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

✅ 1~10번: SQL 명령어 분류 및 기본 개념

- DDL, DML, DCL, TCL 구분
- 기본 테이블 생성 및 제약조건
- 초보자 필수 개념

```
[문제 001]
다음 중 데이터 제어어(DCL)에 해당하는 명령어는?

① INSERT
② RENAME
③ COMMIT
④ REVOKE
```

정답: ④

🧸 쉬운 해설:
④번은 “접근 권한을 회수하는 명령어야!”
REVOKE는 권한을 제어하는 DCL 명령어야!

📚 전문 해설:
DCL(Data Control Language)은 데이터에 대한 접근 권한을 제어하는 명령어 집합입니다.
REVOKE는 사용자에게 부여된 권한을 회수할 때 사용되며, DCL에 속합니다.

보기 설명:

보기 번호	설명	적절성
①	INSERT → DML	❌
②	RENAME → DDL	❌
③	COMMIT → TCL	❌
④	REVOKE → DCL	✅

🧠 **기억법:**
DCL = GRANT / REVOKE → 권한 부여와 회수

필요 암기카드:

- 🃏 카드 1: SQL 명령어 분류 = DDL, DML, DCL, TCL
- 🃏 카드 2: DCL = 권한 제어 (GRANT, REVOKE)

```
[문제 002]
다음 중 아래 내용의 범주에 해당하는 SQL 명령어로 옳지 않은 것은?
① CREATE
② GRANT
③ ALTER
④ DROP
```

> 테이블의 구조를 생성, 변경, 삭제하는 등 데이터 구조를 정의하는데 사용되는 명령어이다.

정답: ②

🧸 **쉬운 해설:**
②번은 “GRANT는 구조가 아니라 권한을 주는 거야!”
그래서 DDL이 아니라 DCL이야!

📚 **전문 해설:**
CREATE, ALTER, DROP은 데이터 구조를 정의하거나 변경하는 DDL 명령어입니다.
GRANT는 권한을 부여하는 DCL 명령어로, 구조 정의와는 관련이 없습니다.

보기 설명:

보기 번호	설명	적절성
①	CREATE → DDL	✅
②	GRANT → DCL	❌

보기 번호	설명	적절성
③	ALTER → DDL	✅
④	DROP → DDL	✅

🧠 **기억법:**
DDL = CREATE / ALTER / DROP / RENAME
DCL = GRANT / REVOKE

필요 암기카드:

- 🃏 카드 1: SQL 명령어 분류 = DDL, DML, DCL, TCL
- 🃏 카드 2: DDL = 구조 정의 / DCL = 권한 제어

[문제 003]
아래 내용에 해당하는 SQL 명령어의 종류를 작성하시오.

> 논리적인 작업의 단위를 묶어 DML에 의해 조작된 결과를 작업 단위(Transaction)별로 제어하는 명령어인 Commit, Rollback, Savepoint 등이 여기에 해당하며, 일부에서는 DCL(Data Control Language)로 분류하기도 한다.

정답: TCL

🧸 **쉬운 해설:**
“작업 단위를 묶어서 처리하는 명령어야!”
Commit, Rollback은 트랜잭션 제어니까 TCL이지!

📚 **전문 해설:**
TCL(Transaction Control Language)은 트랜잭션을 제어하는 명령어 집합입니다.
Commit, Rollback, Savepoint 등이 포함되며, DML 작업의 결과를 확정하거나 취소할 때 사용됩니다.

🧠 **기억법:**
TCL = COMMIT / ROLLBACK / SAVEPOINT

필요 암기카드:

- 🃏 카드 3: TCL = 트랜잭션 제어 명령어
- 🃏 카드 1: SQL 명령어 분류 = DDL, DML, DCL, TCL

[문제 004]
데이터 언어와 SQL 명령어에 대한 설명으로 가장 부적절한 것은?

① 비절차적 데이터 조작어(DML)는 사용자가 무슨 데이터를 원하며, 어떻게 그것을 접근해야 되는지를 명세하는 언어이다.
② DML은 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터베이스를 실질적으로 접근하는데 사용되며 SELECT, INSERT, DELETE, UPDATE 등이 있다.
③ DDL은 스키마, 도메인, 테이블, 뷰, 인덱스를 정의하거나 변경 또는 제거할 때 사용되며 CREATE, ALTER, DROP, RENAME 등이 있다.
④ 호스트 프로그램 속에 삽입되어 사용되는 DML 명령어들을 데이터 부속어(Data Sub Language)라고 한다.

정답: ①

🧸 쉬운 해설:
①번은 “DML은 어떻게 접근하는지도 알려줘!”라고 말하는데,
DML은 ‘무엇을’만 말하고 ‘어떻게’는 말하지 않아!

📚 전문 해설:
DML은 비절차적 언어로, 사용자가 원하는 데이터를 명시하지만
접근 방식은 기술하지 않습니다.
‘어떻게’는 절차적 언어에서 다루는 부분입니다.

보기 설명:

보기 번호	설명	적절성
①	DML이 ‘어떻게’까지 명세한다 → 과도한 설명	❌
②	DML 명령어 설명 정확	✅
③	DDL 명령어 설명 정확	✅
④	DML이 호스트 프로그램에 삽입 → 맞음	✅

🧠 기억법:
DML = SELECT, INSERT, UPDATE, DELETE
❌ 접근 방식은 명세하지 않음

필요 암기카드:

- 🃏 카드 1: SQL 명령어 분류 = DDL, DML, DCL, TCL
- 🃏 카드 4: DML = 데이터 조작 명령어

```
[문제 005]
다음 중 데이터베이스 시스템 언어의 종류와 해당되는 명령어를 바르게 연결한 것을 2개 고르시오.

① DML - SELECT
② TCL - COMMIT
③ DCL - DROP
④ DML - ALTER
```

정답: ①, ②

🧸 **쉬운 해설:**
①,②번은 “SELECT는 데이터를 꺼내고, COMMIT은 저장하는 거야!”
둘 다 정확한 연결이야!

📚 **전문 해설:**
SELECT는 DML, COMMIT은 TCL에 속합니다.
DROP은 DDL이고, ALTER도 DDL이므로 ③,④는 틀렸습니다.

※ 시스템언어
시스템 언어는 데이터베이스에서 데이터를 정의하고, 조작하고, 제어하고, 트랜잭션을 관리하기 위해 사용하는 SQL의 하위 언어들

보기 설명:

보기 번호	설명	적절성
①	DML - SELECT → 정확	✅
②	TCL - COMMIT → 정확	✅
③	DCL - DROP → DROP은 DDL	❌
④	DML - ALTER → ALTER는 DDL	❌

🧠 **기억법:**
DML = SELECT / TCL = COMMIT
DROP, ALTER = DDL

필요 암기카드:

- 🎴 카드 1: SQL 명령어 분류 = DDL, DML, DCL, TCL
- 🎴 카드 3: TCL = 트랜잭션 제어 명령어

[문제 006]
다음 중 아래의 데이터 모델과 같은 테이블 및 PK 제약조건을 생성하는 DDL 문장으로 올바른 것은?
(Oracle 기준)

PRODUCT
PROD_ID: VARCHAR2(10) NOT NULL
PROD_NM VARCHAR2(100) NOT NULL REG DT: DATE NOT NULL REGR NO NUMBER(10) NULL

[IE 표기법]

PRODUCT
PRODUCT() # PROD_ID VARCHAR2(10) +PROD_NM VARCHAR2(100) + REG_DT DATE o REGR NO NUMBER(10)

[Barker 표기법]

①

```
CREATE TABLE PRODUCT
(PROD_ID VARCHAR2(10) NOT NULL,
 PROD_NM VARCHAR2(100) NOT NULL,
 REG DT DATE NOT NULL,
 REGR NO NUMBER(10) NULL);
ALTER TABLE PRODUCT ADD PRIMARY KEY PRODUCT_PK ON (PROD_ID);
```

②

```
CREATE TABLE PRODUCT
(PROD_ID VARCHAR2(10),
 PROD_NM VARCHAR2(100),
 REG_DT DATE,
 REGR_NO NUMBER(10));
ALTER TABLE PRODUCT ADD CONSTRAINT PRODUCT_PK
PRIMARY KEY (PROD_ID);
```

③

```
CREATE TABLE PRODUCT
(PROD_ID VARCHAR2(10) NOT NULL,
 PROD_NM VARCHAR2(100) NOT NULL,
 REG_DT DATE NOT NULL,
 REGR NO NUMBER(10) NULL,
 ADD CONSTRAINT PRIMARY KEY (PROD_ID));
```

④

```
CREATE TABLE PRODUCT
(PROD_ID VARCHAR2(10) NOT NULL,
 PROD_NM VARCHAR2(100) NOT NULL,
 REG_DT DATE NOT NULL,
 REGR_NO NUMBER(10),
 CONSTRAINT PRODUCT_PK PRIMARY KEY (PROD_ID));
```

정답: ④

🧸 **쉬운 해설:**
④번은 “테이블 만들면서 바로 PK도 지정해!”
가장 깔끔하고 정확한 방식이야!

📚 **전문 해설:**
Oracle에서는 CREATE TABLE 문 안에 CONSTRAINT를 포함하여
PK를 지정하는 것이 가장 일반적이고 오류 없이 처리됩니다.

보기 설명:

보기 번호	설명	적절성
①	ALTER 구문 오류 포함	❌
②	ALTER 방식은 가능하지만 덜 직관적	✅
③	CREATE 내부 구문 오류	❌
④	CREATE 내부에 CONSTRAINT 포함 → 정확	✅

🧠 **기억법:**
PK 지정은 → CREATE 내부에서 CONSTRAINT로 처리

필요 암기카드:

- 🃏 카드 5: CREATE TABLE + CONSTRAINT
- 🃏 카드 61: PRIMARY KEY = 고유 이름표

```
[문제 007]
아래와 같이 데이터가 들어있지 않은 왼쪽의 기관분류 테이블 (가)를 오른쪽 기관분류 테이블 (나)처럼
변경하고자 할 때 다음 중 올바른 SQL 문장은? (단, DBMS는 SQLServer 기준)

(가)
[  기관분류  ]
분류 ID:VARCHAR(10) NOT NULL
-------------------------------
분류명 : VARCHAR(10) NOT NULL
등록일자: VARCHAR(10) NULL

(나)
[  기관분류  ]
분류 ID:VARCHAR(10) NOT NULL
-------------------------------
분류명 : VARCHAR(30) NOT NULL
등록일자: DATE NOT NULL
```

```
① ALTER TABLE 기관분류 ALTER COLUMN (분류명 VARCHAR(30), 등록일자 DATE NOT NULL);
② ALTER TABLE 기관분류 ALTER COLUMN (분류명 VARCHAR(30) NOT NULL, 등록일자 DATE NOT NULL);
③ ALTER TABLE 기관분류 ALTER COLUMN 분류명 VARCHAR(30);
  ALTER TABLE 기관분류 ALTER COLUMN 등록일자 DATE NOT NULL;
④ ALTER TABLE 기관분류 ALTER COLUMN 분류명 VARCHAR(30) NOT NULL;
  ALTER TABLE 기관분류 ALTER COLUMN 등록일자 DATE NOT NULL;
```

정답: ④

🧸 **쉬운 해설:**

④번은 “칼럼 하나씩 정확하게 바꿔줘!”

SQLServer에서는 칼럼별로 따로 ALTER 해야 해!

📚 **전문 해설:**

SQLServer에서는 ALTER COLUMN을 사용할 때
칼럼을 하나씩 명시적으로 변경해야 하며,
데이터 타입과 제약조건을 함께 지정해야 합니다.

보기 설명:

보기 번호	설명	적절성
①	괄호 안에 여러 칼럼 → SQLServer 문법 오류	❌
②	괄호 안에 복합 변경 → 문법 오류	❌
③	NOT NULL 누락	❌
④	칼럼별로 정확하게 변경 → ✅	✅

🧠 **기억법:**
SQLServer ALTER COLUMN = **칼럼별로 따로 변경**

필요 암기카드:

- 🃏 카드 5: ALTER TABLE = 구조 변경
- 🃏 카드 64: NOT NULL = 빈칸 금지

```
[문제 008]
다음 중 NULL의 설명으로 가장 부적절한 것은?

① 모르는 값을 의미한다.
② 값의 부재를 의미한다.
③ 공백문자(Empty String) 혹은 숫자 0을 의미한다.
④ NULL과의 모든 비교(IS NULL 제외)는 알 수 없음(Unknown)을 반환한다.
```

정답: ③

🧸 **쉬운 해설:**
③번은 “NULL은 그냥 빈칸이나 0이야!”라고 말하는데,
그건 완전 다른 개념이야!

📚 **전문 해설:**
NULL은 값이 존재하지 않음을 의미하며,
공백 문자나 숫자 0은 실제 값이 존재하는 상태입니다.
NULL은 비교 연산 시 항상 Unknown을 반환합니다.

보기 설명:

보기 번호	설명	적절성
①	모르는 값 → 맞음	✅
②	값의 부재 → 맞음	✅
③	공백/0과 동일 → 틀림	❌
④	비교 결과 Unknown → 맞음	✅

🧠 **기억법:**
NULL ≠ 공백 ≠ 0
NULL은 **존재하지 않는 값**

필요 암기카드:

- 카드 64: NOT NULL = 빈칸 금지
- 카드 81: NULL 특징 = 비교 불가, 값 없음

[문제 009]
아래 테이블 T, S, R이 각각 다음과 같이 선언되었다. 다음 중 DELETE FROM T; 를 수행한 후에 테이블 R에 남아있는 데이터로 가장 적절한 것은?

```
CREATE TABLE T
(C INTEGER PRIMARY KEY,
D INTEGER):


CREATE TABLE S
(B INTEGER PRIMARY KEY,
 C INTEGER REFERENCES T(C) ON DELETE CASCADE);


CREATE TABLE R
(A INTEGER PRIMARY KEY,
 B INTEGER REFERENCES S(B) ON DELETE SET NULL);
```

T 테이블

C	D
1	1
2	1

S 테이블

B	C
1	1
2	1

R 테이블

A	B
1	1
2	2

① (1, NULL)과 (2, 2)
② (1, NULL)과 (2, NULL)
③ (2, 2)
④ (1, 1)

정답: ②

🧸 **쉬운 해설:**
②번은 “부모가 지워지면 자식도 지워지고, 손자도 NULL로 바뀌어!”
CASCADE와 SET NULL이 같이 작동한 결과야!

📚 **전문 해설:**
T 삭제 → S에서 C 참조 CASCADE → S 삭제됨
S 삭제 → R에서 B 참조 SET NULL → R의 B가 NULL로 변경됨

보기 설명:

보기 번호	설명	적절성
①	일부만 NULL → 틀림	❌
②	둘 다 NULL → 정확	✅
③	삭제되지 않음 → 틀림	❌
④	원래 값 유지 → 틀림	❌

🧠 **기억법:**
CASCADE → 자식 삭제
SET NULL → 손자 NULL 처리

필요 암기카드:

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

[문제 010]
다음 중 테이블 생성 시 칼럼별 생성할 수 있는 제약조건(Constraints)에 대한 설명으로 가장 부적절한 것은?

```
① UNIQUE: 테이블 내에서 중복되는 값이 없으며 NULL 입력이 불가능하다.
② PK: 주키로 테이블당 1개만 생성이 가능하다.
③ FK: 외래키로 테이블당 여러 개 생성이 가능하다.
④ NOT NULL: 명시적으로 NULL 입력을 방지한다.
```

정답: ①

🧸 쉬운 해설:
①번은 “UNIQUE는 중복도 안 되고 NULL도 안 돼!”라고 말하는데,
NULL은 허용돼! 중복만 안 되는 거야!

📚 전문 해설:
UNIQUE 제약조건은 중복을 허용하지 않지만,
NULL은 하나 이상 허용됩니다.
PK는 NULL을 허용하지 않으며, 테이블당 1개만 가능합니다.

보기 설명:

보기 번호	설명	적절성
①	UNIQUE는 NULL 허용 → 틀림	❌
②	PK는 테이블당 1개 → 맞음	✅
③	FK는 여러 개 가능 → 맞음	✅
④	NOT NULL은 NULL 방지 → 맞음	✅

🧠 기억법:
UNIQUE = 중복 ❌, NULL ✅
PK = 중복 ❌, NULL ❌

필요 암기카드:

- 🃏 카드 61: PRIMARY KEY = 고유 이름표
- 🃏 카드 63: UNIQUE = 중복 ❌, NULL 가능
- 🃏 카드 64: NOT NULL = 빈칸 금지