

E-Commerce Data Base Design

Requirements

- The e-commerce database needs to manage products, customers, orders, and payments effectively. For products, it should store details like name, price, and inventory. Customer information like name, email, and address should be securely stored, allowing for account creation and updates. Orders need to be tracked, including order ID, date, and items purchased, with options for modification or cancellation. Payment processing should be secure, supporting various methods like cards and PayPal, with transaction details stored for reference. Overall, the system should ensure data security, accuracy, and scalability for a smooth online shopping experience.

Conceptual design

Entities

- Product
- Customer
- Order
- Payment

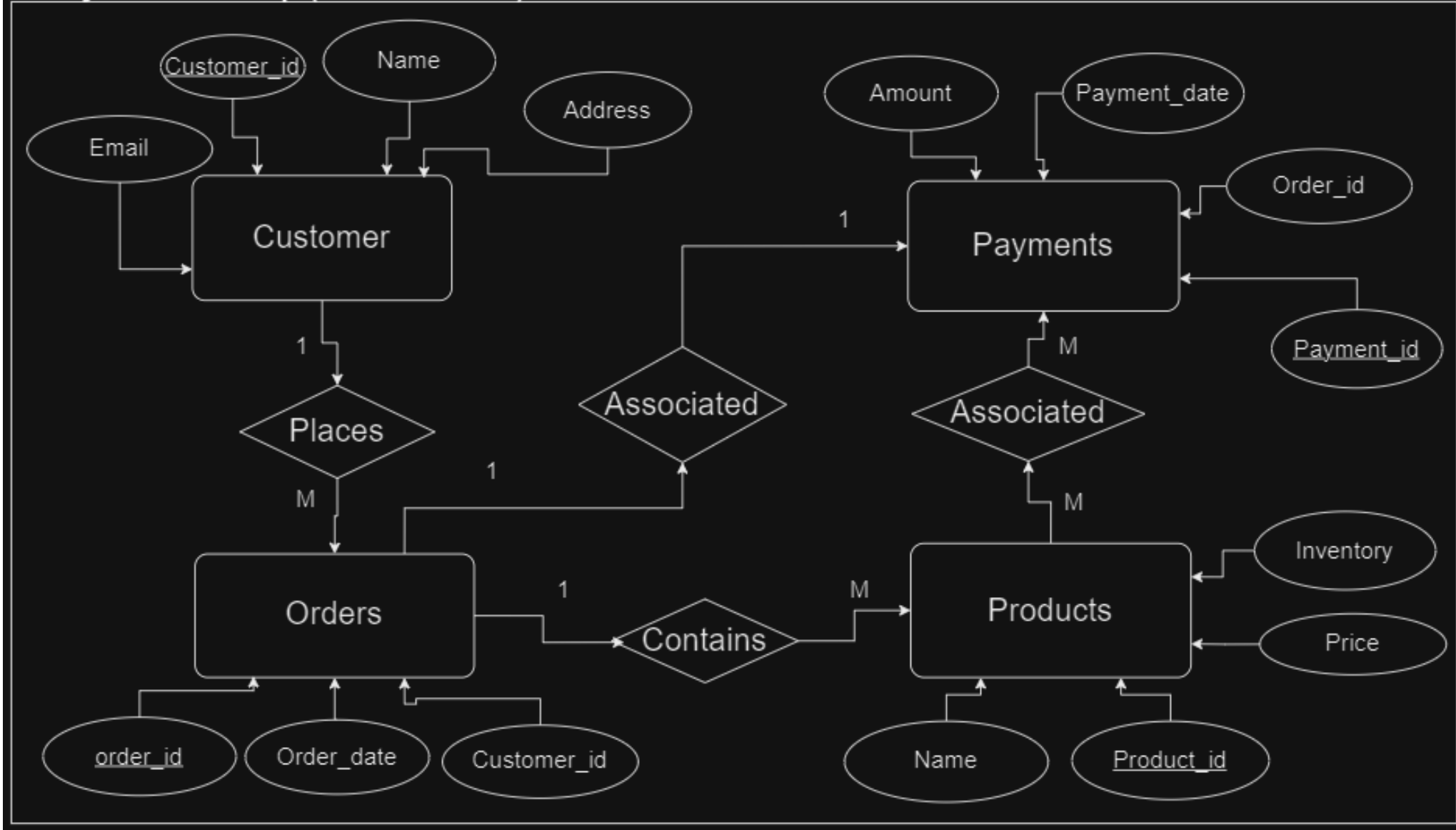
Conceptual Design

Relationships

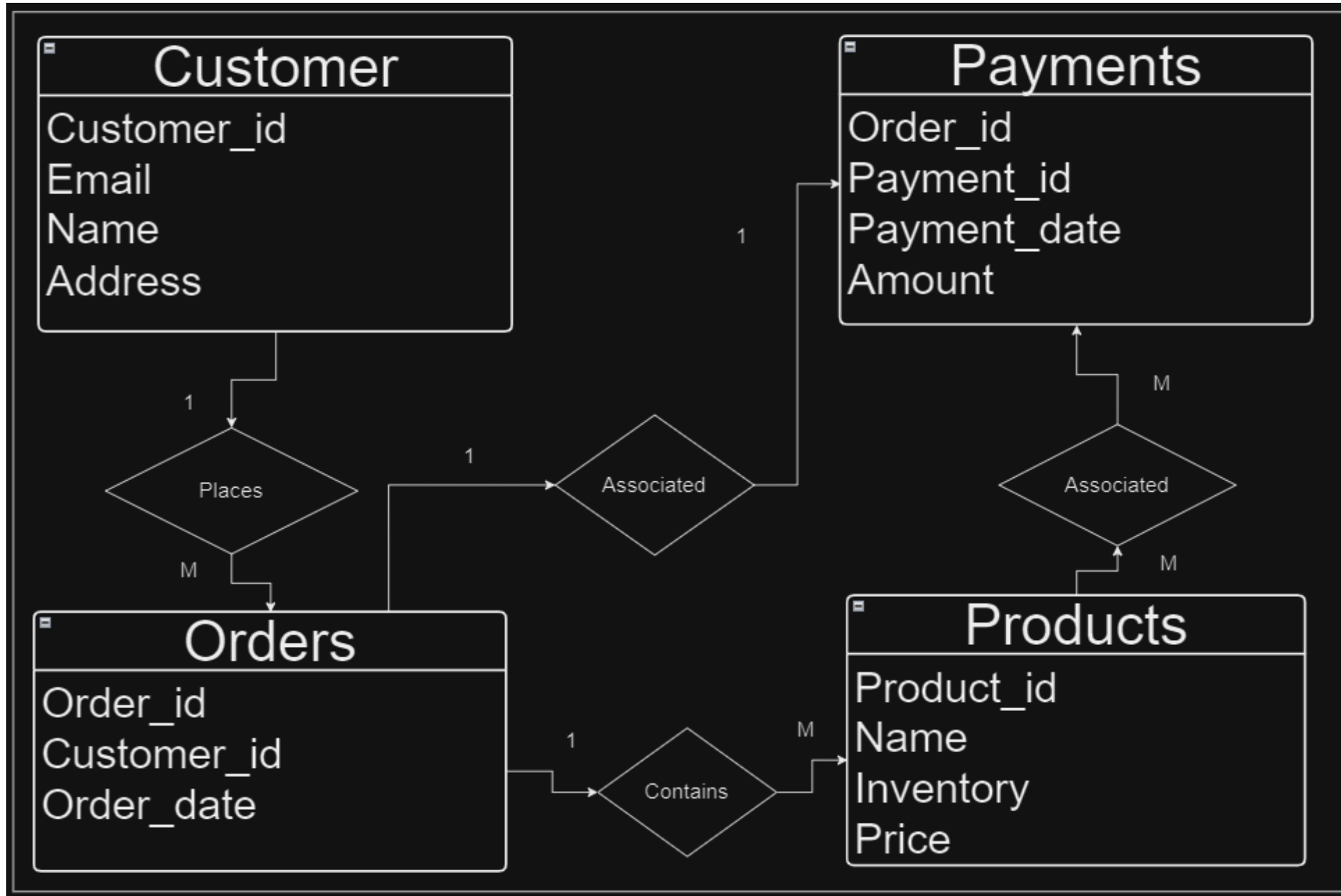
- Product and Order: One-to-Many (Each order can contain multiple products)
- Customer and Order: One-to-Many (Each customer can place multiple orders)
- Order and Payment: One-to-One (Each order is associated with one payment)
- Product and Payment: Many-to-Many (Multiple products can be associated with multiple payments, indicating purchase transactions)

Conceptual Design

Entity-Relationship (E-Commerce)



Logical Design



Physical Design (Implementation)

```
mysql> -- Customer Table
mysql> create table customer(
  -> customer_id int primary key,
  -> name varchar(50),
  -> email varchar(50),
  -> address varchar(50)
  -> );
```

Query OK, 0 rows affected (0.10 sec)

```
mysql> describe customer;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
name	varchar(50)	YES		NULL	
email	varchar(50)	YES		NULL	
address	varchar(50)	YES		NULL	

4 rows in set (0.01 sec)

Physical Design (Implementation)

```
mysql> create table orders(  
    -> order_id int primary key,  
    -> customer_id int,  
    -> order_date date,  
    -> foreign key(customer_id) references customer(customer_id)  
    -> );
```

Query OK, 0 rows affected (0.09 sec)

```
mysql> describe orders;
```

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
order_date	date	YES		NULL	

3 rows in set (0.00 sec)

Physical Design (Implementation)

```
mysql> -- product table
mysql> create table product(
  -> product_id int primary key,
  -> name varchar(50),
  -> price decimal(10,2),
  -> inventory int
  -> );
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> describe product;
```

Field	Type	Null	Key	Default	Extra
product_id	int	NO	PRI	NULL	
name	varchar(50)	YES		NULL	
price	decimal(10,2)	YES		NULL	
inventory	int	YES		NULL	

```
4 rows in set (0.00 sec)
```

Physical Design (Implementation)

```
mysql> -- payment table
mysql> create table payment(
    -> payment_id int primary key,
    -> order_id int,
    -> amount decimal(10,2),
    -> payment_date date,
    -> foreign key(order_id) references orders(order_id)
    -> );
```

Query OK, 0 rows affected (0.09 sec)

```
mysql> describe payment;
```

Field	Type	Null	Key	Default	Extra
payment_id	int	NO	PRI	NULL	
order_id	int	YES	MUL	NULL	
amount	decimal(10,2)	YES		NULL	
payment_date	date	YES		NULL	

4 rows in set (0.00 sec)

Physical Design (Implementation)

```
mysql> select * from customer;
```

customer_id	name	email	address
1	Aarav Patel	aarav@example.com	123, ABC Street, Mumbai
2	Ishaan Singh	ishaan@example.com	456, XYZ Road, Delhi
3	Avani Sharma	avani@example.com	789, PQR Avenue, Bangalore
4	Anaya Gupta	anaya@example.com	1011, LMN Lane, Kolkata
5	Riya Joshi	riya@example.com	1213, EFG Avenue, Chennai

```
5 rows in set (0.00 sec)
```

```
mysql> select * from orders;
```

order_id	customer_id	order_date
1	1	2024-04-12
2	2	2024-04-12
3	3	2024-04-13
4	4	2024-04-13
5	5	2024-04-14

```
5 rows in set (0.00 sec)
```

Physical Design (Implementation)

```
mysql> select * from payment;
```

payment_id	order_id	amount	payment_date
1	1	1500.00	2024-04-12
2	2	800.00	2024-04-12
3	3	3000.00	2024-04-13
4	4	2500.00	2024-04-13
5	5	1200.00	2024-04-14

```
5 rows in set (0.00 sec)
```

```
mysql> select * from product;
```

product_id	name	price	inventory
1	Saree	1500.00	50
2	Kurta	800.00	30
3	Lehenga	3000.00	20
4	Sherwani	2500.00	25
5	Salwar Suit	1200.00	40

```
5 rows in set (0.00 sec)
```

Conclusion

- The database is designed to facilitate the management of products, orders, customers, and payments for an e-commerce platform.
- Relationships exist between the tables, such as orders being associated with customers and payments being linked to orders.
- Referential integrity is maintained through foreign key constraints to ensure data consistency and accuracy.