

Customer(customer_ID(pk), first_name, last_name, email)
Cart(cart_id(pk), customer_id(fk))
Cart_Item(cart_id(pk, fk), shoe_id(pk, fk), quantity)
Employees(employee_id(pk), first_name, email, type, salary)
Transaction(transaction_id(pk), cart_id(fk), payment_method, date, obtain_type)
Inventory_Item(shoe_id(pk, fk), storage_location(pk), quantity)
Shoes(shoe_id(pk), model_id(fk), color_id(fk), size_id(fk), price)
Models(model_id(pk), brand, name)
Colors(color_id(pk), name)
Sizes(size_id(pk), name)

```
CREATE TABLE Customer (  
    customer_ID NUMBER PRIMARY KEY,  
    first_name VARCHAR2(50),  
    last_name VARCHAR2(50),  
    email VARCHAR2(100)  
);
```

```
CREATE TABLE Cart (  
    cart_id NUMBER PRIMARY KEY,  
    customer_id NUMBER,  
    CONSTRAINT fk_customer  
        FOREIGN KEY (customer_id)  
        REFERENCES Customer(customer_ID)  
);
```

```
CREATE TABLE Cart_Item (  
    cart_id NUMBER,  
    shoe_id NUMBER,  
    quantity NUMBER,  
    PRIMARY KEY (cart_id, shoe_id),  
    CONSTRAINT fk_cart  
        FOREIGN KEY (cart_id)  
        REFERENCES Cart(cart_id),  
    CONSTRAINT fk_shoe  
        FOREIGN KEY (shoe_id)  
        REFERENCES Shoes(shoe_id)  
);
```

```
CREATE TABLE Employees (  
    employee_id NUMBER PRIMARY KEY,  
    first_name VARCHAR2(50),  
    email VARCHAR2(100),  
    type VARCHAR2(50)  
);
```

```
CREATE TABLE Purchase (  
    purchase_id NUMBER PRIMARY KEY,  
    transaction_id NUMBER,  
    date DATE,  
    obtain_type VARCHAR2(50)
```

```
purchase_id NUMBER PRIMARY KEY,  
cart_id NUMBER,  
payment_method VARCHAR2(50),  
purchase_date DATE,  
obtain_type VARCHAR2(50),  
CONSTRAINT fk_transaction  
    FOREIGN KEY (cart_id)  
    REFERENCES Cart(cart_id)  
);
```

```
CREATE TABLE Inventory (  
    shoe_id NUMBER,  
    quantity NUMBER,  
    PRIMARY KEY (shoe_id, storage_location),  
    CONSTRAINT fk_inventory  
        FOREIGN KEY (shoe_id)  
        REFERENCES Shoes(shoe_id)  
);
```

```
CREATE TABLE Shoes (  
    shoe_id NUMBER PRIMARY KEY,  
    model_id NUMBER,  
    color_id NUMBER,  
    size_id NUMBER,  
    price NUMBER,  
    CONSTRAINT fk_model  
        FOREIGN KEY (model_id)  
        REFERENCES Models(model_id),  
    CONSTRAINT fk_color  
        FOREIGN KEY (color_id)  
        REFERENCES Colors(color_id),  
    CONSTRAINT fk_size  
        FOREIGN KEY (size_id)  
        REFERENCES Sizes(size_id)  
);
```

```
CREATE TABLE Models (  
    model_id NUMBER PRIMARY KEY,  
    brand VARCHAR2(50),  
    name VARCHAR2(50)  
);
```

```
CREATE TABLE Colors (  
    color_id NUMBER PRIMARY KEY,  
    name VARCHAR2(50)  
);
```

```
CREATE TABLE Sizes (  
    size_id NUMBER PRIMARY KEY,  
    name VARCHAR2(50)
```

```
size_id NUMBER PRIMARY KEY,  
name NUMBER  
);
```

```
CREATE TABLE Monthly_Purchases (Mesyac VARCHAR2(20), Purchase_Count NUMBER,  
Total_Price NUMBER)
```

```
INSERT INTO Customer (customer_ID, first_name, last_name, email)  
SELECT 1, 'Gervonta', 'Davies', 'gervonta@gmail.com' FROM DUAL  
UNION ALL  
SELECT 2, 'Breaking', 'Bad', 'walterwhite@gmail.com' FROM DUAL  
UNION ALL  
SELECT 3, 'Yaslan', 'Ruchanov', 'yaslan@gmail.com' FROM DUAL  
UNION ALL  
SELECT 4, 'Daniil', 'Lyapin', 'vasya@gmail.com' FROM DUAL  
UNION ALL  
SELECT 5, 'Sultan', 'Damir', 'damir@gmail.com' FROM DUAL  
UNION ALL  
SELECT 6, 'Narken', 'Miras', 'miras@gmail.com' FROM DUAL;
```

```
INSERT INTO SIZES (size_ID, name)  
SELECT 1, 37 FROM DUAL  
UNION ALL  
SELECT 2, 38 FROM DUAL  
UNION ALL  
SELECT 3, 39 FROM DUAL  
UNION ALL  
SELECT 4, 39.5 FROM DUAL  
UNION ALL  
SELECT 5, 40 FROM DUAL  
UNION ALL  
SELECT 6, 40.5 FROM DUAL  
UNION ALL  
SELECT 7, 41 FROM DUAL  
UNION ALL  
SELECT 8, 41.5 FROM DUAL  
UNION ALL  
SELECT 9, 42 FROM DUAL;
```

```
INSERT INTO COLORS (color_ID, name)  
SELECT 1, 'White' FROM DUAL  
UNION ALL  
SELECT 2, 'Black' FROM DUAL  
UNION ALL  
SELECT 3, 'Red' FROM DUAL  
UNION ALL  
SELECT 4, 'Yellow' FROM DUAL  
UNION ALL
```

```
SELECT 5, 'Purple' FROM DUAL
UNION ALL
SELECT 6, 'Green' FROM DUAL
UNION ALL
SELECT 7, 'Blue' FROM DUAL
UNION ALL
SELECT 8, 'Orange' FROM DUAL
UNION ALL
SELECT 9, 'Mixed' FROM DUAL;
```

```
INSERT INTO MODELS (model_ID, brand, name)
SELECT 1, 'Nike', 'Air Force 1' FROM DUAL
UNION ALL
SELECT 2, 'Nike', 'Huarache' FROM DUAL
UNION ALL
SELECT 3, 'Nike', 'Air Max 95' FROM DUAL
UNION ALL
SELECT 4, 'Nike', 'Air Max 97' FROM DUAL
UNION ALL
SELECT 5, 'Nike', 'Air Jordan 1' FROM DUAL
UNION ALL
SELECT 6, 'Adidas', 'Ozweego' FROM DUAL
UNION ALL
SELECT 7, 'Nike', 'Air Monarch IV' FROM DUAL
UNION ALL
SELECT 8, 'Barhatniye', 'Tyagi' FROM DUAL
UNION ALL
SELECT 9, 'Adibas', 'Makasin' FROM DUAL
UNION ALL
SELECT 10, 'Nike', 'Air Skepta' FROM DUAL;
```

```
INSERT INTO Shoes (shoe_ID, model_ID, color_ID, size_ID, price)
SELECT 1, 1, 2, 5, 100 FROM DUAL UNION ALL
SELECT 2, 1, 2, 6, 100 FROM DUAL UNION ALL
SELECT 3, 2, 6, 3, 240 FROM DUAL UNION ALL
SELECT 4, 2, 7, 4, 240 FROM DUAL UNION ALL
SELECT 5, 3, 3, 5, 170 FROM DUAL UNION ALL
SELECT 6, 3, 2, 9, 180 FROM DUAL UNION ALL
SELECT 7, 4, 9, 2, 120 FROM DUAL UNION ALL
SELECT 8, 5, 1, 2, 130 FROM DUAL UNION ALL
SELECT 9, 6, 4, 5, 140 FROM DUAL UNION ALL
SELECT 10, 6, 2, 6, 140 FROM DUAL UNION
SELECT 11, 7, 5, 8, 200 FROM DUAL UNION ALL
SELECT 12, 7, 5, 9, 200 FROM DUAL UNION ALL
SELECT 13, 8, 8, 4, 300 FROM DUAL UNION ALL
```

```
SELECT 14, 8, 7, 5, 300 FROM DUAL UNION ALL
SELECT 15, 9, 3, 7, 60 FROM DUAL UNION ALL
SELECT 16, 10, 7, 6, 160 FROM DUAL
INSERT INTO Inventory_item (shoe_ID, quantity)
SELECT 1, 7 FROM DUAL UNION ALL
SELECT 2, 10 FROM DUAL UNION ALL
SELECT 3, 5 FROM DUAL UNION ALL
SELECT 4, 4 FROM DUAL UNION ALL
SELECT 5, 7 FROM DUAL UNION ALL
SELECT 6, 16 FROM DUAL UNION ALL
SELECT 7, 12 FROM DUAL UNION ALL
SELECT 8, 15 FROM DUAL UNION ALL
SELECT 9, 8 FROM DUAL UNION ALL
SELECT 10, 9 FROM DUAL
```

```
INSERT INTO Employees(employee_id, first_name, email, type, salary)
SELECT 1, 'Aldiyar', 'alldiyar@gmail.com', 'admin', 4000 FROM DUAL UNION ALL
SELECT 2, 'Daniil', 'danya@gmail.com', 'admin', 4000 FROM DUAL UNION ALL
SELECT 3, 'Ilyar', 'ilyar@gmail.com', 'courier', 500 FROM DUAL UNION ALL
SELECT 4, 'Talanbek', 'talantbek@gmail.com', 'courier', 500 FROM DUAL UNION ALL
SELECT 5, 'Aidar', 'aidar@gmail.com', 'courier', 500 FROM DUAL UNION ALL
SELECT 6, 'Diyas', 'diyas@gmail.com', 'driver', 800 FROM DUAL UNION ALL
SELECT 7, 'Akzhan', 'akzhan@gmail.com', 'waitress', 100 FROM DUAL UNION ALL
SELECT 8, 'Madi', 'madi@gmail.com', 'security', 800 FROM DUAL UNION ALL
SELECT 9, 'Miras', 'yahagi@gmail.com', 'SMM', 2000 FROM DUAL UNION ALL
SELECT 10, 'Aman', 'aman@gmail.com', 'security', 800 FROM DUAL;
```

```
INSERT INTO CART(cart_id, customer_id)
SELECT 1, 1 FROM DUAL UNION ALL
SELECT 2, 2 FROM DUAL UNION ALL
SELECT 3, 3 FROM DUAL UNION ALL
SELECT 4, 4 FROM DUAL UNION ALL
SELECT 5, 5 FROM DUAL UNION ALL
SELECT 6, 6 FROM DUAL
```

```
;
```

```
INSERT INTO PURCHASE(purchase_id, cart_id, payment_method, purchase_date,
obtain_type)
Select 1, 1, 'kaspi', TO_DATE('2023-02-20', 'yyyy-mm-dd'), 'delievery' FROM DUAL UNION
ALL
```

```

Select 2, 2, 'halyk', TO_DATE('2023-02-25', 'yyyy-mm-dd'), 'pickup' FROM DUAL UNION
ALL
Select 3, 3, 'kaspi', TO_DATE('2023-03-12', 'yyyy-mm-dd'), 'pickup' FROM DUAL UNION
ALL
Select 4, 4, 'cash', TO_DATE('2023-03-24', 'yyyy-mm-dd'), 'delievery' FROM DUAL UNION
ALL
Select 5, 5, 'cash', TO_DATE('2023-03-26', 'yyyy-mm-dd'), 'pickup' FROM DUAL UNION ALL
Select 6, 6, 'halyk', TO_DATE('2023-03-26', 'yyyy-mm-dd'), 'delievery' FROM DUAL;

```

```

INSERT INTO cart_item (cart_id, shoe_id, quantity)
SELECT 1, 4, 1 FROM dual
UNION ALL
SELECT 1, 3, 1 FROM dual
UNION ALL
SELECT 2, 6, 2 FROM dual
UNION ALL
SELECT 3, 1, 1 FROM dual
UNION ALL
SELECT 4, 8, 1 FROM dual
UNION ALL
SELECT 4, 6, 1 FROM dual
UNION ALL
SELECT 5, 2, 1 FROM dual
UNION ALL
SELECT 6, 7, 1 FROM dual
UNION ALL
SELECT 6, 4, 1 FROM dual
UNION ALL
SELECT 6, 3, 1 FROM dual;

```

//Function

```

CREATE OR REPLACE FUNCTION count_records(table_name IN VARCHAR2)
RETURN NUMBER
IS
    total_count NUMBER;
BEGIN
    EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM ' || table_name INTO total_count;
    RETURN total_count;
END;

```

```

Select count_records('shoes') FROM DUAL

```

// User defined exception

```

CREATE OR REPLACE TRIGGER email_check

```

```

BEFORE INSERT ON customer
FOR EACH ROW
DECLARE
    email_error EXCEPTION;
BEGIN
    IF LENGTH(:NEW.email) < 5 OR INSTR(:NEW.email, '@') = 0 THEN
        RAISE email_error;
    END IF;
EXCEPTION
    WHEN email_error THEN
        RAISE_APPLICATION_ERROR(-20001, 'Email must be at least 5 characters and contain
"@@" in it.');
```

```

END;

/ SQL%ROWCOUNT
```

```

CREATE OR REPLACE PROCEDURE count_monthly_purchases AS
    rows_inserted NUMBER;
BEGIN
```

```

    FOR m IN 1..12 LOOP
```

```

        DECLARE
            start_date DATE := TO_DATE('01-' || m || '-' || EXTRACT(YEAR FROM SYSDATE),
'DD-MM-YYYY');
            end_date DATE := LAST_DAY(start_date);
        BEGIN
```

```

            DECLARE
                cur SYS_REFCURSOR;
                count_val NUMBER;
                price_val NUMBER;
            BEGIN
                OPEN cur FOR
                    'SELECT COUNT(*), SUM(s.price) FROM Purchase p, Cart_Item ci, Shoes s ' ||
                    'WHERE p.cart_id = ci.cart_id AND ci.shoe_id = s.shoe_id AND p.purchase_date >=
: start_date AND p.purchase_date <= :end_date'
                    USING start_date, end_date;
```

```

            FETCH cur INTO count_val, price_val;
            CLOSE cur;
```

```

            EXECUTE IMMEDIATE 'INSERT INTO Monthly_Purchases (Mesyac, Purchase_Count,
Total_Price) ' ||
                'VALUES (:mesyac, :count_val, :price_val)' USING TO_CHAR(start_date,
'Month YYYY'), count_val, price_val;
```

```

        rows_inserted := SQL%ROWCOUNT;
        DBMS_OUTPUT.PUT_LINE(rows_inserted || ' row(s) inserted into Monthly_Purchases
table for mesyac ' || TO_CHAR(start_date, 'Month YYYY') || '.');
    END;
    END;
END LOOP;

```

```

    DBMS_OUTPUT.PUT_LINE('Monthly purchase counts and totals have been calculated and
stored in the Monthly_Purchases table. ');
END;
// group by procedure

```

```

CREATE OR REPLACE PROCEDURE group_by_info
IS
BEGIN
    FOR rec IN (
        SELECT type, COUNT(*) AS num_employees
        FROM employees
        GROUP BY type
    )
    LOOP
        DBMS_OUTPUT.PUT_LINE('Type: ' || rec.type || ', Number of Employees: ' ||
rec.num_employees);
    END LOOP;
END;

```

```

// trigger with count of rows

```

```

CREATE OR REPLACE TRIGGER purchase_count_trigger
BEFORE INSERT ON Purchase
FOR EACH ROW
DECLARE
    cnt NUMBER;
BEGIN
    SELECT COUNT(*) INTO cnt
    FROM Purchase
    WHERE TRUNC(purchase_date) = TRUNC(SYSDATE);

    IF cnt >= 2 AND :NEW.obtain_type = 'delievery' THEN
        DBMS_OUTPUT.PUT_LINE('We cannot deliver today, today we already have ' || cnt || '
orders to deliver. ');
    END IF;
END;

```