I am Rakhimov Assanali 210103203

My ERD has 7 entities with their attributes, 7 entities means 7 tables.

And each table has its own primary key. Such as in Users it is User ID,

in Seller it is Seller ID and so on.

Users are divided into 2, they are Seller and Buyer. Seller sells and Buyer

buys. Also, sellers must sell products and sell at least 1 product. And Buyer

has a shopping cart where his products which he can buy in future. Bayer makes

an order and the order must have a product. Also, one order can have only one

buyer. There is also a bank card where information about the buyer's card is stored.

Primary key - values that are unique for the entire table and values must not be duplicated.

Why do my tables follow all 3 normal forms?

First normal form. In 1NF the table must not contain duplicate rows, no arrays or lists of any kind and the column contains data of the same type.

Second normal form. In 2NF the table must be in first normal form, the table must have a primary key and all non-key columns of the table depend on the primary key.

Third normal form. In 3NF the table must be in second normal form and the requirement of the third normal form is that there is no transitive dependency in the tables.

What is the transitive dependency?

Transitive dependency is when non-key columns depend on the values of other non-key columns.

That is, non-key columns must depend only on the primary key.

All my tables follow normal forms.

Below are my codes:

**Procedure which does group by information**

create PROCEDURE *how\_much* (

tip varchar,

total OUT NUMBER

) AS

BEGIN

Select *count*(USER\_ID) into total from USERS

WHERE USERS.TYPE = tip group by USERS.TYPE;

DBMS\_OUTPUT.*PUT\_LINE*(total || ' ' || tip || 's');

END;

DECLARE

who varchar2(255) := 'Buyer';

total number;

Begin

*how\_much*(who, total);

end;

**Function which counts the number of records**

create function *numbers*(

col varchar

)

return number as

total number;

Begin

select *count*(col) into total from BUYER;

return total;

end;

DECLARE

col varchar2(255) := 'Name';

res number;

Begin

res := *numbers*(col);

DBMS\_OUTPUT.*PUT\_LINE*('The number of records: ' || res);

end;

**Procedure which uses SQL%ROWCOUNT to determine the number of rows affected**

CREATE OR REPLACE PROCEDURE *insert\_buyer* (

b\_id number,

b\_name varchar,

b\_surname varchar,

b\_phone number,

b\_card\_id number,

res out number

) AS

BEGIN

INSERT INTO BUYER VALUES (

b\_id, b\_name, b\_surname, b\_phone, b\_card\_id

);

res := sql%rowcount;

END;

DECLARE

res number;

BEGIN

*insert\_buyer*(51, 'Assanali', 'Rakhimov', 88005553535, 444555666, res);

DBMS\_OUTPUT.*PUT\_LINE*('rows: ' || res);

end;

**Add user-defined exception which disallows to enter title of item to be less than 3 characters**

create or replace trigger short\_name

before insert on PRODUCTS

for each row

BEGIN

if *length*(:NEW.PRODUCT\_NAME) < 3 then

*raise\_application\_error*(-20001, 'Length of name should be longer than 3 symbol');

end if;

end;

**Create a trigger before insert on any entity which will show the current number of rows in the table**

create or replace trigger insert\_to\_users

before insert on USERS

for each row

declare

counting number;

begin

select *count*(USER\_ID) into counting from USERS;

DBMS\_OUTPUT.*PUT\_LINE*('Number of row that already have: ' || counting);

end;