

데이터분석 기초 SQL 부트캠프

목차

1. WHERE 구문

비교연산자(<,>=)

논리연산자(AND, OR, NOT)

LIKE, IN ,BETWEEN, ISNULL

2. ORDER BY구문

3. GROUP BY 구문

6.

WHERE구문

비교연산자(<,>,), 논리연산자(AND, OR, NOT)
LIKE, IN ,BETWEEN, ISNULL

비교연산자(<, >, =)

- SQL Code

```
select  
    컬럼  
from  
    테이블  
where  
    조건
```

비교연산자(<, >, =)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE orderlinenumber = 1
;
```

- 결과

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10,100	S24_3969	49	35.29	1
10,101	S18_2795	26	167.06	1
10,102	S18_1367	41	43.13	1
10,103	S24_2300	36	107.34	1
10,104	S12_3148	34	131.44	1
10,105	S24_3816	50	75.47	1
10,106	S700_2834	32	113.9	1
10,107	S12_2823	21	122.0	1
10,108	S24_3856	40	132.0	1
10,109	S18_2870	26	126.72	1
10,110	S18_2795	31	163.69	1
10,111	S18_3136	43	94.25	1
10,112	S10_1949	29	197.16	1
10,113	S32_3522	23	58.82	1

비교연산자(<, >, =)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE productcode =
's24_3969'
;
```

- 결과

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10,100	S24_3969	49	35.29	1
10,110	S24_3969	48	35.29	5
10,124	S24_3969	46	36.11	4
10,138	S24_3969	29	32.82	4
10,149	S24_3969	26	38.57	9
10,162	S24_3969	37	32.82	7
10,173	S24_3969	35	35.7	11
10,182	S24_3969	23	34.88	8
10,193	S24_3969	22	38.16	12
10,204	S24_3969	39	34.88	2
10,214	S24_3969	44	38.57	5
10,227	S24_3969	27	34.88	8
10,242	S24_3969	46	36.52	1

비교연산자(<, >, =)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE quantityordered > 70
;
```

- 결과

orderdetails (8r x 5c)					
orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber	
10,401	S700_2466	85	98.72	10	
10,401	S700_3167	77	73.6	9	
10,404	S12_3990	77	67.03	4	
10,404	S18_3278	90	67.54	6	
10,405	S12_4675	97	115.16	5	
10,405	S24_3856	76	127.79	3	
10,407	S18_1749	76	141.1	2	
10,407	S24_2766	76	81.78	6	

비교연산자(<, >, =)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE quantityordered >=
70
;
```

- 결과

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10,401	S700_2466	85	98.72	10
10,401	S700_3167	77	73.6	9
10,404	S12_3990	77	67.03	4
10,404	S18_3278	90	67.54	6
10,405	S12_4675	97	115.16	5
10,405	S24_3856	76	127.79	3
10,407	S18_1749	76	141.1	2
10,407	S24_2766	76	81.78	6
10,412	S24_2300	70	109.9	10
10,419	S24_3856	70	112.34	8

비교연산자(<, >, =) 실습

- 문제 1. customers 테이블에서 creditLimit이 10000보다 큰 고객들의 이름(customerName)을 조회하세요..

- 정답

customers (98r x 1c)	
customerName	
Atelier graphique	
Signal Gift Stores	
Australian Collectors, Co.	
La Rochelle Gifts	
Baane Mini Imports	
Mini Gifts Distributors Ltd.	
Blauer See Auto, Co.	
Mini Wheels Co.	
Land of Toys Inc.	

논리연산자(AND, OR, NOT)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE quantityordered = 46
AND
productcode = 's24_3969'
;
```

- 결과

orderdetails (3r × 5c)					
orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber	
10,124	s24_3969	46	36.11	4	
10,242	s24_3969	46	36.52	1	
10,368	s24_3969	46	36.52	3	

논리연산자(AND, OR, NOT)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE quantityordered = 46
OR
productcode = 's24_3969'
;
```

- 결과

orderdetails (126r x 5c)					
orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber	
10,100	S24_3969	49	35.29	1	
10,101	S24_2022	46	44.35	2	
10,103	S18_3320	46	86.31	16	
10,109	S18_3232	46	160.87	5	
10,110	S24_2887	46	112.74	10	
10,110	S24_3969	48	35.29	5	
10,115	S12_4473	46	111.39	5	
10,115	S18_2238	46	140.81	4	
10,119	S10_4757	46	112.88	11	
10,120	S10_4698	46	158.8	2	
10,120	S18_2625	46	57.54	4	
10,123	S18_2870	46	114.84	3	
10,124	S24_3969	46	36.11	4	
10,126	S18_2957	46	61.84	14	
10,127	S12_1108	46	193.25	2	

논리연산자(AND, OR, NOT)

- AND 와 OR을 같이 사용하는 경우 우선순위는 AND에 있음
- OR 가 우선적으로 실행되기 위해서는 소괄호 활용()

논리연산자(AND, OR, NOT)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE priceEach = 35.29
AND
productcode = 's24_3969'
OR
quantityordered = 46
;
```

- 결과

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10,100	s24_3969	49	35.29	1
10,110	s24_3969	48	35.29	5
10,280	s24_3969	33	35.29	14
10,420	s24_3969	15	35.29	3
10,394	s24_2840	46	35.36	6
10,124	s24_3969	46	36.11	4
10,242	s24_3969	46	36.52	1
10,368	s24_3969	46	36.52	3
10,190	s32_2206	46	38.62	1
10,215	s18_4668	46	42.76	1

논리연산자(AND, OR, NOT)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE (priceEach = 35.29
AND
productcode = 's24_3969'
)OR
quantityordered = 46
;
```

- 결과

orderdetails (108r x 5c)					
orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber	
10,100	524_3969	49	35.29	1	
10,110	524_3969	48	35.29	5	
10,280	524_3969	33	35.29	14	
10,420	524_3969	15	35.29	3	
10,394	524_2840	46	35.36	6	
10,124	524_3969	46	36.11	4	
10,242	524_3969	46	36.52	1	
10,368	524_3969	46	36.52	3	
10,190	532_2206	46	38.62	1	
10,215	518_4668	46	42.76	1	

- AND의 우선순위가 앞서기 때문에 같은 결과 출력

논리연산자(AND, OR, NOT)

- SQL Code

```
SELECT *
FROM orderdetails
WHERE priceEach = 35.29
AND
(productcode = 's24_3969'
OR
quantityordered = 46 )
;
```

- 결과

orderdetails (4r x 5c)					
orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber	
10,100	s24_3969	49	35.29	1	
10,110	s24_3969	48	35.29	5	
10,280	s24_3969	33	35.29	14	
10,420	s24_3969	15	35.29	3	

- 괄호를 통해서 OR 조건을 우선순위로 하면 다른 결과가 출력

논리연산자(AND, OR, NOT) 실습

- 문제 2: orders 테이블에서 status가 'Shipped' 이거나 'In Process'인 주문의 orderNumber를 조회하세요.

- 정답

orders (309r x 1c)	
orderNumber	
10,100	
10,101	
10,102	
10,103	
10,104	
10,105	
10,106	
10,107	

논리연산자(AND, OR, NOT) 실습

- 문제 3: products 테이블에서 재고(quantityInStock)가 100개 미만이거나 500개 초과인 제품의 productName을 조회하세요.

- 정답

products (107r × 1c)	
productName	
1969 Harley Davidson Ultimate Chopper	
1952 Alpine Renault 1300	
1996 Moto Guzzi 1100i	
2003 Harley-Davidson Eagle Drag Bike	
1972 Alfa Romeo GTA	
1962 LanciaA Delta 16V	
1968 Ford Mustang	
2001 Ferrari Enzo	
1958 Setra Bus	
2002 Suzuki XREO	

논리연산자(AND, OR, NOT) 실습

- 문제 4: products 테이블에서 제품의 가격(buyPrice)이 50보다 크고 재고(quantityInStock)가 100보다 작은 제품의 productName을 조회하세요.

- 정답

products (1r × 1c)	
productName	
1968 Ford Mustang	

LIKE

- **LIKE** 함수는 문자열에서 원하는 문자가 포함되어 있는지를 검색
- **SQL Code**
- **SELECT * FROM table**
- **WHERE LIKE '%CARS'** : CARS 로 끝나는 데이터 검색
- **WHERE LIKE 'CARS%'** : CARS 로 시작하는 데이터 검색
- **WHERE LIKE '%CARS'%** : CARS를 포함하는 데이터 검색

LIKE

- Product table

classicmodels.products: 110 행 (총) (대략적)

» 더

productCode	productName	productLine	productScale	productVendor	productDescription	quantityInStock	buyPrice	MSRP
S10_1678	1969 Harley Davidson Ultimate Chopper	Motorcycles	1:10	Min Lin Diecast	This replica features working kickstand, front suspension...	7,933	48.81	95.7
S10_1949	1952 Alpine Renault 1300	Classic Cars	1:10	Classic Metal Creations	Turnable front wheels; steering function; detailed interio...	7,305	98.58	214.3
S10_2016	1996 Moto Guzzi 1100i	Motorcycles	1:10	Highway 66 Mini Classics	Official Moto Guzzi logos and insignias, saddle bags locat...	6,625	68.99	118.94
S10_4698	2003 Harley-Davidson Eagle Drag Bike	Motorcycles	1:10	Red Start Diecast	Model features, official Harley Davidson logos and insigni...	5,582	91.02	193.66
S10_4757	1972 Alfa Romeo GTA	Classic Cars	1:10	Motor City Art Classics	Features include: Turnable front wheels; steering functio...	3,252	85.68	136.0
S10_4962	1962 LanciaA Delta 16V	Classic Cars	1:10	Second Gear Diecast	Features include: Turnable front wheels; steering functio...	6,791	103.42	147.74
S12_1099	1968 Ford Mustang	Classic Cars	1:12	Autoart Studio Design	Hood, doors and trunk all open to reveal highly detailed i...	68	95.34	194.57
S12_1108	2001 Ferrari Enzo	Classic Cars	1:12	Second Gear Diecast	Turnable front wheels; steering function; detailed interio...	3,619	95.59	207.8
S12_1666	1958 Setra Bus	Trucks and Buses	1:12	Welly Diecast Productions	Model features 30 windows, skylights & glare resistant gl...	1,579	77.9	136.67
S12_2823	2002 Suzuki XREO	Motorcycles	1:12	Unimax Art Galleries	Official logos and insignias, saddle bags located on side o...	9,997	66.27	150.62
S12_3148	1969 Corvair Monza	Classic Cars	1:18	Welly Diecast Productions	1:18 scale die-cast about 10" long doors open, hood ope...	6,906	89.14	151.08

- Productline column을 보면 다양한 상품이 있는 것 같다.
- 어떤 상품이 있는지 확인하기 위해 distinct 를 활용

LIKE

- SQL Code

```
SELECT DISTINCT  
productline  
FROM products  
;
```

- 결과

products (7r x 1c)	
productline	
Classic Cars	
Motorcycles	
Planes	
Ships	
Trains	
Trucks and Buses	
Vintage Cars	

- 결과는 총 7개의 제품이며 자동차 종류는 Classic Cars와 Vintage Cars로 두 종류

LIKE

- SQL Code

```
SELECT *  
FROM products  
WHERE productline = 'vintage cars' OR productline = 'classic cars'  
;
```

```
SELECT *  
FROM products  
WHERE productline LIKE "%cars"  
;
```

- 위 두 코드는 서로 같은 결과를 출력.
- Like '%cars%' 도 가능

LIKE 실습

- 문제 5: employees 테이블에서 jobTitle에 'Sales'라는 단어가 포함된 직원의 firstName과 lastName을 조회하세요.

- 정답

employees (20r x 2c)	
firstName	lastName
Mary	Patterson
William	Patterson
Anthony	Bow
Leslie	Jennings
Leslie	Thompson
Julie	Firrelli
Steve	Patterson
Foon Yue	Tseng

BETWEEN


- **BETWEEN** 함수는 A 이상 B 이하의 데이터를 출력
- **SQL Code**
- `SELECT * FROM table`
`WHERE column1 BETWEEN 10 AND 30`
-> column1 이 10 이상 30 이하인 모든 데이터 출력

BETWEEN

- SQL Code

```
SELECT *
FROM orders
WHERE orderdate
BETWEEN '2003-01-01' AND '2003-01-31'
;
```

- 결과

orders (5r × 7c)		
orderNumber		orderDate
10,100		2003-01-06
10,101		2003-01-09
10,102		2003-01-10
10,103		2003-01-29
10,104		2003-01-31

- BETWEEN A AND B : A 와 B
포함의 데이터 출력

BETWEEN

- SQL Code

```
SELECT *  
FROM orders  
WHERE orderdate BETWEEN '2003-01-01' AND '2003-01-31'  
;
```

```
SELECT *  
FROM orders  
WHERE orderdate >= '2003-01-01' AND orderdate <= '2003-01-31'  
;
```

- 위 두 코드는 서로 같은 결과를 출력.

BETWEEN 실습

- 문제 6: orderdetails 테이블에서 주문 수량(quantityOrdered)이 10개 이상 50개 이하인 주문의 orderNumber를 조회하세요.

- 정답

orderdetails (2,928r × 1c)	
orderNumber	
10,100	
10,100	
10,100	
10,100	
10,101	
10,101	
10,101	
10,101	

NOT BETWEEN

- NOT BETWEEN 함수는 A 이상 B 이하를 제외한 데이터를 출력
- SQL Code
- ```
SELECT * FROM table
WHERE column1 NOT BETWEEN 10 AND 30
```

-> column1 이 10 이상 30 이하 제외 모든 데이터 출력

## NOT BETWEEN

- SQL Code

```
SELECT *
FROM orders
WHERE orderdate NOT BETWEEN '2003-01-01' AND '2003-01-31'
;
```

```
SELECT *
FROM orders
WHERE orderdate < '2003-01-01' OR orderdate > '2003-01-31'
;
```

- 위 두 코드는 서로 같은 결과를 출력.

## IN

- **IN** 은 특정 값이 있을 때 조회
- **SQL Code**
- `SELECT * FROM table`  
`WHERE column1 IN (10, 20, 30)`  
-> column1 이 10, 20, 30 인 데이터 조회


## IN

- SQL Code

```
SELECT *
FROM orders
WHERE orderdate IN ('2003-
02-11', '2003-02-17')
;
```

- WHERE orderdate = '2003-02-11' OR orderdate = '2003-02-17' 과 같은 결과

- 결과

| orders (2r x 7c) |                                                                                     |            |              |             |
|------------------|-------------------------------------------------------------------------------------|------------|--------------|-------------|
| orderNumber      |  | orderDate  | requiredDate | shippedDate |
| 10,105           |                                                                                     | 2003-02-11 | 2003-02-21   | 2003-02-12  |
| 10,106           |                                                                                     | 2003-02-17 | 2003-02-24   | 2003-02-21  |

## IN 실습

- 문제 7: customers 테이블에서 국가(country)가 'USA', 'Canada', 'France' 중 하나인 고객의 customerName을 조회하세요.

- 정답

| customers (51r × 1c)         |  |
|------------------------------|--|
| customerName                 |  |
| Atelier graphique            |  |
| Signal Gift Stores           |  |
| La Rochelle Gifts            |  |
| Mini Gifts Distributors Ltd. |  |
| Mini Wheels Co.              |  |
| Land of Toys Inc.            |  |
| Saveley & Henriot, Co.       |  |
| Muscle Machine Inc           |  |
| Diecast Classics Inc.        |  |



## NOT IN

- NOT IN 은 특정 값이 포함되지 않은 데이터 조회
- SQL Code
- ```
SELECT * FROM table  
WHERE column1 NOT IN (10, 20, 30)
```

-> column1 이 10, 20, 30 이 아닌 데이터 조회

NOT IN

- SQL Code

```
SELECT *  
FROM employees  
WHERE officecode not IN (1,2,3)  
;
```

```
SELECT *  
FROM employees  
WHERE officecode <> 1 and officecode <> 2 and officecode <>3  
;
```

- 위 두 코드는 서로 같은 결과를 출력.

IS NULL / IS NOT NULL

- **IS NULL** : COLUMN 의 값이 NULL 인 데이터 조회
- **IS NOT NULL** : COLUMN 의 값이 NULL 이 아닌 값 조회

```
SELECT *
FROM orders
WHERE comments IS NULL
;
```

orderNumber	orderDate	requiredDate	shippedDate	status	comments
10,100	2003-01-06	2003-01-13	2003-01-10	Shipped	(NULL)
10,102	2003-01-10	2003-01-18	2003-01-14	Shipped	(NULL)
10,103	2003-01-29	2003-02-07	2003-02-02	Shipped	(NULL)

```
SELECT *
FROM orders
WHERE comments IS not NULL
;
```

orderNumber	orderDate	requiredDate	shippedDate	status	comments
10,101	2003-01-09	2003-01-18	2003-01-11	Shipped	Check on availability.
10,107	2003-02-24	2003-03-03	2003-02-26	Shipped	Difficult to negotiate with customer. We need more mark...
10,109	2003-03-10	2003-03-19	2003-03-11	Shipped	Customer requested that FedEx Ground is used for this s...

IS NULL 실습

- 문제 8: employees 테이블에서 상사(reportsTo)가 지정되지 않은 직원의 firstName과 lastName을 조회하세요.

- 정답

employees (1r × 2c)	
firstName	lastName
Diane	Murphy

7.

ORDER BY구문

ORDER BY

- **ORDER BY** : SQL 쿼리의 결과를 특정 컬럼 또는 여러 컬럼을 기준으로 정렬. 기본적으로 오름차순(ASC)으로 정렬되며, 내림차순(DISC)으로 정렬하려면 DESC 키워드를 사용

```
SELECT *  
FROM A  
WHERE ~~  
ORDER BY 컬럼명1 desc, 컬럼명2 asc, ...
```

- ORDER BY 는 WHERE 절(생략 가능) 다음에 사용.

ORDER BY ASC

- SQL Code

```
SELECT LASTNAME,  
FIRSTNAME, OFFICECODE  
FROM employees  
ORDER BY OFFICECODE  
LIMIT 8;
```

- 결과

LASTNAME	FIRSTNAME	OFFICECODE
Murphy	Diane	1
Patterson	Mary	1
Firrelli	Jeff	1
Bow	Anthony	1
Jennings	Leslie	1
Thompson	Leslie	1
Firrelli	Julie	2
Patterson	Steve	2

- 오름차순 ACS는 생략 가능

ORDER BY DESC

- SQL Code

```
SELECT LASTNAME,  
FIRSTNAME, OFFICECODE  
FROM employees  
ORDER BY OFFICECODE DESC  
LIMIT 8;
```

- 결과

LASTNAME	FIRSTNAME	OFFICECODE
Bott	Larry	7
Jones	Barry	7
Patterson	William	6
Fixter	Andy	6
Marsh	Peter	6
King	Tom	6
Nishi	Mami	5
Kato	Yoshimi	5

- OFFICECODE 로 내림차순 정렬

ORDER BY 다중정렬

- SQL Code

```
SELECT *
FROM orderdetails
ORDER BY
quantityordered DESC ,
priceEach asc
;
```

- 결과

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10,405	S12_4675	97	115.16	5
10,404	S18_3278	90	67.54	6
10,401	S700_2466	85	98.72	10
10,404	S12_3990	77	67.03	4
10,401	S700_3167	77	73.6	9
10,407	S24_2766	76	81.78	6
10,405	S24_3856	76	127.79	3
10,407	S18_1749	76	141.1	2

- quantityordered 로 먼저 내림차순 정렬,
이후 priceEach로 오름차순 정렬

ORDER BY 다중정렬

- SQL Code

```
SELECT *
FROM orderdetails
ORDER BY
quantityordered DESC ,
priceEach desc
;
```

- 결과

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10,405	512_4675	97	115.16	5
10,404	518_3278	90	67.54	6
10,401	5700_2466	85	98.72	10
10,401	5700_3167	77	73.6	9
10,404	512_3990	77	67.03	4
10,407	518_1749	76	141.1	2
10,405	524_3856	76	127.79	3
10,407	524_2766	76	81.78	6

- quantityordered 로 먼저 내림차순 정렬,
이후 priceEach로 내림차순 정렬

ORDER BY 실습

- 문제 9: products 테이블에서 제품 이름(productName)을 조회하되, 가격(buyPrice)을 기준으로 내림차순으로 정렬하세요.

정답

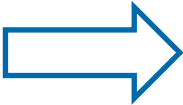
products (110r × 1c)	
productName	
1962 LanciaA Delta 16V	
1998 Chrysler Plymouth Prowler	
1952 Alpine Renault 1300	
1956 Porsche 356A Coupe	
2001 Ferrari Enzo	
1968 Ford Mustang	
1995 Honda Civic	
1970 Triumph Spitfire	
2003 Harley-Davidson Eagle Drag Bike	
1969 Corvair Monza	

8.

GROUP BY구문

GROUP BY

종목명	섹터	상장주식수
삼성전자	반도체	100
셀트리온	바이오	150
SK하이닉스	반도체	200
삼성바이오로직스	바이오	300
NAVER	IT	400
카카오	IT	500



지수	섹터별 상장주식수
반도체	300
바이오	450
반도체	900

- GROUP BY 절은 한 개 이상의 컬럼을 기준으로 결과를 그룹화
- 주로 SUM(), AVG(), COUNT(), MAX(), MIN() 등의 집계 함수와 결합되어 사용

GROUP BY count

- SQL Code

```
SELECT productline,  
count(productline)  
FROM products  
GROUP BY productline  
;
```

- 결과

products (7r × 2c)	
productline	count(productline)
Classic Cars	38
Motorcycles	13
Planes	12
Ships	9
Trains	3
Trucks and Buses	11
Vintage Cars	24

- Products 테이블에서 productLine 별 정보의 수

GROUP BY avg

- SQL Code

```
SELECT productline,  
avg(buyprice)  
FROM products  
GROUP BY productline  
;
```

- 결과

products (7r × 2c)	
productline	avg(buyprice)
Classic Cars	64.446316
Motorcycles	50.685385
Planes	49.629167
Ships	47.007778
Trains	43.923333
Trucks and Buses	56.329091
Vintage Cars	46.06625

- Products 테이블에서 productLine 평균 가격

GROUP BY sum

- SQL Code

```
SELECT productcode,
SUM(quantityordered)
FROM orderdetails
WHERE orderlinenumber = 1
group BY productcode
;
```

- 결과

orderdetails (101r × 2c)	
productcode	SUM(quantityordered)
S10_1678	254
S10_1949	120
S10_2016	106
S10_4698	76
S10_4757	51
S10_4962	48
S12_1108	33
S12_1666	106

- orderdetails 테이블에서 orderlinenumber 가 1인 제품들 중에서 productcode별 주문 수량의 합

GROUP BY max

- SQL Code

```
SELECT productline,  
MAX(msrp)  
FROM products  
GROUP BY productline  
;
```

- 결과

products (7r × 2c)	
productline	MAX(msrp)
Classic Cars	214.3
Motorcycles	193.66
Planes	157.69
Ships	122.89
Trains	100.84
Trucks and Buses	136.67
Vintage Cars	170.0

- Products 테이블에서 productLine 별 최고 msrp(권장 판매 가격)

GROUP BY 실습

- 문제 10: orders 테이블에서 각 상태(status)별로 주문 개수를 구하세요.

- 정답

orders (6r × 2c)	
status	OrderCount
Shipped	303
Resolved	4
Cancelled	6
On Hold	4
Disputed	3
In Process	6

GROUP BY 실습

- 문제 11: orderdetails 테이블에서 각 제품 코드(productCode)별로 주문된 총 수량(quantityOrdered)을 구하세요.

정답

productCode	TotalOrdered
S10_1678	1,057
S10_1949	961
S10_2016	999
S10_4698	985
S10_4757	1,030
S10_4962	932
S12_1099	933
S12_1108	1,019

GROUP BY 실습

- 문제 12: products 테이블에서 제품 라인(productLine)별 제품 개수를 조회하세요.

- 정답

products (7r × 2c)	
productLine	ProductCount
Classic Cars	38
Motorcycles	13
Planes	12
Ships	9
Trains	3
Trucks and Buses	11
Vintage Cars	24

GROUP BY 실습

- 문제 13. "products" 테이블에서 각 제품 라인("productLine")별로 제품의 최대 가격("buyPrice")과 최소 가격("buyPrice")을 계산하세요.

- 정답

products (7r x 3c)		
productLine	maxPrice	minPrice
Classic Cars	103.42	15.91
Motorcycles	91.02	24.14
Planes	77.27	29.34
Ships	82.34	33.3
Trains	67.56	26.72
Trucks and Buses	84.76	24.92
Vintage Cars	86.7	20.61

GROUP BY 실습

- 문제 14. "customers" 테이블에서 각 고객 도시("city")별로 평균 크레딧 한도("creditLimit") 상위 5개를 조회하세요.

- 정답

customers (5r × 2c)	
city	averageCreditLimit
San Rafael	210,500.0
Genève	141,300.0
Manchester	136,800.0
Lyon	123,900.0
Reggio Emilia	121,400.0

GROUP BY 실습

- 문제 15.
"orderdetails"
테이블에서 주문
번호("orderNumber")
별로 총 주문 총액
("priceEach" *
"quantityOrdered")
상위 5개를
계산하세요.

- 정답

orderdetails (5r × 2c)	
orderNumber	totalOrderPrice
10,165	67,392.85
10,287	61,402.0
10,310	61,234.67
10,212	59,830.55
10,207	59,265.14

GROUP BY 실습

- 문제 16. "customers" 테이블에서 각 국가("country")별로 고객 수가 많은 상위 5개를 조회하세요.

- 정답

customers (5r × 2c)	
country	numberOfCustomers
USA	36
Germany	13
France	12
Spain	7
Australia	5

GROUP BY 실습

- 문제 17. "products" 테이블에서 productScale이 '1:10'인 제품 라인("productLine")별로 제품의 평균 가격("buyPrice")을 계산하세요.

- 정답

products (2r × 2c)	
productLine	averagePrice
Classic Cars	95.893333
Motorcycles	69.606667