NAME : OMKAR G. SALVI

GUIDED BY : SAMEER WARSOLKAR SIR

ECOMMERCE ORDERS

SQL PROJECT

om.salvi44@gmail.com

BATCH : DS T-326

ABSTRACT

Ecommerce sales datasets are a valuable resource for analyzing consumer behavior, market trends, and business performance. These datasets typically contain information such as product details, customer demographics, purchase history, and transaction data. By studying these datasets, researchers and businesses can gain insights into customer preferences, identify sales patterns, optimize marketing strategies, and make data-driven decisions to improve profitability.

DESCRIPTION

This database schema is designed to serve as the backend storage for a web application managing an e-commerce platform. By storing data in a relational database, various tasks related to the platform's operations can be performed efficiently and accurately. Some of the benefits of using a database over traditional methods include:

- **Simplified data management :** Updating and modifying product lists, customer details, and other information becomes easier.
- **Automated calculations :** Tasks like calculating due payments can be automated, reducing human error.
- **Enhanced data analysis :** Relational databases provide tools for analyzing data, aiding in decision-making regarding inventory management and other aspects of the e-commerce platform.

Database Tables :-

- **Orders Table :** Contains information about individual orders, such as order ID, order date, customer ID, shipping address, and order status.
- **Order Items Table :** Stores details about the items included in each order, including order ID, product ID, quantity, price, and item status.
- **Customers Table :** Contains information about customers, such as customer ID, name, email address, shipping address, and billing address.
- **Payments Table :** Stores details about payments made for orders, including payment ID, order ID, payment method, payment amount, and payment date.
- **Products Table :** Contains information about products sold on the platform, such as product ID, product name, description, price, quantity in stock, and product category.

ENTITY RELATIONSHIP DIAGRAM (ERD)

The ERD visually represents how these tables are related to each other. For example, an order can have multipleorder items, and a customer can have multiple orders.

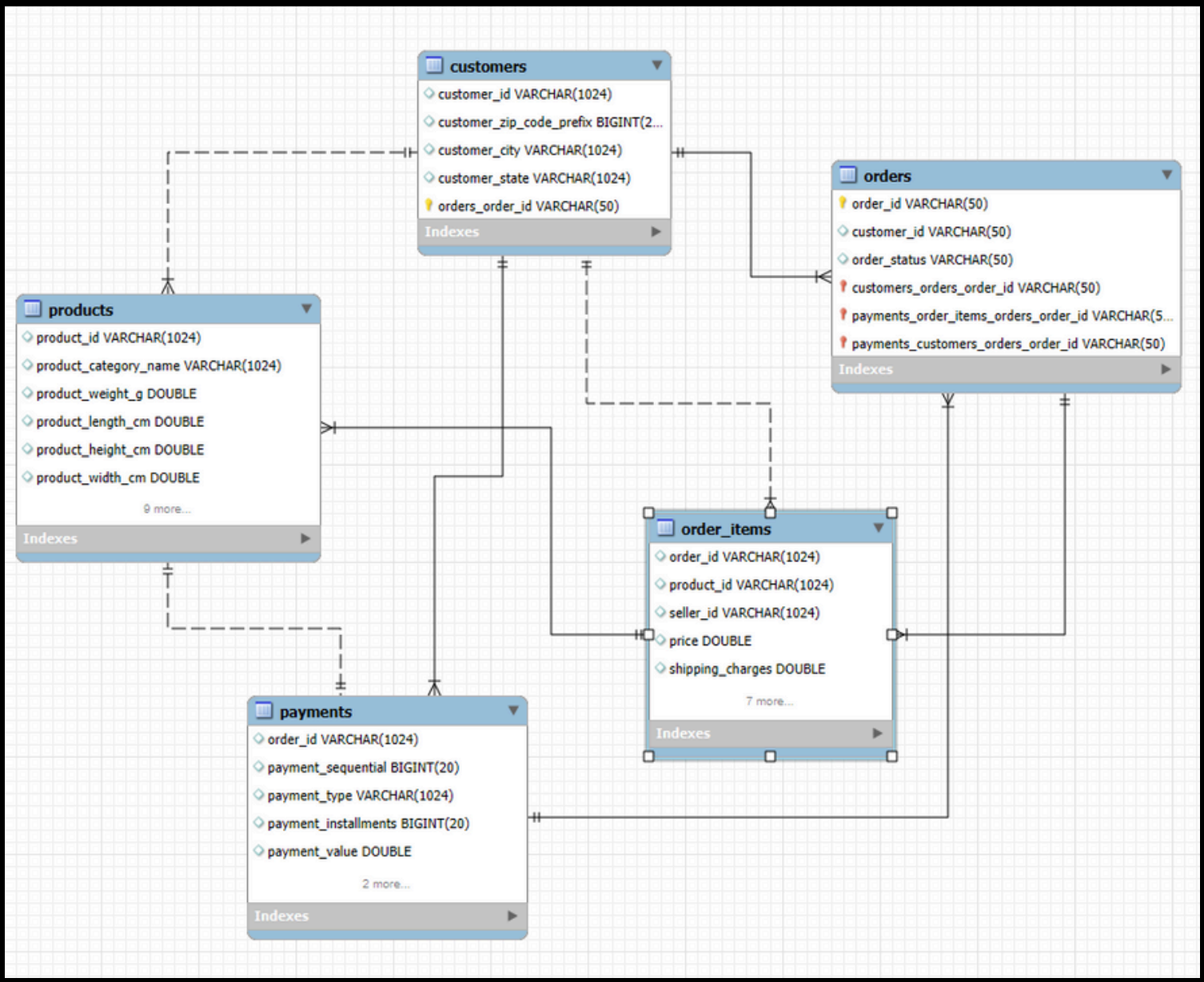


TABLE DESCRIPTION

1. CUSTOMERS :

Field	Type	Null	Key	Default	Extra
customer_id	varchar(1024)	YES		NULL	
customer_zip_code_prefix	bigint(20)	YES		NULL	
customer_city	varchar(1024)	YES		NULL	
customer_state	varchar(1024)	YES		NULL	

2. ORDER_ITEMS :

Field	Type	Null	Key	Default	Extra
order_id	varchar(1024)	YES		NULL	
product_id	varchar(1024)	YES		NULL	
seller_id	varchar(1024)	YES		NULL	
price	double	YES		NULL	
shipping_charges	double	YES		NULL	

3. PAYMENTS :

Field	Type	Null	Key	Default
order_id	varchar(1024)	YES		NULL
payment_sequential	bigint(20)	YES		NULL
payment_type	varchar(1024)	YES		NULL
payment_installments	bigint(20)	YES		NULL
payment_value	double	YES		NULL

4. PRODUCTS :

	Field	Type	Null	Key	Default
▶	product_id	varchar(1024)	YES		NULL
	product_category_name	varchar(1024)	YES		NULL
	product_weight_g	double	YES		NULL
	product_length_cm	double	YES		NULL
	product_height_cm	double	YES		NULL
	product_width_cm	double	YES		NULL

5. ORDERS :

	Field	Type	Null	Key	Default	Extra
▶	order_id	varchar(50)	NO	PRI	NULL	
	customer_id	varchar(50)	YES		NULL	
	order_status	varchar(50)	YES		NULL	

CREATING DATABASE , TABLES & INSERTING VALUES

```
Create database Ecommerce ;
use Ecommerce ;
```

Creating Table Customers :

```
CREATE  TABLE  `Customers`  (
    `customer_id` VARCHAR(1024),
    `customer_zip_code_prefix` BIGINT,
   `customer_city` VARCHAR(1024),
   `customer_state` VARCHAR(1024)
    );
```

Inserting Values into Customers :

```
iNSERT  INTO  `Customers`
(`customer_id`,`customer_zip_code_prefix` , `customer_city` , `custo
mer_state` )
 VALUES
( 'hCT0x9JiGXBQ' , 58125 , 'varzea paulista' , 'SP' ),
( 'PxA7fv9spyhx' , 3112 , 'armacao dos buzios' , 'RJ' ),
( 'g3nXeJkGI0Qw' , 4119 , 'jandira' , 'SP' ),
( 'EOEsCQ6QIpIg' , 18212 ,'uberlandia' , 'MG' ),
( 'mVz5LO2Vd6cL' , 88868 , 'ilhabela' , 'SP' ),
.....
```

Creating Table Order_items :

```
CREATE TABLE `Order_items` (
    `order_id` VARCHAR(1024),
    `product_id` VARCHAR(1024),
    `seller_id` VARCHAR(1024),
    `price` DOUBLE,
    `shipping_charges` DOUBLE
    );
```

Inserting Values into Customers :

```
INSERT INTO `Order_items` ( `order_id` , `product_id` , `seller_id` ,
`price` , `shipping_charges` )
VALUES
( 'Axfy13Hk4PIk' , '90K0C1fIyQUf' , 'ZWM05J9LcBSF' , 223.51,84.65),
( 'v6px92oS8cLG' , 'qejhpMGGVcsl' , 'IjIpYfhUbRQs' , 170.8,23.79),
( 'UIpf9skrhjfm' , 'qUS5d2pEAyxJ' , '77p2EYxcM9MD' ,64.4,17.38),
( 'bwJVWupf2keN' , '639iGvMyv0De' , 'jWzS0ayv9TGf' ,264.5,30.72),
( 'Dd0QnrMk9Cj5' , '1IycYGcsic2F' , 'I1pYW6GBnPMr' , 779.9,30.66),
.....
```

Creating Table Payments :

```
CREATE TABLE `Payments` (
    `order_id` VARCHAR(1024),
    `payment_sequential` BIGINT,
    `payment_type` VARCHAR(1024),
    `payment_installments` BIGINT,
    `payment_value` DOUBLE
    );
```

Inserting Values into Payments :

```
INSERT INTO `Payments`
(`order_id`,`payment_sequential`,`payment_type`,`payment_installments`,`payment_value`)
 VALUES
( 'Axfy13Hk4Plk' , 1 , 'credit_card' , 1 , 259.14),
( 'v6px92oS8cLG' , 1 , 'credit_card' , 8 , 382.39),
( 'Ulpf9skrhjfm' , 1 , 'credit_card' , 4 , 249.25),
( 'bwJVWupf2keN' , 1 , 'credit_card' , 2 , 27.79),
( 'Dd0QnrMk9Cj5' , 1 , 'credit_card' , 1 , 76.15),
.....
```

Creating Table Products :

```
CREATE  TABLE  `Products`  (
    `product_id` VARCHAR(1024),
    `product_category_name` VARCHAR(1024) NULL,
    `product_weight_g` DOUBLE NULL,
    `product_length_cm` DOUBLE NULL,
    `product_height_cm` DOUBLE NULL,
    `product_width_cm` DOUBLE NULL
     );
```

Inserting Values into Products :

```
INSERT  INTO  `Products`
(`product_id`,`product_category_name`,`product_weight_g`,`produc
t_length_cm`,`product_height_cm`,`product_width_cm`)
 VALUES
( '90K0C1fIyQUf' , 'toys' , 491 , 19 , 12 , 16 ),
( 'qejhpMGGVcsl' , 'watches_gifts' , 440 , 18 , 14 , 17 ),
( 'qUS5d2pEAyxJ' , 'costruction_tools_garden' , 2200 , 16 , 16 , 16 ),
( '639iGvMyv0De' , 'toys' , 1450, 68 , 3 , 48 ),
( '1lycYGcsic2F' , 'toys' , 300 , 17 , 4 , 12 ),
.....
```

Creating Table Orders :

```
CREATE TABLE Orders (
    ‘order_id’ VARCHAR(50) PRIMARY KEY,
    ‘customer_id’ VARCHAR(50),
    ‘order_status’ VARCHAR(50)
    );
```

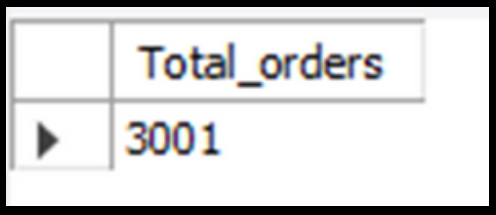
Inserting Values into Orders :

```
INSERT INTO orders (order_id , customer_id , order_status )
VALUES
( 'Axfy13Hk4PIk', 'hCT0x9JiGXBQ' , 'delivered' ),
( 'v6px92oS8cLG', 'PxA7fv9spyhx' , 'delivered' ),
( 'Ulpf9skrhjfm', 'g3nXeJkGI0Qw' , 'delivered' ),
( 'bwJVWupf2keN', 'EOEsCQ6QlpIg' , 'delivered' ),
( 'Dd0QnrMk9Cj5' , 'mVz5LO2Vd6cL' , 'delivered' ),
 ..............
```

QUERIES

1) Count the total number of orders.

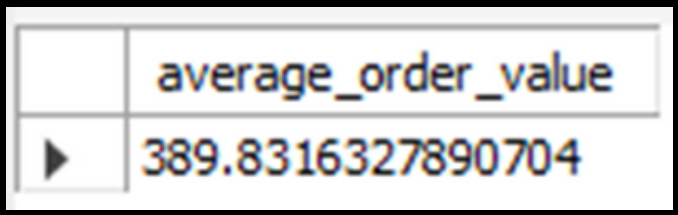
```
select count(*) as Total_orders
from orders ;
```



	Total_orders
▶	3001

2) Find the average order value.

```
select avg(total_price) as average_order_value
from (
        select order_id , sum(price+shipping_charges) as total_price
        from order_items
        group by order_id
        ) as Order_total;
```



	average_order_value
▶	389.8316327890704

3) List the top 5 customers by total orders placed.

```
select c.customer_id, count(*) as Total_orders
from customers c
join orders o on c.customer_id = o.customer_id
group by c.customer_id
order by Total_orders
desc
Limit 5;
```

customer_id	Total_orders
hCT0x9JiGXBQ	2
PxA7fv9spyhx	2
g3nXeJkGI0Qw	2
EOEsCQ6QlpIg	2
mVz5LO2Vd6cL	2

4) List the top 5 customers by total orders placed.

```
select payment_type , count(*) as Total_payments
from payments
group by payment_type
order by Total_payments desc
limit 1;
```

payment_type	Total_payments
credit_card	2217

5) Calculate the total revenue generated by each product category.

```
select p.product_category_name, sum(oi.price +
oi.shipping_charges) as Total_revenue
from products p
join Order_items oi on p.product_id = oi.product_id
group by p.product_category_name;
```

product_category_name	Total_revenue
agro_industry_and_commerce	188.82
air_conditioning	500.24
audio	1491.4299999999998
auto	17970.749999999993
baby	11298.289999999997
bed_bath_table	32610.56
books_general_interest	926.6200000000001
books_imported	277.08

6) Determine the percentage of orders that were cancelled.

```
select count(case when order_status = 'canceled' then 1 end) /
count(*) * 100 as
cancellation rate
from orders;
```

cancellation_rate
0.3999

7) Find the top 10 costly products.

```
select p.product_id, sum(oi.price) as price
from products p
join order_items oi on p.product_id = oi.product_id
group by p.product_id
order by price desc
limit 10;
```

product_id	price
9NwzO0Pm0fDM	417576.87999999966
GvBzGCvvIC2D	184800
ro08JPncYzLh	105443.82000000005
Biwi1BNtUB7l	86589.58000000007
SLTlrWtcYt1m	73212.93
k1G4RHt3R7Nk	53676
CpeCg2UGCL6P	52487.2
CY8OZ3lT9uqB	46384
0vbEvli2JYJu	39405.44999999993
5J7az1rwth4i	27351.000000000015

8) Find the total number of payments made for each order.

```
select o.order_id, COUNT(*) as total_payments
from Orders o
join Payments p on o.order_id = p.order_id
group by o.order_id;
```

order_id	total_payments
00NEIdaOTIgn	1
019iNT2eSEYM	1
05Tyg0HKJJgw	1
072xMq8A0d39	1
07amSeyheu9N	1
07j6mlMhSVtx	1
08dA020btxYK	1

9) Product purchased category name by a specific customer.

```
select p.product_category_name
from Products p
join Order_Items oi on p.product_id = oi.product_id
join Orders o on oi.order_id = o.order_id
where o.customer_id = "PxA7fv9spyhx";
```

product_category_name
watches_gifts

10) Calculate the total revenue generated by each state.

```
select c.customer_state, SUM(oi.price + oi.shipping_charges) as
total_revenue
from Customers c
join Order_Items oi on oi.order_id = oi.order_id
group by c.customer_state;
```

customer_state	total_revenue
AL	30417002.98000056
AM	9359077.840000045
AP	4679538.9199999785
BA	255034871.14002442
CE	79552161.63999416
DF	128687320.30000873
ES	170803170.58001447
GO	142725937.06001115
MA	49135158.659996025
MG	898471472.6394455
MS	58494236.49999326
MT	65513544.87999118
PA	46795389.19999672
PB	42115850.2799981
PE	128687320.30000873
PI	39776080.81999879
PR	346285880.0800262
RJ	925378821.4294021
RN	28077233.52000049
RO	21057925.140000287
RS	342776225.89002615
SC	270243372.63002497
SE	25737464.060000423
SP	2881426089.9956026
TO	18718155.68000022

11) Find the average order value for each payment type

```
select p.payment_type, avg(total_price) as average_order_value
from Payments p
join (
select order_id, SUM(price + shipping_charges) as  total_price
from Order_Items
group by order_id
 ) as order_totals on p.order_id = order_totals.order_id
group by p.payment_type;
```

payment_type	average_order_value
credit_card	377.2641993685155
debit_card	416.83782608695657
voucher	360.39382716049374
wallet	444.3257986111113

12) Find the average order value for each payment type.

```
select p.product_category_name, COUNT(*) as total_orders
from Products p
join Order_Items oi on p.product_id = oi.product_id
group by p.product_category_name
order by total_orders desc
limit 1;
```

	product_category_name	total_orders
▶	toys	4407

13) Identify the top 5 customers with the highest total spending.

```
select c.customer_id, SUM(oi.price + oi.shipping_charges) as
total_spending
from Customers c
join Orders o on c.customer_id = o.customer_id
join Order_Items oi on o.order_id = oi.order_id
group by c.customer_id
order by total_spending desc
limit 5;
```

	customer_id	total_spending
▶	DqV4ZWYBoeIh	9196.52
	WHzroOozslNO	8869.92
	R6ERhHMLglNA	8369.279999999999
	lALEMQErMiPY	8334.699999999999
	cNow1DF5YxjX	8328.16

14) Find the total number of products sold by top 10 seller

```
select seller_id, COUNT(*) AS total_products_sold
from order_items
group by seller_id
order by total_products_sold desc
limit 10 ;
```

seller_id	total_products_sold
RKad98cTxhSb	57
ruq0u2ZpAMDr	50
K0qPVGdA91KO	45
r7VxefoTVHbb	43
yGbPyLPc8PmT	39
TuPm19CMKvrM	35
EGW4UK5bOeEZ	35
UOGIrJtSldvd	34
yHsOFiqVrV1u	30
coi9xkdo66JI	29

15) Determine the number of orders placed for each combination of payment type and installments.

```
select payment_type, payment_installments,
COUNT(DISTINCT order_id) as
total_orders
from payments
group by payment_type, payment_installments;
```

payment_type	payment_installments	total_orders
credit_card	1	690
credit_card	2	364
credit_card	3	299
credit_card	4	210
credit_card	5	153
credit_card	6	121
credit_card	7	38
credit_card	8	122
credit_card	9	16
credit_card	10	187
credit_card	11	1
credit_card	12	7
credit_card	13	1
credit_card	15	1
credit_card	16	1
credit_card	18	1
credit_card	20	1
credit_card	21	1
credit_card	24	3
debit_card	1	46
voucher	1	162
wallet	1	576

CONCLUSION

Overall, the analysis of this e-commerce dataset provides valuable insights into the business's performance and identifies areas for improvement. By implementing the recommended strategies, the business can enhance its operations, increase customer satisfaction, and drive growth.

</> https://github.com/Omkar-Salvi/SQL-PROJECTS