

연습문제

01 다음 내장 함수의 결과를 적으시오.

- ABS(-15)
- CEIL(15.7)
- COS(3.14159)
- FLOOR(15.7)
- LOG(10,100)
- MOD(11,4)
- POWER(3,2)
- ROUND(15.7)
- SIGN(-15)
- TRUNCATE(15.7, 0)
- CHAR(67 USING utf8)
- CONCAT('HAPPY', 'Birthday')
- LOWER('Birthday')
- LPAD('Page 1', 15, '*')
- REPLACE('JACK', 'J', 'BL')
- RPAD('Page 1', 15, '*')
- SUBSTR('ABCDEFGH', 3, 4)
- TRIM(LEADING 0 FROM '00AA00')
- UPPER('Birthday')
- ASCII('A')
- LENGTH('Birthday')
- ADDDATE('2024-02-14', INTERVAL 10 DAY)
- LAST_DAY(SYSDATE())
- NOW()
- DATE_FORMAT(SYSDATE(), '%Y')
- CONCAT(123)
- STR_TO_DATE('12 05 2024', '%d %m %Y')
- CAST('12.3' AS DECIMAL(3,1))
- IF(1=1, 'aa', 'bb')
- IFNULL(123, 345)
- IFNULL(NULL, 123)

02 Mybook 테이블을 생성하고 NULL에 관한 다음 SQL 문에 답하시오. 또한 질의 결과를 보면 서 NULL에 대한 개념도 정리해 보시오.

Mybook

bookid	price
1	10000
2	20000
3	NULL

- (1) SELECT *
FROM Mybook;

연습문제

- (2) SELECT bookid, IFNULL(price, 0)
FROM Mybook;
- (3) SELECT *
FROM Mybook
WHERE price IS NULL;
- (4) SELECT *
FROM Mybook
WHERE price='';
- (5) SELECT bookid, price+100
FROM Mybook;
- (6) SELECT SUM(price), AVG(price), COUNT(*)
FROM Mybook
WHERE bookid >= 4;
- (7) SELECT COUNT(*), COUNT(price)
FROM Mybook;
- (8) SELECT SUM(price), AVG(price)
FROM Mybook;

03 다음 SQL 문을 비어 있는 릴레이션 R(A: int, B: int)에 실행했을 때 결과 값을 적으시오.

(SQL)

```
INSERT INTO R VALUES(NULL, 10);
INSERT INTO R VALUES(12, NULL);
INSERT INTO R VALUES(NULL, NULL);
INSERT INTO R VALUES(10, 12);
```

- (1) SELECT COUNT(A) FROM R;
- (2) SELECT * FROM R WHERE A IN(10, 12, NULL);
- (3) SELECT A, COUNT(*) FROM R GROUP BY A;

연습문제

01 다음 내장 함수의 결과를 적으시오.

- ABS(-15) 15
- CEIL(15.7) 16
- COS(3,14159) -1
- FLOOR(15.7) 15
- LOG(10,100) 2
- MOD(11,4) 3
- POWER(3,2) 9
- ROUND(15.7) 16
- SIGN(-15) -1
- TRUNCATE(15.7, 0) 15
- CHAR(67 USING utf8) C
- CONCAT('HAPPY', 'Birthday') HAPPY Birthday
- LOWER('Birthday') birthday
- LPAD('Page 1', 15, '*') * (15) page 1
- REPLACE('JACK', 'J', 'BL') BLACK
- RPAD('Page 1', 15, '*') page 1 (15)
- SUBSTR('ABCDEFG', 3, 4) CDEF
- TRIM(LEADING 0 FROM '00AA00') AA00
- UPPER('Birthday') BIRTHDAY
- ASCII(A) 65
- LENGTH('Birthday') 8
- ADDDATE(2024-02-14, INTERVAL 10 DAY) 2024-02-24
- LAST_DAY(SYSDATE()) 2025-06-30
- NOW() 2025-06-16 14:23:00
- DATE_FORMAT(SYSDATE(), '%Y') '2025'
- CONCAT(123) 123
- STR_TO_DATE('12 05 2024', '%d %m %Y') 2024-05-12
- CAST('12.3' AS DECIMAL(3,1)) 12.3
- IF(1=1, 'aa', 'bb') aa
- IFNULL(123, 345) 123
- IFNULL(NULL, 123) 123

02 Mybook 테이블을 생성하고 NULL에 관한 다음 SQL 문에 답하시오. 또한 질의의 결과를 보면 서 NULL에 대한 개념도 정리해 보시오.

Mybook

bookid	price
1	10000
2	20000
3	NULL

(1) SELECT *
FROM Mybook;

bookid
1
2
3
price
10000
20000
NULL

연습문제

- (2) SELECT bookid, IFNULL(price, 0)
FROM Mybook;
- (3) SELECT *
FROM Mybook
WHERE price IS NULL;
- (4) SELECT *
FROM Mybook
WHERE price='';
- (5) SELECT bookid, price+100
FROM Mybook;
- (6) SELECT SUM(price), AVG(price), COUNT(*)
FROM Mybook
WHERE bookid >= 4;
- (7) SELECT COUNT(*), COUNT(price)
FROM Mybook;
- (8) SELECT SUM(price), AVG(price)
FROM Mybook;

bookid IFNULL
1 3
2 10000
3 20000
4 0

bookid price
3 NULL

bookid price
1 3
2 10000
3 20000
4 0

bookid price+100
1 103
2 20100
3 20300
4 100

SUM(price), AVG(price), COUNT(*)
30000 15000 3

COUNT(*) COUNT(price)
3 2

SUM(price) AVG(price)
30000 15000

03 다음 SQL 문을 비어 있는 릴레이션 R(A: int, B: int)에 실행했을 때 결과 값을 적으시오.

(SQL)

INSERT INTO R VALUES(NULL, 10);
INSERT INTO R VALUES(12, NULL);
INSERT INTO R VALUES(NULL, NULL);
INSERT INTO R VALUES(10, 12);

A B
12 NULL
10 12

- (1) SELECT COUNT(A) FROM R;
- (2) SELECT * FROM R WHERE A IN(12, 10, NULL);
- (3) SELECT A, COUNT(*) FROM R GROUP BY A;

A COUNT(*)
NULL 2
10 1
12 1

04 마당서점 데이터베이스를 이용해 데이터 출력 개수 및 MySQL의 행번호를 처리하는 다음 SQL 문의 결과를 적으시오.

- (1) SELECT bookid, bookname, price
FROM Book; -- 모든 책 보기 **모든 책 출력**
- (2) SELECT bookid, bookname, price
FROM Book
LIMIT 5; -- 임의로 5개만 보기 **Book 상위 5개의 책만 표시**
- (3) SELECT bookid, bookname, price
FROM Book
ORDER BY price
LIMIT 5; -- price 순으로 5개만 보기 **가장 저렴한 책 5권**
- (4) SET @RNUM:=0;
SELECT bookid, bookname, price, @RNUM:= @RNUM + 1 AS ROWNUM
FROM Book
WHERE @RNUM < 5; -- 임의로 5개만 행번호와 함께 보기 **행번호 붙여서 5개**
- (5) SELECT bookid, bookname, price, @RNUM:= @RNUM + 1 AS ROWNUM
FROM Book, (SELECT @RNUM:=0) R
WHERE @RNUM < 5; -- 임의로 5개만 행번호와 함께 보기 **book 테이블 행번호 1~5**
- (6) SELECT bookid, bookname, price, @RNUM:= @RNUM + 1 AS ROWNUM
FROM (SELECT * FROM Book ORDER BY price) b,
(SELECT @RNUM:=0) R
WHERE @RNUM < 5; -- 행번호와 함께 price 순으로 5개만 보기 **가격 기준, 저렴한 책 5권에서 ROWNUM 붙임**
- (7) SELECT bookid, bookname, price, @RNUM:= @RNUM + 1 AS ROWNUM
FROM (SELECT * FROM Book ORDER BY price) b,
(SELECT @RNUM:=0) R
WHERE @RNUM < 5
LIMIT 5; -- 행번호와 함께 price 순으로 3개만 보기 **가장 저렴한 책 3권, ROWNUM 포함**

05 부속질문에 관한 다음 SQL 문을 수행해 보고, 어떤 질의에 대한 답인지 설명하시오.

- (1) SELECT custid, (SELECT address
FROM Customer cs
WHERE cs.custid = od.custid) 'address'
SUM(saleprice) 'total'
FROM Orders od
GROUP BY od.custid;
외래키 관계로 Customer 테이블에서 주소 찾기.
- (2) SELECT cs, name, s
FROM (SELECT custid, AVG(saleprice) s
FROM Orders
GROUP BY custid) od, Customer cs
WHERE cs.custid = od.custid;
평균 구매 금액을 연결해
- (3) SELECT SUM(saleprice) 'total'
FROM Orders od
WHERE EXISTS (SELECT *
FROM Customer cs
WHERE custid <= 3 AND cs.custid = od.custid);
해당고객이 존재하는지 확인

06 릴레이션 Emp, Dept가 다음과 같이 정의되어 있다. Emp의 deptno는 Dept의 deptno를 참조하는 외래키다. 사원이 1명도 없는 부서(deptno)를 검색하는 질의를 다양한 방법으로 작성했을 때 가장 거리가 먼 질의는 어느 것인가?

(릴레이션)

Emp(empno, ename, job, mgr, hiredate, sal, comm, deptno)
Dept(deptno, dname, loc)

- (1) SELECT deptno FROM Dept
WHERE deptno NOT IN (SELECT deptno FROM Emp);
- (2) SELECT deptno FROM Dept A
WHERE NOT EXISTS
(SELECT * FROM Emp B WHERE A.deptno = B.deptno);

연습문제

```
(3) SELECT B.deptno
FROM EMP A RIGHT OUTER JOIN Dept B ON A.deptno = B.deptno
WHERE empno IS NULL;
✓ SELECT deptno FROM Dept
WHERE deptno != ANY (SELECT deptno FROM EMP);
```

07 다음 세 개의 질의는 각각 부속질의, EXISTS, 조인을 사용하여 '대한민국에 거주하면서 도서를 주문한 적이 있는 고객의 이름'을 찾는 질의다. 세 질의의 결과가 같은지 확인하시오. 이 결과를 참조하여 '대한민국에 거주하면서 도서를 주문한 적이 없는 고객의 이름'을 세 가지 방법으로 작성하시오.

(1) 부속질의 사용

```
SELECT name
FROM Customer
WHERE address LIKE '대한민국%' AND
name IN (SELECT name
FROM Customer
WHERE custid IN (SELECT custid FROM Orders));
```

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

(2) EXISTS 사용

```
SELECT name
FROM Customer c1
WHERE address LIKE '대한민국%' AND
EXISTS (SELECT name
FROM Customer
WHERE custid IN (SELECT custid FROM Orders)
AND c1.name LIKE name);
```

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

(3) 조인 사용

```
SELECT c1.name
FROM Customer c1, Customer c2
WHERE c1.name=c2.name
AND c1.address LIKE '대한민국%'
```

```
SELECT C.name
FROM Customer C
LEFT JOIN Orders O ON C.custid=O.custid
WHERE C.address LIKE '대한민국%'
AND O.custid IS NULL
```

```
AND c2.name IN (SELECT name
FROM Customer
WHERE custid IN (SELECT custid FROM Orders));
```

08 테이블 R, S에 대하여 다음 SQL 문을 수행한 결과를 적으시오.

R		S			
회원번호	AMT	등급	VMIN	VMAX	
1	60000	1	10001	100000	
2	4000	2	1000	10000	
1	3000	3	100	999	

```
(SQL)
SELECT A.회원번호, B.등급
FROM (SELECT 회원번호, SUM(amt)
FROM R GROUP BY 회원번호) A, S
WHERE A.amt BETWEEN B.vmin AND B.vmax
```

회원번호 등급
10001 VMIN
1000 VMAX
10000 VMIN

09 다음 SQL 문의 실행 순서를 번호순으로 적으시오.

```
(SQL)
① SELECT deptno, COUNT(empno)
② FROM Emp
③ WHERE sal >= 500
④ GROUP BY deptno
⑤ HAVING COUNT(empno) > 2
⑥ ORDER BY deptno;
```

① → ② → ③ → ④
→ ⑤ → ⑥

10 뷰의 장점 세 가지를 설명하시오.

복합 데이터
SQL 간단화
논리적 독립성

11 릴레이션 R(A, B)에 대한 뷰(view)가 다음과 같이 정의될 때 SQL 문의 실행 결과를 적으시오.

R(A, B)

A	B
a	1000
a	2000
b	1000
b	NULL
NULL	3000
NULL	NULL

(뷰 생성 스크립트)

```
CREATE VIEW V
```

```
AS
```

```
SELECT * FROM R
```

```
WHERE A = 'a' OR A IS NULL;
```

(SQL)

```
SELECT SUM(B) 합계 FROM R WHERE b >= 2000;
```



12 마당서점 데이터베이스를 이용하여 다음에 해당하는 뷰를 작성하시오.

- (1) 판매가격이 20,000원 이상인 도서의 도서번호, 도서이름, 고객이름, 출판사, 판매가격을 보여주는 highorders 뷰를 생성하시오.
- (2) 생성한 뷰를 이용하여 판매된 도서의 이름과 고객의 이름을 출력하는 SQL 문을 작성하시오.
- (3) highorders 뷰를 변경하고자 한다. 판매가격 속성을 삭제하는 명령을 수행하시오. 삭제 후 (2)번 SQL 문을 다시 수행하시오.

(2번)

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

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

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

```
DROP VIEW highorders;
```

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