```
-- ============================================================
-- TCS ILP PROJECT
-- Online Grocery Ordering System
-- FINAL POLISHED SQL (US_SQL_001 → US_SQL_005)
-- ============================================================

-- =======================
-- CLEANUP
-- =======================
BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE admin_details';
  EXECUTE IMMEDIATE 'DROP TABLE transactions';
  EXECUTE IMMEDIATE 'DROP TABLE product';
  EXECUTE IMMEDIATE 'DROP TABLE login_details';
  EXECUTE IMMEDIATE 'DROP TABLE registration';
EXCEPTION
  WHEN OTHERS THEN NULL;
END;
/

-- ============================================================
-- US_SQL_002 : REGISTRATION TABLE
-- ============================================================
CREATE TABLE registration (
  customer_id      NUMBER PRIMARY KEY,
  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
);

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);

-- ============================================================
-- US_SQL_001 : LOGIN TABLE
```

```sql
-- ============================================================
CREATE TABLE login_details (
   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)  CHECK (is_now_logged_in IN ('Y','N')),
   CONSTRAINT fk_login_cust
      FOREIGN KEY (customer_id) REFERENCES registration(customer_id)
);

INSERT INTO login_details VALUES
(101,SYSTIMESTAMP,SYSTIMESTAMP-INTERVAL '1' DAY,'Pass123','Old1','Y');
INSERT INTO login_details VALUES
(102,SYSTIMESTAMP,SYSTIMESTAMP-INTERVAL '1' DAY,'Pass456','Old2','Y');
INSERT INTO login_details VALUES
(103,SYSTIMESTAMP,SYSTIMESTAMP-INTERVAL '1' DAY,'Pass789','Old3','Y');
INSERT INTO login_details VALUES
(104,SYSTIMESTAMP,SYSTIMESTAMP-INTERVAL '1' DAY,'Pass000','Old4','N');


-- ============================================================
-- US_SQL_003 : PRODUCT TABLE
-- ============================================================
CREATE TABLE product (
   product_id   NUMBER PRIMARY KEY,
   product_name  VARCHAR2(100) NOT NULL,
   price       NUMBER NOT NULL,
   quantity     NUMBER NOT NULL,
   reserved      VARCHAR2(1) CHECK (reserved IN ('Y','N')),
   customer_id   NUMBER NOT NULL,
   CONSTRAINT fk_prod_cust
      FOREIGN KEY (customer_id) REFERENCES registration(customer_id)
);

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


-- ============================================================
-- US_SQL_004 : TRANSACTION TABLE
-- ============================================================
```

```
CREATE TABLE transactions (
   transaction_id NUMBER PRIMARY KEY,
   customer_id   NUMBER NOT NULL,
   product_id    NUMBER NOT NULL,
   total_amount  NUMBER NOT NULL,
   no_of_items   NUMBER NOT NULL,
   CONSTRAINT fk_trx_cust FOREIGN KEY (customer_id)
      REFERENCES registration(customer_id),
   CONSTRAINT fk_trx_prod FOREIGN KEY (product_id)
      REFERENCES product(product_id)
);

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

-- ===========================================================
-- US_SQL_005 : ADMIN TABLE
-- ===========================================================
CREATE TABLE admin_details (
   admin_id      NUMBER PRIMARY KEY,
   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_trx FOREIGN KEY (transaction_id)
      REFERENCES transactions(transaction_id)
);

INSERT INTO admin_details VALUES (1,'Admin@123',101,5001,9001);
INSERT INTO admin_details VALUES (2,'Admin@456',103,5004,9003);

COMMIT;

-- ===========================================================
-- REPORTS / QUERIES
-- ===========================================================
```

```sql
-- US_SQL_001 : First 50% Logged-in Customers
SELECT customer_id
FROM login_details
WHERE is_now_logged_in = 'Y'
AND ROWNUM <= (
    SELECT CEIL(COUNT(*) * 0.5)
    FROM login_details
    WHERE is_now_logged_in = 'Y'
);

-- US_SQL_002 : Update Customer Name Based on Country
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;

-- US_SQL_003 : Distinct Products
SELECT DISTINCT product_name FROM product;

-- US_SQL_003 : Customer–Product Mapping
SELECT r.customer_name, p.product_name
FROM registration r
JOIN product p ON r.customer_id = p.customer_id;

-- US_SQL_004 : Second Highest Transaction
SELECT *
FROM (
    SELECT t.*, ROWNUM rn
    FROM (
        SELECT * FROM transactions ORDER BY total_amount DESC
    ) t
    WHERE ROWNUM <= 2
)
WHERE rn = 2;

-- US_SQL_005 : Final Admin Report
SELECT r.customer_id,
       r.customer_name,
       r.email,
       t.transaction_id,
       p.product_name
```

```
FROM registration r
JOIN transactions t ON r.customer_id = t.customer_id
JOIN product p ON t.product_id = p.product_id;
```