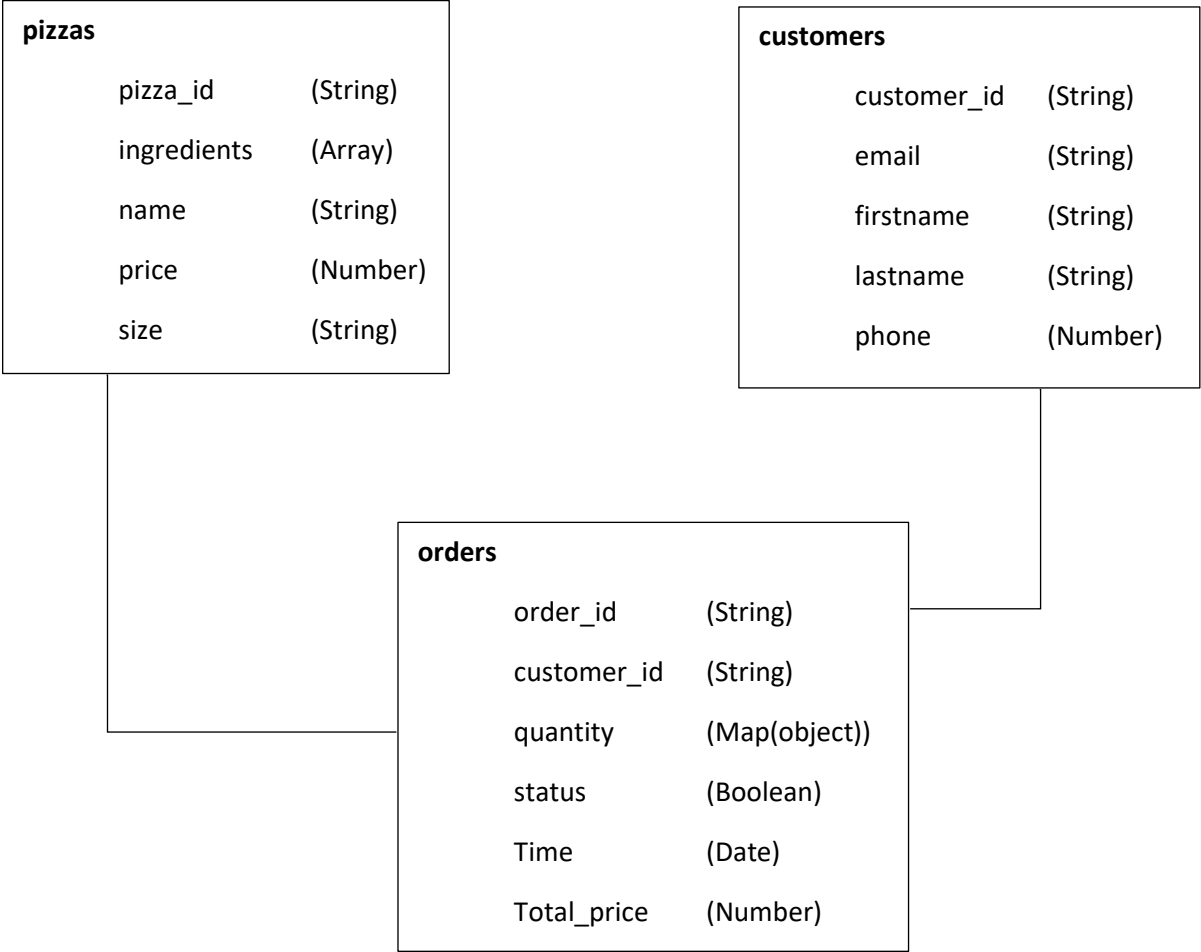


Pizza store

Pizza store เป็นการเก็บข้อมูลการสั่งซื้อพิซซ่า โดยมี ERD ข้อมูลดังนี้



โดยใช้เป็น design pattern ที่ใช้

Attribute pattern: จะใช้กับ order ในส่วนของ quantity เพื่อบอกเมนูและจำนวนพิซซ่าที่สั่งซื้อในรอบนั้น เพื่อให้ง่ายต่อการเรียกใช้งาน

ตัวอย่างข้อมูล

customers

🏠 > customers > c001		
🔗 int305fb088	📁 customers	📄 c001
+ Start collection	+ Add document	+ Start collection
customers >	c001 >	+ Add field
mycategories	c002	email: "john.smith@gmail.com"
myproducts	c003	firstname: "John"
orders	c004	lastname: "Smith"
pizzas	c005	phone: "093-486-7790"
posts	c006	
todos	c007	
users	c008	
zips	c009	
	c010	

pizzas

🏠 > pizzas > p001		
🔗 int305fb088	📁 pizzas	📄 p001
+ Start collection	+ Add document	+ Start collection
customers	p001 >	+ Add field
mycategories	p002	▼ ingredients
myproducts	p003	0 "pepperoni"
orders	p004	1 "mozzarella cheese"
pizzas >	p005	2 "tomato sauce"
posts	p006	3 "dough"
todos	p007	name: "Pepperoni Pizza"
users	p008	price: 12.99
zips	p009	size: "large"
	p010	
	p011	

orders

🏠 > orders > o001		
🔗 int305fb088	📁 orders	📄 o001
+ Start collection	+ Add document	+ Start collection
customers	o001 >	+ Add field
mycategories	o002	customer_id: "c002"
myproducts	o003	▼ quantity
orders >	o004	Meat_Lovers_Pizza: 2
pizzas	o005	Pepperoni_Pizza: 2
posts	o006	Veggie_Pizza: 1
todos	o007	status: true
users	o008	time: January 1, 2022 at 7:00:00 PM UTC+7
zips	o009	total_price: 38.98
	o010	
	o011	
	o012	
	o013	
	o014	

10 Query :

1) Show ordered the most in one order.

```
if (qryId == 1) {  
  title.value = "Show ordered the most in one order.";  
  let max_count = 0  
  qry1 = query(ordersRefs);  
  const querySnap = await getDocs(qry1);  
  
  querySnap.forEach((doc) => {  
    let data = doc.data()  
    let count = 0  
    for(let key in data){  
      count++  
    }  
    if(count>max_count){  
      max_count=count  
      orders.value.push(doc.data())  
    }  
  })  
  console.log(orders.value)  
}
```

ผลลัพธ์

← → localhost:5173/query/1

Pizza store

1. Show ordered the most in one order.
2. Displays pizzas is large-sized.
3. Display orders order by date descending.
4. show pizza that most Expensive and Cheapest Pizza
5. Show orders that have not yet been paid.
6. How much for all seafood pizza sold?
7. Pizza with mozzarella cheese or onions
8. Order total price more than 10 and less then 30
9. The pizza uses chicken as an ingredient and is small in size ,order by name descending .
10. top 3 cheapest pizza.

Show ordered the most in one order.

- customer.x002 ("Veggie_Pizza": 1, "Pepperoni_Pizza": 2, "Meat_Lovers_Pizza": 2) Price : 38.98

2) Displays pizzas is large-sized.

```
else if (qryId == 2) {
  title.value = "Displays pizzas is large-sized.";
  qry1 = query(pizzasRefs, where("size", "==", "large"));
  const querySnap = await getDocs(qry1);
  querySnap.forEach((doc) => {
    pizzas.value.push(doc.data())
  })
  console.log(pizzas.value)
}
```

ผลลัพธ์

The screenshot shows a web browser at localhost:5173/query/2. The page is divided into two columns. The left column, titled 'Pizza store', contains a list of 10 tasks. The right column, titled 'Displays pizzas is large-sized.', displays a list of three pizzas: Pepperoni Pizza (price: 12.99, size: large), Meat Lovers Pizza (price: 14.99, size: large), and Taco Pizza (price: 13.99, size: large). Seafood Pizza (price: 15.99, size: large) is also listed but appears to be cut off.

Pizza store

1. Show ordered the most in one order.
2. Displays pizzas is large-sized.
3. Display orders order by date descending.
4. show pizza that most Expensive and Cheapest Pizza
5. Show orders that have not yet been paid.
6. How much for all seafood pizza sold?
7. Pizza with mozzarella cheese or onions
8. Order total price more than 10 and less then 30
9. The pizza uses chicken as an ingredient and is small in size ,order by name descending .
10. top 3 cheapest pizza.

Displays pizzas is large-sized.

- name : Pepperoni Pizza
price : 12.99
size : large
- name : Meat Lovers Pizza
price : 14.99
size : large
- name : Taco Pizza
price : 13.99
size : large
- name : Seafood Pizza
price : 15.99
size : large

3) Display orders order by date descending.

```
else if (qryId == 3) {
  title.value = "Display orders order by date descending";
  qry1 = query(ordersRefs, orderBy("time", "desc"));
  const querySnap = await getDocs(qry1);
  querySnap.forEach((doc) => {
    orders.value.push(doc.data())
  })
  console.log(orders.value)
}
```

ผลลัพธ์

The screenshot shows a web browser at localhost:5173/query/3. The page is divided into two columns. The left column, titled 'Pizza store', contains a list of 10 tasks. The right column, titled 'Display orders order by date descending', displays a list of 20 orders, each with a customer ID, pizza name, and price. The orders are sorted by date in descending order.

Pizza store

1. Show ordered the most in one order.
2. Displays pizzas is large-sized.
3. Display orders order by date descending.
4. show pizza that most Expensive and Cheapest Pizza
5. Show orders that have not yet been paid.
6. How much for all seafood pizza sold?
7. Pizza with mozzarella cheese or onions
8. Order total price more than 10 and less then 30
9. The pizza uses chicken as an ingredient and is small in size ,order by name descending .
10. top 3 cheapest pizza.

Display orders order by date descending

- customer :x001 ("Buffalo_Chicken_Pizza": 1, "Spinach_and_Feta_Pizza": 1) Price : 22.98
- customer :x007 ("Margherita_Pizza": 2, "BBQ_Chicken_Pizza": 1, "Pepperoni_Pizza": 1) Price : 38.98
- customer :x003 ("Greek_Pizza": 1, "Buffalo_Chicken_Pizza": 2) Price : 34.98
- customer :x004 ("BBQ_Chicken_Pizza": 1, "Greek_Pizza": 1) Price : 22.99
- customer :x004 ("Seafood_Pizza": 1, "Taco_Pizza": 1, "Spinach_and_Feta_Pizza": 1, "BBQ_Chicken_Pizza": 1) Price : 52.96
- customer :x002 ("Spinach_and_Feta_Pizza": 1, "Meat_Lovers_Pizza": 2) Price : 29.98
- customer :x001 ("Hawaiian_Pizza": 1, "Veggie_Pizza": 1) Price : 19.98
- customer :x010 ("Hawaiian_Pizza": 2, "Pepperoni_Pizza": 2) Price : 43.96
- customer :x007 ("Seafood_Pizza": 2, "Spinach_and_Feta_Pizza": 1) Price : 31.98
- customer :x009 ("Spinach_and_Feta_Pizza": 1) Price : 12.99
- customer :x009 ("Buffalo_Chicken_Pizza": 1, "Greek_Pizza": 1) Price : 22.98
- customer :x001 ("Taco_Pizza": 2) Price : 27.98
- customer :x008 ("Hawaiian_Pizza": 2, "BBQ_Chicken_Pizza": 1) Price : 35.97
- customer :x003 ("Hawaiian_Pizza": 1) Price : 8.99
- customer :x002 ("Veggie_Pizza": 1, "Pepperoni_Pizza": 2, "Meat_Lovers_Pizza": 2) Price : 38.98

4) Show pizza that most Expensive and Cheapest Pizza

```
else if(qryId == 4){
  title.value = "Show pizza that most Expensive and Cheapest Pizza"
  qry1 = query(pizzasRefs,orderBy("price","desc"),limit(1))
  qry2 = query(pizzasRefs,orderBy("price"),limit(1))
  const querySnap1 = await getDocs(qry1)
  const querySnap2 = await getDocs(qry2)
  querySnap1.forEach((doc) => {
    pizzas.value.push(doc.data())
  })
  querySnap2.forEach((doc) => {
    pizzas.value.push(doc.data())
  })
  console.log(pizzas.value)
}
```

ผลลัพธ์

The screenshot shows a web browser at localhost:5173/query/4. The page has two columns. The left column, titled "Pizza store", contains a list of 10 tasks. The right column, titled "Show pizza that most Expensive and Cheapest Pizza", displays two pizza items: "Seafood Pizza" with a price of 15.99 and size of large, and "Hawaiian Pizza" with a price of 8.99 and size of small.

name	price	size
Seafood Pizza	15.99	large
Hawaiian Pizza	8.99	small

5) Show orders that have not yet been paid.

```
else if(qryId == 5){
  title.value = "Show orders that have not yet been paid."
  qry1 = query(ordersRefs, where("status","==",false))
  const querySnap1 = await getDocs(qry1);
  querySnap1.forEach((doc) => {
    orders.value.push(doc.data())
  })
  console.log(orders.value)
}
```

ผลลัพธ์

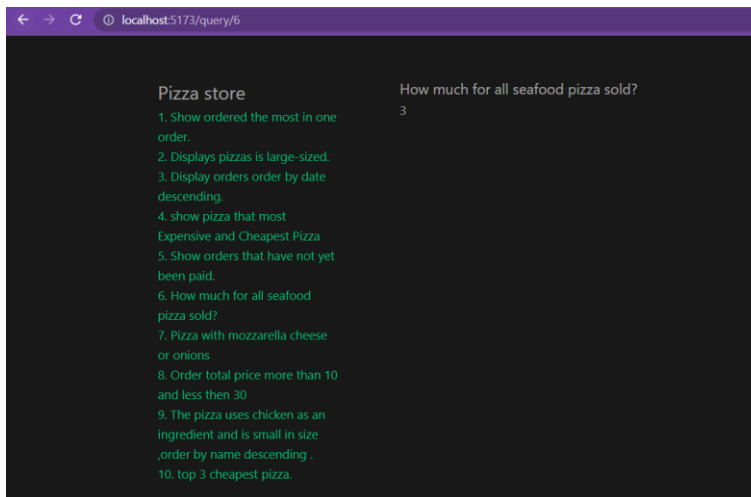
The screenshot shows a web browser at localhost:5173/query/5. The page has two columns. The left column, titled "Pizza store", contains the same list of 10 tasks as the previous screenshot. The right column, titled "Show orders that have not yet been paid.", displays a single order item: "customer x001 ('Spinach_and_Feta_Pizza': 1, 'Buffalo_Chicken_Pizza': 1) Price : 22.98".

customer	items	price
customer x001	('Spinach_and_Feta_Pizza': 1, 'Buffalo_Chicken_Pizza': 1)	22.98

6) How much for all seafood pizza sold?

```
else if(qryId == 6){
  title.value = "How much for all seafood pizza sold?";
  const querySnap = await getDocs(ordersRefs);
  querySnap.forEach((doc) => {
    let data = doc.data()
    // console.log(data.quantity.Seafood_Pizza)
    counts.value = counts.value + (data.quantity.Seafood_Pizza == undefined ? 0 : data.quantity.Seafood_Pizza)
  })
  console.log(counts.value)
}
```

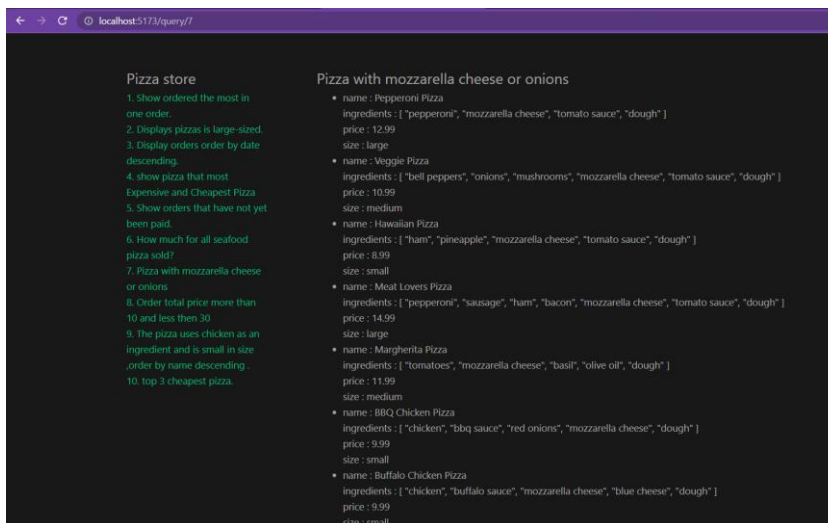
ผลลัพธ์



7) Pizza with mozzarella cheese or onions

```
else if(qryId==7){
  title.value = "Pizza with mozzarella cheese or onions"
  qry1 = query(pizzasRefs, where("ingredients", "array-contains-any", ["onions", "mozzarella cheese"]))
  const querySnap = await getDocs(qry1);
  querySnap.forEach((doc) => {
    pizzas.value.push(doc.data())
  })
  ingredients.value=true
  console.log(pizzas.value)
}
```

ผลลัพธ์



8) Order total price more than 10 and less then 30

```
else if(qryId==8){
  title.value = "Order total price more than 10 and less then 30 "
  qry1 = query(ordersRefs, where("total_price", ">=", 10),where("total_price", "<=", 30) )

  const querySnap = await getDocs(qry1)
  querySnap.forEach((doc) => {
    orders.value.push([doc.data()])
  })

  console.log(orders.value)
}
```

ผลลัพธ์

The screenshot shows a web browser at localhost:5173/query/8. The page is divided into two columns. The left column, titled 'Pizza store', lists 10 queries. The right column, titled 'Order total price more than 10 and less then 30', displays the results for query 8. The results are a list of 6 orders, each with a customer ID, pizza name, and price.

customer	price
customer-z009 ("Spinach_and_Feta_Pizza": 1)	Price : 12.99
customer-z001 ("Hawaiian_Pizza": 1, "Veggie_Pizza": 1)	Price : 19.98
customer-z009 ("Greek_Pizza": 1, "Buffalo_Chicken_Pizza": 1)	Price : 22.98
customer-z001 ("Buffalo_Chicken_Pizza": 1, "Spinach_and_Feta_Pizza": 1)	Price : 22.98
customer-z004 ("BBQ_Chicken_Pizza": 1, "Greek_Pizza": 1)	Price : 22.99
customer-z001 ("Taco_Pizza": 2)	Price : 27.98

9) The pizza uses chicken as an ingredient and is small in size ,order by name descending.

```
else if(qryId==9){
  title.value = "The pizza uses chicken as an ingredient and is small in size ,order by name descending ."
  qry1 = query(pizzasRefs, where("ingredients", "array-contains", "chicken"),where("size","=", "small"),orderBy("name","desc"))
  const querySnap = await getDocs(qry1)
  querySnap.forEach((doc) => {
    pizzas.value.push(doc.data())
  })
  console.log(pizzas.value)
  ingredients.value=true
}
```

ผลลัพธ์

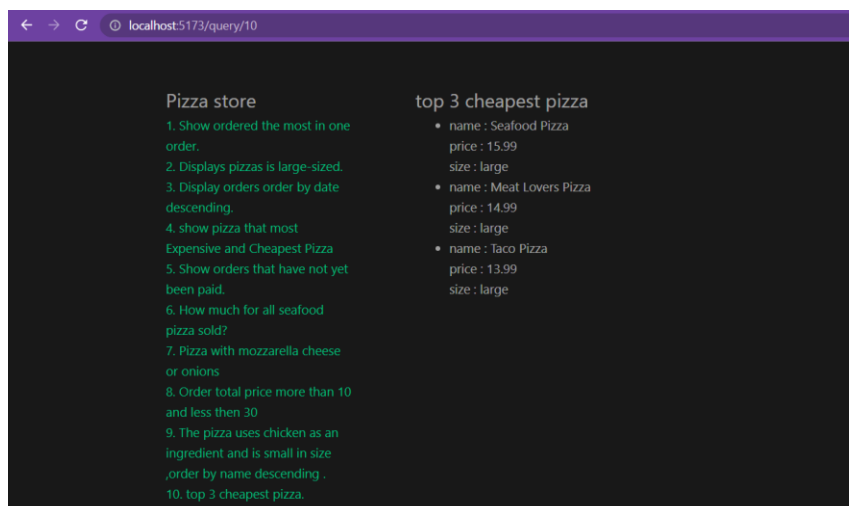
The screenshot shows a web browser at localhost:5173/query/9. The page is divided into two columns. The left column, titled 'Pizza store', lists 10 queries. The right column, titled 'The pizza uses chicken as an ingredient and is small in size ,order by name descending .', displays the results for query 9. The results are a list of 2 pizzas, each with a name, ingredients, price, and size.

name	price	size
name : Buffalo Chicken Pizza	price : 9.99	size : small
name : BBQ Chicken Pizza	price : 9.99	size : small

10) top 3 cheapest pizza.

```
else if(qryId==10){
  title.value = "top 3 cheapest pizza"
  qry1 = query(pizzasRefs, orderBy("price", "desc"), limit(3))
  const querySnap = await getDocs(qry1);
  querySnap.forEach((doc) => {
    pizzas.value.push(doc.data())
  })
  console.log(pizzas.value)
}
```

ผลลัพธ์



The screenshot shows a web browser window with the address bar displaying 'localhost:5173/query/10'. The page content is divided into two columns. The left column, titled 'Pizza store', contains a list of 10 queries. The right column, titled 'top 3 cheapest pizza', displays the results for the 10th query, which is 'top 3 cheapest pizza'. The results are listed as follows:

name	price	size
Seafood Pizza	15.99	large
Meat Lovers Pizza	14.99	large
Taco Pizza	13.99	large