

# 데이터분석 기초 SQL 부트캠프

# 목차

## 12. 서브쿼리

**scalar,row,column,table,exists,**

**correlated**

selec절, from절, where절

## 13. With 구문

# 12.

## 서브쿼리(Sub-query)

Select절, from절, where절

## Scalar Subquery

- 스칼라 서브쿼리는 단일값을 반환
- 일반적으로 select, where, having 절에서 사용
- 모든 서브쿼리는 반드시 괄호 '()' 안에 포함.

## Scalar Subquery

- SQL Code

```
SELECT productName,
buyPrice
FROM products
WHERE buyPrice > (SELECT
AVG(buyPrice) FROM
products);
```

- 결과

productName	buyPrice
1952 Alpine Renault 1300	98.58
1996 Moto Guzzi 1100i	68.99
2003 Harley-Davidson Eagle ...	91.02
1972 Alfa Romeo GTA	85.68
1962 LanciaA Delta 16V	103.42
1968 Ford Mustang	95.34
2001 Ferrari Enzo	95.59

- 평균 가격(buyprice)보다 비싼 상품을 조회
- 서브 쿼리의 결과가 단일 값

## Table Subquery

- 테이블 서브쿼리는 테이블처럼 사용할 수 있는 행과 열을 반환
- 일반적으로 from 절에서 사용.
- From 절에 사용되는 서브쿼리는 별칭 필수

## Table Subquery

- SQL Code

```
SELECT customerNumber,  
order_count  
FROM (SELECT  
customerNumber,  
COUNT(orderNumber) AS  
order_count  
FROM orders GROUP BY  
customerNumber) AS  
subquery  
WHERE order_count >= 5;
```

- 주문 개수가 5개 이상인 고객만 조회
- 서브 쿼리의 결과가 테이블


- 결과

결과 #1 (7r x 2c)	
customerNumber	order_count
114	5
124	17
141	26
145	5
148	5
323	5
353	5

## Scalar Subquery 실습

- 문제 1: 최대 주문 개수를 가진 고객의 ID를 조회

- 정답

orders (1r × 1c)	
customerNumber	
	141



## Table Subquery 실습

- 문제 2: products와 orderdetails 테이블을 이용하여, 각 제품별로 총 주문 금액(quantityOrdered \* priceEach)을 계산하고, 그 결과를 기반으로 상위 5개의 제품만 조회하세요.

- 정답

결과 #1 (5r x 3c)		
productCode	productName	totalAmount
S18_3232	1992 Ferrari 360 Spider red	276,839.98
S12_1108	2001 Ferrari Enzo	190,755.86
S10_1949	1952 Alpine Renault 1300	190,017.96
S10_4698	2003 Harley-Davidson Eagle ...	170,686.0
S12_1099	1968 Ford Mustang	161,531.48

## 상관 Subquery

- 상관 서브쿼리(Correlated Subquery)는 일반 서브쿼리와는 달리 메인 쿼리의 각 행을 참조하여 수행
- 메인 쿼리의 각 행마다 한 번씩 실행되며, 메인 쿼리의 현재 행과 연관된 결과를 반환

## 상관 Subquery

- **특징:**
- **반복 실행:** 상관 서브쿼리는 메인 쿼리의 각 행에 대해 별도로 실행. 따라서 메인 쿼리에 100개의 행이 있다면 상관 서브쿼리도 100번 실행
- **참조:** 상관 서브쿼리는 메인 쿼리의 열을 참조할 수 있음. 이 참조를 통해 서브쿼리는 메인 쿼리의 현재 행에 따라 다른 값을 반환할 수 있음.

## 상관 Subquery

- SQL Code & “각 고객별로 가장 최근의 주문일을 조회하세요”

```
SELECT c.customerName, o.orderDate
FROM customers c, orders o
WHERE c.customerNumber = o.customerNumber
AND o.orderDate = (SELECT MAX(orderDate) FROM orders WHERE
customerNumber = c.customerNumber);
```

- 결과

결과 #1 (98r x 2c)	
customerName	orderDate
King Kong Collectab...	2003-12-01
Men 'R' US Retailers...	2004-01-09
Double Decker Gift ...	2004-01-22

## (상관서브쿼리)Subquery 실습

- 문제 3: 2003년에 주문한 모든 고객의 이름을 조회하세요.

- 정답

customers (74r × 1c)	
customerName	
Atelier graphique	
Signal Gift Stores	
Australian Collectors, Co.	
Baane Mini Imports	
Mini Gifts Distributors Ltd.	

# 13.

## WITH 구문

Common Table Expression(CTE)

## Common Table Expression(CTE)

- WITH문 (Common Table Expression, CTE)과 서브쿼리는 둘 다 SQL 쿼리의 구조를 단순화하고 복잡한 쿼리를 분해하는 데 유용

## With문 vs subquery

- 정의 및 가독성:
- **CTE (WITH문)**: CTE는 쿼리의 시작 부분에 정의되며, 이름을 가진 임시 결과 집합을 생성-> 코드의 가독성 향상, CTE 내에서 정의된 쿼리를 메인 쿼리에서 여러 번 재사용할 수 있음.
- **서브쿼리**: 서브쿼리는 메인 쿼리 내에서 일반적으로 한 번만 사용됩니다.



## Common Table Expression(CTE)

- WITH문 (Common Table Expression, CTE)를 스크립드 맨 먼저 선언을 해준 다음, 메인쿼리에서 사용

```
WITH cte_name AS (  
  -- CTE 정의 쿼리  
)  
  
-- 메인 쿼리  
SELECT * FROM cte_name;
```

## Common Table Expression(CTE)

- WITH문 (Common Table Expression, CTE)를 스크립드 맨 먼저 선언을 해준 다음, 메인쿼리에서 사용

## Common Table Expression(CTE)

- orders 테이블에서 가장 최근 주문을 찾는 쿼리를 작성

```
WITH LatestOrders AS (  
  SELECT customerNumber, MAX(orderDate) AS MaxOrderDate  
  FROM orders  
  GROUP BY customerNumber  
)  
SELECT o.orderNumber, o.orderDate, o.customerNumber  
FROM orders o  
JOIN LatestOrders lo ON o.customerNumber = lo.customerNumber AND  
o.orderDate = lo.MaxOrderDate;
```

## Common Table Expression(CTE)실습

- 문제 5: products 테이블에서 각 제품 라인별로 평균 제품 가격을 계산하세요. 그리고 이 평균 가격보다 높은 가격을 가진 제품들만을 해당 제품 라인별로 조회하세요.

### 정답

productName	productLine	buyPrice	avgPrice
1952 Alpine Renault 1300	Classic Cars	98.58	64.446316
1996 Moto Guzzi 1100i	Motorcycles	68.99	50.685385
2003 Harley-Davidson Eagle ...	Motorcycles	91.02	50.685385
1972 Alfa Romeo GTA	Classic Cars	85.68	64.446316
1962 LanciaA Delta 16V	Classic Cars	103.42	64.446316
1968 Ford Mustang	Classic Cars	95.34	64.446316
2001 Ferrari Enzo	Classic Cars	95.59	64.446316

## Common Table Expression(CTE)실습

- 문제 6: 각 제품 라인별로 제품의 평균 가격과 전체 제품의 평균 가격을 비교하여 전체 평균 가격보다 높은 제품 라인만 조회하세요.

- 정답

products (2r x 3c)		
productLine	avgPrice	totalAvgPrice
Classic Cars	64.446316	54.395182
Trucks and Buses	56.329091	54.395182

## Common Table Expression(CTE)실습

- 문제 7: 각 직원별로  
맡은 고객들의 평균  
크레딧 한도를  
계산하고, 크레딧  
한도의 평균이  
\$100,000 이상인  
직원만 조회하세요.

- 정답

employees (1r x 4c)			
employeeNumber	firstName	lastName	avgCredit
1,165	Leslie	Jennings	100,433.333333