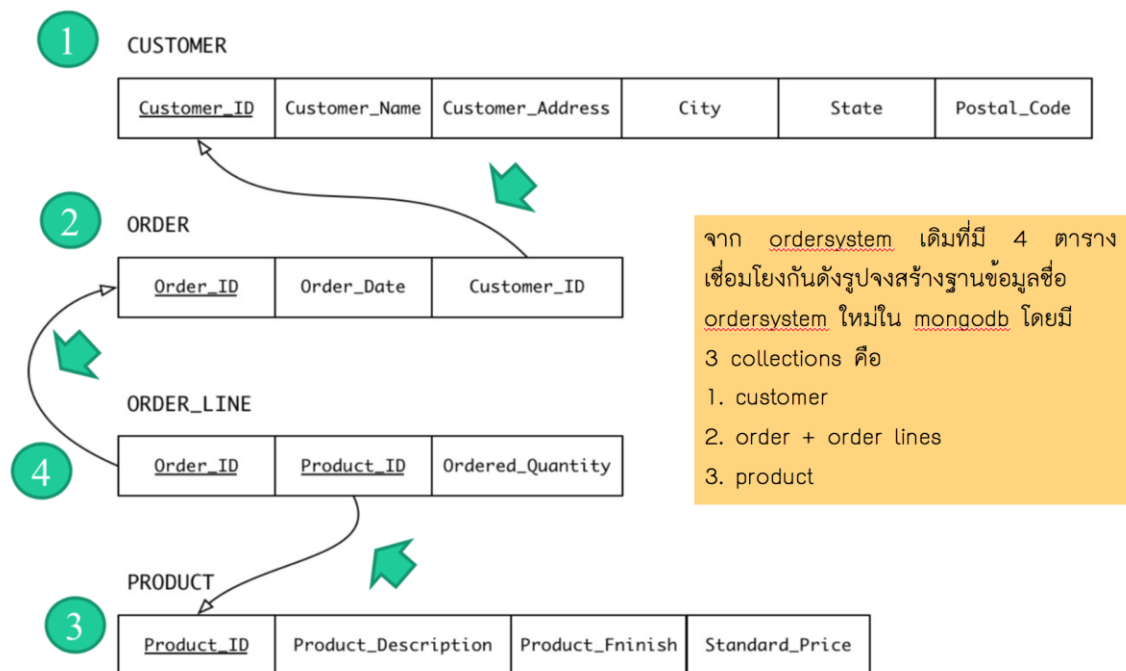


1. ให้ไปที่ pull ไฟล์ข้อมูลเพิ่มเติมจาก https://github.com/wichadak/2110322_DB_SYS โดยจะมีไฟล์ต่างๆ ของ exercise นี้ อยู่ภายใต้ folder ชื่อ MongoDBExercise ซึ่งประกอบด้วยไฟล์สคริปต์ชื่อ run_mongodbimport ใช้นำเข้าข้อมูล และ folder ย่อย ชื่อ ordersystem_for_mongo ที่มีชุดไฟล์ข้อมูล เราจะใช้ไฟล์เหล่านี้มาสร้างฐานข้อมูล ordersystem ในเวอร์ชัน mongodb
2. เดิมใน exercise SQL. ระบบ ordersystem ประกอบด้วย 4 ตาราง คือ ตาราง customer, order, order_lines, และ product ดังรูปต่อไปนี้ แต่ ordersystem เวอร์ชัน mongodb จะเหลือ 3 collections โดยจะรวม order_lines เข้ามาเป็น embedded document ของ order



3. ทำการนำเข้าข้อมูลในแต่ละ collection โดยใช้คำสั่งใน run_mongoimport หรือจะ import ผ่าน GUI ของ compass ก็ได้
4. เมื่อได้ collections ของข้อมูลเรียบร้อยแล้วให้เขียนคำสั่งใน mongodb เพื่อตอบโจทย์ต่อไปนี้
โดยแต่ละข้อให้สร้าง 1 ไฟล์ เก็บคำสั่งของ mongodb
โดยกำหนดชื่อไฟล์เป็น 4_1.txt, 4_2.txt,....
จากนั้นให้ zip ไฟล์ 4_1.txt,...., 4_16.txt และ upload zip file เข้าใน mycourseville

4.1 แสดงจำนวนลูกค้าแยกตาม customer_zipcode โดยเรียงตามจำนวน customers จากมากไปน้อย ถ้าเท่ากันให้เรียงตาม customer_zipcode จากน้อยไปมาก

Expected output

```
[{"_id": 10230,
 "customer_number": 2
},
{"_id": 10310,
 "customer_number": 2
},
{"_id": 10330,
 "customer_number": 2
},
{"_id": 10700,
 "customer_number": 2
},
{"_id": 10600,
 "customer_number": 1
}]
```

4.2 แสดงจำนวนลูกค้าแยกตาม customer_zipcode โดยเรียงตามจำนวน customers จากมากไปน้อย ถ้าเท่ากันให้เรียงตาม customer_zipcode จากน้อยไปมาก โดยแสดงเฉพาะ customer_zipcode ที่มีจำนวนลูกค้ามากกว่า 1

Expected output

```
[{"_id": 10230,
 "customer_number": 2
},
{"_id": 10310,
 "customer_number": 2
},
{"_id": 10330,
 "customer_number": 2
},
{"_id": 10700,
 "customer_number": 2
}]
```

4.3 จงแสดงรายการออเดอร์ของลูกค้ารหัส 10001 โดยเรียงลำดับของรหัสออเดอร์จากน้อยไปมาก

Expected output

```
[{
  "order_id": 3,
  "order_date": "2020-01-10 23:00:00",
  "customer_id": 10001,
  "order_lines": [
    {
      "product_id": 1,
      "ordered_quantity": 10
    },
    {
      "product_id": 3,
      "ordered_quantity": 11
    },
    {
      "product_id": 5,
      "ordered_quantity": 1
    }
  ]
},
{
  "order_id": 4,
  "order_date": "2020-01-11 23:00:00",
  "customer_id": 10001,
  "order_lines": [
    {
      "product_id": 6,
      "ordered_quantity": 10
    }
  ]
},
...
{
  "order_id": 15,
  "order_date": "2020-01-21 23:00:00",
  "customer_id": 10001
}
]
```

4.4 แสดงรหัสสินค้าและคำบรรยายสินค้าที่มีรายการสั่งซื้อมากที่สุด โดยเรียงลำดับตามรหัสสินค้าจากน้อยไปมาก

Expected output

```
[{
  "product_id": 3,
  "product_description": "Cabinet"
},
{
  "product_id": 5,
  "product_description": "Sofabed"
}]
```

4.5 จงแสดงรหัสลูกค้า ชื่อลูกค้า และจำนวนออเดอร์ที่สั่งโดยลูกค้าแต่ละคน โดยเรียงลำดับข้อมูลตามจำนวนออเดอร์จากมากไปน้อย

Expected output

```
[{
  "customer_id": 10002,
  "customer_name": "แสงทอง",
  "num_orders": 20
},
{
  "customer_id": 10001,
  "customer_name": "จอห์น",
  "num_orders": 12
},
{
  "customer_id": 10003,
  "customer_name": "สายฟ้า",
  "num_orders": 5
},
{
  "customer_id": 10008,
  "customer_name": "ปาริชาติ",
  "num_orders": 4
},
{
  "customer_id": 10009,
  "customer_name": "สมวัย",
  "num_orders": 1
}]
```

4.6 จงแสดงรหัสลูกค้า และชื่อลูกค้า ที่มีจำนวนออเดอร์มากอันดับแรก ถ้ามีมากกว่า 1 คนให้เอาชื่อคนที่มีความน้อยสุดเรียงตามลำดับตัวอักษร

Expected output

```
[{
  "customer_id": 10002,
  "customer_name": "แสงทอง"
}]
```

4.7 แสดงรหัสสินค้า คำบรรยายสินค้า และ จำนวนชิ้น ที่ถูกสั่งซื้อมากที่สุด

Expected output

```
[{
  "product_id": 3,
  "product_name": "Cabinet",
  "ordered_quantity": 26
}]
```

4.8 จงแสดงรายชื่อลูกค้าที่มีจำนวนออเดอร์มากที่สุด 3 อันดับแรก และจำนวนออเดอร์ของลูกค้าเหล่านั้น
ถ้ามีจำนวนออเดอร์มากที่สุดเท่ากันให้เรียงตาม customer_name จากน้อยไปมาก

Expected output

```
[{"customer_name": "แสงทอง",  
  "order_count": 20},  
{"customer_name": "จอห์น",  
  "order_count": 12},  
{"customer_name": "สายฟ้า",  
  "order_count": 5}]
```

4.9 จงแสดงรหัสและรายชื่อลูกค้า (ไม่ซ้ำ) ที่มีการสั่งซื้อสินค้าตั้งแต่วันที่ 2020-01-10 ถึง 2020-01-15 โดย
ใช้รหัสลูกค้าในการเรียงลำดับการแสดงผลจากน้อยไปมาก

Expected output

```
[{"customer_id": 10001,  
  "customer_name": "จอห์น"},  
{"customer_id": 10002,  
  "customer_name": "แสงทอง"}]
```

4.10 จงแสดงรายการรหัสสินค้า (product_id) คำอธิบายสินค้า (product_name) และสีของสินค้า
(product_color) ทั้งหมดที่มีสีขาวประกอบ โดย white เป็นตัวใหญ่หรือเล็กก็ได้ โดยให้เรียงลำดับ
ตามรหัสสินค้าจากน้อยไปมาก

Expected output

```
[{"product_id": 3,  
  "product_name": "Cabinet",  
  "product_color": "White Ash"},  
{"product_id": 6,  
  "product_name": "Lazy Boy",  
  "product_color": "White Ash"}]
```

4.11 จงแสดงรายการรหัสและชื่อลูกค้าทั้งหมดที่ยังไม่เคยมีรายการสั่งซื้อของเลย โดยเรียงตามลำดับ customer_id จากน้อยไปมาก

Expected output

```
[{
  "customer_id": 10004,
  "customer_name": "มารศรี"
},
{
  "customer_id": 10005,
  "customer_name": "ธรรณ"
},
{
  "customer_id": 10006,
  "customer_name": "ธนา"
},
{
  "customer_id": 10007,
  "customer_name": "จันทร์เจ้า"
}]
```

4.12 จงแสดงจำนวนออเดอร์ที่เกิดขึ้นระหว่างวันที่ 2020-01-10 ถึง 2020-01-15

Expected output

```
[{
  "total_order": 12
}]
```

4.13 จงแสดงค่า total payment ของออเดอร์หมายเลข 3

Expected output

```
[{
  "customer_id": 3,
  "total_payment": 15500
}]
```

4.14 เพิ่มข้อมูลสินค้าใหม่ต่อไปนี้เข้าตาราง product

product_id, product_name, product_color, produce_price

7,kitchen cabinet,Cherry,1500.00

8,table,Red Oak,550.00

และหลังเพิ่มแล้วให้ทำการ select รายการทั้งหมดของ product ออกมาแสดงด้วย

4.15 ลบรายการทั้งหมดของ customer_id 10001 ออกจากตาราง customer และข้อมูลทั้งหมดของ customer 10001 ในตารางทั้งหมดที่เกี่ยวข้อง จากนั้นทำการ select * from ตาราง customer, ordert, และ order_line ออกมาแสดง

4.16 ทำการ update ราคา product Sofabed จาก 7500.00 เหลือ 5400.00 และทำการ select * from product มาแสดง

5. ให้ความเห็นว่าการ embed ข้อมูลอาจทำในรูปแบบใดอีกในฐานข้อมูล ordersystem ข้างต้น และเลือกใช้การ embed ข้อมูลอย่างไรบ้างในระบบ เพราะอะไร