

E-Commerce Store Database

Term Project – COMP214-Advanced Database Concepts (SEC. 002)

Members

Amirmehrbod Panjnoush 301295854

Jeremy Fernandez 301076715

Maryam Hesam Mohseni 301325443

Dorian Holder-Poulton 301035297

Problem Domain Description

A database for a general e-commerce store to handle various orders from daily customers (Similar to Amazon).

Tables and Attributes

Orders:

1. Order# PK (NUM)
2. Customer# FK (NUM)
3. OrderStatus# (NUM)
4. OrderDate (DATE)
5. ShipDate DATE,
6. ShipAddress (VARCHAR2(50))
7. ShipCity (VARCHAR2(20))
8. ShipProvince (VARCHAR2(20))
9. ShipPostal (VARCHAR2(20))

OrderItems (Table to reduce M - M between Orders and Products):

1. OrderItem# PK (NUM)
2. Order# FK (NUM)
3. Product# FK (NUM)
4. Quantity (NUM)
5. PaidEach (NUM)

Products:

1. Product# PK (NUM)
2. Name (VARCHAR2(20))
3. Description (VARCHAR2(100))
4. Category (CHAR(5))
5. UnitPrice (NUM)
6. RetailPrice (NUM)
7. Stock (NUM)
8. Ordered (NUM)
9. Discount (NUM)

Customers:

1. Customer# PK (NUM)
2. FirstName (VARCHAR(20))
3. LastName (VARCHAR2(20))
4. Address (VARCHAR2(50))
5. City (VARCHAR2(20))
6. Province (VARCHAR2(30))
7. PostalCode (VARCHAR2(30))
8. Email (VARCHAR2(40))
9. Phone# VARCHAR2(15),
10. Username VARCHAR2(15),
11. HashedPassword VARCHAR2(50),
12. Cookie NUMBER(4)

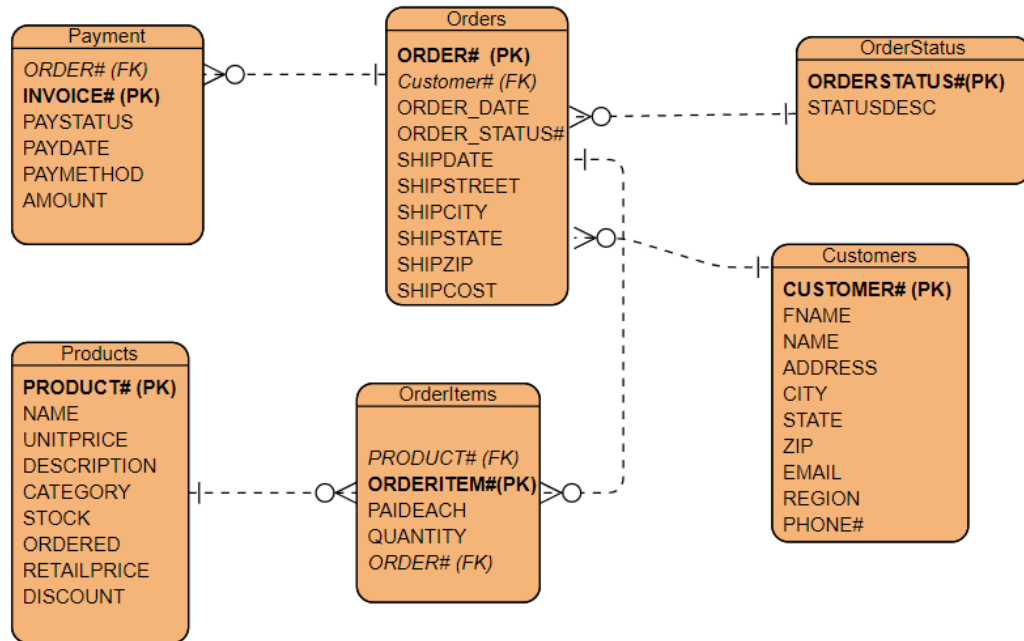
Payment:

1. Invoice# PK (NUM)
2. Order# FK (NUM)
3. Amount (NUM)
4. PayStatus (VARCHAR2(50))
5. PayDate (DATE)
6. PayMethod (VARCHAR2(50))

OrderStatus (Reference table):

1. OrderStatus# PK (NUM)
2. StatusDesc (VARCHAR2(50))

ERD



Database Testing

Sequence Tests

CREATE SEQUENCE order_seq START WITH 1001 INCREMENT BY 1;

ORDER#	CUSTOMER#	ORDERSTATUS#	ORDERDATE	SHIPDATE	SHIPADDRESS	SHIPCITY	SHIPPROVINCE	SHIPPOSTAL
1	1001	1	1 (null)	(null)	(null)	(null)	(null)	(null)
2	1002	3	1 (null)	(null)	(null)	(null)	(null)	(null)
3	1003	2	6 (null)	(null)	(null)	(null)	(null)	(null)
4	1004	2	2 (null)	(null)	(null)	(null)	(null)	(null)
5	1005	5	1 (null)	(null)	(null)	(null)	(null)	(null)
6	1006	4	1 (null)	(null)	(null)	(null)	(null)	(null)
7	1007	8	1 (null)	(null)	(null)	(null)	(null)	(null)
8	1008	6	1 (null)	(null)	(null)	(null)	(null)	(null)
9	1009	10	3 01-DEC-23	(null)	(null)	(null)	(null)	(null)
10	1010	7	5 01-DEC-23	04-DEC-23	890 Maple Ave	Quebec City	QC	G1R 4V7

Index Tests

INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1 SYSTEM	SYS_C007004	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	ORDER#
2 SYSTEM	IDX_ORDERS_CUSTOMER_DATE	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	CUSTOMER#, ORDERDATE

INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1 SYSTEM	SYS_C007010	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	INVOICE#
2 SYSTEM	IDX_PAYMENT_ORDER	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	ORDER#
3 SYSTEM	IDX_PAYMENT_PAYDATE	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	PAYDATE

INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1 SYSTEM	SYS_C007002	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	PRODUCT#
2 SYSTEM	IDX_PRODUCTS_NAME	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	NAME
3 SYSTEM	IDX_PRODUCTS_CATEGORY	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	CATEGORY

Trigger Tests

--Script:

-- Test for update_order_on_confirm_trg

-- We test by generating an invoice for Order#100001 and compare Order and Payment tables before and after

DECLARE

lv_orderid_num Orders.Order#%TYPE := 1001;

BEGIN

INSERT INTO Payment (Invoice#, Order#, Amount, Subtotal, ShipCost)

VALUES (invoice_seq.NEXTVAL, lv_orderid_num, 2853.1, 2848.1, 5);

END;

/

--Script:

-- Test for update_order_on_payment_trg

-- We test by updating the payment record that we created in the previous test

DECLARE

lv_invoiceid_num Payment.Invoice#%TYPE := 100021;

BEGIN

Update Payment

SET PayStatus = 'Paid', PayDate = SYSDATE, PayMethod = 'Debit'

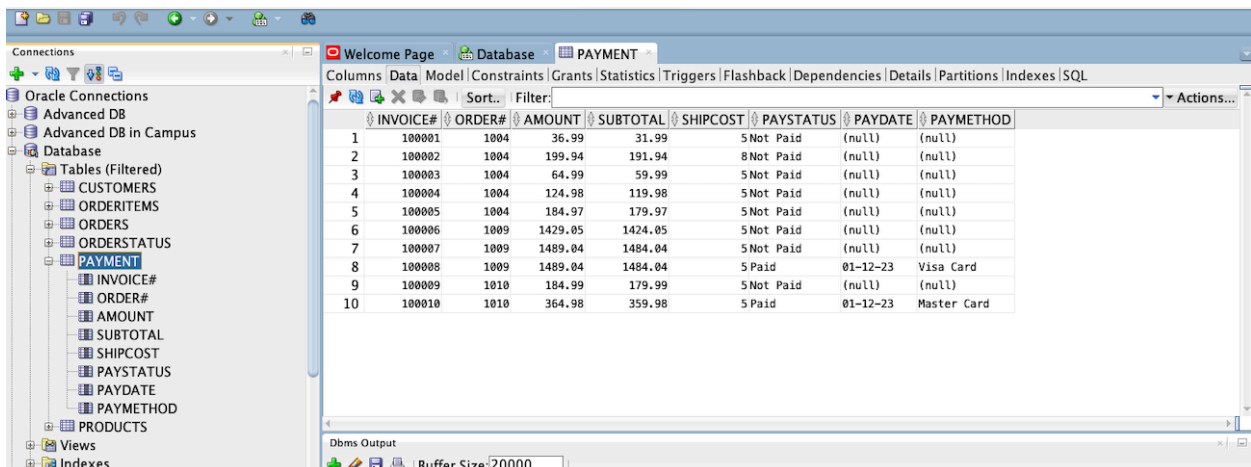
Where Invoice# = lv_invoiceid_num;

COMMIT;

END;

/

Before Insertion:



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Database' tree is expanded, showing the 'PAYMENT' table. The main window displays the 'PAYMENT' table data. The table has columns: INVOICE#, ORDER#, AMOUNT, SUBTOTAL, SHIPCOST, PAYSTATUS, PAYDATE, and PAYMETHOD. The data is as follows:

INVOICE#	ORDER#	AMOUNT	SUBTOTAL	SHIPCOST	PAYSTATUS	PAYDATE	PAYMETHOD
100001	1004	36.99	31.99		5 Not Paid	(null)	(null)
100002	1004	199.94	191.94		8 Not Paid	(null)	(null)
100003	1004	64.99	59.99		5 Not Paid	(null)	(null)
100004	1004	124.98	119.98		5 Not Paid	(null)	(null)
100005	1004	184.97	179.97		5 Not Paid	(null)	(null)
100006	1009	1429.05	1424.05		5 Not Paid	(null)	(null)
100007	1009	1489.04	1484.04		5 Not Paid	(null)	(null)
100008	1009	1489.04	1484.04		5 Paid	01-12-23	Visa Card
100009	1010	184.99	179.99		5 Not Paid	(null)	(null)
100010	1010	364.98	359.98		5 Paid	01-12-23	Master Card

Oracle SQL Developer interface showing the 'ORDERS' table. The left pane displays the database structure with 'ORDERS' selected. The main pane shows the table data with columns: ORDER#, CUSTOMER#, ORDERSTATUS#, ORDERDATE, SHIPDATE, SHIPADDRESS, SHIPCITY, SHIPPROVINCE, and SHIPPOSTAL.

ORDER#	CUSTOMER#	ORDERSTATUS#	ORDERDATE	SHIPDATE	SHIPADDRESS	SHIPCITY	SHIPPROVINCE	SHIPPOSTAL
1	1001	1	1 (null)	(null)	(null)	(null)	(null)	(null)
2	1002	3	1 (null)	(null)	(null)	(null)	(null)	(null)
3	1003	2	6 (null)	(null)	(null)	(null)	(null)	(null)
4	1004	2	2 (null)	(null)	(null)	(null)	(null)	(null)
5	1005	5	1 (null)	(null)	(null)	(null)	(null)	(null)
6	1006	4	1 (null)	(null)	(null)	(null)	(null)	(null)
7	1007	8	1 (null)	(null)	(null)	(null)	(null)	(null)
8	1008	6	1 (null)	(null)	(null)	(null)	(null)	(null)
9	1009	10	2 01-12-23	(null)	(null)	(null)	(null)	(null)
10	1010	7	2 01-12-23	04-12-23	890 Maple Ave	Quebec City QC		G1R 4V7

After Insertion:

Oracle SQL Developer interface showing the 'PAYMENT' table. The left pane displays the database structure with 'PAYMENT' selected. The main pane shows the table data with columns: INVOICE#, ORDER#, AMOUNT, SUBTOTAL, SHIP COST, PAYSTATUS, PAYDATE, and PAYMETHOD.

INVOICE#	ORDER#	AMOUNT	SUBTOTAL	SHIP COST	PAYSTATUS	PAYDATE	PAYMETHOD
1	100001	1004	36.99	31.99	5 Not Paid	(null)	(null)
2	100002	1004	199.94	191.94	8 Not Paid	(null)	(null)
3	100003	1004	64.99	59.99	5 Not Paid	(null)	(null)
4	100004	1004	124.98	119.98	5 Not Paid	(null)	(null)
5	100005	1004	184.97	179.97	5 Not Paid	(null)	(null)
6	100006	1009	1429.05	1424.05	5 Not Paid	(null)	(null)
7	100007	1009	1489.04	1484.04	5 Not Paid	(null)	(null)
8	100008	1009	1489.04	1484.04	5 Paid	01-12-23	Visa Card
9	100009	1010	184.99	179.99	5 Not Paid	(null)	(null)
10	100010	1010	364.98	359.98	5 Paid	01-12-23	Master Card
11	100021	1001	2853.1	2848.1	5 Not Paid	(null)	(null)

Oracle SQL Developer interface showing the 'ORDERS' table. The left pane displays the database structure with 'ORDERS' selected. The main pane shows the table data. A new row has been added at the bottom of the list.

ORDER#	CUSTOMER#	ORDERSTATUS#	ORDERDATE	SHIPDATE	SHIPADDRESS	SHIPCITY	SHIPPROVINCE	SHIPPOSTAL
1	1001	1	2 (null)	(null)	(null)	(null)	(null)	(null)
2	1002	3	1 (null)	(null)	(null)	(null)	(null)	(null)
3	1003	2	6 (null)	(null)	(null)	(null)	(null)	(null)
4	1004	2	2 (null)	(null)	(null)	(null)	(null)	(null)
5	1005	5	1 (null)	(null)	(null)	(null)	(null)	(null)
6	1006	4	1 (null)	(null)	(null)	(null)	(null)	(null)
7	1007	8	1 (null)	(null)	(null)	(null)	(null)	(null)
8	1008	6	1 (null)	(null)	(null)	(null)	(null)	(null)
9	1009	10	2 01-12-23	(null)	(null)	(null)	(null)	(null)
10	1010	7	2 01-12-23	04-12-23	890 Maple Ave	Quebec City QC		G1R 4V7

After Update:

Oracle SQL Developer : Table COMP228_F23_P1Y_48.PAYMENT@Database

Connections

Oracle Connections

- Advanced DB
- Advanced DB in Campus
- Database
 - Tables (Filtered)
 - CUSTOMERS
 - ORDERITEMS
 - ORDERS
 - ORDER#
 - CUSTOMER#
 - ORDERSTATUS#
 - ORDERDATE
 - SHIPDATE
 - SHIPADDRESS
 - SHIPCITY
 - SHIPPROVINCE
 - SHIPPOSTAL
 - ORDERSTATUS
 - PAYMENT

Database

COMP228_F23_P1Y_48.PAYMENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort: | Filter: | Actions...

	INVOICE#	ORDER#	AMOUNT	SUBTOTAL	SHIPCOST	PAYSTATUS	PAYDATE	PAYMETHOD
1	100001	1004	36.99	31.99		5 Not Paid	(null)	(null)
2	100002	1004	199.94	191.94		8 Not Paid	(null)	(null)
3	100003	1004	64.99	59.99		5 Not Paid	(null)	(null)
4	100004	1004	124.98	119.98		5 Not Paid	(null)	(null)
5	100005	1004	184.97	179.97		5 Not Paid	(null)	(null)
6	100006	1009	1429.05	1424.05		5 Not Paid	(null)	(null)
7	100007	1009	1489.04	1484.04		5 Not Paid	(null)	(null)
8	100008	1009	1489.04	1484.04		5 Paid	01-12-23	Visa Card
9	100009	1010	184.99	179.99		5 Not Paid	(null)	(null)
10	100010	1010	364.98	359.98		5 Paid	01-12-23	Master Card
11	100021	1001	2853.1	2848.1		5 Paid	01-12-23	Debit

Connections

Oracle Connections

- Advanced DB
- Advanced DB in Campus
- Database
 - Tables (Filtered)
 - CUSTOMERS
 - ORDERITEMS
 - ORDERS
 - ORDERSTATUS
 - PAYMENT
 - INVOICE#
 - ORDER#
 - AMOUNT
 - SUBTOTAL
 - SHIPCOST
 - PAYSTATUS
 - PAYDATE
 - PAYMETHOD
 - PRODUCTS

Database

COMP228_F23_P1Y_48.ORDERS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort: | Filter: | Actions...

	ORDER#	CUSTOMER#	ORDERSTATUS#	ORDERDATE	SHIPDATE	SHIPADDRESS	SHIPCITY	SHIPPROVINCE	SHIPPOSTAL
1	1001	1	3	01-12-23	(null)	(null)	(null)	(null)	(null)
2	1002	3	1	(null)	(null)	(null)	(null)	(null)	(null)
3	1003	2	6	(null)	(null)	(null)	(null)	(null)	(null)
4	1004	2	2	(null)	(null)	(null)	(null)	(null)	(null)
5	1005	5	1	(null)	(null)	(null)	(null)	(null)	(null)
6	1006	4	1	(null)	(null)	(null)	(null)	(null)	(null)
7	1007	8	1	(null)	(null)	(null)	(null)	(null)	(null)
8	1008	6	1	(null)	(null)	(null)	(null)	(null)	(null)
9	1009	10	2	01-12-23	(null)	(null)	(null)	(null)	(null)
10	1010	7	2	01-12-23	04-12-23	890 Maple Ave	Quebec City QC		G1R 4V7

Messages - Log | Dbms Output

Function Tests

```
-- Test for get_customer_info_sf
DECLARE
  customer_info VARCHAR2(500);
BEGIN
  customer_info := get_customer_info_sf(3); -- Replace 3 with an existing customer ID
  DBMS_OUTPUT.PUT_LINE('Customer Info: ' || customer_info);
END;
/
```

```
-- Test for get_product_info_sf
DECLARE
  product_info VARCHAR2(500);
BEGIN
  product_info := get_product_info_sf(102); -- Replace with an product ID
  DBMS_OUTPUT.PUT_LINE('Product Info: ' || product_info);
END;
/
```

```
-- Test for calculate_order_subtotal_sf
DECLARE
  v_order_subtotal NUMBER;
BEGIN
  v_order_subtotal := calculate_order_subtotal_sf(127);
  DBMS_OUTPUT.PUT_LINE('Order Subtotal: $' || TO_CHAR(v_order_total));
END;
/
```

```
-- Test for calculate_ship_cost_sf
DECLARE
  ship_cost NUMBER;
BEGIN
  ship_cost := calculate_ship_cost_sf(6); -- Replace with a quantity value
  DBMS_OUTPUT.PUT_LINE('Shipping Cost: $' || TO_CHAR(ship_cost));
END;
/
```

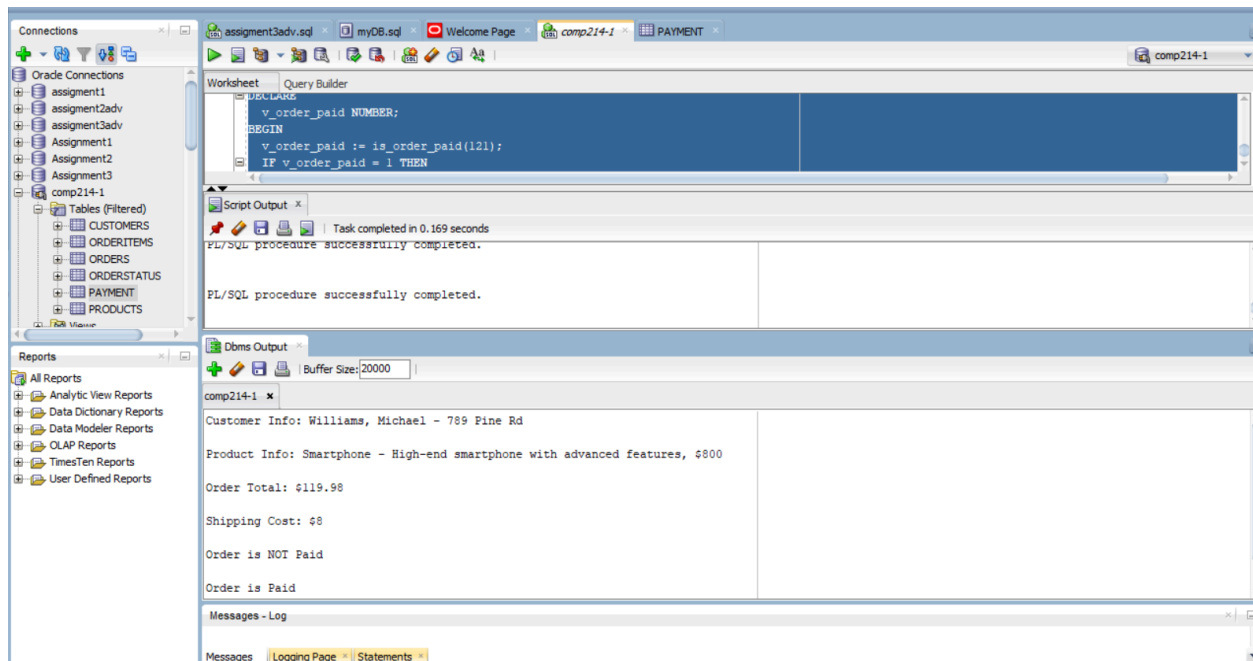
```
-- Test for is_order_paid
DECLARE
  v_order_paid NUMBER;
BEGIN
  v_order_paid := is_order_paid(121);
  IF v_order_paid = 1 THEN
```

```

    DBMS_OUTPUT.PUT_LINE('Order is Paid');
ELSE
    DBMS_OUTPUT.PUT_LINE('Order is NOT Paid');
END IF;
END;
/
DECLARE
    v_order_paid NUMBER;
BEGIN
    v_order_paid := is_order_paid(128);
    IF v_order_paid = 1 THEN
        DBMS_OUTPUT.PUT_LINE('Order is Paid');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Order is NOT Paid');
    END IF;
END;

```

Function Test screenshots:



Tables for reference:

Connections

Oracle Connections

- assignment1
- assignment2adv
- assignment3adv
- Assignment1
- Assignment2
- Assignment3
- comp214-1
 - Tables (Filtered)
 - CUSTOMERS
 - ORDERITEMS
 - ORDERS
 - ORDERSTATUS
 - PAYMENT
 - PRODUCTS

Reports

All Reports

- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

assignment3adv.sql | myDB.sql | Welcome Page | comp214-1 | PAYMENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort: | Filter:

	INVOICE#	ORDER#	AMOUNT	SUBTOTAL	SHIPCOST	PAYSTATUS	PAYDATE	PAYMETHOD
1	100011	121	36.99	31.99	5	Not Paid	(null)	(null)
2	100012	122	199.94	191.94	8	Not Paid	(null)	(null)
3	100013	123	64.99	59.99	5	Not Paid	(null)	(null)
4	100014	124	124.98	119.98	5	Not Paid	(null)	(null)
5	100015	125	184.97	179.97	5	Not Paid	(null)	(null)
6	100016	126	1429.05	1424.05	5	Not Paid	(null)	(null)
7	100017	127	1489.04	1484.04	5	Not Paid	(null)	(null)
8	100018	128	1489.04	1484.04	5	Paid	01-DEC-23	Visa Card
9	100019	129	184.99	179.99	5	Not Paid	(null)	(null)
10	100020	130	364.98	359.98	5	Paid	01-DEC-23	Master Card

Dbms Output

Buffer Size: 20000

comp214-1

Customer Info: Williams, Michael - 789 Pine Rd

Product Info: Smartphone - High-end smartphone with advanced features, \$800

Order Total: \$119.98

Shipping Cost: \$8

Order is NOT Paid

Order is Paid

Messages - Log

Messages | Logging Page | Statements

Click on an identifier with the Control key down to perform "Go to Declaration"

Search

Oracle Connections

- assignment1
- assignment2adv
- assignment3adv
- Assignment1
- Assignment2
- Assignment3
- comp214-1
 - Tables (Filtered)
 - CUSTOMERS
 - ORDERITEMS
 - ORDERS
 - ORDERSTATUS
 - PAYMENT
 - PRODUCTS

PRODUCT#	NAME	DESCRIPTION	CATEGORY	UNITPRICE	RETAILPRICE	STOCK	ORDERED
1	101 Laptop	Powerful laptop for work and entertainment	Electronics	1200	1499.99	50	0
2	102 Smartphone	High-end smartphone with advanced features	Electronics	800	999.99	100	25
3	103 Shirt	Blue cotton shirt for men - Size M	Apparel	29.99	39.99	20	3
4	104 Desk Chair	Comfortable ergonomic chair for your office	Furniture	150	199.99	20	8
5	105 Bluetooth Speaker	Portable speaker with wireless connectivity	Electronics	50	79.99	40	15
6	106 Cookware Set	High-quality non-stick cookware set for your kitchen	Kitchen	120	149.99	15	3
7	107 Digital Camera	Professional-grade digital camera for photography enthusiasts	Electronics	600	799.99	25	7
8	108 Gaming Mouse	High-performance gaming mouse with customizable features	Gaming	40	59.99	60	12
9	109 Fitness Tracker	Track your fitness activities with this advanced fitness tracker	Fitness	80	109.99	35	1
10	110 Blender	Powerful blender for smoothies and food preparation	Appliances	60	89.99	45	18

Connections

- Oracle Connections
 - assignment1
 - assignment2adv
 - assignment3adv
 - Assignment1
 - Assignment2
 - Assignment3
 - comp214-1
 - Tables (Filtered)
 - CUSTOMERS
 - ORDERITEMS**
 - ORDERS
 - ORDERSTATUS
 - PAYMENT
 - PRODUCTS

Reports

- All Reports
 - Analytic View Reports
 - Data Dictionary Reports
 - Data Modeler Reports
 - OLAP Reports
 - TimesTen Reports
 - User Defined Reports

assignment3adv.sql | myDB.sql | Welcome Page | comp214-1 | ORDERITEMS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes

Sort.. | Filter:

	ORDERITEM#	ORDER#	PRODUCT#	QUANTITY	PAIDEACH
1	10011	121	101	2	(null)
2	10012	122	103	2	(null)
3	10013	123	102	1	(null)
4	10014	124	104	3	(null)
5	10015	125	105	2	(null)
6	10016	126	103	1	31.99
7	10017	127	108	2	59.99
8	10018	128	101	1	1424.04
9	10019	129	108	1	59.99
10	10020	130	104	2	179.99

Dbms Output

Buffer Size: 20000

comp214-1

Customer Info: Williams, Michael - 789 Pine Rd

Product Info: Smartphone - High-end smartphone with advanced features, \$800

Order Total: \$119.98

Shipping Cost: \$8

Order is NOT Paid

Connections

- Oracle Connections
 - assignment1
 - assignment2adv
 - assignment3adv
 - Assignment1
 - Assignment2
 - Assignment3
 - comp214-1
 - Tables (Filtered)
 - CUSTOMERS**
 - ORDERITEMS
 - ORDERS
 - ORDERSTATUS
 - PAYMENT
 - PRODUCTS

assignment3adv.sql | myDB.sql | Welcome Page | comp214-1 | CUSTOMERS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	CUSTOMER#	FIRSTNAME	LASTNAME	ADDRESS	CITY	PROVINCE	POSTALCODE	EMAIL	PHONE#	USERNAME	H
1	1	John	Smith	123 Main St	Toronto	ON	M5V 2H1	john.smith@gmail.com	647-555-1234	johnsmith	(nu)
2	2	Emily	Johnson	456 Oak Ave	Vancouver	BC	V6B 3G7	emily.johnson@yahoo.com	778-555-5678	emilyjohnson	(nu)
3	3	Michael	Williams	789 Pine Rd	Montreal	QC	H2Y 1Z6	michael.williams@gmail.com	514-555-9876	michaelwilliams	(nu)
4	4	Sophia	Brown	101 Cedar Ln	Calgary	AB	T2P 0R2	sophia.brown@gmail.com	403-555-5432	sophiabrown	(nu)
5	5	Daniel	Miller	234 Elm St	Ottawa	ON	K1P 5G8	daniel.miller@yahoo.com	613-555-8765	danielmiller	(nu)
6	6	Olivia	Davis	567 Birch Blvd	Edmonton	AB	T5J 2V4	olivia.davis@yahoo.com	780-555-2345	oliviadavis	(nu)
7	7	Ethan	Wilson	890 Maple Ave	Quebec City	QC	G1R 4V7	ethan.wilson@gmail.com	418-555-8765	ethanwilson	(nu)
8	8	Ava	Taylor	111 Pineapple Cres	Winnipeg	MB	R3B 0N2	ava.taylor@gmail.com	204-555-3456	avataylor	(nu)
9	9	Logan	Anderson	222 Banana Dr	Halifax	NS	B3H 2J2	logan.anderson@gmail.com	902-555-6543	logananderson	(nu)
10	10	Emma	White	333 Orange Ave	Victoria	BC	V8W 1B5	emma.white@aol.com	250-555-7890	emmawhite	(nu)
11	11	Maxon	Tones	444 Grape St	Berlin	SW	S4P 3Y8	maxon.tones@gmail.com	306-555-4321	maxontones	(nu)

Procedure Tests

-- Test Procedure: update_stock_sp

```
DECLARE
    v_product_id Products.Product#%TYPE := 101;
    v_new_stock NUMBER := 30;
BEGIN
    update_stock_sp(v_new_stock, v_product_id);
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Stock updated successfully.');
```

EXCEPTION

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
END;
/
```

-- Test Procedure: check_stock_sp

```
DECLARE
    v_product_id Products.Product#%TYPE := 101;
    v_requested_quantity NUMBER := 15;
BEGIN
    check_stock_sp(v_requested_quantity, v_product_id);
    DBMS_OUTPUT.PUT_LINE('Stock check passed. Sufficient stock available.');
```

EXCEPTION

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

-- Test Procedure: calculate_paideach_sp

```
DECLARE
    v_order_id Orders.Order#%TYPE := 1010; -- Replace with the order ID for testing
BEGIN
    calculate_paideach_sp(v_order_id);
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('PaidEach calculated successfully for order ' || v_order_id);
```

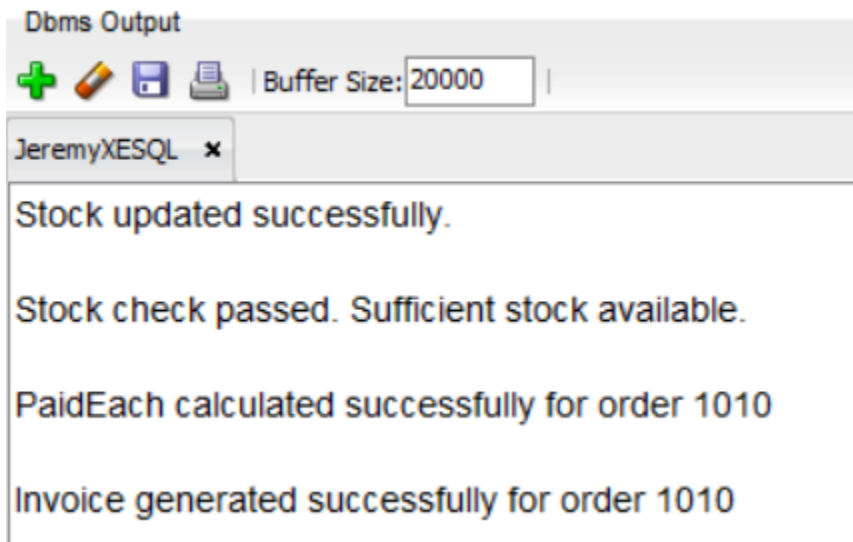
EXCEPTION

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
END;
/
```

```

-- Test Procedure: generate_invoice_sp
DECLARE
    v_order_id Orders.Order#%TYPE := 1010; -- Replace with the order ID for testing
BEGIN
    generate_invoice_sp(v_order_id);
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Invoice generated successfully for order ' || v_order_id);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
END;
/

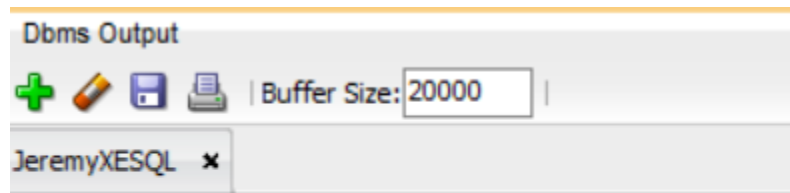
```



	PRODUCT#	NAME	DESCRIPTION	CATEGORY	UNITPRICE	RETAILPRICE	STOCK	ORDERED	DISCOUNT
1	101	Laptop	Powerful laptop for work and entertainment	Electronics	1200	1499.99	80	0	0.05
2	102	Smartphone	High-end smartphone with advanced features	Electronics	800	999.99	100	25	0
3	103	Shirt	Blue cotton shirt for men - Size M	Apparel	29.99	39.99	20	3	0.2
4	104	Desk Chair	Comfortable ergonomic chair for your office	Furniture	150	199.99	20	8	0.1
5	105	Bluetooth Speaker	Portable speaker with wireless connectivity	Electronics	50	79.99	40	15	0.06
6	106	Cookware Set	High-quality non-stick cookware set for your kitchen	Kitchen	120	149.99	15	3	0
7	107	Digital Camera	Professional-grade digital camera for photography enthusiasts	Electronics	600	799.99	25	7	0.05
8	108	Gaming Mouse	High-performance gaming mouse with customizable features	Gaming	40	59.99	60	12	0
9	109	Fitness Tracker	Track your fitness activities with this advanced fitness tracker	Fitness	80	109.99	35	1	0.1
10	110	Blender	Powerful blender for smoothies and food preparation	Appliances	60	89.99	45	18	0.05

Package Tests

```
-- Test Package Procedure: generate_invoice_pp
DECLARE
  v_order_id Orders.Order#%TYPE := 1010; -- Replace with the order ID for testing
BEGIN
  OrderManagement.generate_invoice_pp(v_order_id);
  COMMIT;
  DBMS_OUTPUT.PUT_LINE('Invoice generated successfully for order ' || v_order_id);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
    ROLLBACK;
END;
/
```



Invoice generated successfully for order 1010

I

	INVOICE#	ORDER#	AMOUNT	SUBTOTAL	SHIPCOST	PAYSTATUS	PAYDATE	PAYMETHOD
1	100001	1004	36.99	31.99	5	Not Paid	(null)	(null)
2	100002	1004	199.94	191.94	8	Not Paid	(null)	(null)
3	100003	1004	64.99	59.99	5	Not Paid	(null)	(null)
4	100004	1004	124.98	119.98	5	Not Paid	(null)	(null)
5	100005	1004	184.97	179.97	5	Not Paid	(null)	(null)
6	100006	1009	1429.05	1424.05	5	Not Paid	(null)	(null)
7	100007	1009	1489.04	1484.04	5	Not Paid	(null)	(null)
8	100008	1009	1489.04	1484.04	5	Paid	01-DEC-23	Visa Card
9	100009	1010	184.99	179.99	5	Not Paid	(null)	(null)
10	100010	1010	364.98	359.98	5	Paid	01-DEC-23	Master Card
11	100011	1010	364.98	359.98	5	Not Paid	(null)	(null)
12	100012	1010	364.98	359.98	5	Not Paid	(null)	(null)
13	100013	1010	364.98	359.98	5	Not Paid	(null)	(null)