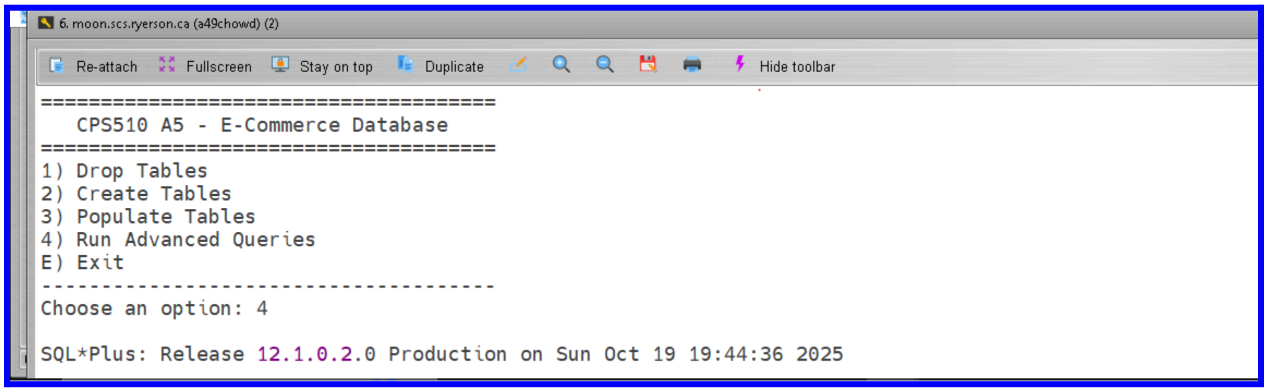
menu.sh

Code:

```
#!/bin/sh

MainMenu() {
    while true
    do
```

```
        clear
        echo "======================================"
        echo "   CPS510 A5 - E-Commerce Database   "
        echo "======================================"
        echo "1) Drop Tables"
        echo "2) Create Tables"
        echo "3) Populate Tables"
        echo "4) Run Advanced Queries"
        echo "E) Exit"
        echo "--------------------------------------"
        echo -n "Choose an option: "
        read CHOICE

        case $CHOICE in
            1) bash drop_tables.sh ;;
            2) bash create_tables.sh ;;
            3) bash populate_tables.sh ;;
            4) bash query_tables.sh ;;
            E|e) exit ;;
            *) echo "Invalid choice. Press Enter to continue."; read ;;
        esac
    done
}

MainMenu
```

Screenshot:



```
======================================
   CPS510 A5 - E-Commerce Database
======================================
1) Drop Tables
2) Create Tables
3) Populate Tables
4) Run Advanced Queries
E) Exit
--------------------------------------
Choose an option: 4

SQL*Plus: Release 12.1.0.2.0 Production on Sun Oct 19 19:44:36 2025
```

Comment:

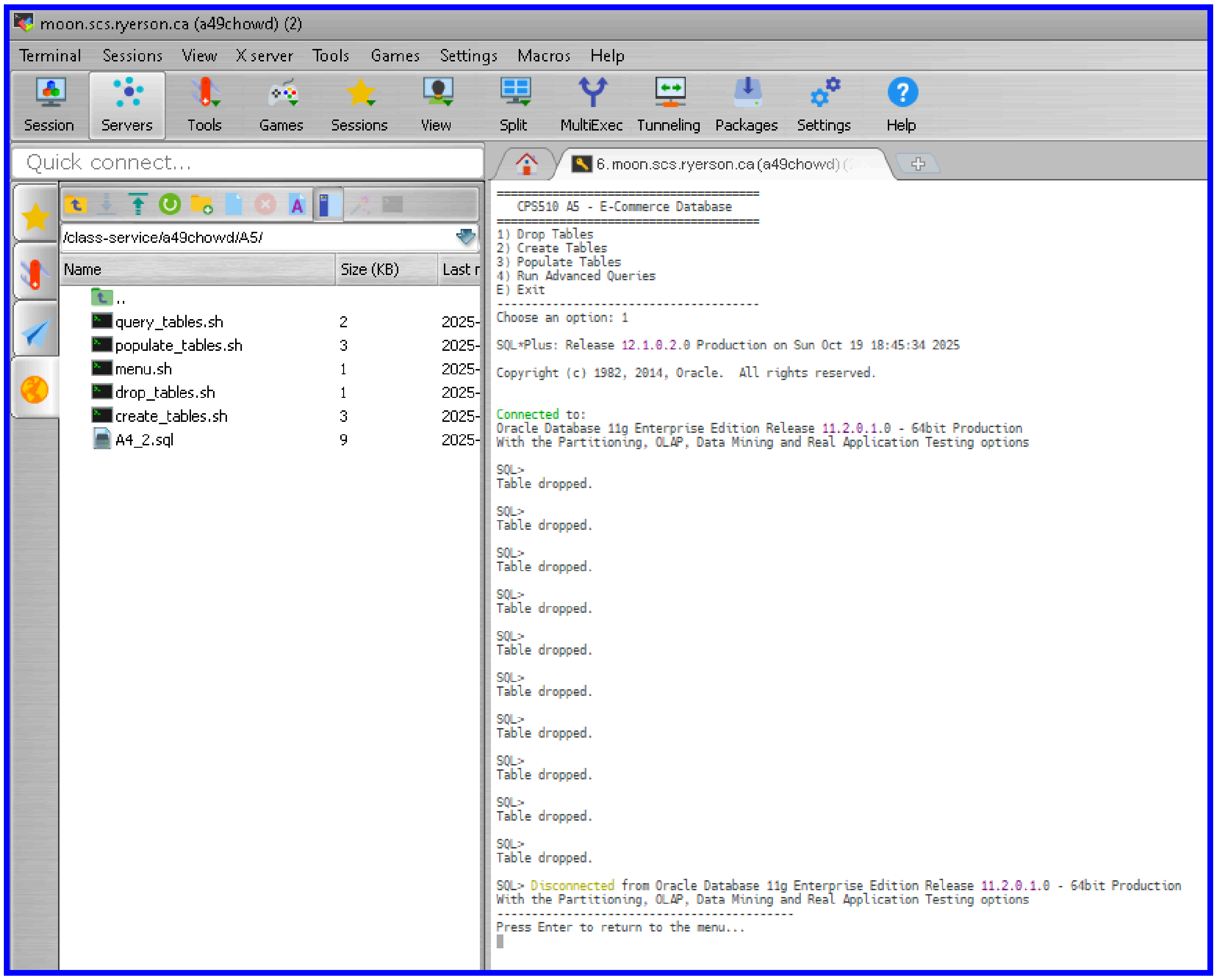
Displays the menu to run all scripts for the user in one place. Uses while loop to make it so menu keeps running until user wants to exist.

drop_tables.sh

Code:

```
#!/bin/sh
sqlplus64
"cs_username/cs_password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(
Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
DROP TABLE RETURNREQUEST CASCADE CONSTRAINTS;
DROP TABLE REVIEW CASCADE CONSTRAINTS;
DROP TABLE REPORT CASCADE CONSTRAINTS;
DROP TABLE PAYMENT CASCADE CONSTRAINTS;
DROP TABLE ORDER_PRODUCT CASCADE CONSTRAINTS;
DROP TABLE ORDERS CASCADE CONSTRAINTS;
DROP TABLE PRODUCT CASCADE CONSTRAINTS;
DROP TABLE STUDENT CASCADE CONSTRAINTS;
DROP TABLE STAFF CASCADE CONSTRAINTS;
DROP TABLE USERS CASCADE CONSTRAINTS;
exit;
EOF

echo "------------------------------------------"
echo "Press Enter to return to the menu..."
read
```

Screenshot:



```
========================================
   CPS510 A5 - E-Commerce Database
========================================
1) Drop Tables
2) Create Tables
3) Populate Tables
4) Run Advanced Queries
E) Exit
----------------------------------------
Choose an option: 1

SQL*Plus: Release 12.1.0.2.0 Production on Sun Oct 19 18:45:34 2025

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
------------------------------------------
Press Enter to return to the menu...
```

Comment:

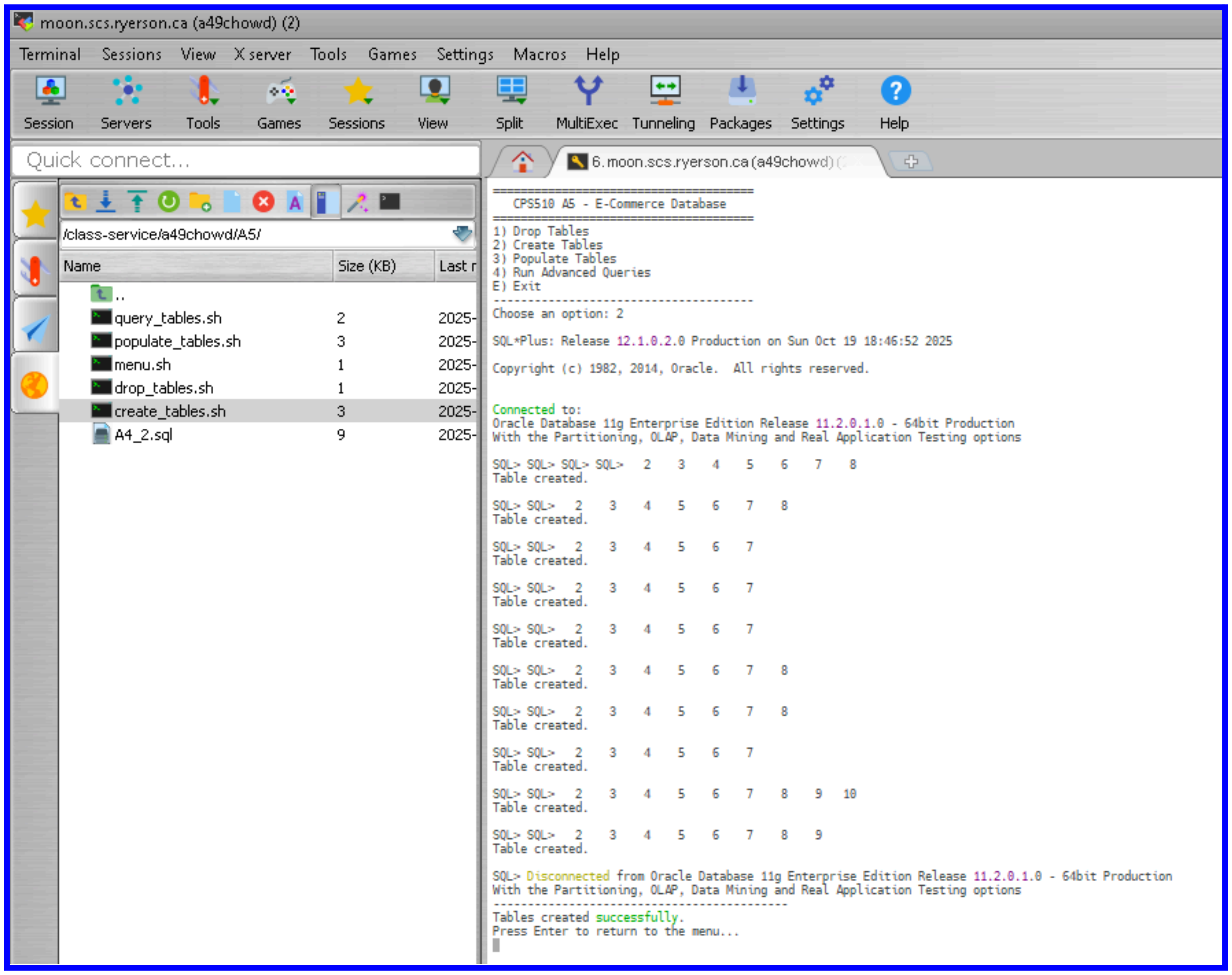
Connects to Oracle database and drops all tables from least dependant to most dependant

create_tables.sh

Code:

```
#!/bin/sh
sqlplus64
"cs_username/cs_password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(
Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
----------------------------------------------------------------
-- CREATE TABLES (structure only)
----------------------------------------------------------------
CREATE TABLE Users (
    UserID     NUMBER PRIMARY KEY,
    FirstName  VARCHAR2(50) NOT NULL,
    LastName   VARCHAR2(50) NOT NULL,
    Email      VARCHAR2(100) UNIQUE NOT NULL,
    Phone      VARCHAR2(15),
    Role       VARCHAR2(20) CHECK (Role IN ('Student','Staff'))
);

CREATE TABLE Staff (
    StaffID    NUMBER PRIMARY KEY,
    Department VARCHAR2(50) NOT NULL,
    Position   VARCHAR2(50),
    HireDate   DATE DEFAULT SYSDATE,
    Salary     NUMBER(10,2) CHECK (Salary >= 0),
    CONSTRAINT fk_staff_user FOREIGN KEY (StaffID) REFERENCES Users(UserID)
);

CREATE TABLE Student (
    StudentID  NUMBER PRIMARY KEY,
    Major      VARCHAR2(50),
    YearLevel  NUMBER(1) CHECK (YearLevel BETWEEN 1 AND 5),
    GPA        NUMBER(3,2) CHECK (GPA BETWEEN 0 AND 4),
    CONSTRAINT fk_student_user FOREIGN KEY (StudentID) REFERENCES Users(UserID)
);

CREATE TABLE Product (
    ProductID     NUMBER PRIMARY KEY,
    Name          VARCHAR2(100) NOT NULL,
    Description   VARCHAR2(255),
    Price         NUMBER(10,2) CHECK (Price > 0),
    StockQuantity NUMBER CHECK (StockQuantity >= 0)
);

CREATE TABLE Orders (
    OrderID   NUMBER PRIMARY KEY,
    UserID    NUMBER NOT NULL,
    OrderDate DATE DEFAULT SYSDATE,
    Status    VARCHAR2(20) CHECK (Status IN ('Pending','Completed','Cancelled')),
    CONSTRAINT fk_orders_user FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

CREATE TABLE Order_Product (
    OrderID   NUMBER NOT NULL,
    ProductID NUMBER NOT NULL,
    Quantity  NUMBER NOT NULL CHECK (Quantity > 0),
    CONSTRAINT pk_order_product PRIMARY KEY (OrderID, ProductID),
    CONSTRAINT fk_op_order FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    CONSTRAINT fk_op_product FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);

CREATE TABLE Payment (
    PaymentID     NUMBER PRIMARY KEY,
    OrderID       NUMBER NOT NULL,
    Amount        NUMBER(10,2) NOT NULL CHECK (Amount > 0),
    PaymentDate   DATE DEFAULT SYSDATE,
    PaymentMethod VARCHAR2(20) CHECK (PaymentMethod IN ('Credit Card','Debit Card','Cash')),
    CONSTRAINT fk_payment_order FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
);

CREATE TABLE Report (
    ReportID      NUMBER PRIMARY KEY,
    StaffID       NUMBER NOT NULL,
    ReportType    VARCHAR2(50) NOT NULL,
    GeneratedDate DATE DEFAULT SYSDATE,
    CONSTRAINT fk_report_staff FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
);

CREATE TABLE Review (
    ReviewID   NUMBER PRIMARY KEY,
    UserID     NUMBER NOT NULL,
    ProductID  NUMBER NOT NULL,
    Rating     NUMBER CHECK (Rating BETWEEN 1 AND 5),
    ReviewComment VARCHAR2(255),
    ReviewDate DATE DEFAULT SYSDATE,
    CONSTRAINT fk_review_user FOREIGN KEY (UserID) REFERENCES Users(UserID),
    CONSTRAINT fk_review_product FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);

CREATE TABLE ReturnRequest (
    ReturnID    NUMBER PRIMARY KEY,
    OrderID     NUMBER NOT NULL,
    ProductID   NUMBER NOT NULL,
    RequestDate DATE DEFAULT SYSDATE,
    Status      VARCHAR2(20) CHECK (Status IN ('Pending','Approved','Rejected')),
    CONSTRAINT fk_return_order FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    CONSTRAINT fk_return_product FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);
exit;
EOF

echo "------------------------------------------"
echo "Tables created successfully."
echo "Press Enter to return to the menu..."
read
```

Screenshot:


moon.scs.ryerson.ca (a49chowd) (2)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
6. moon.scs.ryerson.ca (a49chowd)

```
/class-service/a49chowd/A5/
Name                Size (KB)   Last r
..
query_tables.sh      2          2025-
populate_tables.sh   3          2025-
menu.sh              1          2025-
drop_tables.sh       1          2025-
create_tables.sh     3          2025-
A4_2.sql             9          2025-
```

```
========================================
   CPS510 A5 - E-Commerce Database
========================================
1) Drop Tables
2) Create Tables
3) Populate Tables
4) Run Advanced Queries
E) Exit
----------------------------------------
Choose an option: 2

SQL*Plus: Release 12.1.0.2.0 Production on Sun Oct 19 18:46:52 2025

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9
Table created.

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
------------------------------------------
Tables created successfully.
Press Enter to return to the menu...
```

Comment:

Recreates all tables for the e-commerce system with proper data types primary/foreign keys, and constraints.

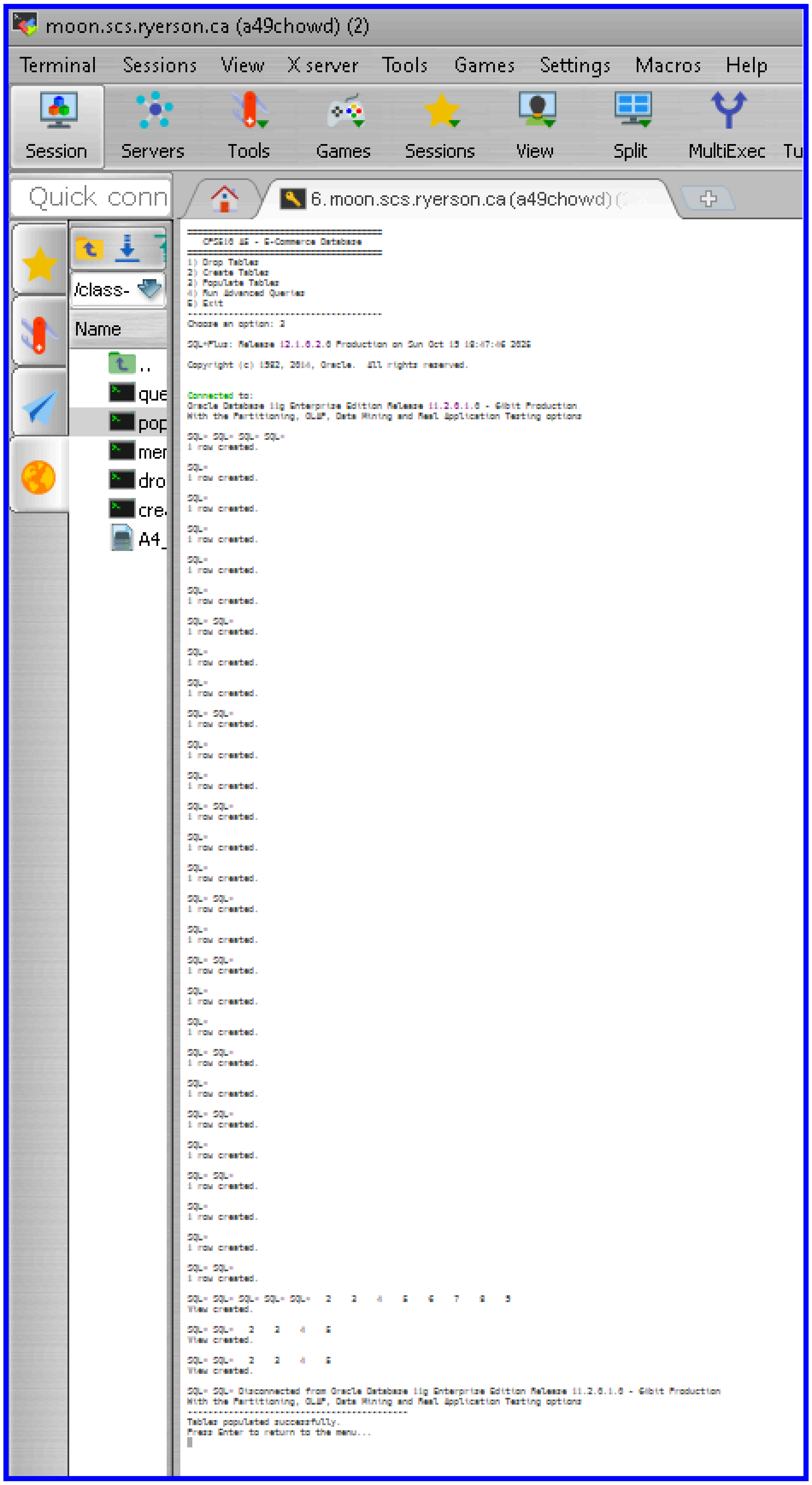
populate_tables.sh

Code:

```
#!/bin/sh
sqlplus64
"cs_username/cs_password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(
Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
------------------------------------------------------------
-- INSERT DATA
------------------------------------------------------------
INSERT INTO Users VALUES (1,'Eshwar','Vetrichelvan','evetrichelvan@torontomu.ca','501111111','Student');
INSERT INTO Users VALUES (2,'Ashwin','Jakanathan','ajakanathan@torontomu.ca','501222222','Student');
INSERT INTO Users VALUES (3,'Jihan','Chowdury','jchowdury@torontomu.ca','501333333','Student');
INSERT INTO Users VALUES (4,'Clara','Lee','clee@torontomu.ca','416444444','Staff');
INSERT INTO Users VALUES (5,'David','Wong','dwong@torontomu.ca','416555555','Staff');
INSERT INTO Users VALUES (6,'Jassar','Surinder','jsurinder@torontomu.ca','416666666','Staff');

INSERT INTO Staff VALUES (4,'IT Services','Technician',DATE '2022-01-15',60000);
INSERT INTO Staff VALUES (5,'Admin','Manager',DATE '2021-06-01',75000);
INSERT INTO Staff VALUES (6,'Admin','Professor',DATE '2023-04-12',75000);

INSERT INTO Student VALUES (1,'Computer Engineering',3,3.2);
INSERT INTO Student VALUES (2,'Computer Engineering',2,3.2);
INSERT INTO Student VALUES (3,'Computer Engineering',3,3.2);

INSERT INTO Product VALUES (101,'Database Textbook','Intro to Oracle SQL',79.99,10);
INSERT INTO Product VALUES (102,'Laptop','14-inch lightweight laptop',899.99,5);
INSERT INTO Product VALUES (103,'Headphones','Noise-cancelling wireless headphones',199.99,15);

INSERT INTO Orders VALUES (5001,1,SYSDATE-7,'Completed');
INSERT INTO Orders VALUES (5002,2,SYSDATE-3,'Completed');

INSERT INTO Order_Product VALUES (5001,101,2);
INSERT INTO Order_Product VALUES (5001,103,1);
INSERT INTO Order_Product VALUES (5002,102,1);

INSERT INTO Payment VALUES (9001,5001,359.97,SYSDATE-6,'Credit Card');
INSERT INTO Payment VALUES (9002,5002,899.99,SYSDATE-2,'Debit Card');

INSERT INTO Report VALUES (7001,4,'Sales Report',SYSDATE-1);
INSERT INTO Report VALUES (7002,5,'User Activity Report',SYSDATE-2);

INSERT INTO Review VALUES (8001,1,101,5,'Great textbook for SQL learning!',SYSDATE-5);
INSERT INTO Review VALUES (8002,2,102,4,'Good laptop, battery health is okay.',SYSDATE-2);
INSERT INTO Review VALUES (8003,1,103,5,'Headphones have clear audio.',SYSDATE-1);

INSERT INTO ReturnRequest VALUES (6001,5001,103,SYSDATE,'Pending');

------------------------------------------------------------
-- CREATE VIEWS
------------------------------------------------------------
CREATE OR REPLACE VIEW Staff_Report_Summary AS
SELECT s.StaffID,
      u.FirstName||' '||u.LastName AS StaffName,
      COUNT(r.ReportID) AS ReportCount,
      MAX(r.GeneratedDate) AS LastGenerated
FROM Staff s
JOIN Users u ON s.StaffID=u.UserID
LEFT JOIN Report r ON s.StaffID=r.StaffID
GROUP BY s.StaffID,u.FirstName,u.LastName;

CREATE OR REPLACE VIEW VW_TOP_RATED_PRODUCTS AS
SELECT p.ProductID,p.Name AS ProductName,AVG(r.Rating) AS AvgRating,COUNT(r.ReviewID) AS
ReviewCount
```

```
FROM Product p JOIN Review r ON p.ProductID=r.ProductID
GROUP BY p.ProductID,p.Name
ORDER BY AvgRating DESC;

CREATE OR REPLACE VIEW VW_SALES_SUMMARY AS
SELECT p.ProductID,p.Name AS ProductName,SUM(op.Quantity) AS
TotalUnitsSold,SUM(op.Quantity*p.Price) AS TotalRevenue
FROM Product p JOIN Order_Product op ON p.ProductID=op.ProductID
GROUP BY p.ProductID,p.Name
ORDER BY TotalRevenue DESC;

exit;
EOF

echo "--------------------------------------------"
echo "Tables populated successfully."
echo "Press Enter to return to the menu..."
read
```

Screenshot:


moon.scs.ryerson.ca (a49chowd) (2)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tu
Quick conn
6. moon.scs.ryerson.ca (a49chowd)
/class-
Name
..
que
pop
mer
dro
cre
A4_

```
==========================================
   CPS510 A5 - E-Commerce Database
==========================================
1) Drop Tables
2) Create Tables
3) Populate Tables
4) Run Advanced Queries
5) Exit
------------------------------------------
Choose an option: 3

SQL*Plus: Release 12.1.0.2.0 Production on Sun Oct 19 18:47:46 2025

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL> SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8    9
View created.

SQL> SQL>   2    3    4    5
View created.

SQL> SQL>   2    3    4    5
View created.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
------------------------------------------
Tables populated successfully.
Press Enter to return to the menu...
```

Comment:

Inserts data into all tables. Populates users, staff, students, products, orders, payments, reports, and reviews and creates 3 views for summary queries.

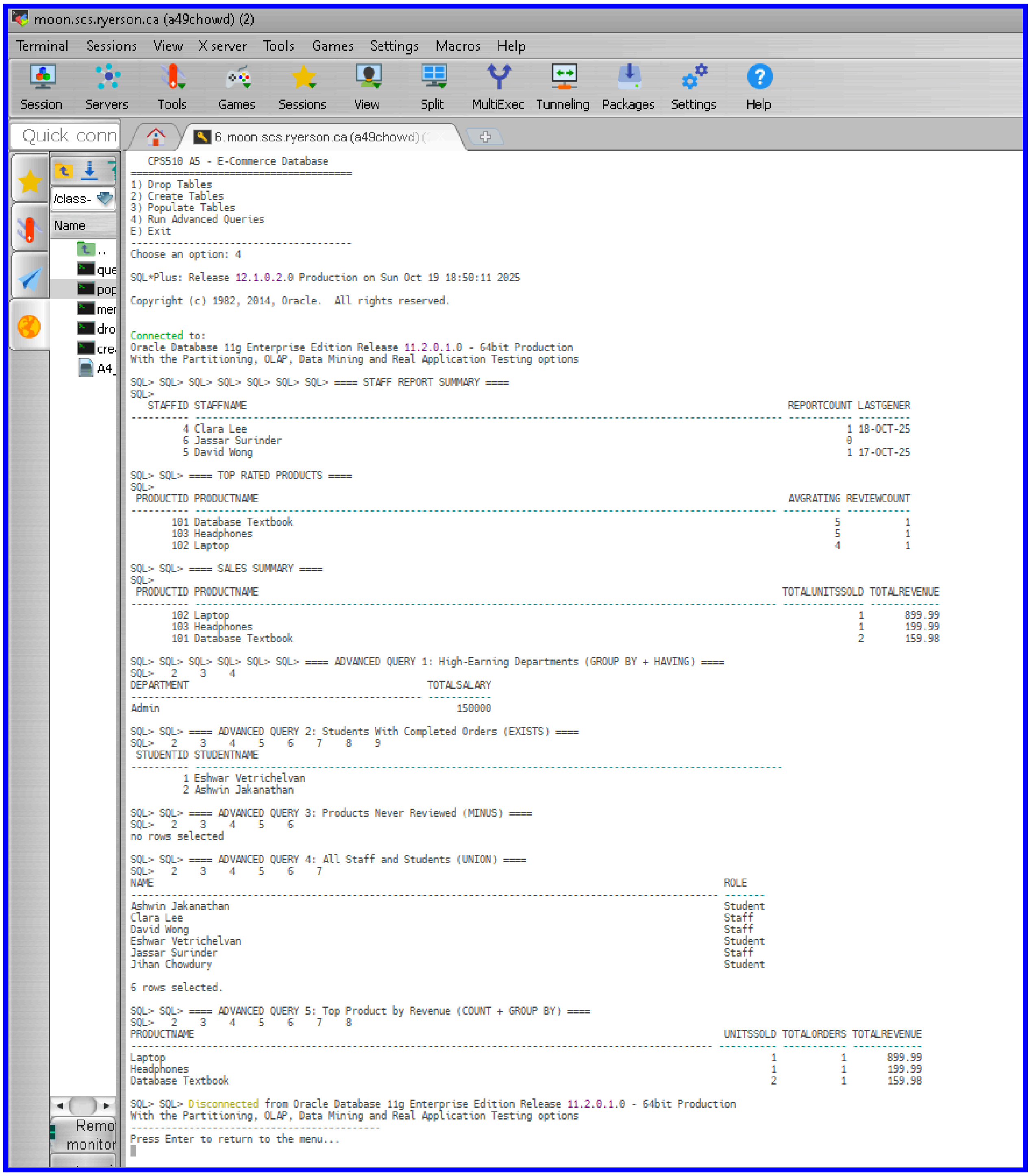
query_tables.sh

Code:

```
 #!/bin/sh
sqlplus64
"cs_username/cs_password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(
Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
SET LINESIZE 200;
SET PAGESIZE 50;

------------------------------------------------------------
-- VIEW QUERIES (3 Views)
------------------------------------------------------------
PROMPT ==== STAFF REPORT SUMMARY ====
SELECT * FROM STAFF_REPORT_SUMMARY;

PROMPT ==== TOP RATED PRODUCTS ====
SELECT * FROM VW_TOP_RATED_PRODUCTS;

PROMPT ==== SALES SUMMARY ====
SELECT * FROM VW_SALES_SUMMARY;

------------------------------------------------------------
-- ADVANCED QUERIES (5 Required for Full Marks)
------------------------------------------------------------

PROMPT ==== ADVANCED QUERY 1: High-Earning Departments (GROUP BY + HAVING) ====
SELECT Department, SUM(Salary) AS TotalSalary
FROM Staff
GROUP BY Department
HAVING SUM(Salary) > 60000;

PROMPT ==== ADVANCED QUERY 2: Students With Completed Orders (EXISTS) ====
SELECT s.StudentID, u.FirstName || ' ' || u.LastName AS StudentName
FROM Student s
JOIN Users u ON s.StudentID = u.UserID
WHERE EXISTS (
   SELECT 1
   FROM Orders o
   WHERE o.UserID = s.StudentID
     AND o.Status = 'Completed'
);
```

```
PROMPT ==== ADVANCED QUERY 3: Products Never Reviewed (MINUS) ====
SELECT Name AS UnreviewedProduct
FROM Product
MINUS
SELECT DISTINCT p.Name
FROM Product p
JOIN Review r ON p.ProductID = r.ProductID;

PROMPT ==== ADVANCED QUERY 4: All Staff and Students (UNION) ====
SELECT u.FirstName || ' ' || u.LastName AS Name, 'Staff' AS Role
FROM Users u
WHERE Role = 'Staff'
UNION
SELECT u.FirstName || ' ' || u.LastName AS Name, 'Student' AS Role
FROM Users u
WHERE Role = 'Student';

PROMPT ==== ADVANCED QUERY 5: Top Product by Revenue (COUNT + GROUP BY) ====
SELECT p.Name AS ProductName,
       SUM(op.Quantity) AS UnitsSold,
       COUNT(op.OrderID) AS TotalOrders,
       SUM(op.Quantity * p.Price) AS TotalRevenue
FROM Product p
JOIN Order_Product op ON p.ProductID = op.ProductID
GROUP BY p.Name
ORDER BY TotalRevenue DESC;

exit;
EOF

echo "------------------------------------------"
echo "Press Enter to return to the menu..."
read
```

Screenshot:

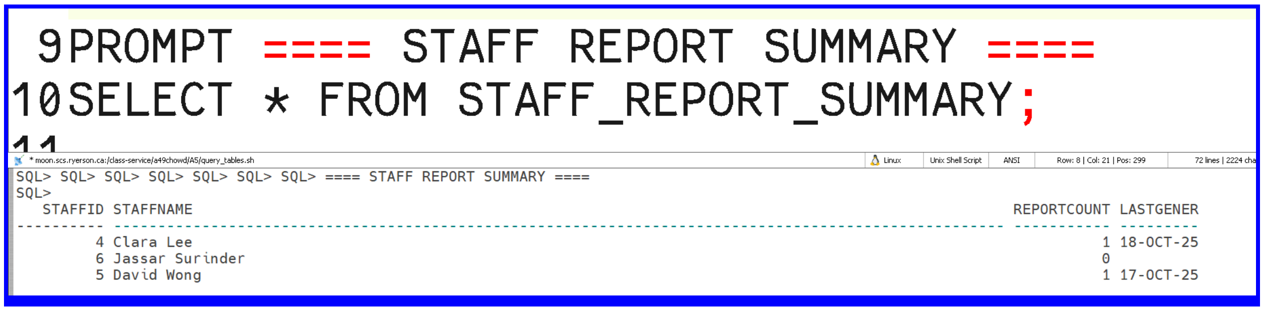

moon.scs.ryerson.ca (a49chowd) (2)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick conn
6. moon.scs.ryerson.ca (a49chowd)
/class-
Name
que
pop
mer
dro
cre
A4_
Remo
monitor

```
   CPS510 A5 - E-Commerce Database
======================================
1) Drop Tables
2) Create Tables
3) Populate Tables
4) Run Advanced Queries
E) Exit
--------------------------------------
Choose an option: 4

SQL*Plus: Release 12.1.0.2.0 Production on Sun Oct 19 18:50:11 2025

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> SQL> SQL> SQL> ==== STAFF REPORT SUMMARY ====
SQL>
   STAFFID STAFFNAME                                                                                          REPORTCOUNT LASTGENER
---------- ---------------------------------------------------------------------------------------------------- ----------- ---------
         4 Clara Lee                                                                                                    1 18-OCT-25
         6 Jassar Surinder                                                                                              0
         5 David Wong                                                                                                   1 17-OCT-25

SQL> SQL> ==== TOP RATED PRODUCTS ====
SQL>
 PRODUCTID PRODUCTNAME                                                                                           AVGRATING REVIEWCOUNT
---------- ---------------------------------------------------------------------------------------------------- ---------- -----------
       101 Database Textbook                                                                                             5           1
       103 Headphones                                                                                                    5           1
       102 Laptop                                                                                                        4           1

SQL> SQL> ==== SALES SUMMARY ====
SQL>
 PRODUCTID PRODUCTNAME                                                                                          TOTALUNITSSOLD TOTALREVENUE
---------- ---------------------------------------------------------------------------------------------------- -------------- ------------
       102 Laptop                                                                                                            1       899.99
       103 Headphones                                                                                                        1       199.99
       101 Database Textbook                                                                                                 2       159.98

SQL> SQL> SQL> SQL> SQL> SQL> ==== ADVANCED QUERY 1: High-Earning Departments (GROUP BY + HAVING) ====
SQL>   2    3    4
DEPARTMENT                                         TOTALSALARY
-------------------------------------------------- -----------
Admin                                                   150000

SQL> SQL> ==== ADVANCED QUERY 2: Students With Completed Orders (EXISTS) ====
SQL>   2    3    4    5    6    7    8    9
 STUDENTID STUDENTNAME
---------- ----------------------------------------------------------------------------------------------------
         1 Eshwar Vetrichelvan
         2 Ashwin Jakanathan

SQL> SQL> ==== ADVANCED QUERY 3: Products Never Reviewed (MINUS) ====
SQL>   2    3    4    5    6
no rows selected

SQL> SQL> ==== ADVANCED QUERY 4: All Staff and Students (UNION) ====
SQL>   2    3    4    5    6    7
NAME                                                                                                  ROLE
----------------------------------------------------------------------------------------------------- -------
Ashwin Jakanathan                                                                                     Student
Clara Lee                                                                                             Staff
David Wong                                                                                            Staff
Eshwar Vetrichelvan                                                                                   Student
Jassar Surinder                                                                                       Staff
Jihan Chowdury                                                                                        Student

6 rows selected.

SQL> SQL> ==== ADVANCED QUERY 5: Top Product by Revenue (COUNT + GROUP BY) ====
SQL>   2    3    4    5    6    7    8
PRODUCTNAME                                                                                           UNITSSOLD TOTALORDERS TOTALREVENUE
---------------------------------------------------------------------------------------------------- ---------- ----------- ------------
Laptop                                                                                                        1           1       899.99
Headphones                                                                                                    1           1       199.99
Database Textbook                                                                                             2           1       159.98

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
----------------------------------------
Press Enter to return to the menu...
```

Comment:

Runs the 3 view queries and 5 advanced queries.

Simple and Advanced Queries


* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Script
ANSI
Row: 8 | Col: 21 | Pos: 299
72 lines | 2224 cha

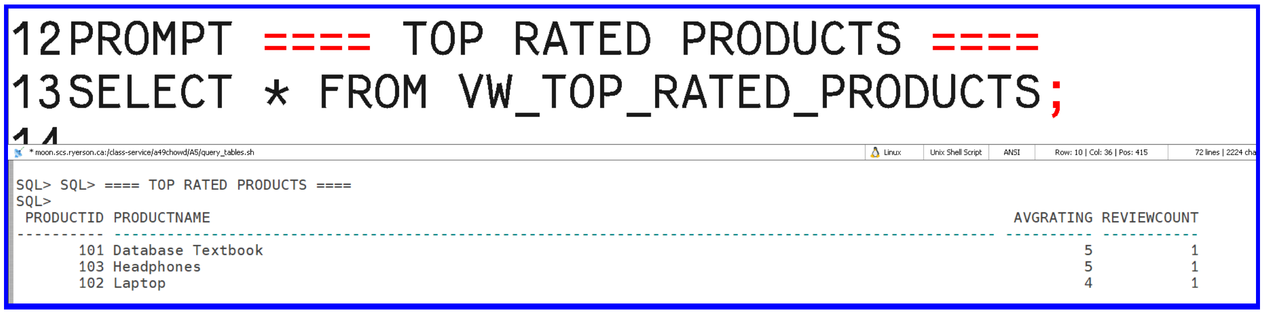
```
PROMPT ==== STAFF REPORT SUMMARY ====
SELECT * FROM STAFF_REPORT_SUMMARY;
```

```
SQL> SQL> SQL> SQL> SQL> SQL> SQL> ==== STAFF REPORT SUMMARY ====
SQL>
   STAFFID STAFFNAME                                                                                          REPORTCOUNT LASTGENER
---------- -------------------------------------------------------------------------------------------------- ----------- ---------
         4 Clara Lee                                                                                                    1 18-OCT-25
         6 Jassar Surinder                                                                                              0
         5 David Wong                                                                                                   1 17-OCT-25
```

Comment:

Shows how many reports each staff member has created

Shows all

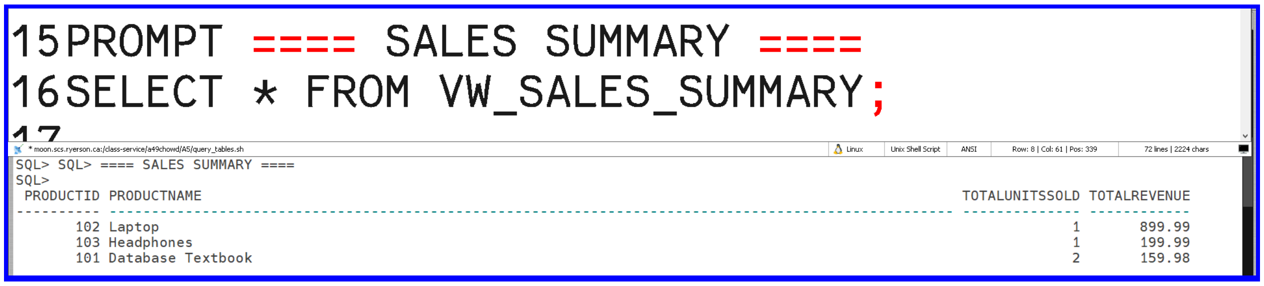

* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Script
ANSI
Row: 10 | Col: 36 | Pos: 415
72 lines | 2224 cha

```
PROMPT ==== TOP RATED PRODUCTS ====
SELECT * FROM VW_TOP_RATED_PRODUCTS;
```

```
SQL> SQL> ==== TOP RATED PRODUCTS ====
SQL>
 PRODUCTID PRODUCTNAME                                                                                   AVGRATING REVIEWCOUNT
---------- ---------------------------------------------------------------------------------------------- --------- -----------
       101 Database Textbook                                                                                      5           1
       103 Headphones                                                                                             5           1
       102 Laptop                                                                                                 4           1
```

Comment:

Lists all products with average rating in descending order and number of reviews.

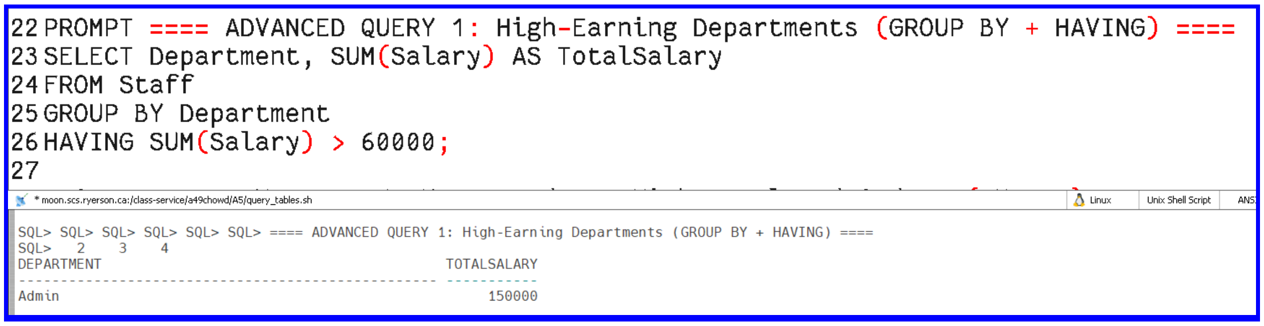

* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Script
ANSI
Row: 8 | Col: 61 | Pos: 339
72 lines | 2224 chars

```
PROMPT ==== SALES SUMMARY ====
SELECT * FROM VW_SALES_SUMMARY;
```

```
SQL> SQL> ==== SALES SUMMARY ====
SQL>
 PRODUCTID PRODUCTNAME                                                                               TOTALUNITSSOLD TOTALREVENUE
---------- ----------------------------------------------------------------------------------------- -------------- ------------
       102 Laptop                                                                                                 1       899.99
       103 Headphones                                                                                             1       199.99
       101 Database Textbook                                                                                      2       159.98
```

Comment:

Calculates and shows total revenue and units sold for each product.

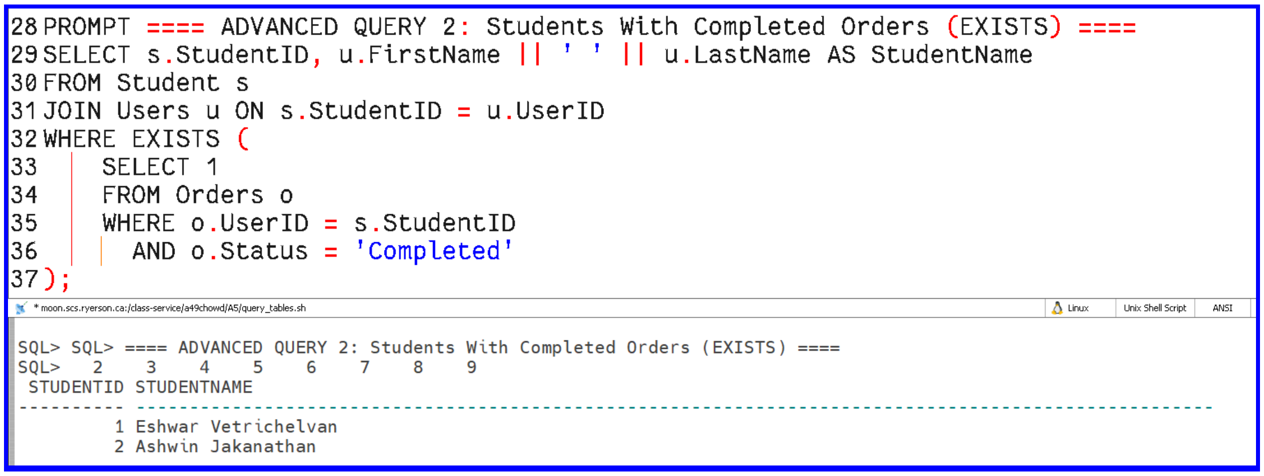

* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Script
ANSI

```
PROMPT ==== ADVANCED QUERY 1: High-Earning Departments (GROUP BY + HAVING) ====
SELECT Department, SUM(Salary) AS TotalSalary
FROM Staff
GROUP BY Department
HAVING SUM(Salary) > 60000;

```

```
SQL> SQL> SQL> SQL> SQL> SQL> ==== ADVANCED QUERY 1: High-Earning Departments (GROUP BY + HAVING) ====
SQL>   2    3    4
DEPARTMENT                                           TOTALSALARY
-------------------------------------------------- -----------
Admin                                                     150000
```

Comment:

Groups all staff by department and sums their salaries and only displays the departments with salary > 60000.

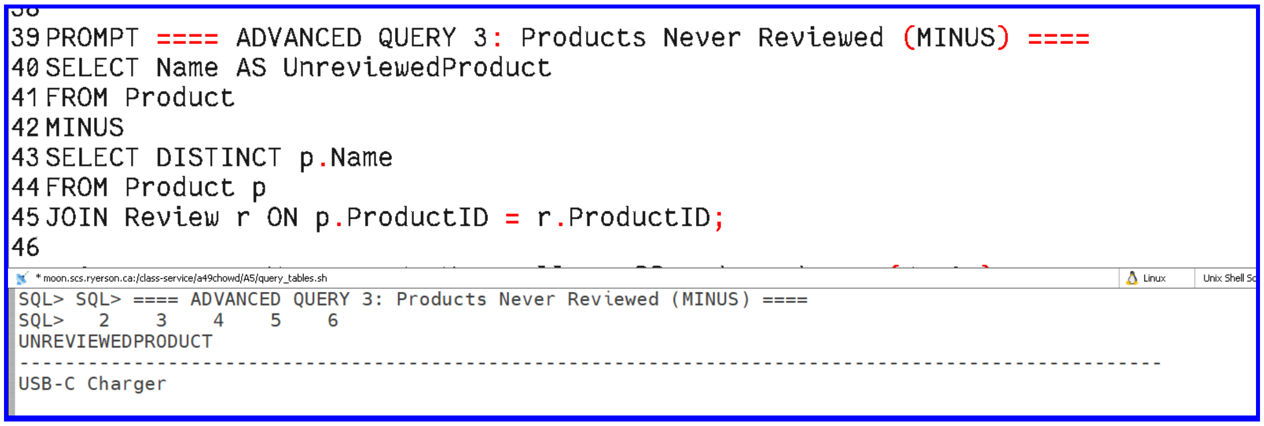

* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Script
ANSI

```
PROMPT ==== ADVANCED QUERY 2: Students With Completed Orders (EXISTS) ====
SELECT s.StudentID, u.FirstName || ' ' || u.LastName AS StudentName
FROM Student s
JOIN Users u ON s.StudentID = u.UserID
WHERE EXISTS (
    SELECT 1
    FROM Orders o
    WHERE o.UserID = s.StudentID
      AND o.Status = 'Completed'
);
```

```
SQL> SQL> ==== ADVANCED QUERY 2: Students With Completed Orders (EXISTS) ====
SQL>   2    3    4    5    6    7    8    9
 STUDENTID STUDENTNAME
---------- ----------------------------------------------------------------------------------------------------
         1 Eshwar Vetrichelvan
         2 Ashwin Jakanathan
```

Comment:

Lists students have at least one completed order. Checks if at least one record exists of a student inside the orders and returns that student.

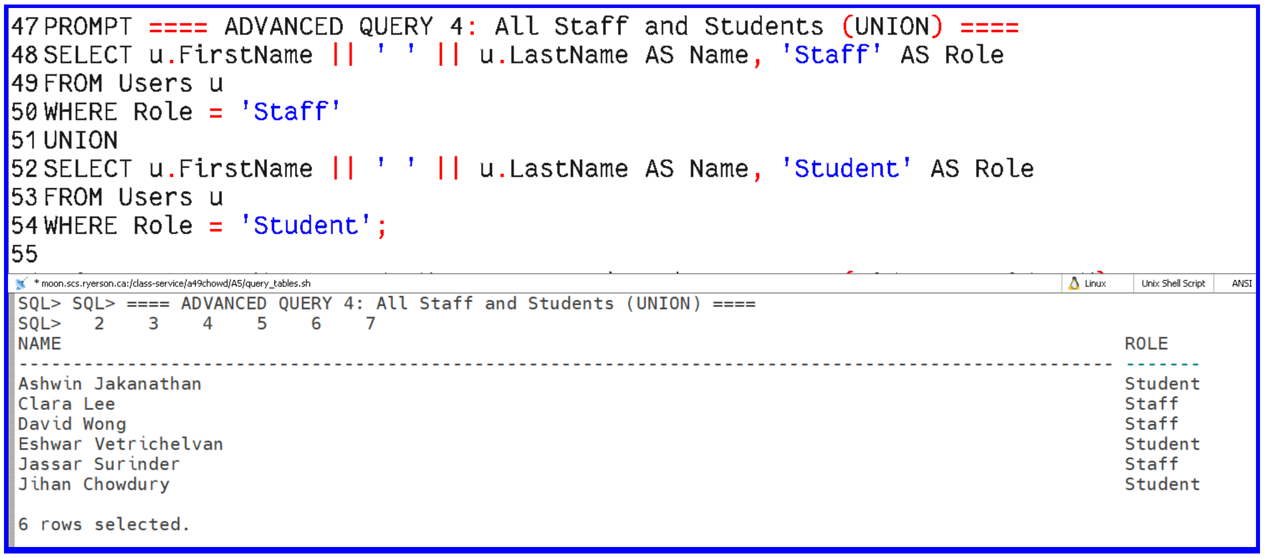

* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Sc

```
PROMPT ==== ADVANCED QUERY 3: Products Never Reviewed (MINUS) ====
SELECT Name AS UnreviewedProduct
FROM Product
MINUS
SELECT DISTINCT p.Name
FROM Product p
JOIN Review r ON p.ProductID = r.ProductID;

```

```
SQL> SQL> ==== ADVANCED QUERY 3: Products Never Reviewed (MINUS) ====
SQL>   2    3    4    5    6
UNREVIEWEDPRODUCT
----------------------------------------------------------------------------------------------------
USB-C Charger
```

Comment:

Shows products that have no reviews. Selects all products, and then uses MINUS to remove the products that appear in the Review table.

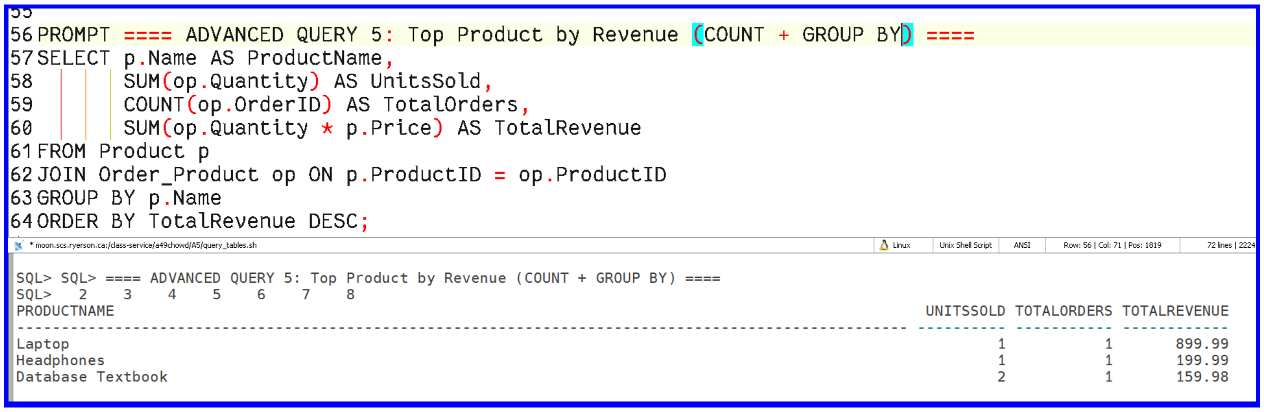

* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Script
ANSI

```
PROMPT ==== ADVANCED QUERY 4: All Staff and Students (UNION) ====
SELECT u.FirstName || ' ' || u.LastName AS Name, 'Staff' AS Role
FROM Users u
WHERE Role = 'Staff'
UNION
SELECT u.FirstName || ' ' || u.LastName AS Name, 'Student' AS Role
FROM Users u
WHERE Role = 'Student';
```

```
SQL> SQL> ==== ADVANCED QUERY 4: All Staff and Students (UNION) ====
SQL>   2    3    4    5    6    7
NAME                                                                                                  ROLE
---------------------------------------------------------------------------------------------------- -------
Ashwin Jakanathan                                                                                     Student
Clara Lee                                                                                             Staff
David Wong                                                                                            Staff
Eshwar Vetrichelvan                                                                                   Student
Jassar Surinder                                                                                       Staff
Jihan Chowdury                                                                                        Student

6 rows selected.
```

Comment:

Shows all staff and student names in a single list using UNION.


* moon.scs.ryerson.ca:/class-service/a49chowd/A5/query_tables.sh
Linux
Unix Shell Script
ANSI
Row: 56 | Col: 71 | Pos: 1819
72 lines | 2224

```
PROMPT ==== ADVANCED QUERY 5: Top Product by Revenue (COUNT + GROUP BY) ====
SELECT p.Name AS ProductName,
       SUM(op.Quantity) AS UnitsSold,
       COUNT(op.OrderID) AS TotalOrders,
       SUM(op.Quantity * p.Price) AS TotalRevenue
FROM Product p
JOIN Order_Product op ON p.ProductID = op.ProductID
GROUP BY p.Name
ORDER BY TotalRevenue DESC;
```

```
SQL> SQL> ==== ADVANCED QUERY 5: Top Product by Revenue (COUNT + GROUP BY) ====
SQL>   2    3    4    5    6    7    8
PRODUCTNAME                                                                                      UNITSSOLD TOTALORDERS TOTALREVENUE
---------------------------------------------------------------------------------------------------- ---------- ----------- ------------
Laptop                                                                                                   1           1       899.99
Headphones                                                                                               1           1       199.99
Database Textbook                                                                                        2           1       159.98
```

Comment:

Shows each product’s total units sold, no. of orders, and total revenue.