This mascara is amazing! No clumping and makes my lashes look so long!', 'Vitamin
E, jojoba oil', '28 oz', 'Bottle', 'Available', '2200 S ATLANTIC BL', 'LOS ANGELES', 'CA',
'91754', 'P');
```

INSERT INTO Beauty_Products VALUES (3, 'Facial Cleanser', 'Gentle cleanser for all skin types', 8.99, 75, 'Skincare', 'I love how this cleanser leaves my skin feeling soft and clean!', 'Water, glycerin, aloe vera', '4.2 oz', 'Bottle', 'Available', '2980 E CAPITOL EXPY STE 60', 'SAN JOSE', 'CA', '95148', 'M');
INSERT INTO Beauty_products VALUES (4, 'Hair Spray', 'Flexible hold hairspray for all hair types', 6.99, 40, 'Haircare', 'This hairspray keeps my style in place without feeling crunchy or stiff!', 'Alcohol, water, panthenol', '8 oz', 'Can', 'Available', '11954 CARMEL MOUNTAIN ROAD', 'SAN DIEGO', 'CA', '92128', 'P');
INSERT INTO Beauty_Products VALUES(5, 'Sunscreen', 'Broad spectrum SPF 50 sunscreen lotion', 15.99, 40, 'Skincare', 'This is my new go-to sunscreen! No white cast and doesn''t feel heavy on my skin', 'Zinc oxide, octinoxate', '3.4 oz', 'Tube', 'Available', '13424 BISCAYNE BLVD', 'MIAMI', 'FL', '33181', 'M');

MANUFACTURED TABLE: (Ankita Savaliya)

CREATE TABLE Manufactured
(MProductID INT NOT NULL,
ManufacName VARCHAR2(50),
ManufacDate DATE,
ExpiryDate DATE,
SafetyWarning VARCHAR2(80),
CONSTRAINT Manufactured_PK PRIMARY KEY(MProductID),
CONSTRAINT Manufactured_FK4 FOREIGN KEY(MProductID) REFERENCES Beauty_Products (ProductID));

INSERT INTO Manufactured VALUES
(1,'AB Beauty Products', TO_DATE('01-10-2022','MM-DD-YYYY'), TO_DATE('07-22-2024','MM-DD-YYYY'),'For external use only. Avoid contact with eyes. Keep out of reach of children')
INSERT INTO Manufactured

VALUES (3,'AB Beauty Products',TO_DATE('01-10-2022','MM-DD-YYYY'),TO_DATE('07-22-2024','MM-DD-YYYY'),'For external use only. Avoid contact with eyes. Keep out of reach of children')

INSERT INTO Manufactured

VALUES (5,'AB Beauty Products',TO_DATE('03-05-2021','MM-DD-YYYY'),TO_DATE('07-22-2024','MM-DD-YYYY'),'For external use only Do not use on damaged or broken skin')

PURCHASED TABLE: (Ankita Savaliya)

CREATE TABLE Purchased

(PProductID INT NOT NULL,

VendorName VARCHAR2(50),

BrandName VARCHAR2(50),

VendorContactNo VARCHAR2(15),

VendorEmailId VARCHAR2(25),

TrackingID VARCHAR (20));

ALTER TABLE Purchased

ADD CONSTRAINT Purchased_PK PRIMARY KEY(PProductID);

ALTER TABLE Purchased

ADD CONSTRAINT Purchased_FK4 FOREIGN KEY(PProductID) REFERENCES Beauty_Products (ProductID);

INSERT INTO Purchased

VALUES (2,'Ziya Beauty Stores','Dior','626-234-7546', 'ZiyaBeauty12@gmail.com','934234')

INSERT INTO Purchased

VALUES (4,'Sanya E-Beauty','Sanya','626-546-7546', 'sanya12@gmail.com','943523')

EMPLOYEE TABLE: (Akanksha Khaire)

```
CREATE TABLE Employee
(
   EmployeeID INT NOT NULL,
   EmployeeName VARCHAR2(25),
   Street VARCHAR2(50),
   City VARCHAR2(50),
   State VARCHAR2(2),
   PostalCode VARCHAR2(10),
   DateOfBirth DATE,
   Gender VARCHAR2(6),
   Salary NUMBER (9,2),
   CONSTRAINT EmployeePK PRIMARY KEY (EmployeeID),
   CONSTRAINT Gender CHECK (Gender IN ('Male','Female','M','F')),
   CONSTRAINT Salary CHECK (Salary > 0)
);

INSERT INTO Employee VALUES ('1','Jeniffer Smith','13 Any St, Apt A','LA','CA','91001','09-JAN-55','M','50000');
INSERT INTO Employee VALUES ('2','Johny Ward','12 Any St Apt B','LA','CA','91002','08-DEC-45','M','30000');
INSERT INTO Employee VALUES ('3','Gloria Jenkins','14 Any St Apt C','LA', 'CA','91003','19-JUL-58','F','45000');
INSERT INTO Employee VALUES ('4','Aaron Cox','15 Any St Apt D','LA','CA','91004','20-JUN-31','F','35000');
INSERT INTO Employee VALUES ('5','Teresa Butler','16 Any St Apt E','LA','CA','91005','15-SEP-52','M','56000');
```

Database Structure:

DESC Customer; (Bhumika Suvagia)

```
Name         Null?    Type
------------ -------- -------------
CUSTOMERID   NOT NULL NUMBER
FIRSTNAME             VARCHAR2(50)
LASTNAME              VARCHAR2(50)
MIDDLENAME            VARCHAR2(50)
STREET                VARCHAR2(100)
CITY                  VARCHAR2(50)
STATE                 VARCHAR2(50)
POSTALCODE            VARCHAR2(10)
EMAIL                 VARCHAR2(100)
PHONENUMBER           VARCHAR2(20)
```

DESC Order; (Bhumika Suvagia)

```
Name         Null?    Type
------------ -------- ------------
ORDERNUMBER NOT NULL NUMBER(38)
ORDERAMOUNT          NUMBER(10,2)
ORDERDATE            DATE
ORDERSTATUS          VARCHAR2(50)
CUSTOMERID           NUMBER(38)
EMPLOYEEID           NUMBER(38)
```

DESC OrderLine; (Akanksha Khaire)

```
Name            Null?    Type
--------------- -------- ----------
ORDERNUMBER     NOT NULL NUMBER(38)
PRODUCTID       NOT NULL NUMBER(38)
ORDEREDQUANTITY          NUMBER(38)
```

DESC Beauty_Products; (Ankita Savaliya)

```
Name             Null?    Type
---------------- -------- -------------
PRODUCTID        NOT NULL NUMBER(38)
PRODUCTNAME               VARCHAR2(50)
DESCRIPTION               VARCHAR2(50)
PRICE                     NUMBER(10,2)
QUANTITY                  NUMBER(38)
CATEGORYNAME              VARCHAR2(30)
PRODUCTREVIEW             VARCHAR2(100)
INGREDIENTS               VARCHAR2(50)
NETWEIGHT                 VARCHAR2(10)
PACKAGINGTYPE             VARCHAR2(20)
PRODAVAILABILITY          VARCHAR2(15)
STREET                    VARCHAR2(50)
CITY                      VARCHAR2(25)
STATE                     CHAR(2)
POSTALCODE                VARCHAR2(5)
PRODUCTTYPE               VARCHAR2(5)
```

DESC Manufactured; (Ankita Savaliya)

```
Name          Null?    Type
------------- -------- ------------
MPRODUCTID    NOT NULL NUMBER(38)
MANUFACNAME            VARCHAR2(50)
MANUFACDATE            DATE
EXPIRYDATE             DATE
SAFETYWARNING          VARCHAR2(80)
```

DESC Purchased; (Ankita Savaliya)

```
Name            Null?    Type
--------------- -------- ------------
PPRODUCTID      NOT NULL NUMBER(38)
VENDORNAME               VARCHAR2(50)
BRANDNAME                VARCHAR2(50)
VENDORCONTACTNO          VARCHAR2(15)
VENDOREMAILID            VARCHAR2(25)
TRACKINGID               VARCHAR2(20)
```

DESC Employee; (Akanksha Khaire)

```
Name          Null?    Type
------------- -------- ------------
EMPLOYEEID    NOT NULL NUMBER(38)
EMPLOYEENAME           VARCHAR2(25)
STREET                 VARCHAR2(50)
CITY                   VARCHAR2(50)
STATE                  VARCHAR2(2)
POSTALCODE             VARCHAR2(10)
DATEOFBIRTH            DATE
GENDER                 VARCHAR2(6)
SALARY                 NUMBER(9,2)
```

Database Entity Instances:

SELECT * FROM Customer; (Bhumika Suvagia)

	CUSTOMERID	FIRSTNAME	LASTNAME	MIDDLENAME	STREET	CITY	STATE	POSTALCODE	EMAIL	PHONENUMBER
1	1	Bhumika	Suvagia	M	5022 S Power RD	Mesa	AZ	85212	johndoe@email.com	555-1234
2	2	Ankita	Savaliya	(null)	3935 Grand Ave	Chino Hills	CA	91710	janesmith@email.com	555-5678
3	3	Akanksha	Khair	T	2219 Park Ave	Tustin	CA	92782	bob.johnson@email.com	555-9876
4	4	Sara	Lee	J	200 Towne Center Blvd	Sanford	FL	32771	sara.lee@email.com	555-5555
5	5	David	Nguyen	H	1201 F St	Washington	DC	20004	david.nguyen@email.com	555-4321

SELECT * FROM Orders; (Bhumika Suvagia)

	ORDERNUMBER	ORDERAMOUNT	ORDERDATE	ORDERSTATUS	CUSTOMERID	EMPLOYEEID
1	1001	500	22-MAR-15	Pending	1	1
2	1002	750	01-APR-17	Completed	2	2
3	1003	125	22-MAY-15	Cancelled	3	1
4	1004	1000	15-FEB-20	Pending	1	3
5	1005	250	03-MAY-19	Completed	4	2

SELECT * FROM Beauty_Products; (Ankita Savaliya)

	PRODUCTID	PRODUCTNAME	DESCRIPTION	PRICE	QUANTITY	CATEGORYNAME
1	2	Mascara	Lengthening mascara in black	14.99	30	Eye
2	3	Facial Cleanser	Gentle cleanser for all skin types	8.99	75	Skincare
3	4	Hair Spray	Flexible hold hairspray for all hair types	6.99	40	Haircare
4	5	Sunscreen	Broad spectrum SPF 50 sunscreen lotion	15.99	40	Skincare
5	1	Lipstick	Long-lasting lipstick in a bold red shade	9.99	50	Makeup

	PRODUCTREVIEW	INGREDIENTS	NETWEIGHT	PACKAGINGTYPE
1	This mascara is amazing! No clumping and makes my lashes look so long!	Vitamin E, jojoba oil	28 oz	Bottle
2	I love how this cleanser leaves my skin feeling soft and clean!	Water, glycerin, aloe vera	4.2 oz	Bottle
3	This hairspray keeps my style in place without feeling crunchy or stiff!	Alcohol, water, panthenol	8 oz	Can
4	This is my new go-to sunscreen! No white cast and doesn't feel heavy on my skin	Zinc oxide, octinoxate	3.4 oz	Tube
5	Love this lipstick! Goes on smooth and lasts for hours	Castor oil	12 oz	Tube

INGREDIENTS	NETWEIGHT	PACKAGINGTYPE	PRODAVAILABILITY	STREET	CITY	STATE	POSTALCODE	PRODUCTTYPE
Vitamin E, jojoba oil	28 oz	Bottle	Available	2200 S ATLANTIC BL	LOS ANGELES	CA	91754	P
Water, glycerin, aloe vera	4.2 oz	Bottle	Available	2980 E CAPITOL EXPY STE 60	SAN JOSE	CA	95148	M
Alcohol, water, panthenol	8 oz	Can	Available	11954 CARMEL MOUNTAIN ROAD	SAN DIEGO	CA	92128	P
Zinc oxide, octinoxate	3.4 oz	Tube	Available	13424 BISCAYNE BLVD	MIAMI	FL	33181	M
Castor oil	12 oz	Tube	Available	356 OLD STEESE HWY	FAIRBANKS	AK	99701	M

SELECT * FROM Manufactured; (Ankita Savaliya)

	MPRODUCTID	MANUFACNAME	MANUFACDATE	EXPIRYDATE	SAFETYWARNING
1	1	AB Beauty Products	22-MAY-22	22-JUL-22	For external use only
2	3	AB Beauty Products	10-JAN-22	22-JUL-24	For external use only. Avoid contact with eyes.Keep out of reach of children
3	5	AB Beauty Products	05-MAR-21	22-JUL-24	For external use only Do not use on damaged or broken skin

SELECT * FROM Purchased; (Ankita Savaliya)

	PPRODUCTID	VENDORNAME	BRANDNAME	VENDORCONTACTNO	VENDOREMAILID	TRACKINGID
1	2	Ziya Beauty Stores	Dior	626-234-7546	ZiyaBeauty12@gmail.com	934234
2	4	Sanya E-Beauty	Sanya	626-546-7546	sanya12@gmail.com	943523

SELECT * FROM Employee; (Akanksha Khaire)

	EMPLOYEEID	EMPLOYEENAME	STREET	CITY	STATE	POSTALCODE	DATEOFBIRTH	GENDER	SALARY
1	1	Jeniffer Smith	13 Any St, Apt A	LA	CA	91001	09-JAN-55	M	50000
2	2	Johny Ward	12 Any St Apt B	LA	CA	91002	08-DEC-45	M	30000
3	3	Gloria Jenkins	14 Any St Apt C	LA	CA	91003	19-JUL-58	F	45000
4	4	Aaron Cox	15 Any St Apt D	LA	CA	91004	20-JUN-31	F	35000
5	5	Teresa Butler	16 Any St Apt E	LA	CA	91005	15-SEP-52	M	56000

SELECT * FROM OrderLine; (Akanksha Khaire)

	ORDERNUMBER	PRODUCTID	ORDEREDQUANTITY
1	1001	1	2
2	1001	2	1
3	1002	3	3
4	1002	4	2
5	1003	5	1

Insert, Update, Delete and Views:

INSERTION (Ankita Savaliya)

INSERT Value in Customer Table

	CUSTOMERID	FIRSTNAME	LASTNAME	MIDDLENAME	STREET	CITY	STATE	POSTALCODE	EMAIL	PHONENUMBER
1	1	Bhumika	Suvagia	M	5022 S Power RD	Mesa	AZ	85212	johndoe@email.com	555-1234
2	2	Ankita	Savaliya	(null)	3935 Grand Ave	Chino Hills	CA	91710	janesmith@email.com	555-5678
3	3	Akanksha	Khair	T	2219 Park Ave	Tustin	CA	92782	bob.johnson@email.com	555-9876
4	4	Sara	Lee	J	200 Towne Center Blvd	Sanford	FL	32771	sara.lee@email.com	555-5555
5	5	David	Nguyen	H	1201 F St	Washington	DC	20004	david.nguyen@email.com	555-4321

SQL Statement:

INSERT INTO BSUVAGI.Customer VALUES (6, 'Jenny', 'Sammy', 'F', '356 OLD STEESE HWY', 'FAIRBANKS', 'AK', '99701', 'JennySam46@gmail.com', '(907) 459-2355');

SELECT * FROM BSUVAGI.Customer;

SQL Output:

	CUSTOMERID	FIRSTNAME	LASTNAME	MIDDLENAME	STREET	CITY	STATE	POSTALCODE	EMAIL	PHONENUMBER
1	1	Bhumika	Suvagia	M	5022 S Power RD	Mesa	AZ	85212	johndoe@email.com	555-1234
2	2	Ankita	Savaliya	(null)	3935 Grand Ave	Chino Hills	CA	91710	janesmith@email.com	555-5678
3	3	Akanksha	Khair	T	2219 Park Ave	Tustin	CA	92782	bob.johnson@email.com	555-9876
4	4	Sara	Lee	J	200 Towne Center Blvd	Sanford	FL	32771	sara.lee@email.com	555-5555
5	5	David	Nguyen	H	1201 F St	Washington	DC	20004	david.nguyen@email.com	555-4321
6	6	Jenny	Sammy	F	356 OLD STEESE HWY	FAIRBANKS	AK	99701	JennySam46@gmail.com	(907) 459-2355

UPDATE: (Ankita Savaliya)

Update value in Employee Table

	EMPLOYEEID	EMPLOYEENAME	STREET	CITY	STATE	POSTALCODE	DATEOFBIRTH	GENDER	SALARY
1	1	Jeniffer Smith	13 Any St, Apt A	LA	CA	91001	09-JAN-55	M	50000
2	2	Johny Ward	12 Any St Apt B	LA	CA	91002	08-DEC-45	M	30000
3	3	Gloria Jenkins	14 Any St Apt C	LA	CA	91003	19-JUL-58	F	45000
4	4	Aaron Cox	15 Any St Apt D	LA	CA	91004	20-JUN-31	F	35000
5	5	Teresa Butler	16 Any St Apt E	LA	CA	91005	15-SEP-52	M	56000

SQL Statement:

UPDATE AKHAIRE3.Employee SET Street='12625 FREDERICK ST',City='RIVERSIDE',State='CA',PostalCode='92503' WHERE EmployeeID=2;

UPDATE AKHAIRE3.Employee SET Street='301 E US HIGHWAY 82',
City='MCKINNEY',State='FL',PostalCode='32811' WHERE EmployeeID=3;

UPDATE AKHAIRE3.Employee SET Street='2135 UNION ST',City='SAN
FRANCISCO',State='CA',PostalCode='94123' WHERE EmployeeID=4;

SQL Output:

	EMPLOYEEID	EMPLOYEENAME	STREET	CITY	STATE	POSTALCODE	DATEOFBIRTH	GENDER	SALARY
1	1	Jeniffer Smith	13 Any St, Apt A	LA	CA	91001	09-JAN-55	M	50000
2	2	Johny Ward	12625 FREDERICK ST	RIVERSIDE	CA	92503	08-DEC-45	M	30000
3	3	Gloria Jenkins	4736 S KIRKMAN RD	ORLANDO	FL	32811	19-JUL-58	F	45000
4	4	Aaron Cox	2135 UNION ST	SAN FRANCISCO	CA	94123	20-JUN-31	F	35000
5	5	Teresa Butler	16 Any St Apt E	LA	CA	91005	15-SEP-52	M	56000

DELETE (Akanksha Khaire)

DELETE Value in Employee Table

	EMPLOYEEID	EMPLOYEENAME	STREET	CITY	STATE	POSTALCODE	DATEOFBIRTH	GENDER	SALARY
1	1	Jeniffer Smith	13 Any St, Apt A	LA	CA	91001	09-JAN-55	M	50000
2	2	Johny Ward	12 Any St Apt B	LA	CA	91002	08-DEC-45	M	30000
3	3	Gloria Jenkins	14 Any St Apt C	LA	CA	91003	19-JUL-58	F	45000
4	4	Aaron Cox	15 Any St Apt D	LA	CA	91004	20-JUN-31	F	35000
5	5	Teresa Butler	16 Any St Apt E	LA	CA	91005	15-SEP-52	M	56000

SQL Statement:

DELETE FROM EMPLOYEE WHERE EmployeeID=5;

SELECT * FROM Employee;

SQL Output:

	EMPLOYEEID	EMPLOYEENAME	STREET	CITY	STATE	POSTALCODE	DATEOFBIRTH	GENDER	SALARY
1	1	Jeniffer Smith	13 Any St, Apt A	LA	CA	91001	09-JAN-55	M	50000
2	2	Johny Ward	12 Any St Apt B	LA	CA	91002	08-DEC-45	M	30000
3	3	Gloria Jenkins	14 Any St Apt C	LA	CA	91003	19-JUL-58	F	45000
4	4	Aaron Cox	15 Any St Apt D	LA	CA	91004	20-JUN-31	F	35000

VIEWS:

Create View for Customers and Orders: (Bhumika Suvagia)

SQL Statement:

-- Create the view

CREATE VIEW CustomerOrders AS

SELECT c.CustomerID, c.FirstName, o.OrderNumber, o.OrderDate, o.OrderAmount

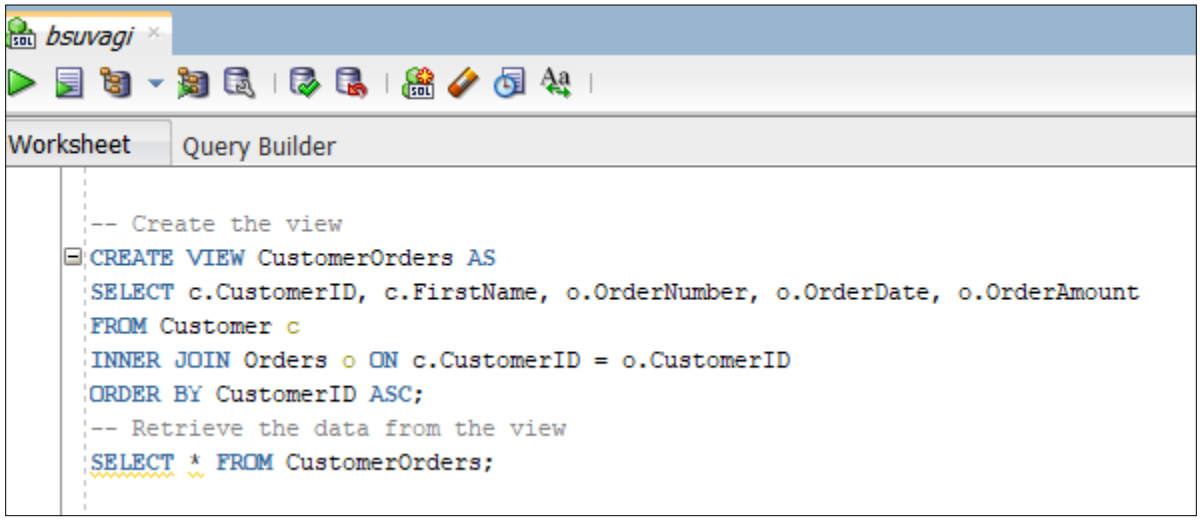
FROM Customer c

INNER JOIN Orders o ON c.CustomerID = o.CustomerID

ORDER BY CustomerID ASC;

-- Retrieve the data from the view

SELECT * FROM CustomerOrders;



SQL Output:

	CUSTOMERID	FIRSTNAME	ORDERNUMBER	ORDERDATE	ORDERAMOUNT
1	1	Bhumika	1001	22-MAR-15	500
2	1	Bhumika	1004	15-FEB-20	1000
3	2	Ankita	1002	01-APR-17	750
4	3	Akanksha	1003	22-MAY-15	125
5	4	Sara	1005	03-MAY-19	250

Create view for Manufactured and Purchased Products: (Bhumika Suvagia)

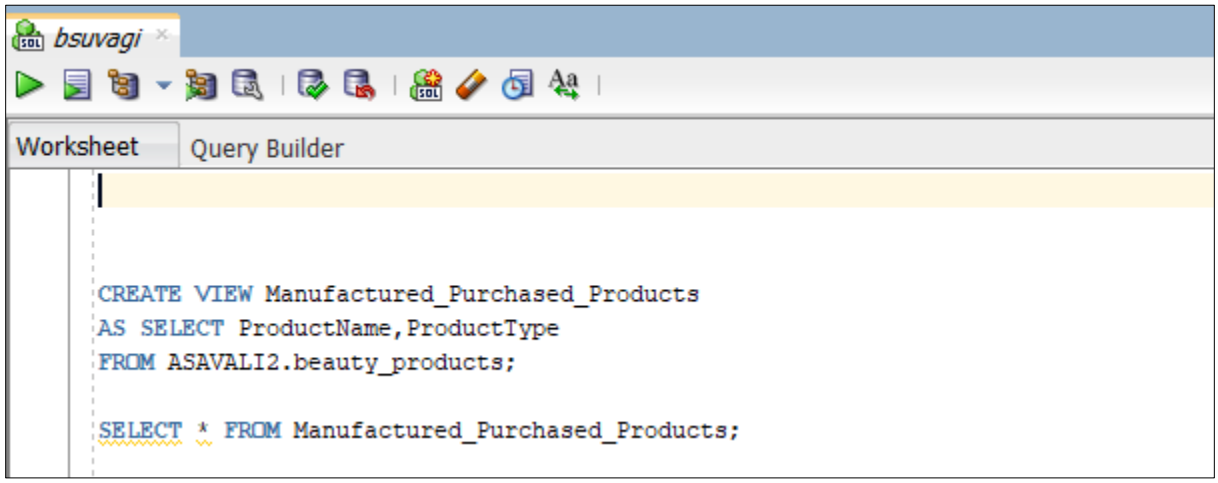
SQL Statement:

CREATE VIEW Manufactured_Purchased_Products

AS SELECT ProductName,ProductType

FROM ASAVALI2.beauty_products;

SELECT * FROM Manufactured_Purchased_Products;



SQL Output:

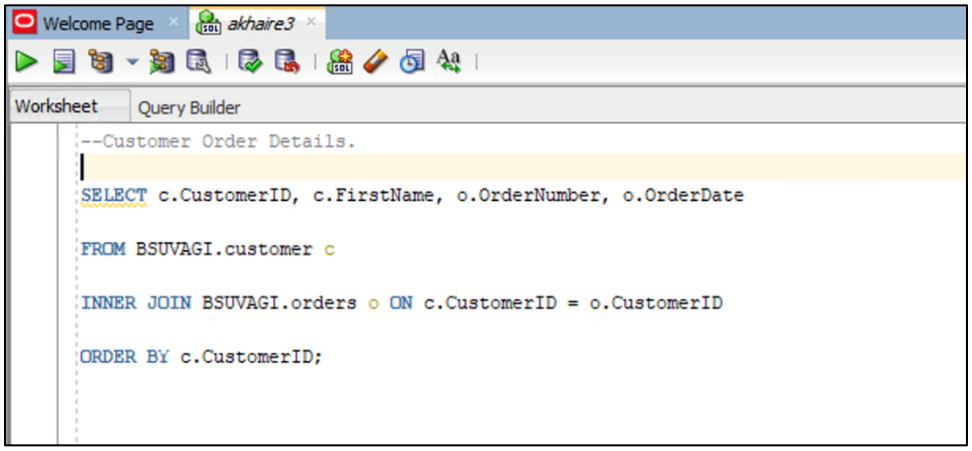
	PRODUCTNAME	PRODUCTTYPE
1	Mascara	P
2	Facial Cleanser	M
3	Hair Spray	P
4	Sunscreen	M
5	Lipstick	M

SQL Test Statements:

1.Display the customer order information which includes customer firstname, order number and order date. (Akanksha Khaire)

SQL Statement: (Using Inner Join and Order by)

```
--Customer Order Details.
SELECT c.CustomerID, c.FirstName, o.OrderNumber, o.OrderDate
FROM BSUVAGI.customer c
INNER JOIN BSUVAGI.orders o ON c.CustomerID = o.CustomerID
ORDER BY c.CustomerID
```



SQL Output:

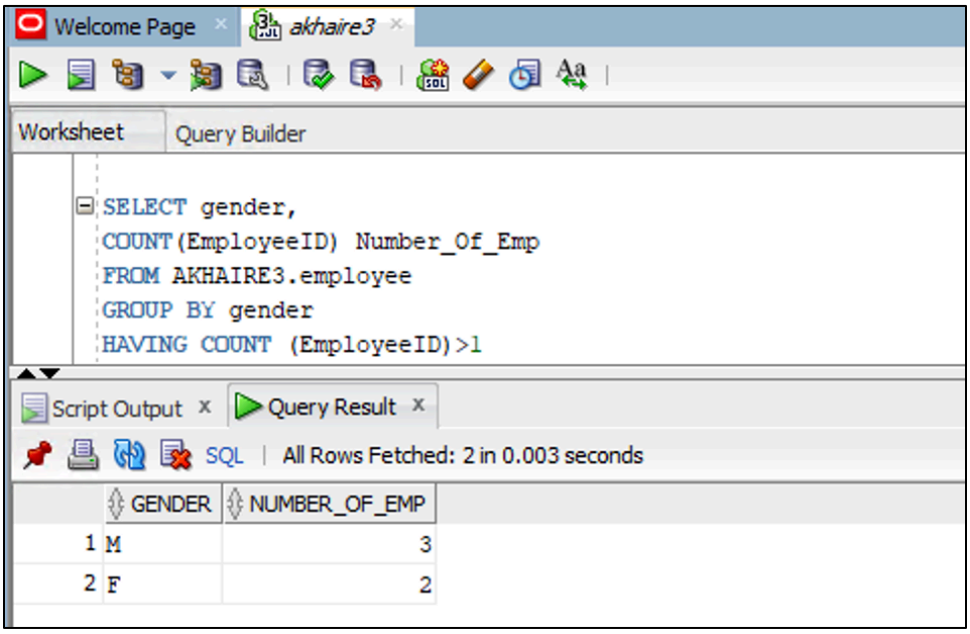
	CUSTOMERID	FIRSTNAME	ORDERNUMBER	ORDERDATE
1	1	Bhumika	1001	22-MAR-15
2	1	Bhumika	1004	15-FEB-20
3	2	Ankita	1002	01-APR-17
4	3	Akanksha	1003	22-MAY-15
5	4	Sara	1005	03-MAY-19

2.Count the number of male and female employees who process the orders. (Akanksha Khaire)

SQL Statement: (Using Group by, Count(), and Having)

```
SELECT gender,
COUNT(EmployeeID) Number_Of_Emp
FROM AKHAIRE3.employee
GROUP BY gender
HAVING COUNT (EmployeeID)>1
```

SQL Output:



3.Retrieve Customer Name from Customer table where Order Number equals to 1003. (Akanksha Khaire)

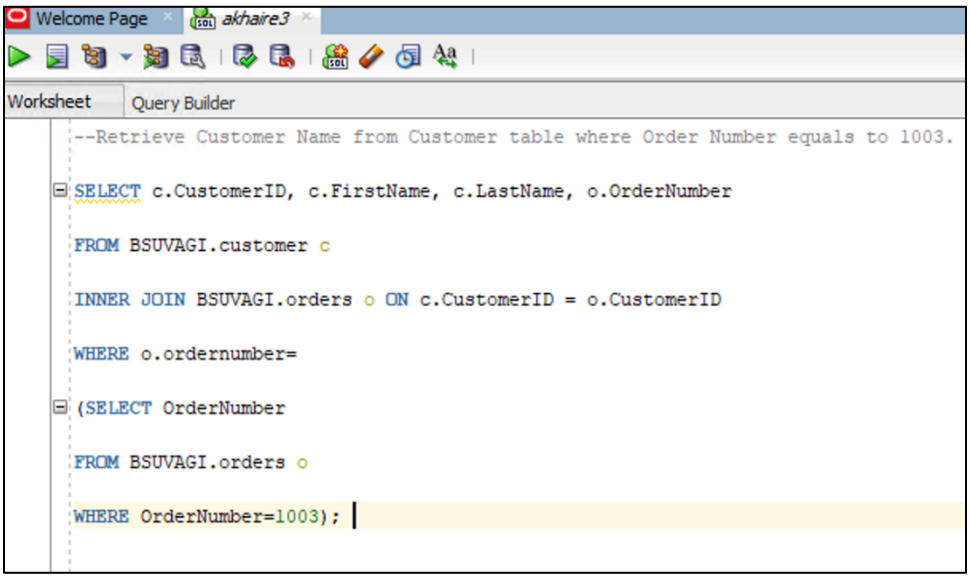
SQL Statement: (Using Inner Join and Subquery)

```
--Retrieve Customer Name from Customer table where Order Number equals to 1003.
SELECT c.CustomerID, c.FirstName, c.LastName, o.OrderNumber
FROM BSUVAGI.customer c
INNER JOIN BSUVAGI.orders o ON c.CustomerID = o.CustomerID
WHERE o.ordernumber=
```

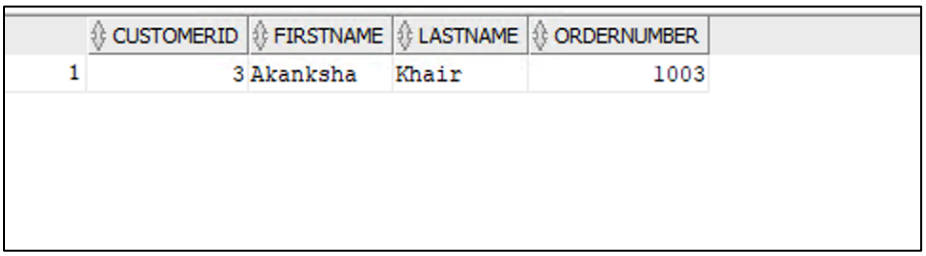
(SELECT OrderNumber

FROM BSUVAGI.orders o

WHERE OrderNumber=1003);



SQL Output:



	CUSTOMERID	FIRSTNAME	LASTNAME	ORDERNUMBER
1	3	Akanksha	Khair	1003

4.Display the order numbers and the number of product types placed for each order

(Bhumika Suvagia)

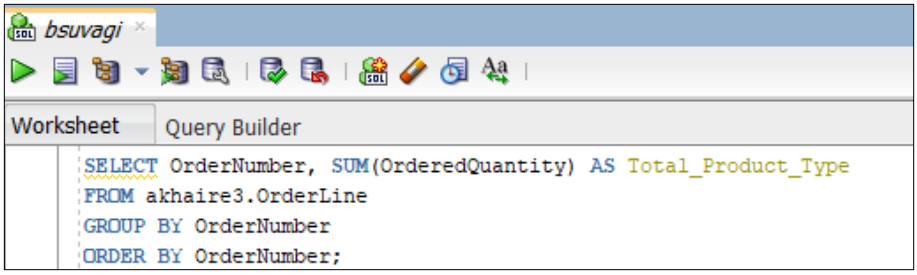
SQL Statement: (Using Group by and Order by)

SELECT OrderNumber, SUM(OrderedQuantity) AS Total_Product_Type

FROM akhaire3.OrderLine

GROUP BY OrderNumber

ORDER BY OrderNumber;



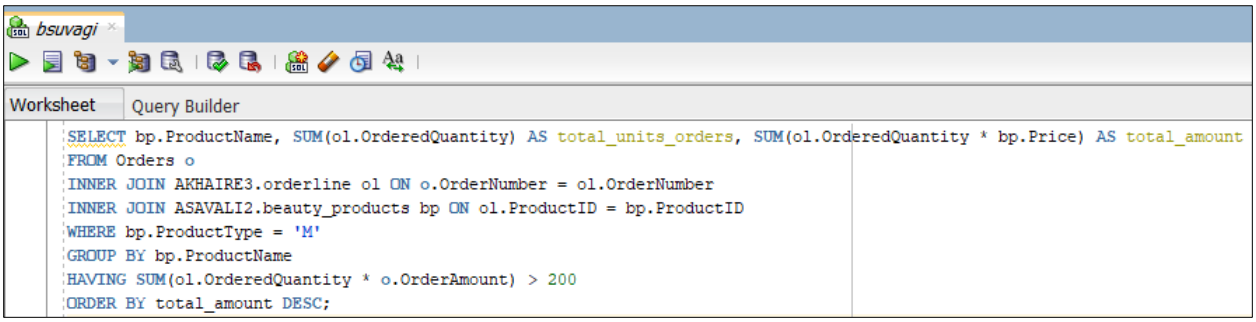
SQL Output:

	ORDERNUMBER	TOTAL_PRODUCT_TYPE
1	1001	3
2	1002	5
3	1003	1

5.What are the top 2 best-selling manufactured beauty products by total amount, and how many units of each product were ordered? (Bhumika Suvagiya)

SQL Statement: (Using Inner Join, Group by, Having, Sum(), and Order by)

```
SELECT bp.ProductName, SUM(ol.OrderedQuantity) AS total_units_orders,
SUM(ol.OrderedQuantity * bp.Price) AS total_amount
FROM Orders o
INNER JOIN AKHAIRE3.orderline ol ON o.OrderNumber = ol.OrderNumber
INNER JOIN ASAVALI2.beauty_products bp ON ol.ProductID = bp.ProductID
WHERE bp.ProductType = 'M'
GROUP BY bp.ProductName
HAVING SUM(ol.OrderedQuantity * o.OrderAmount) > 200
ORDER BY total_amount DESC;
```



SQL Output:

	PRODUCTNAME	TOTAL_UNITS_ORDERS	TOTAL_AMOUNT
1	Facial Cleanser	3	26.97
2	Lipstick	2	19.98

6.Display a list of orders with their order status, indicating whether they are pending or completed. (Bhumika Suvagia)

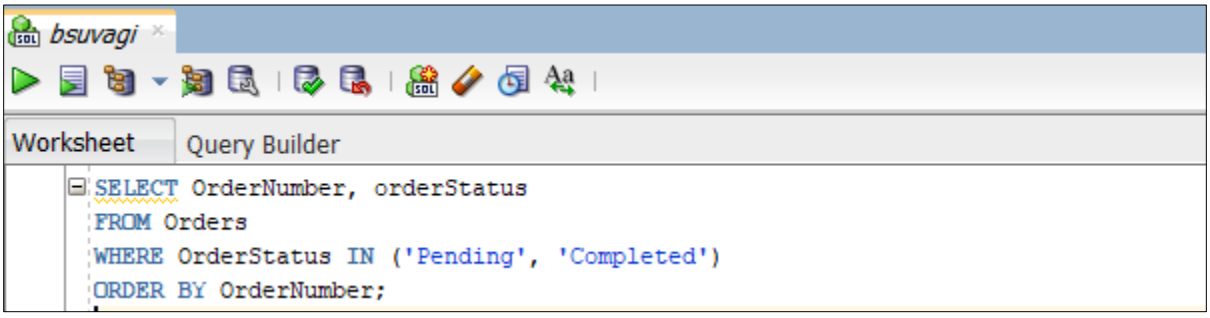
SQL Statement: (Using Order by)

SELECT OrderNumber, orderStatus

FROM Orders

WHERE OrderStatus IN ('Pending', 'Completed')

ORDER BY OrderNumber;



SQL Output:

	ORDERNUMBER	ORDERSTATUS
1	1001	Pending
2	1002	Completed
3	1004	Pending
4	1005	Completed

7.Display Top 3 Employees by Salary (Ankita Savaliya)

SQL Statement: (Using Common Type Expression and Subquery)

WITH CTE AS(

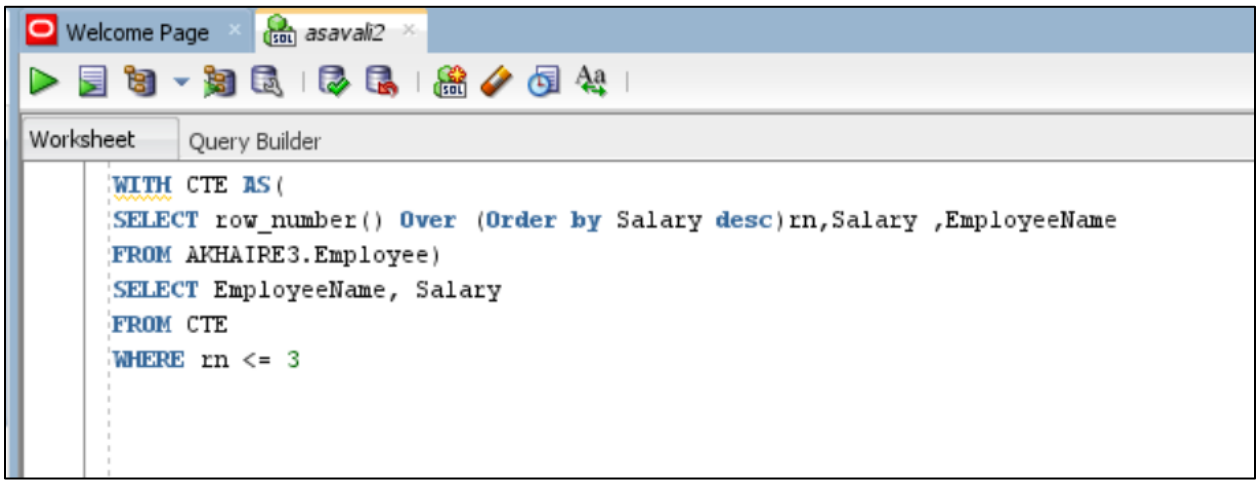
SELECT row_number() Over (Order by Salary desc)rn,Salary ,EmployeeName

FROM AKHAIRE3.Employee)

SELECT EmployeeName, Salary

FROM CTE

WHERE rn <= 3



SQL Output:

	EMPLOYEENAME	SALARY
1	Teresa Butler	56000
2	Jeniffer Smith	50000
3	Gloria Jenkins	45000

8.List Products details Whether it is Manufactured or Purchased? (Ankita Savaliya)

SQL Statement: (Using Union, Inner Join, Subquery, and Order by)

SELECT

B.ProductName,Case WHEN ProductType = 'M' then 'Manufactured' WHEN ProductType = 'P' then 'Purchased' END ProductType,A.company

FROM

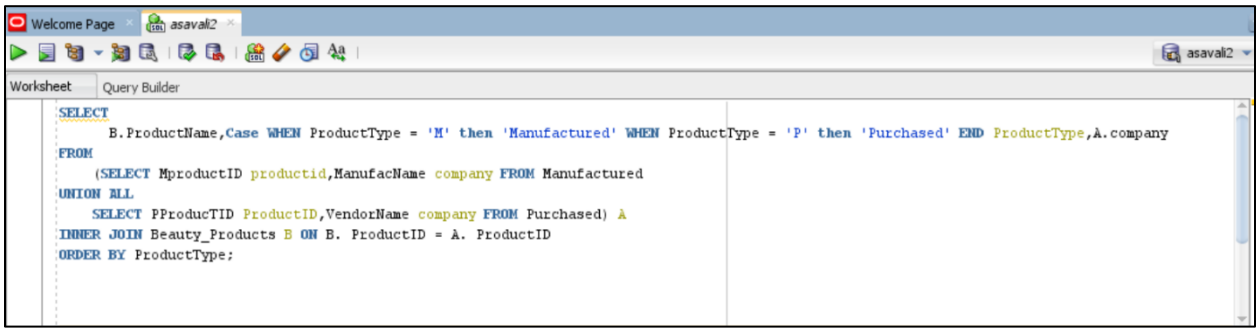
(SELECT MproductID productid,ManufacName company FROM Manufactured

UNION ALL

SELECT PProducTID ProductID,VendorName company FROM Purchased) A

INNER JOIN Beauty_Products B ON B. ProductID = A. ProductID

ORDER BY ProductType;



SQL Output:

	PRODUCTNAME	PRODUCTTYPE	COMPANY
1	Sunscreen	Manufactured	AB Beauty Products
2	Facial Cleanser	Manufactured	AB Beauty Products
3	Lipstick	Manufactured	AB Beauty Products
4	Mascara	Purchased	Ziya Beauty Stores
5	Hair Spray	Purchased	Sanya E-Beauty

PL/SQL Blocks:

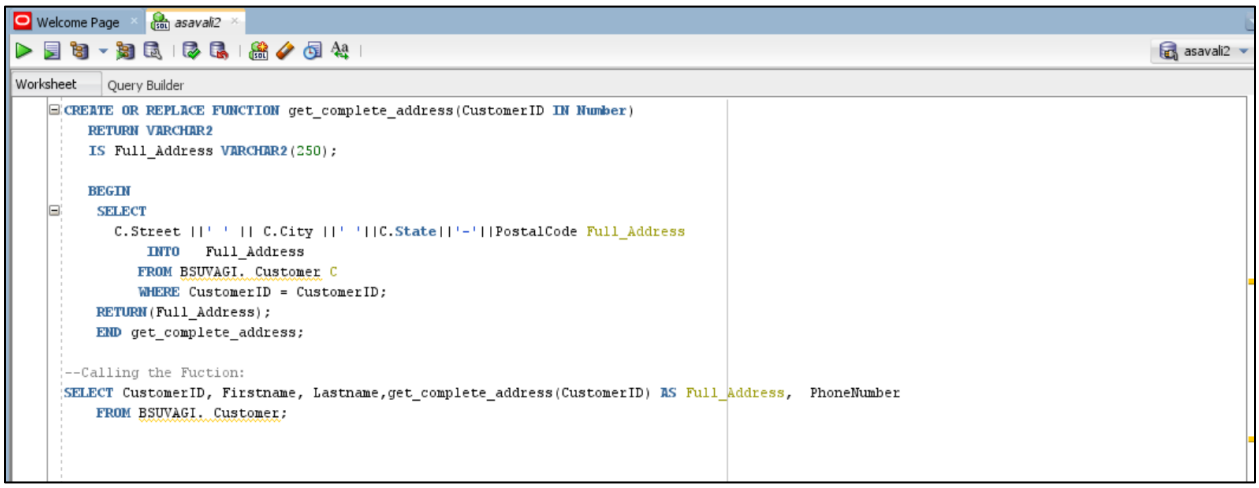
Functions:

1. Function for getting Full address by Customer ID: (Ankita Savaliya)

```
CREATE OR REPLACE FUNCTION get_complete_address(CustomerID IN Number)
  RETURN VARCHAR2
  IS Full_Address VARCHAR2(250);
  BEGIN
   SELECT
          C.Street ||' ' || C.City ||' '||C.State||'-'||PostalCode Full_Address
          INTO   Full_Address
         FROM BSUVAGI. Customer C
         WHERE CustomerID = CustomerID;
   RETURN(Full_Address);
   END get_complete_address;
```

Calling the Fuction:

SELECT CustomerID, Firstname, Lastname,get_complete_address(CustomerID) AS Full_Address, PhoneNumber FROM BSUVAGI. Customer;



Output:

	CUSTOMERID	FIRSTNAME	LASTNAME	FULL_ADDRESS	PHONENUMBER
1	1	Bhumika	Suvagia	5022 S Power RD Mesa AZ-85212	555-1234
2	2	Ankita	Savaliya	3935 Grand Ave Chino Hills CA-91710	555-5678
3	3	Akanksha	Khair	2219 Park Ave Tustin CA-92782	555-9876
4	4	Sara	Lee	200 Towne Center Blvd Sanford FL-32771	555-5555
5	5	David	Nguyen	1201 F St Washington DC-20004	555-4321
6	6	Jenny	Sammy	356 OLD STEESE HWY FAIRBANKS AK-99701	(907) 459-2355

2. Function for getting Order Status by Order ID: (Bhumika Suvagia)

```
CREATE OR REPLACE FUNCTION get_order_status
  (OrderID IN orders.orderNumber%TYPE)
  RETURN orders.orderstatus%TYPE
IS
  v_order_status orders.orderstatus%TYPE;
BEGIN
  SELECT Orderstatus INTO v_order_status
  FROM orders WHERE orderNumber = OrderID
  FETCH FIRST 1 ROWS ONLY;
  RETURN v_order_status;
END get_order_status;
```

```
Function GET_ORDER_STATUS compiled
```

-- Declare a bind variable

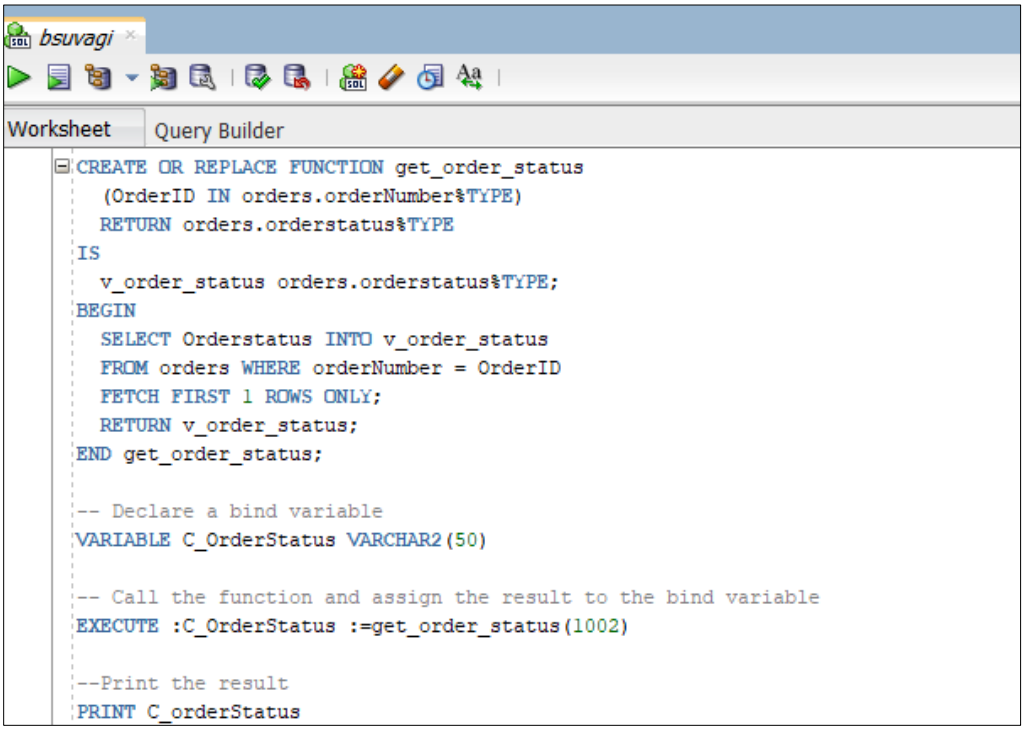
VARIABLE C_OrderStatus VARCHAR2(50)

-- Call the function and assign the result to the bind variable

EXECUTE :C_OrderStatus :=get_order_status(1002)

--Print the result

PRINT C_orderStatus

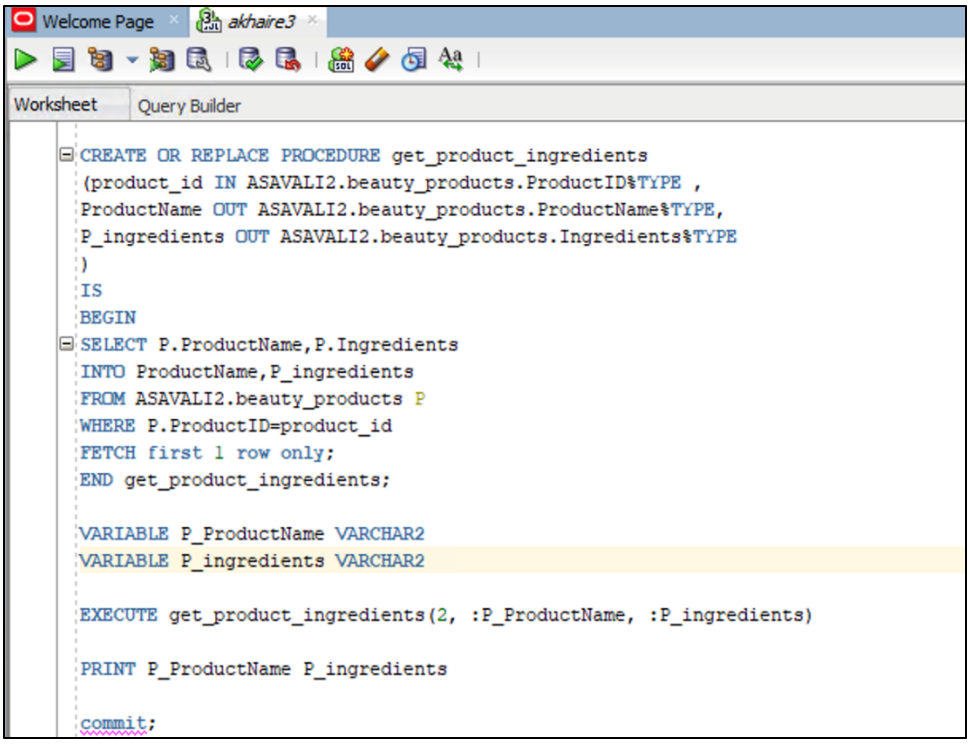


Output:

```
C_ORDERSTATUS
--------------------------------------------------
Completed
```

3. Function for checking the product availability by Product Name: (Akanksha Khaire)

```
CREATE OR REPLACE FUNCTION check_product_availability(ProductName IN
VARCHAR2)
RETURN VARCHAR2
IS
v_available VARCHAR2(10);
BEGIN
SELECT 'Available' INTO v_available
FROM ASAVALI2.beauty_products
WHERE productname = ProductName
FETCH FIRST 1 ROW ONLY;
RETURN v_available;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RETURN 'Not Available';
END;

VARIABLE P_status VARCHAR2(20);
EXEC :P_status := check_product_availability('Sunscreen');
PRINT P_status;
```




```
Function CHECK_PRODUCT_AVAILABILITY compiled
```

Output:

```
PL/SQL procedure successfully completed.


P_STATUS
--------------------------------------------------------------------------------
Available
```

4. Function for Getting Salary by Employee ID: (Ankita Savaliya)

```
CREATE OR REPLACE FUNCTION get_employee_salary
  (employee_id IN AKHAIRE3.employee.EmployeeID%TYPE)
  RETURN NUMBER
IS
  emp_salary AKHAIRE3.employee.Salary%TYPE:=0;
BEGIN
```

```
  SELECT Salary INTO emp_salary
  FROM AKHAIRE3.employee WHERE EmployeeID = employee_id;
RETURN emp_salary;
END get_employee_salary;
```


```
Function GET_EMPLOYEE_SALARY compiled
```

```
-- Declare a bind variable
VARIABLE b_Employee_Salary NUMBER

- Call the function and assign the result to the bind variable
exec :b_Employee_Salary :=get_employee_salary(2)

--Print the result
PRINT b_Employee_Salary
```

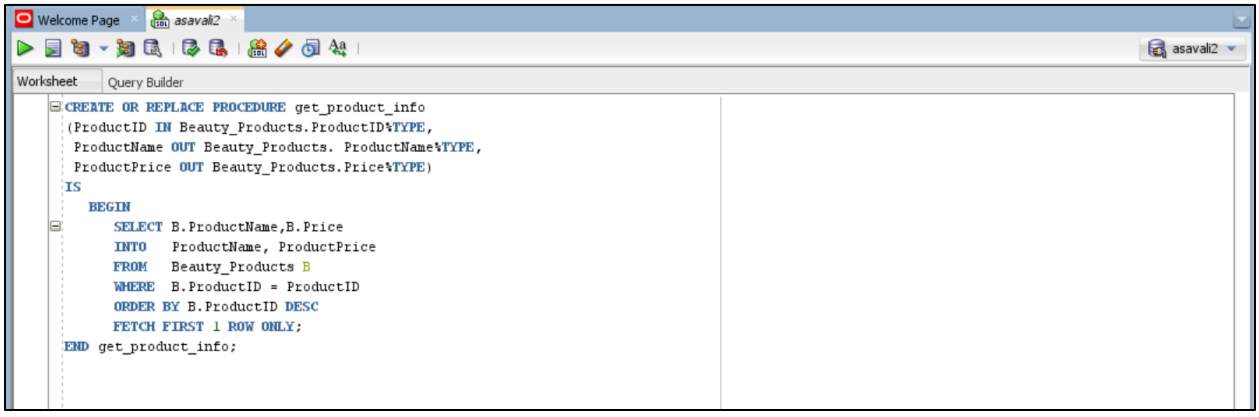
Output:

```
B_EMPLOYEE_SALARY
-----------------
            30000
```

Procedures:

1. Procedure for getting Product Information by Product ID: (Ankita Savaliya)

```
CREATE OR REPLACE PROCEDURE get_product_info
(ProductID IN Beauty_Products.ProductID%TYPE,
 ProductName OUT Beauty_Products. ProductName%TYPE,
 ProductPrice OUT Beauty_Products.Price%TYPE)
IS
  BEGIN
     SELECT B.ProductName,B.Price
     INTO   ProductName, ProductPrice
     FROM   Beauty_Products B
     WHERE  B.ProductID = ProductID
     ORDER BY B.ProductID DESC
     FETCH FIRST 1 ROW ONLY;
END get_product_info;
```



```
Procedure GET_PRODUCT_INFO compiled
```

--Declare a bind variable

VARIABLE B_ProductName VARCHAR2(100)

VARIABLE B_ProductPrice NUMBER

-- Call the function and assign the result to the bind variable

EXECUTE get_product_info(1, :B_ProductName, :B_ProductPrice)

--Print the result

PRINT B_ProductName B_ProductPrice

Output:

```
B_PRODUCTNAME
--------------------------------------------------------------------------------
Sunscreen

B_PRODUCTPRICE
--------------
         15.99
```

2. Procedure for getting Customer Order Information by Customer ID: (Bhumika Suvagia)

CREATE OR REPLACE PROCEDURE get_customer_orders(

CustID IN Customer.CustomerID%TYPE,

OrderNumber OUT orders.ordernumber%TYPE,

```
    OrderedQuantity OUT NUMBER,
    OrderAmount OUT NUMBER
)
IS
BEGIN
    SELECT
        O.OrderNumber,
        SUM(Ol.OrderedQuantity) AS OrderedQuantity,
        SUM(p.price * ol.orderedquantity) AS OrderAmount
    INTO OrderNumber, OrderedQuantity, OrderAmount
    FROM orders O
    INNER JOIN AKHAIRE3.orderline Ol ON O.OrderNumber = Ol.OrderNumber
    INNER JOIN ASAVALI2.beauty_products P ON Ol.ProductID = P.ProductID
    WHERE O.CustomerID = CustID
    GROUP BY O.OrderNumber
    ORDER BY O.OrderNumber DESC
    FETCH FIRST 1 ROWS ONLY;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            OrderNumber := '0';
            OrderedQuantity := '0';
            OrderAmount := '0.00';
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLCODE || ' - ' || SQLERRM);
            OrderNumber := NULL;
            OrderedQuantity := NULL;
            OrderAmount := NULL;
END get_customer_orders;
```

```
Procedure GET_CUSTOMER_ORDERS compiled
```

-- Declare a bind variable

VARIABLE C_OrderNumber NUMBER

VARIABLE C_OrderedQuantity NUMBER

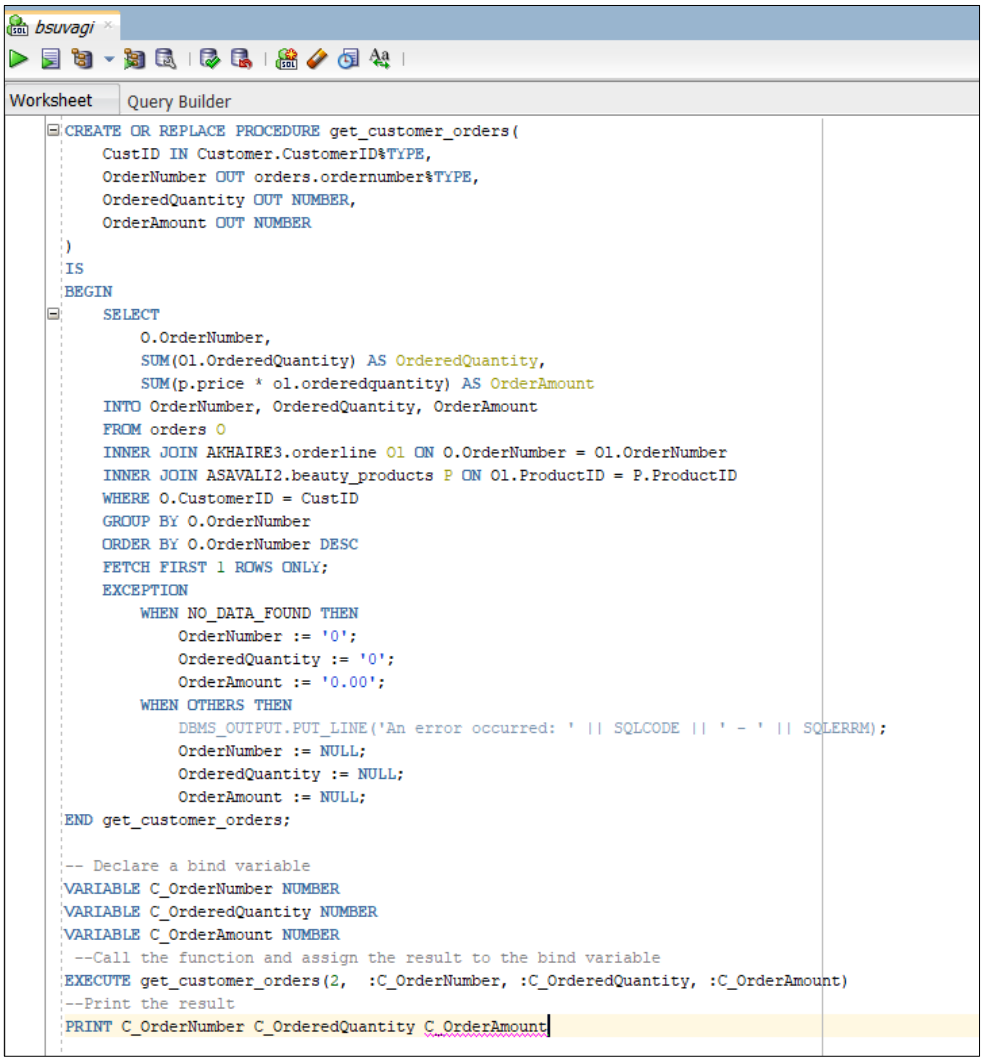
VARIABLE C_OrderAmount NUMBER

--Call the function and assign the result to the bind variable

EXECUTE get_customer_orders(2, :C_OrderNumber, :C_OrderedQuantity, :C_OrderAmount)

--Print the result

PRINT C_OrderNumber C_OrderedQuantity C_OrderAmount

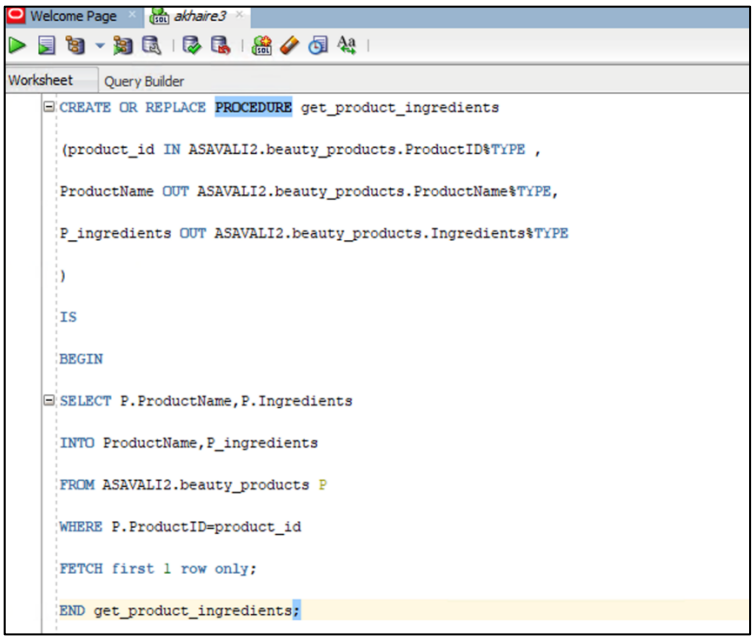


Output:

```
C_ORDERNUMBER
-------------
         1002

C_ORDEREDQUANTITY
-----------------
                5

C_ORDERAMOUNT
-------------
        40.95
```

3. Procedure for getting the product ingredients by Product ID: (Akanksha Khaire)

```
CREATE OR REPLACE PROCEDURE get_product_ingredients
(product_id IN ASAVALI2.beauty_products.ProductID%TYPE ,
ProductName OUT ASAVALI2.beauty_products.ProductName%TYPE,
P_ingredients OUT ASAVALI2.beauty_products.Ingredients%TYPE
)
IS
BEGIN
SELECT P.ProductName,P.Ingredients
INTO ProductName,P_ingredients
FROM ASAVALI2.beauty_products P
WHERE P.ProductID=product_id
FETCH first 1 row only;
END get_product_ingredients;
```



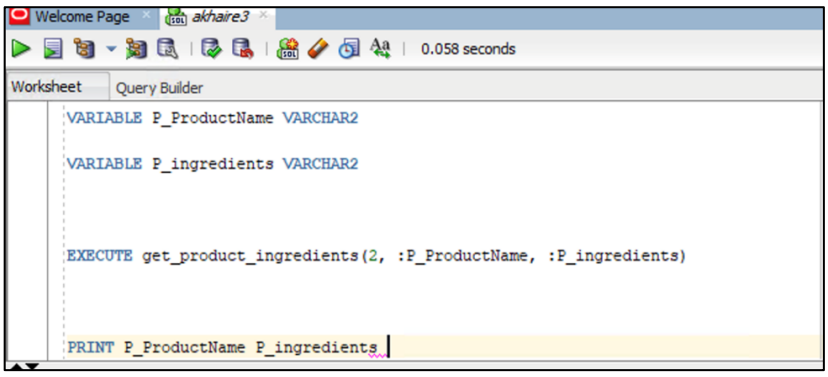

```
PL/SQL procedure successfully completed.
```

VARIABLE P_ProductName VARCHAR2

VARIABLE P_ingredients VARCHAR2

EXECUTE get_product_ingredients(2, :P_ProductName, :P_ingredients)

PRINT P_ProductName P_ingredients



Output:

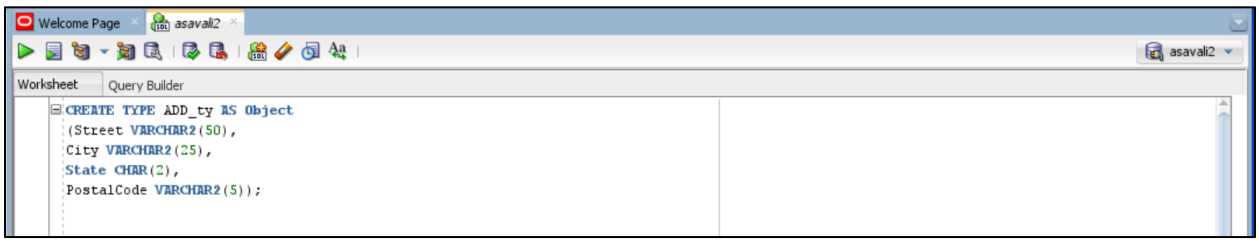
```
P_PRODUCTNAME
--------------------------------------------------------------------------------
Mascara

P_INGREDIENTS
--------------------------------------------------------------------------------
Vitamin E, jojoba oil
```

Object-Relational Database Management System (ORDBMS):

Create Customer information table with user defined object types and display customer full name using user-define method. (Ankita Savaliya)

This creates a type named ADD_ty that contains four attributes: Street, City, State, and PostalCode

```
CREATE TYPE ADD_ty AS Object
(Street VARCHAR2(50),
City VARCHAR2(25),
State CHAR(2),
PostalCode VARCHAR2(5));
```

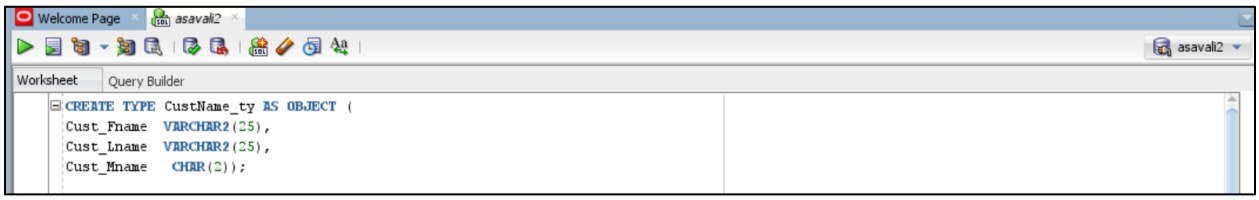



```
Type ADD_TY compiled
```

This creates a type named CustName_ty that contains three attributes: Cust_Fname, Cust_Lname, and Cust_Mname

```
CREATE TYPE CustName_ty AS OBJECT (
```

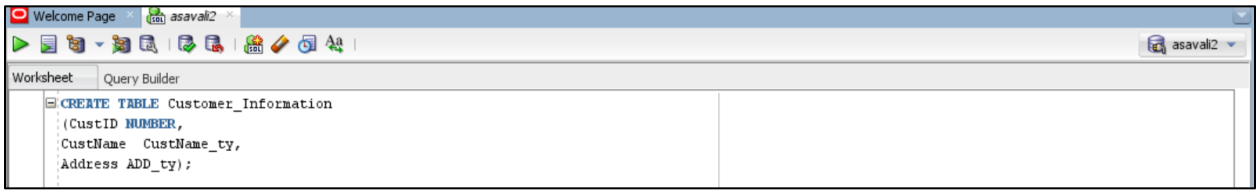
```
Cust_Fname    VARCHAR2(25),
Cust_Lname    VARCHAR2(25),
Cust_Mname    CHAR(2));
```




```
Type CUSTNAME_TY compiled
```

This creates a table named Customer_Information with three columns: CustID, CustName, and Address

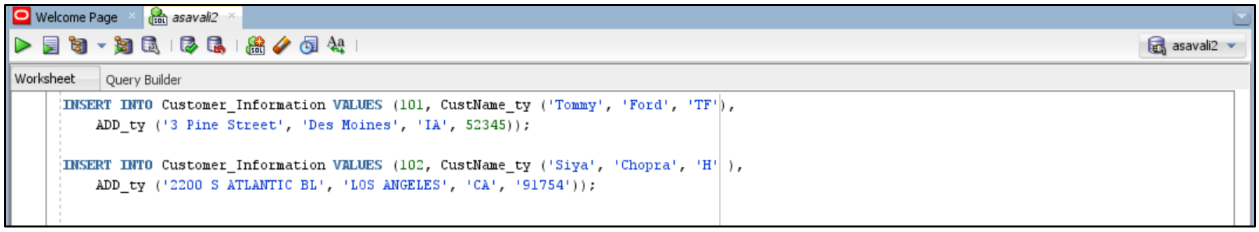
```
CREATE TABLE Customer_Information
(CustID NUMBER,
CustName  CustName_ty,
Address ADD_ty);
```




```
Table CUSTOMER_INFORMATION created.
```

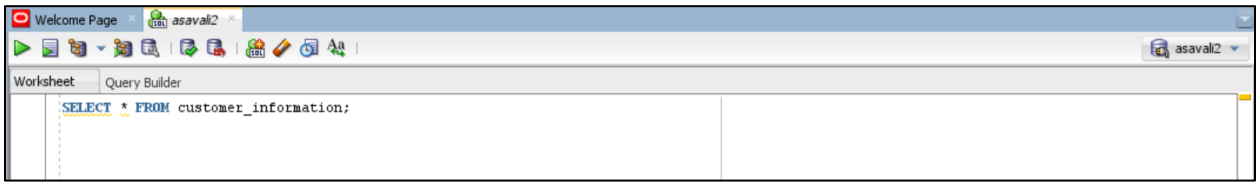
Insert value using constructor functions for the CustName_ty and ADD_ty object types

```
INSERT INTO Customer_Information VALUES (101, CustName_ty ('Tommy', 'Ford', 'TF'),
ADD_ty ('3 Pine Street', 'Des Moines', 'IA', 52345));
INSERT INTO Customer_Information VALUES (102, CustName_ty ('Siya', 'Chopra', 'H' ),
ADD_ty ('2200 S ATLANTIC BL', 'LOS ANGELES', 'CA', '91754'));
```



Display all the information in customer_information table

SELECT * FROM customer_information;

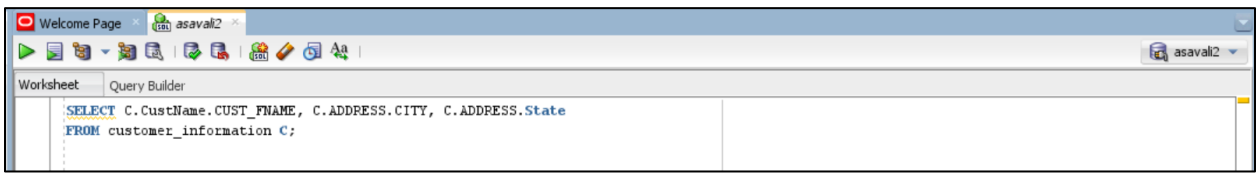



```
    CUSTID
----------
CUSTNAME(CUST_FNAME, CUST_LNAME, CUST_MNAME)
--------------------------------------------------------------------------------
ADDRESS(STREET, CITY, STATE, POSTALCODE)
--------------------------------------------------------------------------------
       101
CUSTNAME_TY('Tommy', 'Ford', 'TF')
ADD_TY('3 Pine Street', 'Des Moines', 'IA', '52345')

       102
CUSTNAME_TY('Siya', 'Chopra', 'H ')
ADD_TY('2200 S ATLANTIC BL', 'LOS ANGELES', 'CA', '91754')
```

Retrieve specific information from the customer_information table

SELECT C.CustName.CUST_FNAME, C.ADDRESS.CITY, C.ADDRESS.State

FROM customer_information C;

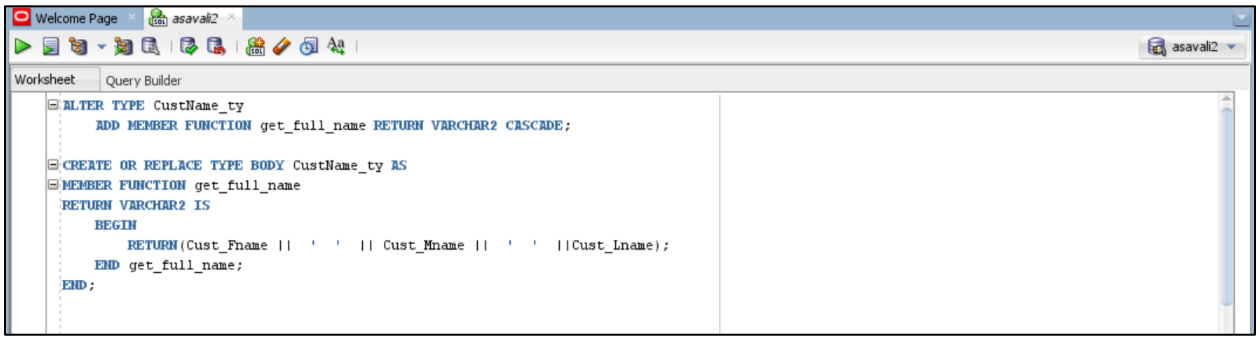


Output:

	CUSTNAME.CUST_FNAME	ADDRESS.CITY	ADDRESS.STATE
1	Tommy	Des Moines	IA
2	Siya	LOS ANGELES	CA

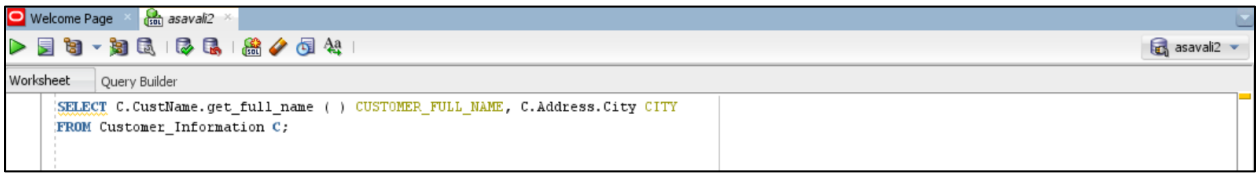
Create a user-defined method reusing CustName_ty object

```
ALTER TYPE CustName_ty
ADD MEMBER FUNCTION get_full_name RETURN VARCHAR2 CASCADE;

CREATE OR REPLACE TYPE BODY CustName_ty AS
MEMBER FUNCTION get_full_name
RETURN VARCHAR2 IS
        BEGIN
                RETURN(Cust_Fname || ' ' || Cust_Mname || ' ' ||Cust_Lname);
        END get_full_name;
END;
```

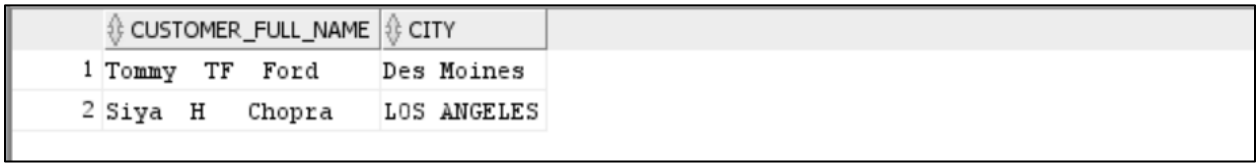



```
Type CUSTNAME_TY altered.


Type Body CUSTNAME_TY compiled
```

SELECT C.CustName.get_full_name ( ) CUSTOMER_FULL_NAME, C.Address.City CITY
FROM Customer_Information C;



Output:



	CUSTOMER_FULL_NAME	CITY
1	Tommy TF Ford	Des Moines
2	Siya H Chopra	LOS ANGELES

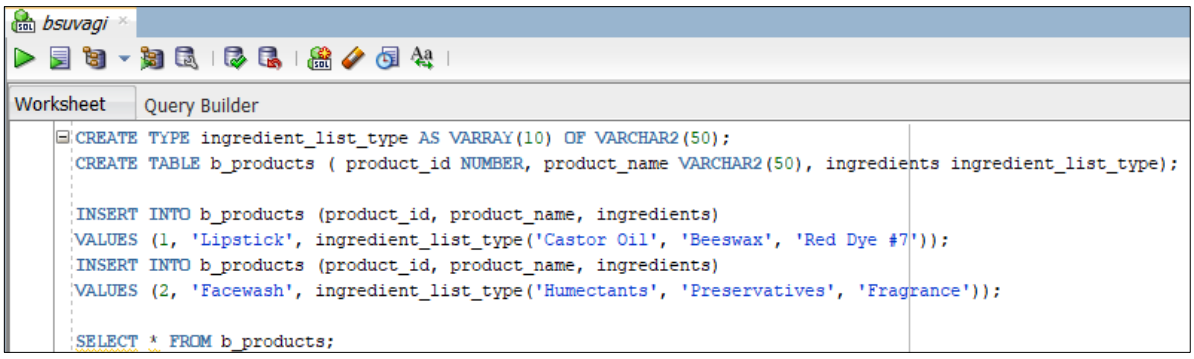
VARRAY:

Inserting product ingredients: (Bhumika Suvagia)

CREATE TYPE ingredient_list_type AS VARRAY(10) OF VARCHAR2(50);
CREATE TABLE b_products (product_id NUMBER, product_name VARCHAR2(50),
ingredients ingredient_list_type);

INSERT INTO b_products (product_id, product_name, ingredients)
VALUES (1, 'Lipstick', ingredient_list_type('Castor Oil', 'Beeswax', 'Red Dye #7'));
INSERT INTO b_products (product_id, product_name, ingredients)
VALUES (2, 'Facewash', ingredient_list_type('Humectants', 'Preservatives', 'Fragrance'));

SELECT * FROM b_products;

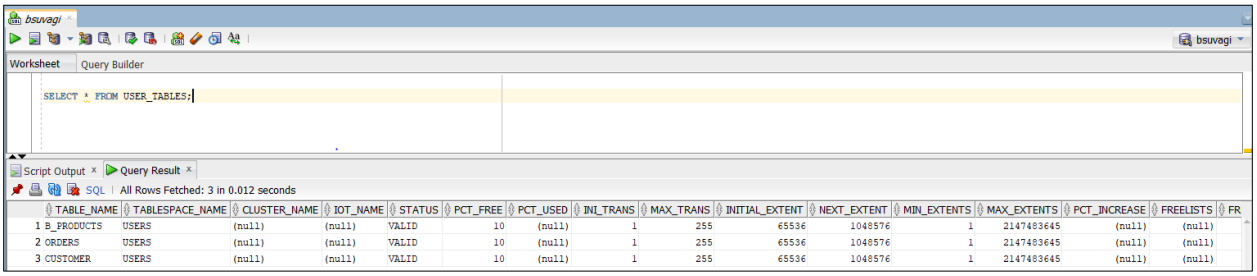


Output:

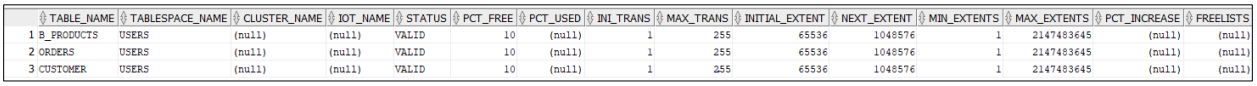
```
PRODUCT_ID PRODUCT_NAME
---------- --------------------------------------------------
INGREDIENTS
--------------------------------------------------------------------------------
         2 Facewash
INGREDIENT_LIST_TYPE('Humectants', 'Preservatives', 'Fragrance')
```

List objects in User Account:

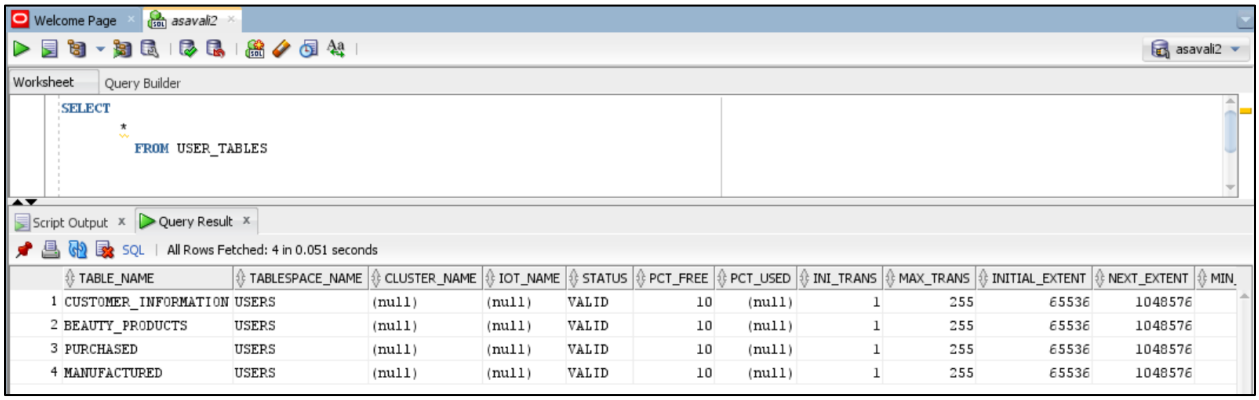
SELECT * FROM USER_TABLES;



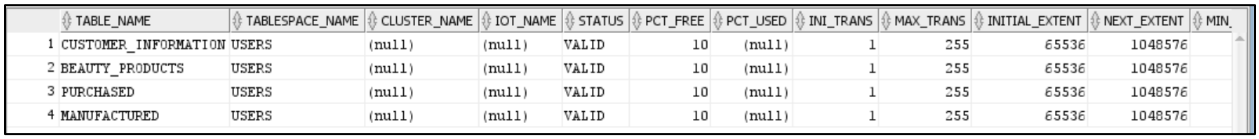
	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	PCT_INCREASE	FREELISTS	FR
1	B_PRODUCTS	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	1	2147483645	(null)	(null)	
2	ORDERS	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	1	2147483645	(null)	(null)	
3	CUSTOMER	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	1	2147483645	(null)	(null)	



	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	PCT_INCREASE	FREELISTS
1	B_PRODUCTS	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	1	2147483645	(null)	(null)
2	ORDERS	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	1	2147483645	(null)	(null)
3	CUSTOMER	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	1	2147483645	(null)	(null)



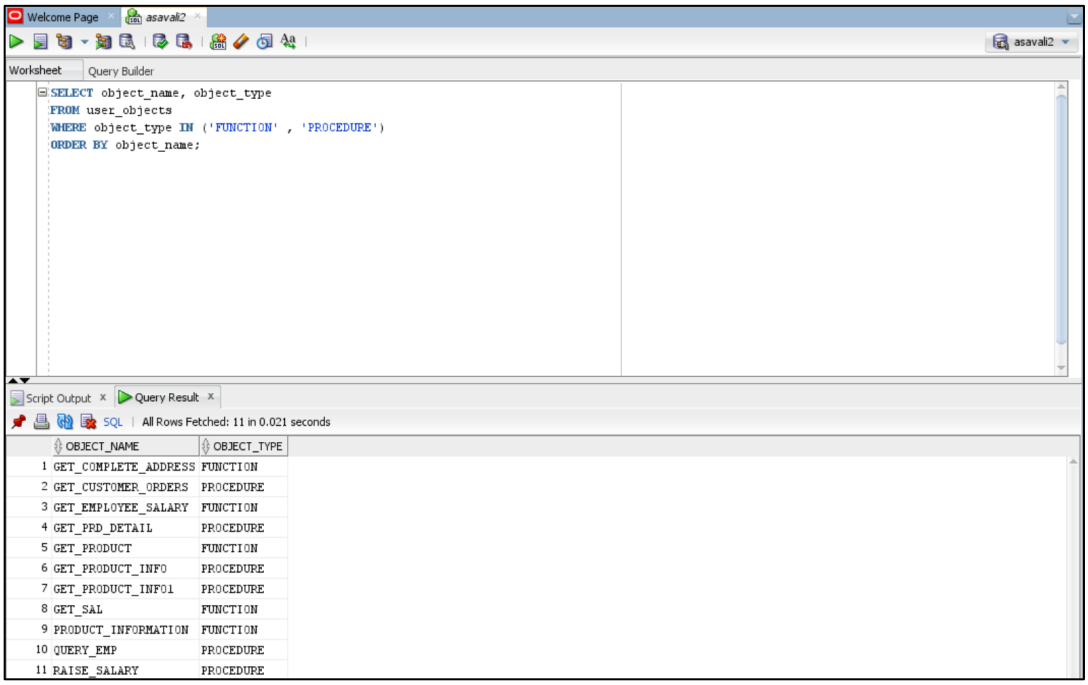
	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT	MIN
1	CUSTOMER_INFORMATION	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	
2	BEAUTY_PRODUCTS	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	
3	PURCHASED	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	
4	MANUFACTURED	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	



	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT	MIN
1	CUSTOMER_INFORMATION	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	
2	BEAUTY_PRODUCTS	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	
3	PURCHASED	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	
4	MANUFACTURED	USERS	(null)	(null)	VALID	10	(null)	1	255	65536	1048576	

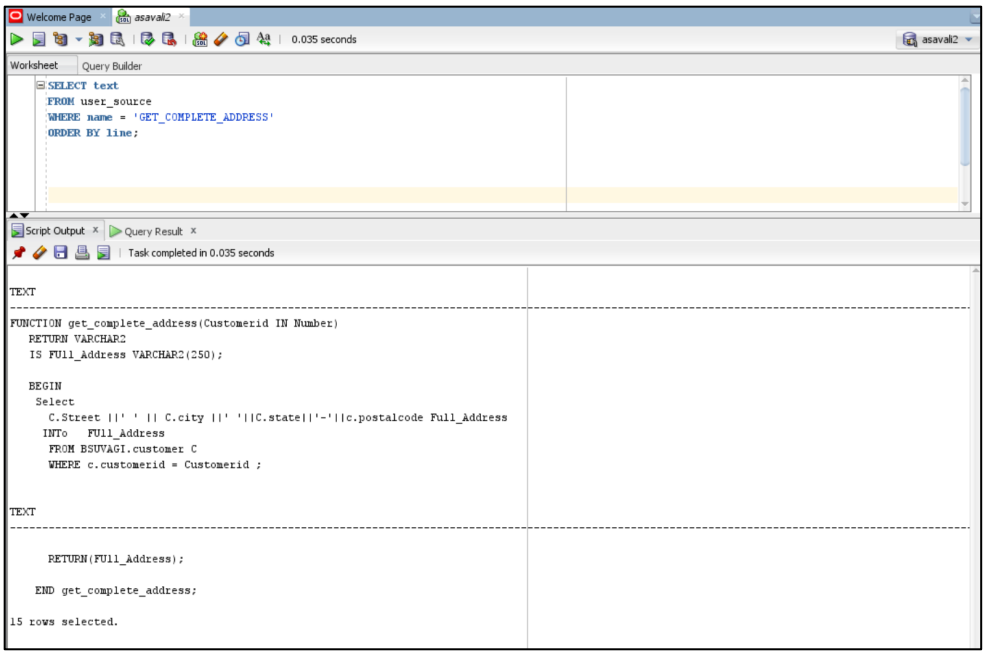
List all Procedures and Functions:

```
SELECT object_name, object_type
FROM user_objects
WHERE object_type IN ('FUNCTION' , 'PROCEDURE')
ORDER BY object_name;
```

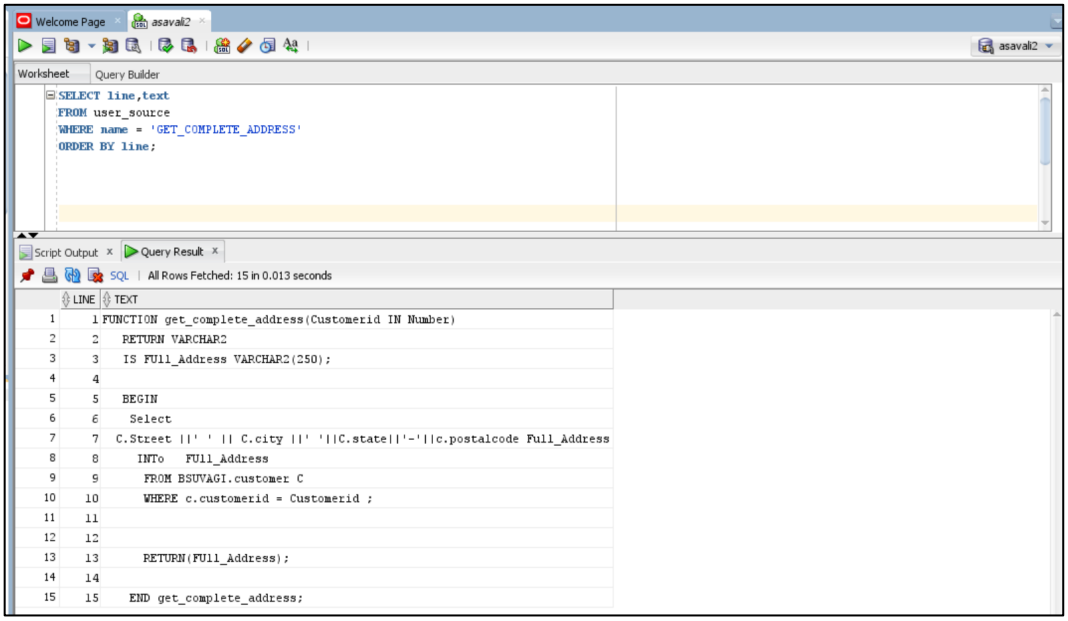


List the code of Procedures and Functions:

```
SELECT text
FROM user_source
WHERE name = 'GET_COMPLETE_ADDRESS'
ORDER BY line;
```


Welcome Page
asavali2
0.035 seconds
asavali2
Worksheet
Query Builder
SELECT text
FROM user_source
WHERE name = 'GET_COMPLETE_ADDRESS'
ORDER BY line;
Script Output
Query Result
Task completed in 0.035 seconds
TEXT
FUNCTION get_complete_address(Customerid IN Number)
RETURN VARCHAR2
IS Full_Address VARCHAR2(250);
BEGIN
Select
C.Street ||' ' || C.city ||' '||C.state||'-'||c.postalcode Full_Address
INTo Full_Address
FROM BSUVAGI.customer C
WHERE c.customerid = Customerid ;
TEXT
RETURN(Full_Address);
END get_complete_address;
15 rows selected.

Code by Line:


Welcome Page
asavali2
asavali2
Worksheet
Query Builder
SELECT line,text
FROM user_source
WHERE name = 'GET_COMPLETE_ADDRESS'
ORDER BY line;
Script Output
Query Result
SQL
All Rows Fetched: 15 in 0.013 seconds
LINE
TEXT
1 FUNCTION get_complete_address(Customerid IN Number)
2 RETURN VARCHAR2
3 IS Full_Address VARCHAR2(250);
4
5 BEGIN
6 Select
7 C.Street ||' ' || C.city ||' '||C.state||'-'||c.postalcode Full_Address
8 INTo Full_Address
9 FROM BSUVAGI.customer C
10 WHERE c.customerid = Customerid ;
11
12
13 RETURN(Full_Address);
14
15 END get_complete_address;