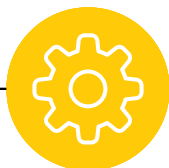
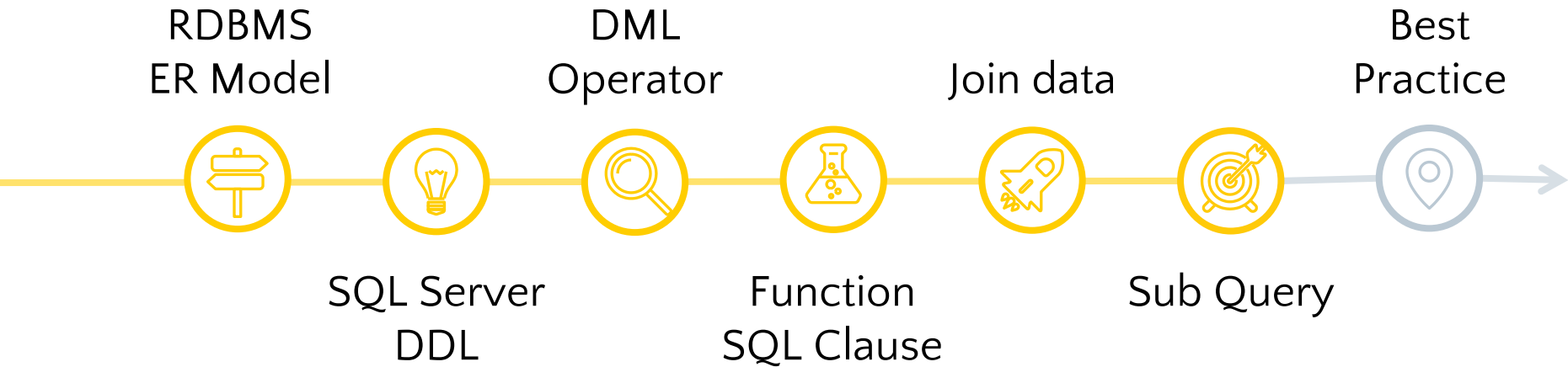


Welcome back



SQL *Essentials*

Roadmap



Previous lecture

- INNER JOIN
- OUTER JOIN
- EXCLUDING JOIN
- SELF JOIN
- CROSS JOIN
- UNION & UNION ALL

What will we explore today?

- Sub queries
- Advance operators
- Rules of sub query
- Practice

Sub query

- A **sub-query**, also called an **inner query**, is a SQL query nested inside a larger query.
- It's query inside query

How it's work

- Inner query is independent of outer query.
- Inner query is executed first and the results are stored.
- Outer query then runs on the stored results.
- Exception **Correlated subqueries**

Example 1

Results		Messages						
	ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	1	Nguyễn	Văn	Huân	7	8	9	2000-10-15
2	2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	4	Nguyễn	NULL	Truong	NULL	5	7	1999-10-15

```
SELECT LastName, Physic,  
FROM Student
```

```
(SELECT AVG(Physic * 1.0) FROM Student) AS 'AVG OF Physic'
```

2

LastName	Physic
Huân	8
Hiếu	4
Huệ	5
Truong	5

3

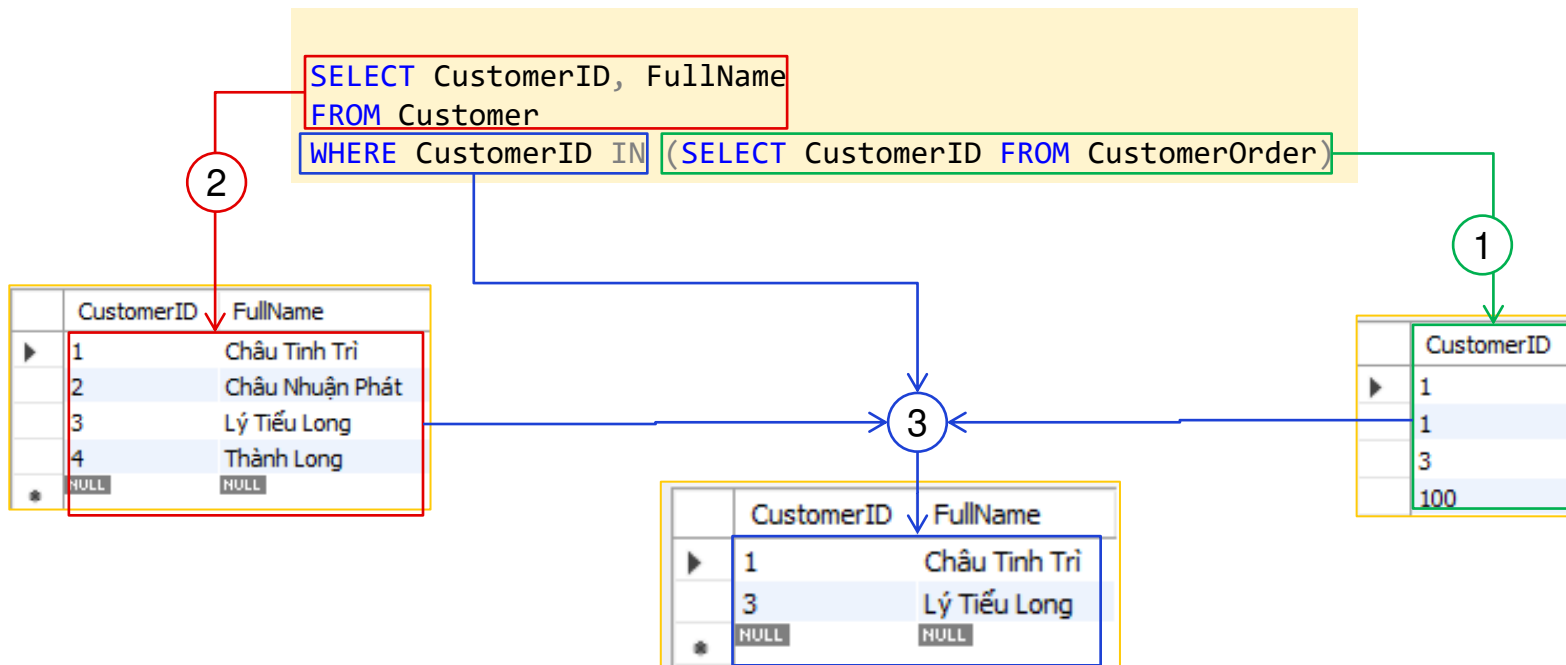
	LastName	Physic	AVG OF Physic
▶	Huân	8	5.50000
	Hiếu	4	5.50000
	Huệ	5	5.50000
	Trương	5	5.50000

1

AVG OF Physic = 5.500000

3

Example 2



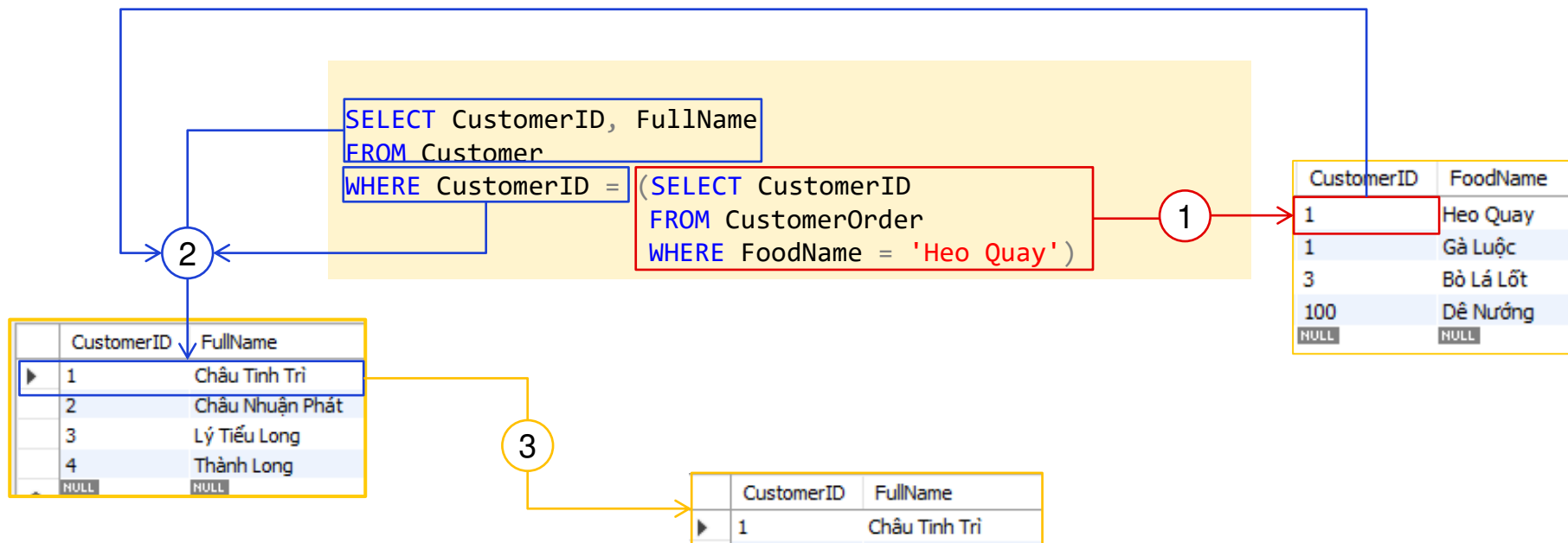
Type of Sub Query

- Single row sub query
- Multiple row sub query
- Multiple column sub query
- Corrolated sub query
- Nested sub query

Single row query

- A single row subquery returns zero or one row to the outer SQL statement.
- You can place a subquery in a WHERE, HAVING or FROM clause of a SELECT statement.

Single row query



Exercise

Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

DeliveryAddress

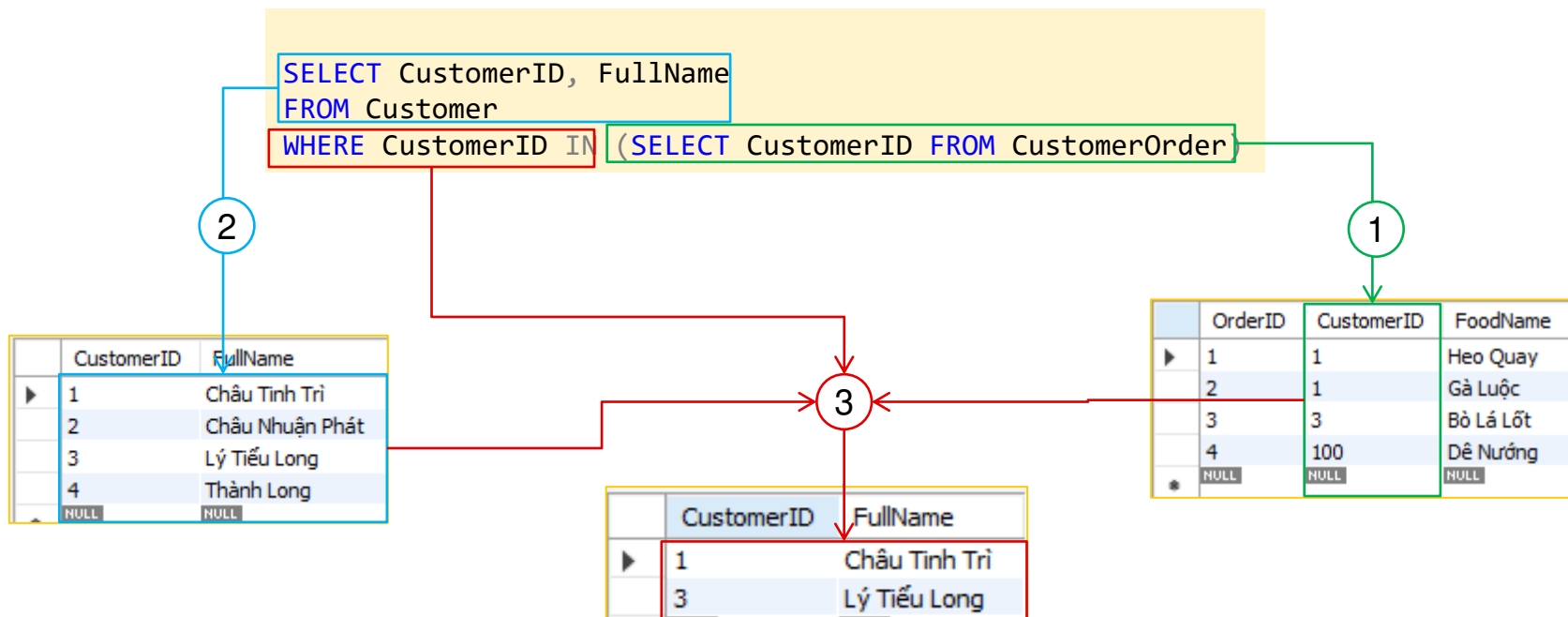
ID	FullAddress
1	TP. HCM
2	TP. HA NOI

Query CustomerID, OrderID, FoodName from CustomerOrder with Delivery FullAddress = "TP. HCM" using single row subquery

Multiple row subquery

- Multiple row subquery returns one or more rows to the outer SQL statement.
- Use the IN, ANY, or ALL operator in outer query to handle a subquery that returns multiple rows.

Multiple row subquery



Excercise

Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

DeliveryAddress

ID	FullAddress
1	TP. HCM
2	TP. HA NOI

Query CustomerID, OrderID, FoodName from CustomerOrder with Delivery FullAddress = "TP. HCM" or "TP. HA NOI" using multiple row subquery

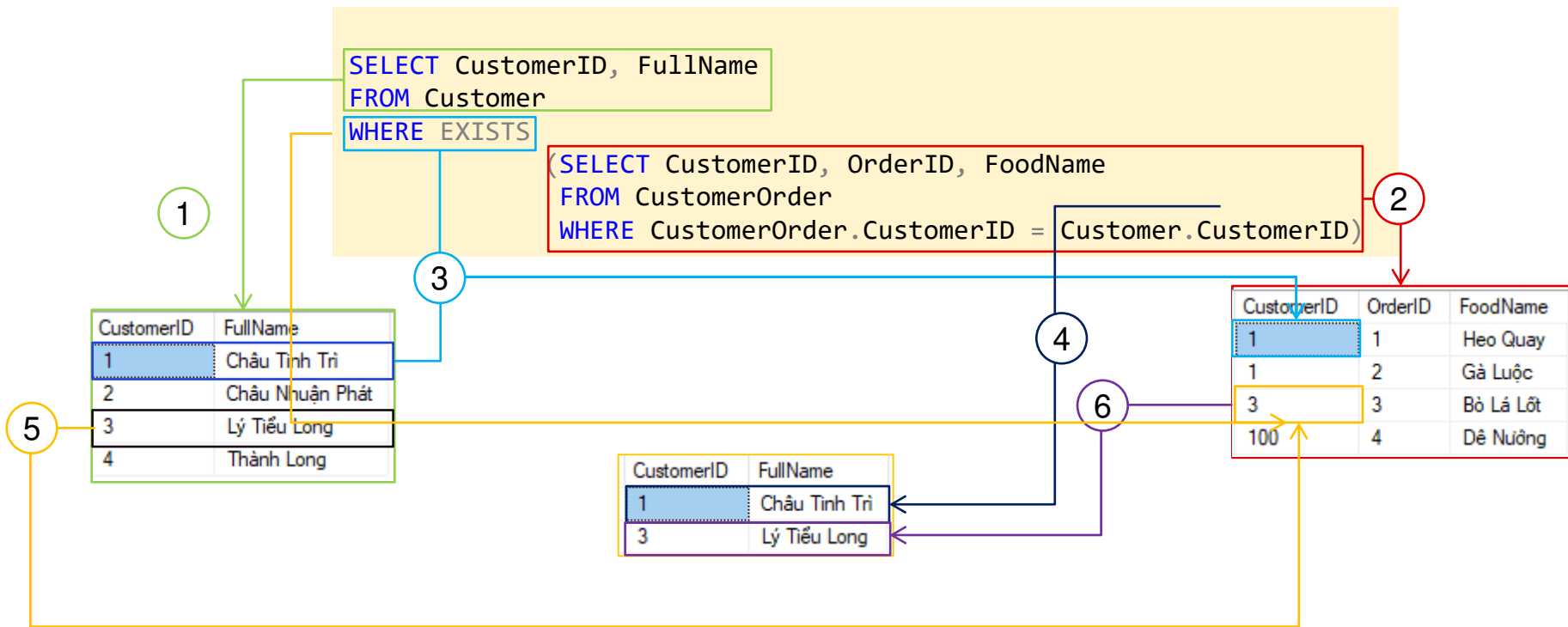
MULTIPLE COLUMN

- Multiple columns subquery returns more columns to outer SQL statement.
- Use the EXISTS operator in outer query to handle a subquery that returns multiple columns.

CORRELATED SUBQUERY

- Normal subquery executes the subquery first and provides the value to the outer query, whereas **correlated subquery** references a column in the outer query and **executes the subquery once for each row in the outer query..**

How it work



Excercise

```
CREATE TABLE Employee(  
    ID int PRIMARY KEY AUTO_INCREMENT,  
    FullName varchar(20) NOT NULL,  
    ManagerID int  
);  
  
INSERT INTO Employee (FullName, ManagerID)  
VALUES  
    ('HÀO CEO', NULL),  
    ('HƯỜNG DRECTOR', 1),  
    ('MÃN CTO', 1),  
    ('HUY CULI', 3)
```

Excercise 1

	ID	FullName	ManagerID
▶	1	HÀO CEO	NULL
	2	HƯƠNG DRECTOR	1
	3	MÃN CTO	1
	4	HUY CULI	3

Query the manager who has at least 1 employee

Excercise 2

	ID	FullName	ManagerID
▶	1	HÀO CEO	NULL
	2	HƯƠNG DRECTOR	1
	3	MÃN CTO	1
	4	HUY CULI	3

**Query the manager infomation (ID, FullName)
who has at least 2 employee**

NESTED SUBQUERY

```
USE LECTURE5 JOIN DEMO;
```

```
SELECT CustomerID, FullName  
FROM Customer
```

```
WHERE CustomerID IN (
```

```
SELECT CustomerID  
FROM CustomerOrder  
WHERE DeliveryAddressID =
```

```
(SELECT ID  
FROM DeliveryAddress  
WHERE FullAddress = 'TP. HA NOI')
```

```
)
```

	CustomerID	FullName
▶	1	Châu Tinh Trì
	2	Châu Nhuận Phát
	3	Lý Tiểu Long
	4	Thành Long

	OrderID	CustomerID	FoodName	DeliveryAddressID
▶	1	1	Hèo Quay	1
	2	1	Gà Luộc	1
	3	3	Bò Lá Lốt	2
	4	100	Dê Nướng	2

	ID	FullAddress
▶	1	TP. HCM
	2	TP. HA NOI

SOME RULE OF SUB QUERY

- Enclose a subquery in parenthesis.
- A subquery must include a SELECT clause and a FROM clause.
- Subqueries that return more than one row can only be used with multiple value operators, such as the IN, ALL, ANY operator.
- A subquery can include WHERE, GROUP BY, and HAVING clauses.
- You can include an ORDER BY clause only when a TOP clause is included.
- You can nest subqueries up to 32 levels.

Advanced Operators

- EXISTS
- ALL
- IN
- ANY

IN Operator

- The IN operator allows you to specify multiple values in a WHERE clause.
- The IN operator is a shorthand for multiple OR conditions.

```
SELECT ID, FullName  
FROM Employee  
WHERE ManagerID IN (1, 2, 3);
```

EXISTS Operator

- Used to test for the existence of any record in a subquery.
- The **EXISTS** operator returns TRUE if the subquery returns one or more records.

```
SELECT e.ID, e.FullName  
FROM Employee e  
WHERE EXISTS (  
    SELECT 1  
    FROM Employee m  
    WHERE m.ManagerID = e.ID  
);
```

ANY Operator

- Allow you to perform a comparison between a single column value and a range of other values.
- ANY means that the condition will be true if the operation is true for any of the values in the range.

```
SELECT ID, FullName  
FROM Employee  
WHERE ManagerID > ANY (SELECT ID FROM  
Employee WHERE ID IN (2, 4, 6));
```

ALL Operator

- Returns a boolean value as a result
- Returns TRUE if ALL of the subquery values meet the condition
- Be used with SELECT, WHERE and HAVING statements

```
SELECT ID, FullName  
FROM Employee  
WHERE ManagerID > ALL (SELECT ID FROM  
Employee WHERE ID IN (1, 2));
```

IN vs ANY

- You must place an =, <>, <, >, <=, or >= operator before ANY

Pratice time

```
CREATE TABLE Customer
```

```
(  
  CustID INT AUTO_INCREMENT PRIMARY KEY,  
  FirstName VARCHAR(10) NOT NULL,  
  LastName VARCHAR(10) NOT NULL,  
  Gender VARCHAR(10) NOT NULL,  
  DateOfBirth DATE,  
  Address VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE OrderFood
```

```
(  
  OrderID INT AUTO_INCREMENT PRIMARY KEY,  
  CustomerID INT NOT NULL,  
  Total DECIMAL(10, 2) NOT NULL,  
  DeliveryAddress VARCHAR(100) NOT NULL,  
  OrderDate DATE,  
  CHECK (Total > 0)  
);
```

Practice time (sub query)

- Get the customer list that never order
- Get the customer list that have at least 2 order
- Get the customer list that have at least 2 order in 2022 & at least 1 order in the March



Thank you!



Any questions?

Extra Resources

Name	Link
became SQL god?	https://www.w3schools.com/sql/default.asp