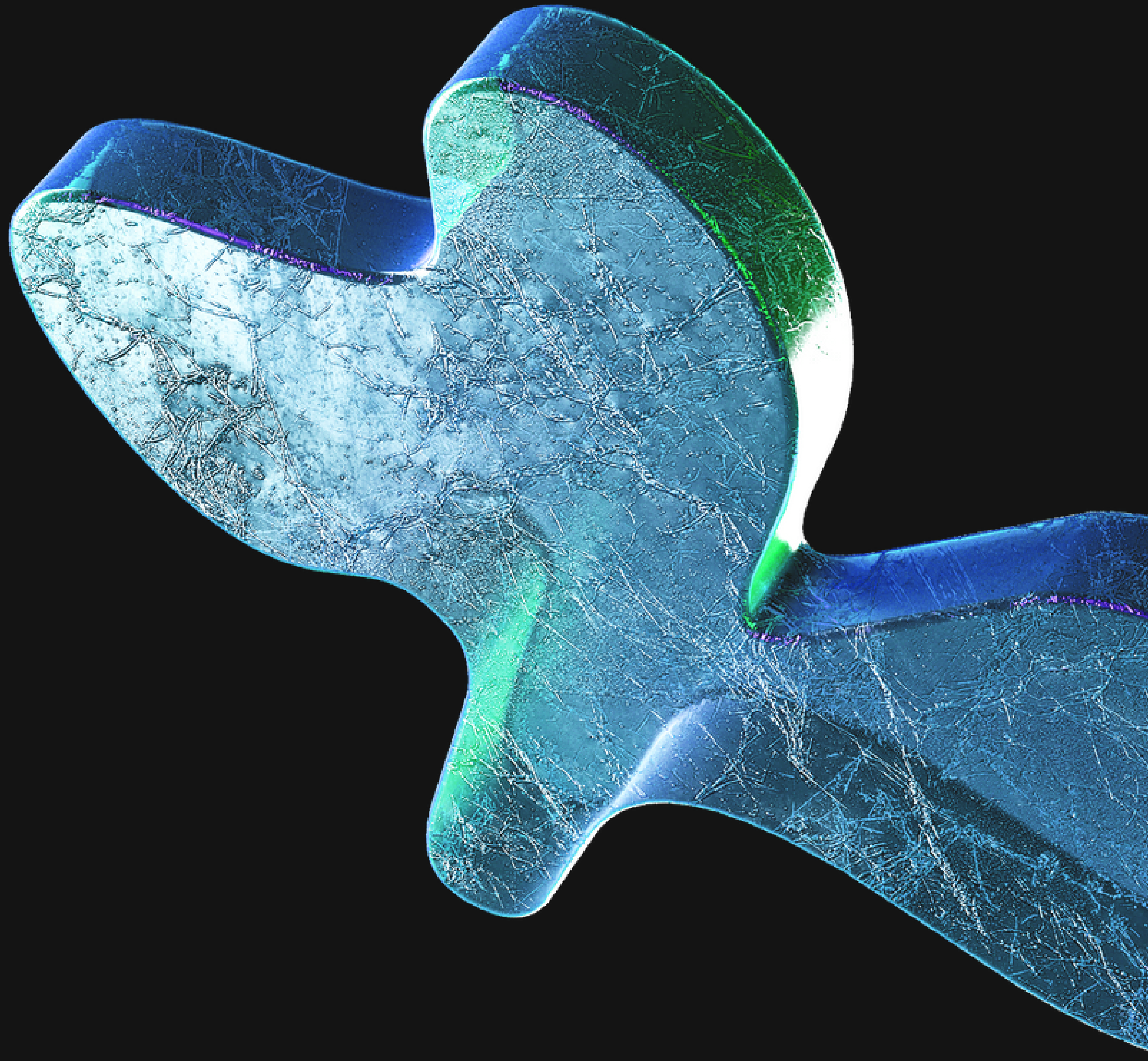
COFFEE SHOP

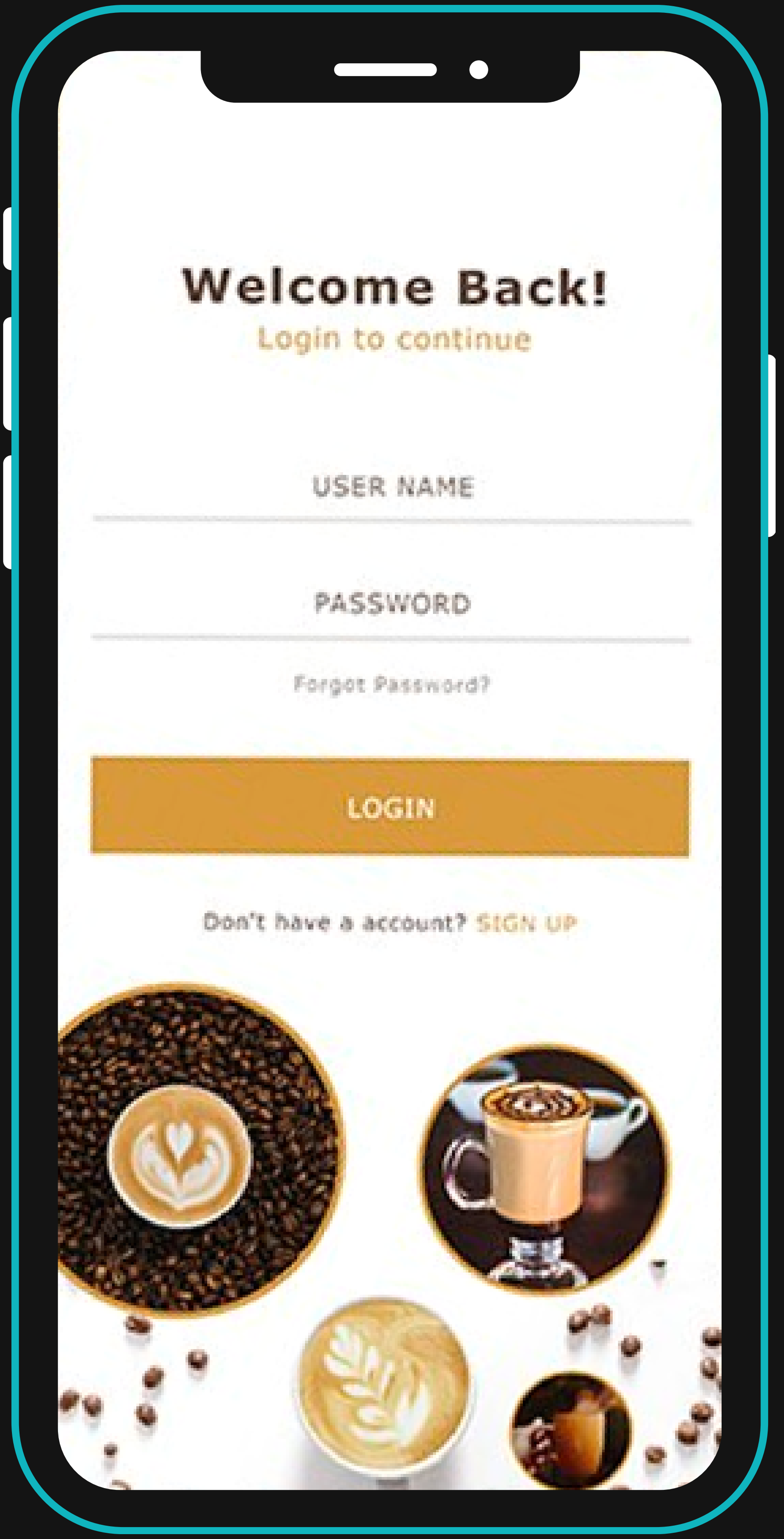


Done by:

1. Amir Amirgali 210103069 (Exception, Transaction, Triggers)
2. Druskuldinova Zhanelya 210103031 (Procedures, Functions, Report)
3. Ospan Abylay 210103075 (Create and insert tables, triggers)
4. Umbetzhan Sholpan 210103023 (Functions, Procedures, Buisiness process)
5. Yerzhankyzy Salima 210103066 (ER Diagram, Report, Cursors, Records)

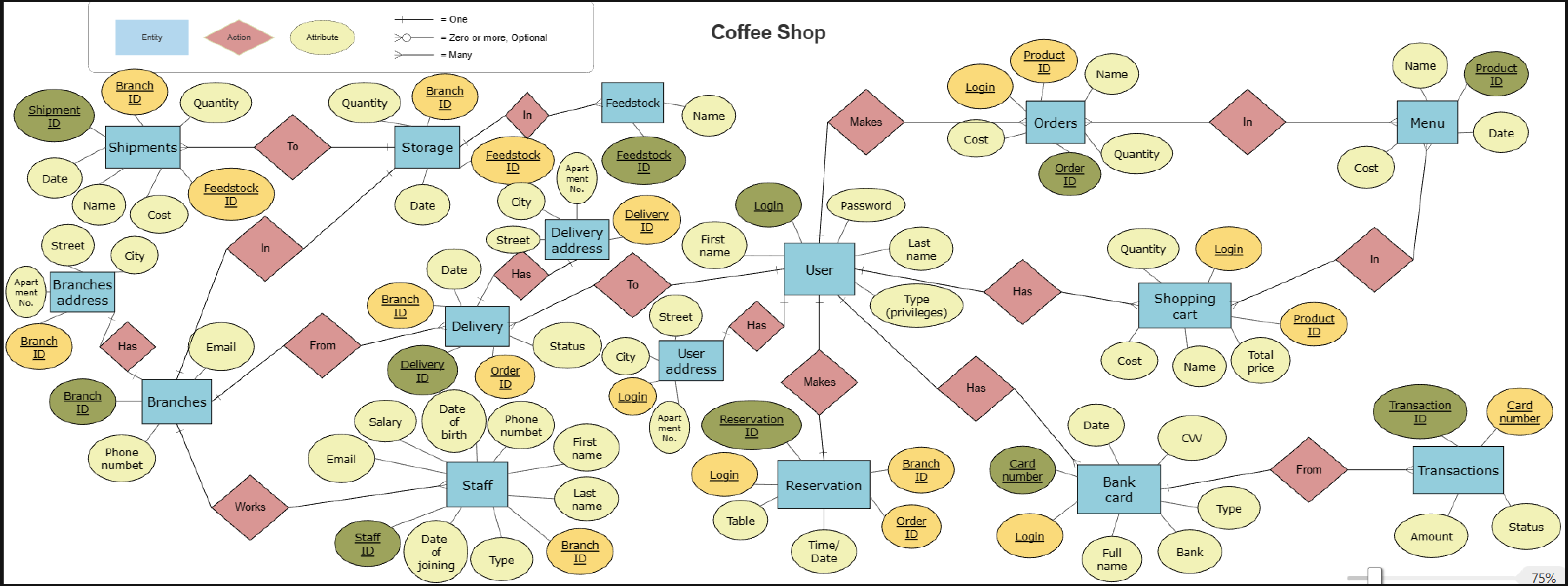
The process of using of our shop:

As a inspiration of our project we chose the “Coffee shop”

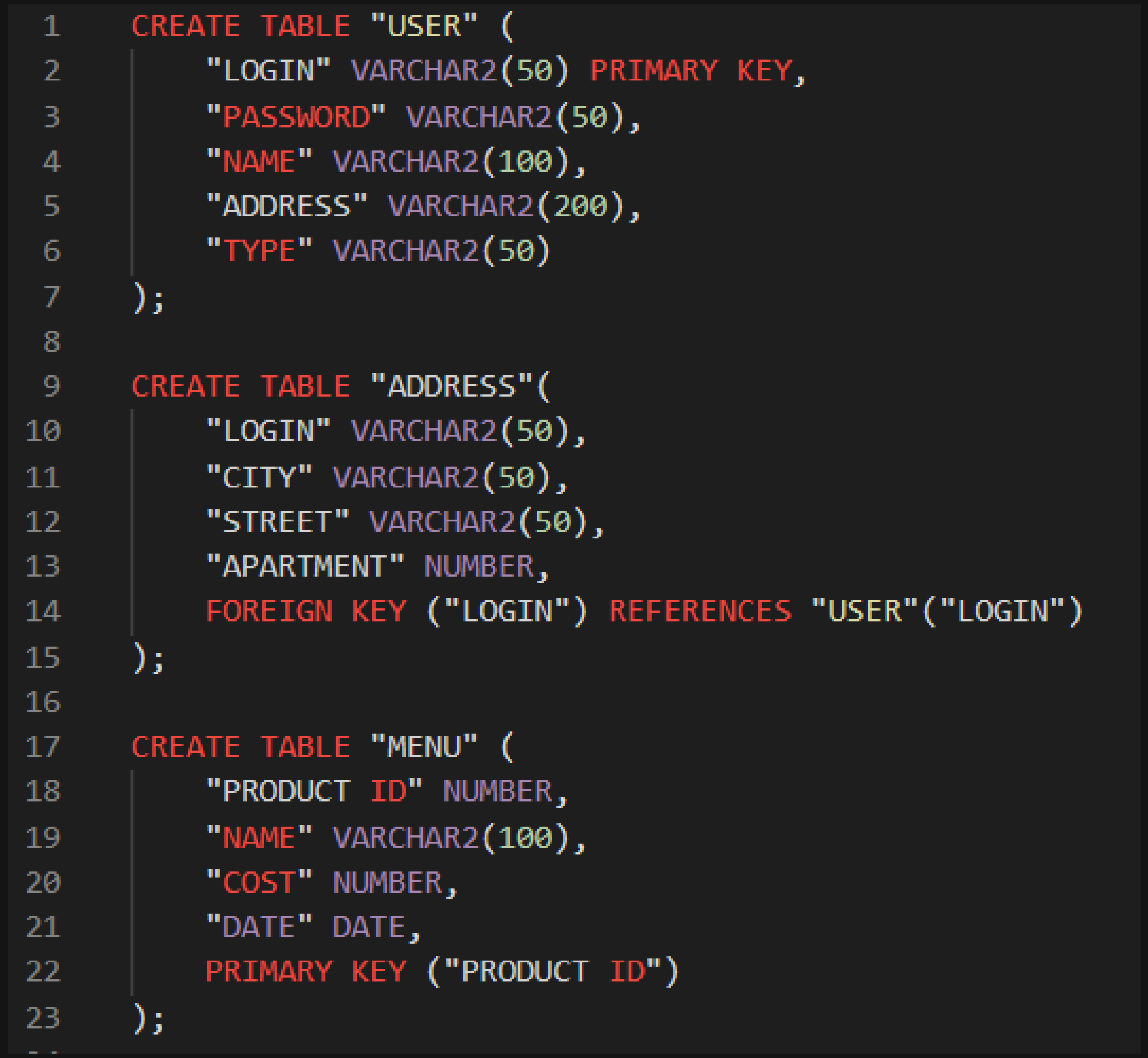


1. Client opens the shop
2. Signing up (fill in the information about themselves and add a bank card)
3. Loging in into their account
4. Scrolls through the menu
5. Client can choose the items and put them into the shopping card
6. After client is able to make order (in process of buying the cost of the order will be withdrawn from the clients' card)
7. Client has 2 choices: Picking up themselves / Delivery from the nearest branch (if client chose 'delivery' after payment, the courier will start transporting the order)
8. Also, client can book table in any branch
9. When 'delivery' status will be succeed, transaction also will be ended, otherwise order will be canceled and money will be returned to the client.

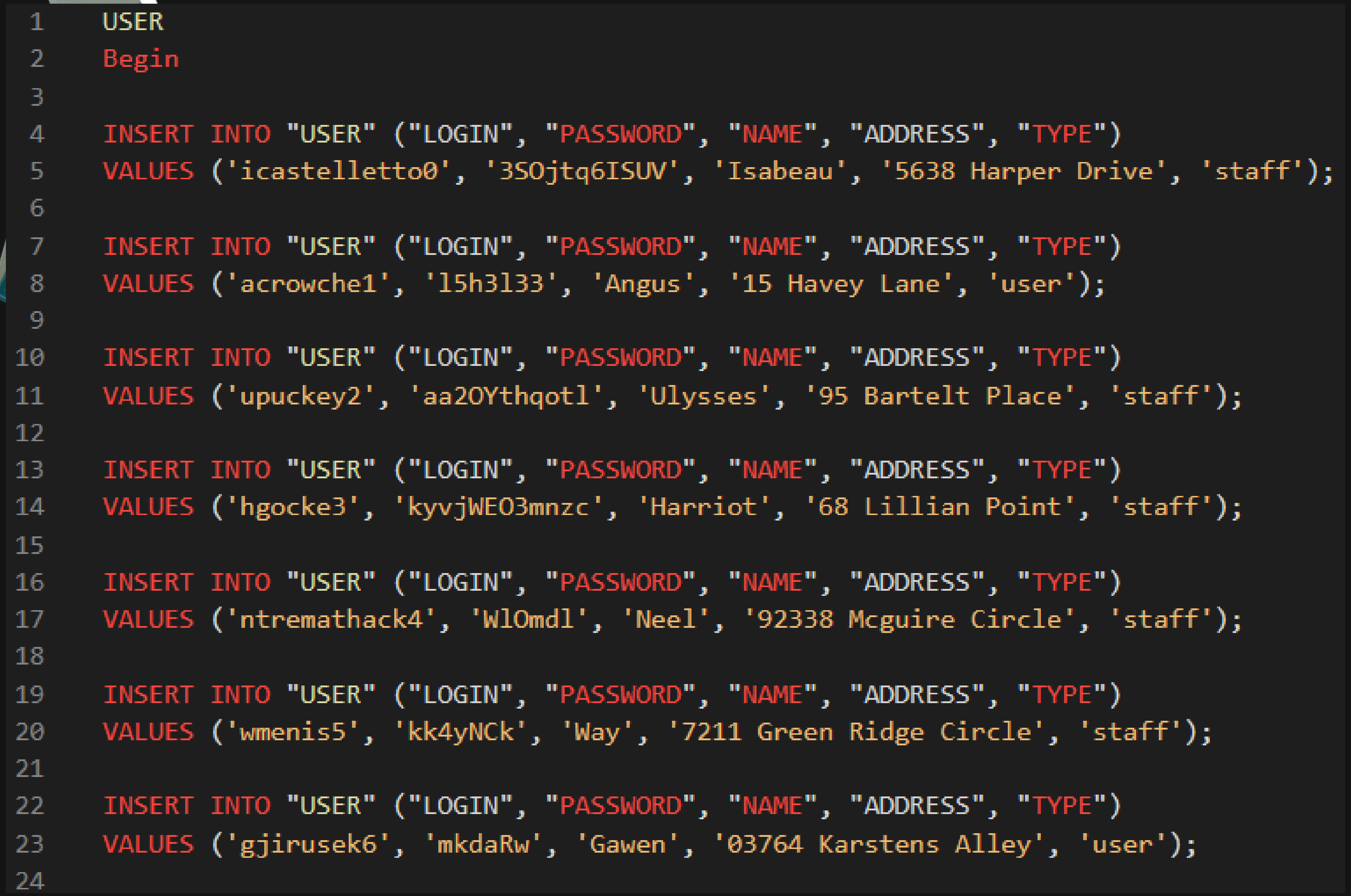
ER Diagram


Entity
Action
Attribute
= One
= Zero or more, Optional
= Many
Coffee Shop
Shipments
Shipment ID
Branch ID
Quantity
Date
Name
Cost
Feedstock ID
To
Storage
Quantity
Branch ID
Date
In
Feedstock
Name
Feedstock ID
Feedstock ID
Apart ment No.
City
Street
Delivery address
Delivery ID
Has
Makes
Orders
Login
Product ID
Name
Cost
Quantity
Order ID
In
Menu
Name
Product ID
Date
Cost
In
User
Login
Password
First name
Last name
Type (privileges)
To
Has
Shopping cart
Quantity
Login
Product ID
Cost
Name
Total price
Street
City
Apart ment No.
Branches address
Branch ID
Has
Branches
Branch ID
Phone number
Email
From
Delivery
Date
Branch ID
Delivery ID
Order ID
Status
Works
Staff
Salary
Date of birth
Phone numbet
First name
Last name
Email
Staff ID
Date of joining
Type
Branch ID
Street
City
User address
Login
Apart ment No.
Has
Makes
Reservation
Reservation ID
Login
Table
Time/ Date
Branch ID
Order ID
Has
Bank card
Date
CW
Card number
Login
Full name
Bank
Type
From
Transactions
Transaction ID
Card number
Amount
Status
75%

Creating and inserting



```sql
CREATE TABLE "USER" (
    "LOGIN" VARCHAR2(50) PRIMARY KEY,
    "PASSWORD" VARCHAR2(50),
    "NAME" VARCHAR2(100),
    "ADDRESS" VARCHAR2(200),
    "TYPE" VARCHAR2(50)
);

CREATE TABLE "ADDRESS"(
    "LOGIN" VARCHAR2(50),
    "CITY" VARCHAR2(50),
    "STREET" VARCHAR2(50),
    "APARTMENT" NUMBER,
    FOREIGN KEY ("LOGIN") REFERENCES "USER"("LOGIN")
);

CREATE TABLE "MENU" (
    "PRODUCT ID" NUMBER,
    "NAME" VARCHAR2(100),
    "COST" NUMBER,
    "DATE" DATE,
    PRIMARY KEY ("PRODUCT ID")
);
```



```sql
USER
Begin

INSERT INTO "USER" ("LOGIN", "PASSWORD", "NAME", "ADDRESS", "TYPE")
VALUES ('icastelletto0', '3SOjtq6ISUV', 'Isabeau', '5638 Harper Drive', 'staff');

INSERT INTO "USER" ("LOGIN", "PASSWORD", "NAME", "ADDRESS", "TYPE")
VALUES ('acrowche1', 'l5h3l33', 'Angus', '15 Havey Lane', 'user');

INSERT INTO "USER" ("LOGIN", "PASSWORD", "NAME", "ADDRESS", "TYPE")
VALUES ('upuckey2', 'aa2OYthqotl', 'Ulysses', '95 Bartelt Place', 'staff');

INSERT INTO "USER" ("LOGIN", "PASSWORD", "NAME", "ADDRESS", "TYPE")
VALUES ('hgocke3', 'kyvjWEO3mnzc', 'Harriot', '68 Lillian Point', 'staff');

INSERT INTO "USER" ("LOGIN", "PASSWORD", "NAME", "ADDRESS", "TYPE")
VALUES ('ntremathack4', 'Wl0mdl', 'Neel', '92338 Mcguire Circle', 'staff');

INSERT INTO "USER" ("LOGIN", "PASSWORD", "NAME", "ADDRESS", "TYPE")
VALUES ('wmenis5', 'kk4yNCk', 'Way', '7211 Green Ridge Circle', 'staff');

INSERT INTO "USER" ("LOGIN", "PASSWORD", "NAME", "ADDRESS", "TYPE")
VALUES ('gjirusek6', 'mkdaRw', 'Gawen', '03764 Karstens Alley', 'user');
```

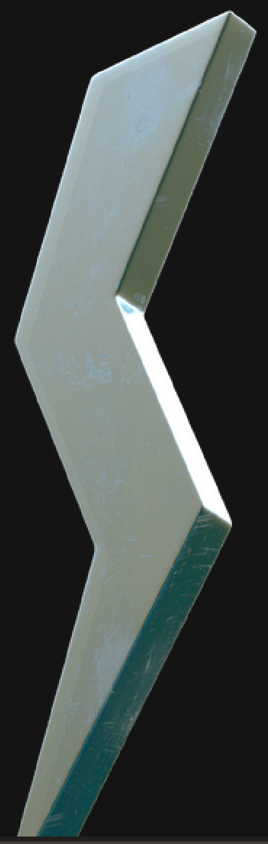
Trigger

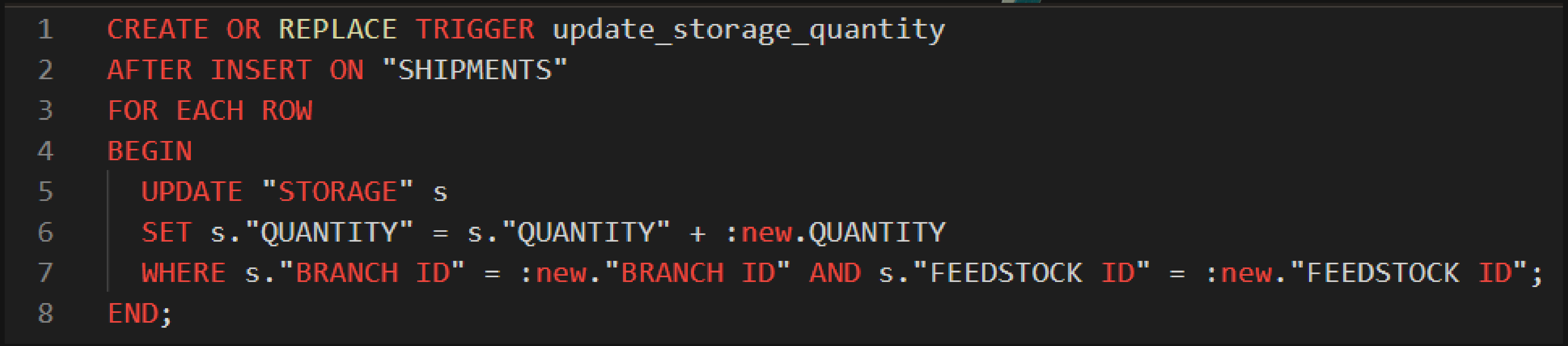
```
Current number of rows in the table: 21

1 row(s) inserted.
```

```
CREATE OR REPLACE TRIGGER show_row_count_menu
BEFORE INSERT ON "MENU"
FOR EACH ROW
DECLARE
  row_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO row_count FROM "MENU";
  DBMS_OUTPUT.PUT_LINE('Current number of rows in the table: ' || row_count);
END;
```

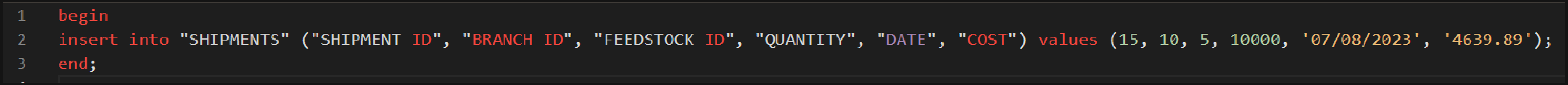
```
begin
insert into "MENU" ("PRODUCT ID", "NAME", "COST") values (22, 'ICED AMERICANO L', '27.76');
end;
```

Trigger



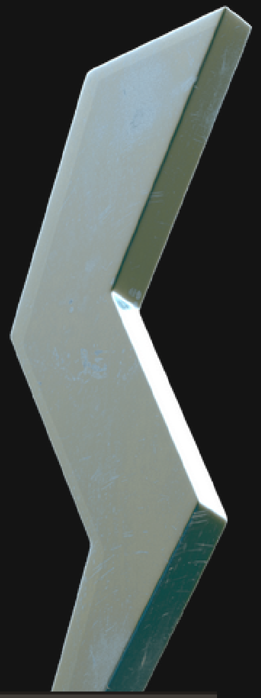


```
CREATE OR REPLACE TRIGGER update_storage_quantity
AFTER INSERT ON "SHIPMENTS"
FOR EACH ROW
BEGIN
    UPDATE "STORAGE" s
    SET s."QUANTITY" = s."QUANTITY" + :new.QUANTITY
    WHERE s."BRANCH ID" = :new."BRANCH ID" AND s."FEEDSTOCK ID" = :new."FEEDSTOCK ID";
END;
```

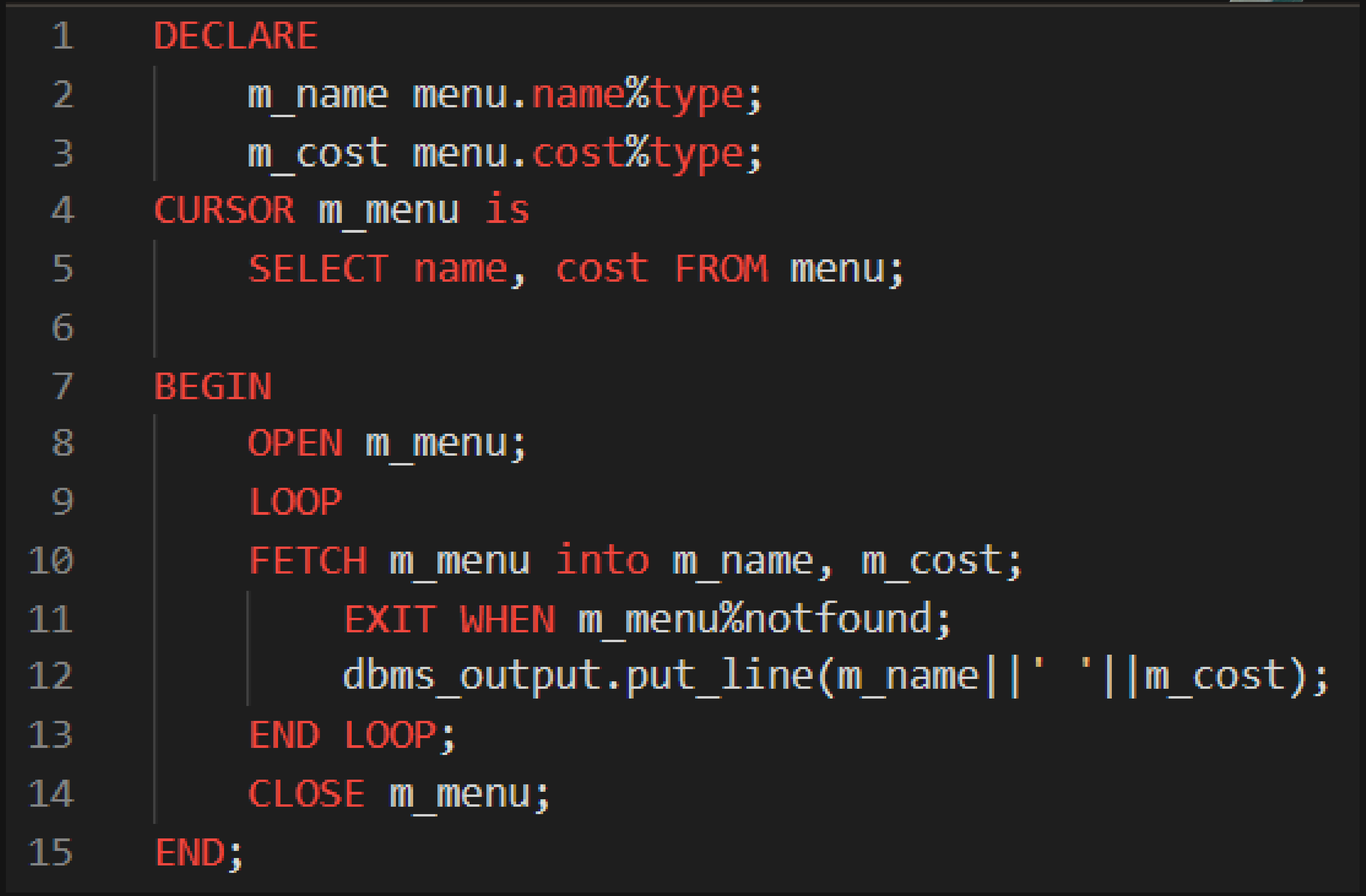


```
begin
insert into "SHIPMENTS" ("SHIPMENT ID", "BRANCH ID", "FEEDSTOCK ID", "QUANTITY", "DATE", "COST") values (15, 10, 5, 10000, '07/08/2023', '4639.89');
end;
```

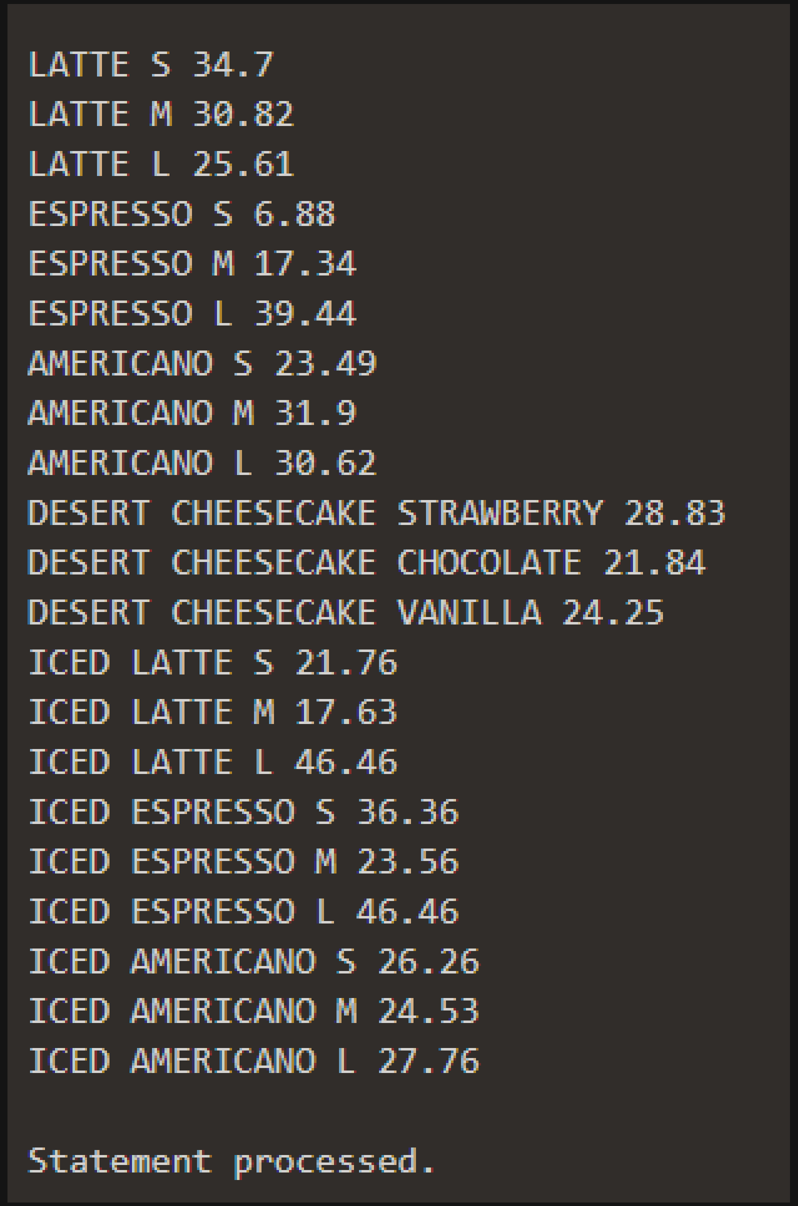
Cursor



This cursor shows the entire menu along with the price

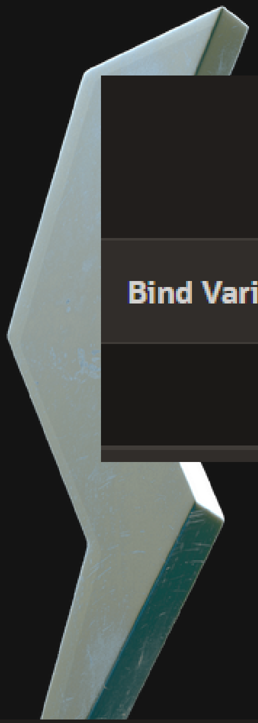


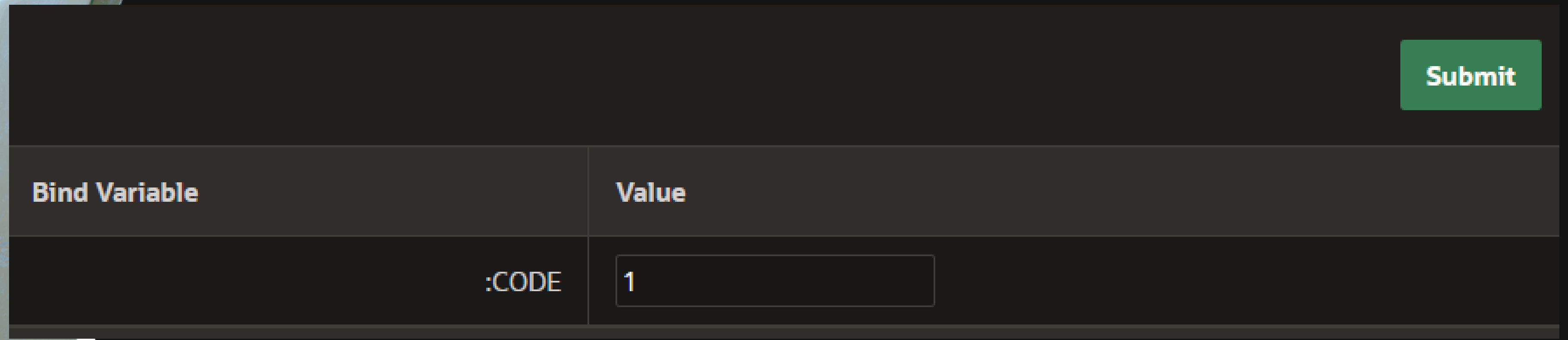
```
DECLARE
    m_name menu.name%type;
    m_cost menu.cost%type;
CURSOR m_menu is
    SELECT name, cost FROM menu;

BEGIN
    OPEN m_menu;
    LOOP
    FETCH m_menu into m_name, m_cost;
        EXIT WHEN m_menu%notfound;
        dbms_output.put_line(m_name||' '||m_cost);
    END LOOP;
    CLOSE m_menu;
END;
```



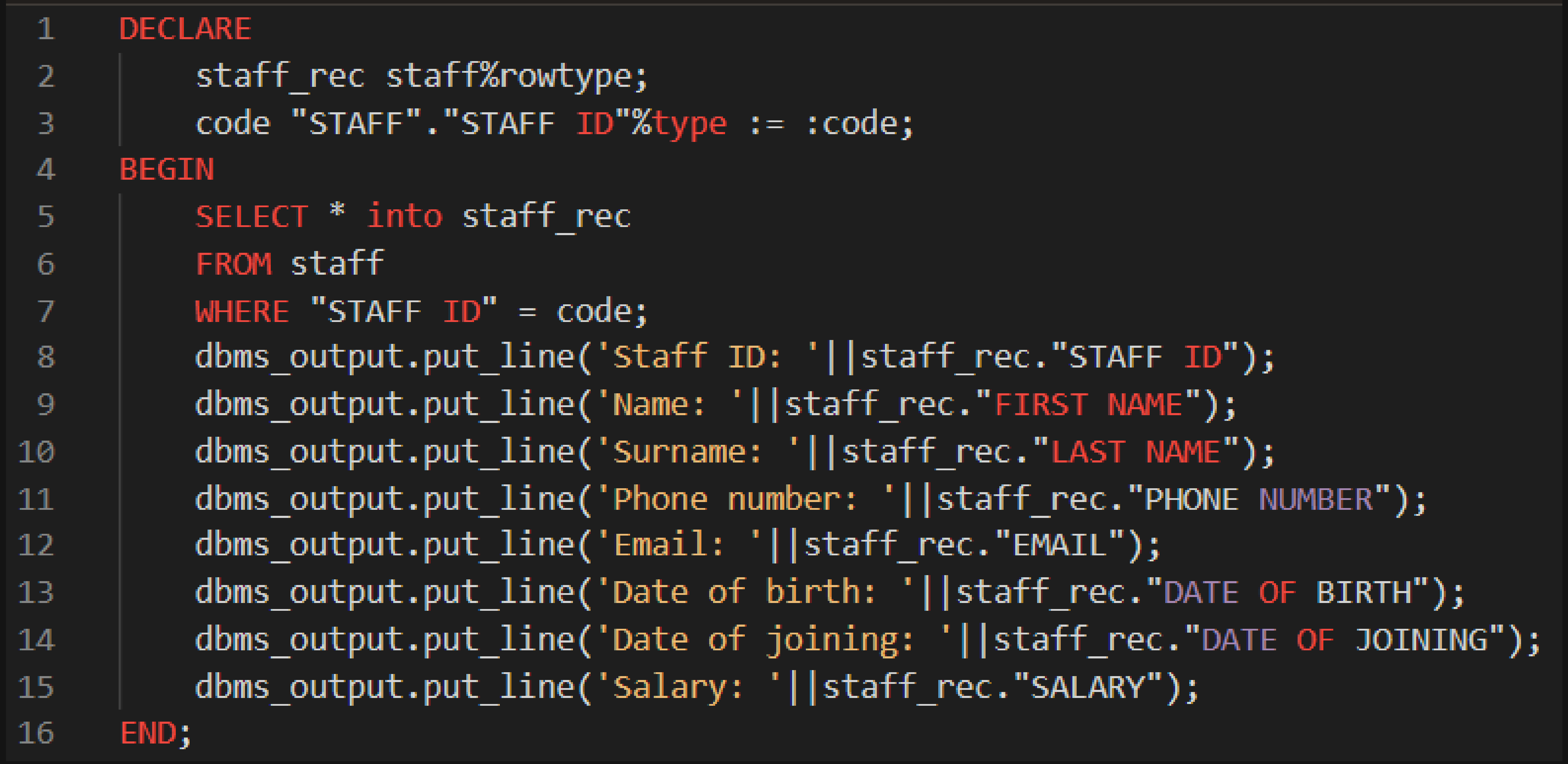
```
LATTE S 34.7
LATTE M 30.82
LATTE L 25.61
ESPRESSO S 6.88
ESPRESSO M 17.34
ESPRESSO L 39.44
AMERICANO S 23.49
AMERICANO M 31.9
AMERICANO L 30.62
DESERT CHEESECAKE STRAWBERRY 28.83
DESERT CHEESECAKE CHOCOLATE 21.84
DESERT CHEESECAKE VANILLA 24.25
ICED LATTE S 21.76
ICED LATTE M 17.63
ICED LATTE L 46.46
ICED ESPRESSO S 36.36
ICED ESPRESSO M 23.56
ICED ESPRESSO L 46.46
ICED AMERICANO S 26.26
ICED AMERICANO M 24.53
ICED AMERICANO L 27.76

Statement processed.
```

Table-based records

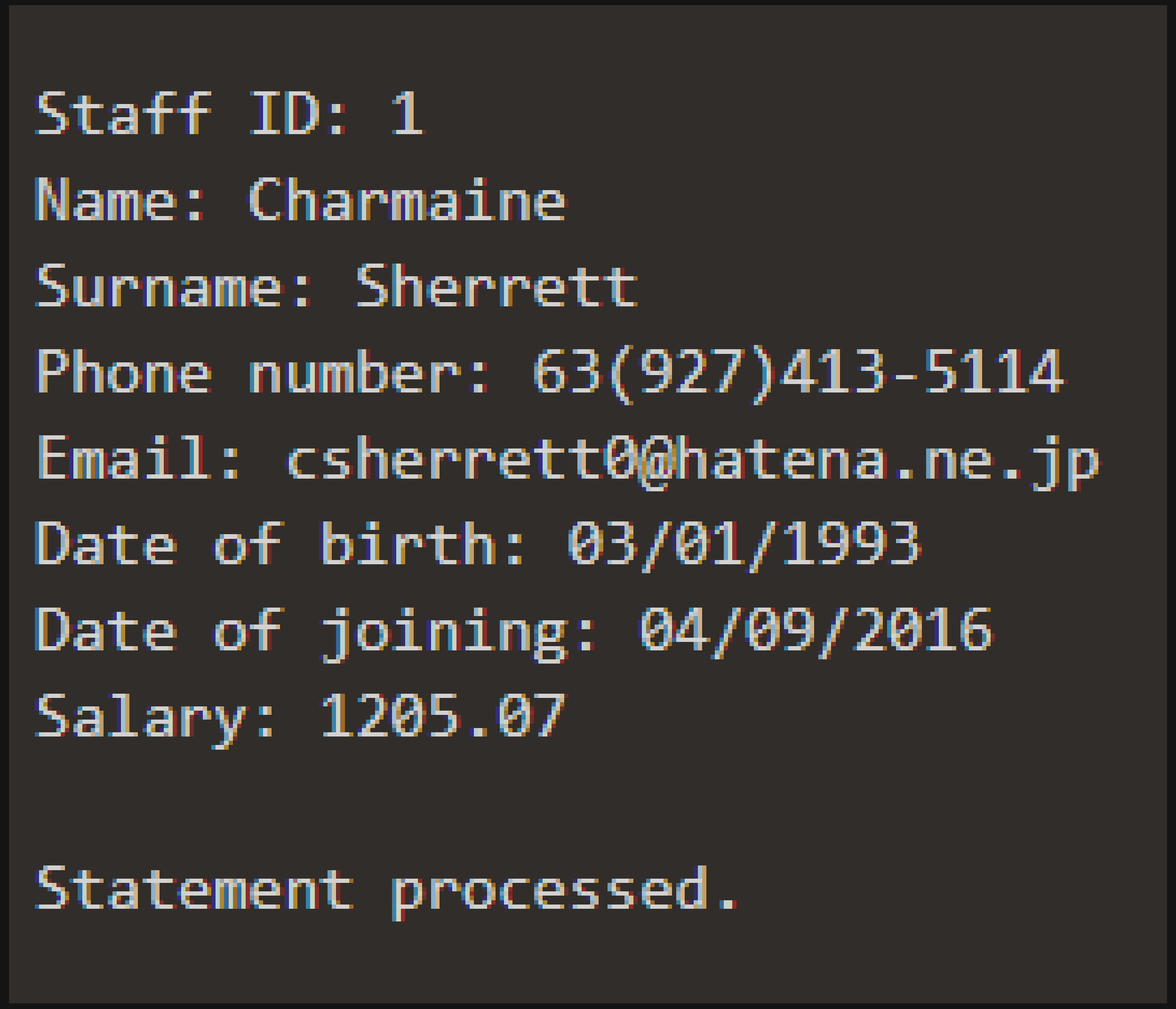




Bind Variable	Value
:CODE	1

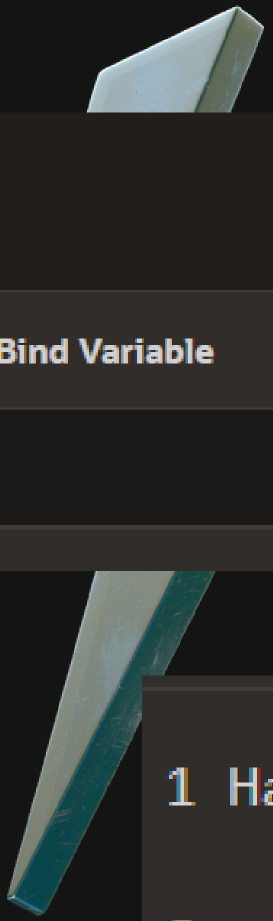


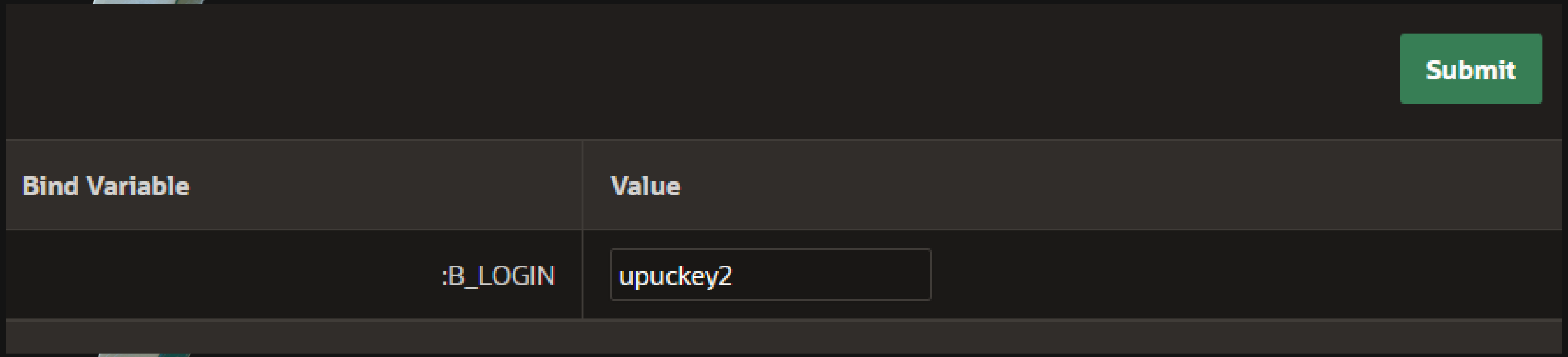
```
DECLARE
    staff_rec staff%rowtype;
    code "STAFF"."STAFF ID"%type := :code;
BEGIN
    SELECT * into staff_rec
    FROM staff
    WHERE "STAFF ID" = code;
    dbms_output.put_line('Staff ID: '||staff_rec."STAFF ID");
    dbms_output.put_line('Name: '||staff_rec."FIRST NAME");
    dbms_output.put_line('Surname: '||staff_rec."LAST NAME");
    dbms_output.put_line('Phone number: '||staff_rec."PHONE NUMBER");
    dbms_output.put_line('Email: '||staff_rec."EMAIL");
    dbms_output.put_line('Date of birth: '||staff_rec."DATE OF BIRTH");
    dbms_output.put_line('Date of joining: '||staff_rec."DATE OF JOINING");
    dbms_output.put_line('Salary: '||staff_rec."SALARY");
END;
```



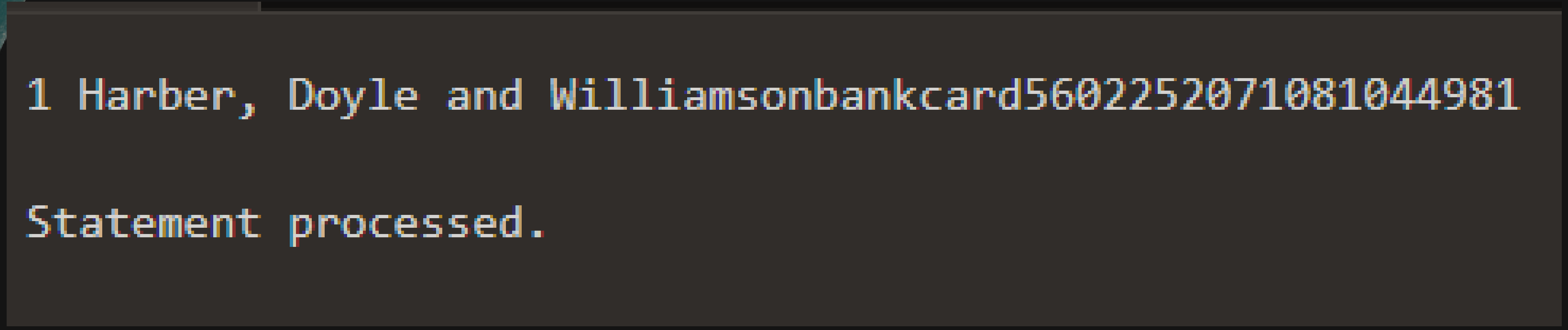
```
Staff ID: 1
Name: Charmaine
Surname: Sherrett
Phone number: 63(927)413-5114
Email: csherrett0@hatena.ne.jp
Date of birth: 03/01/1993
Date of joining: 04/09/2016
Salary: 1205.07

Statement processed.
```

Cursor-based records

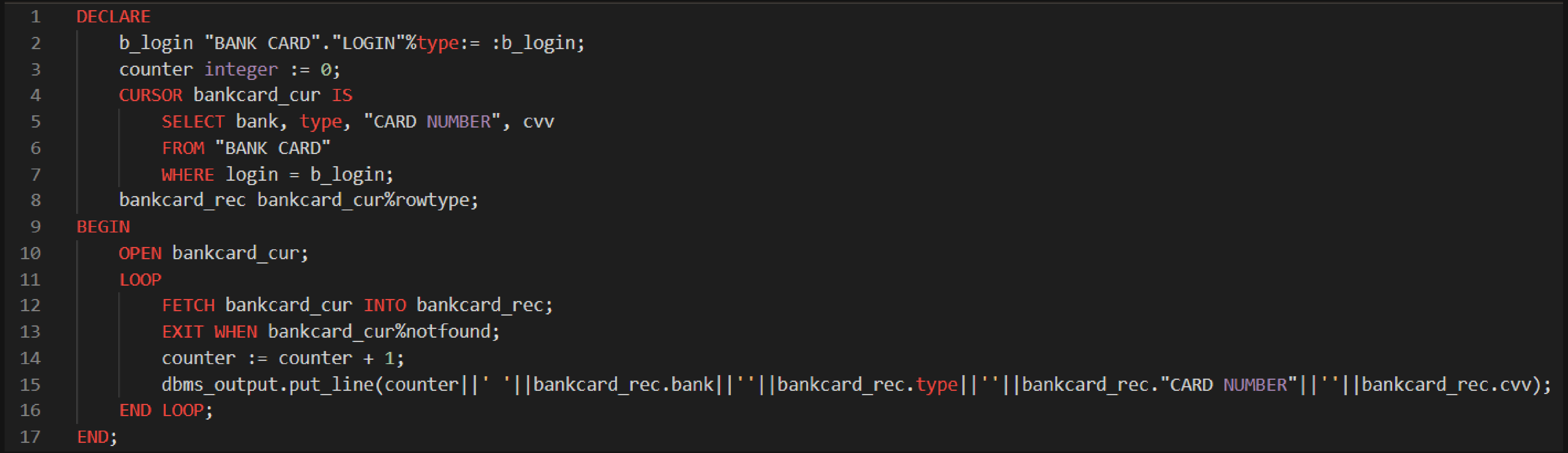




Bind Variable	Value
:B_LOGIN	upuckey2

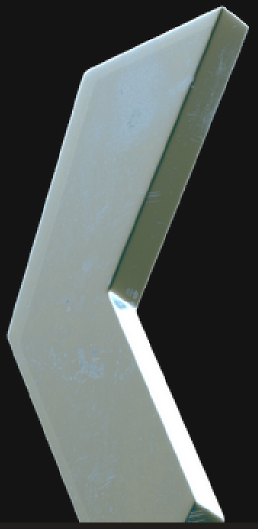


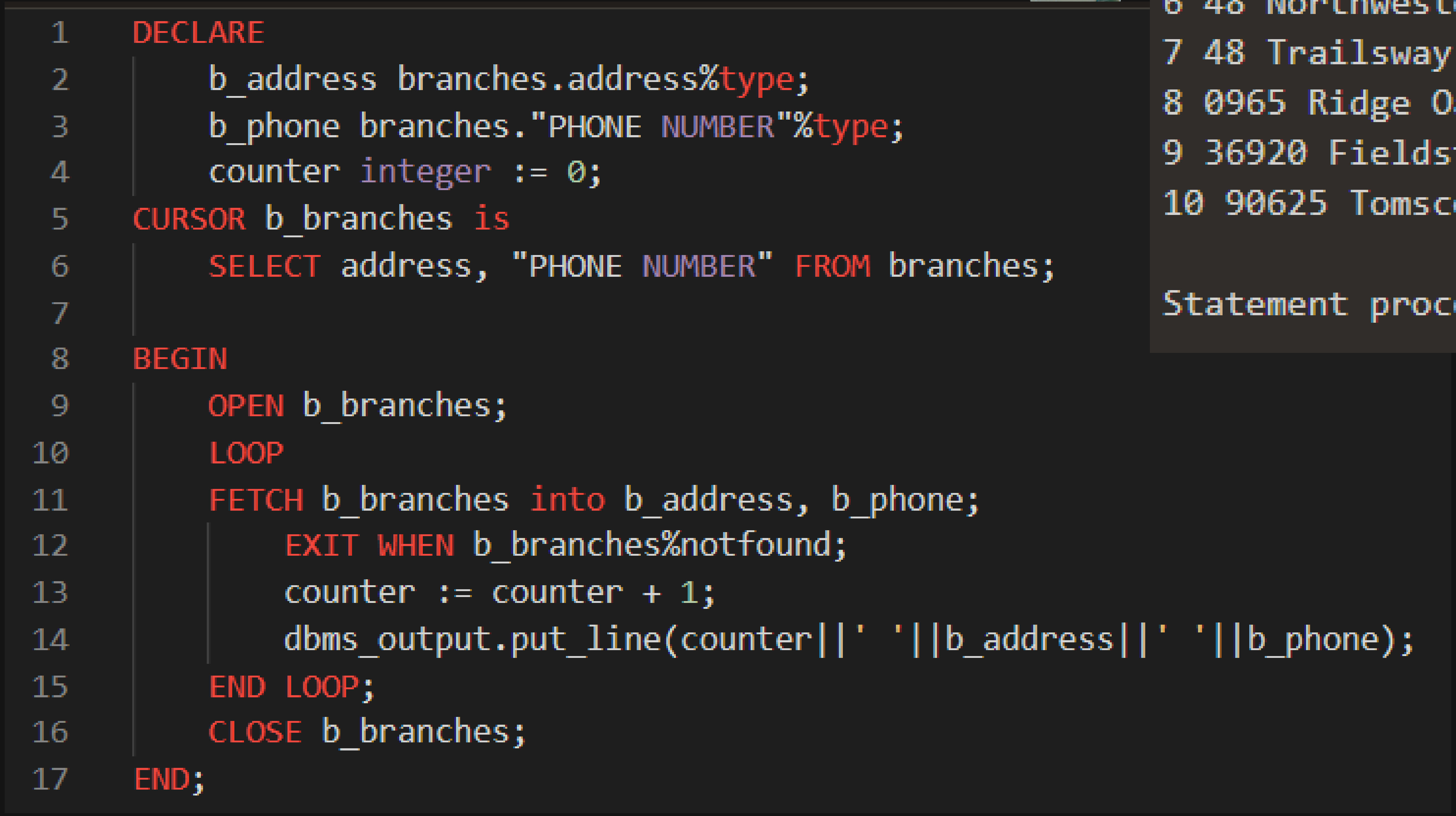
```
1 Harber, Doyle and Williamsonbankcard5602252071081044981

Statement processed.
```



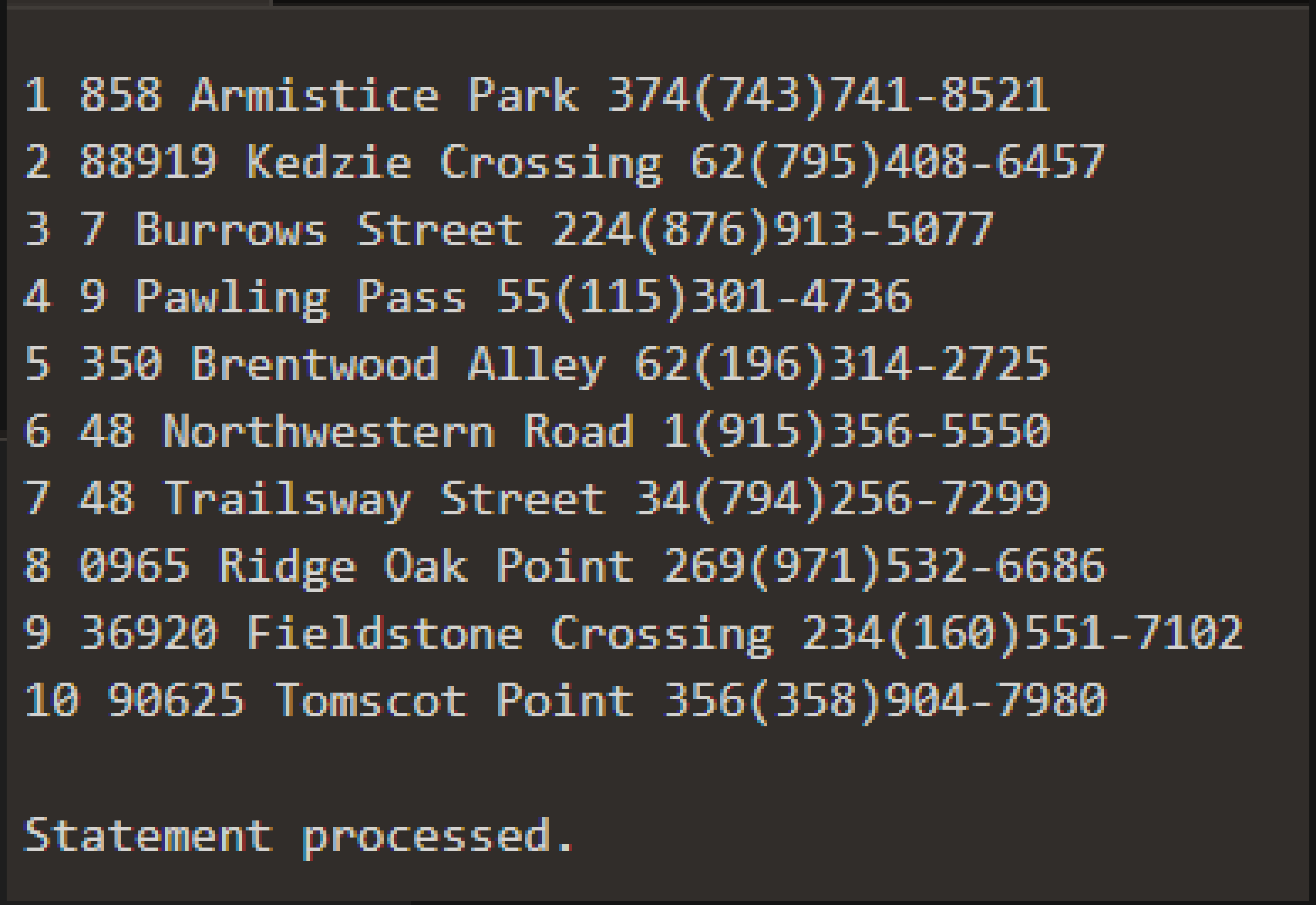
```
DECLARE
    b_login "BANK CARD"."LOGIN"%type:= :b_login;
    counter integer := 0;
    CURSOR bankcard_cur IS
        SELECT bank, type, "CARD NUMBER", cvv
        FROM "BANK CARD"
        WHERE login = b_login;
    bankcard_rec bankcard_cur%rowtype;
BEGIN
    OPEN bankcard_cur;
    LOOP
        FETCH bankcard_cur INTO bankcard_rec;
        EXIT WHEN bankcard_cur%notfound;
        counter := counter + 1;
        dbms_output.put_line(counter||' '||bankcard_rec.bank||''||bankcard_rec.type||''||bankcard_rec."CARD NUMBER"||''||bankcard_rec.cvv);
    END LOOP;
END;
```

Cursor



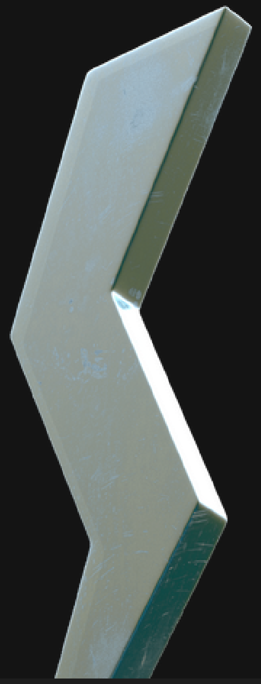


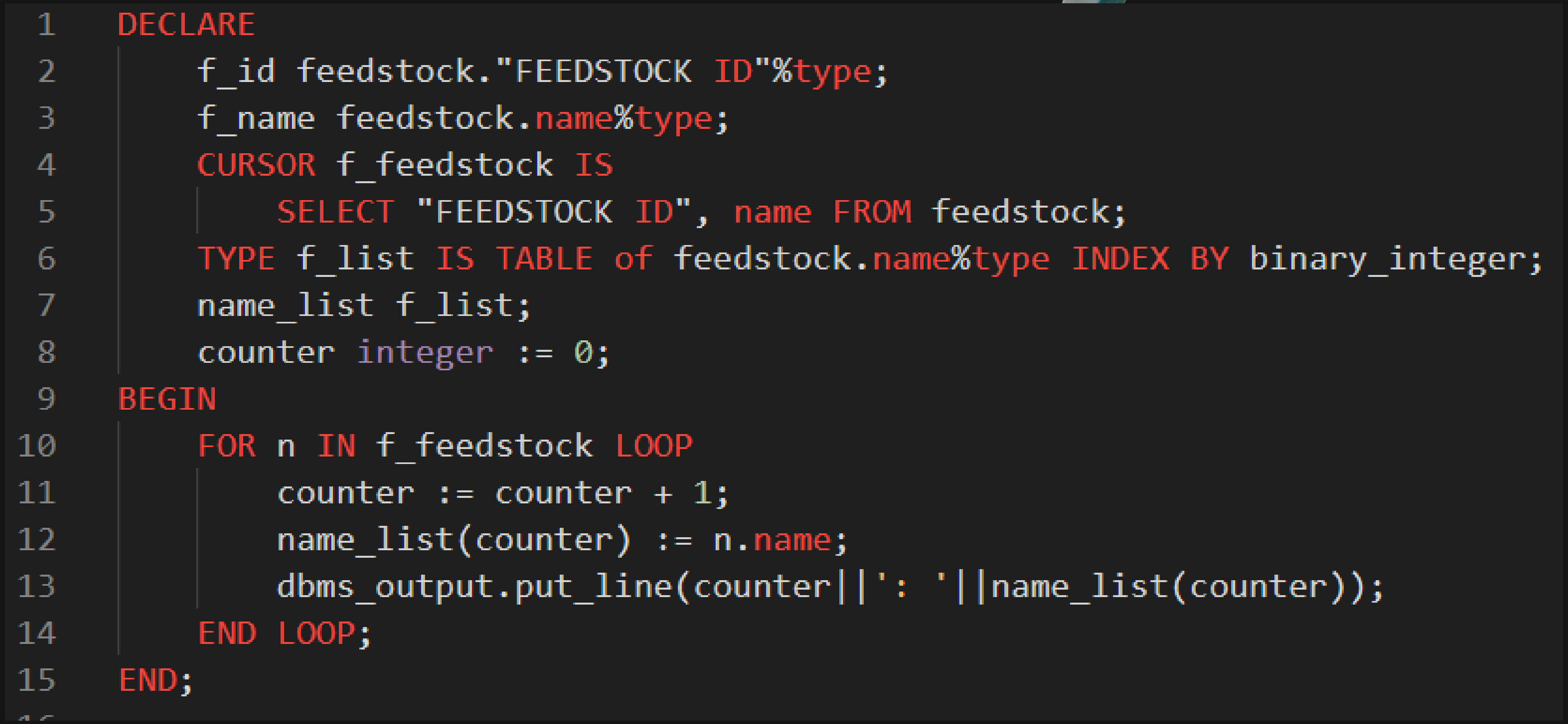
```
DECLARE
    b_address branches.address%type;
    b_phone branches."PHONE NUMBER"%type;
    counter integer := 0;
CURSOR b_branches is
    SELECT address, "PHONE NUMBER" FROM branches;

BEGIN
    OPEN b_branches;
    LOOP
    FETCH b_branches into b_address, b_phone;
        EXIT WHEN b_branches%notfound;
        counter := counter + 1;
        dbms_output.put_line(counter||' '||b_address||' '||b_phone);
    END LOOP;
    CLOSE b_branches;
END;
```



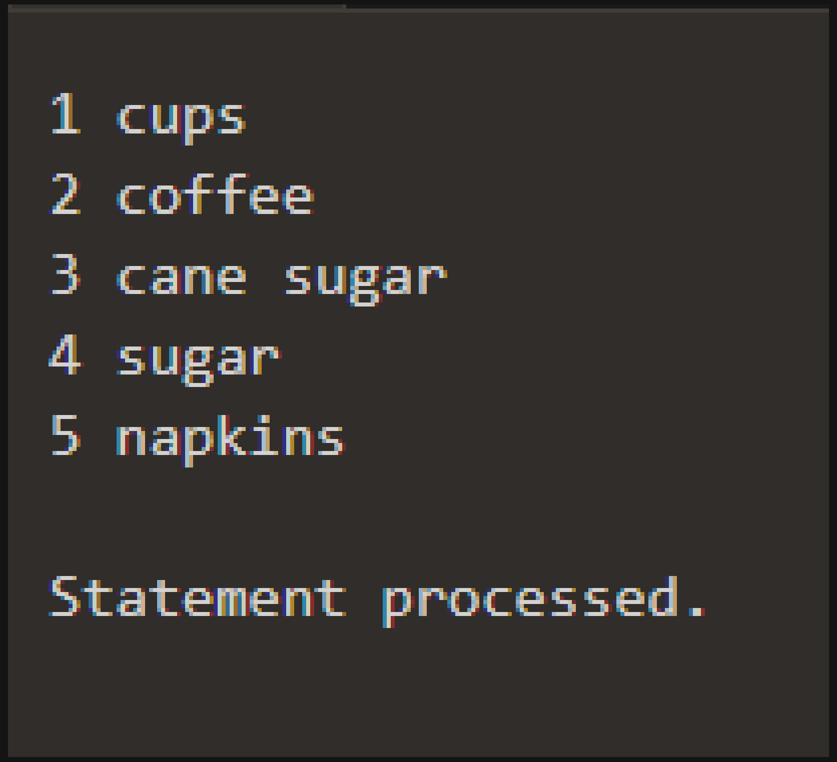
```
1 858 Armistice Park 374(743)741-8521
2 88919 Kedzie Crossing 62(795)408-6457
3 7 Burrows Street 224(876)913-5077
4 9 Pawling Pass 55(115)301-4736
5 350 Brentwood Alley 62(196)314-2725
6 48 Northwestern Road 1(915)356-5550
7 48 Trailsway Street 34(794)256-7299
8 0965 Ridge Oak Point 269(971)532-6686
9 36920 Fieldstone Crossing 234(160)551-7102
10 90625 Tomscot Point 356(358)904-7980

Statement processed.
```

Collections+ Cursor



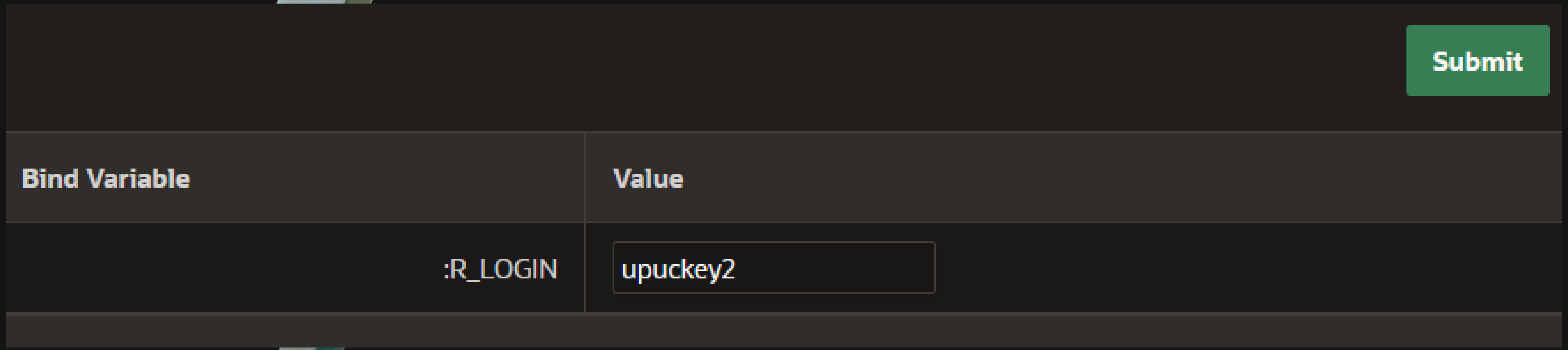


```
DECLARE
    f_id feedstock."FEEDSTOCK ID"%type;
    f_name feedstock.name%type;
    CURSOR f_feedstock IS
        SELECT "FEEDSTOCK ID", name FROM feedstock;
    TYPE f_list IS TABLE of feedstock.name%type INDEX BY binary_integer;
    name_list f_list;
    counter integer := 0;
BEGIN
    FOR n IN f_feedstock LOOP
        counter := counter + 1;
        name_list(counter) := n.name;
        dbms_output.put_line(counter||': '||name_list(counter));
    END LOOP;
END;
```

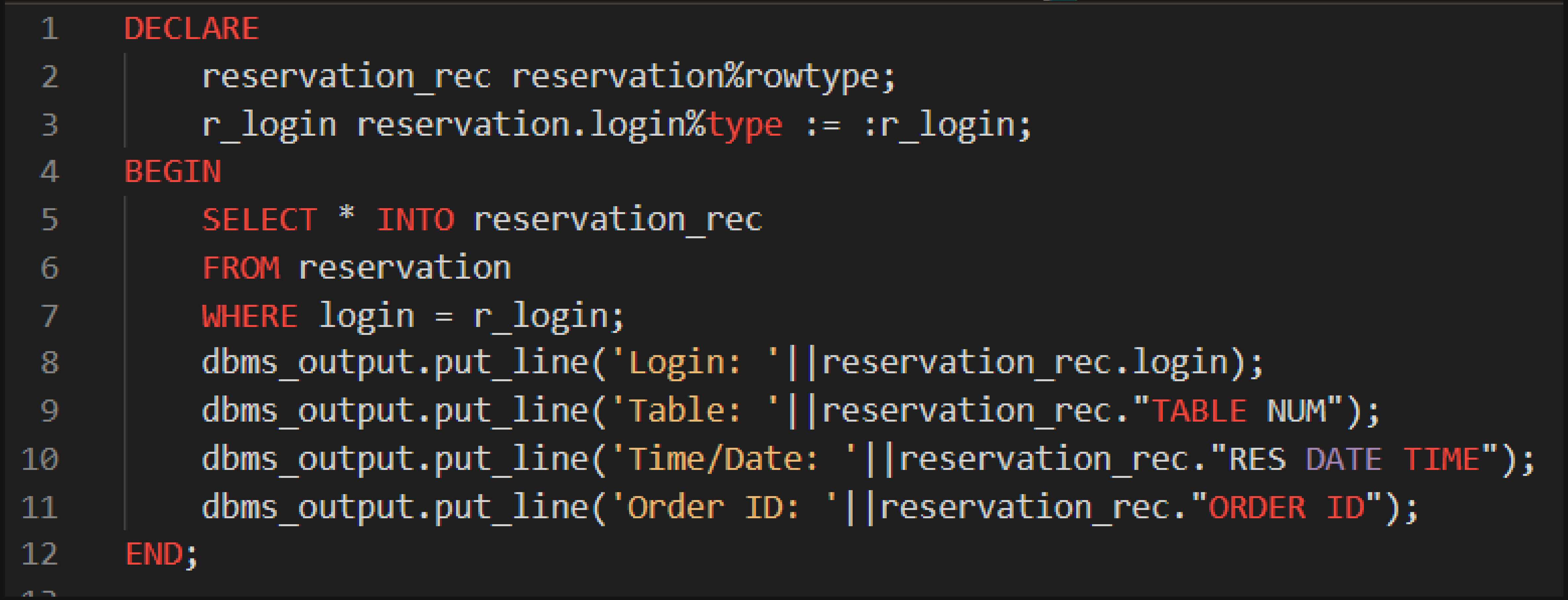


```
1 cups
2 coffee
3 cane sugar
4 sugar
5 napkins

Statement processed.
```

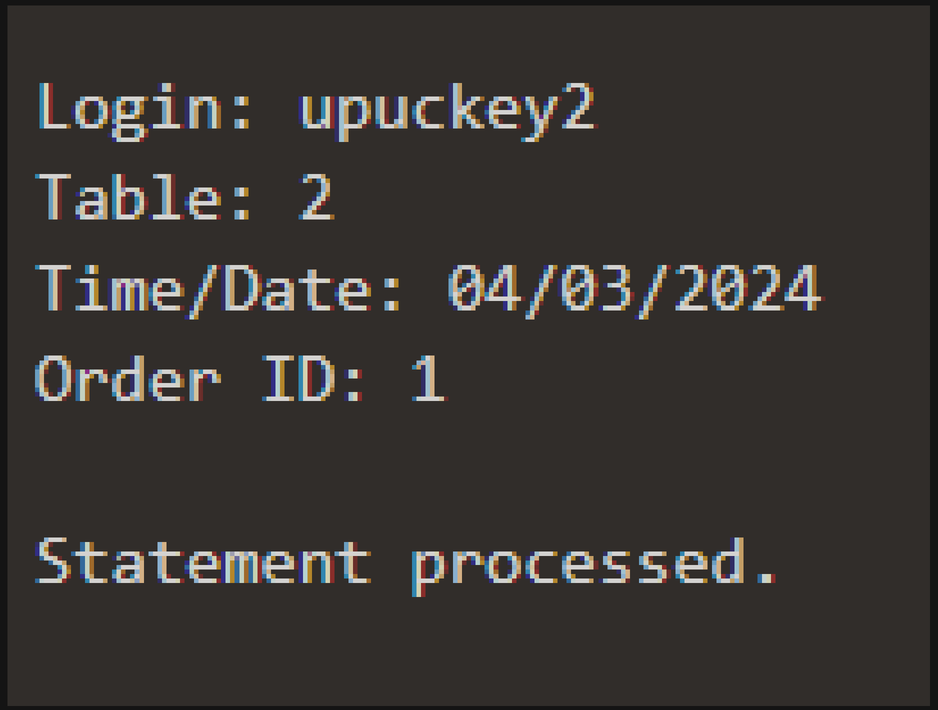
Table-based records



Bind Variable	Value
:R_LOGIN	upuckey2

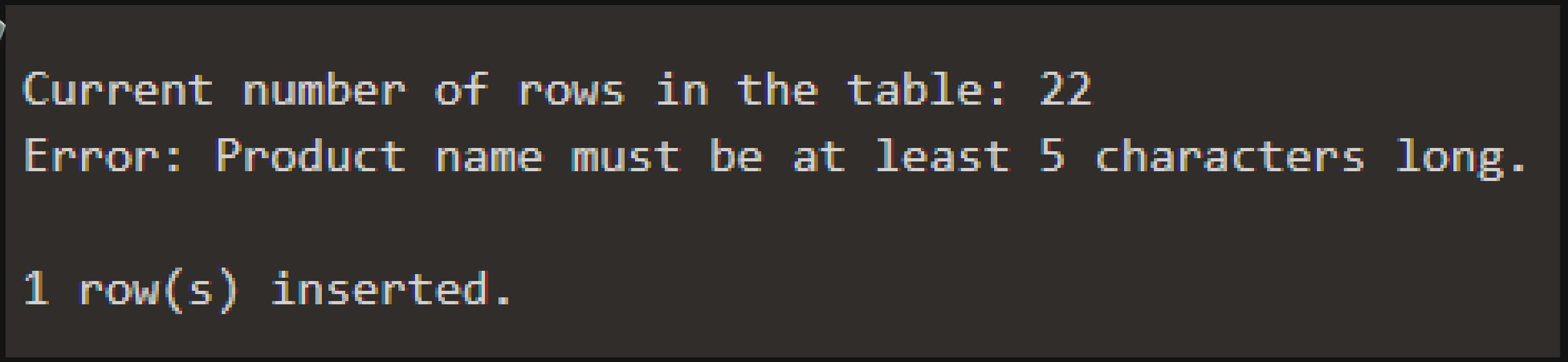


```
DECLARE
    reservation_rec reservation%rowtype;
    r_login reservation.login%type := :r_login;
BEGIN
    SELECT * INTO reservation_rec
    FROM reservation
    WHERE login = r_login;
    dbms_output.put_line('Login: '||reservation_rec.login);
    dbms_output.put_line('Table: '||reservation_rec."TABLE NUM");
    dbms_output.put_line('Time/Date: '||reservation_rec."RES DATE TIME");
    dbms_output.put_line('Order ID: '||reservation_rec."ORDER ID");
END;
```

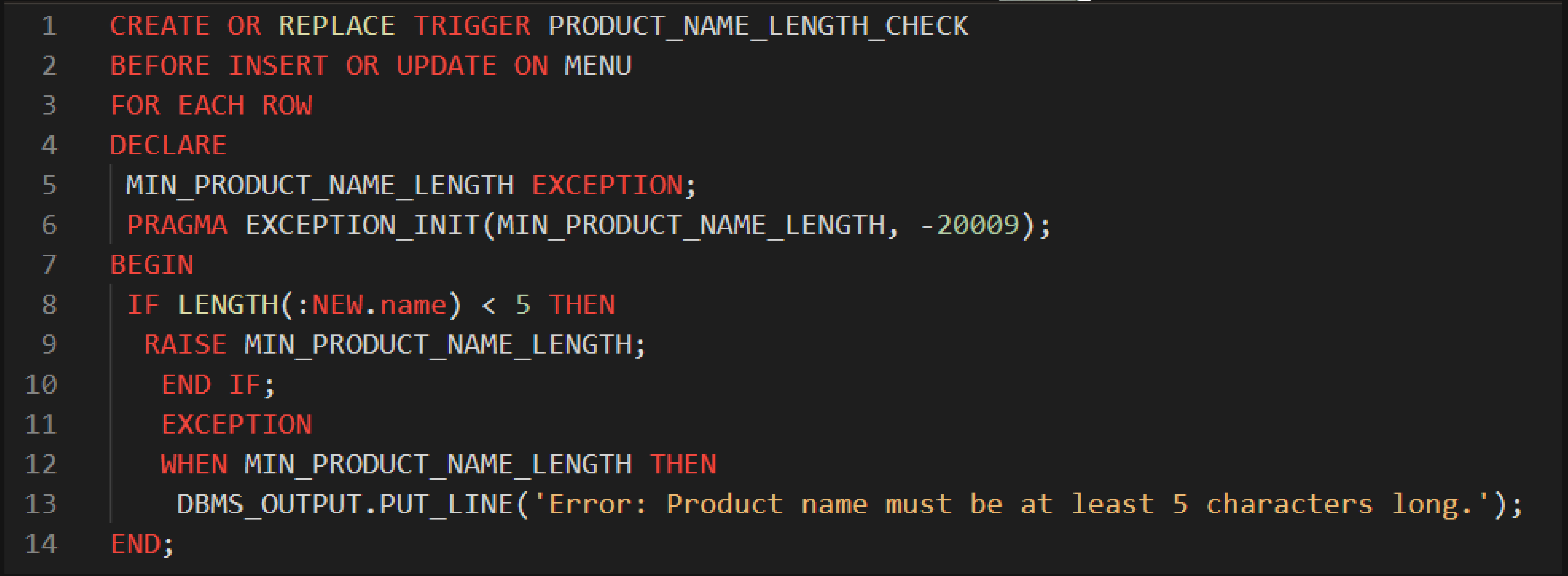


```
Login: upuckey2
Table: 2
Time/Date: 04/03/2024
Order ID: 1

Statement processed.
```

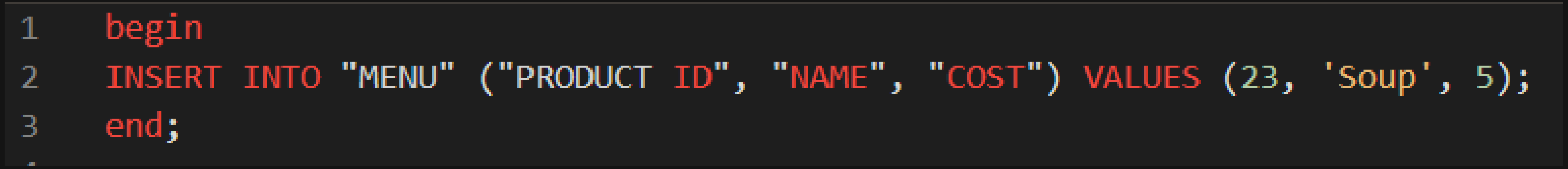
Trigger



```
Current number of rows in the table: 22
Error: Product name must be at least 5 characters long.

1 row(s) inserted.
```

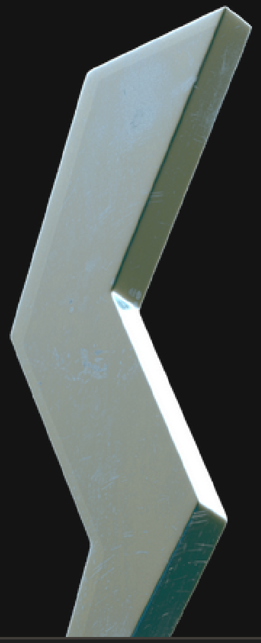


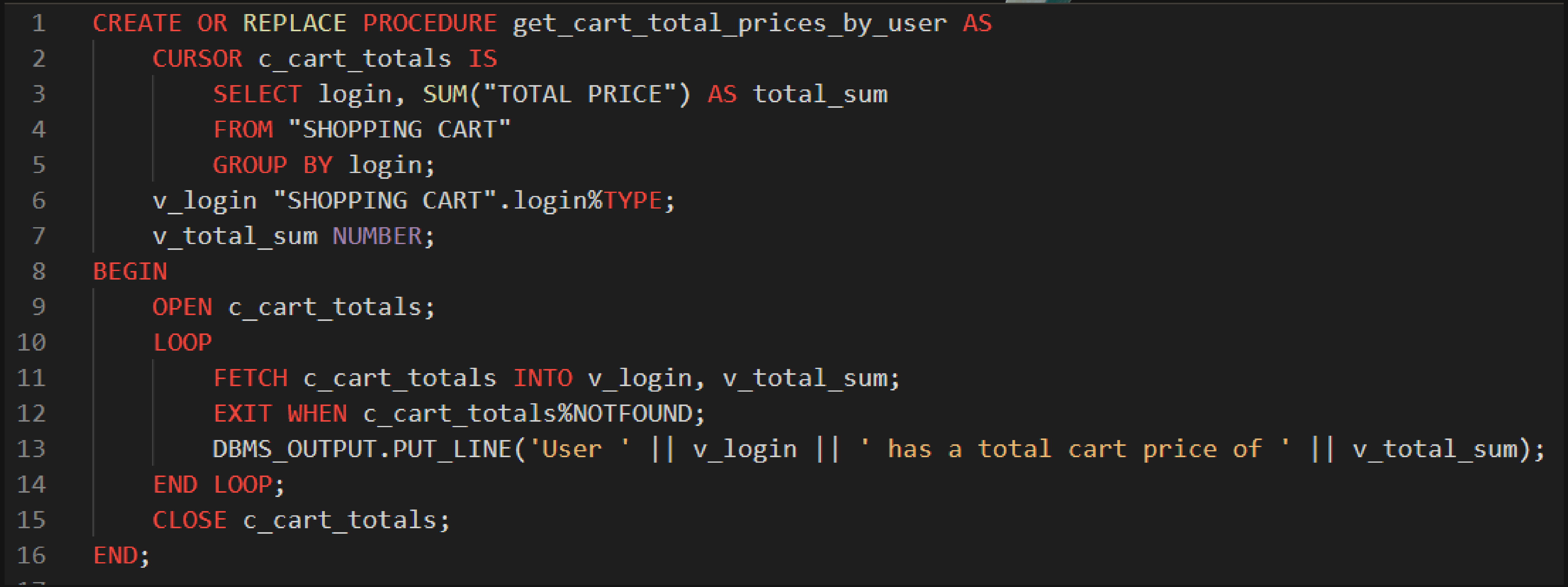
```
CREATE OR REPLACE TRIGGER PRODUCT_NAME_LENGTH_CHECK
BEFORE INSERT OR UPDATE ON MENU
FOR EACH ROW
DECLARE
 MIN_PRODUCT_NAME_LENGTH EXCEPTION;
 PRAGMA EXCEPTION_INIT(MIN_PRODUCT_NAME_LENGTH, -20009);
BEGIN
 IF LENGTH(:NEW.name) < 5 THEN
  RAISE MIN_PRODUCT_NAME_LENGTH;
   END IF;
   EXCEPTION
   WHEN MIN_PRODUCT_NAME_LENGTH THEN
    DBMS_OUTPUT.PUT_LINE('Error: Product name must be at least 5 characters long.');
END;
```



```
begin
INSERT INTO "MENU" ("PRODUCT ID", "NAME", "COST") VALUES (23, 'Soup', 5);
end;
```

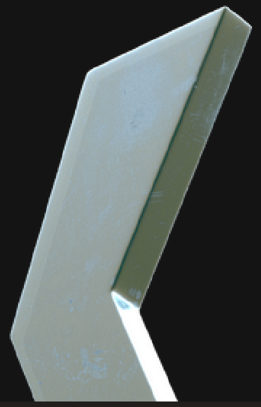
Procedure

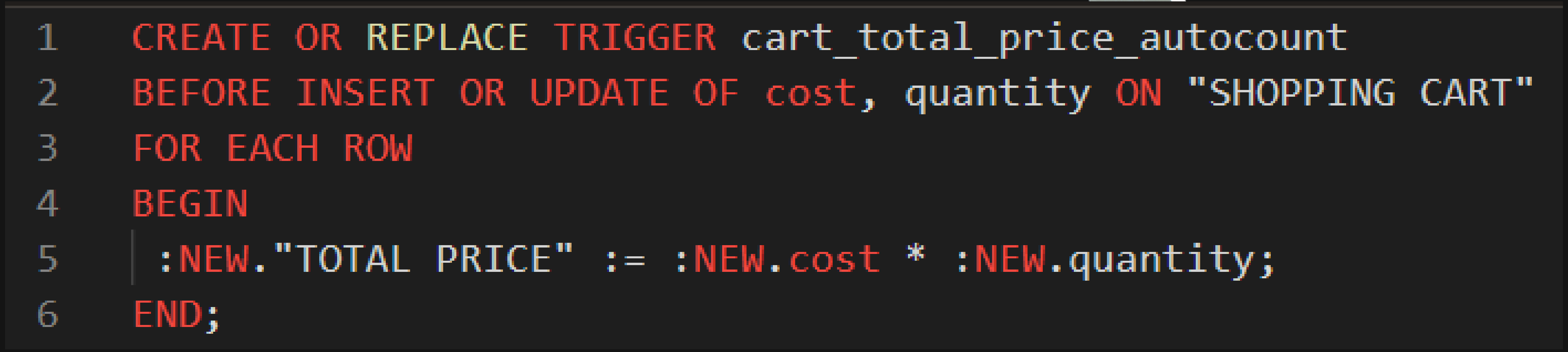




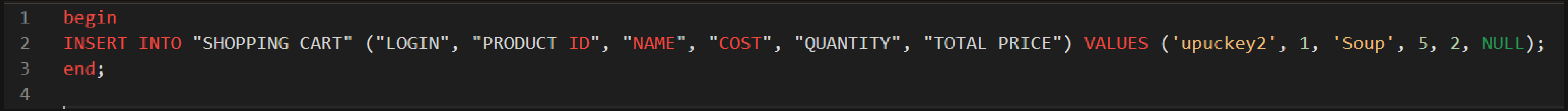
```sql
CREATE OR REPLACE PROCEDURE get_cart_total_prices_by_user AS
    CURSOR c_cart_totals IS
        SELECT login, SUM("TOTAL PRICE") AS total_sum
        FROM "SHOPPING CART"
        GROUP BY login;
    v_login "SHOPPING CART".login%TYPE;
    v_total_sum NUMBER;
BEGIN
    OPEN c_cart_totals;
    LOOP
        FETCH c_cart_totals INTO v_login, v_total_sum;
        EXIT WHEN c_cart_totals%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('User ' || v_login || ' has a total cart price of ' || v_total_sum);
    END LOOP;
    CLOSE c_cart_totals;
END;
```

Trigger

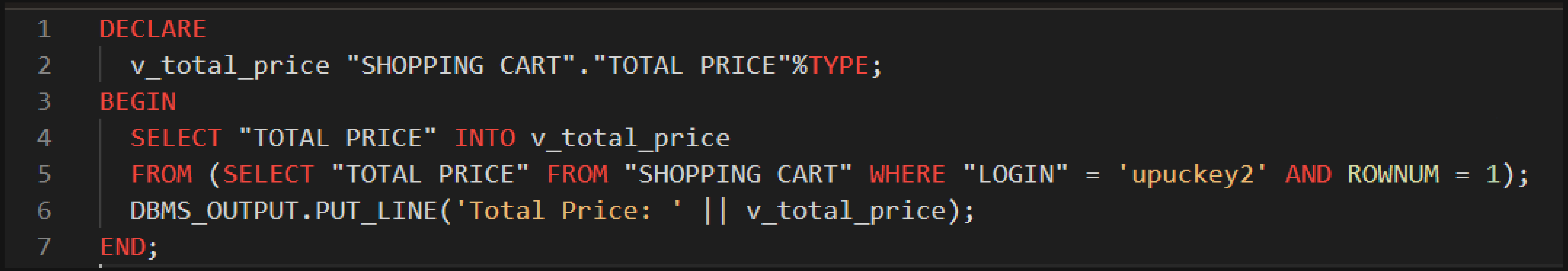




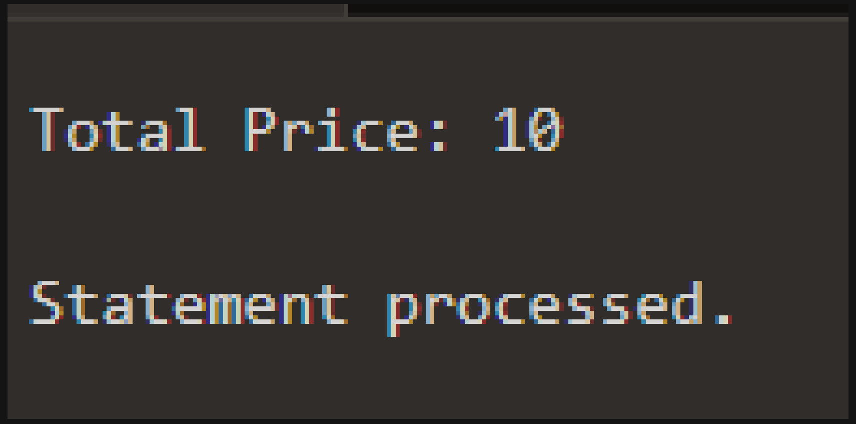
```
CREATE OR REPLACE TRIGGER cart_total_price_autocount
BEFORE INSERT OR UPDATE OF cost, quantity ON "SHOPPING CART"
FOR EACH ROW
BEGIN
  :NEW."TOTAL PRICE" := :NEW.cost * :NEW.quantity;
END;
```



```
begin
INSERT INTO "SHOPPING CART" ("LOGIN", "PRODUCT ID", "NAME", "COST", "QUANTITY", "TOTAL PRICE") VALUES ('upuckey2', 1, 'Soup', 5, 2, NULL);
end;

```

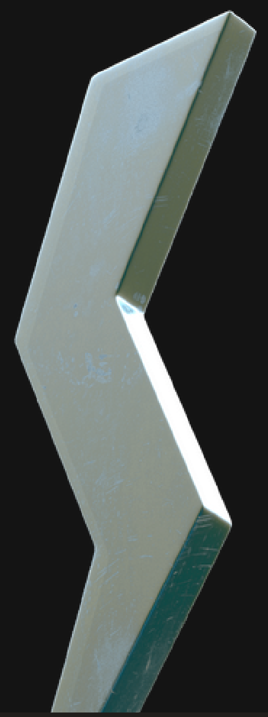


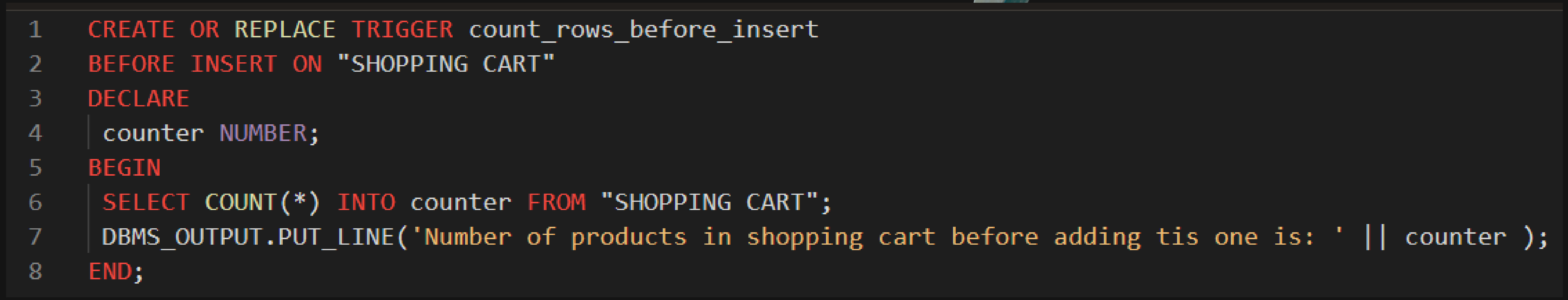
```
DECLARE
  v_total_price "SHOPPING CART"."TOTAL PRICE"%TYPE;
BEGIN
  SELECT "TOTAL PRICE" INTO v_total_price
  FROM (SELECT "TOTAL PRICE" FROM "SHOPPING CART" WHERE "LOGIN" = 'upuckey2' AND ROWNUM = 1);
  DBMS_OUTPUT.PUT_LINE('Total Price: ' || v_total_price);
END;
```



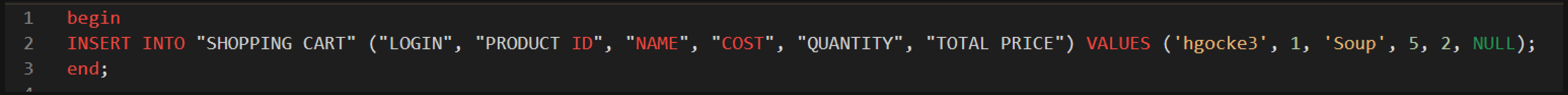
```
Total Price: 10

Statement processed.
```

Trigger

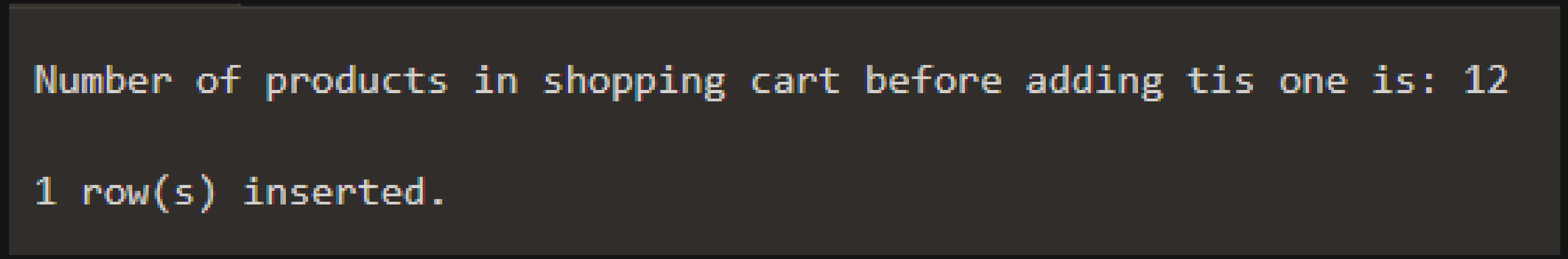




```sql
CREATE OR REPLACE TRIGGER count_rows_before_insert
BEFORE INSERT ON "SHOPPING CART"
DECLARE
  counter NUMBER;
BEGIN
  SELECT COUNT(*) INTO counter FROM "SHOPPING CART";
  DBMS_OUTPUT.PUT_LINE('Number of products in shopping cart before adding tis one is: ' || counter );
END;
```

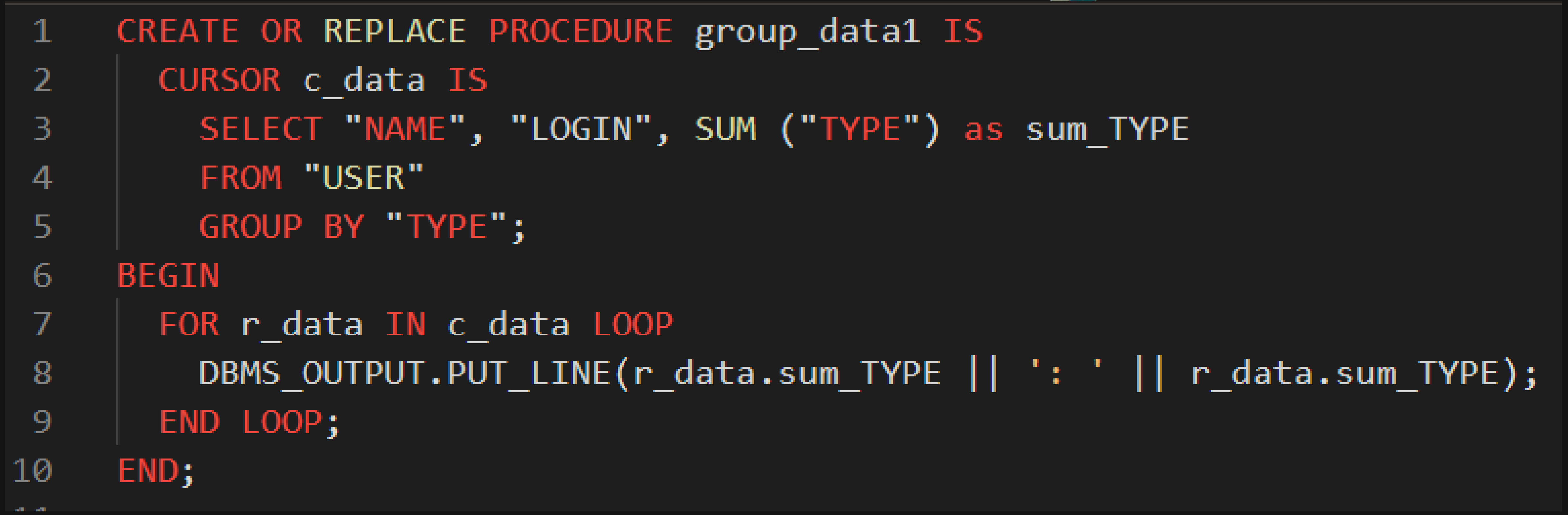


```sql
begin
INSERT INTO "SHOPPING CART" ("LOGIN", "PRODUCT ID", "NAME", "COST", "QUANTITY", "TOTAL PRICE") VALUES ('hgocke3', 1, 'Soup', 5, 2, NULL);
end;
```



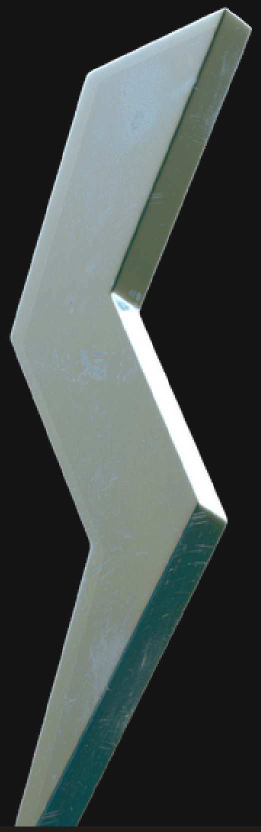
```
Number of products in shopping cart before adding tis one is: 12

1 row(s) inserted.
```

Procedure



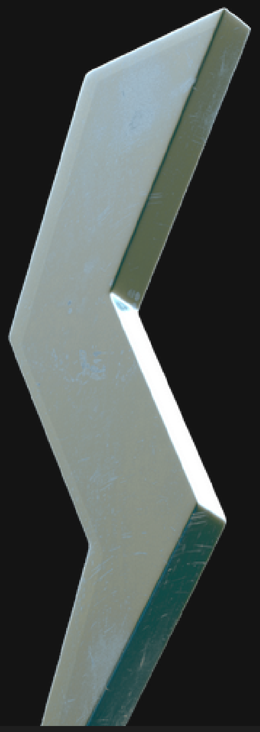
```
CREATE OR REPLACE PROCEDURE group_data1 IS
  CURSOR c_data IS
    SELECT "NAME", "LOGIN", SUM ("TYPE") as sum_TYPE
    FROM "USER"
    GROUP BY "TYPE";
BEGIN
  FOR r_data IN c_data LOOP
    DBMS_OUTPUT.PUT_LINE(r_data.sum_TYPE || ': ' || r_data.sum_TYPE);
  END LOOP;
END;
```

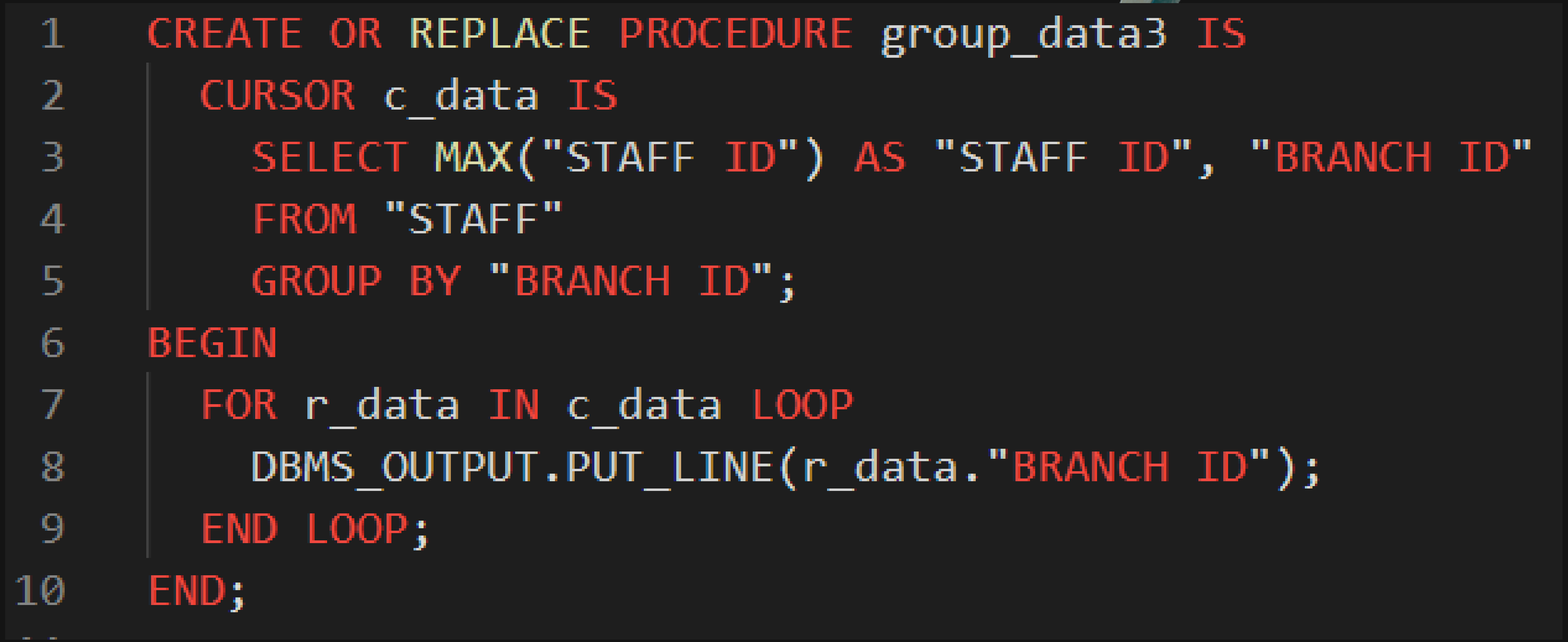
Procedure



```
CREATE OR REPLACE PROCEDURE group_data2 IS
  CURSOR c_data IS
    SELECT "BRANCH ID", "FEEDSTOCK ID", SUM("QUANTITY") as sum_QUANTITY
    FROM "STORAGE"
    GROUP BY "BRANCH ID", "FEEDSTOCK ID";
BEGIN
  FOR r_data IN c_data LOOP
    DBMS_OUTPUT.PUT_LINE(r_data."BRANCH ID" || ', ' ||  r_data."FEEDSTOCK ID" || ': ' || r_data.sum_QUANTITY);
  END LOOP;
END;
```

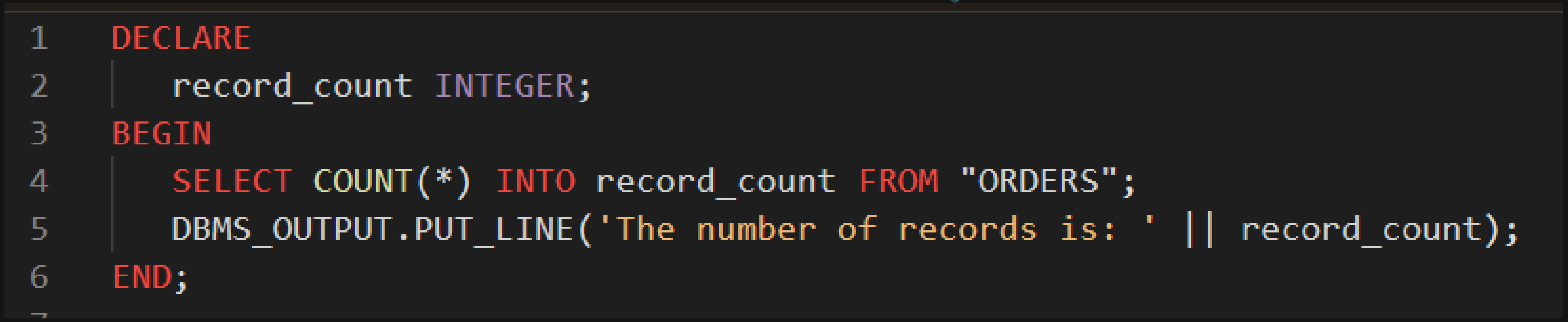
Procedure





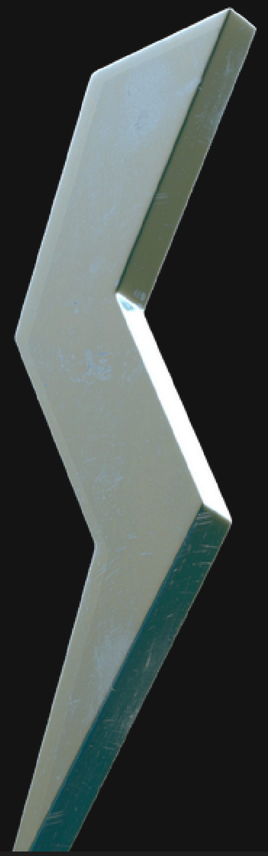
```
CREATE OR REPLACE PROCEDURE group_data3 IS
  CURSOR c_data IS
    SELECT MAX("STAFF ID") AS "STAFF ID", "BRANCH ID"
    FROM "STAFF"
    GROUP BY "BRANCH ID";
BEGIN
  FOR r_data IN c_data LOOP
    DBMS_OUTPUT.PUT_LINE(r_data."BRANCH ID");
  END LOOP;
END;
```

Record counter



```
DECLARE
    record_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO record_count FROM "ORDERS";
    DBMS_OUTPUT.PUT_LINE('The number of records is: ' || record_count);
END;
```

Procedure



```sql
CREATE OR REPLACE PROCEDURE delete_succeed_transactions IS
  deleted_count NUMBER;
BEGIN
  DELETE FROM "TRANSACTIONS"
  WHERE status = 'SUCCESS';
  deleted_count := SQL%ROWCOUNT;

  DBMS_OUTPUT.PUT_LINE('Number of successful transactions deleted: ' || deleted_count);
END;
```