



49회 기출문제

1과목

■ 문제 1. 아래의 정규화 결과에 알맞은 정규화는?

정답: 2차 정규화

해설: 식별자 컬럼간의 종속성 해결

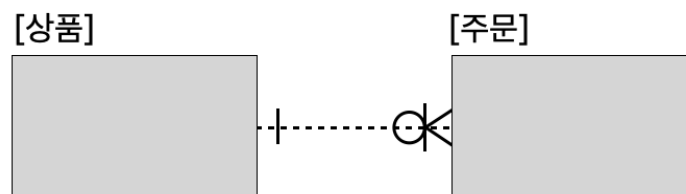
■ 문제 2. 속성이 가질 수 있는 데이터 값의 범위를 무엇이라 하는가?

정답: 도메인

■ 문제 3. 주식별자를 구성하는 속성에서 하나만이라도 제거되면 유일성을 만족하지 못하는 특성은 무엇인가?

정답: 최소성

■ 문제 4. 다음 보기 중 아래 ERD에 대한 설명으로 바르지 않은 것은?



정답: 3번 하나의 상품은 항상 주문을 가진다

해설: 하나의 상품은 주문을 가질수도 or 가지지 않을수도 있다. (O = or)



알아두기

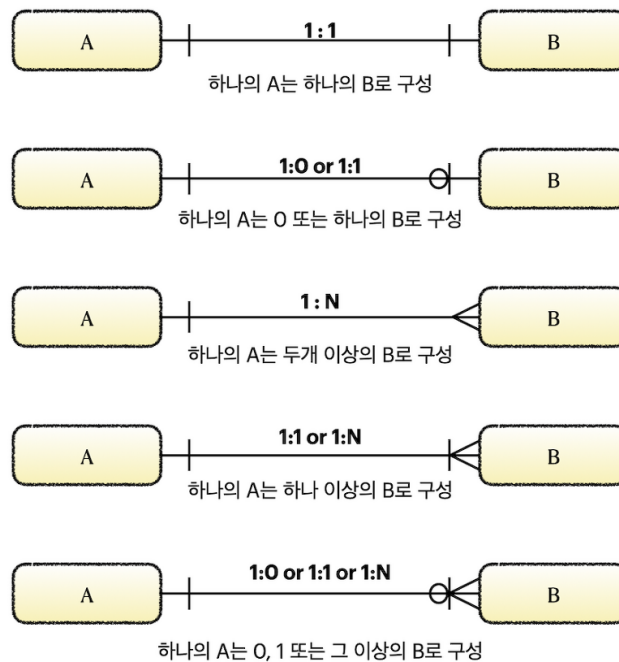
ERD 실선(Identifying) / 점선(Non-Identifying) 의 차이

1. 실선 (식별관계)

- 부모테이블의 PK가 외래키로써, 자식테이블의 PK에 포함이 되는 경우
- 부모, 자식 관계

2. 점선 (비식별관계)

- 부모테이블의 PK가 외래키로써, 자식테이블의 PK가 아닌 *일반속성* 이 되는 경우
- 부모, 자식 관계가 아닌 모든 경우를 표시함



두 엔터티간 관계에서 수행되는 경우의 수 (관계차수)

■ 문제 5. 도메인의 특징이 아닌 것 ?

정답: FK 제약 조건

■ 문제 6. 식별자 관계 틀린 선지 고르는 문제

정답: 4번. 식별자관계는 자식의 주식별자가 부모로부터 독립되어있고, 비식별자 관계는 자식의 주식별자가 부모에게 종속되어있다.

■ 문제 7. 보기에 해당하는 속성을 고르시오.



업무상 필요한 데이터 이외에 데이터 모델링을 위해 업무를 규칙화하기 위한 속성을 새로 만들거나 변형하여 정의하는 속성은 무엇인가?

정답: 설계 속성

■ 문제 8. 아래의 설명으로 알맞은 것은?



두 개 이상의 부모 엔터티로부터 발생되고 자주 내용이 바뀌거나 데이터량이 증가된다.
분석초기 단계에서는 잘 나타나지 않으며 상세 설계 단계나 프로세스와 상관모델링을 진행하면서 도출될 수 있다.

정답: 행위 엔터티

■ 문제 9. 아래의 식별자 분류에 대한 설명으로 적절한 것은?

정답: 주민등록번호

■ 단답형 1. 두 개의 엔터티간 관계에서 참여자의 수를 표현하는 것을 무엇이라 하는가?

정답: 관계차수 (Cardinality)

2과목 (객관식 35문제 / 단답형 5문제)

■ 문제 10. PLAYER 테이블에서 선수명과 팀명은 오름차순, 연봉은 내림차순으로 조회하는 SQL로 바른것은?

1) SELECT 선수명, 팀명, 연봉 FROM ORDER BY 선수명 DESC, 팀명 DESC, 연봉 ASC

2) SELECT 선수명, 팀명, 연봉 FROM ORDER BY 선수명 ASC, 팀명, 3 DESC

3) SELECT 선수명, 팀명, 연봉 FROM ORDER BY 선수명 ASC, 팀명 ASC, 연봉

4) SELECT 선수명, 팀명, 연봉 FROM ORDER BY 선수명, 팀명, DESC 연봉

해설: 선수명 ASC(오름차순), 팀명(default 오름차순), 3 DESC(3번째 열 내림차순)

■ 문제 11. SQL 의 분류로 적절하지 않은 것은?

정답: DML = RENAME

■ 문제 12. COMMIT을 하기 전 DML에 대한 설명으로 적절하지 않은 것은?

정답: 다른 유저가 데이터를 수정할 수 있다.

■ 문제 13. 아래의 테이블에서 순방향으로 계층형 쿼리를 구성할 때 올바른 것은?

정답: CONNECT BY PRIOR EMP_NO = MAAGER_NO

해설:

PRIOR 자식 = 부모 형태를 사용하면 계층 구조의 부모 데이터에서 자식 데이터(부모 -> 자식) 방향으로 전개하는 **순방향 전개**를 함.

💡알아두기

CONNECT BY PRIOR 자식 = 부모 (부모 → 자식) (순 ↓ 방향)

CONNECT BY PRIOR 부모 = 자식 (자식 → 부모) (역 ↑ 방향)

CONNECT BY 자식 = PRIOR 부모 (자식 → 부모) (역 ↑ 방향)

CONNECT BY 부모 = PRIOR 자식 (부모 → 자식) (순 ↓ 방향)

■ 문제 14. ORACLE DBMS 에서 아래의 NULL 처리 함수에 대한 설명으로 올바른 것은?

A : 컬럼값이 NULL이면 B를 출력

B : 두개의 컬럼 값이 같으면 NULL

C : NULL이 나올 때 까지

정답: NVL / NULLIF / COALESCE

■ 문제 15. 다음 중 아래 SQL의 실행결과로 가장 적절한 것은? (노랭이 2과목 55번)

[TBL]

ID

100

100

200

200

200

999

999

[SQL]

SELECT ID

FROM TBL

GROUP BY ID

HAVING COUNT(*) = 2

ORDER BY (CASE WHEN ID = 999 THEN 0 ELSE ID END)

1) 2)

ID ID

--- ---

100 999

999 100

3) 4)

ID ID

--- ---

100 999

200 200

999 100

정답: 2번 999, 100 순서로 나오는 SQL 결과

(CASE WHEN ID = 999 THEN 0 ELSE ID END) 정답 999(첫번째행), 100(두번째행)

■ 문제 16. 아래의 테이블에서 오류가 발생하는 SQL을 고르시오.

[TEAM]

ID COL1

1 A

2 B

3 C

4 D

[STADIUM]

ID	COL2
1	A
2	B

- 1) SELECT * FROM TEAM A INNER JOIN STADIUM B ON A.ID = B.ID
- 2) SELECT * FROM TEAM A INNER JOIN STADIUM B USING(ID)
- 3) SELECT * FROM TEAM A NATURAL JOIN STADIUM B
- 4) SELECT * FROM TEAM A INNER JOIN STADIUM B ON (ID)**

해설: **Error : ORA-00920: invalid relational operator**

ON절은 두개의 컬럼조건이 사용되어야 함.

■ 문제 17. 아래의 테이블에서 사원이 매니저 연봉이 3000 이하인 결과를 찾는 SQL이 아닌 것은?

정답: NOT EXISTS (SELECT 'X' FROM ... WHERE A.MGR_NO = B.EMP_NO WHERE ... > 3000)

해설:

NULL이 포함된 경우 NULL이 출력되는 조건절 NOT EXISTS를 구분하는 문제

■ 문제 18. 아래의 SQL 에 대한 결과로 올바른것을 고르시오.

```
SELECT ...  
FROM ...  
WHERE (A, B) IN (SELECT A, B FROM ...)
```

정답: B의 값중 하나가 NULL 인 결과

해설:

NOT IN 의 경우 NULL 이 있을 때 0 ROWS 로 처리됨

■ 문제 19. 아래의 SQL결과를 출력하는 SQL을 완성하십시오. (집계 관련 문제)



- ROLLUP과 같은 구조이나 실제 전체 합이 존재하지 않는 형태
- GROUPING SET 으로 구성되거나 ROLLUP 이 부분적으로 사용
- 문제 보기 중 GROUPING SET (회원번호, (회원번호,상품), ()) 이런 형태일 경우 () 이 부분으로 전체 합이 도출됨

정답: GROUP BY 회원번호, ROLLUP (상품)

■ 문제 20. 아래의 SQL 결과를 출력하는 SQL을 완성하시오. (ROLLUP 컬럼의 순서 문제)

중간 컬럼으로만 GROUP BY 가 한번, 그 이후는 순서대로 첫번째, 세번째 컬럼이 나오는 형태

정답: ROLLUP (고객등급, 첫번째컬럼, 세번째컬럼)

■ 문제 21. 아래의 SQL에 대한 결과로 올바른 것은? (HAVING 절의 조건을 만족하지 않을 때의 문제)

```
SELECT COUNT(*)
  FROM ...
 WHERE ...
  GROUP BY ID
HAVING COUNT(*) > 3
```

정답: 1번 (만족하는 결과가 없으므로 "데이터가 출력되지 않는다.")

해설: 집계 결과중 HAVING 을 만족하지 못하므로 0 ROWS 를 반환한다.

■ 문제 22. (주문번호, 회원번호, 주문금액) 테이블의 데이터에 대한 SQL 결과가 아래와 같을 때 올바른 SQL은?

[결과]

회원번호	총주문금액

12345	500000

정답: SELECT ... FROM (SELECT ... FROM ... ORDER BY ... DESC) WHERE ROWNUM = 1

해설:

총 주문 금액이 가장 큰 1건이 나오는, ROWNUM=1 과 ORDER BY 를 어떻게 써야 하는지에 대한 문제

INLINE VIEW 에서 ORDER BY 결과가 나오고 메인쿼리에서 ROWNUM = 1 이 작동하면 위의 결과가 나옴.

■ 문제 23. 아래의 테이블 (주문번호, 회원번호, 주문금액) 에서 연봉의 순서에 따른 출력결과가 다음과 같을 때 SQL을 완성하시오.

[결과]

X	100	0
Y	200	100
X	300	200

정답: LAG(COL1, 1, 0)

해설:

LAG, LEAD (이전값, 다음값) 를 묻는 문제

맨위에 0이 있으므로 LAG를 작성할때 LAG(COL1, 1, 0) 이렇게 데이터가 없을 경우 가져오는 값을 명시해야함.

#💡알아두기

```
LAG(expr [,offset] [,default]) OVER([partition_by_clause] order_by_clause)
LEAD(expr [,offset] [,default]) OVER([partition_by_clause] order_by_clause)
```




LAG 함수 : 이전 행의 값을 리턴

LEAD 함수 : 다음 행의 값을 리턴

expr : 대상 컬럼명

offset : 값을 가져올 행의 위치 기본값은 1, 생략가능

default : 값이 없을 경우 기본값, 생략가능

partition_by_clause : 그룹 컬럼명, 생략가능

order_by_clause : 정렬 컬럼명, 필수

■ 문제 24. 다음 중 아래와 같은 테이블에서 SQL을 실행할 때 결과로 가장 적절한 것은? (노랭이 113번)

[테이블 : 고객]

고객번호(PK)	고객명
001	홍길동
002	이순신
003	강감찬
004	이상화
005	이규혁

[테이블 : 월별매출]

고객번호(PK)	고객명	매출액
202201	001	200
202201	002	300
202201	003	250
202201	004	300
202201	005	250
202202	001	150
202202	002	150
202202	004	200
202202	005	100
202203	002	100
202203	003	100
202203	004	200
202203	005	350

```

SELECT 고객번호
      , 고객명
      , 매출액
      , RANK( ) OVER(ORDER BY 매출액 DESC) AS 순위
FROM (
      SELECT A.고객번호 ,MAX(A.고객명) AS 고객명
            , SUM(B.매출액) AS 매출액
      FROM 고객 A INNER JOIN 월별매출 B
      ON (A.고객번호 = B.고객번호)
      GROUP BY A.고객번호
    )
ORDER BY 순위;

```

1)

고객번호	고객명	매출액	순위
005	이규혁	700	1
004	이상화	700	1
002	이순신	550	3
001	홍길동	350	4
003	강감찬	350	4

2)

고객번호	고객명	매출액	순위
005	이규혁	700	1
004	이상화	700	2
002	이순신	550	3
001	홍길동	350	4
003	강감찬	350	5

3)

고객번호	고객명	매출액	순위
005	이규혁	700	1
004	이상화	700	1
002	이순신	550	2
001	홍길동	350	3
003	강감찬	350	3

4)

고객번호	고객명	매출액	순위
003	강감찬	350	1
001	홍길동	350	1
002	이순신	550	2
004	이상화	700	3
005	이규혁	700	3

정답: 1번

해설:

RANK 함수는 ORDER BY 를 포함한 쿼리문에서 특정 항목에 대한 순위를 구하는 함수이며 동일한 값에는 동일한 순위를 부여함.

■ 문제 25. 아래의 SQL 결과로 올바른것을 고르시오.

[결과]

1

1

2

정답: **DENSE_RANK()** ...

해설:

정렬 순서에 따른 랭킹이 동일 등수 허용, 연속된 등수가 나오는 DENSE_RANK

■ 문제 26. 아래의 요구사항을 만족하는 SQL을 작성하시오.

...부서는 사원이 없을 경우 사원수를 0으로 표시한다.

[SQL]

```
SELECT ...  
FROM EMP A ( ) JOIN DEPT B ...
```

정답: **RIGHT OUTER**해설: **OUTER JOIN** 에서 **LEFT** / **RIGHT** 구분을 묻는 문제

■ 문제 27. 다음 중 B_User가 아래의 작업을 수행할 수 있도록 권한을 부여하는 DCL로 가장 적절한 것은? (노랭이 2과목 120번)

```
UPDATE A_User.TB_A  
SET col1 = 'AAA'  
WHERE col2 = 3;
```

- 1) GRANT SELECT, UPDATE TO B_User;
- 2) REVOKE SELECT ON A_User.TB_A FROM B_User;
- 3) DENY UPDATE ON A_User.TB_A TO B_User;

4) GRANT SELECT, UPDATE ON A_User.TB_A TO B_User;

해설:

WHERE 조건의 데이터를 찾기 위한 SELECT 권한 및 데이터 변경을 위한 UPDATE 권한이 필요.

GRANT(권한부여), REVOKE(권한회수)

DCL(Data Control Language) 데이터 제어어 에 속함.

■ 문제 28. 아래 SQL 수행 후 테이블 X에 대한 SELECT 권한을 가지고 있는 유저는?

[SQL]

```
DBA : ... GRANT U1
DBA : ... GRANT U2 WITH GRANT OPTION
U2 : ... GRANT U3 ...
DBA : ... REVOKE U2 RESTRICT
```

정답: DBA, U1, U2, U3

해설:

REVOKE 대상 USER에게 권한을 부여받은 유저가 있다면 REVOKE 가 실패하게 됨.

종류	설명
WITH GRANT OPTION	자신이 부여받은 권한에서 다른 사용자에게 권한 부여 가능, 권한을 회수할 경우 연쇄적으로 권한이 회수된다.
CASCADE	권한제거 명령어 로 부여된 권한 트리를 모두 제거한다.
RESTRICT	권한제거 명령어 로 지정한 권한을 제거 할 경우 타 사용자에게 영향이 가면 실패 하도록 처리하는 명령어

■ 문제 29. 아래의 SQL에 대해서 에러가 나지 않는것은?

[SQL]

```
CREATE TABLE T1 (
    A number default 9 primary key,
    B number not null,
    C number
);
```

1) INSERT INTO T1 VALUES (7,1)

2) ...

3) ...

4) INSERT INTO T1 (B, C) VALUES (9,1)

해설:

DEFAULT는 아무 값을 입력하지 않아도 기본 값으로 설정한 값이 자동으로 입력되는 제약조건으로 NOT NULL과 마찬가지로 많이 사용된다.

A 가 DEFAULT 9 을 가지고 있으므로 보기 4번의 쿼리는 정상적으로 작동된다.

#💡알아두기

오라클 DEFAULT 제약조건 추가
[컬럼명] [타입] DEFAULT [기본값]

📌 테이블 생성할 때 선언
CREATE TABLE T1(
 A NUMBER DEFAULT 9
);

📌 테이블 생성 후 제약조건 추가할 때
ALTER TABLE T1 MODIFY A DEFAULT 9

📌 DEFAULT 제약조건 삭제
ALTER TABLE T1 MODIFY A DEFAULT NULL

삭제 시 DEFAULT 기본값을 NULL로 작성하는 이유는 최초 테이블 삽입 시 값이 없을 때 NULL이 들어가기 때문이다.

■ 문제 30. 아래의 SQL결과로 올바른 것은?

```
INSERT ... 1
INSERT ... 2
SAVEPOINT X
UPDATE ...
DELETE ...
ROLLBACK X
INSERT ... 3

[SQL]
SELECT AVG(COL1) FROM ...
```

정답: 2

해설:

$$(1+2+3)/3 = 2$$

■ 문제 31. TABLE에 대한 설명으로 적절하지 않은 것은?

정답: PK (중복을 허용하되 NULL은 허용하지 않는다.)

해설:

PK(Primary Key / 기본 키) 는 아래와 같은 특징을 갖는다.

값이 중복되지 않는다.

값을 반드시 입력해야 한다.

테이블 데이터의 고유 인식번호 이다. (테이블당 하나의 기본키만 지정)

■ 문제 32. 아래의 SQL에 대한 결과로 올바른 것은? (WINDOWS FUNCTION 3개에 대한 문제)

[TABLE_32]

COL1	COL2	COL3
100	100	100
200	100	100
300	200	100
400	100	100
500	300	100
600	400	100
700	200	100

[SQL]

```
SELECT MAX(COL1) OVER(ORDER BY COL1 ROWS
                        BETWEEN UNBOUNDED PRECEDING AND 2 FOLLOW
ING) AS MAX,
       SUM(COL2) OVER(ORDER BY COL1 ROWS
                        BETWEEN 1PRECEDING AND CURRENT ROW) AS S
UM,
       FIRST_VALUE(COL1) OVER(ORDER BY COL1 RANGE
                               BETWEEN 200 PRECEDING AND 200 FOLLOWING)
```

```
AS FIRST
FROM TABLE_32;
```

정답:

[RESULT]

MAX	SUM	FIRST
300	100	100
400	200	100
500	300	100
600	300	200
700	400	300
700	700	400
700	600	500

해설:

- start_point와 end_point의 구성
 - start_point는 end_point와 같거나 작은 값이 들어감
 - Default값은 RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
 - UNBOUNDED PRECEDING : start_point 만 들어갈 수 있으며, 파티션의 first row
 - UNBOUNDED FOLLOWING : end_point만 들어갈 수 있으며, 파티션의 last row
 - CURRENT ROW : start, end_point 둘다 가능. 윈도우는 CUREENT ROW에서 start 하거나 end 함
 - value_expr PRECEDING

> CURRENT ROW 전의 물리적/논리적 OFFSET, 상수나 표현식(EXPRESSION)이 들어가며 양수임.

> RANGE에 사용시, DATE 컬럼에 대한 INTERVAL LITERAL일 수 있음

- value_expr FOLLOWING

> CURRENT ROW 후의 물리적/논리적 OFFSET, value_expr PRECEDING 와 성격 동일함

>> MAX : 파티션(전체 로우의 첫번째 행) 이 시작점이며 2 FOLLOWING 의 기준은 현재 로우에서 시작됨.

즉, 각 행의 뒤의 2개열까지가 끝점이 됨.

>> SUM : 시작점은 현재 로우의 1번째 전 행 부터 끝점은 현재 로우로 SUM 이 됨

>> FIRST_VALUE : 시작점은 현재 로우의 범위 기준으로 200 단위 전(500 이면 300부터) 부터 끝점은 현재 로우의 범위 기준으로 200 단위 후 (500이면 700까지) 의 범위에서 첫번째 값

■ 문제 33. 아래의 테이블과 데이터에 대한 SQL결과로 올바른 것은?

[TAB1]

COL1	COL2
------	------

1	A
---	---

1	A
---	---

1	B
---	---

1	B
---	---

```
SELECT COUNT(COL1), COUNT(COL2)
FROM (
    SELECT DISTINCT COL1, COL2
    FROM TAB1
)
```

정답: 2, 2

■ 문제 34. 아래의 SQL에 대한 설명으로 가장 적절한 것은? (NULL이 있을때 AVG와 COUNT가 어떻게 되는지 묻는 문제)

```
SELECT SUM(C2) / COUNT(*) AVG1,
       SUM(C2) / COUNT(C2) AVG2,
       AVG(C2) AVG3
```

정답: AVG2와 AVG3가 같다. (NULL이 있으면 빠지고 평균)

■ 문제 35. 아래의 테이블이 있을 때 SQL 결과로 올바른 것은? (CROSS JOIN된 이후 결과에서 조인된 테이블 NULL 사용)

[TAB1]

ID	NAME
1	SMITH
2	ALICE
3	SCOTT
4	...

[TAB2]

ID	RULE
1	S%
2	%T%

```
[SQL]
SELECT *
  FROM TAB1 A, TAB2 B
 WHERE A.NAME LIKE B.RULE;
```

정답:

ID	NAME	ID	RULE
1	SMITH	1	S%
3	SCOTT	1	S%
1	SMITH	2	%T%
3	SCOTT	2	%T%

해설:

RULE 컬럼의 내용이 들어가 LIKE 절 완성 -> CROSS JOIN후 LIKE절 체크

■ 문제 36. 아래의 데이터에 대한 SQL 결과가 다른 것은?

1. 1) SELECT COUNT(DISTINCT COL1)

2) **SELECT DISTINCT COUNT(*)**

3) SELECT * FROM ... GROUP BY ...

4) ...

■ 문제 37. 아래의 SQL결과로 올바른 것은?

```
SELECT SUBSTR('123456789', -4, 2) FROM DUAL;
```

정답: 67

해설:

SUBSTR('문자열', 시작위치, 길이)

4 (뒤에서 4번째 자리수 부터), 2자리 까지 이므로 67

(뒤에서 부터 문자열을 자를때 시작위치에 마이너스 값을 입력해줌)

뒤에서부터 문자열을 자를 때는 시작위치에 음수(-) 값을 입력
SUBSTR('문자열', 시작위치, 길이)

SUBSTR('123456789', 2): 문자열의 두 번째 위치(2)부터 문자열 끝까지 (결과:
SUBSTR('123456789', 2, 3): 문자열의 두 번째 위치(2)부터 세 자리(4)까지 (
SUBSTR('123456789', -5): 문자열의 뒤에서 두 번째 위치(5)부터 문자열 끝까지
SUBSTR('123456789', -5, 3): 문자열의 뒤에서 두 번째 위치(8)부터 문자열 끝까지

■ 문제 38. 아래의 SQL결과로 올바른 것은? (LENGTH 결과 찾는 문제)

```
SELECT LENGTH('SQL EXPERT') FROM DUAL;
```

정답: 10

■ 문제 39. 아래의 요구사항에 맞는 SQL로 적절한 것은? (요구사항에 맞춘 SQL 작성)

[요구사항]

등급은 F 또는 G 이며, 범위는 2000 ~ 50000 사이 ..

정답: AND (COL1 = F OR COL1 = G) AND ...

■ 문제 40. 아래의 데이터에 대한 SQL결과로 올바른 것은? (NULL 이 있을 때 오라클에서 정렬 되는 순서)

[TABLE40]

COL1

```
-----  
NULL  
NULL  
1  
2  
3
```

```
[SQL]  
SELECT * FROM TABLE40 ORDER BY COL1 ASC;
```

정답: 오라클 NULL은 가장 큰값으로 인식되므로 ASC로 정렬할 경우 맨 마지막에 나오게 된다.

```
COL1  
-----  
1  
2  
3  
NULL  
NULL
```

```
SELECT * FROM TABLE40 ORDER BY COL1 DESC;
```

#DESC 일 경우 아래와 같은 결과가 나옴.

```
COL1  
-----  
NULL  
NULL  
3  
2  
1
```

■ 문제 41. SQL 집합 연산자에서 교집합에 해당하는 것은?

정답: INTERSECT

■ 문제 42. 아래의 SQL에 대한 설명으로 가장 적절한 것은?

(부서별 연봉이 제일 높은 사람 찾는 내용)

■ 문제 43. 아래의 SQL구문에 대한 결과로 알맞은 것은? (MERGE INTO 구문에 대한 설명)

[TABLE_43_1]

ID COL1

1 A

2 B

[TABLE_43_2]

ID COL2

1 C

2 D

3 E

4 F

[SQL]

MERGE INTO TABLE_43_1

USING TABLE_43_2

ON ...

MATCH ... UPDATE

NOT MATCH INSERT

■ 문제 44. 아래와 같은 SQL문에 대해 삽입이 성공하는 SQL 문은? (노랭이 2과목 20번)

```
CREATE TABLE TBL (  
    ID NUMBER PRIMARY KEY,  
    AMT NUMBER NOT NULL,  
    DEGREE VARCHAR2(1)  
)
```

가. INSERT INTO TBL VALUES(1, 100)

나. INSERT INTO TBL(ID, AMT, DEGREE) VALUES(2, 200, 'AB')

다. INSERT INTO TBL(ID, DEGREE) VALUES(4, 'X')

라. INSERT INTO TBL(ID, AMT) VALUES(3, 300)

마. INSERT INTO TBL VALUES(5, 500, NULL)

1) 가, 나

2) 나, 다

3) 다, 라

4) 라, 마

해설:

가 - 삽입 컬럼을 명시하지 않았을 경우 모든 컬럼을 삽입해야 함

나 - DEGREE 컬럼의 길이는 VARCHAR2(1) 이다. 'AB'는 컬럼 길이를 초과함

다 - NOT NULL 컬럼인 AMT 컬럼을 명시하지 않았음

■ **단답형 2. 다른 테이블에서 파생된 테이블로, 물리적 데이터가 저장되는 것이 아닌 논리적으로 존재하는 것을 무엇이라 하는가?** 정답: **VIEW (뷰)**

■ **단답형 3. 아래의 SQL 결과를 작성하시오.**(COUNT(*), HAVING > 2 이며 GROUP BY 관련된 쿼리)

[TAB1]

ID	COL1
A	1
A	2
B	3
B	4
C	5
C	6
C	7
D	8

[SQL]

```
SELECT COUNT(*)
FROM TAB1
GROUP BY ID
HAVING COUNT(*) > 2;
```

정답: 3

해설:

ID, COUNT(*) 결과를 HAVING 조건절 체크한 최종결과는 '3'

```
SQL> SELECT COUNT(*) FROM TAB1 GROUP BY ID HAVING COUNT(*) > 2;
```

```
COUNT(*)  
-----  
3
```

■ **단답형 4.** 아래의 SQL결과가 나올 수 있도록 빈칸을 채우시오.

```
SELECT ...  
FROM ...  
WHERE COL ...  
OR COL ( )
```

정답: IS NULL

■ **단답형 5.** 아래의 SQL 결과에 맞는 SQL 을 작성하시오.

[결과]
ROLLUP 에 관련된 문제

정답: ROLLUP

■ **단답형 6.** NTILE(3) 의 결과를 채워넣으시오. (총 7건의 로우)

[결과]

NT	COUNT(*)
(ㄱ)	(ㄴ)
2	2
3	2

정답: ㄱ:1 / ㄴ:3