━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

\*\*Linux(ubuntu) 및 PuTTY를 활용한 MySQL 실습 환경 구성

및 집계함수 명령어 실습 \*\*

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

과목명 : DB 기초 및 실습

강사 : 강사님

과정 : AI를 활용한 보안전문가 양성과정

이름 :

제출일 : 2025년 7월 7일

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

**Linux 접속**

**> putty > Connection > SSH >**

**Auth > Credentials > Private Key 선택**

**ubuntu 로 접속**

**$ su root**

**$ docker ps**

**docker container 접속**

**$ docker exec -it mysql\_server /bin/bash**

**mysql 접속**

**$ mysql -uroot -p**

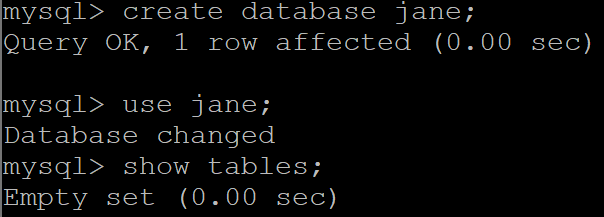
**mysql > show databases;**

**mysql > use onestar;**

**mysql > show tables;**

**mysql > select \* from [table 명];**

**CREATE DATABASE jane;**

****

**집계함수**

-- 판매 데이터를 저장할 테이블 생성

CREATE TABLE sales (

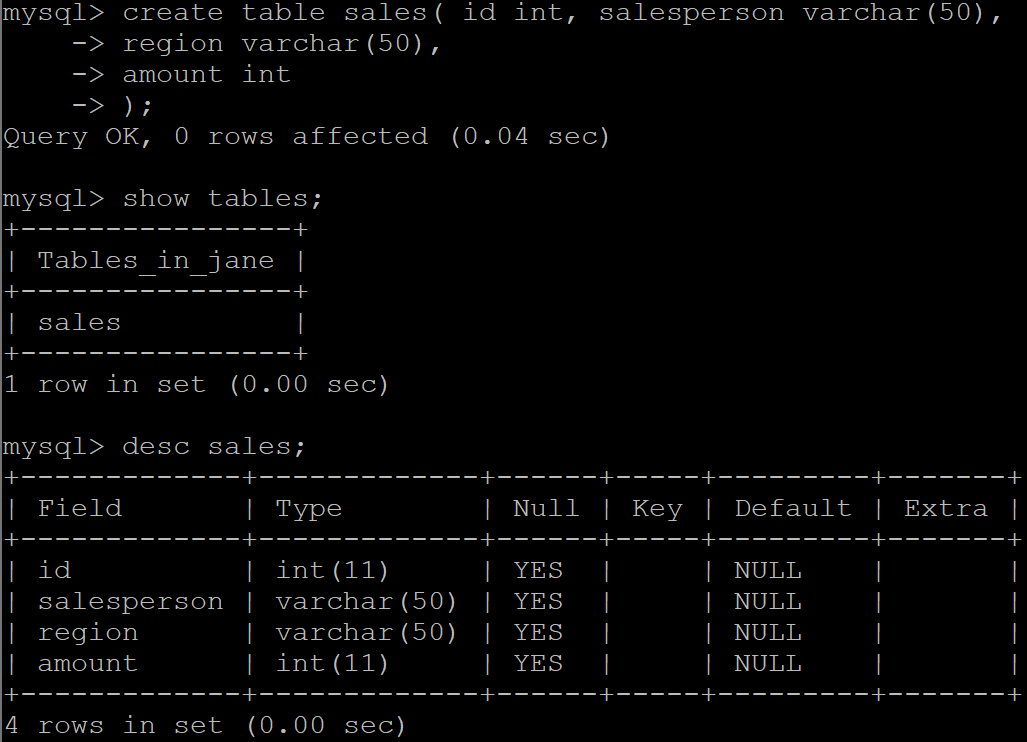
id INT,

salesperson VARCHAR(50),

region VARCHAR(50),

amount INT

);



-- 테스트용 데이터 삽입

INSERT INTO sales (id, salesperson, region, amount) VALUES

(1, 'Alice', 'Seoul', 100),

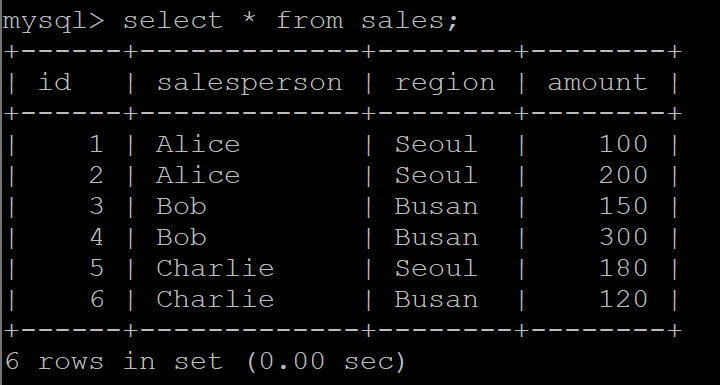
(2, 'Alice', 'Seoul', 200),

(3, 'Bob', 'Busan', 150),

(4, 'Bob', 'Busan', 300),

(5, 'Charlie', 'Seoul', 180),

(6, 'Charlie', 'Busan', 120);

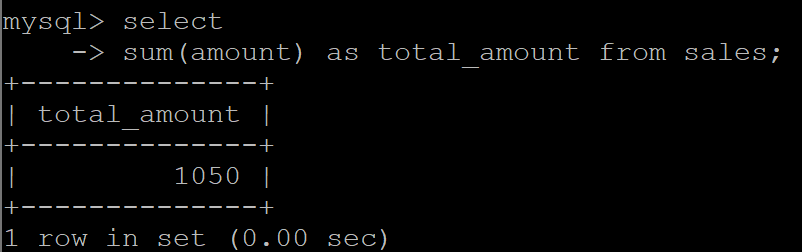


-- 전체 판매 금액의 총합을 구함

SELECT

SUM(amount) AS total\_amount

FROM sales;

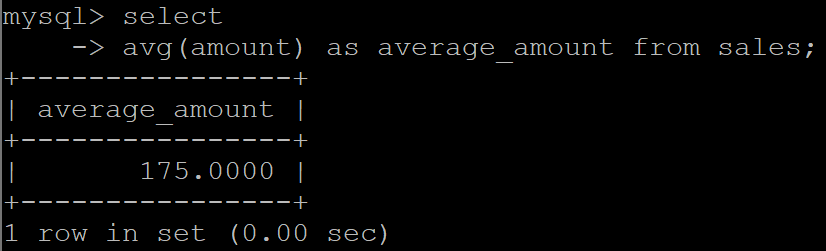


-- 전체 판매 금액의 평균을 구함

SELECT

AVG(amount) AS average\_amount

FROM sales;

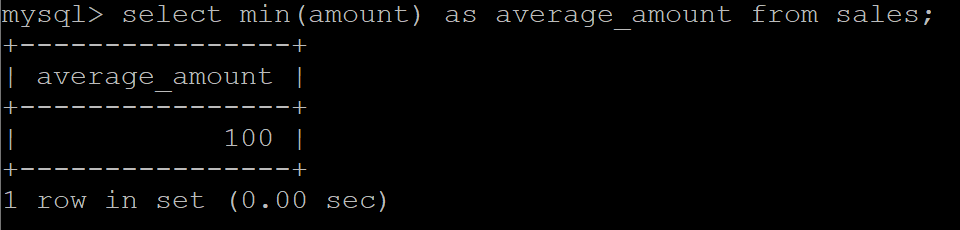


-- 판매 금액 중 가장 작은 값을 구함

SELECT

MIN(amount) AS min\_amount

FROM sales;

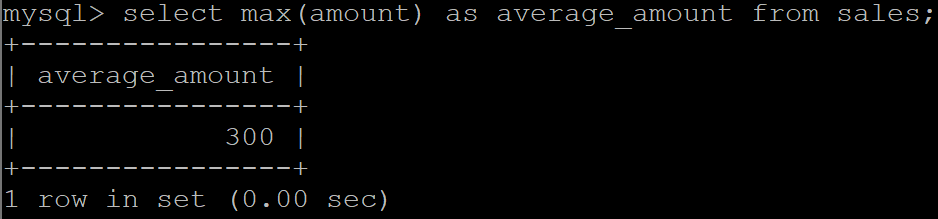


-- 판매 금액 중 가장 큰 값을 구함

SELECT

MAX(amount) AS max\_amount

FROM sales;



CREATE TABLE Orders ( OrderID INT PRIMARY KEY, CustomerID VARCHAR(10), ProductName VARCHAR(50), Amount INT );

INSERT INTO Orders (OrderID, CustomerID, ProductName, Amount) VALUES (1, 'C001', 'Laptop', 1500000),

(2, 'C002', 'Smartphone', 1000000), (3, 'C003', 'Tablet', 700000), (4, 'C004', 'Smartwatch', 500000),

(5, 'C005', 'Desktop', 2000000), (6, 'C006', 'Monitor', 700000), (7, 'C007', 'Wireless Earbuds', 300000), (8, 'C008', 'Smartphone', 1000000), (9, 'C009', 'Tablet', 700000);

**ROW\_NUMBER()**

SET @row\_num := 0;

SELECT

OrderID,

Amount,

@row\_num := @row\_num + 1 AS RowNum

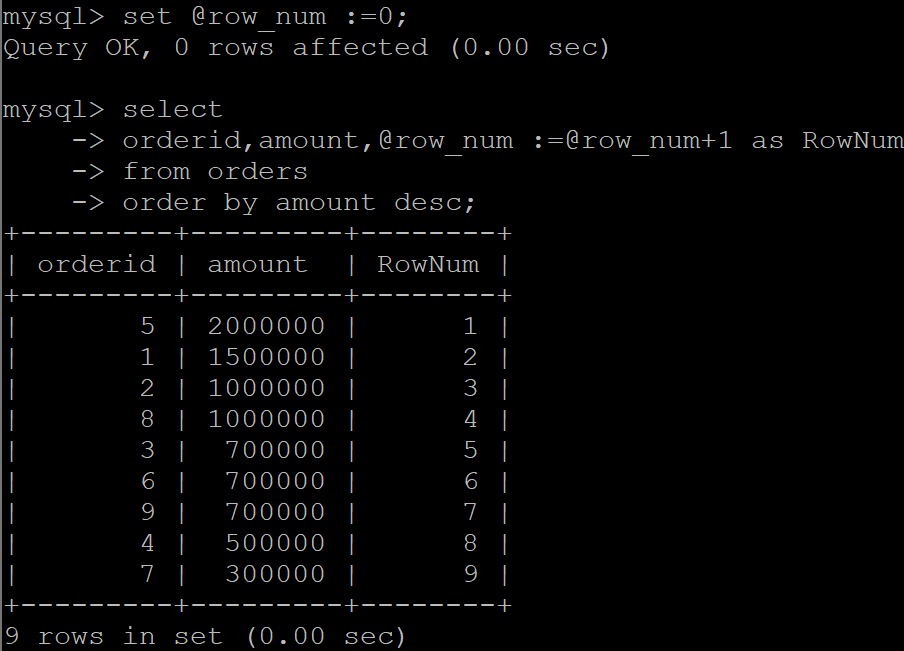
FROM Orders

ORDER BY Amount DESC;

//

@row\_num := @row\_num + 1 AS RowNum

* 한 줄 select 할 때 마다 숫자 1개씩 더 해라



**RANK();**

SET @rank := 0;

SET @prev\_amt := NULL;

SET @count := 0;

SELECT

OrderID,

CustomerID,

ProductName,

Amount,

@rank := IF(@prev\_amt = Amount, @rank, @count + 1) AS rank,

@count := @count + 1,

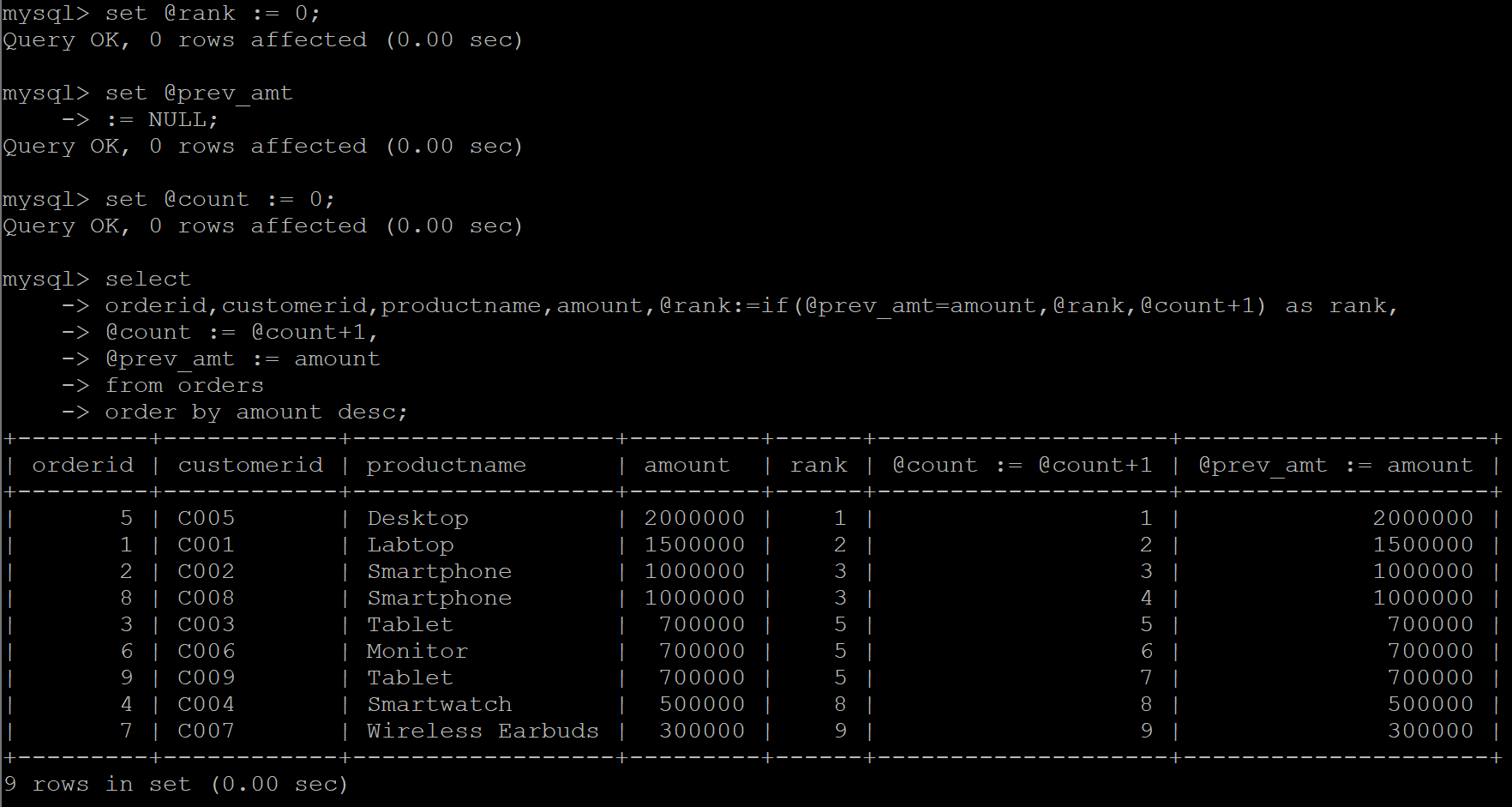
@prev\_amt := Amount

FROM Orders

ORDER BY Amount DESC;

//

@rank := IF(@prev\_amt = Amount, @rank, @count + 1) AS rank,



**DRANK();**

SET @drank := 0;

SET @prev\_amt := NULL;

SELECT

OrderID,

CustomerID,

ProductName,

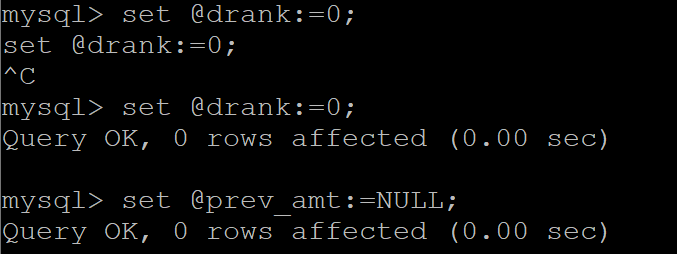
Amount,

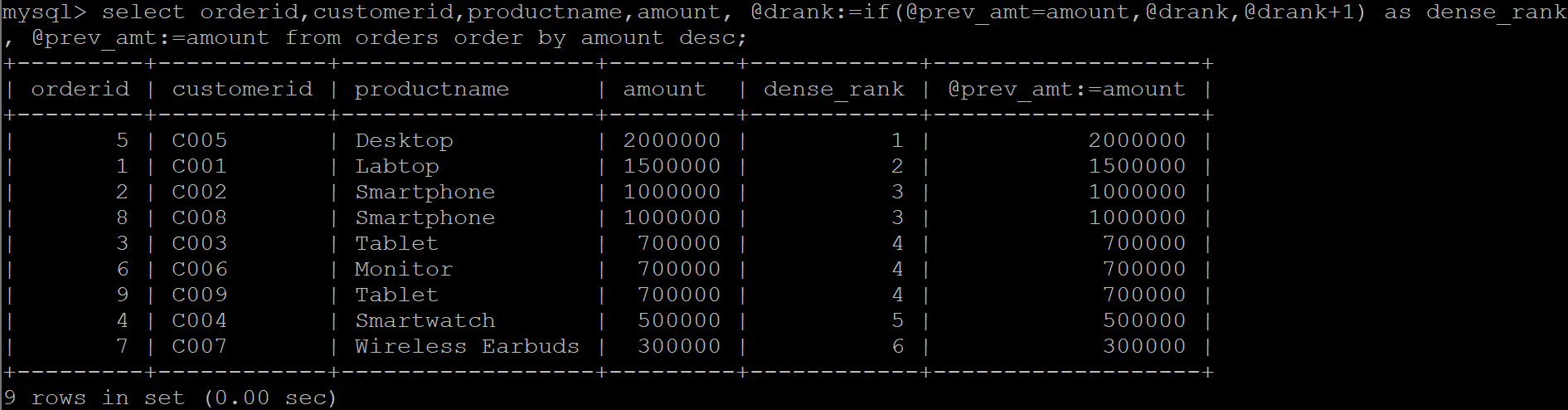
@drank := IF(@prev\_amt = Amount, @drank, @drank + 1) AS dense\_rank,

@prev\_amt := Amount

FROM Orders

ORDER BY Amount DESC;





**NTILE(n)**

SELECT COUNT(\*) INTO @total FROM Orders;

SET @n := 4;

SET @per\_group := CEIL(@total / @n);

SET @row := 0;

SELECT

OrderID,

CustomerID,

ProductName,

Amount,

FLOOR((@row := @row + 1 - 1) / @per\_group) + 1 AS ntile\_4

FROM Orders

ORDER BY Amount DESC;

// 전체를 n개의 그룹으로 나눈 값을 출력

