Endterm from DBMS-2

CARS

210103006 Toregul Akzhali
210103206 Baibolat Bekarys
210103264 Tilabek Yerdaulet
210103251Temirkanat Zhandarbek
210103119 Mussafar Mustafa

Presentation structure:

STEP 1

Introduction.

Explain our project, introduce our project

STEP 2

ER diagram

Explain ER diagram, how they are connected to each other.

STEP 3

Normal Forms

Explanation of why the structure follows normal forms

STEP 4

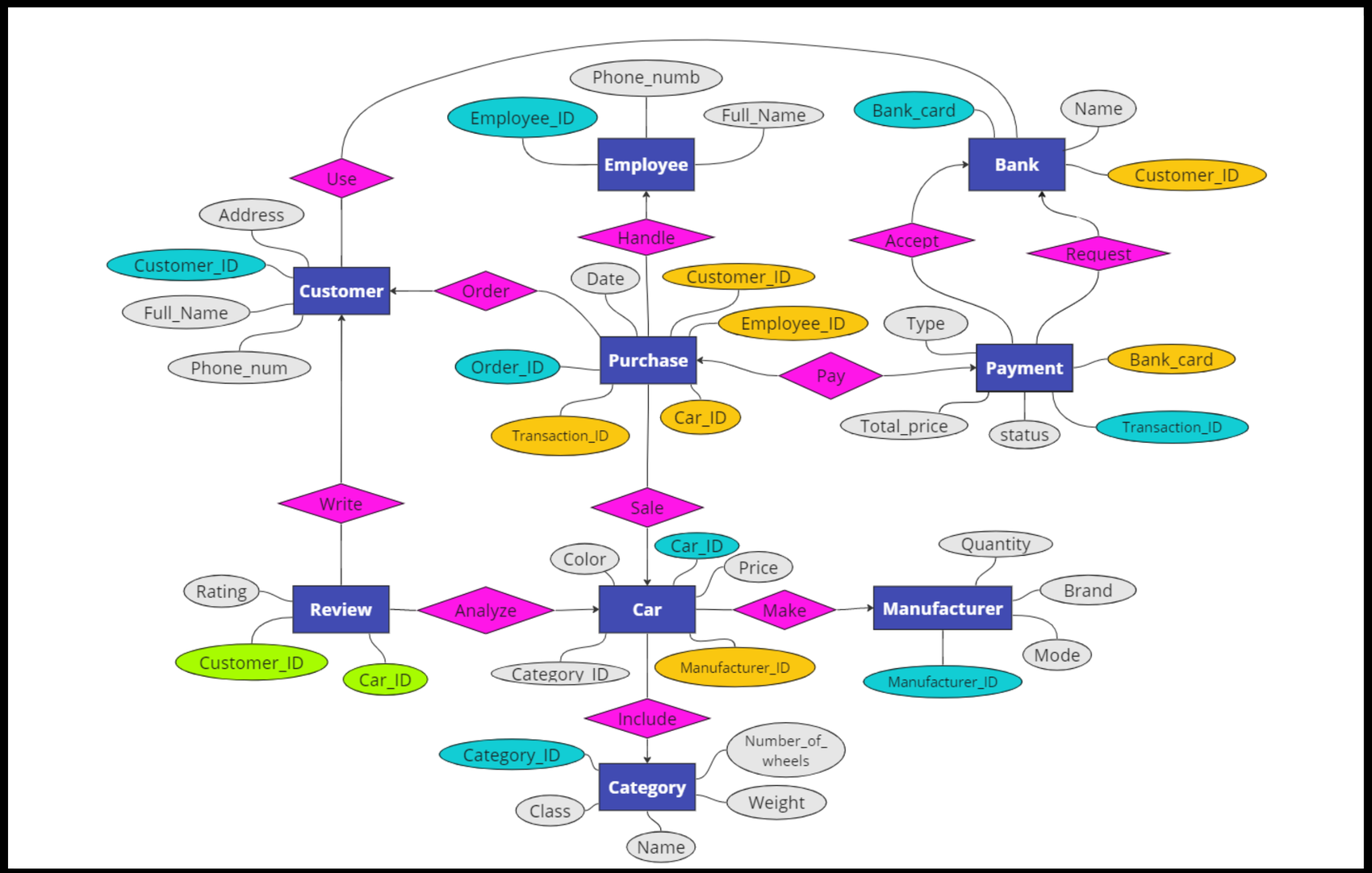
Coding

Explanation and coding part of each item

introduction

The transport industry is developing at the highest level in the world.And now we have developed a project according to which, with the development of online commerce, you can get transport quickly, easily, and conveniently. In addition, each company in the car interior, manufacturers produce different cars.In this project, customers can get the car they need in the desired category, from the right manufacturer, through a bank or without it and write a review and see a rating of car.

In the project created 9 entity ER diagrams, compiled 1, 2, 3 Normal form tables, recorded 1 trigger, 2 procedures, 1 function, 1exception by coding



ERD

Blue is Primary key
Orange is Foreign key
Green is Primary and Foreign key
Indigo is Entity
Pink is Relation

Link to ERd (miro)

https://miro.com/welcomeonboard/WjVydk8yQ0xFOTg2T3NwN0dEN0Q4elVCVVp0THdPRDE1RDhXdmNCRVFKcnpxNUlBd3ZsYVZrQXp0aFIyaXJ1dnwzNDU4NzY0NTQxMTg2MzY4Njk1fDI=?share_link_id=536641103717

Tables Columns

Car (Car_ID, Color, Price, Cotegory_ID, Manufacturer_ID)

Review (Car_ID, Customer_ID, Rating)

Purchase (Order_ID, Customer_ID, Employee_ID, Transaction_ID, Car_ID, Date)

Payment (Transaction_ID, Card_number, Total_price, Type, Status)

Employee (Employee_ID, Phone_number, Full_name)

Customer (Customer_ID, Phone_number, Full_name, Address)

Category (Category_ID, Class, Weight, Number_of_wheels, Name)

Manufacturer (Manufacturer_ID, Model, Quantity, Brand)

Bank (Bank_card, Name, Customer_ID)

Tables Relations

1) Relation(handle) between Purchase and Employee (many-to-one)
2) Relation(pay) between Purchase and Payment (one-to-one)
3) Relation(accept,request) between Payment and Bank (many-to-one)
4) Relation(sale) between Purchase and Car (many-to-one)
5) Relation(make) between Car and Manufacturer (many-to-one)
6) Relation(include) between Car and Category (many-to-one)
7) Relation(order) between Purchase and Customer (many-to-one)
8) Relation(write) between Customer and Review (one-to-many)
9) Relation(analyze) between Review and Car (many-to-one)
10) Relation(use) between Bank and Customer (many-to-many)

Normal Form

Explanation of why the structure follows normal forms

1NF

1NF IS A PROPERTY OF A RELATION IN A RELATIONAL DATABASE . A RELATION IS IN THE FIRST NORMAL FORM IF AND ONLY IF NO ATTRIBUTE DOMAIN HAS RELATIONS AS ELEMENTS. OR, MORE INFORMALLY, NO TABLE COLUMN CAN HAVE TABLES AS VALUES

2NF

FOR A TABLE TO BE IN 2NF, THERE ARE TWO REQUIREMENTS
-THE DATABASE IS IN FIRST NORMAL FORM
-ALL NONKEY ATTRIBUTES IN THE TABLE MUST BE FUNCTIONALLY DEPENDENT ON THE ENTIRE PRIMARY KEY

3NF

A TABLE SATISFIES 3NF, IF AND ONLY IF
FOR ANY NON-TRIVIAL X->Y
-EITHER X IS A SUPERKEY
-OR EACH ATTRIBUTE IN Y IS CONTAINED IN A KEY

Our tables are in 1, 2, 3NF. Because there is one information in each row. And each non-key attribute is functionally dependent on the key. And the non-key attribute is non-transitively dependent on the primary key

1)CAR

{Car_id} -> {color, price, category_id, manufacturer_id}

CAR_ID	COLOR	PRICE	CATEGORY_ID	MANUFACTURER_ID
1	White	500000	2	2
2	Black	650000	1	1
3	Gray	500000	3	3
4	Dark Blue	1500000	5	5

2)CATEGORY

{Category_id} -> {class, weight, number_of_wheels, name}

CATEGORY_ID	CLASS	WEIGHT	NUMBER_OF_WHEELS	NAME
1	A	400	4	Sedan
2	B	600	4	Minivan
3	D	400	4	Bus
4	E	1000	8	Truck

3)CUSTOMER

{Customer_id} -> {phone_number, full_name, address}

CUSTOMER_ID	PHONE_NUMBER	FULL_NAME	ADDRESS
1	87004554545	Onaibaev Dias	Tole bi 12
2	87024945854	Karasayev Dias	Vernye 21
3	87013744957	Tagybai Aset	Samuryk 74
4	87022945930	Kozy-Korpesh Bekzat	Satbaev 22

4)EMPLOYEE

{Employee_id} -> {phone_number, full_name}

EMPLOYEE_ID	PHONE_NUMBER	FULL_NAME
1	87021495824	Temirkanat Zhandarbek
2	87006647896	Toregul Akzhali
3	87078737458	Baibolat Bekarys
4	87026005121	Tilebek Erdaulet

5)MANUFACTURER

{Manufacturer_id} -> {model, quantity, brand}

MANUFACTURER_ID	MODEL	QUANTITY	BRAND
1	Prada-200	130	Toyota
2	Prada-300	145	Toyota
3	E221	150	Mercedes
4	M5	125	BMW

6)PURCHASE

{Oreder_id} -> {customer_id,employee_id, transaction_id, car_id, date}

OREDER_ID	CUSTOMER_ID	EMPLOYEE_ID	TRANSACTION_ID	CAR_ID	DATE
1	2	2	402	1	04/20/2023
2	1	1	401	2	07/07/2022
3	3	3	403	3	11/21/2021
4	4	4	404	4	02/01/2019

7)Review

{Customer_id, Car_id} -> {rating}

CAR_ID	CUSTOMER_ID	RATING
1	1	50
2	2	90
3	3	100
4	4	70

8)PAYMENT

{Transaction_id} -> {bank_card, total_price, type, status}

TRANSACTION_ID	BANK_CARD	TOTAL_PRICE	TYPE	STATUS
401	50	1000000	bank_card	accept
402		1200000	cash	
403	52	1500000	bank_card	declained
404	53	900000	bank_card	acceprt

9)BANK

{Bank_card} -> {name}

BANK_CARD	NAME	CUSTOMER_ID
50	halyk	1
51	kaspi	4
52	otbasy	11
53	red	3

--Procedure which does group by information

Language PL/SQL Rows 10 Clear Command Find Tables

```
--GROUP BY procedure: create or replace
Create or Replace Procedure group_by_Procedure (
    tbl in varchar2,
    col in varchar2,
    counter out number
)
As
    sql_query varchar2(2000);
Begin
    sql_query := 'Select Count(*) From (Select Count(*) From ' || tbl || ' Group by ' || col ||')';
    Execute Immediate sql_query into counter;
    dbms_output.put_line('The number of ' || col || 's in table ' || tbl || ' is equal to: ' || counter);
End;

--declare
Declare
    counter number;
    tbl varchar2(100);
    col varchar2(100);
Begin
    tbl:='bank';
    col:='name';
    group_by_Procedure(tbl, col, counter);
End;
```

Results Explain Describe Saved SQL History

```
The number of names in table bank is equal to: 6

Statement processed.
```

--Function which counts the number of records

Language PL/SQL Rows 10 Clear Command Find Tables

```
Create or Replace Function count_records_Function (tbl in varchar2)
   return number is
   counter number;
   sql_query varchar2(2000);
Begin
   sql_query := 'Select Count(*) From ' || tbl;
   Execute Immediate sql_query into counter;
   return counter;
End;

--declare
Declare
    counter number;
    tbl varchar2(100);
Begin
   tbl:='bank';
   counter:= count_records_Function(tbl);
   dbms_output.put_line('The number of records in table ' || tbl || ' is equal to: ' || counter);
End;
```

Results Explain Describe Saved SQL History

```
The number of records in table bank is equal to: 20

Statement processed.
```

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

Language PL/SQL Rows 10 Clear Command Find Tables

```
Create or Replace Procedure delete_multiple_rows_ROWCOUNT(
   cond in varchar2,
   tbl in varchar2
) is
   counter number;
   sql_query varchar2(2000);
Begin
   sql_query := 'Delete From ' || tbl || ' WHERE ' || cond ;
   Execute Immediate sql_query;
   counter := SQL%ROWCOUNT;
   dbms_output.put_line('Rows affected: ' || counter);
End;


--declare
Begin
   delete_multiple_rows_ROWCOUNT('rating = 80', 'review');
End;
```

Results Explain Describe Saved SQL History

```
Rows affected: 2

Statement processed.
```

210103206@stu.sdu.edu.kz olmesjan en

--Add user-defined exception which disallows to enter title of item (e.g. book) to be less than 5 characters

Language PL/SQL Rows 10 Clear Command Find Tables

```
Create or Replace Procedure new_Employee(
   employee_id_1 in number,
   phone_number_1 in number,
   full_name_1 in varchar2
)  AS
   sql_query varchar2(2000);
   ex_invalid Exception;
Begin
If Length(full_name_1) < 5 Then
   Raise ex_invalid;
Else
   sql_query := 'Insert into Employee(employee_id, phone_number, full_name)
   values('||employee_id_1||', '''||phone_number_1||''', '||full_name_1||')';
   Execute Immediate sql_query;
End If;
   Exception When ex_invalid Then
   dbms_output.put_line('The number of characters in full_name is not enough to be required');
End;


--declare
Begin
   new_Employee(20, 87056650001, 'Asan');
End;
```

Results Explain Describe Saved SQL History

```
The number of characters in full_name is not enough to be required

Statement processed.
```

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

Language PL/SQL Rows 10 Clear Command Find Tables

```
Create or Replace Trigger count_rows_Trigger
Before Insert On Review
For Each Row
Declare
   counter number;
Begin
   Select Count(*) into counter From Review;
   dbms_output.put_line('The number of rows before inserting new value is equal to: '||counter);
End;

Begin
   Insert into Review(Car_id, Customer_id, Rating) Values (9, 16, 55);
End;
```

Results Explain Describe Saved SQL History

```
The number of rows before inserting new value is equal to: 16

1 row(s) inserted.
```

Conclusion

The transportation management system is an innovative project that offers fast and reliable transportation services. It includes nine entities with unique attributes and is designed using Normal Forms tables to ensure efficient data management. The project uses automated triggers, procedures, functions, and exceptions to provide a streamlined approach to transportation management, setting new standards for the industry.

THANKS FOR YOUR ATTENTION