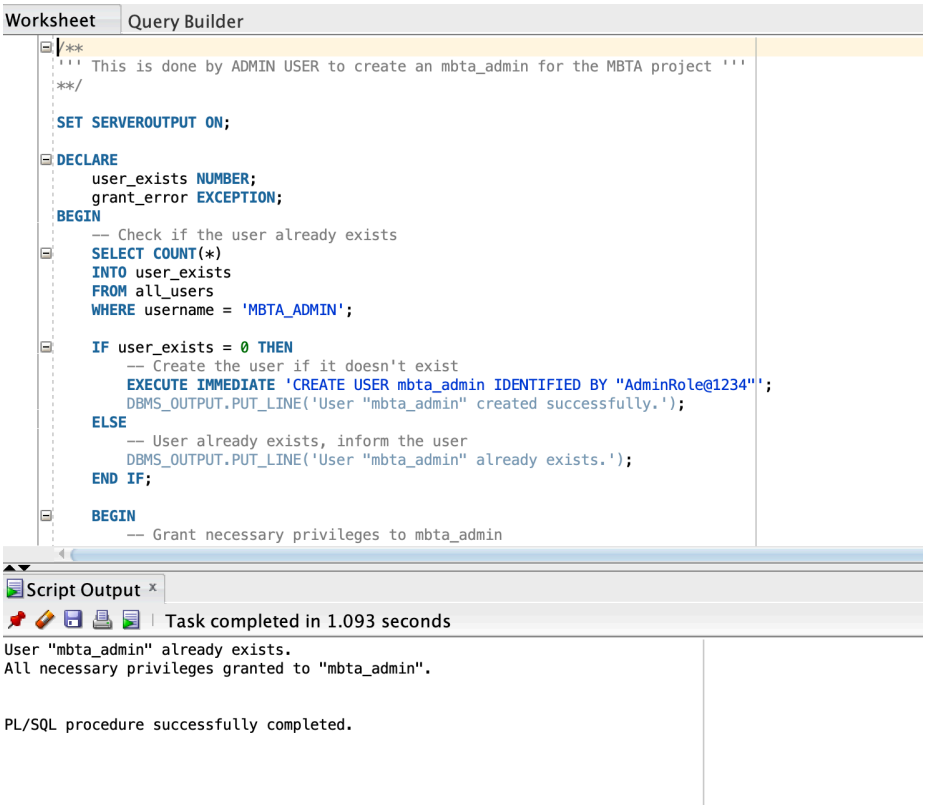
DMDD Phase 3
Team 16

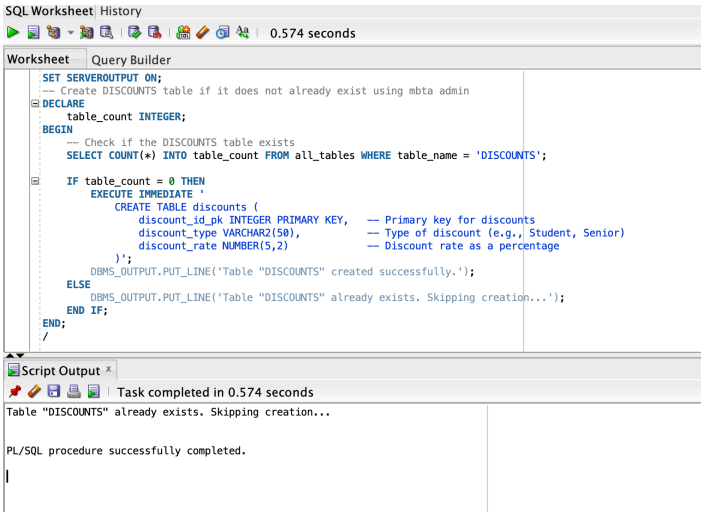
Step 1: Create MBTA_Admin user using oracle admin
File Name: Roles/Mbta_Admin_User_Creation_1.sql



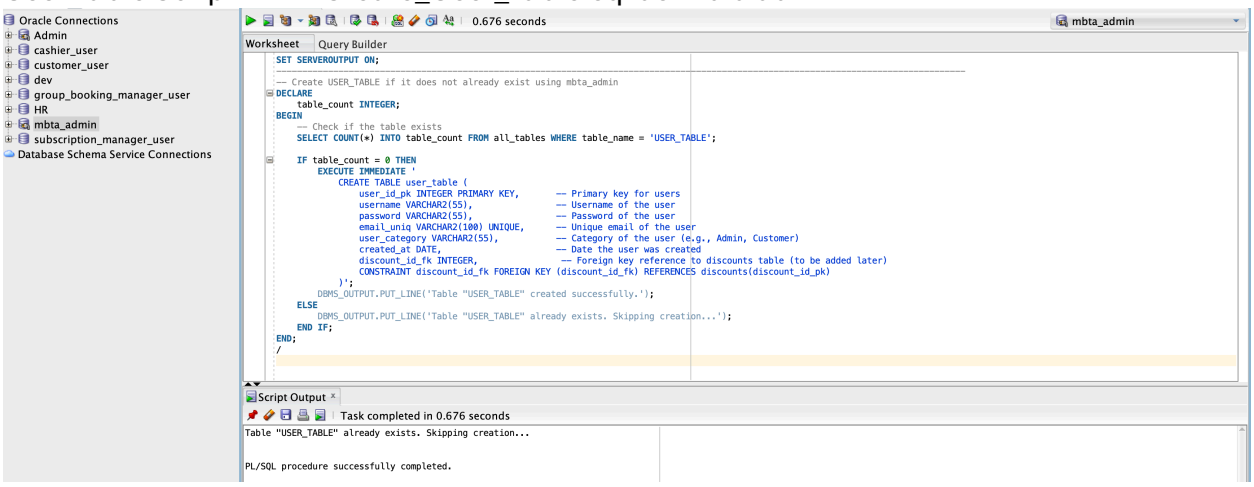
Login to Mbta_admin and then start creating tables

Step 2(DDL) : Create Tables in the order mentioned using MBTA_Admin to avoid foreign Key constraints

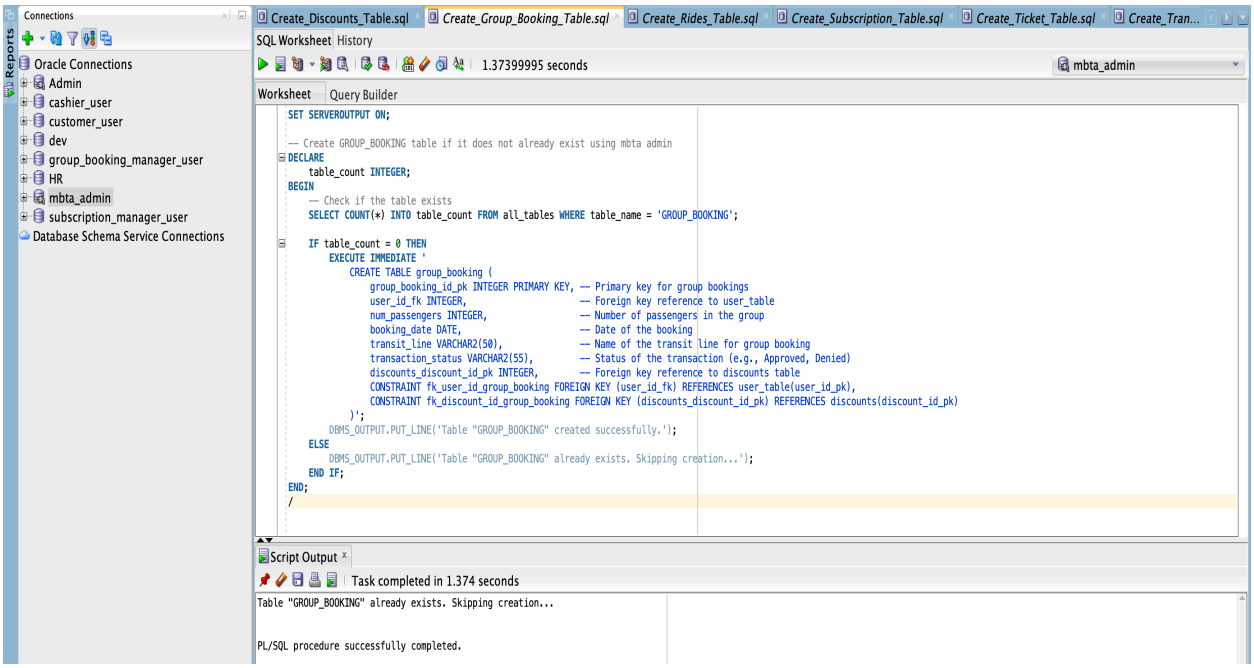
- Discount Table script: DDL/Create_Discounts_Table.sql



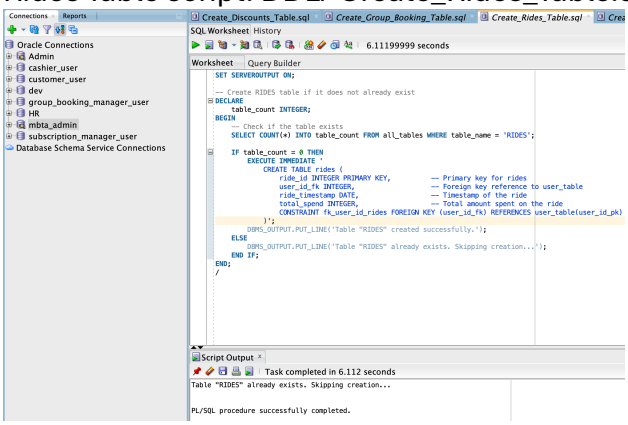
- User Table Script: DDL/Create_User_Table.sql as mbta admin



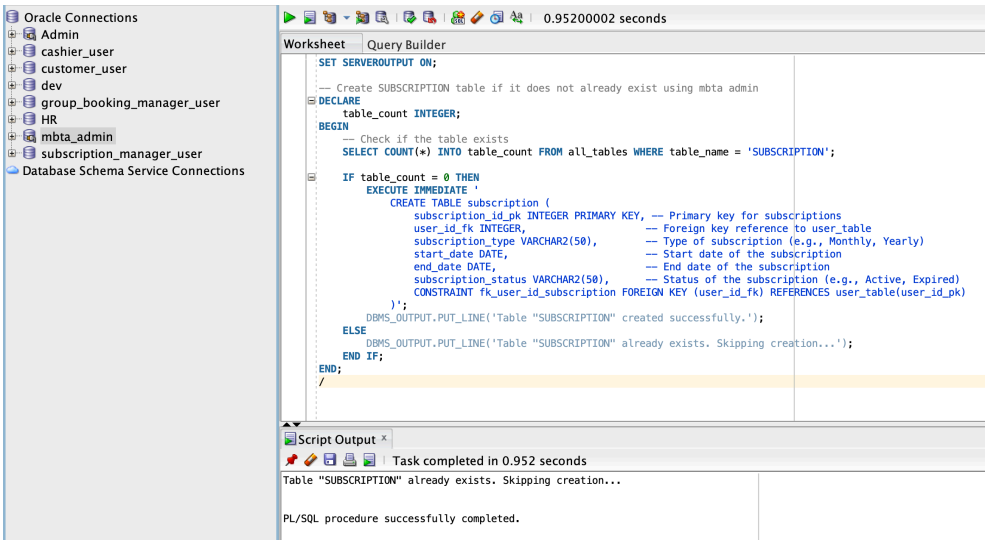
- Group Booking table script: DDL/Create_Group_Booking_Table.sql as mbta admin



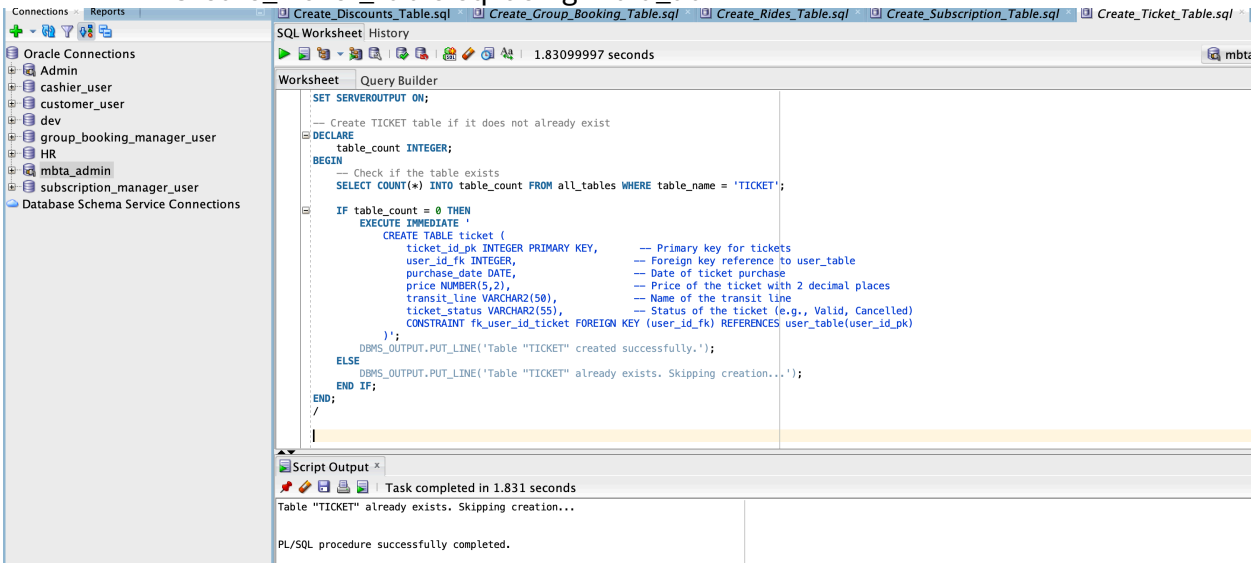
- Rides Table script: DDL/ Create_Rides_Table.sql using mbta_admin



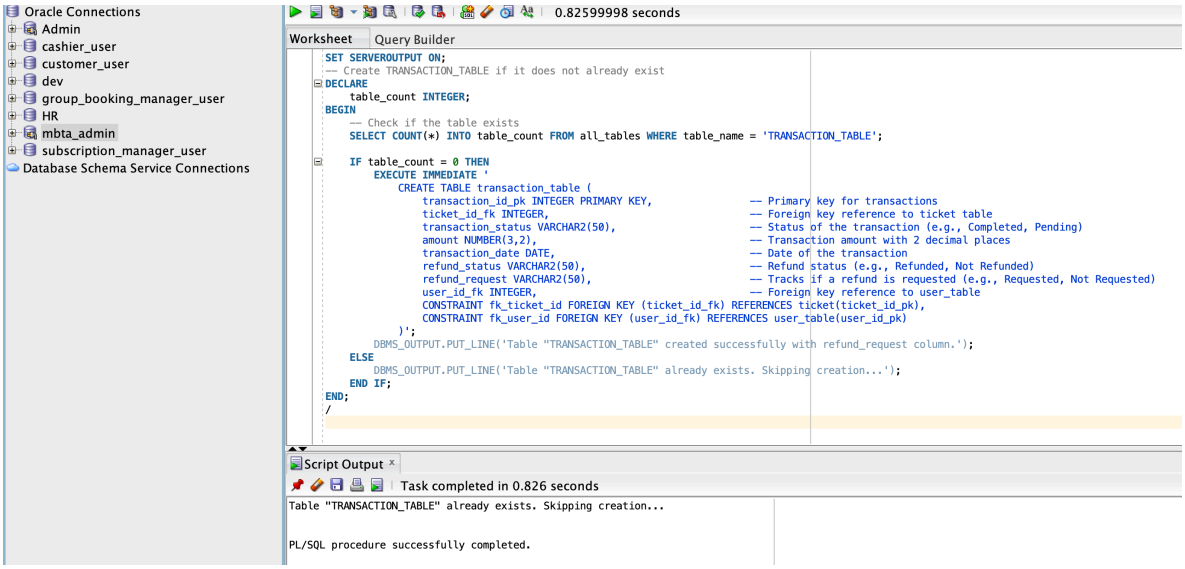
- subcription table: DDL/ Create_Subscription_Table.sql using mbta_admin



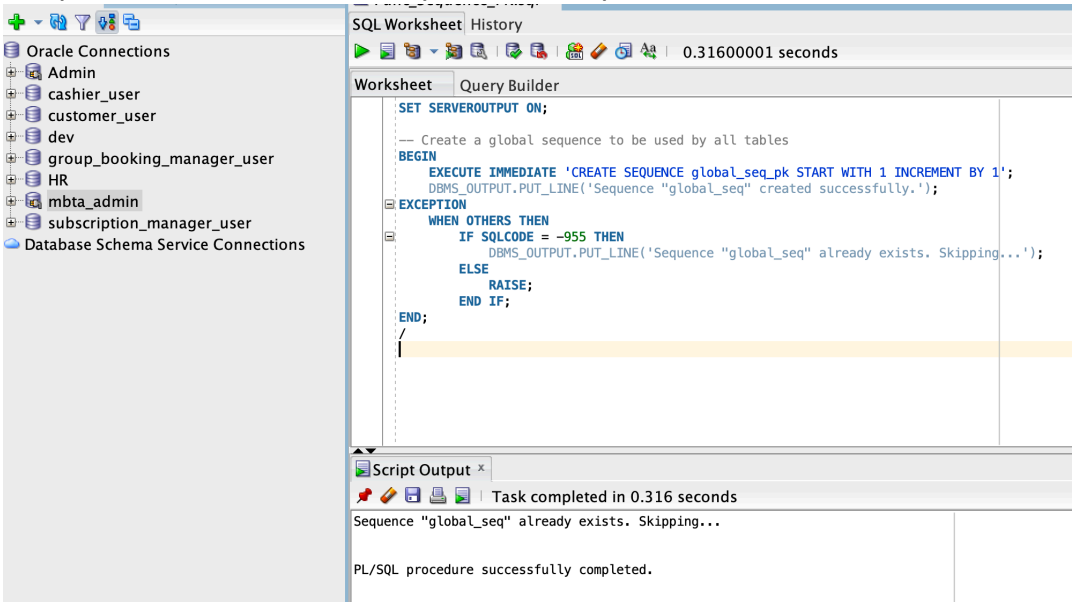
- DDL/Create_Ticket_Table.sql using mbta_admin



- DDL/ Create_Transaction_Table.sql using mbta_admin

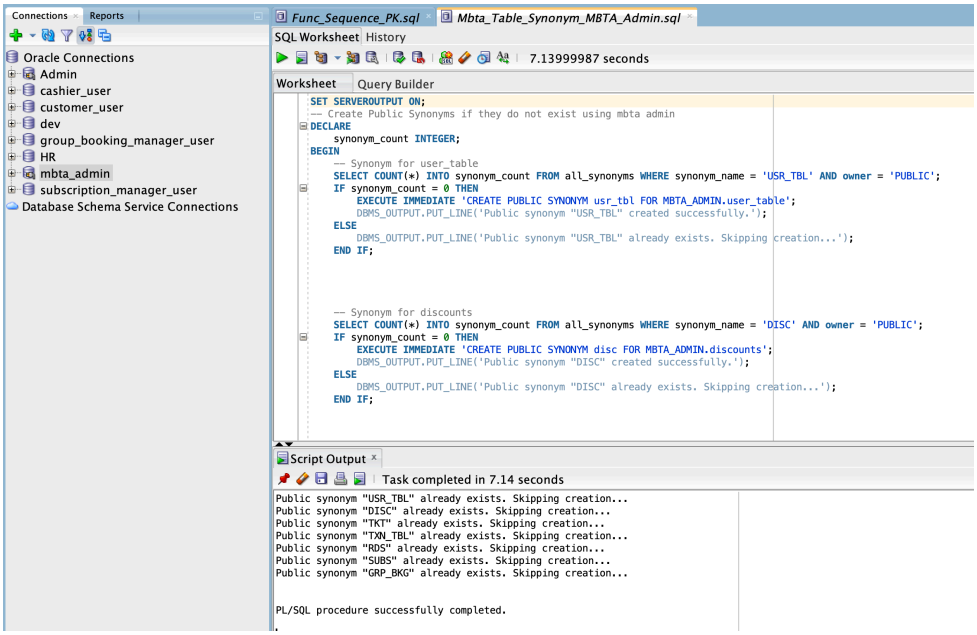


Step 3 : Execute **sequence function only once** to generate primary key for each table when insertion happens (use mbta admin)
File path: Functions/ Func_Sequence_PK.sql



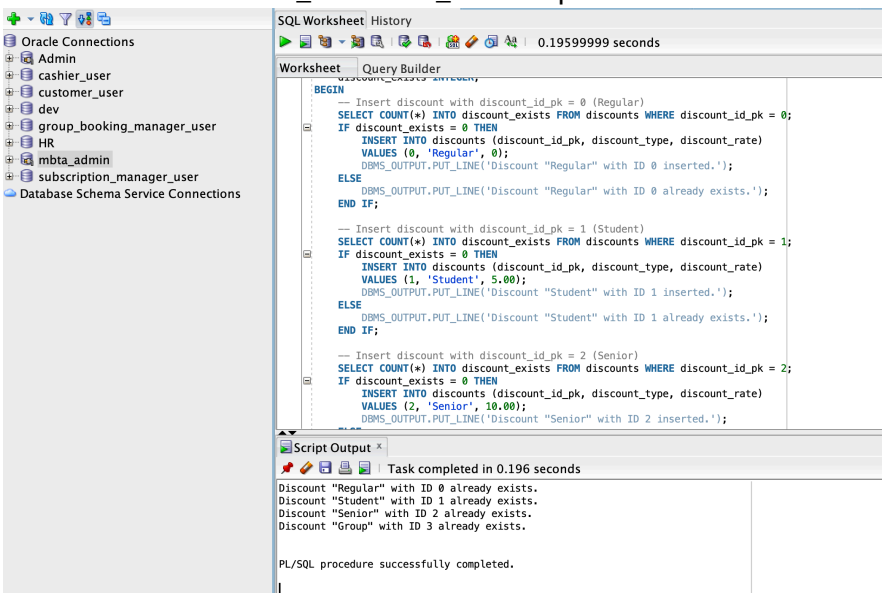
Step 4: CREATE Synonym for all the tables using mbta admin

File Path: Roles/Mbta_Table_Synonym_MBTA_Admin.sql

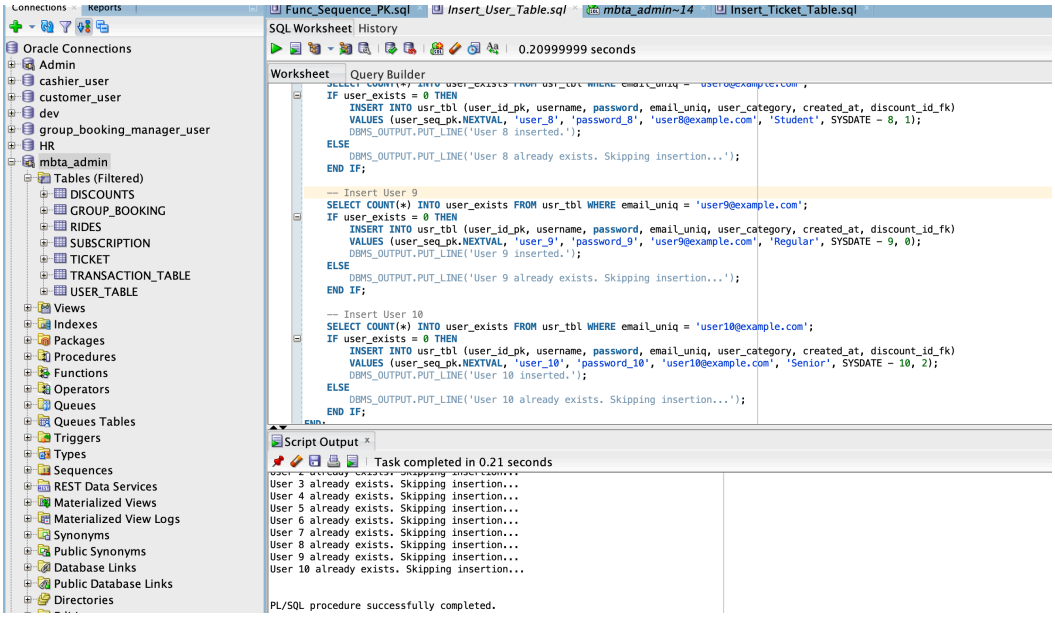


STEP 5(DML): Insert Data Into Tables using mbta admin
***** Insert data in the order mentioned**

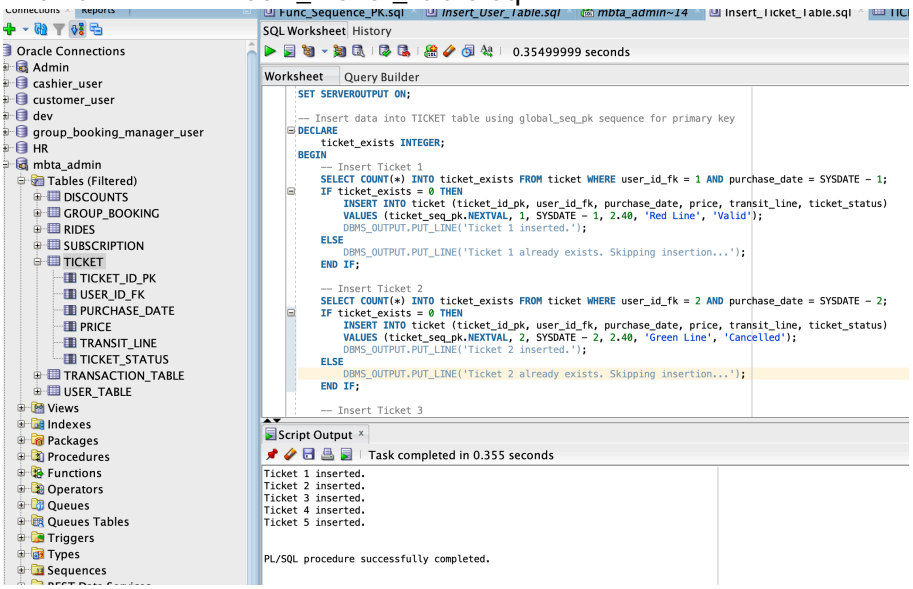
- File Path: DML/ Insert_Discount_Table.sql:



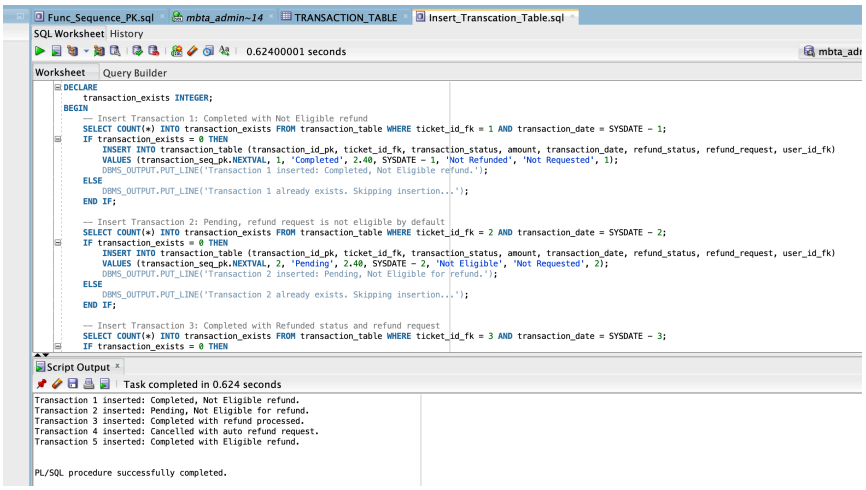
- File Path: DML/ Insert_User_Table.sql



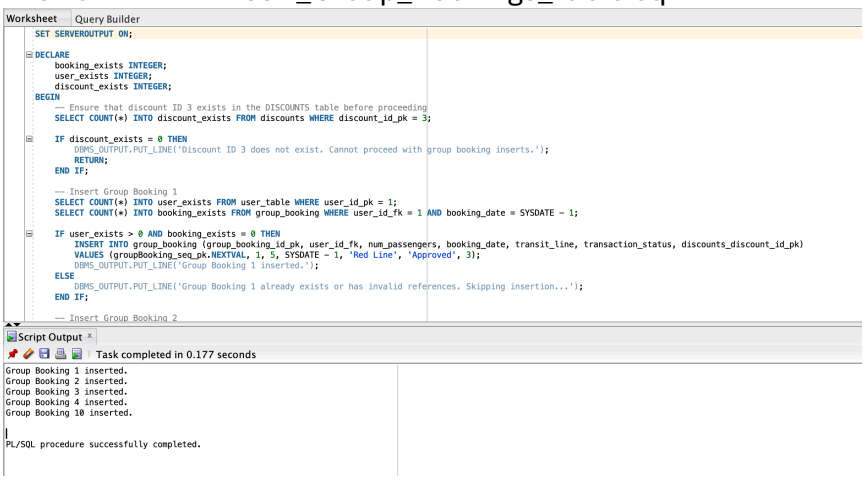
- File Path: DML/ Insert_Ticket_Table.sql



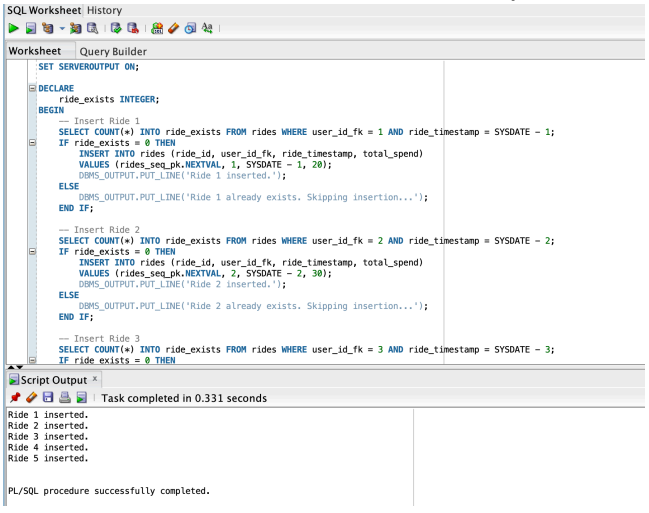
- File Path: DML/ Insert_Transcation_Table.sql



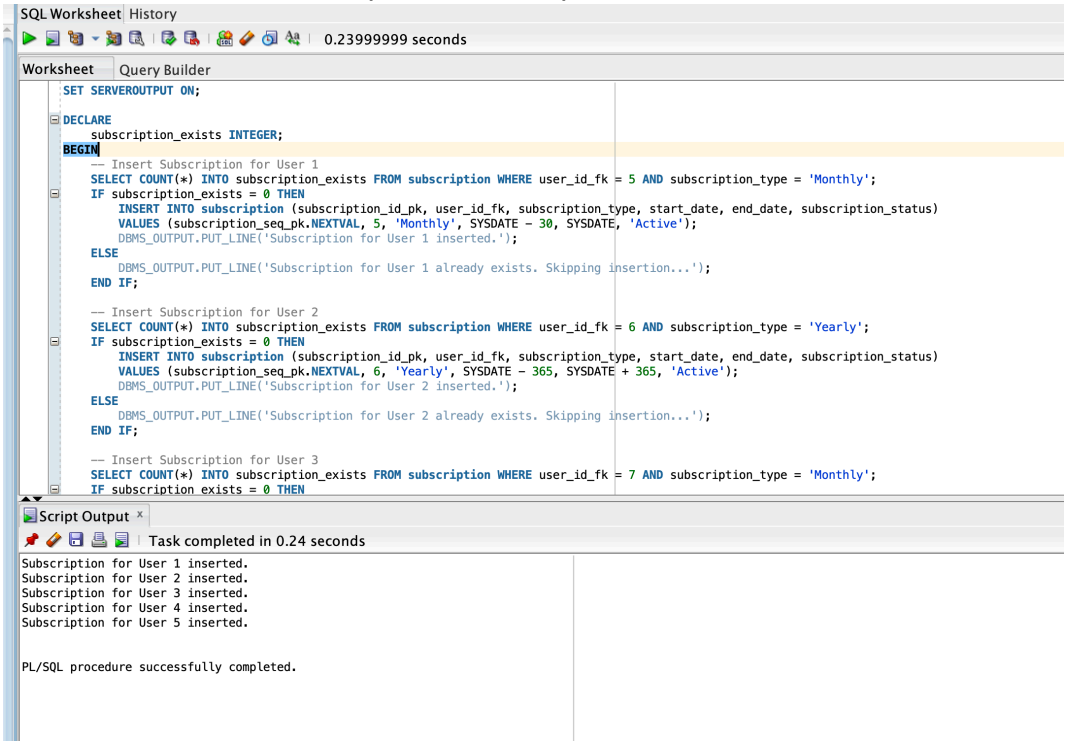
- File Path: DML/ Insert_Group_Bookings_Table.sql



- File Path: DML/ Insert_Rides_Table.sql



File Path: DML/ Insert_Subscription_Table.sql

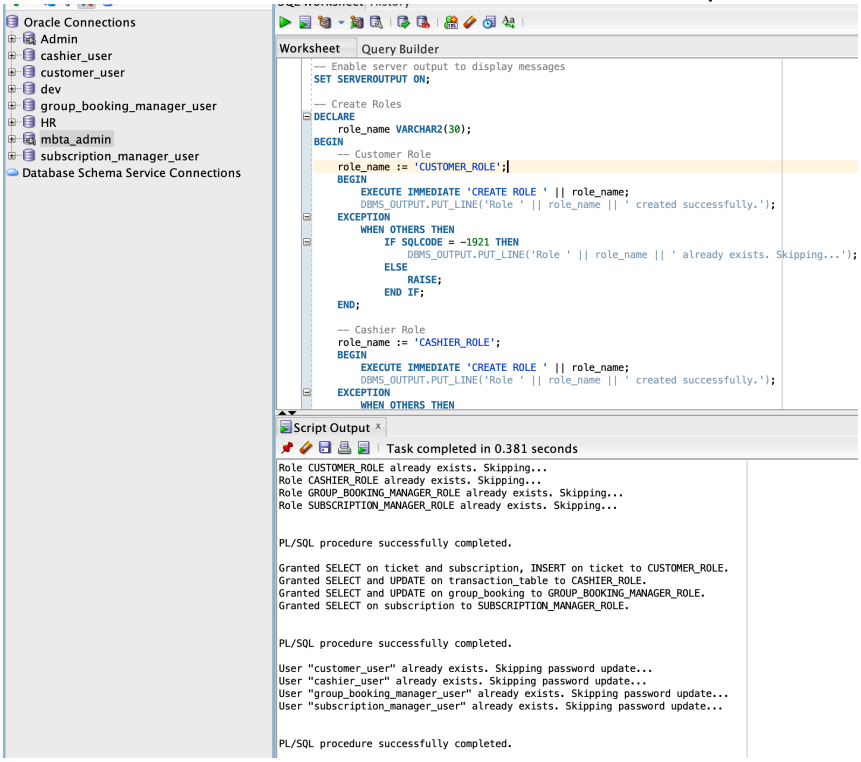


```
SET SERVEROUTPUT ON;

DECLARE
    subscription_exists INTEGER;
BEGIN
    -- Insert Subscription for User 1
    SELECT COUNT(*) INTO subscription_exists FROM subscription WHERE user_id_fk = 5 AND subscription_type = 'Monthly';
    IF subscription_exists = 0 THEN
        INSERT INTO subscription (subscription_id_pk, user_id_fk, subscription_type, start_date, end_date, subscription_status)
        VALUES (subscription_seq_pk.NEXTVAL, 5, 'Monthly', SYSDATE - 30, SYSDATE, 'Active');
        DBMS_OUTPUT.PUT_LINE('Subscription for User 1 inserted.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Subscription for User 1 already exists. Skipping insertion...');
    END IF;

    -- Insert Subscription for User 2
    SELECT COUNT(*) INTO subscription_exists FROM subscription WHERE user_id_fk = 6 AND subscription_type = 'Yearly';
    IF subscription_exists = 0 THEN
        INSERT INTO subscription (subscription_id_pk, user_id_fk, subscription_type, start_date, end_date, subscription_status)
        VALUES (subscription_seq_pk.NEXTVAL, 6, 'Yearly', SYSDATE - 365, SYSDATE + 365, 'Active');
        DBMS_OUTPUT.PUT_LINE('Subscription for User 2 inserted.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Subscription for User 2 already exists. Skipping insertion...');
    END IF;

    -- Insert Subscription for User 3
    SELECT COUNT(*) INTO subscription_exists FROM subscription WHERE user_id_fk = 7 AND subscription_type = 'Monthly';
    IF subscription_exists = 0 THEN
```

```
Subscription for User 1 inserted.
Subscription for User 2 inserted.
Subscription for User 3 inserted.
Subscription for User 4 inserted.
Subscription for User 5 inserted.


PL/SQL procedure successfully completed.
```

Step 6: Create Users and Roles (login as mbta admin)

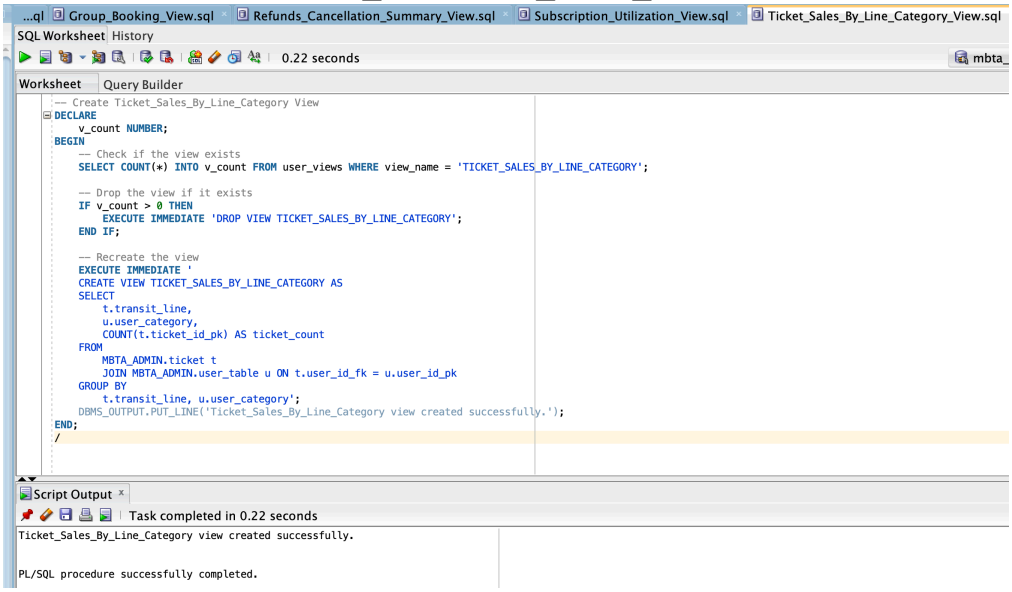
File Path: Role/ User_Creation_and_Grant_Roles.sql



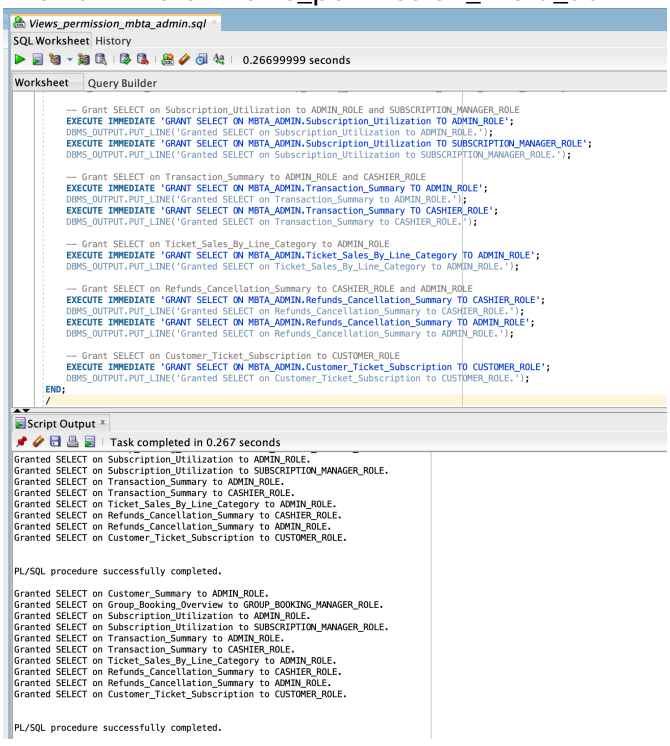
```
-- Enable server output to display messages
SET SERVEROUTPUT ON;

-- Create Roles
DECLARE
    role_name VARCHAR2(30);
BEGIN
    -- Customer Role
    role_name := 'CUSTOMER_ROLE';
    BEGIN
        EXECUTE IMMEDIATE 'CREATE ROLE ' || role_name;
        DBMS_OUTPUT.PUT_LINE('Role ' || role_name || ' created successfully.');
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE = -1921 THEN
                DBMS_OUTPUT.PUT_LINE('Role ' || role_name || ' already exists. Skipping...');
            ELSE
                RAISE;
            END IF;
    END;

    -- Cashier Role
    role_name := 'CASHIER_ROLE';
    BEGIN
        EXECUTE IMMEDIATE 'CREATE ROLE ' || role_name;
        DBMS_OUTPUT.PUT_LINE('Role ' || role_name || ' created successfully.');
    EXCEPTION
        WHEN OTHERS THEN
```

```
Role CUSTOMER_ROLE already exists. Skipping...
Role CASHIER_ROLE already exists. Skipping...
Role GROUP_BOOKING_MANAGER_ROLE already exists. Skipping...
Role SUBSCRIPTION_MANAGER_ROLE already exists. Skipping...


PL/SQL procedure successfully completed.

Granted SELECT on ticket and subscription, INSERT on ticket to CUSTOMER_ROLE.
Granted SELECT and UPDATE on transaction_table to CASHIER_ROLE.
Granted SELECT and UPDATE on group_booking to GROUP_BOOKING_MANAGER_ROLE.
Granted SELECT on subscription to SUBSCRIPTION_MANAGER_ROLE.


PL/SQL procedure successfully completed.

User "customer_user" already exists. Skipping password update...
User "cashier_user" already exists. Skipping password update...
User "group_booking_manager_user" already exists. Skipping password update...
User "subscription_manager_user" already exists. Skipping password update...


PL/SQL procedure successfully completed.
```

Step 7: Create Views (Login as Mbta admin) (exec all views as admin)
File Path : `Views/Views_Creation_mbta_admin/`

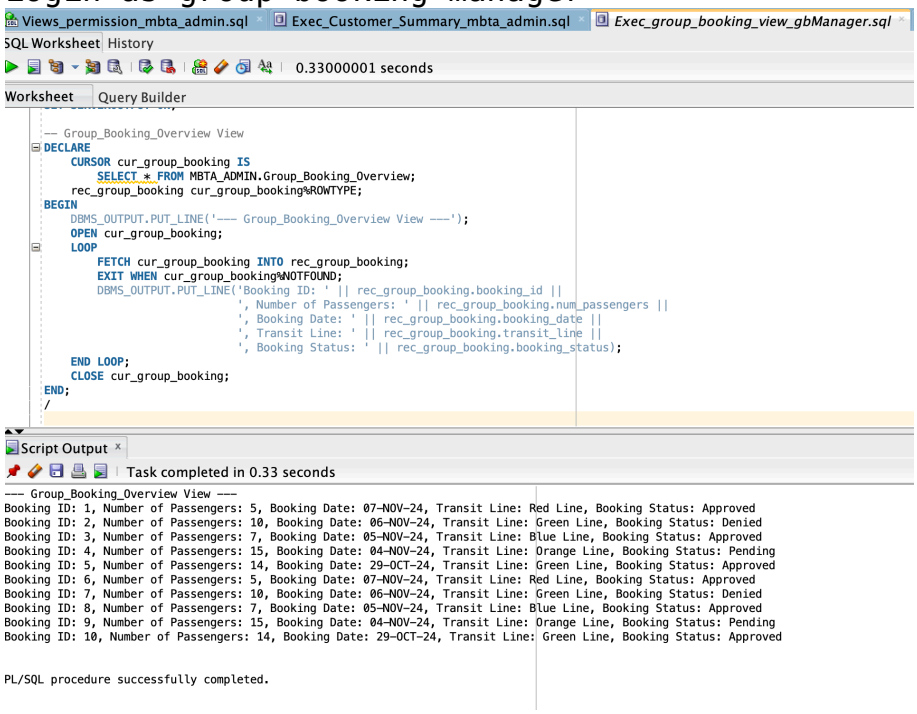


Step 8: Execute VIEWS related permissions (LOGIN AS MBTA ADMIN)
File Path: Role/ Views_permission_mbta_admin.sql

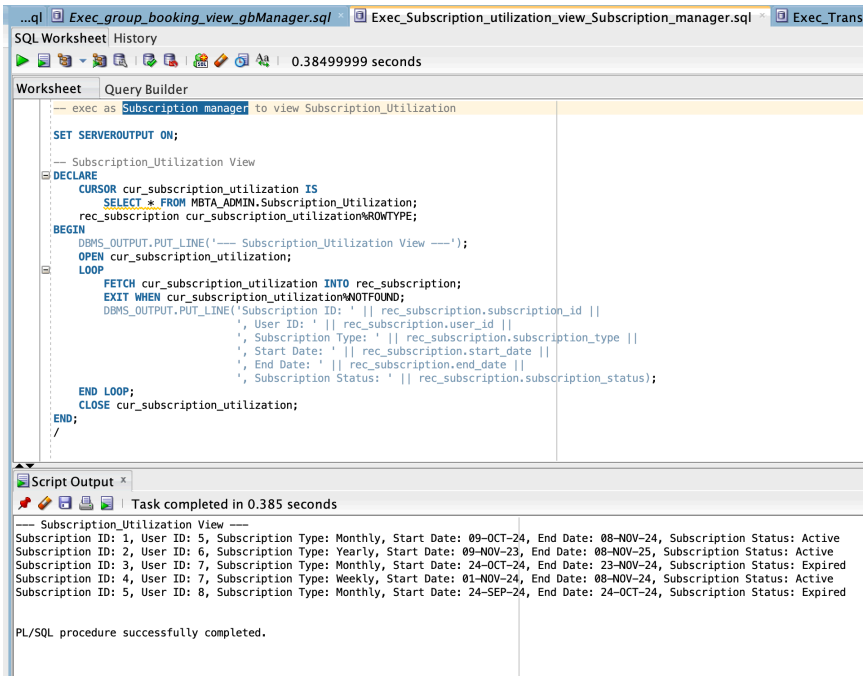


Step 9: **Exec Views**

File Path : `Views/Exec_Views/ Exec_group_booking_view_gbManager`
`Login as group booking manager`



File Path: `Views/Exec_Views/`
`Exec_Subscription_utilization_view_Subscription_manager.sql`



File Path: `Views/Exec_Views/Exec_Transaction_Summary_view_cashier_user.sql`