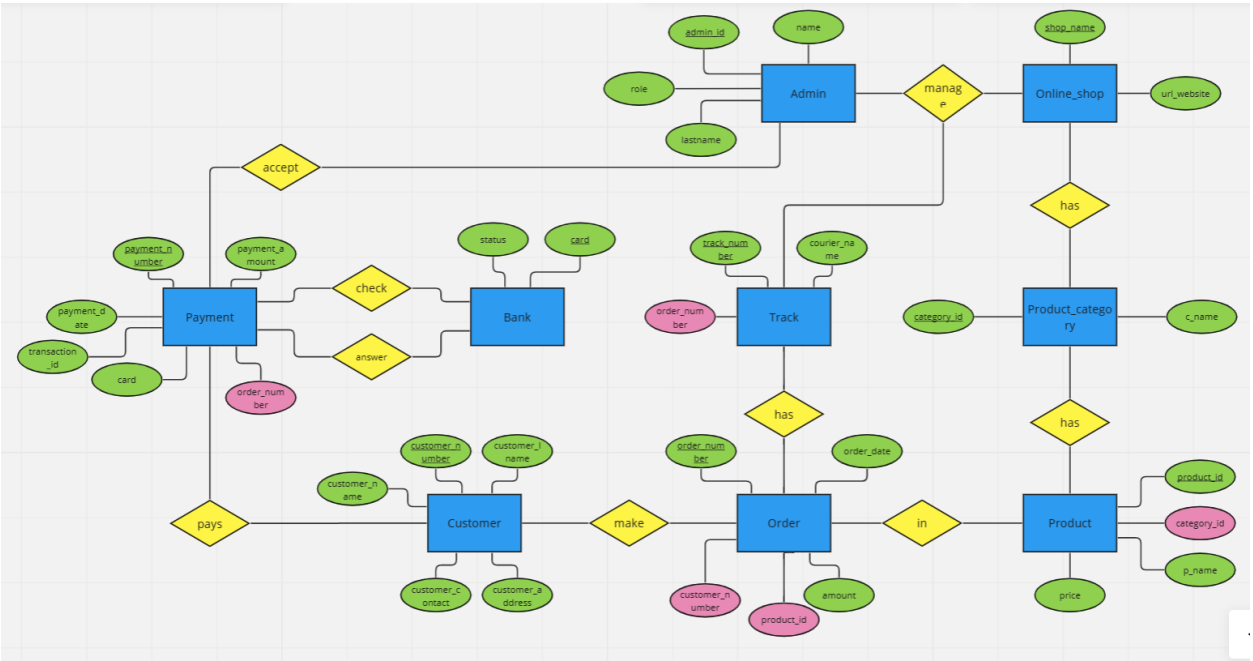
Introduction :

Online book store . In our project , we made an analogue of an online book store . How our system works . First of all , the customer enters our website and chooses a genre later , makes an order from books . After that , he makes the payment , after the transaction is completed and confirmation from the bank comes than Admins approve and send the order to the courier .

Since we still have a small shop . There are no employees , we work for ourselves , admins are engaged in all the business of the store.

In our system, data is stored in 9 tables. They are (Admin , Online_shop , Product_category , Product , Order, Customer , Track , Bank , Payment).

In the next turn, let's get acquainted with the relationships between the tables.



The blue quadrilateral depicted here is entity , and the yellow diamonds are the relationship between them , and the military circles are attributes , and I marked the pink ones like that foreign keys!!!

1.Relationship between Admin and Online_shop 1:1

why 1:1 . because only one person can manage one store at a time, just like one store can have only one admin

2. Relationship between Online_shop and Product_category 1:M

Because one store can have many categories

3. Relationship between Product_category and Product 1:M

Because in one category we have a lot of products

4. Relationship between Product and Order M:1

In one order we can have the lot of products

5. Relationship between Customer and Order 1:M

One customer can do lot of orders

6. Relationship between Customer and Payment 1:M

One customer can do many pays

7. Relationship between Payment and Bank M:M

lot of cards can do lot of trans actions so bank can check lot of transactions

8. Relationship between Order and Track 1:1

One order can ACCEPTED ONLY ONE TRACK NUMBER SO IT IS WILL BE 1:1

9. Relationship between Admin and Payment 1:M

One admin can checked and do transport lot of orders and payment

10 Relationship between Admin and Track 1:M

One admin can checked and do transport og products

1)

admin_id	name	lastname	role
1	Gaukhar	Bazarbayeva	manage
2	Aidana	Kuanova	manage

admin_id ➞ name

admin_id ➞ lastname

admin_id ➞ role

admin_id ➞ admin_id, name, lastname, role

2)

shop_name	url_website
booktopia	http://booktopia.com

shop_name ➞ url_website

3)

category_id	c_name
1101	fantasy
1102	drama
1103	horror
1104	comedy
1105	detective

category_id ➞ c_name

4)

category_id	product_id	p_name	price
1101	110167	Six of Crows	4500
1101	110124	Stardust	7600
1101	110118	The Last Unicorn	3800
1101	110143	The Fifth Season	5500
1101	110139	Kindred	8700

category_id	product_id
1101	110167
1101	110124
1101	110118
1101	110143
1101	110139

product_id	p_name	price
110167	Six of Crows	4500
110124	Stardust	7600
110118	The Last Unicorn	3800
110143	The Fifth Season	5500
110139	Kindred	8700

category_id ➟ product_id

product_id ➟ p_name

product_id ➟ price

product_id ➟ p_name, price

5)

customer_number	customer_name	customer_lname	customer_contact	customer_address
23456	Amelia	Carter	87013492034	California 31
23345	Vincent	Watting	87786423190	Texas 20
23445	Frazer	Masselin	87012319857	Florida 97
44533	Denise	Avon	87752973081	New York 12
23521	Shelly	Minard	87772083429	Pennsylvania 67

customer_number ➟ customer_name

customer_number ➟ customer_lname

customer_number ➟ customer_contact

customer_number ➟ customer_address

customer_number ➟ customer_name, customer_lname, customer_contact, customer_address

6)

order_number	customer_number	amount	order_date	product_id
21003	23456	4500	11 april 12:20	110167
21004	23345	7600	11 april 15:00	110124
21005	23445	3800	12 april 10:30	110118
21006	44533	5500	12 april 13:45	110143
21007	23521	8700	13 april 16:05	110139

order_number ➡ customer_number

order_number ➡ amount

order_number ➡ order_date

order_number ➡ product_id

order_number ➡ customer_number, amount, order_date, product_id

payment_number	order_number	payment_amount	payment_date	transaction_id	card
421	21003	4500	11 april 12:20	308	4 400 430 108 837 880
422	21004	7600	11 april 15:00	619	4 400 430 114 680 750
423	21005	3800	12 april 10:30	372	4 400 430 169 363 060

payment_number	order_number	payment_amount	payment_date	transaction_id
421	21003	4500	11 april 12:20	308
422	21004	7600	11 april 15:00	619
423	21005	3800	12 april 10:30	372

transaction_id	card
308	4 400 430 108 837 880
619	4 400 430 114 680 750
372	4 400 430 169 363 060

payment_number ➡ order_number

payment_number ➡ payment_amount

payment_number ➡ payment_date

payment_number ➡ transaction_id

payment_number ➡ order_number, payment_amount, payment_date, transaction_id

transaction_id ➡ card

1.Functions :

```
CREATE OR REPLACE FUNCTION number_of_category_product(c_id IN NUMBER)
RETURN NUMBER
IS num_product NUMBER;
BEGIN
 SELECT COUNT(*) INTO num_product FROM PRODUCT WHERE category_id = c_id;
Return num_product ;
END;
```

```
Declare
num_product NUMBER ;
Begin
num_product := number_of_category_product(1102);
dbms_output.put_line(num_product || ' Products we have in category 1102');
END ;
```

EXSEPTION :

```
DECLARE
num NUMBER := :num;
or_num orderr.order_number%type;
amount_sum orderr.amount%type ;
invalid_number_order EXCEPTION;
BEGIN
IF LENGTH(num) > 5 THEN
RAISE invalid_number_order;
ELse
 SELECT order_number, amount INTO or_num, amount_sum
FROM orderr
WHERE order_number = num;
 DBMS_OUTPUT.PUT_LINE ('Order number is' || or_num || ' amount is ' || amount_sum);
```

```
END IF;

EXCEPTION

WHEN invalid_number_order THEN

DBMS_OUTPUT.PUT_LINE('Invalid order name .');

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

END;
```

Trigger

```
CREATE OR REPLACE TRIGGER currrent_rowse
BEFORE INSERT ON any_table
FOR EACH ROW
declare
row_insert NUMBER ;
BEGIN
Select Count(*) into row_insert From Admin;
DBMS_OUTPUT.PUT_LINE(row_insert || ' its the number of rows.');

END;



Begin
Insert into Admin(admin_id , name , lastname , role) values(6 , 'Gna' , 'Oasarova' , 'anage');
END ;
```

Procedures :

```
CREATE OR REPLACE PROCEDURE determine_rowss( c_number IN NUMBER , status Varchar2)
IS
row_insert INTEGER;
BEGIN
Insert Into BANK(card , status) values(c_number , status) ;
row_insert := SQL%ROWCOUNT;
DBMS_OUTPUT.PUT_LINE(row_insert || ' row inserted.');
END;

Begin
determine_rowss(4400225563528854 , 'accept');
END ;
```

Procedure 2

```
CREATE OR REPLACE PROCEDURE group_by_Column (
column_name IN VARCHAR2
)
```

```
IS
BEGIN
FOR row IN (
SELECT column_name, COUNT(*) AS row_count
FROM Customer
GROUP BY column_name
) LOOP
DBMS_OUTPUT.PUT_LINE('Column value: '  row.column_name  ', Row count: ' || row.row_count);
END LOOP;
END;



BEGIN
group_by_Column('Carter');
END ;
```

There will be a full explanation in the video(we apologize, I hope you will understand))))