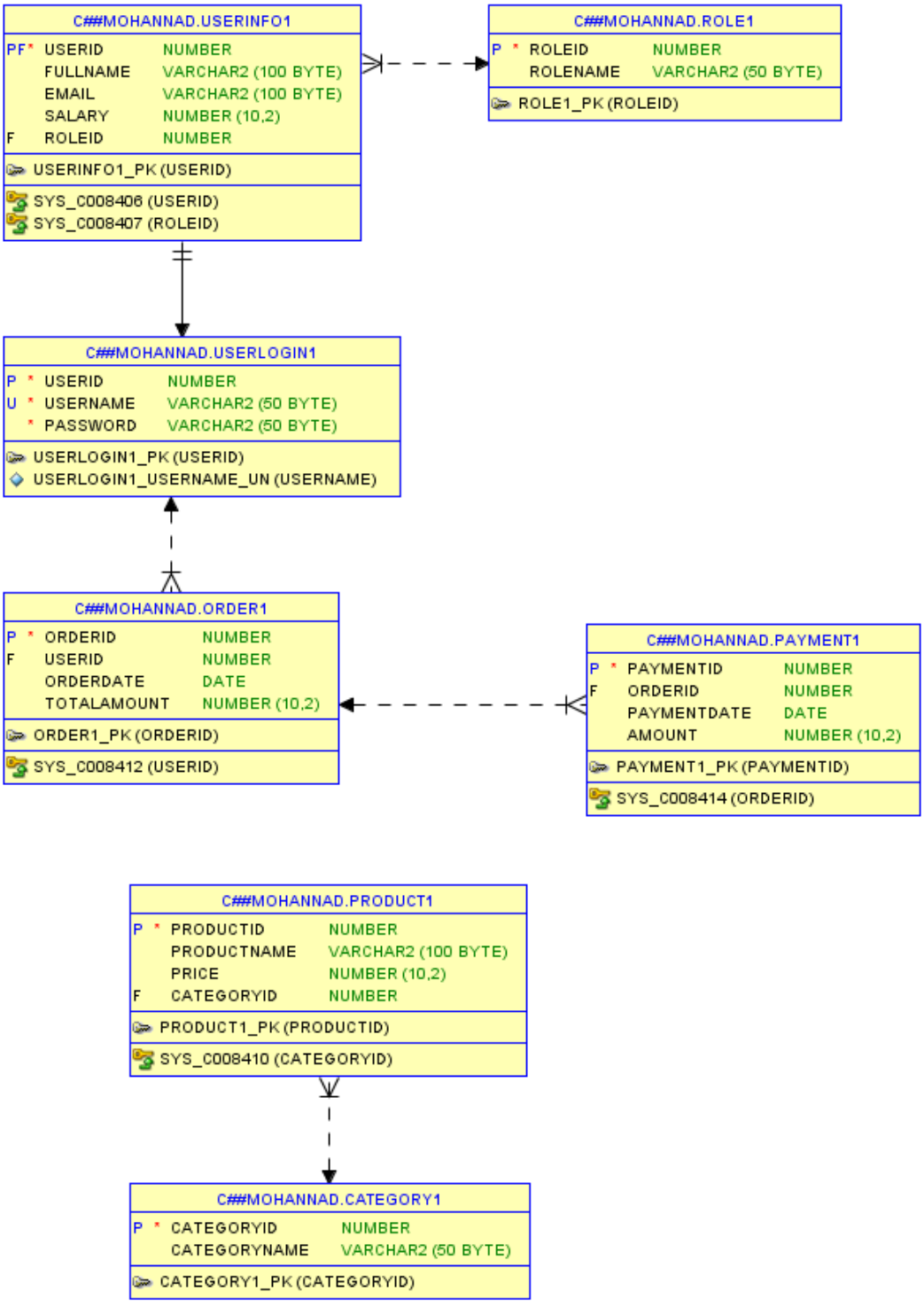
Assignment 9:

Class Diagram:



1- Display user information by username and password:

```
create or replace procedure display_user(username in varchar2, password in varchar2)
is
v_fullname VARCHAR2(100);
v_email VARCHAR2(100);
v_salary NUMBER;
v_roleid NUMBER;
begin
select userinfo1.fullname, userinfo1.email, userinfo1.salary, userinfo1.roleid
into v_fullname, v_email, v_salary, v_roleid
from userinfo1
inner join userlogin1 on userinfo1.userid = userlogin1.userid
where userlogin1.username = username AND userlogin1.password = password;

DBMS_OUTPUT.PUT_LINE('Full Name: ' || v_fullname);
DBMS_OUTPUT.PUT_LINE('Email: ' || v_email);
DBMS_OUTPUT.PUT_LINE('Salary: ' || v_salary);
DBMS_OUTPUT.PUT_LINE('Role ID: ' || v_roleid);

end display_user;
```

```
BEGIN
   display_user('user1', 'pass1');
END;
```

```
END;
Error report -
ORA-01422: exact fetch returns more than requested number of rows
ORA-06512: at "C##MOHANNAD.DISPLAY_USER", line 8
ORA-06512: at line 2
01422. 00000 -  "exact fetch returns more than requested number of rows"
*Cause:    The number specified in exact fetch is less than the rows returned.
*Action:   Rewrite the query or change number of rows requested
```

I kept getting this error, which indicates that multiple rows were fetched, I checked the logic and it seems fine, only the user who's info was entered as parameter should be returned.

2- Display All Product Names and Prices:

```
create or replace procedure display_all_products
is
v_product_name varchar2(100);
v_price number;
begin
for product_record in (select productname, price from Product1) loop
   DBMS_OUTPUT.PUT_LINE('Product Name: ' || product_record.productname || ' | Price: ' || product_record.price);
end loop;
end display_all_products;
```

```
BEGIN
   display_all_products;
END;
```

```
Procedure DISPLAY_ALL_PRODUCTS compiled

Product Name: Smartphone | Price: 300
Product Name: T-shirt | Price: 20
Product Name: Microwave | Price: 100
Product Name: Novel Book | Price: 15
Product Name: Toy Car | Price: 10


PL/SQL procedure successfully completed.
```

3- Display Products by Product Name:

```
create or replace procedure display_products_by_name(product_name in
varchar2)
is
   v_product_name VARCHAR2(100);
   v_price NUMBER;
begin
   for product_record in (
      select productname, price
      from Product1
      where productname = product_name
   ) loop
      DBMS_OUTPUT.PUT_LINE('Product Name: ' || product_record.productname
|| ' | Price: ' || product_record.price);
   end loop;
END display_products_by_name;
```

```
BEGIN
   display_products_by_name('Smartphone');
END;
```

```
Procedure DISPLAY_PRODUCTS_BY_NAME compiled

Product Name: Smartphone | Price: 300


PL/SQL procedure successfully completed.
```

4- Display Products by Price:

```
create or replace procedure display_products_by_price(min_price in number,
max_price in number)
is
   v_product_name varchar2(100);
   v_price number;
begin
   for product_record in (
      select productname, price
      from Product1
      where price between min_price AND max_price
   ) LOOP
      DBMS_OUTPUT.PUT_LINE('Product Name: ' || product_record.productname
|| ' | Price: ' || product_record.price);
```

```
    END LOOP;
END display_products_by_price;
```

```
BEGIN
    display_products_by_price(12, 200);
END;
```

```
Procedure DISPLAY_PRODUCTS_BY_PRICE compiled

Product Name: T-shirt | Price: 20
Product Name: Microwave | Price: 100
Product Name: Novel Book | Price: 15


PL/SQL procedure successfully completed.
```

5- Display Users with Their Roles

```
create or replace procedure display_users_with_roles
is
    v_fullname varchar2(100);
    v_email varchar2(100);
    v_role_name varchar2(100);
begin
    for user_record in (
        select userinfo1.fullname, userinfo1.email, role1.rolename
        from userinfo1
        inner join role1 on userinfo1.roleid = role1.roleid
    ) LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Full Name: ' || user_record.fullname ||
                         ' | Email: ' || user_record.email ||
                         ' | Role: ' || user_record.rolename);
  END LOOP;
END display_users_with_roles;
```

```
BEGIN
  display_users_with_roles;
END;
```

```
Procedure DISPLAY_USERS_WITH_ROLES compiled

Full Name: John Doe | Email: john@example.com | Role: Customer
Full Name: Jane Smith | Email: jane@example.com | Role: Admin
Full Name: Bob Brown | Email: bob@example.com | Role: Delivery
Full Name: Alice White | Email: alice@example.com | Role: Supplier
Full Name: Charlie Black | Email: charlie@example.com | Role: Manager


PL/SQL procedure successfully completed.
```