

## 융합 미디어 과제 4장

21100248장성빈

1.(1)15 (2)16 (3)-1 (4)15 (5)2 (6)3 (7)9 (8)16 (9)-1 (10)15 (11)c  
(12)birthday (13)\*\*\*\*\*page1 (14)BLACK (15)page1\*\*\*\*\* (16)CDEF  
(17)AAOO (18)BIRTHDAY (19)65 (20)8BYTE (21)2,24 (22)4,30  
(23)2025.6.14. (24)2025 (25)123 (26)2024-05-12 (27)12.3 (28)aa  
(29)123 (30)123

2.

1.모든행 출력

2.1 | 10000

2 | 20000

3 | 0

3. bookid = 3

4. 아무것도 안 나옴.

5. NULL

6. NULL, NULL, 0

7. 3, 2

8. SUM = 30000, AVG = 15000

3.

1. 2

2. A:12,10

B:NULL,12

3.A:NULL,10,12

COUNT:2,1,1

4.

1.전체 행이 그대로 출력됨

2.테이블에서 상위 5개만 출력

3.테이블 기준 가격이 가장 낮은 5개가 출력됨.

4.행번호를 하나씩 증가시키며 5개만 출력

5.초기화한 변수로 행번호를 생성 + 5개까지

6.가격 순으로 정렬한 후 행번호 추가, 5개까지 출력

7. 가격 순 정렬 + ROWNUM 추가 후, 그 중 3개만 출력

5.

1.고객별 주소와 총 구매금액 확인

2.고객별 평균 구매금액 확인

3.custid ≤ 3인 고객들의 구매 금액 총합

6. 2번

7.

```
1.SELECT name
   FROM Customer
  WHERE address LIKE '대한민국%'
 AND custid NOT IN (
    SELECT custid FROM Orders
  );
```

```
2.SELECT name
   FROM Customer c1
  WHERE address LIKE '대한민국%'
 AND NOT EXISTS (
    SELECT *
    FROM Orders o
   WHERE o.custid = c1.custid
  );
```

```
3.SELECT c.name
   FROM Customer c
  LEFT OUTER JOIN Orders o ON c.custid = o.custid
  WHERE c.address LIKE '대한민국%'
 AND o.custid IS NULL;
```

8.

회원번호	등급
----	--
1	1
2	2

9. 2,3,4,5,1,6

10.

1. 보안 강화
2. 복잡한 쿼리 단순화
3. 논리적 독립성 유지

11. 5000

12.

```
1.CREATE VIEW highorders AS
SELECT
    b.bookid,
    b.bookname,
    c.name AS customer_name,
    b.publisher,
    o.saleprice
FROM
    Orders o
    JOIN Book b ON o.bookid = b.bookid
    JOIN Customer c ON o.custid = c.custid
WHERE
    o.saleprice >= 20000;
```

```
2.SELECT
    bookname,
    customer_name
FROM
    highorders;
```

```
3.DROP VIEW highorders;
```

```
CREATE VIEW highorders AS
SELECT
    b.bookid,
    b.bookname,
    c.name AS customer_name,
    b.publisher
FROM
    Orders o
    JOIN Book b ON o.bookid = b.bookid
    JOIN Customer c ON o.custid = c.custid
WHERE
    o.saleprice >= 20000;
```

```
SELECT
    bookname,
    customer_name
FROM
    highorders;
```