

## < < DB 쪽지시험 예상문제 > >

### [ 개념 PART ]

Q . 다음 보기를 DB에서 연산자 우선 순위를 알맞게 나열하시오. (01\_SELECT)

[보기]	연결 연산자	OR (논리연산자)	산술 연산자
	BETWEEN AND / NOT BETWEEN AND	AND (논리연산자)	NOT (논리연산자)
	IS NULL / IS NOT NULL, LIKE, IN / NOT IN		비교 연산자

1. 산술 연산자
2. 연결 연산자
3. 비교 연산자
4. IS NULL / IS NOT NULL, LIKE, IN / NOT IN
5. BETWEEN AND / NOT BETWEEN AND
6. NOT (논리연산자)
7. AND (논리연산자)
8. OR (논리연산자)

Q . 다음 연산자들을 우선 순위에 맞게 정렬하시오. (01\_SELECT)

[보기]	산술연산자	AND	비교연산자	OR
------	-------	-----	-------	----

1. 산술연산자
2. 비교연산자
3. AND
4. OR

Q . 자바와 DB에서 null의 차이를 서술하시오. (01\_SELECT)

JAVA에서 NULL : 참조하는 객체가 없다.

DB에서 NULL : 컬럼 값이 없다.

Q . IN 연산자의 정의를 쓰시오. (01\_SELECT)

비교하려는 값과 목록에 작성된 값 중 일치하는 것이 있는지를 조회하는 연산자

Q . 숫자 관련 함수를 나타내는 FLOOR와 TRUNC를 정의하고 차이점을 서술하시오.

(02\_FUNCTION)

FLOOR : 소수점 첫째 자리에서 내림 처리

TRUNC : 특정 위치 아래를 버림(절삭)

차이점: 음수일 때 결과 값이 다르다.

**Q . TO\_DATE 함수에서 연도 표기 패턴 'Y' 와 'R' 의 차이를 설명하시오. (02\_FUNCTION)**

Y는 현재 세기 기준,

R는 1 세기 기준으로 절반 이상이면 이전 세기, 절반 미만이면 현재 세기 기준으로 연도를 표기한다.

**Q . 연도에 대한 포맷 문자 'Y'와 'R'에 대하여 서술하시오. (02\_FUNCTION)**

Y는 무조건 현재 세기(21 세기)를 반영하고, R은 1 세기 기준으로 50 년 이상이면 이전 세기(1900 년대), 50 년 미만이면 현재 세기(2000 년대)를 반영한다.

**Q . 다음 집합 연산에 대해 설명하시오. (03\_GROUPBY\_HAVING)**

UNION : OR 같은 개념 (합집합) --> 중복 제거

INTERSECT : AND 같은 개념 (교집합)

UNION ALL : OR 결과 값에 AND 결과 값이 더해진 거(합집합 + 교집합) --> 중복 미제거

MINUS : 차집합 개념

**Q . 집합 연산에 대해 설명하시오. (03\_GROUPBY\_HAVING)**

UNION : OR 같은 개념 (합집합) --> 중복 제거

INTERSECT : AND 같은 개념 (교집합)

UNION ALL : OR 결과 값에 AND 결과 값이 더해진 거(합집합 + 교집합) --> 중복 미제거

MINUS : 차집합 개념

**Q . SELECT문 해석 순서를 나열하시오. (03\_GROUPBY\_HAVING)**

A : SELECT 컬럼명 AS 별칭, 계산식, 함수식

B : FROM 참조할 테이블명

C : WHERE 컬럼명 | 함수식 비교연산자 비교값

D : GROUP BY 그룹을 묶을 컬럼명

E : HAVING 그룹함수식 비교연산자 비교값

F : ORDER BY 컬럼명 | 별칭 | 컬럼순번 정렬방식 [NULLS FIRST | LAST];

B - C - D - E - A - F

**Q . SELECT 문 해석 순서를 ( ) 안에 알맞게 작성하시오. (03\_GROUPBY\_HAVING)**

(5) : SELECT 컬럼명 AS 별칭, 계산식, 함수식

(1) : FROM 참조할 테이블명

(2) : WHERE 컬럼명 | 함수식 비교연산자 비교값

(3) : GROUP BY 그룹을 묶을 컬럼명

(4) : HAVING 그룹함수식 비교연산자 비교값

(6) : ORDER BY 컬럼명 | 별칭 | 컬럼순번 정렬방식 [NULLS FIRST | LAST];

(1) - (2) - (3) - (4) - (5) - (6)

**Q . 다음 SELECT문 해석 순서를 적으세요. (03\_GROUPBY\_HAVING)**

[SELECT] 컬럼명 AS 별칭, 계산식, 함수식

[FROM] 참조할 테이블명

[WHERE] 컬럼명 | 함수식 비교연산자 비교값

[GROUP BY] 그룹을 묶을 컬럼명

[HAVING] 그룹함수식 비교연산자 비교값

[ORDER BY] 컬럼명 | 별칭 | 컬럼순번 정렬방식 [NULLS FIRST | LAST];

FROM > WHERE > GROUP BY > HAVING > SELECT > ORDER BY

**Q . 해석 순서가 어떻게 되는지 나열하시오. (03\_GROUPBY\_HAVING)**

[SELECT] 컬럼명 AS 별칭, 계산식, 함수식

[FROM] 참조할 테이블명

[WHERE] 컬럼명 | 함수식 비교연산자 비교값

[GROUP BY] 그룹을 묶을 컬럼명

[HAVING] 그룹함수식 비교연산자 비교값

[ORDER BY] 컬럼명 | 별칭 | 컬럼순번 정렬방식 [NULLS FIRST | LAST];

FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDER BY

**Q . INNER JOIN과 OUTER JOIN의 차이점을 설명하시오. (04\_JOIN)**

INNER JOIN은 연결에 사용되는 컬럼값이 NULL이면 조회 결과에 포함되지 않지만

OUTER JOIN은 두 테이블의 지정하는 컬럼값이 일치하지 않는 행도 조회 결과에 포함된다.

**Q . 내부 조인과 외부 조인의 차이점을 설명하시오. (04\_JOIN)**

내부 조인 : 연결되는 컬럼의 값이 일치하는 행들만 조인됨

외부 조인 : 두 테이블의 지정하는 컬럼 값이 일치하지 않는 행도 조인에 포함을 시킴

**Q . 내부 조인과 외부조인의 특징과 문제점을 서술하시오. (04\_JOIN)**

내부 조인 : 연결되는 컬럼의 값이 일치하는 행들만 조인됨.

외부 조인 : 두 테이블의 지정하는 컬럼값이 일치하지 않는 행도 조인에 포함을 시킴

문제점 : 연결에 사용되는 컬럼의 값이 NULL이면 조회 결과에 포함되지 않음

**Q . 내부 조인의 문제점을 서술하시오. (04\_JOIN)**

연결에 사용되는 컬럼의 값이 NULL이면 조회 결과에 포함되지 않는다.

**Q . ROLLBACK의 정의를 쓰시오. (06\_DML(INSERT,UPDATE,DELETE))**

마지막 COMMIT 시점으로 돌아감.

**Q . DELETE와 TRUNCATE의 차이점을 서술하시오. (06\_DML(INSERT,UPDATE,DELECT))**

DELETE는 ROLLBACK으로 복구되지만 TRUNCATE는 ROLLBACK으로 복구가 안 된다.

**Q . 테이블의 전체 행을 삭제하는 구문을 적으시오. (06\_DML(INSERT,UPDATE,DELECT))**

TRUNCATE

**Q . 메모리 버퍼(트랜잭션)에 임시 저장된 데이터 변경 사항을 DB에 반영하는 단어를 작성하시오. (06\_DML(INSERT,UPDATE,DELECT))**

COMMIT

**Q . 테이블에 내가 선택한 컬럼에 대한 값만 INSERT할 때 사용하는 식을 작성하시오. (06\_DML(INSERT,UPDATE,DELECT))**

INSERT INTO 테이블명(컬럼명, 컬럼명, 컬럼명,...) VALUES (데이터 1, 데이터 2, 데이터 3, ...);

**Q . UNIQUE 제약조건에 대해 설명하시오. (08\_DDL(CREATE))**

컬럼에 입력값에 대해서 중복을 제한하는 제약조건.

UNIQUE 제약조건이 설정된 컬럼에 NULL값은 중복 삽입 가능하다.

**Q . FOREIGN KEY를 테이블 레벨로 사용하고자 한다. 이 때 식을 작성하시오. (08\_DDL(CREATE))**

[CONSTRAINT 이름] FOREIGN KEY (적용할 컬럼명) REFERENCES 참조할 테이블명 [(참조할 컬럼)]  
[삭제를]

**Q . DML, DDL, TCL의 정의를 각각 서술하시오. (06 & 07 & 08 & 09)**

DML : 데이터 조작 언어(데이터 삽입, 수정, 삭제)

DDL : 데이터 정의 언어(객체 생성, 수정, 삭제)

TCL : 트랜잭션 제어 언어(COMMIT, ROLLBACK, SAVEPOINT)

DML(Data Manipulation Language) : 데이터 조작 언어로 테이블에 데이터 값을 삽입(INSERT), 수정(UPDATE) 삭제(DELECT)하는 구문

DDL(DATA DEFINITION LANGUAGE) : 데이터 정의 언어로 객체를 생성(CREATE), 수정(ALTER), 삭제(DROP)

TCL(TRANSACTION CONTROL LANGUAGE) : 트랜잭션 제어 언어로 트랜잭션 종료 후 저장(COMMIT), 트랜잭션 취소(ROLLBACK), 임시저장(SAVEPOINT)이 있다.

## [ 문제 PART ]

Q . 다음 코드의 문제점을 서술하고 올바르게 고치시오. (01\_SELECT)

```
SELECT EMP_ID, EMP_NAME, PHONE FROM EMPLOYEE WHERE PHONE = NULL;
```

문제점 : **NULL**은 = 비교연산자로 연산할 수 없다.

올바르게 고친 코드 :

```
SELECT EMP_ID, EMP_NAME, PHONE FROM EMPLOYEE WHERE PHONE IS NULL;
```

Q . 아래 코드는 원하는 조건으로 나오지 않는다. 그 원인과 해결방법은? (01\_SELECT)

-- EMPLOYEE 테이블에서 전화번호가 없는 사원의 사번, 이름, 전화번호 조회

```
SELECT EMP_ID, EMP_NAME, PHONE
```

```
FROM EMPLOYEE
```

```
WHERE PHONE = NULL;
```

원인 : **NULL**을 비교하려면 **NULL** 처리 연산자를 써야한다.

해결방법 : **WHERE PHONE IS NULL;**

Q . 아래 코드는 원하는 조건으로 나오지 않는다. 그 원인과 해결방법은? (01\_SELECT)

-- EMPLOYEE 테이블에서 DEPT\_CODE가 D9 또는 D6 이고

-- 급여가 270 만 이상인 사원의 이름, 이메일, 부서코드, 입사일, 급여를 조회

```
SELECT EMP_NAME, EMAIL, DEPT_CODE, HIRE_DATE, SALARY
```

```
FROM EMPLOYEE
```

```
WHERE DEPT_CODE = 'D9' OR DEPT_CODE = 'D6'
```

```
AND SALARY >= 2700000;
```

원인 : 연산자 우선 순위에서 **AND**가 **OR**보다 우선 순위가 높아서 **OR**연산자 보다 **AND**연산자를 먼저 실행되기 때문이다.

해결방법 : **WHERE (DEPT\_CODE = 'D9' OR DEPT\_CODE = 'D6')**

Q . EMPLOYEE 테이블에서 근무 년수가 20 년 이상인 직원의 전체 정보 조회하시오.(EXTRACT 사용) (02\_FUNCTION)

```
SELECT * FROM EMPLOYEE
```

```
WHERE EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM HIRE_DATE) >= 20;
```

Q . SELECT LENGTH('HELLO WORLD') FROM DUAL; 의 반환 값은? (02\_FUNCTION)

11

Q . 'GOOD BYE' 문자열의 길이를 반환하는 SQL문을 작성하시오. (02\_FUNCTION)

```
SELECT 'GOOD BYE', LENGTH('GOOD BYE') FROM DUAL;
```

Q . 다음 쿼리문 실행 시 오류가 발생했다. 번호를 올바르게 해석 순서로 변경하시오. (테이블은 신경쓰지 않는다.) (03\_GROUPBY\_HAVING)

1	SELECT DEPT_CODE , FLOOR(AVG(SALARY))
2	FROM EMPLOYEE
3	ORDER BY DEPT_CODE
4	GROUP BY DEPT_CODE
5	HAVING AVG(SALARY) >= 3000000
6	WHERE DEPT_CODE = 'D5'
	;

1 - 2 - 6 - 4 - 5 - 3

Q . 다음 코드를 실행한 결과, 다음과 같은 오류가 발생했다. 오류의 원인을 서술하고 코드를 수정하시오. (03\_GROUPBY\_HAVING)

-- ORA-00933: SQL 명령어가 올바르게 종료되지 않았습니다

```
SELECT SUBSTR(TERM_NO, 1, 4) 년도, ROUND(AVG(POINT), 1) "년도 별 평점"
FROM TB_GRADE
GROUP BY SUBSTR(TERM_NO, 1, 4)
WHERE STUDENT_NO = 'A112113'
ORDER BY 년도;
```

원인 : GROUP BY와 WHERE의 순서가 잘못되어서 오류가 발생했다.

올바르게 수정한 코드 :

```
SELECT SUBSTR(TERM_NO, 1, 4) 년도, ROUND(AVG(POINT), 1) "년도 별 평점"
FROM TB_GRADE
WHERE STUDENT_NO = 'A112113'
GROUP BY SUBSTR(TERM_NO, 1, 4)
ORDER BY 년도;
```

Q . EMPLOYEE 테이블에서 부서 별로 같은 직급인 사원의 급여 합계를 조회하고, 부서 코드 오름차순으로 정렬 (03\_GROUPBY\_HAVING)

부서 코드 : DEPT\_CODE,

직급 : JOB\_CODE,

급여 : SALARY

```
SELECT DEPT_CODE , JOB_CODE , SUM(SALARY)
FROM EMPLOYEE
GROUP BY DEPT_CODE , JOB_CODE
ORDER BY DEPT_CODE ;
```

Q . 아래의 코드를 실행했는데 의도한 결과와 다르게 조회되었다. 오류의 원인을 서술하시오.

(04\_JOIN)

SELECT EMP\_ID 사번, EMP\_NAME 이름, JOB\_NAME 직급명, DEPT\_TITLE 부서명, LOCAL\_NAME  
근무지역명, SALARY 급여

FROM EMPLOYEE

JOIN DEPARTMENT ON (DEPT\_ID = DEPT\_CODE)

NATURAL JOIN JOB

JOIN LOCATION ON (LOCATION\_ID = LOCAL\_CODE)

WHERE LOCAL\_NAME LIKE '%ASIA%' AND JOB\_NAME = '대리';

원인 : 자연 조인은 테이블간에 동일한 형식을 갖는 공통 컬럼이 반드시 하나만 존재해야 한다.

Q . 직급 별 가장 나이가 많은 직원을 조회하는 SQL문을 작성하시오. (05\_SUBQUERY)

SELECT MAX(FLOOR(MONTHS\_BETWEEN(SYSDATE,  
TO\_DATE(SUBSTR(EMP\_NO,1,6),'RRMMDD' ) )/12)) 나이  
FROM EMPLOYEE;

Q . 테이블에 기록된 값을 변경하려고 한다. EMPLOYEE2 테이블에서 장쯔위(EMP\_NAME)의  
JOB\_CODE와 SAL\_LEVEL을 심봉선과 같도록 바꿔라. (06\_DML(INSERT,UPDATE,DELETE))

UPDATE EMPLOYEE2

SET (JOB\_CODE, SAL\_LEVEL) = (SELECT JOB\_CODE, SAL\_LEVEL FROM EMPLOYEE2

WHERE EMP\_NAME = '심봉선')

WHERE EMP\_NAME = '장쯔위';

Q . MEMBER2 테이블의 데이터를 전체 삭제하고, 해당 테이블도 삭제하시오. (06 & 08)

DELETE FROM MEMBER2;

DROP TABLE MEMBER2;

Q . 테이블을 생성해보아라. (08\_DDL(CREATE))

테이블명 : MEMBER

컬럼명 : MEMBER\_CODE 숫자

컬럼명 : MEMBER\_NAME 한글 최대 15 글자 NULL 허용X

CREATE TABLE MEMBER(

MEMBER\_CODE NUMBER

MEMBER\_NAME VARCHAR2(45) NOT NULL

);

Q . 아래 테이블을 참고하고 SQL문 내용을 진행 후, 지워진 부서들 이름을 쓰시오. (07\_TCL)

[DEPARTMENT3 테이블]

DEPT_CODE	DEPT_TITLE	LOCATION_ID
D1	인사관리부	L1
D2	회계관리부	L1
D3	마케팅부	L1
D4	국내영업부	L1
D5	해외영업 1 부	L2
D6	해외영업 2 부	L3
D7	해외영업 3 부	L4
D8	기술지원부	L5
D9	총무부	L1

[SQL문]

**DELETE FROM** DEPARTMENT3

**WHERE** DEPT\_ID = 'D3';

**SAVEPOINT** SP1; -- 마케팅부(D3) 삭제 시점에 SAVEPOINT 지정

**DELETE FROM** DEPARTMENT3

**WHERE** DEPT\_TITLE = '해외영업 3 부';

**COMMIT;**

**DELETE FROM** DEPARTMENT3

**WHERE** DEPT\_TITLE = '해외영업 1 부';

**SAVEPOINT** SP2;

**DELETE FROM** DEPARTMENT3

**WHERE** DEPT\_ID = 'D4';

**ROLLBACK;**

마케팅부, 해외영업 3 부



Q . 테이블 생성시 MEMBER\_ID에 최대로 들어갈 수 있는 한글 글자 수는? (08\_DDL(CREATE))

```
CREATE TABLE MEMBER2(  
MEMBER_ID VARCHAR2(30),  
MEMBER_SSN NUMBER  
);
```

10 글자

Q . 주어진 테이블명과 컬럼명으로 테이블을 생성하시오. (08\_DDL(CREATE))

테이블명: KH\_CLASS

컬럼명: CLASS\_CODE 숫자 기본키

컬럼명:CLASS\_NAME 한글 최대 4 글자 널허용X

```
CREATE TABLE KH_CLASS(  
CLASS_CODE NUMBER PRIMARY KEY,  
CLASS_NAME VARCHAR2(12) NOT NULL  
);
```