

```

-- =====
-- TCS ILP PROJECT: Online Grocery Ordering System
-- COMPLETE SCRIPT (US_SQL_001 to US_SQL_005)
-- =====

-- 1. CLEANUP (Drop tables if they exist to avoid errors)
-- Dropping in reverse order of dependencies
BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE Admin';
    EXECUTE IMMEDIATE 'DROP TABLE "Transaction"';
    EXECUTE IMMEDIATE 'DROP TABLE Product';
    EXECUTE IMMEDIATE 'DROP TABLE Login';
    EXECUTE IMMEDIATE 'DROP TABLE Registration';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

-- =====
-- MODULE: US_SQL_002 (Registration Table)
-- =====
CREATE TABLE Registration (
    "Customer ID"   NUMBER PRIMARY KEY NOT NULL,
    "Customer Name" VARCHAR2(100) NOT NULL,
    "Email"         VARCHAR2(100) NOT NULL,
    "Password"      VARCHAR2(50) NOT NULL,
    "Address"       VARCHAR2(200) NOT NULL,
    "Contact Number" NUMBER NOT NULL
);

-- Sample Data for Registration
INSERT INTO Registration VALUES (101, 'Sam', 'sam@gmail.com', 'Pass123', 'RZ/12-4,
Mahavir Enclave, New Delhi, India', 9876543210);
INSERT INTO Registration VALUES (102, 'John', 'john@yahoo.com', 'Pass456', '123 Apple St,
New York, US', 8877665544);
INSERT INTO Registration VALUES (103, 'Priya', 'priya@tcs.com', 'Pass789', 'Tech Park,
Bangalore, India', 9123456780);
INSERT INTO Registration VALUES (104, 'Mike', 'mike@usmail.com', 'Pass000', '456 Orange
Ave, California, US', 7766554433);

-- =====
-- MODULE: US_SQL_001 (Login Table)
-- =====
CREATE TABLE Login (

```

```

"Customer ID"    NUMBER NOT NULL,
"Last Login"     TIMESTAMP NOT NULL,
"Last Logout"    TIMESTAMP NOT NULL,
"updated password" VARCHAR2(50) NOT NULL,
"old password"   VARCHAR2(50) NOT NULL,
"is_now_logged_in" VARCHAR2(1) NOT NULL,
CONSTRAINT fk_login_cust FOREIGN KEY ("Customer ID") REFERENCES
Registration("Customer ID")
);

```

-- Sample Data for Login

```

INSERT INTO Login VALUES (101, SYSTIMESTAMP, SYSTIMESTAMP - 1, 'Pass123',
'OldPass1', 'Y');
INSERT INTO Login VALUES (102, SYSTIMESTAMP, SYSTIMESTAMP - 1, 'Pass456',
'OldPass2', 'Y');
INSERT INTO Login VALUES (103, SYSTIMESTAMP, SYSTIMESTAMP - 1, 'Pass789',
'OldPass3', 'Y');
INSERT INTO Login VALUES (104, SYSTIMESTAMP, SYSTIMESTAMP - 1, 'Pass000',
'OldPass4', 'N');

```

```

-- =====
-- MODULE: US_SQL_003 (Product Table)
-- =====
CREATE TABLE Product (
  "Product ID"    NUMBER PRIMARY KEY NOT NULL,
  "Product Name"  VARCHAR2(100) NOT NULL,
  "Price"         NUMBER NOT NULL,
  "Quantity"      NUMBER NOT NULL,
  "Reserved"      VARCHAR2(50) NOT NULL,
  "Customer ID"   NUMBER NOT NULL,
  CONSTRAINT fk_prod_cust FOREIGN KEY ("Customer ID") REFERENCES
Registration("Customer ID")
);

```

-- Sample Data for Product

```

INSERT INTO Product VALUES (5001, 'Milk', 60, 2, 'Reserved', 101);
INSERT INTO Product VALUES (5002, 'Bread', 40, 1, 'Reserved', 101);
INSERT INTO Product VALUES (5003, 'Butter', 55, 5, 'Reserved', 102);
INSERT INTO Product VALUES (5004, 'Rice', 100, 3, 'Reserved', 103);
INSERT INTO Product VALUES (5005, 'Wheat', 200, 1, 'Reserved', 104);

```

```

-- =====
-- MODULE: US_SQL_004 (Transaction Table)
-- =====

```

```

CREATE TABLE "Transaction" (
    "Transaction ID" NUMBER PRIMARY KEY NOT NULL,
    "Customer ID"  NUMBER NOT NULL,
    "Product ID"   NUMBER NOT NULL,
    "Total Amount" NUMBER NOT NULL,
    "no_of_items"  NUMBER NOT NULL,
    CONSTRAINT fk_trans_cust FOREIGN KEY ("Customer ID") REFERENCES
Registration("Customer ID"),
    CONSTRAINT fk_trans_prod FOREIGN KEY ("Product ID") REFERENCES Product("Product
ID")
);

```

-- Sample Data for Transaction

```

INSERT INTO "Transaction" VALUES (9001, 101, 5001, 5000, 5);
INSERT INTO "Transaction" VALUES (9002, 102, 5003, 3000, 3);
INSERT INTO "Transaction" VALUES (9003, 103, 5004, 7000, 7); -- Highest
INSERT INTO "Transaction" VALUES (9004, 101, 5002, 6000, 4); -- Second Highest
INSERT INTO "Transaction" VALUES (9005, 104, 5005, 2000, 1);

```

```

-- =====
-- MODULE: US_SQL_005 (Admin Table)
-- =====

```

```

CREATE TABLE Admin (
    "Admin ID"      NUMBER PRIMARY KEY NOT NULL,
    "Admin Password" VARCHAR2(50) NOT NULL,
    "Customer ID"   NUMBER NOT NULL,
    "Product ID"    NUMBER NOT NULL,
    "Transaction ID" NUMBER NOT NULL,
    CONSTRAINT fk_adm_cust FOREIGN KEY ("Customer ID") REFERENCES
Registration("Customer ID"),
    CONSTRAINT fk_adm_prod FOREIGN KEY ("Product ID") REFERENCES Product("Product
ID"),
    CONSTRAINT fk_adm_trans FOREIGN KEY ("Transaction ID") REFERENCES
"Transaction"("Transaction ID")
);

```

-- Sample Data for Admin

```

INSERT INTO Admin VALUES (1, 'AdminPass1', 101, 5001, 9001);
INSERT INTO Admin VALUES (2, 'AdminPass2', 103, 5004, 9003);

```

COMMIT;

```

-- =====
-- QUERIES / REPORTS (Solution to User Stories)

```

```

-- =====

-- -----
-- US_SQL_001: Report of First 50% logged in IDs
-- -----
SELECT "Customer ID" FROM Login
WHERE "is_now_logged_in" = 'Y'
AND ROWNUM <= (SELECT COUNT(*) / 2 FROM Login WHERE "is_now_logged_in" = 'Y');

-- -----
-- US_SQL_002: Update Customer Name based on Address
-- -----
UPDATE Registration
SET "Customer Name" = CASE
    WHEN "Address" LIKE '%US%' THEN 'US_' || "Customer Name"
    WHEN "Address" LIKE '%India%' THEN 'IN_' || "Customer Name"
    ELSE "Customer Name"
END;

-- Check the update
SELECT * FROM Registration;

-- -----
-- US_SQL_003: Distinct Products & Customer Mapping
-- -----
-- Report 1: Distinct Products
SELECT DISTINCT "Product Name" FROM Product;

-- Report 2: Product to Customer Mapping
SELECT r."Customer Name", p."Product Name", p."Reserved"
FROM Registration r
JOIN Product p ON r."Customer ID" = p."Customer ID";

-- -----
-- US_SQL_004: Display Second Highest Transaction
-- -----
-- Logic: Sort descending, pick the one that is 2nd in the list.
SELECT * FROM (
    SELECT a.*, ROWNUM rnum FROM (
        SELECT * FROM "Transaction" ORDER BY "Total Amount" DESC
    ) a WHERE ROWNUM <= 2
) WHERE rnum = 2;

-- -----

```

-- US\_SQL\_005: Display Customer Details, Trans ID, Product Name

-----

-- Join Registration, Transaction, and Product tables

SELECT

    r."Customer ID",

    r."Customer Name",

    r."Email",

    t."Transaction ID",

    p."Product Name"

FROM Registration r

JOIN "Transaction" t ON r."Customer ID" = t."Customer ID"

JOIN Product p ON t."Product ID" = p."Product ID";