

Group Project

ELECTRONIC APPLIANCE DATABASE

Ryan Lam, Sahithya Vallabhaneni, Honesty Werinwo, Zhirat

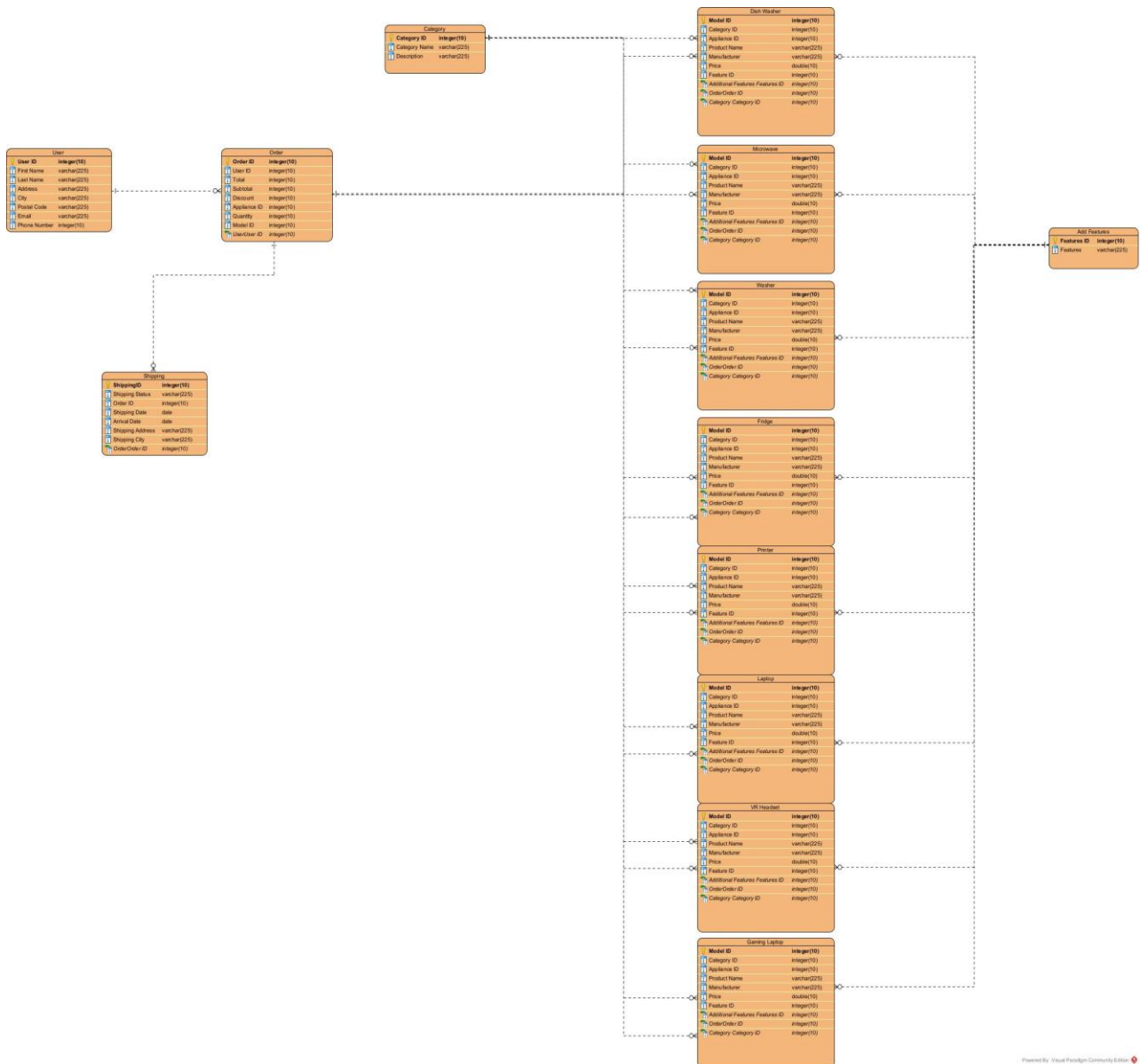
Software Engineering Technology/Technician, Centennial College

COMP214: Advanced Database

Professor Z. Malik

Introduction

Our database solution aims to streamline the management of electronic appliances and orders for a fictional electronics appliance store, that models a real-life electronic appliance store. We used Home Depot – www.homedepot.com.ca for our problem domain. The key components of the database project include: 13 Tables (EA_USER, EA_ORDER, EA_CATEGORY, EA_SHIPPING, EA_ADDFEATURES, EA_MICROWAVE, EA_PRINTER, EA_LAPTOP, EA_GAMINGLAPTOP, EA_WASHER, EA_DISHWASHER, EA_FRIDGE, EA_VRHEADSET), ER diagram, Sequence, Indexes, Triggers and Packages (procedures and functions). In this database, the EA_USER table is used to store all the information of the user. Orders from the user will be stored into the EA_ORDER table. Moreover, the shipping information of the orders will be stored in the EA_SHIPPING table. To classify each product, the EA_CATEGORY and EA_ADDFEATURES table is implemented to specify what kind of special features that the product have. Here is the ERD diagram that show the detailed relationship of each table.



Tables

EA_USER

UserID	FirstName	LastName	Address	City	PostalCode	Email	PhoneNumber
1001	John	Doe	123 Main St	New York	A1B2C3	john.doe@example.com	1234567890
1002	Jane	Smith	456 Maple Ave	Los Angeles	G1H2A3	jane.smith@example.com	9876543210
1003	Michael	Johnson	789 Oak Rd	Chicago	E4H0H2	michael.johnson@example.com	5678901234
1005	David	Brown	888 Elm Blvd	Houston	L1P0H2	david.brown@example.com	9012345678
1006	Sarah	Lee	222 Birch Ln	Miami	M1P2A1	sarah.lee@example.com	3456789012
1007	James	Jones	333 Cedar Dr	Dallas	T0E2P8	james.jones@example.com	6789012345
1008	Emma	Wilson	444 Spruce Ave	Seattle	K1H2A6	emma.wilson@example.com	8901234567
1009	Andrew	Taylor	777 Aspen Rd	Atlanta	A2H1B2	andrew.taylor@example.com	2345678901
1010	Olivia	Anderson	999 Birch Ln	Denver	E2H2A2	olivia.anderson@example.com	4567890123
1004	Emily	Williams	555 Pine St	Toronto	M2K0H1	emily.williams@example.com	7890123456

EA_ORDER

OrderID	UserID	Total	Subtotal	Discount	ApplianceID	Quantity	ModelID
10000001	1001	500	450	50	100003	2	1001
10000002	1002	800	720	80	200005	1	1002
10000003	1003	350	300	50	200010	3	1003
10000004	1004	900	800	100	300004	1	1004
10000005	1005	600	500	100	400010	2	1005
10000006	1006	200	180	20	500009	1	1006
10000007	1007	750	700	50	600010	3	1007
10000008	1008	400	350	50	700005	1	1008
10000009	1009	300	270	30	800008	2	1009
10000010	1010	650	600	50	100004	2	1010

EA_CATEGORY

CategoryID	CategoryName	CategoryDescription
1	KitchenAppliance	Appliances used in the kitchen for cooking and food preparation.
2	HomeAppliance	Appliances used in household chores and maintenance.
3	OfficeAppliance	Appliances used in office settings for productivity and organization.
4	GamingAppliance	Appliances used for gaming and entertainment purposes.

EA_Shipping

ShippingID	ShippingStatus	OrderID	ShippingDate	ArrivalDate	ShippingAddress	ShippingCity
101	Shipped	10000001	27-JUN-23	30-JUN-23	123 Main Street	Toronto
102	Out for Delivery	10000002	28-JUN-23	01-JUL-23	456 Elm Avenue	Vancouver
103	Delivered	10000003	29-JUN-23	02-JUL-23	789 Oak Road	Montreal
104	In Transit	10000004	30-JUN-23	03-JUL-23	321 Pine Court	Calgary
105	Shipped	10000005	01-JUL-23	04-JUL-23	987 Maple Lane	Ottawa
106	Out for Delivery	10000006	02-JUL-23	05-JUL-23	654 Cedar Street	Edmonton
107	Delivered	10000007	03-JUL-23	06-JUL-23	234 Birch Avenue	Quebec City
108	In Transit	10000008	04-JUL-23	07-JUL-23	876 Willow Place	Winnipeg
109	Shipped	10000009	05-JUL-23	08-JUL-23	543 Fir Drive	Halifax
110	Out for Delivery	10000010	06-JUL-23	09-JUL-23	789 Cedar Court	Saskatoon
101	Shipped	10000001	27-JUN-23	30-JUN-23	123 Main Street	Toronto

EA_MICROWAVE

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
100001	1	101	Compact Microwave	Panasonic	200	1
100002	1	102	Countertop Microwave	LG	150	2
100003	1	103	Over-the-Range Microwave	Whirlpool	300	1
100004	1	104	Built-in Microwave	LG	380	1
100005	1	105	Smart Microwave	Panasonic	250	2
100006	1	106	Microwave with Grill	LG	280	1
100007	1	107	Microwave with Convection	Whirlpool	330	1
100008	1	108	Retro-Style Microwave	LG	200	2
100009	1	109	Compact Microwave	Panasonic	160	1
100010	1	110	Family-Sized Microwave	LG	350	2

EA_PRINTER

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
200001	3	201	Inkjet Printer	hp	100	1
200002	3	202	Laser Printer	Canon	250	3
200003	3	203	All-in-One Printer	Xerox	180	1
200004	3	204	Wireless Printer	Brother	130	3
200005	3	205	Color Printer	Epson	150	1
200006	3	206	Photo Printer	hp	200	3
200007	3	207	Portable Printer	Canon	90	1
200008	3	208	High-Speed Printer	Xerox	300	3
200009	3	209	Large Format Printer	Brother	600	1
200010	3	210	Multifunction Printer	Epson	190	3

EA_FRIDGE

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
300001	1	301	Top Freezer Refrigerator	LG	700	1
300002	1	302	Bottom Freezer Refrigerator	General Electric	900	2
300003	1	303	French Door Refrigerator	KitchenAid	1300	1
300004	1	304	Side-by-Side Refrigerator	Whirlpool	1200	2
300005	1	305	Compact Refrigerator	Panasonic	200	1
300006	1	306	Counter-Depth Refrigerator	LG	1500	2
300007	1	307	Smart Refrigerator	General Electric	1600	1
300008	1	308	Mini Fridge	KitchenAid	130	2
300009	1	309	Freezerless Refrigerator	Whirlpool	900	1
300010	1	310	Convertible Refrigerator	Panasonic	1000	2

EA_LAPTOP

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
400001	3	401	Ultra-Thin Laptop	Lenovo	1000	1
400002	3	402	Gaming Laptop	HP	1900	3
400003	3	403	2-in-1 Convertible Laptop	Acer	1300	1
400004	3	404	Business Laptop	Asus	1500	3
400005	3	405	Budget Laptop	Dell	600	1
400006	3	406	High-Performance Laptop	HP	1600	3
400007	3	407	Touchscreen Laptop	Asus	900	1
400008	3	408	MacBook	HP	2000	3
400009	3	409	Chromebook	Asus	400	1
400010	3	410	Lightweight Laptop	HP	800	3

EA_VRHEADSET

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
500001	4	501	Wireless VR Headset	Sony	800	1
500002	4	502	Gaming VR Headset	Xbox	1000	3
500003	4	503	High-Resolution VR Headset	Google	1500	1
500004	4	504	Portable VR Headset	Steam	600	3
500005	4	505	Professional VR Headset	Sony	1900	1
500006	4	506	All-in-One VR Headset	Xbox	1200	3
500007	4	507	Budget VR Headset	Google	300	1
500008	4	508	Advanced VR Headset	Steam	1700	3
500009	4	509	Comfortable VR Headset	Sony	900	1
500010	4	510	VR Headset with Motion Controllers	Xbox	1300	3

EA_GAMINGLAPTOP

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
600001	4	601	High-Performance Gaming Laptop	ASUS	2000	1
600002	4	602	Ultra-Slim Gaming Laptop	MSI	1800	3
600003	4	603	RGB Gaming Laptop	Acer	1500	1
600004	4	604	Powerful Gaming Laptop	Lenovo	2300	3
600005	4	605	Affordable Gaming Laptop	Dell	1300	1
600006	4	606	Portable Gaming Laptop	HP	1600	3
600007	4	607	VR-Ready Gaming Laptop	Razer	2500	1

600008	4	608	15-inch Gaming Laptop	Alienware	1700	3
600009	4	609	144Hz Gaming Laptop	ASUS	1900	1
600010	4	610	17-inch Gaming Laptop	MSI	2200	3

EA_WASHER

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
700001	2	701	Front-Load Washing Machine	LG	700	2
700002	2	702	Top-Load Washing Machine	Samsung	600	3
700003	2	703	Smart Washing Machine	Whirlpool	900	2
700004	2	704	Energy-Efficient Washer	Bosch	800	3
700005	2	705	Compact Washing Machine	GE Appliances	500	2
700006	2	706	Large Capacity Washer	Maytag	1000	3
700007	2	707	Quick Wash Washing Machine	Haier	750	2
700008	2	708	Steam Washing Machine	Electrolux	850	3
700009	2	709	Quiet Washing Machine	Frigidaire	900	2
700010	2	710	Automatic Washing Machine	LG	900	3

EA_DISHWASHER

ModelID	CategoryID	ApplianceID	ProductName	Manufacturer	Price	FeatureID
800001	2	711	Smart Dishwasher	Bosch	900	2
800002	2	712	Compact Dishwasher	KitchenAid	600	3
800003	2	713	Energy-Efficient Dishwasher	Whirlpool	800	2
800004	2	714	Stainless Steel Dishwasher	LG	800	3
800005	2	715	Silent Dishwasher	Samsung	700	2
800006	2	716	Large Capacity Dishwasher	GE Appliances	1000	3
800007	2	717	Quick Wash Dishwasher	Maytag	750	2
800008	2	718	Built-In Dishwasher	Electrolux	850	3
800009	2	719	Rack Adjusting Dishwasher	Frigidaire	900	2
800010	2	720	Automatic Dishwasher	LG	900	3

EA_ADDFEATURES

FeaturesID	Features
1001	Wi-Fi Connectable
1002	No Special Feature
1003	Can Connect to Phone

Triggers

After Insert trigger

SQL Code:

```
CREATE OR REPLACE TRIGGER trg_Washer
```

```
AFTER INSERT ON EA_ORDER
```

```
FOR EACH ROW
```

BEGIN

UPDATE EA_ORDER

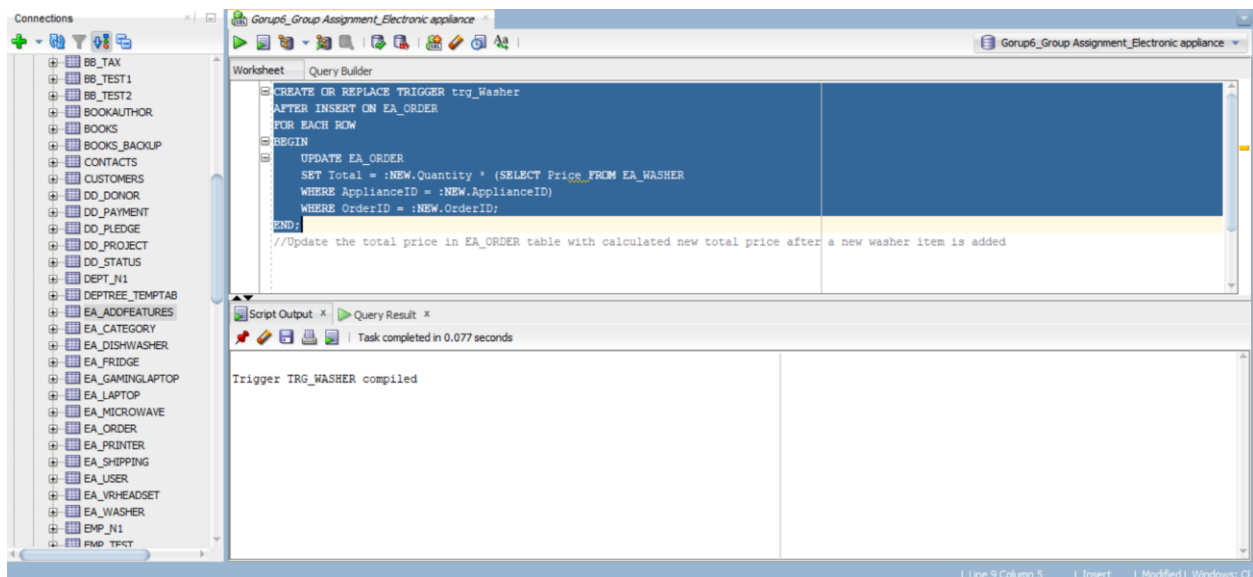
SET Total = :NEW.Quantity * (SELECT Price FROM EA_WASHER

WHERE ApplianceID = :NEW.ApplianceID)

WHERE OrderID = :NEW.OrderID;

END;

//Update the total price in EA_ORDER table with calculated new total price after a new washer item is added



Before Delete trigger

SQL Code:

CREATE OR REPLACE TRIGGER trg__fridge_model_del

BEFORE DELETE ON EA_FRIDGE

FOR EACH ROW

DECLARE

lv_OrderCount NUMBER;

BEGIN

SELECT COUNT(*) INTO lv_OrderCount

FROM EA_ORDER

WHERE ApplianceID = :OLD.ApplianceID;

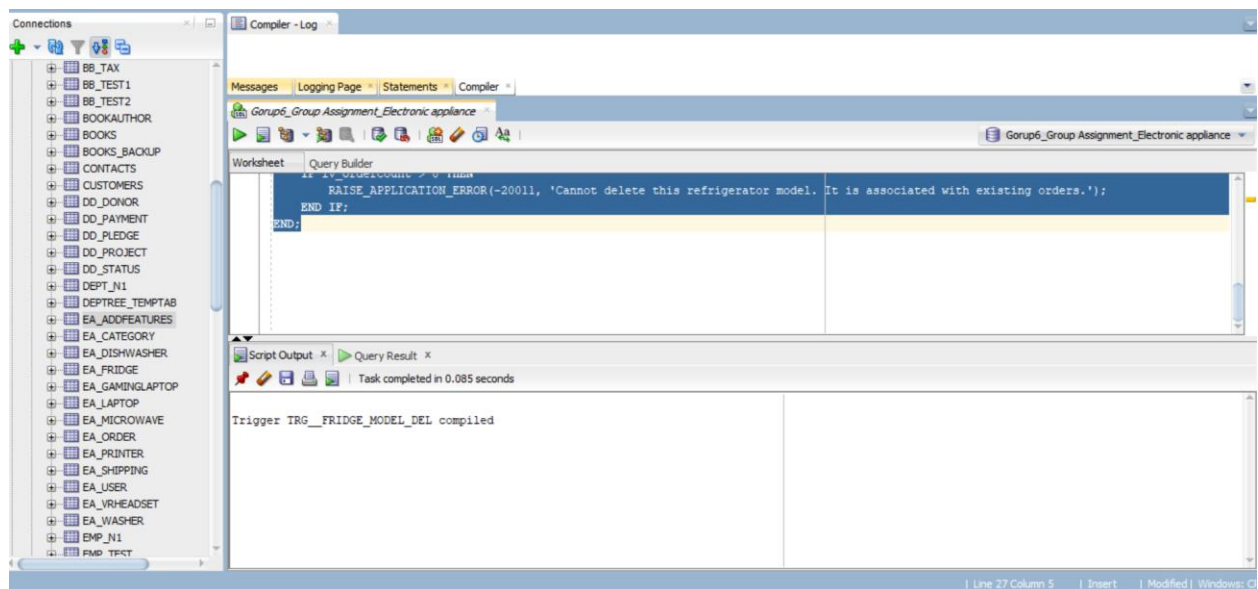
IF lv_OrderCount > 0 THEN

RAISE_APPLICATION_ERROR(-20011, 'Cannot delete this refrigerator model. It is associated with existing orders.');

END IF;

END;

//This trigger checks if the model of fridge to be deleted is associated with any orders.



After Delete trigger

SQL Code:

CREATE OR REPLACE TRIGGER trg_gaminglaptop_del

AFTER DELETE ON EA_GAMINGLAPTOP

FOR EACH ROW

BEGIN

```

UPDATE EA_ORDER

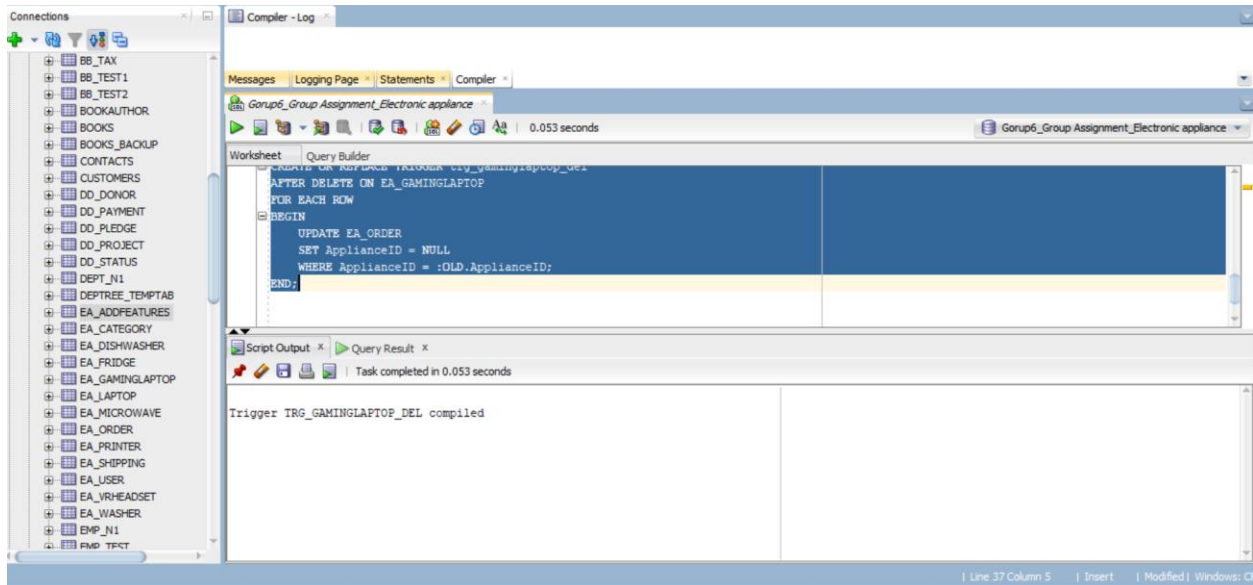
SET ApplianceID = NULL

WHERE ApplianceID = :OLD.ApplianceID;

END;

// After a gaming laptop model is deleted from the EA_GAMINGLAPTOP table, this trigger
updates the ApplianceID of associated orders to NULL

```



Sequences

```
CREATE SEQUENCE grpsequence
```

```
MAXVALUE 1100
```

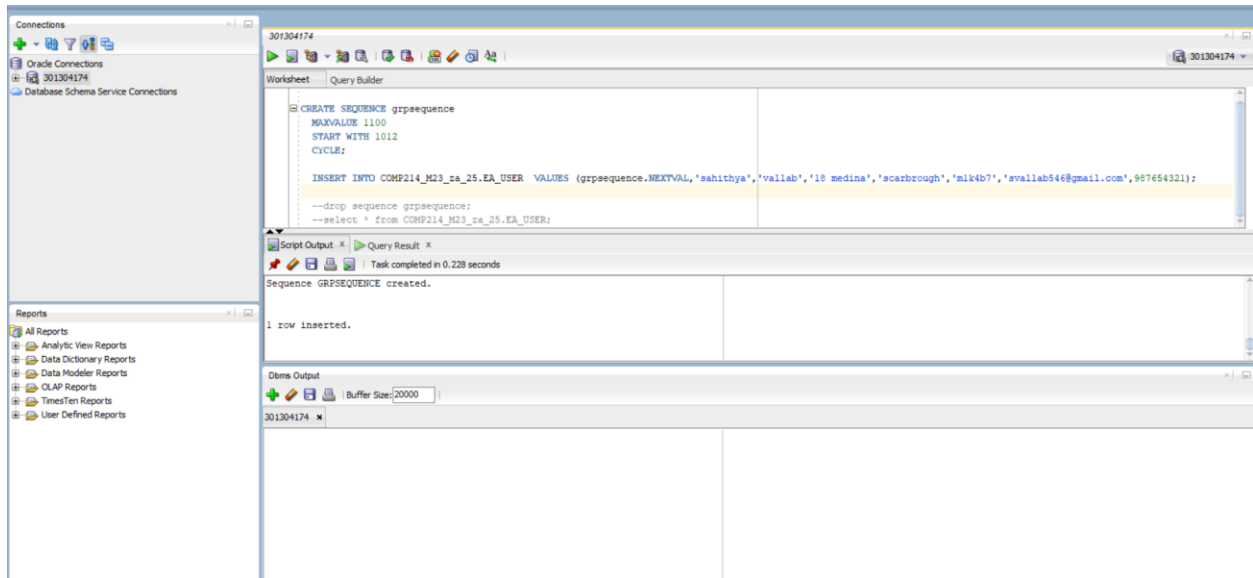
```
START WITH 1012
```

```
CYCLE;
```

```
INSERT INTO COMP214_M23_zs_25.EA_USER VALUES (grpsequence.NEXTVAL,'sahithya','vallab','18
medina','scarbrough','m1k4b7','svallab546@gmail.com','987654321);
```

```
--drop sequence grpsequence;
```

```
--select * from COMP214_M23_zs_25.EA_USER;
```



INDEXES

```
CREATE INDEX idx_product_price ON COMP214_M23_zs_25.EA_DISHWASHER(ProductName, Price);
```

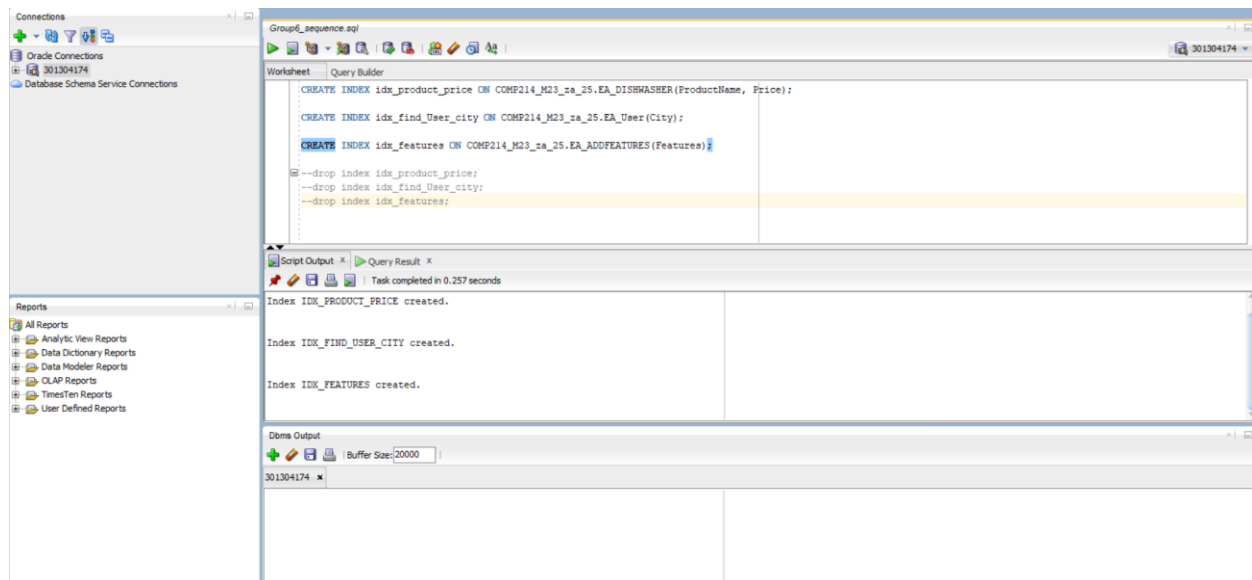
```
CREATE INDEX idx_find_User_city ON COMP214_M23_zs_25.EA_User(City);
```

```
CREATE INDEX idx_features ON COMP214_M23_zs_25.EA_ADDFEATURES(Features);
```

```
--drop index idx_product_price;
```

```
--drop index idx_find_User_city;
```

```
--drop index idx_features;
```



PROCEDURES

CREATE OR REPLACE PACKAGE printer_shipping_pkg AS

-- Procedure to insert printer details

PROCEDURE insert_printer_details(

 p_ModelID COMP214_M23_zs_25.EA_PRINTER.ModelID%TYPE,
 p_CategoryID COMP214_M23_zs_25.EA_PRINTER.CategoryID%TYPE,
 p_ApplianceID COMP214_M23_zs_25.EA_PRINTER.ApplianceID%TYPE,
 p_ProductName COMP214_M23_zs_25.EA_PRINTER.ProductName%TYPE,
 p_Manufacturer COMP214_M23_zs_25.EA_PRINTER.Manufacturer%TYPE,
 p_Price COMP214_M23_zs_25.EA_PRINTER.Price%TYPE,
 p_FeatureID COMP214_M23_zs_25.EA_PRINTER.FeatureID%TYPE

);

-- Procedure to get shipping details by ID

PROCEDURE get_shipping_details(p_shipping_id IN NUMBER);

END printer_shipping_pkg;

/

```
CREATE OR REPLACE PACKAGE BODY printer_shipping_pkg AS
```

```
-- Procedure to insert printer details
```

```
PROCEDURE insert_printer_details(
```

```
    p_ModelID COMP214_M23_zs_25.EA_PRINTER.ModelID%TYPE,
```

```
    p_CategoryID COMP214_M23_zs_25.EA_PRINTER.CategoryID%TYPE,
```

```
    p_ApplianceID COMP214_M23_zs_25.EA_PRINTER.ApplianceID%TYPE,
```

```
    p_ProductName COMP214_M23_zs_25.EA_PRINTER.ProductName%TYPE,
```

```
    p_Manufacturer COMP214_M23_zs_25.EA_PRINTER.Manufacturer%TYPE,
```

```
    p_Price COMP214_M23_zs_25.EA_PRINTER.Price%TYPE,
```

```
    p_FeatureID COMP214_M23_zs_25.EA_PRINTER.FeatureID%TYPE
```

```
) AS
```

```
BEGIN
```

```
    INSERT INTO COMP214_M23_zs_25.EA_PRINTER (ModelID, CategoryID, ApplianceID, ProductName,  
    Manufacturer, Price, FeatureID)
```

```
    VALUES (p_ModelID, p_CategoryID, p_ApplianceID, p_ProductName, p_Manufacturer, p_Price, p_FeatureID);
```

```
COMMIT;
```

```
    DBMS_OUTPUT.PUT_LINE('Printer details inserted successfully.');
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
```

```
        ROLLBACK;
```

```
END insert_printer_details;
```

```
-- Procedure to get shipping details by ID
```

```
PROCEDURE get_shipping_details(p_shipping_id IN NUMBER) AS
```

```
    v_shipping_details COMP214_M23_zs_25.EA_SHIPPING%ROWTYPE;
```

```
-- Declare a cursor to retrieve shipping details
```

```
CURSOR shipping_cursor IS
```



```

SELECT *

FROM COMP214_M23_zä_25.EA_SHIPPING

WHERE ShippingID = p_shipping_id;

BEGIN

OPEN shipping_cursor;

FETCH shipping_cursor INTO v_shipping_details;


IF shipping_cursor%NOTFOUND THEN

    DBMS_OUTPUT.PUT_LINE('Shipping details not found for the given ShippingID.');
```

ELSE

```

    DBMS_OUTPUT.PUT_LINE('Shipping ID: ' || v_shipping_details.ShippingID);
    DBMS_OUTPUT.PUT_LINE('Shipping Status: ' || v_shipping_details.ShippingStatus);
    DBMS_OUTPUT.PUT_LINE('Order ID: ' || v_shipping_details.OrderID);
    DBMS_OUTPUT.PUT_LINE('Shipping Date: ' || v_shipping_details.ShippingDate);
    DBMS_OUTPUT.PUT_LINE('Arrival Date: ' || v_shipping_details.ArrivalDate);
    DBMS_OUTPUT.PUT_LINE('Shipping Address: ' || v_shipping_details.ShippingAddress);
    DBMS_OUTPUT.PUT_LINE('Shipping City: ' || v_shipping_details.ShippingCity);
END IF;


CLOSE shipping_cursor;

EXCEPTION

WHEN OTHERS THEN

    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

END get_shipping_details;

END printer_shipping_pkg;

/

//anonymus block

BEGIN

-- Call the insert_printer_details procedure with appropriate parameter values
printer_shipping_pkg.insert_printer_details(
```

```

p_ModelID => 200013,

p_CategoryID => 3,

p_ApplianceID => 211,

p_ProductName => 'Dot Matrix Printer',

p_Manufacturer => 'Epson',

p_Price => 120,

p_FeatureID => 1

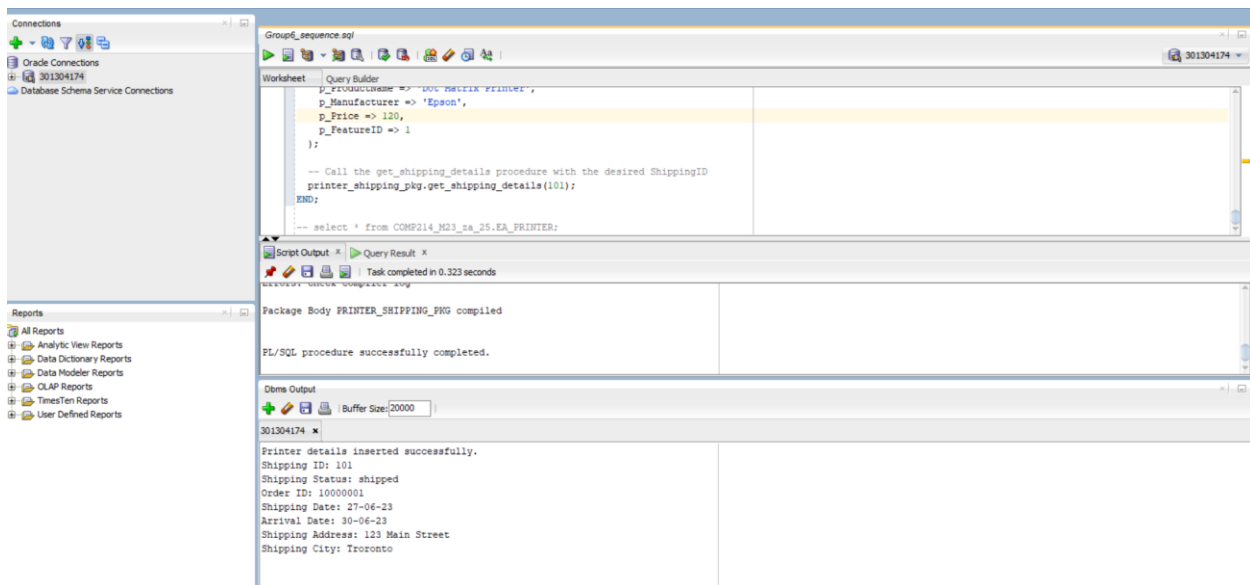
);

-- Call the get_shipping_details procedure with the desired ShippingID
printer_shipping_pkg.get_shipping_details(101);

END;

-- select * from COMP214_M23_zs_25.EA_PRINTER;

```



FUNCTIONS

CREATE OR REPLACE PACKAGE EA_PACKAGE AS

```
-- Function using cursor and exception handling to get order details
```

```
FUNCTION get_order_details(order_id IN COMP214_M23_z_25.EA_ORDER.OrderID%TYPE) RETURN  
COMP214_M23_z_25.EA_ORDER%ROWTYPE;
```

```
FUNCTION get_microwave_details(appliance_id IN COMP214_M23_z_25.EA_MICROWAVE.ApplianceID%TYPE)  
RETURN COMP214_M23_z_25.EA_MICROWAVE%ROWTYPE;
```

```
-- Function to calculate total price for a specific order
```

```
FUNCTION calculate_order_total(order_id IN COMP214_M23_z_25.EA_ORDER.OrderID%TYPE) RETURN  
NUMBER;
```

```
END EA_PACKAGE;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY EA_PACKAGE AS
```

```
FUNCTION get_order_details(order_id IN COMP214_M23_z_25.EA_ORDER.OrderID%TYPE) RETURN  
COMP214_M23_z_25.EA_ORDER%ROWTYPE IS
```

```
order_rec COMP214_M23_z_25.EA_ORDER%ROWTYPE;
```

```
order_cursor SYS_REFCURSOR;
```

```
BEGIN
```

```
OPEN order_cursor FOR
```

```
SELECT *
```

```
FROM COMP214_M23_z_25.EA_ORDER
```

```
WHERE OrderID = order_id;
```

```
FETCH order_cursor INTO order_rec;
```

```
CLOSE order_cursor;
```

```
IF order_cursor%NOTFOUND THEN
```

```
RAISE_APPLICATION_ERROR(-20001, 'Order not found.');
```

```
END IF;
```

```
RETURN order_rec;
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```

        RAISE_APPLICATION_ERROR(-20002, 'Order not found.');
```

WHEN OTHERS THEN

```

    IF order_cursor IS NOT NULL THEN

        CLOSE order_cursor;

    END IF;

END get_order_details;
```

```

FUNCTION get_microwave_details(appliance_id IN COMP214_M23_zs_25.EA_MICROWAVE.ApplianceID%TYPE)
RETURN COMP214_M23_zs_25.EA_MICROWAVE%ROWTYPE IS

    microwave_rec COMP214_M23_zs_25.EA_MICROWAVE%ROWTYPE;

BEGIN

    SELECT *

    INTO microwave_rec

    FROM COMP214_M23_zs_25.EA_MICROWAVE

    WHERE ApplianceID = appliance_id;

    RETURN microwave_rec;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20003, 'Microwave details not found.');
```

WHEN OTHERS THEN

```

    RAISE;

END get_microwave_details;
```

-- Function to calculate total price for a specific order

```

FUNCTION calculate_order_total(order_id IN COMP214_M23_zs_25.EA_ORDER.OrderID%TYPE) RETURN
NUMBER IS

    total_amount COMP214_M23_zs_25.EA_ORDER.Total%TYPE;

BEGIN

    SELECT Total

    INTO total_amount
```

```

FROM COMP214_M23_zä_25.EA_ORDER

WHERE OrderID = order_id;

IF total_amount > 100 THEN
    RETURN total_amount - 20;
ELSE
    RETURN total_amount;
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20004, 'Order not found.');
```

WHEN OTHERS THEN

```

    RAISE;
END calculate_order_total;
END EA_PACKAGE;
/
--annonymus block to call functions

DECLARE

order_id_to_fetch COMP214_M23_zä_25.EA_ORDER.OrderID%TYPE := 10000001;
appliance_id_to_fetch COMP214_M23_zä_25.EA_MICROWAVE.ApplianceID%TYPE := 101;
order_total NUMBER;
order_rec COMP214_M23_zä_25.EA_ORDER%ROWTYPE;
microwave_rec COMP214_M23_zä_25.EA_MICROWAVE%ROWTYPE;
BEGIN

    -- Calling get_order_details function
    BEGIN
        order_rec := EA_PACKAGE.get_order_details(order_id_to_fetch);
        IF order_rec.OrderID IS NOT NULL THEN
            DBMS_OUTPUT.PUT_LINE('Order Details:');
            DBMS_OUTPUT.PUT_LINE('OrderID: ' || order_rec.OrderID);
```

```

    DBMS_OUTPUT.PUT_LINE('UserID: ' || order_rec.UserID);

    DBMS_OUTPUT.PUT_LINE('Total: ' || order_rec.Total);

ELSE

    DBMS_OUTPUT.PUT_LINE('Order not found.');
```

END IF;

```

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('Order not found.');
```

WHEN OTHERS THEN

```

    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);

END;
```

-- Calling get_microwave_details function

```

BEGIN

    microwave_rec := EA_PACKAGE.get_microwave_details(appliance_id_to_fetch);

    IF microwave_rec.ApplianceID IS NOT NULL THEN

        DBMS_OUTPUT.PUT_LINE('Microwave Details:');

        DBMS_OUTPUT.PUT_LINE('ApplianceID: ' || microwave_rec.ApplianceID);

        DBMS_OUTPUT.PUT_LINE('ProductName: ' || microwave_rec.ProductName);

        DBMS_OUTPUT.PUT_LINE('Manufacturer: ' || microwave_rec.Manufacturer);

    ELSE

        DBMS_OUTPUT.PUT_LINE('Microwave details not found.');
```

END IF;

```

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('Microwave details not found.');
```

WHEN OTHERS THEN

```

    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);

END;
```

-- Calling calculate_order_total function

BEGIN

```
order_total := EA_PACKAGE.calculate_order_total(order_id_to_fetch);
```

```
DBMS_OUTPUT.PUT_LINE('Calculated Order Total: ' || order_total);
```

EXCEPTION

WHEN OTHERS THEN

```
DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
```

END;

END;

/

The screenshot displays the Oracle SQL Developer environment. On the left, the 'Connections' pane shows a connection to '301304174'. Below it, the 'Reports' pane lists various report types. The main workspace is titled 'Group0_sequence.sql' and contains a PL/SQL procedure. The procedure starts with a 'WHEN NO_DATA_FOUND THEN' block, followed by a 'WHEN OTHERS THEN' block. The main body of the procedure begins with a comment '-- Calling calculate_order_total function', followed by a 'BEGIN' block. Inside the 'BEGIN' block, the line 'order_total := EA_PACKAGE.calculate_order_total(order_id_to_fetch);' is highlighted in yellow. This is followed by 'DBMS_OUTPUT.PUT_LINE('Calculated Order Total: ' || order_total);'. An 'EXCEPTION' block follows, with a 'WHEN OTHERS THEN' block containing 'DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);'. The procedure ends with 'END;' and a forward slash '/' to execute it. Below the code editor, the 'Script Output' pane shows 'Task completed in 1.12 seconds' and 'PL/SQL procedure successfully completed.'. At the bottom, the 'DBMS Output' pane shows the output for connection '301304174', displaying 'Microwave Details: ApplianceID: 101, ProductName: Compact Microwave, Manufacturer: Panasonic'.

```
WHEN NO_DATA_FOUND THEN
  DBMS_OUTPUT.PUT_LINE('Microwave details not found.');
```

```
WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
```

```
-- Calling calculate_order_total function
BEGIN
  order_total := EA_PACKAGE.calculate_order_total(order_id_to_fetch);
  DBMS_OUTPUT.PUT_LINE('Calculated Order Total: ' || order_total);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
```

```
/
```

Task completed in 1.12 seconds

PL/SQL procedure successfully completed.

DBMS Output

301304174 x

Microwave Details:
ApplianceID: 101
ProductName: Compact Microwave
Manufacturer: Panasonic