# **SQL BASIC**

구간	문제 번호	주제	난이도
1~10	001~010	SQL 명령어 분류 및 기본	★ 초급
11~20	011~020	테이블 생성 및 제약조건	★ ★ 초~중급
21~30	021~030	관계 설정 및 트랜잭션	★ ★ 중급
31~40	031~040	SELECT 조건 및 NULL 처리	★ ★ ☆ 중~고급
41~50	041~050	고급 함수 및 날짜/CASE	★ ★ ☆ 고급
51~64	051~064	실전 SELECT + GROUP BY/HAVING + JOIN	★ ★ ★ ☆ 실무형

# ☑ 11~20번: 테이블 생성, 제약조건, 데이터 조작

- 테이블 명명 규칙, DEFAULT, INDEX, PK/FK
- INSERT/DELETE/ALTER 관련 실습 문제

#### [문제 011]

다음 중 물리적 테이블 명으로 가장 적절한 것은?

- ① EMP\_10
- ② 100-EMP
- 3 EMP-100
- 4 100\_EMP

#### 정답: ①

### 🐣 쉬운 해설:

①번은 "숫자로 시작하지 않고, 특수문자도 없지!" SQL에서 테이블 이름은 문자로 시작하고, 밑줄(\_)은 허용돼!

### **!!** 전문 해설:

SQL에서 테이블명은 문자로 시작해야 하며, 숫자나 특수문자(- 등)로 시작하거나 포함되면 오류가 발생할 수 있습니다. 밑줄(\_)은 허용됩니다.

보기 번호	설명	적절성
1	문자로 시작, 밑줄 사용 → 적절	<u>~</u>
2	숫자로 시작, 하이픈 포함 → 부적절	×
3	하이픈 포함 → 부적절	×
4	숫자로 시작 → 부적절	×

테이블명 = 문자로 시작 + 밑줄은 OK + 특수문자 X

#### 필요 암기카드:

• 👂 카드 83: 테이블 명명 규칙 = 문자 시작, 특수문자 🔀, 밑줄 OK

#### [문제 012]

아래와 같은 테이블 구조를 정의하려고 한다. 이때 아직 부서가 정의되지 않은 사원은 기본부서(코드 : '0000')로 배치하고, 입사일자(JOIN DATE) 기준으로 많은 조회가 발생하므로 입사일자에 Index를 생성하려고 한다.

다음 중 올바른 SQL 문장을 2개 고르시오

#### **EMP**

### EMP\_NO: VARCHAR2(10) NOT NULL

EMP NM. VARCHAR2(30) NOT NULL

DEPT\_CODE: VARCHAR2(4) NOT NULL

JOIN DATE: DATE NOT NULL REGIST DATE: DATE NULL

```
① CREATE TABLE EMP
(EMP_NO VARCHAR2(10) PRIMARY KEY,
EMP NM VARCHAR2(30) NOT NULL,
DEPT CODE VARCHAR2(4) DEFAULT '0000' NOT NULL,
JOIN DATE DATE NOT NULL,
REGIST DATE DATE NULL);
CREATE INDEX IDX EMP 01 ON EMP (JOIN DATE);
2 CREATE TABLE EMP
(EMP NO VARCHAR2(10) PRIMARY KEY,
EMP_NM VARCHAR2(30) NOT NULL,
DEPT CODE VARCHAR2(4) DEFAULT '0000',
JOIN DATE DATE NOT NULL,
REGIST DATE DATE );
CREATE INDEX IDX_EMP_01 ON EMP (JOIN_DATE);
3 CREATE TABLE EMP
(EMP_NO VARCHAR2(10) NOT NULL,
EMP_NM VARCHAR2(30) NOT NULL,
DEPT_CODE VARCHAR2(4) DEFAULT '0000' NOT NULL,
JOIN_DATE DATE NOT NULL,
REGIST_DATE DATE):
ALTER TABLE EMP ADD CONSTRAINT EMP_PK PRIMARY KEY
(EMP_NO):
CREATE INDEX IDX_EMP_01 ON EMP (JOIN_DATE):
4 CREATE TABLE EMP
(EMP_NO VARCHAR2(10) NOT NULL PRIMARY KEY,
EMP_NM VARCHAR2(30) NOT NULL,
DEPT_CODE VARCHAR2(4) DEFAULT '0000' NOT NULL,
JOIN_DATE DATE NOT NULL,
REGIST DATE DATE NULL );
ALTER TABLE EMP ADD CONSTRAINT EMP_PK PRIMARY KEY
(EMP_NO):
CREATE INDEX IDX_EMP_01 ON EMP (JOIN_DATE):
```

#### 정답: ①, ③

#### 🧸 쉬운 해설:

①,③번은 "기본값도 잘 넣고, 인덱스도 만들고, PK도 잘 지정했어!" 문법도 정확하고, 요구사항도 충족해!

#### 🌉 전문 해설:

입사일자에 인덱스를 생성하고, DEPT\_CODE에 기본값을 지정하며, EMP\_NO에 대해 PK를 설정해야 합니다.
①과 ③은 이 조건을 모두 만족합니다.

보기 번호	설명	적절성
1)	DEFAULT, NOT NULL, 인덱스 설정 → ☑	<u> </u>
2	NOT NULL 누락, DEFAULT 누락	×
3	PK 별도 설정, 인덱스 생성 → ☑	
4	PK 중복 설정 → 문법 오류	×

기본값 + 인덱스 + PK = 정확한 테이블 정의

#### 필요 암기카드:

- 参 카드 5: CREATE TABLE + CONSTRAINT
- 🔹 카드 84: DEFAULT = 기본값 설정
- 🔹 카드 85: CREATE INDEX = 성능 향상

#### [문제 013]

다음 중 아래와 같은 문장으로 '학생' 테이블을 생성한 후, 유효한 튜플(Tuple)들을 삽입하였다. SQL 1, SQL 2 문장의 실행 결과로 가장 적절한 것은?

생성) create table 학생 (학번 char (8) primary key, 장학금 integer):

SQL1: select count(\*) from 학생 SQL2: select count(학번) from 학생

- ® SQL1, SQL2 문장의 실행 결과는 다를 수 있으며, 그 이유는 장학금 속성(Attribute)에 널(Null)값이 존재할 수 있기 때문이다.
- ② SQL1, SQL2 문장의 실행 결과는 항상 다르다.
- ③ SQL1, SQL2 문장의 실행 결과는 항상 같다.
- ® SQL1, SQL2 문장의 실행 결과는 다를 수 있으며, 그 이유는 학번 속성(Attribute)에 널(Null)값이 존재할 수 있기 때문이다.

#### 정답: ③

#### 🐣 쉬운 해설:

③번은 "학번은 PK니까 NULL이 없어!" 그래서 COUNT(\*)와 COUNT(학번)은 항상 같아!

### 뷐 전문 해설:

COUNT(\*)는 전체 행 수를 세고,

COUNT(칼럼)는 NULL이 아닌 값만 셉니다. 하지만 학번은 PK이므로 NULL이 될 수 없기 때문에 두 결과는 항상 같습니다.

#### 보기 설명:

보기 번호	설명	적절성
1)	장학금은 상관없음	×
2	항상 다르다 → 틀림	×
3	PK는 NULL 불가 → 항상 같음	<u> </u>
(4)	학번에 NULL 가능 → 틀림	×

### 🧠 기억법:

COUNT(\*) vs COUNT(PK) → 항상 같음 (PK는 NULL 💢)

#### 필요 암기카드:

- 🔹 카드 86: COUNT(\*) vs COUNT(컬럼) 차이
- 🔹 카드 61: PRIMARY KEY = 고유 이름표, NULL 🗶

#### [문제 014]

다음 중 외래키에 대한 설명으로 가장 부적절한 것을 2개 고르시오.

- ① 테이블 생성시 설정할 수 있다.
- ② 외래키 값은 널 값을 가질 수 없다.
- ③ 한 테이블에 하나만 존재해야 한다.
- ④ 외래키 값은 참조 무결성 제약을 받을 수 있다.

#### 정답: ②, ③

### 🐣 쉬운 해설:

②,③번은 "외래키는 NULL 안 되고, 하나만 있어야 해!"라고 말하는데, 사실 NULL도 가능하고 여러 개도 만들 수 있어!

### 💄 전문 해설:

외래키는 NULL 값을 가질 수 있으며, 한 테이블에 여러 개의 외래키를 설정할 수 있습니다. 참조 무결성 제약도 적용됩니다.

보기 번호	설명	적절성
1)	테이블 생성 시 설정 가능	<u> </u>
2	외래키는 NULL 가능 → 틀림	×
3	외래키는 여러 개 가능 → 틀림	×
4	참조 무결성 적용 가능 → 맞음	

FK = NULL 가능 + 여러 개 가능

#### 필요 암기카드:

- 🔹 카드 62: FOREIGN KEY = 연결 다리
- 🦠 카드 87: FK 특징 = NULL 가능, 다중 설정 가능

#### [문제 015]

다음 중 데이터베이스 테이블의 제약조건(Constraint)에 대한 설명으로 가장 부적절한 것은?

- ① CHECK 제약조건은 특정 컬럼의 무결성을 유지하기 위해 설정한다.
- ② PRIMARY KEY는 반드시 테이블당 하나만 정의할 수 있다.
- ③ UNIQUE로 지정된 모든 컬럼은 NULL 값을 가질 수 없다.
- ④ FOREIGN KEY는 다른 테이블의 PK를 참조하여 관계를 정의한다.

#### 정답: ③

#### 🧸 쉬운 해설:

③번은 "UNIQUE는 중복도 안 되고 NULL도 안 돼!"라고 말하는데, 사실 NULL은 허용돼! 중복만 안 되는 거야!

#### 🍱 전문 해설:

UNIQUE 제약조건은 중복을 허용하지 않지만, NULL은 하나 이상 허용됩니다.

PK는 NULL을 허용하지 않으며, 테이블당 하나만 설정 가능합니다.

보기 번호	설명	적절성
1)	CHECK 제약조건 설명 정확	
2	PK는 테이블당 하나만 가능 → 맞음	

보기 번호	설명	적절성
3	UNIQUE는 NULL 허용 → 틀림	×
4	FK는 다른 테이블의 PK 참조 → 맞음	

UNIQUE = 중복 X, NULL ✓ PK = 중복 X, NULL X

### 필요 암기카드:

• 🔹 카드 61: PRIMARY KEY = 고유 이름표

• ▼ 카드 63: UNIQUE = 중복 X, NULL 가능

• 🔹 카드 64: NOT NULL = 빈칸 금지

#### [문제 016]

EMP 테이블에서 COMM 칼럼을 삭제하고자 할 때, 아래 SQL 문장의 빈칸에 들어갈 내용은?

# EMP 테이블

MGR	ENAME	SAL	СОММ
7566	FORD	3000	
7566	SCOTT	3000	
7698	JAMES	950	
7698	ALLEN	1600	300
7698	WARD	1250	500
7698	TURNER	1500	0
7698	MARTIN	1250	1400
7782	MILLER	1300	
7839	ADAMS	1100	
7839	BLAKE	2850	
7839	JONES	2975	
7839	CLARK	2450	

MGR	ENAME	SAL	СОММ
7839	SMITH	800	

[ ① ] TABLE EMP

[ ② ] COMM

정답: ALTER, DROP COLUMN

### 🐣 쉬운 해설:

"ALTER로 테이블을 바꾸고, DROP COLUMN으로 칼럼을 지워!" 이게 칼럼 삭제의 기본 공식이야!

### **軕** 전문 해설:

칼럼을 삭제할 때는 ALTER TABLE 명령어와 함께 DROP COLUMN을 사용해야 합니다.

### 🧠 기억법:

칼럼 삭제 = ALTER TABLE + DROP COLUMN

### 필요 암기카드:

- 🔹 카드 5: ALTER TABLE = 구조 변경
- 🔹 카드 88: DROP COLUMN = 칼럼 제거

## [문제 017]

아래 7개의 SQL 문장이 성공적으로 수행되었다고 할 때, A, B, C 세 개의 SQL 문장을 차례대로 실행하면 A와 C의 SELECT 문장 수행결과는 각각 무엇인가?

```
CREATE TABLE 부서(부서번호 CHAR(10), 부서명 CHAR(10), PRIMARY KEY(부서번호));

CREATE TABLE 직원(직원번호 CHAR(10), 소속부서 CHAR(10), PRIMARY KEY(직원번호), FOREIGN KEY(소속부서) REFERENCES 부서(부서번호) ON DELETE CASCADE);

INSERT INTO 부서 VALUES('10', '영업과');
INSERT INTO 부서 VALUES('20', '기획과');

INSERT INTO 직원 VALUES('1000', '10');
INSERT INTO 직원 VALUES('2000', '20');
INSERT INTO 직원 VALUES('3000', '20');
COMMIT:
```

```
A. SELECT COUNT(직원번호) FROM 직원
B. DELETE FROM 부서 WHERE 부서번호 = '20':
C. SELECT COUNT(직원번호) FROM 직원
COMMIT;
```

- ① 3, null
- 2 3, 1
- 3 3, 2
- 4 3, 3

정답: ② (A: 3, C: 1)

#### 🐣 쉬운 해설:

②번은 "부서 20을 지우면 그 부서 직원도 같이 지워져!" ON DELETE CASCADE가 작동한 결과야!

#### 🌗 전문 해설:

부서번호 20 삭제  $\rightarrow$  해당 부서를 참조하는 직원도 삭제됨 직원 수: 삭제 전 3명  $\rightarrow$  삭제 후 1명

보기 번호	A 결과	C 결과	적절성
1)	3	NULL → 결과 없음	×
2	3	1 → 정확	<u> </u>
3	3	2 → 틀림	×
(4)	3	3 → 삭제 안 됨	×

ON DELETE CASCADE = 부모 삭제 → 자식도 삭제

#### 필요 암기카드:

- 🔹 카드 62: FOREIGN KEY = 연결 다리
- 🔊 카드 82: ON DELETE 옵션 = CASCADE / SET NULL

[문제 018]

STADIUM 테이블의 이름을 STADIUM\_JSC로 변경하는 SQL을 작성하시오. (ANSI 표준 기준)

정답: RENAME STADIUM TO STADIUM JSC

### 🐣 쉬운 해설:

"RENAME으로 테이블 이름을 바꿔!" TO 다음에 새 이름을 쓰면 돼!

#### 🍱 전문 해설:

ANSI SQL 표준에서는 테이블 이름 변경 시 RENAME 명령어를 사용합니다.

#### 🧠 기억법:

테이블 이름 변경 = RENAME A TO B

#### 필요 암기카드:

- 🔹 카드 89: RENAME = 테이블 이름 변경
- 🔹 카드 5: ALTER TABLE = 구조 변경

#### [문제 019]

표준 SQL(SQL:1999)에서 테이블 생성시 참조관계를 정의하기 위해 외래키(Foreign Key)를 선언한다. 관계형 데이터베이스에서 Child Table의 FK 데이터 생성시 Parent Table에 PK가 없는 경우, Child Table 데이터 입력을 허용하지 않는 참조동작(Referential Action)인 것은?

- ① CASCADE
- ② RESTRICT
- 3 AUTOMATIC
- **4** DEPENDENT

#### 정답: ④

#### 🐣 쉬운 해설:

④번은 "부모 없으면 자식도 못 생겨!" DEPENDENT는 부모 없으면 입력 자체가 안 돼!

#### ■ 전문 해설:

DEPENDENT는 부모 테이블에 참조 대상이 없으면 자식 테이블에 데이터를 입력할 수 없도록 제한합니다.

※ DEPENDENT는 부모가 없으면 자식도 못 생기는 원칙

예)

```
--- 부모 테이블: 고객

CREATE TABLE 고객 (
고객ID INT PRIMARY KEY,
고객명 VARCHAR(100)
);

-- 자식 테이블: 주문

CREATE TABLE 주문 (
주문ID INT PRIMARY KEY,
고객ID INT,
주문일자 DATE,
FOREIGN KEY (고객ID) REFERENCES 고객(고객ID)
-- DEPENDENT 참조동작을 가정: 부모에 고객ID가 없으면 주문 입력 불가
);
```

#### 보기 설명:

보기 번호	설명	적절성
1)	CASCADE → 삭제 시 자식도 삭제	×
2	RESTRICT → 삭제 제한	×
3	AUTOMATIC → 표준 아님	×
4	DEPENDENT $\rightarrow$ 참조 없으면 입력 불가	

### 🧠 기억법:

DEPENDENT = 부모 없으면 자식 생성 

★

#### 필요 암기카드:

- 🔹 카드 82: ON DELETE 옵션 = CASCADE / SET NULL / RESTRICT
- 🛊 카드 90: DEPENDENT = 참조 없으면 입력 불가

```
[문제 020]
아래와 같은 SQL문에 대해 삽입이 성공하는 SQL문은?
```

```
CREATE TABLE TBL

(
ID NUMBER PRIMARY KEY,

AMT NUMBER NOT NULL,

DEGREE VARCHAR2(1)
)
```

```
    INSERT INTO TBL VALUES(1, 100)
    INSERT INTO TBL(ID, AMT, DEGREE) VALUES(2, 200, 'AB')
    INSERT INTO TBL(ID, DEGREE) VALUES(4, 'X')
    INSERT INTO TBL(ID, AMT) VALUES(3, 300)
    INSERT INTO TBL VALUES(5, 500, NULL)
```

- 1) 1, 2
- (2) 2. 3
- 3 3, 4
- 4 4, 5

### 정답: ④

### 🐣 쉬운 해설:

④번은 "필수값만 넣었고, NULL도 안 빠졌어!" ID와 AMT만 있으면 삽입 성공이야!

### 💵 전문 해설:

AMT는 NOT NULL이므로 반드시 입력해야 하며,

DEGREE는 NULL 가능

- ①번은 DEGREE 누락  $\rightarrow$  오류
- ②번은 DEGREE 길이 초과 → 오류
- ③번은 AMT 누락 → 오류
- ⑤번은 DEGREE NULL → 가능

보기 번호	보기 번호 설명	
1)	DEGREE 누락 → 오류	×
2	DEGREE 길이 초과 → 오류	×
3	AMT 누락 → 오류	×

보기 번호	설명	적절성
4	필수값만 입력 → 성공	
(5)	NULL 허용 $\rightarrow$ 성공 가능	

NOT NULL  $\rightarrow$  반드시 입력 VARCHAR2(1)  $\rightarrow$  한 글자만 허용

# 필요 암기카드:

• 🏂 카드 64: NOT NULL = 빈칸 금지

• 🔹 카드 91: INSERT 조건 = 필수값 확인