

Ecommerce Database System

Database Management System 2

Prepared by:

Zhanar Mazhit, ID: 210103161

Nurbol Akhmetov, ID: 210101084

Agenda

- 01 Introduction to the System
- 02 ER Diagram
- 03 Normalization in Tables
- 04 Queries

Introduction To the System

Entities: Customer, User_Info, Payment, Shopping_Cart, Orders, Shipping, Rating, Product, Category, Courier

Customer table's attributes: Customer_ID (PK), First_Name, Last_Name, Birth_Date, Email, Gender, Phone_Number.

User_Info table's attributes: Username (PK), Password.

Payment table's attributes: Payment_ID (PK), Customer_ID, IBAN, Provider.

Shopping_Cart table's attributes: Customer_ID (PK), Product_ID, Quantity.

Orders table's attributes: Order_ID (PK), Customer_ID, Date, Product_ID, Quantity, Payment_ID, Status, Shipping_ID, Courier_ID.

Shipping table's attributes: Shipping_ID (PK), Customer_ID, City, Address, Zip_Code.

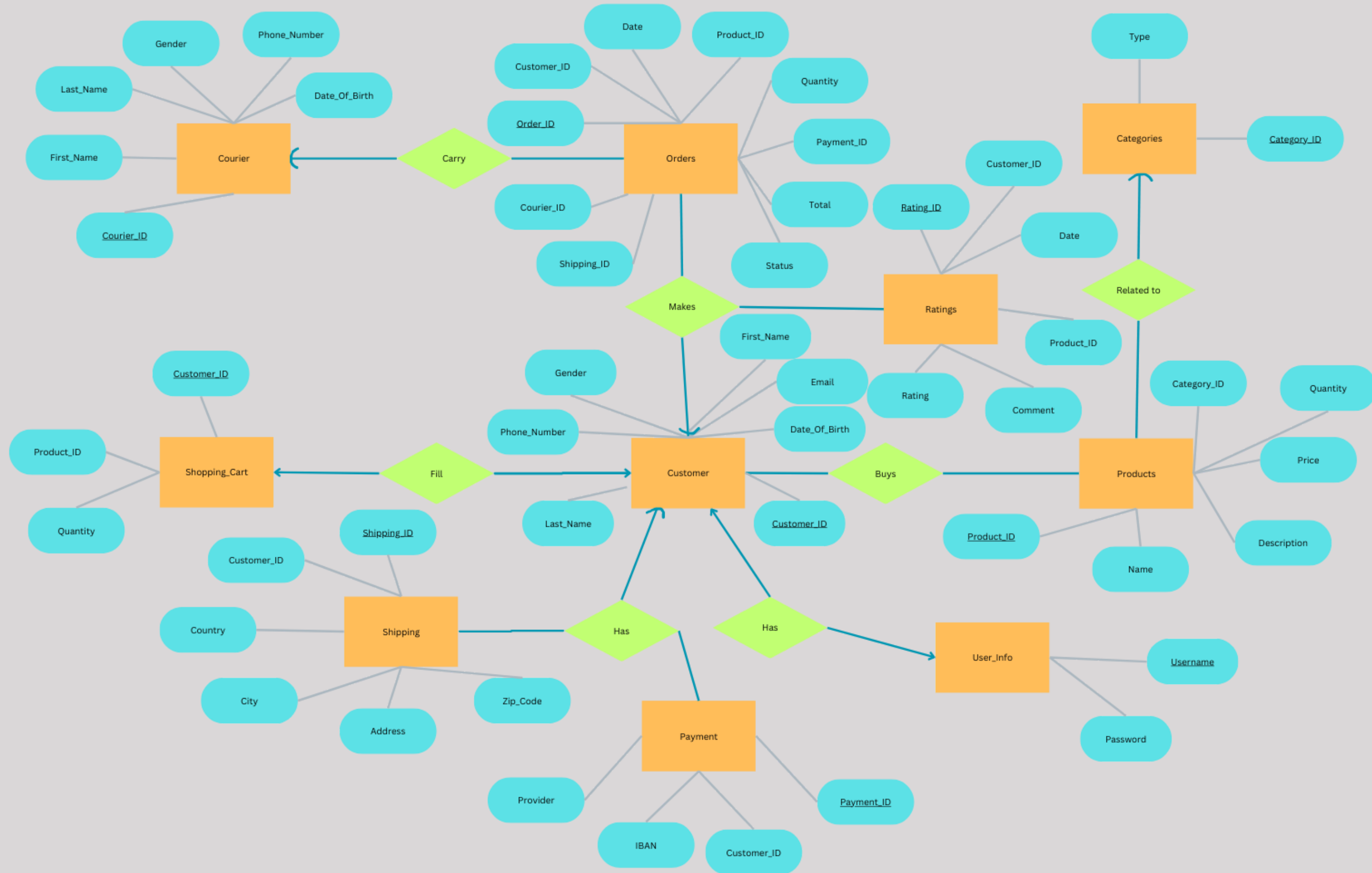
Rating table's attributes: Rating_ID (PK), Customer_ID, Product_ID, Comment_Text, Rating, Date.

Product table's attributes: Product_ID (PK), Name, Description, Price, Category_ID, Quantity.

Category table's attributes: Category_ID (PK), Type.

Courier table's attributes: Courier_ID (PK), First_Name, Last_Name, Gender, Phone_Number, Date_Of_Birth.

ER Diagram



Normalization in Tables

Every entity follows the 1NF because each table cell has exactly one value and each record is unique, meaning that there are no duplicate rows. 2NF rules are followed, as well, because all non-key attributes are fully functional dependent on the primary key which is defined in each entity as ID. And, finally, all tables are in 3NF because there are no transitive functional dependencies. The non-key values in the tables cannot define other non-key values.

Queries

2. Function that counts the number of orders that a particular customer made:

Declaration:

```
CREATE OR REPLACE FUNCTION  
count_orders(p_id IN NUMBER)  
RETURN NUMBER IS
```

```
count_orders INT;
```

```
BEGIN
```

```
SELECT COUNT(*) INTO count_orders  
FROM Orders
```

```
WHERE Customer_ID = p_id;
```

```
RETURN count_orders;
```

```
END;
```

Execution:

```
DECLARE
```

```
c_id INT := 1499397;
```

```
output INT;
```

```
BEGIN
```

```
output := count_orders(c_id);
```

```
dbms_output.put_line(output);
```

```
END;
```

Queries

4. User-defined exception that raises when an entered title's length is less than 5 characters:

DECLARE

product_name Product.Product_Name%TYPE :=
'Mars';

custom_exception EXCEPTION;

BEGIN

IF length(product_name) < 5 THEN

RAISE custom_exception;

END IF;

EXCEPTION

WHEN custom_exception THEN

dbms_output.put_line('Length of product name
must be greater than 5!');

END;

Queries

5. Trigger that shows the quantity of rows in the table after inserting a new row:

Creation of trigger:

```
CREATE OR REPLACE TRIGGER show_row_count
BEFORE INSERT ON Category
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('Current row count: ' ||
SQL%ROWCOUNT);
END;
```

Insertion in the table:

```
INSERT INTO CATEGORY (Category_id) VALUES (12)
```


The background is a solid dark teal color. It features several sets of thin, white, curved lines that flow across the frame. One set of lines starts from the top left and curves towards the right. Another set starts from the bottom left and curves towards the right. A third set starts from the top right and curves towards the bottom right. These lines create a sense of movement and depth.

Thank You