

01 데이터베이스 기본



1 트랜잭션 ★★☆

(1) 트랜잭션

1 트랜잭션(Transaction)의 개념

인가받지 않은 사용자로부터 데이터를 보장하기 위해 DBMS가 가져야 하는 특성이자, 데이터베이스 시스템에서 하나의 논리적 기능을 정상적으로 수행하기 위한 작업의 기본 단위이다.

2 트랜잭션의 특성 [2020년 1회]

▼ 트랜잭션의 특성

특성	설명	주요기법
원자성 (Atomicity)	<ul style="list-style-type: none"> 분해가 불가능한 작업의 최소단위 연산 전체가 성공 또는 실패(All or Nothing) 하나라도 실패할 경우 전체가 취소되어야하는 특성 	<ul style="list-style-type: none"> Commit/Rollback 회복성 보장
일관성 (Consistency)	<ul style="list-style-type: none"> 트랜잭션이 실행 성공 후 항상 일관된 데이터베이스 상태를 보존해야하는 특성 	<ul style="list-style-type: none"> 무결성 제약조건 동시성 제어
격리성 (Isolation)	<ul style="list-style-type: none"> 트랜잭션 실행 중 생성하는 연산의 중간 결과를 다른 트랜잭션이 접근 불가한 특성 	<ul style="list-style-type: none"> Read Uncommitted Read Committed Repeatable Read Serializable
영속성 (Durability)	<ul style="list-style-type: none"> 성공이 완료된 트랜잭션의 결과는 영속적으로 데이터베이스에 저장하는 특성 	회복기법

두음쌤 한마디
트랜잭션의 특성
『ACID』
Atomicity / Consistency /
Isolation / Durability

잠깐! 알고가기

데이터베이스의 고립화 수준
(Isolation Level)

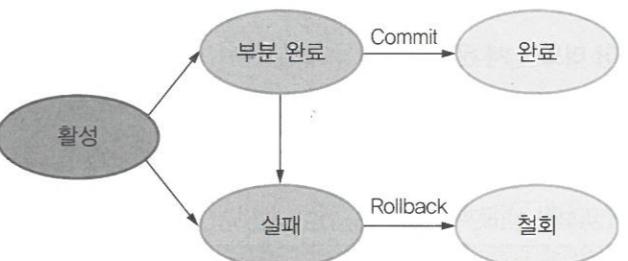
① Read Uncommitted
한 트랜잭션에서 연산(갱신) 중
인(아직 커밋되지 않은) 데이터를
다른 트랜잭션이 읽는 것을 허용.
연산(갱신) 중인 데이터에 대한
연산은 불허한다.

② Read Committed
한 트랜잭션에서 연산(갱신)을
수행할 때, 연산이 완료될 때까
지 연산대상 데이터에 대한 읽기
를 제한. 연산이 완료되어 커밋
된 데이터는 다른 트랜잭션이 읽
는 것을 허용한다.

③ Repeatable Read
선행 트랜잭션이 특정 데이터를
읽을 때, 트랜잭션 종료 시까지
해당 데이터에 대한 갱신/삭제를
제한한다.

④ Serializable Read
선행 트랜잭션이 특정 데이터 영
역을 순차적으로 읽을 때, 해당
데이터 영역 전체에 대한 접근을
제한한다.

3 트랜잭션의 상태 변화



▲ 트랜잭션의 상태전이도

▼ 트랜잭션의 상태 설명

상태	설명
활동 상태(Active)	초기 상태. 트랜잭션이 실행 중일 때 가지는 상태
부분 완료 상태 (Partially Committed)	마지막 명령문이 실행된 후에 가지는 상태
완료 상태(Committed)	트랜잭션이 성공적으로 완료된 후 가지는 상태
실패 상태(Failed)	정상적인 실행이 더 이상 진행될 수 없을 때 가지는 상태
철회 상태(Aborted)	트랜잭션이 취소되고 데이터베이스가 트랜잭션 시작 전 상태로 환원된 상태

두음쌤 한마디
트랜잭션의 상태
『활부완실철』
활동 / 부분완료 / 완료 / 실패
/ 철회 상태
→ 이번달 할부(활부)금 완전
싫조~(완실철)

4 트랜잭션 제어

트랜잭션 제어언어는 TCL(Transaction Control Language)이라고 하며,
트랜잭션의 결과를 허용하거나 취소하는 목적으로 사용되는 언어를 지칭한다.

▼ TCL 명령어

명령어	핵심	설명
커밋(COMMIT)	트랜잭션 확정	트랜잭션을 메모리에 영구적으로 저장하는 명령어
롤백(ROLLBACK)	트랜잭션 취소	트랜잭션 내역을 저장 무효화시키는 명령어
체크포인트(CHECKPOINT)	저장 시기 설정	ROLLBACK을 위한 시점을 지정하는 명령어

두음쌤 한마디
TCL 명령어
『커롤체』
커밋 / 롤백 / 체크 포인트

5 병행 제어(일관성 주요 기법)

① 병행 제어(Concurrency Control) 개념

병행 제어는 다수 사용자 환경에서 여러 트랜잭션을 수행할 때, 데이터베이스
일관성 유지를 위해 상호작용을 제어하는 기법이다.

② 병행 제어의 목적

- 데이터베이스의 공유를 최대화한다.
- 시스템의 활용도를 최대화한다.

- 데이터베이스의 일관성을 유지한다.
- 사용자에 대한 응답시간을 최소화한다.

③ 병행 제어 미보장 시 문제점

병행 제어 미보장 시 문제점은 갱신 손실, 현황 파악오류, 모순성, 연쇄복귀가 있다.

▼ 병행 제어 미보장 시 문제점

문제점	설명
갱신 손실 (Lost Update)	먼저 실행된 트랜잭션의 결과를 나중에 실행된 트랜잭션이 덮어쓸 때 발생하는 오류
현황 파악오류 (Dirty Read)	트랜잭션의 중간 수행 결과를 다른 트랜잭션이 참조하여 발생하는 오류
모순성 (Inconsistency)	두 트랜잭션이 동시에 실행되어 데이터베이스의 일관성이 결여되는 오류
연쇄복귀 (Cascading Rollback)	복수의 트랜잭션이 데이터 공유 시 특정 트랜잭션이 처리를 취소할 경우 트랜잭션이 처리한 곳의 부분을 취소하지 못하는 오류

④ 병행 제어 기법의 종류

- 병행 제어 기법에는 로킹, 타임 스탬프 순서가 있다.

▼ 병행 제어 기법의 종류

기법	설명
로킹 (Locking)	<ul style="list-style-type: none"> 같은 자원을 액세스하는 다중 트랜잭션 환경에서 DB의 일관성과 무결성을 유지하기 위해 트랜잭션의 순차적 진행을 보장하는 직렬화 기법 로킹의 특징은 다음과 같음 <ul style="list-style-type: none"> 데이터베이스, 파일, 레코드 등은 로킹 단위가 될 수 있음 로킹 단위가 작아지면 데이터베이스 공유도가 증가 로킹 단위가 작아지면 로킹 오버헤드가 증가 한꺼번에 로킹할 수 있는 객체의 크기를 로킹 단위라고 함
낙관적 검증	<ul style="list-style-type: none"> 트랜잭션이 어떠한 검증도 수행하지 않고 일단 트랜잭션을 수행하고, 트랜잭션 종료 시 검증을 수행하여 데이터베이스에 반영하는 기법
타임 스탬프 순서 (Time Stamp Ordering)	<ul style="list-style-type: none"> 트랜잭션과 트랜잭션이 읽거나 갱신한 데이터에 대해 트랜잭션이 실행을 시작하기 전에 타임 스탬프(Time Stamp)를 부여하여 부여된 시간에 따라 트랜잭션 작업을 수행하는 기법
다중버전 동시성 제어 (MVCC; Multi Version Concurrency Control)	<ul style="list-style-type: none"> 트랜잭션의 타임스탬프와 접근하려는 데이터의 타임스탬프를 비교하여 직렬 가능성이 보장되는 적절한 버전을 선택하여 접근하도록 하는 기법

6 데이터베이스 고립화 수준(격리성 주요 기법)

① 고립화 수준(Isolation Level) 개념

- 고립화 수준은 다른 트랜잭션이 현재의 데이터에 대한 무결성을 해치지 않기 위해 임금을 설정하는 정도이다.

② 고립화 수준 종류

- 고립화 수준 종류는 Read Uncommitted, Read Committed, Repeatable Read, Serializable Read가 있다.

▼ 고립화 수준 종류

수준	설명
Read Uncommitted	<ul style="list-style-type: none"> 한 트랜잭션에서 연산(갱신) 중인(아직 커밋되지 않은) 데이터를 다른 트랜잭션이 읽는 것을 허용하는 수준 연산(갱신) 중인 데이터에 대한 연산은 불허
Read Committed	<ul style="list-style-type: none"> 한 트랜잭션에서 연산(갱신)을 수행할 때, 연산이 완료될 때까지 연산 대상 데이터에 대한 읽기를 제한하는 수준 연산이 완료되어 커밋된 데이터는 다른 트랜잭션이 읽는 것을 허용
Repeatable Read	<ul style="list-style-type: none"> 선행 트랜잭션이 특정 데이터를 읽을 때, 트랜잭션 종료 시까지 해당 데이터에 대한 갱신·삭제를 제한하는 수준
Serializable Read	<ul style="list-style-type: none"> 선행 트랜잭션이 특정 데이터 영역을 순차적으로 읽을 때, 해당 데이터 영역 전체에 대한 접근 제한하는 수준

7 회복 기법(영속성 주요 기법)

① 회복 기법(Recovery) 개념

- 회복 기법은 트랜잭션을 수행하는 도중 장애로 인해 손상된 데이터베이스를 손상되기 이전의 정상적인 상태로 복구시키는 작업이다.

② 회복 기법 종류 [2020년 4회]

- 회복 기법 종류에는 로그 기반 회복 기법인 지연 갱신 회복 기법, 즉각(즉시) 갱신 회복 기법, 체크 포인트 회복 기법, 그림자 페이징 회복 기법이 있다.

▼ 회복 기법 종류

기법	설명
로그 기반 회복 기법	지연 갱신 회복 기법과 즉각 갱신 회복 기법이 있음
즉각 갱신 회복 기법 (Immediate Update)	<ul style="list-style-type: none"> 트랜잭션이 완료되기 전까지 데이터베이스에 기록하지 않는 기법 트랜잭션 수행 중 갱신 결과를 바로 DB에 반영하는 기법



두음쌤 한마다

회복 기법 종류

- 「회」로체크
회복 기법(로그 기반 회복 기법 / 체크 포인트 회복 기법 / 그림자 페이징 회복 기법)
→ 무전기의 회로를 체크 해 봐라!



두음쌤 한마다

병행 제어 미보장 시 문제점
「갱현모연」
갱신 손실 / 현황 파악오류 /
모순성 / 연쇄복귀
→ 갱현아! 애들 다 모였(모연)나?



두음쌤 한마다

병행 제어 기법
「로 낙타다」
로킹 / 낙관적 검증 / 타임 스탬프 순서 / 다중버전 동시성
→ 록(로) 콘서트에서 낙타를 탄다.

기법	설명
체크 포인트 회복 기법(Checkpoint Recovery)	장애 발생 시 검사점 이후에 처리된 트랜잭션에 대해서만 장애 발생 이전의 상태로 복원시키는 회복 기법
그림자 페이징 회복 기법(Shadow Paging Recovery)	데이터베이스 트랜잭션 수행 시 복제본을 생성하여 데이터베이스 장애 시 이를 이용해 복구하는 기법

(2) DDL

① 데이터 정의어(DDL; Data Definition Language)의 개념

- 데이터 정의어는 데이터를 정의하는 언어로서 ‘데이터를 담는 그릇을 정의하는 언어’이다.
- 테이블과 같은 데이터 구조를 정의하는 데 사용되는 명령어들로 특정 구조를 생성, 변경, 삭제, 이름을 바꾸는 데이터 구조와 관련된 명령어들을 데이터 정의어라고 부른다.

② DDL의 대상 [2020년 3회]

▼ DDL 대상

DDL 대상	설명				
도메인 (Domain)	<ul style="list-style-type: none"> 하나의 속성이 가질 수 있는 원자값들의 집합 속성의 데이터 타입과 크기, 제약조건 등의 정보 				
외부 스키마 (External Schema)	<ul style="list-style-type: none"> 데이터베이스의 구조, 제약조건 등의 정보를 담고 있는 기본적인 구조 스키마는 외부/개념/내부 3계층으로 구성되어 있음 				
스키마 (Schema)	<table border="1"> <tr> <td>개념 스키마 (Conceptual Schema)</td> <td> <ul style="list-style-type: none"> 사용자나 개발자의 관점에서 필요로 하는 데이터베이스의 논리적 구조 사용자 뷰를 나타냄 서브 스키마로 불림 </td> </tr> <tr> <td>내부 스키마 (Internal Schema)</td> <td> <ul style="list-style-type: none"> 데이터베이스의 전체적인 논리적 구조 전체적인 뷰를 나타냄 개체 간의 관계, 제약조건, 접근 권한, 무결성, 보안에 대해 정의 </td> </tr> </table>	개념 스키마 (Conceptual Schema)	<ul style="list-style-type: none"> 사용자나 개발자의 관점에서 필요로 하는 데이터베이스의 논리적 구조 사용자 뷰를 나타냄 서브 스키마로 불림 	내부 스키마 (Internal Schema)	<ul style="list-style-type: none"> 데이터베이스의 전체적인 논리적 구조 전체적인 뷰를 나타냄 개체 간의 관계, 제약조건, 접근 권한, 무결성, 보안에 대해 정의
개념 스키마 (Conceptual Schema)	<ul style="list-style-type: none"> 사용자나 개발자의 관점에서 필요로 하는 데이터베이스의 논리적 구조 사용자 뷰를 나타냄 서브 스키마로 불림 				
내부 스키마 (Internal Schema)	<ul style="list-style-type: none"> 데이터베이스의 전체적인 논리적 구조 전체적인 뷰를 나타냄 개체 간의 관계, 제약조건, 접근 권한, 무결성, 보안에 대해 정의 				
테이블(Table)	데이터 저장 공간				
뷰(View)	하나 이상의 물리 테이블에서 유도되는 가상의 테이블				
인덱스(Index)	검색을 빠르게 하기 위한 데이터 구조				



두음쌤 한마다

DDL 대상

「도스테부인」

도메인 / 스키마 / 테이블 / 뷰 / 인덱스
→ 18세기 독일에 도스테 부인(부인)이라는 사람이 있었다.

학습 Point

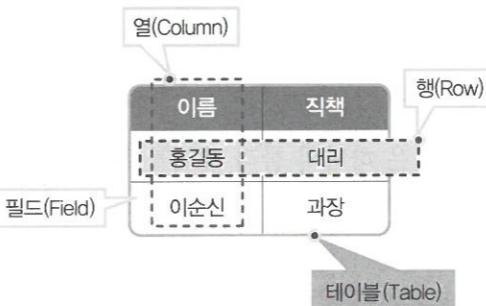
스키마는 시험문제에 나왔기 때문에 개념을 익혀두시기 바랍니다. 그리고 테이블, 뷰, 인덱스는 문법을 작성하실 수 있어야 합니다.

① 테이블(Table)

② 테이블의 개념

- 테이블은 데이터를 저장하는 항목인 필드(Field)들로 구성된 데이터의 집합체이다.
- 하나의 DB 내에 여러 개의 테이블로 구성될 수 있고, 레이션(Relation) 혹은 엔티티(Entity)라고도 불린다.

④ 테이블의 용어



▲ 테이블의 용어

▼ 테이블 관련 용어

용어	설명
튜플(Tuple) / 행(Row)	<ul style="list-style-type: none"> 테이블 내의 행을 의미하며 레코드(Record)라고도 함 튜플은 릴레이션(Relation)에서 같은 값을 가질 수 없음
애트리뷰트(Attribute) / 열(Column)	<ul style="list-style-type: none"> 테이블 내의 열을 의미 열의 개수를 디그리(Degree)라고 함
식별자(Identifier)	<ul style="list-style-type: none"> 여러 개의 집합체를 담고 있는 관계형 데이터베이스에서 각각의 구분 할 수 있는 논리적인 개념
카디널리티(Cardinality)	튜플(Tuple)의 개수
차수(Degree)	애트리뷰트(Attribute)의 개수
도메인(Domain)	하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자값들의 집합

② 뷰(View)

③ 뷰의 개념

- 뷰는 논리 테이블로서 사용자에게(생성 관점 아닌 사용 관점에서) 테이블과 동일하다.
- 아래 그림에서 ‘테이블 A’와 ‘테이블 B’는 물리 테이블을 의미하고, ‘뷰 C’는 두 개의 테이블을 이용하여 생성한 뷰를 의미한다.

핵인싸 퀴즈



1 C++, 자바에서는 참일 경우 ()으로, 거짓일 경우 ()으로 표현한다.

2 C++에서 문자 하나를 저장하고자 할 때 데이터 타입 선언은 ()이라고 한다.

3 ()은/는 프로그래밍 언어에서 실수치, 정수 자료형과 같은 여러 종류의 데이터를 식별하는 형태이다.

4 C언어에서 double은 ()값을 저장할 때 사용하는 데이터 타입이다.

5 ()은/는 저장하고자 하는 어떠한 값을 있을 때, 그 값을 주 기억 장치에 기억하기 위한 공간이다.

정답 1. true, false 2. char 3. 데이터 타입 4. 실수 5. 변수

TABLE A

컬럼 1	컬럼 2	컬럼 3	컬럼 4	컬럼 5	컬럼 6

TABLE B

컬럼 1	컬럼 2	컬럼 3

VIEW C

▲ 뷰의 개념

- 뷰는 '테이블 A'와 같은 하나의 물리 테이블로부터 생성 가능하며, 다수의 테이블 또는 다른 뷰를 이용해 만들 수 있다.
- 뷰와 같은 결과를 만들기 위해 조인 기능을 활용할 수 있으나, 뷰가 만들어져 있다면 사용자는 조인 없이 하나의 테이블을 대상으로 하는 단순한 질의어를 사용할 수 있다.

④ 뷰의 특징

▼ 뷰의 특징

특징	설명
논리적 데이터 독립성 제공	데이터베이스에 영향을 주지 않고 애플리케이션이 원하는 형태로 데이터에 접근 가능
데이터 조작 연산 간소화	애플리케이션이 원하는 형태의 논리적 구조를 형성하여 데이터 조작 연산을 간소화 예 회원 테이블에서 우수 회원을 뷰로 생성하여 활용
보안 기능(접근제어) 제공	특정 필드만을 선택해 뷰를 생성할 경우 애플리케이션은 선택되지 않은 필드의 조회 및 접근 불가
뷰 변경 불가	뷰 정의는 ALTER 문을 이용하여 변경할 수 없음 (뷰는 CREATE 문을 사용하여 정의, 뷰를 제거할 때에는 DROP 문을 사용)

⑤ 뷰의 목적

- 뷰를 사용하는 주된 이유는 단순 질의어를 사용할 수 있기 때문이다.
- FROM 절에 있는 하나의 뷰를 통해 뷰를 구성하는 복수의 테이블을 대체하는 단순성에 그 의의가 있다.
- 테이블의 중요 데이터 일부만을 제공할 수 있는 장단점이 있다.

▼ 뷰의 장점과 단점

구분	장/단점	설명
장점	논리적 독립성 제공	• 뷰는 논리 테이블(테이블의 구조가 변경되어도 뷰를 사용하는 응용 프로그램은 변경하지 않아도 됨)
	사용자 데이터 관리 용이	• 복수 테이블에 존재하는 여러 종류의 데이터에 대해 단순한 질의어 사용이 가능
	데이터 보안의 용이	• 중요 보안 데이터를 저장 중인 테이블에는 접근 불허 • 해당 테이블의 일부 정보만을 볼 수 있는 뷰에는 접근을 허용 • 보안 데이터에 대한 접근 제어 가능
단점	뷰 자체 인덱스 불가	• 인덱스는 물리적으로 저장된 데이터를 대상으로 하기에 논리적 구성인 뷰 자체는 인덱스를 가지지 못함
	뷰 정의의 변경 불가	• 뷰의 정의를 변경하려면 뷰를 삭제하고 재생성
	데이터 변경 제약 존재	• 뷰의 내용에 대한 삽입, 삭제, 변경 제약이 있음

③ 인덱스(Index)

ⓐ 인덱스의 개념

- 인덱스는 데이터를 빠르게 찾을 수 있는 수단으로서, 테이블에 대한 조회 속도를 높여 주는 자료 구조이다.
- 인덱스는 테이블의 특정 레코드 위치를 알려 주는 용도로 사용한다.

Index_name

Index(이름)	주소
김구	5
이육사	4
이순신	6
을지문덕	3
장보고	1
홍길동	2

Table_Avengers

Index(이름)	주소	생년월일	출생
장보고	1	19850111	완도
홍길동	2	19131130	장성
을지문덕	3	20001201	서울
이육사	4	19930105	평택
김구	5	19870712	해주
이순신	6	19500322	거제

▲ 인덱스 개념

ⓑ 인덱스의 특징

- 기본 키(PK; Primary Key) 컬럼은 자동으로 인덱스가 생성된다.
- 연월일이나 이름을 기준으로 하는 인덱스는 자동으로 생성되지 않는다.
- 테이블의 컬럼에 인덱스가 없는 경우, 테이블의 전체 내용을 검색한다(테이블 전체 스캔; Table Full Scan).
- 인덱스가 생성되어 있을 때 데이터를 빠르게 찾을 수 있다(인덱스 범위 스캔; Index Range Scan).
- 조건절에 '='로 비교되는 컬럼을 대상으로 인덱스를 생성하면 검색 속도를 높일 수 있다.



DDL 명령어
「크알드트」
CREATE / ALTER / DROP / TRUNCATE
→ 크리스마스를 위한 계란
알 두 트럭

④ 인덱스의 종류

▼ 인덱스 종류

유형	설명
순서 인덱스 (Ordered Index)	<ul style="list-style-type: none"> 데이터가 정렬된 순서로 생성되는 인덱스 B-Tree 알고리즘 활용(오름차순/내림차순 지정 가능)
해시 인덱스 (Hash Index)	<ul style="list-style-type: none"> 해시 함수에 의해 직접 데이터에 키 값으로 접근하는 인덱스 데이터 접근 비용이 균일, 튜플(Row) 양에 무관
비트맵 인덱스 (Bitmap Index)	<ul style="list-style-type: none"> 각 컬럼에 적은 개수 값이 저장된 경우 선택하는 인덱스 수정 변경이 적을 경우 유용(생년월일, 상품번호 등)
함수기반 인덱스 (Functional Index)	<ul style="list-style-type: none"> 수식이나 함수를 적용하여 만든 인덱스
단일 인덱스 (Singed Index)	<ul style="list-style-type: none"> 하나의 컬럼으로만 구성한 인덱스 주 사용 컬럼이 하나일 경우 사용
결합 인덱스 (Concatenated Index)	<ul style="list-style-type: none"> 두 개 이상의 컬럼으로 구성한 인덱스 WHERE 조건으로 사용하는 빈도가 높은 경우 사용
클러스터드 인덱스 (Clutered Index)	<ul style="list-style-type: none"> 기본 키(PK) 기준으로 레코드를 묶어서 저장하는 인덱스 저장 데이터의 물리적 순서에 따라 인덱스가 생성 특정 범위 검색 시 유리함

⑤ 인덱스의 스캔 방식

▼ 인덱스 스캔 방식

구분	설명	개념도
인덱스 범위 스캔 (Index Range Scan)	인덱스 루트 블록에서 리프 블록까지 수직적으로 탐색한 후에 리프 블록을 필요한 범위만 스캔하는 방식	
인덱스 전체 스캔 (Index Full Scan)	수직적 탐색 없이 인덱스 리프 블록을 처음부터 끝까지 수평적으로 탐색하는 방식	
인덱스 단일 스캔 (Index Unique Scan)	수직적 탐색만으로 데이터를 찾는 스캔 방식	
인덱스 생략 스캔 (Index Skip Scan)	선두 컬럼이 조건 절에 빠졌어도 인덱스를 활용하는 스캔방식	

③ DDL 명령어

- DDL 명령어는 CREATE, ALTER, DROP, TRUNCATE가 있다.

▼ DDL 명령어

구분	DDL 명령어	설명
생성	CREATE	데이터베이스 오브젝트 생성하는 명령어
수정	ALTER	데이터베이스 오브젝트 변경하는 명령어
삭제	DROP	데이터베이스 오브젝트 삭제하는 명령어
	TRUNCATE	데이터베이스 오브젝트 내용 삭제하는 명령어

④ TABLE 관련 DDL

① CREATE TABLE

- CREATE TABLE은 테이블을 생성하는 명령이다.
- CREATE TABLE에서 하나의 컬럼(속성)에 대해 '컬럼명 데이터타입 제약조건'으로 구성된다.

▼ CREATE TABLE 기본문법

```
CREATE TABLE 테이블명
(
    컬럼명 데이터타입 [제약조건],
    ...
);
```

▼ CREATE TABLE 상세문법

```
CREATE TABLE 테이블명
(
    컬럼명 데이터타입 PRIMARY KEY, -- 기본키 설정
    컬럼명 데이터타입 FOREIGN KEY REFERENCES 참조테이블(기본키), -- 외래키 설정
    컬럼명 데이터타입 UNIQUE,
    컬럼명 데이터타입 NOT NULL,
    컬럼명 데이터타입 CHECK(조건식), -- 제약조건 설정
    컬럼명 데이터타입 DEFAULT 값
);
```

예

```
CREATE TABLE 사원
(
    사번 VARCHAR(10) PRIMARY KEY,
    업무 VARCHAR(20) FOREIGN KEY REFERENCES 부서(부서코드),
    이름 VARCHAR(10) UNIQUE,
    생년월일 CHAR(8) NOT NULL,
    성별 CHAR(1) CHECK (성별 = 'M' OR 성별 = 'F'),
    입사일 DATE DEFAULT SYSDATE -- SYSDATE는 현재시간/날짜
);
```

- CREATE TABLE에 들어갈 수 있는 제약조건으로는 기본 키(PRIMARY KEY), 외래 키(FOREIGN KEY), UNIQUE, NOT NULL, CHECK, DEFAULT가 있다.

▼ CREATE TABLE 제약조건

제약조건	설명
PRIMARY KEY	<ul style="list-style-type: none"> 테이블의 기본 키를 정의 유일하게 테이블의 각 행을 식별
FOREIGN KEY	<ul style="list-style-type: none"> 외래 키를 정의 참조 대상을 테이블(컬럼명)로 명시 열과 참조된 테이블의 열 사이의 외래 키 관계를 적용하고 설정
UNIQUE	<ul style="list-style-type: none"> 테이블 내에서 얻은 유일한 값을 갖도록 하는 제약조건
NOT NULL	<ul style="list-style-type: none"> 해당 컬럼은 NULL 값을 포함하지 않도록 하는 제약조건
CHECK	<ul style="list-style-type: none"> 개발자가 정의하는 제약조건 참(TRUE)이어야 하는 조건을 지정
DEFAULT	<ul style="list-style-type: none"> 데이터를 INSERT 할 때 해당 컬럼의 값을 넣지 않는 경우 기본값으로 설정해 주는 제약조건

② ALTER TABLE

- ALTER TABLE은 테이블을 수정하는 명령이다.

③ ALTER TABLE 컬럼 추가

- 테이블에 필요한 컬럼을 추가하는 문법이다.
- CREATE TABLE의 컬럼에 사용되는 제약조건인 PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT를 ALTER TABLE에서도 사용할 수 있다.

▼ ALTER TABLE 컬럼 추가 문법

```
ALTER TABLE 테이블명 ADD 컬럼명 데이터타입 [제약조건];
```

예 ALTER TABLE 사원 ADD 전화번호 VARCHAR(11) UNIQUE;

→ 사원 테이블의 전화번호라는 컬럼에 대해 타입이 VARCHAR(11)이면서 UNIQUE 제약조건을 걸도록 추가

④ ALTER TABLE 컬럼 수정

- 테이블에 필요한 컬럼을 수정하는 문법이다.
- 테이블 생성을 위한 CREATE 문에 제약조건을 명시 후에 ALTER를 통해 테이블 제약조건의 변경이 가능하다.

▼ ALTER TABLE 컬럼 추가 문법

```
ALTER TABLE 테이블명 MODIFY 컬럼명 데이터타입 [제약조건];
```

예 ALTER TABLE 사원 MODIFY 이름 VARCHAR(30) NOT NULL;

→ 사원 테이블의 이름이라는 컬럼에 대해 타입이 VARCHAR(30)이면서 NOT NULL 제약조건을 걸도록 수정

⑤ ALTER TABLE 컬럼 삭제

- 테이블에 필요한 컬럼을 삭제하는 문법이다.

▼ ALTER TABLE 컬럼 삭제 문법

```
ALTER TABLE 테이블명 DROP 컬럼명;
```

예 ALTER TABLE 사원 DROP 생년월일;

→ 사원 테이블에 생년월일이라는 컬럼을 삭제

③ DROP TABLE

- DROP TABLE은 테이블을 삭제하는 명령이다.

▼ DROP TABLE 문법

```
DROP TABLE 테이블명 [CASCADE | RESTRICT];
```

예 DROP TABLE 사원;

→ 사원 테이블 삭제

- DROP TABLE 명령어의 옵션에는 CASCADE와 RESTRICT가 있다.
- CASCADE와 RESTRICT의 경우 외래 키(FOREIGN KEY)가 걸려 있을 때 해당한다.

▼ DROP TABLE 명령어 옵션

옵션	설명
CASCADE	참조하는 테이블까지 연쇄적으로 제거하는 옵션
RESTRICT	다른 테이블이 삭제할 테이블을 참조 중이면 제거하지 않는 옵션

④ TRUNCATE TABLE

- TRUNCATE TABLE은 테이블 내의 데이터들을 삭제하는 명령이다.

▼ TRUNCATE TABLE 문법

```
TRUNCATE TABLE 테이블명;
```

예 TRUNCATE TABLE 사원;

→ 사원 테이블 내의 모든 데이터를 삭제

5 VIEW 관련 DDL

① CREATE VIEW

- CREATE VIEW는 뷰를 생성하는 명령이다.

▼ CREATE VIEW 문법

CREATE VIEW 뷰이름 **AS**

조회쿼리;

예) 사원 테이블에 사번, 업무, 이름, 생년월일, 성별, 입사일이라는 컬럼이 있을 때 뷰 테이블 작성

CREATE VIEW 사원뷰 **AS**

SELECT 사번, 이름

FROM 사원

WHERE 성별 = 'M';

사원 테이블에서 성별 값이 'M'을 가진 사번, 이름으로 생성된
사원뷰라는 이름의 뷰 생성

학습 Point

CREATE VIEW를 만들 때 AS 뒤에 조회쿼리의 결과가 뷰로 만들어지기 때문에 SELECT 문을 익힌 후에 다시 보시면 이해가 더 쉽습니다.

잠깐 알고가기

UNION

집합연산자로 중복 행이 제거된
쿼리 결과 집합이다.

ORDER BY 절

속성을 정렬하고자 할 때 사용
한다(ASC: 오름차순, DESC: 내
림차순, ASC, DESC 키워드 생
략 시 오름차순 정렬).

학습 Point

VIEW는 수정(ALTER)할 수 없습니다. 삭제(DROP) 후에 다시
생성(CREATE)하거나, CREATE OR REPLACE VIEW 명령을
통해 기존의 뷰를 교체해야 합니다.

- VIEW 테이블의 SELECT 문에는 UNION이나 ORDER BY 절을 사용할 수 없다.
- 컬럼명을 기술하지 않으면 SELECT 문의 컬럼명이 자동으로 사용된다.

② CREATE OR REPLACE VIEW

- CREATE OR REPLACE VIEW는 뷰를 교체하는 명령이다.
- OR REPLACE라는 키워드를 추가하는 것을 제외하고는 CREATE VIEW와
사용 방법이 동일하다.

▼ CREATE OR REPLACE VIEW 기본문법

CREATE OR REPLACE VIEW 뷰이름 **AS**

조회쿼리;

③ DROP VIEW

▼ DROP VIEW 기본문법

DROP VIEW 뷰이름;

6 INDEX 관련 DDL

① CREATE INDEX

- CREATE INDEX는 인덱스를 생성하는 명령이다.
- UNIQUE는 생략 가능하고, 인덱스 걸린 컬럼에 중복 값을 허용하지 않는다.
- 복수 컬럼을 인덱스로 걸 수 있다.

▼ CREATE INDEX 문법

CREATE [UNIQUE] INDEX 인덱스명 **ON** 테이블명(컬럼명1, 컬럼명2, ...);

예) **CREATE INDEX** 사번인덱스 **ON** 사원(사번);

→ 사원 테이블의 사번 컬럼에 대해 사번인덱스라는 인덱스 명으로 인덱스 생성

② ALTER INDEX

- ALTER INDEX는 인덱스를 수정하는 명령어이다.
- 일부 DBMS는 ALTER INDEX를 제공하지 않는다.
- 기존 인덱스를 삭제하고 신규 인덱스를 생성하는 방식으로 사용을 권고
한다.

▼ ALTER INDEX 문법

ALTER [UNIQUE] INDEX 인덱스명 **ON** 테이블명(컬럼명1, 컬럼명2, ...);

예) **ALTER INDEX** 사번인덱스 **ON** 사원(사번);

→ 사원 테이블의 사번 컬럼에 대해 사번인덱스라는 인덱스 명으로 인덱스 생성

③ DROP INDEX

- DROP INDEX는 인덱스를 삭제하는 명령어이다.

▼ DROP INDEX 문법

DROP INDEX 인덱스명;

예) **DROP INDEX** 사번인덱스;

→ 사번인덱스라는 인덱스 명을 가지고 있는 인덱스를 삭제

(3) DML

1 데이터 조작어(DML; Data Manipulation Language)의 개념

- 데이터 조작어는 데이터베이스에 저장된 자료들을 입력, 수정, 삭제, 조회하는 언어이다.

2 DML 명령어

- 데이터 조작어(DML)의 유형에는 SELECT, INSERT, UPDATE, DELETE
가 있다.

학습 Point

Create, Read, Update, Delete
의 앞글자를 따서 CRUD라고 부
르고, CRUD는 데이터베이스 기
본 연산이다.



두음쌤 한마디

데이터베이스 DML

『세인업데』

SELECT / INSERT /

UPDATE / DELETE

→ 내 친구 세인이 집에 업데

▼ DML의 유형

유형	동작	설명
SELECT	조회	테이블 내 칼럼에 저장된 데이터를 조회
INSERT	삽입	테이블 내 칼럼에 데이터를 추가
UPDATE	갱신	테이블 내 칼럼에 저장된 데이터를 수정
DELETE	삭제	테이블 내 칼럼에 저장된 데이터를 삭제

③ SELECT(데이터 조회) 명령어

① SELECT 명령어 개념

- SELECT 명령어는 데이터의 내용을 조회할 때 사용하는 명령어이다.
- SELECT 명령어는 SELECT 절, FROM 절, WHERE 절, GROUP BY 절, HAVING 절, ORDER BY 절로 구성된다.

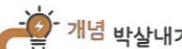
```
SELECT [ALL | DISTINCT] 속성명1, 속성명2...
  FROM 테이블명1, ...
  [WHERE 조건]
  [GROUP BY 속성명1, ...]
  [HAVING 그룹조건]
  [ORDER BY 속성 [ASC | DESC] ];
```

▼ SELECT 명령어

구분	설명		
		<ul style="list-style-type: none"> • 검색하고자 하는 속성명, 계산식을 기술 • 속성명 별칭은 AS를 사용하며 생략 가능함 • 2개 이상의 테이블을 대상으로 검색할 때는 '테이블명, 속성명'으로 표현 • 술어 부분은 ALL이 기본값 	
SELECT 절	ALL	<ul style="list-style-type: none"> • 모든 튜플을 검색할 때 사용 • SELECT 뒤에 명시하지 않은 경우 ALL로 인식 	
	DISTINCT	<ul style="list-style-type: none"> • 중복된 속성이 조회될 경우 그중 한 개만 검색(SELECT 뒤에 명시된 속성이 중복될 경우 한 개만 검색) 	
		<ul style="list-style-type: none"> • 질의에 의해 검색될 데이터들을 포함하는 테이블명을 기술 	
FROM 절			
WHERE 절	<ul style="list-style-type: none"> • 검색할 조건을 기술 		
GROUP BY 절	<ul style="list-style-type: none"> • 속성값을 그룹으로 분류하고자 할 때 사용 		
HAVING 절	<ul style="list-style-type: none"> • GROUP BY에 의해 분류한 후 그룹에 대한 조건 지정 		
ORDER BY 절	<ul style="list-style-type: none"> • 속성값을 정렬하고자 할 때(ASC: 오름차순, DESC: 내림차순, ASC, DESC 키워드 생략 시 오름차순 정렬) 사용 		

② SELECT 절 [2020년 1회]

- SELECT 절은 테이블에서 출력할 컬럼을 명시하기 위해 사용한다.



■ SELECT 절 예제

- 아래 테이블은 성적 테이블이라고 했을 때 쿼리 및 결과는 다음과 같다.

[성적 테이블]

이름	과목	학점
김철수	C언어	A
한유리	자료구조	A
신짱구	자료구조	A
이훈이	알고리즘	B

- 성적 테이블에 대해 쿼리와 쿼리 결과는 다음과 같다.

SELECT 과목 FROM 성적;	성적 테이블에서 과목을 모두 출력 과목 C언어 자료구조 자료구조 알고리즘
SELECT DISTINCT 과목 FROM 성적;	성적 테이블에서 과목 중에 중복된 값을 제거하여 출력 과목 C언어 자료구조 알고리즘
SELECT 과목 FROM 성적 WHERE 학점 = 'A'	성적 테이블에서 학점이 'A'인 과목을 모두 출력 과목 C언어 자료구조 자료구조
SELECT DISTINCT 과목 FROM 성적 WHERE 학점 = 'A'	성적 테이블에서 과목 중에 중복된 값을 제거하여 출력 과목 C언어 자료구조
SELECT COUNT(DISTINCT 과목) FROM 성적;	성적 테이블에서 과목 중에 중복된 값을 제거한 튜플의 갯수를 출력 COUNT(DISTINCT 과목) 3
SELECT DISTINCT 과목, 학점 FROM 성적;	성적 테이블에서 과목, 학점 두 컬럼이 모두 같은 값이면 제거 과목 학점 C언어 A 자료구조 A 알고리즘 B



데이터베이스 DML

『세인업데』

SELECT / INSERT /

UPDATE / DELETE

→ 내 친구 세인이 집에 업데



두음쌤 한마디

SELECT 명령문

『셀프 웨 구해오』

SELECT / FROM /

WHERE / GROUP BY /

HAVING / ORDER BY

→ 셀프 웨이터를 구해오라

잠깐! 알고가기

SELECT 명령문에서 "DISTINCT"는 중복된 튜플이 있으면 그 중 첫 번째 한 개만 검색한다.

학습 Point

2020년 제1회 기사 실기시험에서 DISTINCT를 이용한 계산이나왔습니다. 눈여겨 봐두길 바랍니다.

④ WHERE 절 [2020년 2회]

- WHERE 절 조건에서는 비교, 범위, 집합, 패턴, NULL, 복합조건이 있다.

▼ WHERE 절 문법

구분	연산자	설명
비교	=	값이 같은 경우 조회
	<, !=	값이 다른 경우 조회
	<, <=, >=	비교 연산에 해당하는 데이터 조회
범위	BETWEEN	<p>컬럼 BETWEEN 값1 AND 값2</p> <ul style="list-style-type: none"> 값1보다 크거나 같고, 값2보다 작거나 같은 데이터 조회 다음과 동일한 결과 <p>컬럼 >= 값1 AND 컬럼 <= 값2</p>
	IN	<p>컬럼 IN (값1, 값2, ...)</p> <p>컬럼이 IN 안에 포함된 경우의 데이터 조회</p>
집합	NOT IN	<p>컬럼 NOT IN (값1, 값2, ...)</p> <p>컬럼이 IN 안에 포함되어 있지 않은 경우의 데이터 조회</p>
패턴	LIKE	컬럼 LIKE 패턴
		컬럼이 패턴에 포함된 경우의 데이터 조회
		% 0개 이상의 문자열과 일치
		[] 1개의 문자와 일치
		[^] 1개의 문자와 불일치
NULL	IS NULL	<p>컬럼 IS NULL</p> <p>컬럼이 NULL인 데이터 조회</p>
	IS NOT NULL	<p>컬럼 IS NOT NULL</p> <p>컬럼이 NULL이 아닌 데이터 조회</p>
	AND	<p>조건1 AND 조건2</p> <p>조건1과 조건2 모두를 만족하는 데이터 조회</p>
	OR	<p>조건1 OR 조건2</p> <p>조건1과 조건2 둘 중 하나를 만족하는 데이터 조회</p>
복합조건	NOT, !	<p>NOT 조건</p> <p>조건에 해당하지 않는 데이터 조회</p>

개념 박살내기

■ WHERE 절 예제

BETWEEN	<pre>SELECT * FROM PRODUCT WHERE PRICE BETWEEN 5000 AND 8000;</pre> <p>→ 상품(PRODUCT) 테이블에서 가격(PRICE)이 5000보다 크거나 같고 8000보다 작거나 같은 튜플을 조회</p>
IN	<pre>SELECT * FROM PRODUCT WHERE PRICE IN (4000, 5000, 6000);</pre> <p>→ 상품(PRODUCT) 테이블에서 가격(PRICE)이 4000 또는 5000 또는 6000인 튜플을 조회</p>
LIKE	<pre>SELECT * FROM PRODUCT WHERE NAME LIKE '정보%';</pre> <p>→ 상품(PRODUCT) 테이블에서 이름(NAME)이 '정보'로 시작되는 문자열</p>
LIKE	<pre>SELECT * FROM PRODUCT WHERE NAME LIKE '[ABCD]%' ;</pre> <p>→ 첫 번째 문자가 'A' 또는 'B' 또는 'C' 또는 'D'인 문자열과 일치하는 문자열 검색</p>
IS NULL	<pre>SELECT * FROM PRODUCT WHERE PRICE IS NULL;</pre> <p>→ 상품(PRODUCT) 테이블에서 가격(PRICE)과 NULL 값인 경우의 튜플을 조회</p>

⑤ GROUP BY 절

- GROUP BY 절은 속성값을 그룹으로 분류하고자 할 때 사용한다.

개념 박살내기

■ GROUP BY 절 예제

- 아래 테이블은 급여 테이블이라고 했을 때 쿼리 및 결과는 다음과 같다.

[급여 테이블]

이름	직책	부서	급여
김철수	차장	마케팅	5000
한유리	차장	전산	4800
신짱구	사원	마케팅	2500
이훈이	사원	마케팅	2700

학습 Point

WHERE 조건과 BETWEEN 모두 기존 필기에서 기출되었던 부분입니다. 실기에도 나올 수 있으니 학습하시길 권장합니다.

학습 Point

SELECT 절에서 「컬럼명 AS 명칭」 형태로 쓰면 결과 출력 시 컬럼명이 별칭으로 표기됩니다. 추가로 AS는 생략 가능합니다.

<code>SELECT 직책, COUNT(직책), SUM(급여) FROM 급여 GROUP BY 직책;</code>	GROUP BY 절에 명시된 직책을 기준으로 그룹을 묶었을 때 직책별 건수(COUNT)와 급여의 합계(SUM)
<code>SELECT 부서, SUM(급여) AS 급여합계 FROM 급여 GROUP BY 부서;</code>	GROUP BY 절에 명시된 부서를 기준으로 그룹을 묶었을 때, 부서별 급여의 합계(SUM)
<code>SELECT 직책, 부서, SUM(급여) AS 급여합계 FROM 급여 GROUP BY 직책, 부서;</code>	GROUP BY 절에 명시된 직책, 부서를 기준으로 그룹을 묶었을 때 각 직책, 부서에 대한 급여 합계(SUM)
<code>SELECT COUNT(*) FROM 급여;</code>	GROUP BY 절이 없을 경우 전체 테이블이 하나의 그룹이 되고, 그룹에 해당하는 총 튜플의 수(COUNT)를 출력

④ HAVING 절

HAVING 절은 GROUP BY에 의해 분류한 후 그룹에 대한 조건을 지정할 때 사용한다.

<code>SELECT * FROM 성적 ORDER BY 이름;</code>	[성적 테이블]
<code>SELECT * FROM 성적 ORDER BY 과목, 이름;</code>	[과목별 성적 테이블]
<code>SELECT * FROM 성적 ORDER BY 학점 DESC, 이름 ASC;</code>	[학점별 성적 테이블]

⑤ ORDER BY 절

- ORDER BY 절은 속성값을 정렬하고자 할 때 사용한다.



■ ORDER BY 절 예제

- 아래 테이블은 성적 테이블이라고 했을 때 쿼리 및 결과는 다음과 같다.

[성적 테이블]

이름	과목	학점
김철수	C언어	A
한유리	자료구조	A
신짱구	자료구조	A
이훈이	알고리즘	B

- ORDER BY 절에 ASC와 DESC가 명시되어 있지 않은 경우 ASC가 기본값

- 이름에 대해 오름차순 정렬
- 문자열일 경우 가나다순으로 정렬

이름	과목	학점
김철수	C언어	A
신짱구	자료구조	A
이훈이	알고리즘	B
한유리	자료구조	A

- ORDER BY 절에 2개 이상의 속성이 있는 경우 먼저 선언된 속성으로 정렬 후, 같은 값일 때 다음 속성으로 정렬

이름	과목	학점
김철수	C언어	A
이훈이	알고리즘	B
신짱구	자료구조	A
한유리	자료구조	A

- ORDER BY 절에 2개 이상의 속성에 대해서 오름차순과 내림차순을 섞어서 사용할 수 있음

- 학점은 내림차순으로 정렬한 후 같은 값일 때 이름을 오름차순으로 정렬

이름	과목	학점
이훈이	알고리즘	B
김철수	C언어	A
신짱구	자료구조	A
한유리	자료구조	A

핵인싸 퀴즈

6 ()은/는 같은 타입의 변수들로 이루어진 집합이다.

7 자바에서 char형 배열 5개를 선언할 경우 `char[] arr = ()` `char[5]`라고 선언한다.

8 ()은/는 변수의 주솟값을 저장하는 공간이다.

9 `int *a = &p;`라고 할 때 `a`는 ()이고, `&p`는 `p`의 ()이다.

10 ()은/는 자바에서 중복된 원소를 허용하지 않는 집합(Set)의 성질을 가진 클래스로 순서가 중요하지 않은 성질을 가진다.

정답 6. 배열 7. new 8. 포인터 9. 표 인터 변수, 주소(값) 10. HashSet

② 조인(Join)

ⓐ 조인 개념

- 조인은 두 개 이상의 테이블을 연결하여 데이터를 검색하는 방법이다.
- 두 릴레이션으로부터 관련된 튜플들을 결합하여 하나의 튜플로 만드는 가장 대표적인 데이터 연결 방법이다.

▼ 논리적 조인 유형(상세)

유형	설명
내부 조인 (Inner Join)	공통 존재 컬럼의 값이 같은 경우를 추출하는 기법
외부 조인 (Outer Join)	외부 조인의 종류로는 왼쪽 외부 조인, 오른쪽 외부 조인, 완전 외부 조인이 있음
	왼쪽 외부 조인 (Left Outer Join) 왼쪽 테이블의 모든 데이터와 오른쪽 테이블의 동일 데이터를 추출하는 기법
	오른쪽 외부 조인 (Right Outer Join) 오른쪽 테이블의 모든 데이터와 왼쪽 테이블의 동일 데이터를 추출하는 기법
완전 외부 조인 (Full Outer Join)	양쪽의 모든 데이터를 추출하는 기법
교차 조인 (Cross Join)	조인 조건이 없는 모든 데이터 조합을 추출하는 기법
셀프 조인 (Self Join)	자기 자신에게 별칭을 지정한 후 다시 조인하는 기법

ⓑ 내부 조인

<code>SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2, ... FROM 테이블1 A [INNER] JOIN 테이블2 B ON 조인조건 [WHERE 검색조건];</code>	<ul style="list-style-type: none"> 같은 이름의 컬럼이 여러 테이블에 있을 경우 '별칭.컬럼명' 형태로 명시 INNER라는 키워드는 생략해도 내부 조인이 됨 검색조건을 추가할 경우 조인된 값에서 해당 조건에 맞는 결과만 출력되도록 설정
--	---

ⓒ 왼쪽 외부 조인

<code>SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2, ... FROM 테이블1 A LEFT [OUTER] JOIN 테이블2 B ON 조인조건 [WHERE 검색조건];</code>	<ul style="list-style-type: none"> OUTER라는 키워드는 생략해도 왼쪽 외부 조인이 됨 검색조건을 추가할 경우 조인된 값에서 해당 조건에 맞는 결과만 출력되도록 설정
---	--

ⓓ 오른쪽 외부 조인

<code>SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2, ... FROM 테이블1 A RIGHT [OUTER] JOIN 테이블2 B ON 조인조건 [WHERE 검색조건];</code>	<ul style="list-style-type: none"> OUTER라는 키워드는 생략해도 오른쪽 외부 조인이 됨 검색조건을 추가할 경우 조인된 값에서 해당 조건에 맞는 결과만 출력되도록 설정
--	---

ⓔ 완전 외부 조인

<code>SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2, ... FROM 테이블1 A FULL [OUTER] JOIN 테이블2 B ON 조인조건 [WHERE 검색조건];</code>	<ul style="list-style-type: none"> OUTER라는 키워드는 생략해도 완전 외부 조인이 됨 검색조건을 추가할 경우 조인된 값에서 해당 조건에 맞는 결과만 출력되도록 설정
---	--

ⓕ 교차 조인

<code>SELECT 컬럼1, 컬럼2, ... FROM 테이블1 CROSS JOIN 테이블2</code>	<ul style="list-style-type: none"> 조인 조건이 없는 모든 데이터 조합을 추출하기 때문에 ON절이 없음
---	---

ⓖ 셀프 조인

<code>SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2, ... FROM 테이블1 A [INNER] JOIN 테이블1 B ON 조인조건 [WHERE 검색조건];</code>	<ul style="list-style-type: none"> 같은 테이블명을 쓰고 별칭만 A, B와 같이 다르게 함 검색조건을 추가할 경우 조인된 값에서 해당 조건에 맞는 결과만 출력되도록 설정
--	---

개념 박살내기

▣ 조인 예시

- 도서와 도서가격이라는 테이블이 있을 때 조인 쿼리와 결과를 보여준다.

[도서]		[도서가격]	
책번호	책명	책번호	가격
111	운영체제	111	20,000
222	자료구조	222	25,000
555	컴퓨터구조	333	10,000
		444	15,000

핵인싸 퀴즈



11

()은/는 자바에서 크기가 가변적으로 변하는 선형리스트의 성질을 가진 클래스이며, 순서가 중요하며, 인덱스를 통해 접근이 가능하다.

12

()은/는 자바에서 데이터를 저장하는 노드가 이전 노드와 다음 노드의 상태를 알고 있는 링크드 리스트 자료구조를 구현한 클래스이다.

13

()은/는 자바에서 키와 값으로 구성된 객체를 저장하는 구조로 되어 있는 자료구조를 구현한 클래스이다.

14

()은/는 파이썬에서 중복된 원소를 허용하지 않는 집합의 성질을 가지고 있는 자료구조이다.

15

()은/는 파이썬에서 크기가 가변적으로 변하는 선형리스트의 성질을 가지고 있는 자료구조이다.

정답 11. ArrayList 12. LinkedList 13. HashMap 14. 세트형 15. 리스트형

핵인싸 퀴즈 ! ?

16 ()은/는 파이썬에서 초기에 선언한 값에서 값을 생성, 삭제, 수정이 불가능한 형태의 자료구조이다.

17 ()은/는 파이썬에서 키와 값으로 구성된 객체를 저장하는 구조로 되어 있는 자료구조이다.

18 파이썬에서 `a = [1, 2, 3, 4, 5]`라고 할 때 `a[-1]`의 값은 ()이다.

19 파이썬에서 `a = [1, 2, 3, 4, 5]`라고 할 때 `a[0]`의 값은 ()이다.

20 ()은/는 리스트에서 여러 개의 데이터에 동시에 접근하는 기법이다.

정답 16. 튜플형 17. 딕셔너리형 18. 5
19. 1 20. 리스트 슬라이싱

① 내부 조인

```
SELECT A.책번호, A.책명, B.가격
FROM 도서 A JOIN 도서가격 B
ON A.책번호 = B.책번호;
```

• '도서' 테이블은 A라는 별칭으로, '도서가격' 테이블은 B라는 명칭으로 설정 후 책번호가 같은 것끼리 조인

[결과]

A.책번호	A.책명	B.가격
111	운영체제	20,000
222	자료구조	25,000

② 왼쪽 외부 조인

```
SELECT A.책번호, A.책명, B.책번호, B.가격
FROM 도서 A LEFT JOIN 도서가격 B
ON A.책번호 = B.책번호;
```

• '도서' 테이블은 A라는 별칭으로, '도서가격' 테이블은 B라는 명칭으로 설정 후 왼쪽에 위치한 '도서' 테이블 기준으로 왼쪽 외부 조인

[결과]

A.책번호	A.책명	B.책번호	B.가격
111	운영체제	111	20,000
222	자료구조	222	25,000
555	컴퓨터구조	NULL	NULL

③ 오른쪽 외부 조인

```
SELECT A.책번호, A.책명, B.책번호, B.가격
FROM 도서 A RIGHT JOIN 도서가격 B
ON A.책번호 = B.책번호;
```

• '도서' 테이블은 A라는 별칭으로, '도서가격' 테이블은 B라는 명칭으로 설정 후 오른쪽에 위치한 '도서가격' 테이블 기준으로 오른쪽 외부 조인

[결과]

A.책번호	A.책명	B.책번호	B.가격
111	운영체제	111	20,000
222	자료구조	222	25,000
NULL	NULL	333	10,000
NULL	NULL	444	15,000

④ 완전 외부 조인

```
SELECT A.책번호, A.책명, B.책번호, B.가격
FROM 도서 A FULL JOIN 도서가격 B
ON A.책번호 = B.책번호;
```

• '도서' 테이블은 A라는 별칭으로, '도서가격' 테이블은 B라는 명칭으로 설정 후 완전 외부 조인

[결과]

A.책번호	A.책명	B.책번호	B.가격
111	운영체제	111	20,000
222	자료구조	222	25,000
NULL	NULL	333	10,000
NULL	NULL	444	15,000
555	컴퓨터구조	NULL	NULL

⑤ 교차 조인

```
SELECT A.책번호, A.책명, B.책번호, B.가격
FROM 도서 A CROSS JOIN 도서가격 B;
```

• '도서' 테이블은 A라는 별칭으로, '도서가격' 테이블은 B라는 별칭으로 설정 후 조건 설정 없이 교차 조인

[결과]

A.책번호	A.책명	B.책번호	B.가격
111	운영체제	111	20,000
111	운영체제	222	25,000
111	운영체제	333	10,000
111	운영체제	444	15,000
222	자료구조	111	20,000
222	자료구조	222	25,000
222	자료구조	333	10,000
222	자료구조	444	15,000
555	컴퓨터구조	111	20,000
555	컴퓨터구조	222	25,000
555	컴퓨터구조	333	10,000
555	컴퓨터구조	444	15,000

⑥ 셀프 조인

[도서]

책번호	책명	선수과목_책번호
111	운영체제	222
222	자료구조	555
555	컴퓨터구조	NULL

```
SELECT A.책번호, A.책명, B.책번호, B.책명
FROM 도서 A JOIN 도서 B
ON A.선수과목_책번호 = B.책번호;
```

• '도서' 테이블을 A, B라는 별칭으로 따로 설정
• 테이블 내의 다른 컬럼을 조건으로 두어 셀프 조인 수행

[결과]

A.책번호	A.책명	B.책번호	B.가격
111	운영체제	222	자료구조
222	자료구조	555	컴퓨터구조

21 a = [1, 2, 3, 4, 5]로 a를 정의한 후에 a[1 : 4] 같은 ()이다.
[2, 3, 4] ?

22 a = [1, 2, 3, 4, 5]로 a를 정의한 후에 a[0 : 4 : 2] 같은 ()이다.

23 ()은/는 변수, 상수, 함수 등 서로 구분하기 위해서 사용되는 이름이다.

24 ()은/는 컴퓨터 프로그래밍 언어에서 이미 문법적인 용도로 사용되고 있기는 단어로 식별자로 사용할 수 없다. 데이터 타입(int, float, …), 조건문(if, switch, case), 반복문(while, for, do), 루프 제어 명령문(break, continue), 함수 반환값(return)에 들어가는 키워드들이 이에 해당한다.

25 ()은/는 식별자 표기 시에 여러 단어가 이어지면 첫 단어 시작만 소문자로 표시하고, 각 단어의 첫 글자는 대문자로 지정하는 표기법이다.

정답 21. [1, 2, 3] 22. [1, 3] 23. 식별자
24. 예약어 25. 카멜 표기법

핵인싸 퀴즈

26 ()은/는 식별자 표기 시에 여러 단어가 이어지면 각 단어의 첫 글자는 대문자로 자정하는 표기법이다.

27 ()은/는 식별자 표기 시에 여러 단어가 이어지면 단어 사이에 언더바를 넣는 표기법이다.

28 ()은/는 식별자 표기 시에 두어에 자료형을 붙이는 표기법으로, int형일 경우 n, char형일 경우 c, 문자열일 경우 sz를 붙인다.

29 ()은/는 왼쪽 값은 오른쪽 값으로 나눈 나머지를 계산하는 연산자이다.

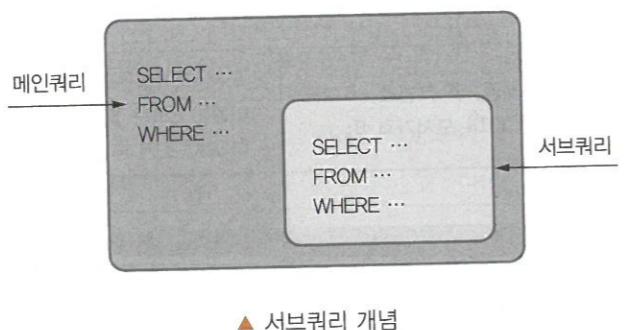
30 ()은/는 왼쪽 값을 오른쪽 값만큼 비트를 왼쪽으로 이동하는 연산자이다.

정답 26. 패스칼 표기법 27. 스네이크 표기법 28. 헝가리안 표기법 29. % 30. <<

③ 서브쿼리(Sub-Query)

① 서브쿼리 개념

- 서브쿼리는 SQL 문 안에 포함된 또 다른 SQL 문이다.
- 서브쿼리의 용도는 알려지지 않은 기준을 위한 검색을 위해 사용한다.
- 메인쿼리와 서브쿼리 관계는 주종 관계로서, 서브쿼리에 사용되는 컬럼 정보는 메인쿼리의 컬럼 정보를 사용할 수 있으나 역으로는 성립하지 않는다.



④ 서브쿼리 유형

▼ 서브쿼리 유형(위치 기준)

서브쿼리 종류	설명
SELECT 절 서브쿼리	<ul style="list-style-type: none"> 서브쿼리가 SELECT 절 안에 들어있는 형태 스칼라 서브쿼리(Scalar Sub-Query)라고도 불림 SELECT 절에 오는 서브쿼리는 반드시 단일 행을 리턴해야 함 SUM, COUNT, MIN, MAX 등과 같은 집계 함수가 많이 쓰임
FROM 절 서브쿼리	<ul style="list-style-type: none"> 서브쿼리가 FROM 절 안에 들어있는 형태 인라인 뷰(Inline Views)라고 불림 뷰(View)처럼 결과가 동적으로 생성된 테이블 형태로 사용할 수 있음
WHERE 절 서브쿼리	<ul style="list-style-type: none"> 서브쿼리가 WHERE 절 안에 들어있는 형태 중첩 서브쿼리(Nested Sub-Query)라고도 불림

개념 박살내기

■ 서브 쿼리 예제

도서와 도서가격이라는 테이블이 있을 때 서브쿼리와 결과를 보여준다.

[도서]		[도서가격]	
책번호	책명	책번호	가격
111	운영체제	111	20,000
222	자료구조	222	25,000
555	컴퓨터구조	333	10,000
		444	15,000

① SELECT 절 서브쿼리

```
SELECT (SELECT MAX(가격)
        FROM 도서 A
       WHERE A.책번호 = B.책번호
         AND 책명='자료구조')
      FROM 도서가격 B;
```

- A라는 이름으로 바꾼 '도서 테이블'의 '책번호'와 B라는 이름으로 바꾼 '도서가격' 테이블의 '책번호'와 같으면서 '책명'이 '자료구조'인 테이블에서의 책의 '가격' 중 가장 큰 값을 출력

[결과]

가격
25,000

② FROM 절 서브쿼리

```
SELECT MAX(가격)
  FROM 도서가격 A,
       (SELECT 책번호
        FROM 도서
       WHERE 책명='자료구조') B
      WHERE A.책번호 = B.책번호
```

- (SELECT 책번호 FROM 도서 WHERE 책명='자료구조')라는 결과 값을 B라는 이름으로 명명하면 B라는 테이블처럼 사용할 수 있음
- A라는 이름으로 바꾼 '도서 가격'의 '책번호'와 B라는 이름의 서브쿼리 결과값 중 '책번호'가 같은 책의 최대 '가격'을 출력

[결과]

가격
25,000

③ WHERE 절 서브쿼리

```
SELECT MAX(가격)
  FROM 도서가격
 WHERE 책번호 IN (SELECT 책번호
                    FROM 도서
                   WHERE 책명='자료구조');
```

- (SELECT 책번호 FROM 도서 WHERE 책명='자료구조') 서브쿼리에서 나온 '책번호'와 '도서가격' 테이블에 있는 '책번호'와 같은 책의 최대 '가격'을 출력

[결과]

가격
25,000

핵인싸 퀴즈

31 ()은/는 왼쪽에 있는 값이 오른쪽에 있는 값과 다르면 침을 반환하고, 그렇지 않으면 거짓을 반환하는 연산자이다.

32 ()은/는 모든 비트의 값을 반대로 바꾸는 반전 기능을 하는 연산자이다.

33 ()은/는 두 값은 비트로 연산하여 같은 비트의 값이 서로 다르면 해당 비트 값이 1이 되고, 그렇지 않으면 0이 되는 연산자(XOR 연산자)이다.

34 ()은/는 왼쪽의 변수에 오른쪽의 값을 대입하는 연산자이다.

35 a의 값이 0일 printf("%d", a++);의 출력값은 ()이다.

정답 31. != 32. ~ 33. ^ 34. = 35. 0

핵인싸 퀴즈

36 b의 값이 5일 printf("%d",
++b);의 출력값은 ()이다.

37 ()은/는 조건이 참일 경우
물음표(?)와 콜론(:) 사이의 값을
반환하고, 조건이 거짓일 경우
콜론(:)과 세미콜론(;) 사이의 값
을 반환하는 연산자이다.

38 a=5>3?2:1; 일 때, 삼항 연
산자를 실행 후 a의 값은 ()
이다.

39 C언어 printf 함수에서 문자
열을 출력하는 포맷 스트링은
()이다.

40 C언어 printf 함수에서 10진
수를 출력하는 포맷 스트링은
()이다.

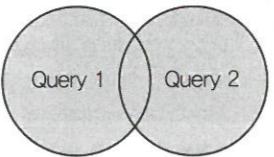
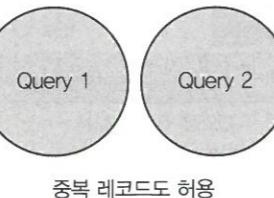
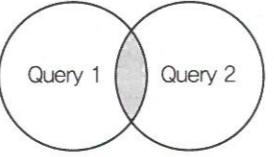
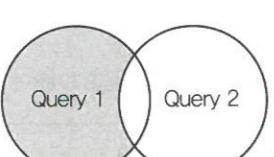
정답 36. 6 37. 삼항 연산자 38. 2
39. %s 40. %d

④ 집합 연산자(Set Operator)

① 집합 연산자 개념

- 집합 연산자는 테이블을 집합 개념으로 보고, 두 테이블 연산에 집합 연산자를 사용하는 방식이다.
- 집합 연산자는 여러 질의 결과를 연결하여 하나로 결합하는 방식을 사용한다.(집합 연산자는 2개 이상의 질의 결과를 하나의 결과로 만들어 준다.)

▼ 집합 연산자 유형

집합 연산자	구성도	설명
UNION		중복 행이 제거된 쿼리 결과를 반환하는 집합 연산자 중복 레코드를 제외
UNION ALL		중복 행이 제거되지 않은 쿼리 결과를 반환 하는 집합 연산자 중복 레코드도 허용
INTERSECT		두 쿼리 결과에 공통적으로 존재하는 결과를 반환하는 집합 연산자 중복 레코드만 포함
MINUS		첫 쿼리에 있고 두 번째 쿼리에는 없는 결과를 반환하는 집합 연산자 비교 레코드 제외

개념 박살내기

■ 집합 연산자 예제

[EMP 테이블]

EMPNO	ENAME	JOB	SAL
1001	홍길동	대리	1000
1002	임꺽정	과장	1500
1003	장길산	차장	2000
1004	강은미	부장	2500

① UNION

```
SELECT ENAME
  FROM EMP
 WHERE SAL <= 2000
UNION
SELECT ENAME
  FROM EMP
 WHERE SAL >= 1500;
```

EMP 테이블에서 SAL이 2000보다 작거나 같은 직원의 이름과 EMP 테이블에서 SAL이 1500보다 크거나 같은 직원의 이름의 합집합을 중복을 제거한 후 결과를 반환한다.

- SAL이 2000보다 작거나 같은 직원의 이름을 조회결과는 다음과 같다.

ENAME
홍길동
임꺽정
장길산

- SAL이 1500보다 크거나 같은 직원의 이름을 조회결과는 다음과 같다.

ENAME
임꺽정
장길산
강은미

- UNION의 결과는 다음과 같다.

ENAME
홍길동
임꺽정
장길산
강은미

② UNION ALL

```
SELECT ENAME
  FROM EMP
 WHERE SAL <= 2000
UNION ALL
SELECT ENAME
  FROM EMP
 WHERE SAL >= 1500;
```

EMP 테이블에서 SAL이 2000보다 작거나 같은 직원의 이름과 EMP 테이블에서 SAL이 1500보다 크거나 같은 직원의 이름의 합집합을 중복을 포함하여 결과를 반환한다.

핵인싸 퀴즈

41 C++에서 std::cout, std::cin 등을 사용하기 위해서는 () 헤더를 선언해야 한다.

42 C++에서 출력 함수로 가장 많이 사용하는 함수는 ()이다.

43 C언어는 개행을 할 때 printf 안에 () 문자열을 넣어준다.

44 C 언어는 std::cout 함수에 \n 문자열을 넣어도 되지만, () 함수를 사용할 수 있다.

45 자바는 출력 후 개행을 하기 위해 System.out.() 함수를 사용한다.

정답 41. iostream.h 42. std::cout 43. \n 44. std::endl 45. println

핵심 퀴즈



46 파이썬에서 값을 출력하기 위한 함수로 () 함수가 있다.

47 C언어에서 입력함수인 scanf를 사용하기 위해서는 () 헤더를 선언해야 한다.

48 ()은/는 조건이 참인지 거짓인지에 따라 경로를 선택하는 조건문이다.

49 C, C++, 자바에서는 if ~ else if ~ else 문이 파이썬에서는 if ~ () ~ else 문과 같다.

50 switch 문의 식이 어떠한 case의 값도 만족하지 않으면 ()으로 진입해 명령문 실행 한다.

정답 46. print 47. stdio.h 48. if 문
49. elif 50. default

- SAL이 2000보다 작거나 같은 직원의 이름을 조회결과로 '홍길동', '임꺽정', '장길산'이 나온다.
- SAL이 1500보다 크거나 같은 직원의 이름을 조회결과로 '임꺽정', '장길산', '강은미'가 나온다.
- UNION ALL은 중복된 값이 제거되지 않으며 결과는 아래와 같다.

ENAME
홍길동
임꺽정
장길산
임꺽정
장길산
강은미

③ INTERSECT

```
SELECT ENAME
  FROM EMP
 WHERE SAL <= 2000
INTERSECT
SELECT ENAME
  FROM EMP
 WHERE SAL >= 1500;
```

EMP 테이블에서 SAL이 2000보다 작거나 같은 직원의 이름과 EMP 테이블에서 SAL이 1500보다 크거나 같은 직원의 이름의 교집합 결과를 반환한다.

- SAL이 2000보다 작거나 같은 직원의 이름을 조회결과로 '홍길동', '임꺽정', '장길산'이 나온다.
- SAL이 1500보다 크거나 같은 직원의 이름을 조회결과로 '임꺽정', '장길산', '강은미'가 나온다.
- INTERSECT는 중복된 값만 나타나며 결과는 아래와 같다.

ENAME
임꺽정
장길산

④ MINUS

```
SELECT ENAME
  FROM EMP
 WHERE SAL <= 2000
MINUS
SELECT ENAME
  FROM EMP
 WHERE SAL >= 1500;
```

EMP 테이블에서 SAL이 2000보다 작거나 같은 직원의 이름과 EMP 테이블에서 SAL이 1500보다 크거나 같은 직원의 이름의 차집합 결과를 반환한다.

- SAL이 2000보다 작거나 같은 직원의 이름을 조회결과로 '홍길동', '임꺽정', '장길산'이 나온다.
- SAL이 1500보다 크거나 같은 직원의 이름을 조회결과로 '임꺽정', '장길산', '강은미'가 나온다.
- MINUS는 데이터 집합을 기준으로 다른 데이터 집합과 공통 항목을 제외한 결과만 추출한다.

ENAME
홍길동

4 INSERT(데이터 삽입) 명령어

INSERT는 데이터의 내용을 삽입할 때 사용하는 명령어이다.

```
INSERT INTO 테이블명(속성명1, ...)
VALUES (데이터1, ...);
```

- 속성과 데이터 개수, 데이터 타입이 일치해야 함
- 속성명은 생략 가능
- 속성의 타입이 숫자인 경우 데이터는 따옴표를 붙이지 않아도 되며, 문자열인 경우 따옴표를 붙여야 함

예 [학생] 테이블에 학번이 6677, 성명 '장길산', 학년이 3학년, 수강과목은 '수학' 인 학생을 삽입

```
→ INSERT INTO 학생(학번, 성명, 학년, 수강과목)
      VALUES (6677, '장길산', 3, '수학');
```

5 UPDATE(데이터 변경) 명령어

UPDATE는 데이터의 내용을 변경할 때 사용하는 명령어이다.

```
UPDATE 테이블명
      SET 속성명 = 데이터, ...
      WHERE 조건;
```

- UPDATE 명령문은 WHERE 절을 통해 어떤 조건이 만족할 경우에만 특정 컬럼의 값을 수정하는 용도로 자주 사용됨

예 [학생] 테이블에 장길산의 주소를 인천으로 수정

```
→ UPDATE 학생
      SET 주소 = '인천'
      WHERE 이름 = '장길산';
```

6 DELETE(데이터 삭제) 명령어 [2020년 4회]

DELETE는 데이터의 내용을 삭제할 때 사용하는 명령어이다.

```
DELETE FROM 테이블명
      WHERE 조건;
```

- 모든 레코드를 삭제할 때는 WHERE 절 없이 DELETE만 사용
- 레코드를 삭제해도 테이블 구조는 남아 있어서 디스크에서 테이블을 완전히 삭제하는 DROP 명령과는 다름

예 [학생] 테이블에 장길산에 대한 튜플을 삭제

```
→ DELETE FROM 학생
      WHERE 이름 = '장길산'
```

(4) DCL

① 데이터 제어어(DCL; Data Control Language)의 개념

- 데이터 제어어는 데이터베이스 관리자가 데이터 보안, 무결성 유지, 병행 제어, 회복을 위해 관리자(DBA)가 사용하는 제어용 언어이다.
- 데이터 제어어의 유형에는 GRANT, REVOKE가 있다.

▼ DCL의 유형

유형	동작	설명
GRANT	사용 권한 부여	관리자(DBA)가 사용자에게 데이터베이스에 대한 권한을 부여하는 명령어
REVOKE	사용 권한 회수	관리자(DBA)가 사용자에게 부여했던 권한을 회수하기 위한 명령어

① GRANT(권한 부여) 명령어

- GRANT는 데이터베이스 관리자(DBA; Database Administrator)가 사용자에게 데이터베이스에 대한 권한을 부여하는 명령어이다.

GRANT 권한 ON 테이블 TO 사용자; 관리자가 사용자에게 테이블에 대한 권한을 부여

예 관리자가 사용자 장길산에게 '학생' 테이블에 대해 UPDATE 할 수 있는 권한 부여
→ GRANT UPDATE ON 학생 TO 장길산;

② REVOKE(권한 회수) 명령어

- REVOKE는 데이터베이스 관리자(DBA)가 사용자에게 부여했던 권한을 회수하기 위한 명령어이다.

REVOKE 권한 ON 테이블 FROM 사용자 관리자가 사용자에게 부여했던 테이블에 대한 권한을 회수

예 관리자가 사용자 장길산에게 '학생' 테이블에 대해 UPDATE 할 수 있는 권한을 회수
→ REVOKE UPDATE ON 학생 FROM 장길산;



두음샘 한마디

GRANT 명령어

「그온투」

GRANT 권한 ON 테이블 TO 사용자



두음샘 한마디

REVOKE 명령어

「리온프」

REVOKE 권한 ON 테이블 FROM 사용자

NCS
지/피/지/기

기출문제

[단답형]

01 ▶ 2020년 1회
트랜잭션의 특성 중 일관성, 지속성 외 2개의 특성을 쓰시오.

①

②

[해설]

• 원자성은 하나라도 실패할 경우 전체가 취소되어야 하는 특성이다.

• 격리성(또는 고립성)은 트랜잭션 실행 중 생성하는 연산의 중간 결과를 다른 트랜잭션이 접근 불가한 특성이다.

[약술형]

02 ▶ 2020년 2회
트랜잭션 Rollback에 대해 설명하시오.

[해설]

Rollback (ROLLBACK) 오류가 발생했을 때, 오류 이전의 특정 시점(SAVEPOINT, CHECKPOINT) 상태로 되돌려주는 제어어

커밋 (Commit) 트랜잭션을 메모리에 영구적으로 저장하는 제어어

체크포인트 (CHECKPOINT) ROLLBACK을 위한 시점을 지정하는 제어어

[단답형]

03 ▶ 2020년 1회
STUDENT 테이블에 컴퓨터과 학생 50명, 인터넷과 학생 100명, 사무자동화학과 학생 50명의 정보가 저장되어 있을 때, 다음 SQL 문의 실행 결과의 투플 수는?
(단, DEPT 칼럼은 학과명이다.)

- ① SELECT DEPT FROM STUDENT;
- ② SELECT DISTINCT DEPT FROM STUDENT
- ③ SELECT COUNT(DISTINCT DEPT) FROM STUDENT WHERE DEPT= '컴퓨터과'

①

②

③

해설 • 단순 SELECT(조건검색)의 경우 전체 테이블의 튜플을 검색하기 때문에 200건($=50+100+50$)이 조회된다.

• DISTINCT(중복제거)의 경우 동일한 튜플을 제거하고 검색하기 때문에 '컴퓨터과' 1건, '인터넷과' 1건, '사무자동화학과' 1건으로 총 3건이 조회된다.

• GROUP BY가 없을 경우 전체 테이블에서의 튜플 개수를 COUNT를 통해 출력하기 때문에 출력값을 표현한 1개가 튜플이다.

[단답형]

04 ▶ 2020년 2회
학생 테이블은 학번(VARCHAR), 이름(VARCHAR), 학년(NUMBER), 수강과목(VARCHAR), 점수(NUMBER), 연락처(VARCHAR)를 속성으로 가진다. 아래 조건을 만족하는 SQL 문을 작성하시오.

- 1) 학생 테이블에서 3, 4학년인 학번, 이름을 조회한다.
- 2) IN 연산자 사용해야 한다.

[학생 테이블]

학번	이름	학년	수강과목	점수	연락처
1000	김정미	1	알고리즘	90	010-1111-2222
2000	강은미	2	데이터베이스	95	010-2222-2222
3000	홍길동	3	전산수학	90	010-3333-3333
4000	장길산	4	운영체제	95	010-4444-4444

[해설] WHERE 절에서 IN은 다음과 같이 작성한다.

컬럼 IN (값1, 값2, ...)

컬럼이 IN 안에 포함된 경우의 데이터 조회한다.

[단답형]

05 ▶ 2020년 4회
데이터베이스의 회복(Recovery) 기법 중 Rollback 시 Redo, Undo가 모두 실행되는 트랜잭션 처리법으로 트랜잭션 수행 중 간신희 결과를 바로 DB에 반영하는 기법은 무엇인가?

해설 • 회복 기법 중 로그 기반 회복 기법에는 자연 간신희 회복 기법, 즉각 간신희 회복 기법이 있으며, 트랜잭션 수행 중에 간신희 결과를 바로 DB에 반영하는 기법은 즉각 간신희 회복 기법이다.



NCS

지/피/지/기

기출문제

- ▶ 2020년 2회
06 STUDENT 테이블의 NAME 속성에 IDX_NAME 이름으로 인덱스 생성하는 SQL 문을 작성하시오.

[STUDENT 테이블]

STID	NAME	SCORE	DEPTID
1000	김정미	90	1
2000	강은미	95	2
3000	홍길동	90	3
4000	장길산	95	4

해설 CREATE INDEX 문법은 다음과 같다.

```
CREATE [UNIQUE] INDEX 인덱스명 ON 테이블명(컬럼명1,
    컬럼명2, ...);
```

- ▶ 2020년 3회
07 학생 테이블에서 이름이 민수인 튜플을 삭제하는 SQL 문을 작성하시오.

[학생] 테이블

학번	이름	점수	과목이름
1000	김정미	90	알고리즘
2000	강은미	95	데이터베이스
3000	홍길동	90	전산수학
4000	민수	95	운영체제

정답

01. 원자성(Atomicity), 격리성 또는 고립성(Isolation) 02. 트랜잭션 처리 중 오류가 발생했을 때, 오류 이전의 특정 시점(SAVEPOINT, CHECKPOINT) 상태로 되돌려주는 제어어(명령어)이다. 03. ① 200, ② 3, ③ 1 04. SELECT 학번, 이름 FROM 학생 WHERE 학번 IN (3, 4); 05. 즉각 갱신 회복 기법 06. CREATE INDEX IDX_NAME ON STUDENT(NAME); 07. DELETE FROM 학생 WHERE 이름 = '민수'; 08. 데이터베이스의 구조, 제약조건 등의 정보를 담고 있는 기본적인 구조이다. 09. ① ALTER, ② ADD

해설 튜플을 삭제하는 경우 DELETE 명령어를 사용한다.

```
DELETE FROM 테이블명
WHERE 조건;
```

- ▶ 2020년 3회
08 DB 스키마에 대해서 서술하시오.

해설

도메인	<ul style="list-style-type: none"> 하나의 속성이 가질 수 있는 원자값들의 집합 속성의 데이터 타입과 크기, 제약조건 등의 정보
스키마	<ul style="list-style-type: none"> 데이터베이스의 구조, 제약조건 등의 정보를 담고 있는 기본적인 구조
테이블	<ul style="list-style-type: none"> 데이터 저장 공간 하나 이상의 물리 테이블에서 유도되는 가상의 테이블
뷰	<ul style="list-style-type: none"> 검색을 빠르게 하기 위한 데이터 구조
인덱스	

- ▶ 2020년 3회
09 학생 테이블에 주소 속성을 추가하는 SQL 문을 작성하시오.

```
( ① ) TABLE 학생 ( ② ) 주소 VARCHAR(20);
```

①

②

해설 ALTER TABLE 컬럼 추가하기 위한 문법은 다음과 같다.

```
ALTER TABLE 테이블명 ADD 컬럼명 데이터타입 [제약조건];
```



NCS

천/기/누/설

예상문제

- ▶ 2020년 2회
01 트랜잭션에 대해 서술하시오.

- ▶ 2020년 3회
02 다음 괄호 () 안에 알맞은 말을 쓰시오.

스키마는 사용자나 개발자의 관점에서 필요로 하는 데이터베이스의 논리적 구조를 나타낸 (①) 스키마, 데이터베이스의 전체적인 논리적 구조를 나타낸 (②) 스키마, 물리적 저장장치의 관점에서 보는 데이터베이스의 구조를 나타낸 (③) 스키마가 있다.

①

②

③

- ▶ 2020년 3회
03 인덱스(Index)에 대해 서술하시오.

외부 스키마 (External Schema)	<ul style="list-style-type: none"> 사용자나 개발자의 관점에서 필요로 하는 데이터베이스의 논리적 구조 사용자 뷰를 나타냄 서브 스키마로 불림
개념 스키마 (Conceptual Schema)	<ul style="list-style-type: none"> 데이터베이스의 전체적인 논리적 구조 전체적인 뷰를 나타냄 개체 간의 관계, 제약 조건, 접근 권한, 무결성, 보안에 대해 정의
내부 스키마 (Internal Schema)	<ul style="list-style-type: none"> 물리적 저장장치의 관점에서 보는 데이터베이스 구조 실제로 데이터베이스에 저장될 레코드의 형식을 정의하고 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 등을 표현

- ▶ 2020년 3회
04 사원 테이블의 '사번' 컬럼에 대해 '사번인덱스'라는 인덱스 명으로 인덱스를 생성하고자 한다. 알맞은 쿼리를 쓰시오.

해설 CREATE INDEX 문법은 다음과 같다.

```
CREATE [UNIQUE] INDEX 인덱스명 ON 테이블명(컬럼명1, 컬럼명2, ...);
```

- ▶ 2020년 3회
05 다음은 사람이라는 테이블을 만들려고 한다. 사람이라는 테이블에는 이름, 성별이라는 컬럼이 있는데, 이름은 VARCHAR(10) 데이터 타입을 가지고, 성별은 CHAR(1) 데이터 타입을 가지고 있다. 성별은 추가로 'M'과 'F' 값만 가질 수 있도록 제약 조건을 걸고자 한다. 알맞은 쿼리를 작성하시오.

해설 CREATE TABLE의 상세 문법은 다음과 같다.

```
CREATE TABLE 테이블명 (
    컬럼명 데이터타입 PRIMARY KEY, -- 기본 키 설정
    컬럼명 데이터타입 FOREIGN KEY REFERENCES 참조테이블(기본 키), -- 외래 키 설정
    컬럼명 데이터타입 UNIQUE,
    컬럼명 데이터타입 NOT NULL,
    컬럼명 데이터타입 CHECK(조건식) -- 제약 조건 설정
    컬럼명 데이터타입 DEFAULT 값
);
```



NCS

천/기/누/설

예상문제

- 06 [기술형] 사원 테이블에 VARCHAR(11) 타입으로 전화번호라는 컬럼을 추가하려고 한다. 알맞은 쿼리를 작성하시오.

해설 ALTER TABLE에서 컬럼을 추가하는 문법은 다음과 같다.

```
ALTER TABLE 테이블명 ADD 컬럼명 데이터타입 [제약조건];
```

- 07 [기술형] 사원 테이블에 사번, 업무, 이름, 생년월일, 성별, 입사 일이라는 컬럼이 있을 때 사원 테이블에서 성별 값이 'M'을 가진 사번, 이름으로 생성된 '사원부'라는 이름의 뷰 생성하는 쿼리를 작성하시오.

해설 뷰를 생성하기 위해서는 다음과 같은 문법을 사용한다.

```
CREATE VIEW 뷰이름 AS  
조회쿼리;
```

- 08 [기술형] 다음 테이블은 학생 테이블의 일부이다. 해당 테이블에서 전공만 출력하는 쿼리를 쓰시오. 단, 전공명은 중복되지 않아야 한다.

[학생]

이름	전공	생년월일
이완용	일본어학과	580607
박영효	일본어학과	610612
기철	몽골어학과	840101
안중근	국문학과	790902

해설 • SELECT 명령어 문법은 다음과 같다.

```
SELECT [ALL | DISTINCT] 속성명1, 속성명2...  
FROM 테이블명1, ...  
[WHERE 조건]  
[GROUP BY 속성명1, ...]  
[HAVING 그룹조건]  
[ORDER BY 속성 [ASC | DESC] ];
```

• DISTINCT라는 키워드를 붙이면 중복된 속성이 조회될 경우 그중 한 개만 검색(SELECT 뒤에 명시된 속성이 중복될 경우 한 개만 검색)한다.

- 09 [기술형] 다음은 학생 테이블의 일부이다. '이' 씨 성을 가진 사람의 학번을 출력하는 쿼리를 작성하시오.

[학생]

이름	전공	학번
이완용	일본어학과	580607
박영효	일본어학과	610612
기철	몽골어학과	840101
안중근	국문학과	790902

해설 • WHERE 절에서 패턴을 이용할 경우 LIKE 키워드를 사용한다.

컬럼 LIKE 패턴

• LIKE의 패턴에 쓰이는 와일드카드는 다음과 같다.

%	0개 이상의 문자열과 일치
[]	1개의 문자와 일치
[^]	1개의 문자와 불일치
_	특정 위치의 1개의 문자와 일치

• LIKE 사용을 위한 예제는 다음과 같다.

```
WHERE 이름 LIKE '이%': '이'로 시작하는 사람 검색  
WHERE 이름 LIKE '%이': '이'로 끝나는 사람 검색  
WHERE 이름 LIKE '%이%': '이'가 들어가는 사람 검색  
WHERE 이름 LIKE '이_': '이'로 시작되고 '이' 뒤에 1글자만 있는 사람 검색  
WHERE 이름 LIKE '이__': '이'로 시작되고 '이' 뒤에 2글자만 있는 사람 검색  
WHERE 이름 LIKE '이_신': '이*신'이라는 사람 검색
```

[기술형]

- 10 다음은 학생 테이블에서 주소 컬럼이 있다. 주소 컬럼이 NULL이 아닌 주솟값을 출력하는 쿼리를 작성하시오.

해설 컬럼이 NULL이 아닌 데이터 조회할 경우 WHERE 절에서 IS NOT NULL 구문을 사용한다.

```
컬럼 IS NOT NULL;
```

<고객> 테이블

이름	나이	성별	주소
홍길동	20	남	경기도
임꺽정	59	남	전라도
유관순	35	여	경상도
나혜석	41	여	충청도
이순신	33	남	강원도

해설 • BETWEEN 문법은 다음과 같다.

```
컬럼 BETWEEN 값1 AND 값2
```

• 값1보다 크거나 같고, 값2보다 작거나 같은 데이터를 조회한다.

- 13 [기술형] 다음은 성적 테이블의 일부이다. 이름, 과목, 성적을 성적순(내림차순)으로 출력하는 쿼리를 작성하시오.

[성적]

이름	과목	성적
문무왕	프로그래밍	4.0
문무왕	알고리즘	4.5
장보고	알고리즘	3.5
장보고	자료구조	3.0

해설 ORDER BY 절은 속성값을 정렬하고자 할 때 사용한다. (ASC: 오름차순, DESC: 내림차순, ASC, DESC 키워드 생략 시 오름차순 정렬)

```
ORDER BY 속성 [ASC | DESC]
```



NCS

천/기/누/설

예상문제

- 14 [약술형] 다음과 같이 도서와 도서가격이라는 테이블이 있다. 책 번호 컬럼을 기준으로 Left Outer Join을 하여 [결과] 테이블과 같이 나오도록 쿼리를 작성하시오.

[도서]

책번호	책명
111	운영체제
222	자료구조
555	컴퓨터구조

[도서가격]

책번호	가격
111	20,000
222	25,000
333	10,000
444	15,000

[결과]

책번호	책명	책번호	가격
111	운영체제	111	20,000
222	자료구조	222	25,000
555	컴퓨터구조	NULL	NULL

해설 • 왼쪽 외부 조인 문법은 다음과 같다.

```
SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2, ...
FROM 테이블1 A LEFT [OUTER] JOIN 테이블2 B
ON 조인조건
[WHERE 검색조건];
```

• '도서' 테이블은 A라는 별칭으로, '도서가격' 테이블은 B라는 명칭으로 설정 후 왼쪽에 위치한 '도서' 테이블 기준으로 왼쪽 외부 조인한다.

- 15 [단답형] 다음은 직원 테이블이다. 다음 쿼리를 실행 결과 튜플 수는 얼마인지 쓰시오.

[직원]

사번	이름	연봉
1	홍길동	1,000
2	임꺽정	1,500
3	장길산	2,000

[쿼리]

```
SELECT * FROM 직원 WHERE SAL >= 1500
UNION ALL
SELECT * FROM 직원 WHERE SAL >= 2000;
```

해설 • 왼쪽 외부 조인 문법은 다음과 같다.

```
SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2, ...
FROM 테이블1 A LEFT [OUTER] JOIN 테이블2 B
ON 조인조건
[WHERE 검색조건];
```

• '도서' 테이블은 A라는 별칭으로, '도서가격' 테이블은 B라는 명칭으로 설정 후 왼쪽에 위치한 '도서' 테이블 기준으로 왼쪽 외부 조인한다.

- 16 [약술형] 다음은 테이블 및 컬럼명에 대한 명세이다. 사원명이 홍길동, 나이가 24, 급여가 300인 직원을 직원 테이블에 삽입하는 쿼리를 작성하시오.

- 테이블명: EMPLOYEE(직원 테이블)
- 컬럼명: NAME(사원명), AGE(나이), SALARY(급여)

해설 INSERT 명령문은 다음과 같다.

```
INSERT INTO 테이블명(속성명1, ...)
VALUES(데이터1, ...);
```

- 17 [약술형] 다음은 테이블 및 컬럼명에 대한 명세이다. 급여가 300 이하인 직원들을 직원 테이블에서 삭제하는 쿼리를 작성하시오.

- 테이블명: EMPLOYEE(직원 테이블)
- 컬럼명: NAME(사원명), AGE(나이), SALARY(급여)

- 18 관리자가 사용자 장길산에게 '학생' 테이블에 대해 UPDATE할 수 있는 권한을 부여하는 쿼리를 작성하시오.

해설 DELETE 명령문은 다음과 같다.

GRANT 명령어	그온투	GRANT 권한 ON 테이블 TO 사용자
-----------	-----	------------------------

해설 DELETE 명령문은 다음과 같다.

```
DELETE FROM 테이블명
WHERE 조건;
```

정답

- 데이터베이스 시스템에서 하나의 논리적 기능을 정상적으로 수행하기 위한 작업의 기본 단위이다.
- ① 외부, ② 개념, ③ 내부
- 데이터를 빠르게 찾을 수 있는 수단으로서 테이블에 대한 조회 속도를 높여 주는 자료 구조이다.
- CREATE INDEX 사번인덱스 ON 사원(사번);
- CREATE TABLE 사원 (이름 VARCHAR(10), 성별 CHAR(1) CHECK (성별 = 'M' OR 성별 = 'F'));
- ALTER TABLE 사원 ADD 전화번호 VARCHAR(11);
- CREATE VIEW 사원뷰 AS SELECT 사번, 이름 FROM 사원 WHERE 성별 = 'M';
- SELECT DISTINCT 전공 FROM 학생;
- SELECT 학번 FROM 학생 WHERE 이름 LIKE '이%';
- SELECT * FROM 교수 WHERE 전공 IN ('컴퓨터공학', '전자공학');
- SELECT 이름 FROM 고객 WHERE 나이 BETWEEN 50 AND 59 AND 성별 = '남';
- SELECT * FROM 성적 ORDER BY 성적 DESC;
- SELECT A.책번호, A.책명, B.책번호, B.가격 FROM 도서 A LEFT JOIN 도서가격 B ON A.책번호 = B.책번호;
- INSERT INTO EMPLOYEE(NAME, AGE, SALARY) VALUES('홍길동', 24, 300);
- DELETE FROM EMPLOYEE WHERE SALARY <= 300;
- GRANT UPDATE ON 학생 TO 장길산;

O2 응용 SQL 작성하기



1 집계성 SQL 작성 ★★★

(1) 데이터 분석 함수의 개념

학습 Point
데이터 분석 함수의 종류가 단답형 형태로 나올 가능성이 있습니다. 설명을 보고 함수를 떠올릴 수 있을 정도로 학습하세요!

- 총합, 평균 등의 데이터 분석을 위해서는 복수 행 기준의 데이터를 모아서 처리하는 것을 목적으로 하는 다중 행 함수이다.
- 데이터 분석을 위한 다중 행 함수의 공통적인 특성은 아래와 같다.

- 단일 행을 기반으로 산출하지 않고 복수 행을 그룹별로 모아 놓고 그룹당 단일 계산 결과를 반환한다.
- GROUP BY 구문을 활용하여 복수 행을 그룹핑한다.
- SELECT, HAVING, ORDER BY 등의 구문에 활용한다.

(2) 데이터 분석 함수의 종류

SQL 표준에서는 데이터 투플 간의 상호 연관 및 계산 분석을 위한 세 가지 함수가 있다.

▼ 데이터 분석 함수 종류

함수	설명
집계 함수	여러 행 또는 테이블 전체 행으로부터 하나의 결괏값을 반환하는 함수
그룹 함수	소그룹 간의 소계 및 중계 등의 중간 합계 데이터를 산출하는 함수
윈도 함수	데이터베이스를 사용한 온라인 분석 처리 용도로 사용하기 위해서 표준 SQL에 추가된 기능

(3) 집계 함수

① 집계 함수(Aggregate Function)의 개념

여러 행 또는 테이블 전체 행으로부터 하나의 결괏값을 반환하는 함수이다.

② 집계 함수 구문

▼ 집계 함수 구문

```
SELECT 컬럼1, 컬럼2, ..., 집계함수
  FROM 테이블명
 [WHERE 조건]
 GROUP BY 컬럼1, 컬럼2, ...
 [HAVING 조건식(집계함수 포함)]
```

- WHERE 조건으로 지정된 데이터 집합으로부터 그룹화된 집합에 대한 조건 선택 시에 HAVING을 사용하는 것이다.
- GROUP BY 구문 뒤에는 테이블을 구분하는 컬럼을 기재하여 그룹화한다.
- HAVING 구문은 그룹화된 집합에 대한 조건 지정 시 사용하고, 상수나 집약 함수, 집약 키를 사용할 수 있다.

① GROUP BY 구문

- SQL에서는 WHERE 구문을 활용하여 조건별 대상 ROW를 선택한다.
- 복수 ROW 대상의 데이터 분석 시 그룹핑 대상이 되는 부분을 선별할 필요가 있다.
- GROUP BY는 그와 같은 경우에 사용하며, 아래와 같은 특성을 가진다.

- NULL값을 가지는 ROW는 제외한 후 산출한다.
- SELECT에서 사용하는 것과 같은 ALIAS 사용이 불가하다.
- WHERE 구문 안에 포함되지 않는다.
- WHERE 구문은 GROUP BY보다 먼저 실행되고, 대상이 되는 단일 행을 사전에 선별하는 역할을 한다.

- GROUP BY 구문은 실제 구체적 데이터 분석값을 보고자 하는 컬럼 단위를 선정할 때 사용되는 기준이 되며, 이 부분의 조정을 통해 사용자가 원하는 분석 데이터를 볼 수 있게 해 준다.

② HAVING 구문

- HAVING 구문은 WHERE 구문 내에는 사용할 수 없는 집계 함수의 구문을 적용하여 복수 행의 계산 결과를 조건별로 적용하는 데 사용된다.

- 일반적으로 GROUP BY 뒤에 기재하면, GROUP BY 구문의 기준 항목이나 소그룹 집계 함수를 활용한 조건을 적용하는 데 사용한다.
- 쉽게 생각하면 GROUP BY 및 집계 함수에 대한 WHERE 구문이다.

③ 집계 함수의 종류

집계 특성상 숫자 유형의 계산에 사용되는 것이 대다수이나, MAX/MIN 또는 COUNT와 같이 문자열 유형의 최대/최소나 건수 계산 등에도 사용한다.

▼ 집계 함수 종류

집계 함수	내용
COUNT	복수 행의 줄 수를 반환하는 함수
SUM	복수 행의 해당 컬럼 간의 합계를 계산하는 함수
AVG	복수 행의 해당 컬럼 간의 평균을 계산하는 함수
MAX	복수 행의 해당 컬럼 중 최댓값을 계산하는 함수
MIN	복수 행의 해당 컬럼 중 최솟값을 계산하는 함수
STDDEV	복수 행의 해당 컬럼 간의 표준편차를 계산하는 함수
VARIAN	복수 행의 해당 컬럼 간의 분산을 계산하는 함수

④ 집계 함수 활용 예시

아래 테이블은 학생(STUDENT) 테이블이라고 했을 때 집계 함수에 대한 쿼리 및 결과는 다음과 같다.

학생명	국어	영어
유리	100	80
철수	50	70
영식	80	90
미선	70	100

▼ 집계 함수 예시

쿼리	설명
<code>SELECT COUNT(*) FROM STUDENT WHERE 국어 >= 80</code>	국어 점수가 80점 이상인 학생들(국어 >= 80)의 숫자를 알려줌 (2명)
<code>SELECT SUM(국어), AVG(영어) FROM STUDENT</code>	국어 점수의 합(300점), 영어 점수의 평균 값(85점)을 알려줌
<code>SELECT MAX(국어), MIN(국어) FROM STUDENT</code>	국어 점수 최고점(100점)과 최저점(50점)을 알려줌
<code>SELECT STDDEV(국어), VARIAN(국어) FROM STUDENT</code>	국어 점수의 표준편차(18.03), 분산(325)을 알려줌

(4) 그룹 함수

① 그룹 함수(Group Function)의 개념

테이블의 전체 행을 하나 이상의 컬럼을 기준으로 컬럼 값에 따라 그룹화하여 그룹별로 결과를 출력하는 함수이다.

② 그룹 함수의 유형

① ROLLUP 함수

④ ROLLUP 함수 개념

- ROLLUP에 의해 지정된 컬럼은 소계(소그룹의 합계) 등 중간 집계 값을 산출하기 위한 그룹 함수이다.
- 지정 컬럼의 수보다 하나 더 큰 레벨만큼의 중간 집계 값이 생성된다.
- ROLLUP의 지정 컬럼은 계층별로 구성되기 때문에 순서가 바뀌면 수행 결과가 바뀜을 유의한다.

▼ ROLLUP 함수 구문

```
SELECT 컬럼1, 컬럼2, ..., 집계 함수
  FROM 테이블명
  [WHERE ...]
  GROUP BY [컬럼 ...] ROLLUP 컬럼
  [HAVING ...]
  [ORDER BY ...]
```

- 소계 집계 대상이 되는 컬럼을 ROLLUP 뒤에 기재하고, 소계 집계 대상이 아닌 경우 GROUP BY 뒤에 기재한다.
- SELECT 뒤에 포함되는 컬럼이 GROUP BY 또는 ROLLUP 뒤에 기재되어야 한다는 점만 숙지하고 쿼리를 작성한다.
- ORDER BY 구문을 활용해 계층 내 정렬에 사용이 가능하며, SQL의 결과를 보다 체계적으로 보여 준다.

④ ROLLUP 함수 사례

부서별 연봉 테이블인 DEPT_SALARY에서 부서명-직위에 해당되는 연봉 정보, 부서별 연봉 합계, 전체 연봉 합계를 나타내는 쿼리이다.

학습 Point

그룹 함수 역시 중요도가 높습니다. ROLLUP, CUBE, GROUPING SETS 함수의 개념과 문법을 확인해가며 익혀두시길 권장합니다.

56 ()은/는 프로그램에서 프로그램이 끝없이 동작하는 것으로, 반복문에 종료 조건이 없거나, 종료 조건과 만날 수 없을 때 생긴다.

57 ()은/는 서로 연관된 정수형 상수들의 집합으로, 정수형 상수에 이름을 붙여서 코드를 이해하기 쉽게 해준다.

58 ()은/는 사용자가 기본 태입을 가지고 새롭게 정의할 수 있는 사용자 정의 자료형이다.

59 C언어에서 열거체를 선언하기 위한 키워드는 ()이다.

60 C언어에서 구조체를 선언하기 위한 키워드는 ()이다.

정답 56. 무한 루프 57. 열거체 58. 구조체 59. enum 60. struct

▼ 부서별 연봉 테이블

부서명(DEPT)	직위(JOB)	연봉(SALARY)
마케팅부	부장	4,000
마케팅부	차장	2,000
마케팅부	과장	1,500
기획부	부장	3,800
기획부	차장	1,800

▼ ROLLUP 함수 사례

```
SELECT DEPT, JOB, SUM(SALARY)
  FROM DEPT_SALARY
 GROUP BY ROLLUP(DEPT, JOB);
```

▼ ROLLUP 결괏값

부서명(DEPT)	직위(JOB)	연봉(SALARY)
마케팅부	부장	4,000
마케팅부	차장	2,000
마케팅부	과장	1,500
마케팅부		7,500
기획부	부장	3,800
기획부	차장	1,800
기획부		5,600
		13,100

② CUBE 함수

① CUBE 함수 개념

- CUBE는 결합 가능한 모든 값에 대해 다차원 집계를 생성하는 그룹 함수이다.
- 연산이 많아 시스템에 부담을 준다.

▼ CUBE 활용 구문

구문	설명
<pre>SELECT 컬럼명1, ..., 집계 함수 FROM 테이블명 [WHERE ...] GROUP BY [컬럼명1, ...] CUBE(컬럼명a, ...) [HAVING ...] [ORDER BY ...]</pre>	<ul style="list-style-type: none"> ROLLUP은 소계, 중간 집계를 나타내 주지만, CUBE는 결합 가능한 모든 값에 대해 다차원 집계 생성 CUBE는 세분화된 소계가 구해짐

④ CUBE 함수 사례

- 부서별 연봉 테이블인 DEPT_SALARY에서 부서명-직위에 해당되는 연봉 정보, 부서별 연봉 합계, 직위별 연봉 합계, 전체 연봉 합계를 나타내는 쿼리이다.

▼ 부서별 연봉 테이블

부서명(DEPT)	직위(JOB)	연봉(SALARY)
마케팅부	부장	4,000
마케팅부	차장	2,000
마케팅부	과장	1,500
기획부	부장	3,800
기획부	차장	1,800

▼ CUBE 함수 사례

```
SELECT DEPT, JOB, SUM(SALARY)
  FROM DEPT_SALARY
 GROUP BY CUBE(DEPT, JOB);
```

▼ CUBE 결괏값

부서명(DEPT)	직위(JOB)	연봉(SALARY)
		13,100
	부장	7,800
	차장	3,800
	과장	1,500
마케팅부		7,500
마케팅부	부장	4,000
마케팅부	차장	2,000
마케팅부	과장	1,500
기획부		5,600
기획부	부장	3,800
기획부	차장	1,800

③ GROUPING SETS 함수

② GROUPING SETS 함수 개념

- 집계 대상 컬럼들에 대한 개별 집계를 구할 수 있으며, ROLLUP이나 CUBE와는 달리 컬럼 간 순서와 무관한 결과를 얻을 수 있는 그룹 함수이다.

61 ()은/는 사용자가 직접 새로운 함수를 정의하여 사용하는 방법이다.

62 ()은/는 사용자 정의 함수가 종료될 때 반환 값을 넘겨주기 위해 사용하는 키워드이다.

63 ()은/는 함수 자신이 자신을 부르는 함수이다.

64 ()은/는 외부의 모든 클래스에서 접근이 가능한 접근 제어자이다.

65 ()은/는 같은 패키지 내부에 있는 클래스, 하위 클래스(상속받은 경우)에서 접근이 가능한 접근 제어자이다.

정답 61. 사용자 정의 함수 62. return 63. 재귀함수 64. public 65. protected

- GROUPING SETS를 이용해 다양한 소계 집합을 만들 수 있다.
- ORDER BY를 사용하여 집계 대상 그룹과의 표시 순서를 조정하여 체계적으로 보여줄 수 있다.

▼ GROUPING SETS 활용 구문

구문	설명
<pre>SELECT 컬럼명1, ..., 집계 함수 FROM 테이블명 [WHERE ...] GROUP BY [컬럼명1, ...] GROUPING SETS(컬럼명1, ...) [HAVING ...] [ORDER BY ...]</pre>	개별 집계를 구할 수 있으며, ROLLUP 계층구조와 달리 평등한 관계라 순서에 상관없이 동일한 결과

④ GROUPING SETS 함수 사례

부서별 연봉 테이블인 DEPT_SALARY에서 부서별 연봉 합계, 직위별 연봉 합계를 나타내는 쿼리이다.

▼ 부서별 연봉 테이블

부서명(DEPT)	직위(JOB)	연봉(SALARY)
마케팅부	부장	4,000
마케팅부	차장	2,000
마케팅부	과장	1,500
기획부	부장	3,800
기획부	차장	1,800

▼ GROUPING SETS 함수 사례

```
SELECT DEPT, JOB, SUM(SALARY)
  FROM DEPT_SALARY
 GROUP BY GROUPING SETS(DEPT, JOB);
```

▼ GROUPING SETS 결괏값

부서명(DEPT)	직위(JOB)	연봉(SALARY)
		13,100
	부장	7,800
	차장	3,800
	과장	1,500
마케팅부		7,500
기획부		5,600

(5) 윈도 함수

1 윈도 함수의 개념

- 윈도 함수는 데이터베이스를 사용한 온라인 분석 처리 용도로 사용하기 위해서 표준 SQL에 추가된 함수이다.
- 윈도 함수를 OLAP 함수라고도 한다.

2 윈도 함수의 구문

▼ 윈도 함수의 구문

구문	설명
<pre>SELECT 함수명(파라미터) OVER ([PARTITION BY 컬럼1, ...] [ORDER BY 컬럼A, ...]) FROM 테이블명</pre>	<ul style="list-style-type: none"> PARTITION BY는 선택 항목이며, 순위를 정할 대상 범위의 컬럼을 설정 PARTITION BY 구에는 GROUP BY 구가 가진 집약 기능이 없으며, 이로 인해 레코드가 줄어들지 않음 PARTITION BY를 통해 구분된 레코드 집합을 윈도라고 함 윈도 함수에는 OVER 문구가 필수적으로 포함 ORDER BY 뒤에는 SORT 컬럼을 입력(어떤 열을 어떤 순서로 순위를 정할지를 지정)

3 윈도 함수의 분류

- 윈도 함수에는 순위 함수, 행 순서 함수, 그룹 내 비율 함수가 있다.

▼ 윈도 함수의 분류

분류	설명
순위 함수	<ul style="list-style-type: none"> 레코드의 순위를 계산하는 함수 RANK, DENSE_RANK, ROW_NUMBER 함수가 존재
행 순서 함수	<ul style="list-style-type: none"> 레코드에서 가장 먼저 나오거나 가장 뒤에 나오는 값, 이전/이후의 값들을 출력하는 함수 FIRST_VALUE, LAST_VALUE, LAG, LEAD 함수가 존재
그룹 내 비율 함수	<ul style="list-style-type: none"> 백분율을 보여주거나 행의 순서별 백분율 등 비율과 관련된 통계를 보여주는 함수 RATIO_TO_REPORT, PERCENT_RANK 함수가 존재

잠깐! 알고가기

OLAP(OnLine Analytical Processing)
의사결정 지원 시스템으로, 사용자가 동일한 데이터를 여러 기준을 이용하는 다양한 방식으로 바라보면서 다차원 데이터 분석을 할 수 있도록 도와주는 기술이다.

학습 Point

윈도 함수는 집계 함수, 그룹 함수보다는 중요도가 낮습니다. 두 음쌍의 도움을 받아 빠르게 학습하세요!



두음쌍 한마디

원도 함수의 분류
「순행비」
순위 함수 / 행 순서 함수 / 그룹 내 비율 함수
→ 집 근처를 순행 비행하다

핵인싸 퀴즈

66 ()은/는 같은 클래스 내에서만 접근이 가능한 접근 제어자로 하위 클래스에서도 접근이 불가능한 접근 제어자이다.

67 ()은/는 C++, 자바에서 현재 객체를 가리키는 포인터로 클래스 내부의 변수와 함수를 가리킬 수 있다.

68 ()은/는 파이썬에서 현재 객체를 가리키는 포인터로 클래스 내부의 변수와 함수를 가리킬 수 있다.

69 C++에서 new에 의해 생성된 인스턴스는 메모리 누수(Leak)가 발생하지 않도록 () 키워드를 통해 메모리 해제를 해주어야 한다.

70 ()은/는 해당 클래스의 객체가 생성될 때 자동으로 호출되는 특수한 종류의 메서드이다.

정답 66. private 67. this 68. delete 69. 생성자

① 순위 함수

- 순위 함수에는 RANK, DENSE_RANK, ROW_NUMBER 함수가 있다.

▼ 순위 함수

순위 함수	설명
RANK	<ul style="list-style-type: none"> 특정 항목(컬럼)에 대한 순위를 구하는 함수 동일 순위의 레코드 존재 시 후순위는 넘어감(2위가 3개인 레코드인 경우: 2위, 2위, 2위, 5위, 6위, ...)
DENSE_RANK	<ul style="list-style-type: none"> 레코드의 순위를 계산 동일 순위의 레코드 존재 시에도 후순위를 넘어가지 않음(2위가 3개인 레코드인 경우: 2위, 2위, 2위, 3위, 4위, ...)
ROW_NUMBER	<ul style="list-style-type: none"> 레코드의 순위를 계산 동일 순위의 값이 존재해도 이와 무관하게 연속 번호를 부여(2위가 3개인 레코드인 경우: 2위, 3위, 4위, 5위, 6위, ...)

- 다음은 직원 테이블인 EMPLOYEE에서 연봉 컬럼인 SALARY를 기준으로 RANK 함수, DENSE_RANK, ROW_NUMBER 함수를 사용한 쿼리와 결과이다.

▼ 순위 함수 예제

```
SELECT NAME,
       SALARY,
       RANK() OVER (ORDER BY SALARY DESC) A,
       DENSE_RANK() OVER (ORDER BY SALARY DESC) B,
       ROW_NUMBER() OVER (ORDER BY SALARY DESC) C
  FROM EMPLOYEE;
```

▼ 순위 함수 결과

이름(NAME)	연봉(SALARY)	A	B	C
장길산	3000	1	1	1
임꺽정	2500	2	2	2
홍길동	2500	2	2	3
김철수	2200	4	3	4
한유리	2000	5	4	5

② 행 순서 함수

- 행 순서 함수는 FIRST_VALUE, LAST_VALUE, LAG, LEAD 함수가 있다.

▼ 행 순서 함수

행 순서 함수	설명
FIRST_VALUE	<ul style="list-style-type: none"> 파티션별 윈도에서 가장 먼저 나오는 값을 찾음 집계 함수의 MIN과 동일한 결과를 출력
LAST_VALUE	<ul style="list-style-type: none"> 파티션별 윈도에서 가장 늦게 나오는 값을 찾음 집계 함수의 MAX와 동일한 결과 출력
LAG	<ul style="list-style-type: none"> 파티션별 윈도에서 이전 로우의 값 반환
LEAD	<ul style="list-style-type: none"> 파티션별 윈도에서 이후 로우의 값 반환

- 다음은 직원 테이블인 EMPLOYEE에서 연봉 컬럼인 SALARY를 기준으로 FIRST_VALUE, LAST_VALUE, LAG, LEAD 함수를 사용한 쿼리와 결과이다.

▼ 행 순서 함수 예제

```
SELECT NAME,
       SALARY,
       FIRST_VALUE(NAME) OVER (ORDER BY SALARY DESC) A,
       LAST_VALUE(NAME) OVER (ORDER BY SALARY DESC) B,
       LAG(NAME) OVER (ORDER BY SALARY DESC) C,
       LEAD(NAME) OVER (ORDER BY SALARY DESC) D
  FROM EMPLOYEE;
```

▼ 행 순서 함수 예제

이름(NAME)	연봉(SALARY)	A	B	C	D
장길산	3000	장길산	한유리		임꺽정
임꺽정	2500	장길산	한유리	장길산	홍길동
홍길동	2500	장길산	한유리	임꺽정	김철수
김철수	2200	장길산	한유리	홍길동	한유리
한유리	2000	장길산	한유리	김철수	

③ 그룹 내 비율 함수

- 그룹 내 비율 함수는 RATIO_TO_REPORT, PERCENT_RANK 함수가 있다.

▼ 그룹 내 비율 함수

그룹 내 비율 함수	설명
RATIO_TO_REPORT	<ul style="list-style-type: none"> 주어진 그룹에 대해 합을 기준으로 각 로우의 상대적 비율을 반환하는 함수 결값값은 0~1의 범위 값을 가짐 OVER 팔호 안에 컬럼 생략 시 그룹은 테이블 전체가 대상
PERCENT_RANK	<ul style="list-style-type: none"> 주어진 그룹에 대해 제일 먼저 나오는 것을 0으로, 제일 늦게 나오는 것을 1로 하여, 값이 아닌 행의 순서별 백분율을 구하는 함수 결값값은 0~1의 범위 값을 가짐

핵인싸 퀴즈

71 파이썬에서 생성자임을 알리기 위해 ()이라는 메서드명을 사용한다.

72 ()은/는 객체의 수명이 끝났을 때 객체를 제거하기 위한 목적으로 사용되는 메서드이다.

73 C++에서는 소멸자를 호출하기 위해서 클래스명과 동일한 메서드명을 가지고 ()기호를 사용한다.

74 파이썬에서 소멸자는 () ()라는 메서드명을 사용한다.

75 ()은/는 어떤 객체가 있을 때 그 객체의 변수와 메서드를 다른 객체가 물려받는 기능이다.

정답 71. __init__ 72. 소멸자 73. ~ 74. __del__ 75. 상속

NCS
지/피/지/기

기출문제

- 다음은 직원 테이블인 EMPLOYEE에서 연봉 컬럼인 SALARY를 기준으로 RATIO_TO_REPORT, PERCENT_RANK 함수를 사용한 쿼리와 결과이다.

▼ 그룹 내 비율 함수 예제

```
SELECT NAME,
       SALARY,
       RATIO_TO_REPORT(SALARY) OVER () A,
       PERCENT_RANK() OVER (ORDER BY SALARY DESC) B
  FROM EMPLOYEE;
```

▼ 그룹 내 비율 함수 결과

이름(NAME)	연봉(SALARY)	A	B
장길산	3000	0.2459	0
임꺽정	2500	0.2049	0.25
홍길동	2500	0.2049	0.25
김철수	2200	0.1803	0.75
한유리	2000	0.1639	1

[예술형]

▶ 2020년 3회

- ✓ 01 다음 조건을 만족하면서 과목별 점수의 평균이 90 이 상인 과목이름, 최소점수, 최대점수를 구하는 SQL 문을 작성하시오.

- 대소문자를 구분하지 않습니다.
- WHERE 구문을 사용하지 않습니다.
- GROUP BY, HAVING 구문을 반드시 사용합니다.
- 세미콜론(;)은 생략 가능합니다.
- 별칭(AS)를 사용해야 합니다.(별칭 사용 시 별칭은 작은 따옴표를 써야 합니다.)
- 집계 함수를 사용해야 합니다.

[성적]

과목코드	과목이름	학점	점수
1000	컴퓨터과학	A+	95
2000	운영체제	B+	85
1000	컴퓨터과학	B+	85
2000	운영체제	B	80

[결과]

과목이름	최소점수	최대점수
컴퓨터과학	85	95

해설 ▶ 과목별 점수의 평균을 구하기 위해서는 GROUP BY 절과 HAVING 절을 이용한다. 최소점수를 구하기 위해서는 MIN 함수. 최대점수를 구하기 위해서는 MAX 함수를 이용한다.

정답

01. SELECT 과목이름, MIN(점수) AS 최소점수, MAX(점수) AS 최대점수 FROM 성적 GROUP BY 과목이름 HAVING AVG(점수) >= 90;
02. SELECT 학과, COUNT(학과) AS '학과별tuple수' FROM 학생 GROUP BY 학과; 또는 SELECT 학과, COUNT(*) AS '학과별tuple수' FROM 학생 GROUP BY 학과;

[예술형]

▶ 2020년 4회

- 02 다음 조건을 만족하면서 학과별로 튜플 수가 얼마인지 구하는 SQL 문을 작성하시오.

- 대소문자를 구분하지 않습니다.
- WHERE 구문을 사용하지 않습니다.
- GROUP BY를 사용합니다.
- 세미콜론(;)은 생략 가능합니다.
- 별칭(AS)를 사용해야 합니다.(별칭 사용 시 별칭은 작은 따옴표를 써야 합니다.)
- 집계 함수를 사용해야 합니다.

[학생]

학과	학생
전기	이순신
컴퓨터	안중근
컴퓨터	윤봉길
전자	이봉창
전자	강우규

[결과]

학과	학과별tuple수
전기	1
컴퓨터	2
전자	2

해설 ▶ 학과에 대한 집계를 하기 위해서는 GROUP BY 절에 학과를 지정한다.
 • 학과별 튜플의 개수를 구하기 위해서는 COUNT 함수를 사용하며, COUNT 함수에는 행의 개수를 판단할 수 있는 속성을 파라미터로 넣는다.
 • COUNT(학과), COUNT(*) 모두 행의 개수를 판단할 수 있으므로 둘 중 아무거나 사용해도 무방하다.



NCS

천/기/누/설

예상문제

- 01 [기술형] 다음은 급여 테이블이다. 부서명과 부서별 급여의 평균을 구하는 쿼리를 구하시오.

[급여]

이름	직책	부서	급여
문무왕	차장	마케팅	5,000
박혁거세	차장	전산	4,800
장보고	사원	마케팅	4,000
김유신	사원	마케팅	4,000

해설》 · GROUP BY 절은 속성값을 그룹으로 분류하고자 할 때 사용한다.

- 부서별로 구분해야 하므로 GROUP BY 절에는 '부서' 컬럼이 들어가야 하며, SELECT 절에서는 '부서'별 평균이 어떤 값인지 알아야 하므로 '부서'와 AVG(급여)가 포함되어야 한다.

- 02 [기술형] 다음은 급여 테이블이다. 부서와 직책별 급여의 합계를 구하는 쿼리를 구하시오.

[급여]

이름	직책	부서	급여
문무왕	차장	마케팅	5,000
박혁거세	차장	전산	4,800
장보고	사원	마케팅	4,000
김유신	사원	마케팅	4,000

해설》 · GROUP BY 절은 속성값을 그룹으로 분류하고자 할 때 사용한다.

- 부서별로 구분해야 하므로 GROUP BY 절에는 '부서', '직책' 컬럼이 들어가야 하며, SELECT 절에서는 '부서'와 '직책'별 평균이 어떤 값인지 알아야 하므로 '부서', '직책' 컬럼과 SUM(급여)가 포함되어야 한다.

- 03 [기술형] 다음은 학생 테이블의 일부이다. 평균 성적이 4.0을 초과하는 학생의 이름을 출력하는 쿼리를 작성하시오.

[학생]

이름	과목	성적
문무왕	프로그래밍	4.5
문무왕	알고리즘	4.5
장보고	알고리즘	3.5
장보고	자료구조	4.5

해설》 · GROUP BY 절은 속성값을 그룹으로 분류하고자 할 때 사용한다.

- 이름에 따라 평균 성적을 구하려는 쿼리를 구하기 위해서는 GROUP BY 절에 '이름' 컬럼을 추가해야 한다.
- HAVING 절은 GROUP BY에 의해 분류한 후 그룹에 대한 조건을 지정할 때 사용한다.
- 평균 성적이 4.0을 초과하는지 확인하기 위해 HAVING 절에 평균 성적이 4.0보다 크다라는 조건의 쿼리를 작성한다.
- 이름만 출력해야 하므로 SELECT 절에는 '이름' 컬럼만 조회한다.

[결과]

학교명	학년	인원
A	1	51
A	2	29
A	3	15
A		95
B	1	33
B	2	44
B	3	72
B		149
		244

[결과]

이름	점수	등수
장길산	100	1
임꺽정	100	1
홍길동	90	2
김철수	80	3
한유리	70	4

해설》 · 동일 순위의 레코드 존재 시에도 후순위를 넘어가지 않는 경우(2위가 3개인 레코드인 경우: 2위, 2위, 2위, 3위, 4위, ...) DENSE_RANK 함수를 사용한다.

- DENSE_RANK는 윈도 함수이고, 윈도 함수 문법은 다음과 같다.

```
SELECT 함수명(파라미터)
    OVER ([PARTITION BY 컬럼1, ...])
    [ORDER BY 컬럼A, ...]
    FROM 테이블명;
```

- ORDER BY 뒤에는 SORT할 컬럼을 입력(어떤 열을 어떤 순서대로 순위를 매길지를 지정)한다.

- 05 [기술형] 다음은 학생 테이블이다. 윈도 함수를 이용하여 결과 테이블과 동일하게 나오도록 쿼리를 작성하시오.

[학생]

이름	점수
장길산	100
임꺽정	100
홍길동	90
김철수	80
한유리	70

정답

- SELECT 부서, 직책, AVG(급여) FROM 급여 GROUP BY 부서;
- SELECT 부서, 직책, SUM(급여) AS 급여 FROM 급여 GROUP BY 부서, 직책;
- SELECT 이름, 과목, SUM(급여) AS 급여 FROM 학생 GROUP BY 이름 HAVING AVG(성적) > 4.0;
- SELECT 학교명, 학년, SUM(인원) AS 인원 FROM 학교 GROUP BY 학교명, 학년 ROLLUP;
- SELECT 이름, 점수, DENSE_RANK() OVER(ORDER BY 점수 DESC) AS 등수 FROM 학생;



단원종합문제

[기술형]
01 커밋(COMMIT)에 대해 서술하시오.

해설 >	롤백 (ROLLBACK)	오류가 발생했을 때, 오류 이전의 특정 시점(SAVEPOINT, CHECKPOINT) 상태로 되돌려주는 제어어
	커밋 (COMMIT)	트랜잭션 메모리에 영구적으로 저장하는 제어어
	체크포인트 (CHECKPOINT)	ROLLBACK을 위한 시점을 지정하는 제어어

[단답형]
02 STUDENT 테이블에 컴퓨터과 학생 50명, 인터넷과 학생 100명, 사무자동화학과 학생 50명의 정보가 저장되어 있을 때, 다음 쿼리 문을 실행한 결과 튜플 수는? (단, DEPT 컬럼은 학과명이다.)

- ① SELECT DEPT FROM STUDENT;
- ② SELECT DISTINCT DEPT FROM STUDENT;
- ③ SELECT COUNT(DISTINCT DEPT) FROM STUDENT WHERE DEPT='컴퓨터과';

- ①
- ②
- ③

해설 > • 단순 SELECT(조건검색)의 경우 전체 테이블의 튜플을 검색하기 때문에 200건($=50+100+50$)이 조회된다.
• DISTINCT(중복제거)의 경우 동일한 튜플을 제거하고 검색하기 때문에 '컴퓨터과' 1건, '인터넷과' 1건, '사무자동화학과' 1건으로 총 3건이 조회된다.
• GROUP BY가 없을 경우 전체 테이블에서 튜플 개수를 COUNT를 통해 출력하기 때문에 출력값을 표현한 1개가 튜플이다.

[기술형]
03 사원 테이블에 생년월일이라는 컬럼을 삭제하고자 한다. 알맞은 쿼리를 작성하시오.

해설 > ALTER TABLE에서 컬럼을 삭제하는 문법은 다음과 같다.
ALTER TABLE 테이블명 DROP 컬럼명;

[기술형]
04 다음은 사람이라는 테이블을 만들려고 한다. 사람이라는 테이블에는 이름, 생년월일이라는 컬럼이 있는데, 이름은 VARCHAR(10) 데이터 타입을 가지고, 생년월일은 CHAR(8) 데이터 타입을 가지고 있다. 생년월일은 추가로 NULL을 가질 수 없도록 제약 조건을 걸고자 한다. 알맞은 쿼리를 작성하시오.

[기술형]
05 아래와 같이 학생 테이블이 존재할 때, 뷰를 생성하는 SQL을 작성하시오.

- 학생 테이블에서 전공이 '전산'인 학번, 이름으로 학생 뷰라는 이름의 뷰를 생성
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[학생] 테이블

학번	이름	전공
202001	홍길동	전산
202002	김정미	전산
202003	장길산	전산
202004	강은미	경영

[학생부] 뷰

학번	이름
202001	홍길동
202002	김정미
202003	장길산

[기술형]
06 아래와 같이 테이블이 존재할 때, 인덱스를 생성하는 SQL 문을 작성하시오

- 학생 테이블의 학번 컬럼에 대해 학번인덱스라는 인덱스명으로 인덱스를 생성
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[학생] 테이블

학번	이름	학과
2020001	홍길동	전산
2020002	김정미	전산
2020003	장길산	보안
2020004	강은미	보안

해설 > • 인덱스를 생성하는 명령은 다음과 같다.

CREATE INDEX 인덱스명 ON 테이블명(컬럼명1, 컬럼명2, ...);

• 인덱스명은 학번인덱스이고, 테이블명은 학생. 컬럼은 학번으로 해서 인덱스를 생성한다.

해설 > CREATE TABLE의 상세 문법은 다음과 같다.

```
CREATE TABLE 테이블명
(
    컬럼명 데이터타입 PRIMARY KEY, -- 기본키 설정
    컬럼명 데이터타입 FOREIGN KEY REFERENCES 참조테이블(기본키), -- 외래키 설정
    컬럼명 데이터타입 UNIQUE,
    컬럼명 데이터타입 NOT NULL,
    컬럼명 데이터타입 CHECK(조건식) -- 제약조건 설정
    컬럼명 데이터타입 DEFAULT 값
);
```

해설 > • 뷰를 생성하는 명령은 다음과 같다.

CREATE VIEW 뷰이름 AS
조회쿼리

- 학생 테이블에서 전공이 '전산'인 학번, 이름으로 학생뷰라는 이름의 뷰를 생성하므로
- CREATE VIEW 학생부 AS 뒤에
- SELECT 절에는 학번, 이름, FROM 절에는 사원, WHERE 절은 전공이 '전산'으로 한다.



단원종합문제

[기술형]
07 아래와 같이 조건을 만족하는 테이블을 생성하는 SQL 문을 작성하시오.

- 테이블명은 교수
- VARCHAR 타입의 사이즈가 10인 컬럼 이고 PRIMARY KEY인 컬럼
- VARCHAR 타입의 사이즈가 10인 이름 컬럼
- VARCHAR 타입의 사이즈가 8이면서 Null을 허용하지 않는 임용일 컬럼
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

해설 ▶ 테이블을 생성하는 명령은 다음과 같다.

```
CREATE TABLE 테이블명
(
    컬럼명 데이터타입 [제약조건],
    ...
);
```

[기술형]
08 성적 테이블에 학점 컬럼을 추가하는 SQL 문을 작성하시오.

- 학점 컬럼은 VARCHAR 타입이고 사이즈는 4이고 NULL을 허용하지 않음
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[성적] 테이블

과목코드	과목이름	점수
1000	컴퓨터과학	95
2000	운영체제	85
1000	컴퓨터과학	85
2000	운영체제	80

[결과]

과목코드	과목이름	점수	학점
1000	컴퓨터과학	95	A+
2000	운영체제	85	B+
1000	컴퓨터과학	85	B+
2000	운영체제	80	B

해설 ▶ TABLE에 컬럼을 추가하는 문법은 다음과 같다.

```
ALTER TABLE 테이블명 ADD 컬럼명 데이터타입 [제약조건];
    • 테이블명은 성적, 컬럼명은 학점, 데이터타입은 VARCHAR, 사
    이즈는 4, NULL을 허용하지 않으므로 NOT NULL로 작성한다.
```

[기술형]
09 학생 테이블에서 전산과를 출력하는 SQL 문을 작성하시오.

- WHERE 절 조건을 이용하여 학과가 전산과인 행을 출력
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[학생] 테이블

학번	이름	학과
20001	김철수	전산과
20002	한유리	전산과
20003	장길산	보안학과
20004	홍길동	보안학과

[결과]

학번	이름
20001	김철수
20002	한유리

해설 ▶ WHERE 절에서 학과가 전산과인 행을 출력한다.

[기술형]

10 급여 테이블에서 부서별의 급여합계를 출력하는 SQL 문을 작성하시오.

- 급여합계는 급여 컬럼의 값들의 합이며 AS를 사용하여 '급여합계'로 출력
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[급여] 테이블

이름	부서	급여
김철수	마케팅	5000
장길산	마케팅	4000
홍길동	전산	3000
한유리	전산	2000

[결과]

부서	급여합계
마케팅	9000
전산	5000

[기술형]

11 급여 테이블에서 부서별의 급여합계가 6,000을 출력하는 SQL 문을 작성하시오.

- 급여합계는 급여 컬럼의 값들의 합이며 AS를 사용하여 '급여합계'로 출력
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[급여] 테이블

이름	부서	급여
김철수	마케팅	5000
장길산	마케팅	4000
홍길동	전산	3000
한유리	전산	2000

[결과]

부서	급여합계
마케팅	9000

해설 ▶ 부서별 급여합계를 구하기 위해서는 GROUP BY를 사용 한다.

- SELECT 절에서 컬럼명을 급여합계로 출력하기 위해 SUM(급여) 뒤에 AS를 사용한다.



단원종합문제

[기술형] **12** 주어진 급여테이블에서 다음을 만족하는 SQL을 작성하시오.

- SELECT 절에 '*'을 사용하지 않음
- 학점은 내림차순 정렬, 같은 값을 때 이름을 오름차순으로 정렬함
- 대소문자를 구분하지 않음
- 세미콜론(;)은 생략 가능

[급여] 테이블

이름	과목	학점
김철수	C언어	A
한유리	자료구조	A
신짱구	자료구조	A
이훈이	알고리즘	B

[결과]

이름	과목	학점
이훈이	알고리즘	B
김철수	C언어	A
신짱구	자료구조	A
한유리	자료구조	A

해설 • ORDER BY 절은 속성값을 정렬하고자 할 때 사용한다.
(ASC: 오름차순, DESC: 내림차순, ASC, DESC 키워드 생략 시
오름차순 정렬)

ORDER BY 속성 [ASC | DESC]

- 학점은 내림차순, 그 다음에 학점이 같을 경우 이름이 오름차순으로 ORDER BY 절에 학점, 이름 순서로 쿼리를 작성한다.

[기술형] **13** 주어진 도서 테이블과 도서가격 테이블을 내부 조인(inner join)하는 SQL 문을 작성하시오.

- FROM 절에서 도서는 A, 도서가격은 B로 별칭을 줌
- 도서 테이블의 책번호와 도서가격 테이블의 책번호는 ON 절에서 조인 조건으로 사용
- 대소문자를 구분하지 않음
- 세미콜론(;)은 생략 가능

[도서] 테이블

책번호	책명
111	운영체제
222	자료구조
555	컴퓨터구조

[도서가격] 테이블

책번호	가격
111	20000
222	25000
333	10000
444	15000

[결과]

책번호	책명	가격
111	운영체제	20000
222	자료구조	25000

해설 • 내부 조인 문법은 다음과 같다.

```
SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2,
...
FROM 테이블1 A [INNER] JOIN 테이블2 B
ON 조인조건
[WHERE 검색조건]
```

[기술형] **14** 주어진 도서 테이블과 도서가격 테이블을 완전 외부 조인(full outer join)하는 SQL 문을 작성하시오.

- FROM 절에서 도서는 A, 도서가격은 B로 별칭을 줌
- 도서 테이블의 책번호와 도서가격 테이블의 책번호는 ON 절에서 조인 조건으로 사용
- 대소문자를 구분하지 않음
- 세미콜론(;)은 생략 가능

[도서] 테이블

책번호	책명
111	운영체제
222	자료구조
555	컴퓨터구조

[도서가격] 테이블

책번호	책명
111	20000
222	25000
333	10000
444	15000

[결과]

A.책번호	A.책명	B.책번호	B.가격
111	운영체제	111	20,000
222	자료구조	222	25,000
NULL	NULL	333	10,000
NULL	NULL	444	15,000
555	컴퓨터구조	NULL	NULL

해설 • 완전 외부 조인(full outer join) 문법은 아래와 같다.

```
SELECT A.컬럼1, A.컬럼2, ..., B.컬럼1, B.컬럼2,
...
FROM 테이블1 A FULL [OUTER] JOIN 테이블2 B
ON 조인조건
[WHERE 검색조건]
```

- 완전 외부 조인을 사용하기 위해 FULL OUTER JOIN으로 하고, ON 조건에서 JOIN 조건을 작성한다.

[기술형] **15** 도서 테이블과 도서가격 테이블에서 아래를 만족하는 SQL 문을 작성하시오.

- FROM 절에서 도서는 A, 도서가격은 B로 별칭을 줌
- WHERE 절에서 IN 연산자를 사용하여 IN 연산자의 조건에 서브쿼리를 사용
- 책명이 자료구조인 가격 중에 가장 비싼 값을 도서가격 테이블에서 서브쿼리함
- 대소문자를 구분하지 않음
- 세미콜론(;)은 생략 가능

[도서] 테이블

책번호	책명
111	운영체제
222	자료구조
555	컴퓨터구조

[도서가격] 테이블

책번호	책명	가격
111	운영체제	20000
222	자료구조	25000
333	10000	10,000
444	15000	15,000

[결과]

가격
25000



단원종합문제

[의술형]
16 서브쿼리하는 SQL 문을 작성하시오.

- FROM 절에서 도서는 A, 도서가격은 B로 별칭을 줌
- SELECT 절에서 서브쿼리를 사용
- 책명이 자료구조인 가격 중에 가장 비싼 값을 도서가격 테이블에서 서브쿼리함
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[도서] 테이블

책번호	책명
111	운영체제
222	자료구조
555	컴퓨터구조

[도서가격] 테이블

책번호	책명
111	20000
222	25000
333	10000
444	15000

[결과]

가격
25000

해설> 도서 테이블에서 자료구조인 책 중에서 가장 비싼 가격을 구하기 위해 MAX 를 사용하여 특정 값을 구한 후 도서가격 테이블을 서브쿼리한다.

[의술형]
17 데이터의 내용을 삽입하는 SQL 문을 작성하시오.

- INSERT 문을 이용
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[도서] 테이블

학번	성명	학년	수강과목
1000	홍길동	1	수학
2000	이순신	2	알고리즘

[결과]

학번	성명	학년	수강과목
1000	홍길동	1	수학
2000	이순신	2	알고리즘
3000	장길산	3	수학

해설> INSERT 명령어는 다음과 같다.

```
INSERT INTO 테이블명(속성명1, ...) VALUES (데이터1, ...)
```

[의술형]
18 데이터의 내용을 변경하는 SQL 문을 작성하시오.

- UPDATE 문을 이용
- WHERE 절에서 학번이 1000인 데이터만 수정
- SET에서 수학을 영어로 변경
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[도서] 테이블

학번	성명	학년	수강과목
1000	홍길동	1	수학

[결과]

학번	성명	학년	수강과목
1000	홍길동	1	영어

해설> UPDATE 명령어는 다음과 같다.

```
UPDATE 테이블명 SET 속성명 = 데이터, ...
WHERE 조건;
```

[의술형]
19 데이터의 내용을 삭제하는 SQL 문을 작성하시오.

- DELETE 문을 이용
- WHERE 절에서 학번이 3000인 데이터 삭제
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[도서] 테이블

학번	성명	학년	수강과목
1000	홍길동	1	수학
2000	이순신	2	알고리즘
3000	장길산	3	수학

[결과]

학번	성명	학년	수강과목
1000	홍길동	1	수학

해설> DELETE 명령어는 다음과 같다.

```
DELETE FROM 테이블명
WHERE 조건;
```

- 도서 테이블에서 데이터를 삭제하기 위해 WHERE 절에서 학번이 3000인 값을 구한다.

[의술형]
20 데이터베이스 관리자가 권한을 부여하는 DCL 문을 작성하시오.

- 관리자가 사용자 홍길동에게 '사원'테이블에 대해 SELECT할 수 있는 권한을 부여
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

해설> 권한을 부여하기 위해 GRANT 명령어를 사용한다.

```
GRANT 권한 ON 테이블 TO 사용자;
```

[의술형]
21 데이터베이스 관리자가 권한을 회수하는 DCL 문을 작성하시오.

- 관리자가 사용자 장길산에게 '학생' 테이블에 대해 UPDATE할 수 있는 권한을 회수
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

해설> 권한을 부여하기 위해 GRANT 명령어를 사용한다.

```
REVOKE 권한 ON 테이블 FROM 사용자
```



단원종합문제

[의술형]
22 순위함수를 사용하는 SQL 문을 작성하시오.

- RANK 함수를 이용하여 연봉을 높은 순에서 낮은 순으로 정렬함
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[사원] 테이블

이름	연봉
장길산	3000
임꺽정	2500
홍길동	2200
김철수	2000

[결과]

이름	연봉	순위
장길산	3000	1
임꺽정	2500	2
홍길동	2200	3
김철수	2000	4

해설 • 순위를 구하기 위해 RANK를 이용할 수 있다. RANK는 특정항목(컬럼)에 대한 순위를 구하여 동일 순위의 레코드 존재 시 후 순위는 넘어간다.
• SELECT 절에서 연봉을 높은 순에서 낮은 순으로 정렬하고 RANK 함수를 이용한다.

[의술형]
23 행 순서 함수를 사용하는 SQL 문을 작성하시오.

- FIRST_VALUE 행 순서 함수를 이용함
- FIRST_VALUE 행 순서 함수에서 연봉을 높은 순에서 낮은 순으로 정렬함
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[사원] 테이블

이름	연봉
장길산	3000
임꺽정	2500
홍길동	2200
김철수	2000

[결과]

이름	연봉	A
장길산	3000	장길산
임꺽정	2500	장길산
홍길동	2200	장길산
김철수	2000	장길산

해설 • FIRST_VALUE 행 순서 함수는 파티션별 윈도에서 가장 먼저 나오는 값을 찾는다.
• 집계함수의 MIN과 동일한 결과를 출력한다. SELECT 절에서 연봉을 높은 순에서 낮은 순으로 정렬하고 FIRST_VALUE 행 순서 함수를 사용한다.

[의술형]
24 사원 테이블에서 그룹 내 비율 함수를 사용하여 결과를 출력하는 SQL 문을 작성하시오.

- RATIO_TO_REPORT 함수를 이용함
- RATIO_TO_REPORT 함수에서 연봉을 높은 순에서 낮은 순으로 정렬함
- 대소문자를 구분하지 않음
- 세미콜론(:)은 생략 가능

[사원] 테이블

이름	연봉
장길산	3000
임꺽정	2500
홍길동	2500
김철수	2200
한유리	2000

[결과]

이름	연봉	A
장길산	3000	0.2459
임꺽정	2500	0.2049
홍길동	2500	0.2049
김철수	2200	0.1803
한유리	2000	0.1639

해설 • RATIO_TO_REPORT 함수는 주어진 그룹에 대해 합을 기준으로 각 로우의 상대적 비율을 반환하는 함수이다.
• SELECT 절에서 연봉을 높은 순에서 낮은 순으로 정렬하고 RATIO_TO_REPORT 함수를 이용한다.

[의술형]
25 다음은 테이블 및 컬럼명에 대한 명세이다. 나이가 25세 이하인 직원들의 급여를 250으로 변경하는 쿼리를 작성하시오.

- 테이블명: EMPLOYEE(직원 테이블)
- 컬럼명: NAME(사원명), AGE(나이), SALARY(급여)

해설 UPDATE 명령문은 다음과 같다.

```
UPDATE 테이블명
SET 속성명=데이터, ...
WHERE 조건;
```

[의술형]
26 관리자가 사용자 홍길동에게 '학생' 테이블에 대해 조회 할 수 있는 권한을 회수하는 쿼리를 작성하시오.

해설 REVOKE 명령어
리온프 REVOKE 권한 ON 테이블 FROM 사용자

[의술형]
27 DBMS에서 지원하는 기능이며, 일련의 쿼리들을 마치 하나의 함수처럼 실행하기 위한 쿼리의 집합으로 반환 값을 출력하지 않는 기능은 무엇인지 쓰시오.

프로시저	일련의 쿼리들을 마치 하나의 함수처럼 실행하기 위한 쿼리의 집합
사용자 정의함수	일련의 SQL 처리를 수행하고, 수행 결과를 단일 값으로 반환할 수 있는 절차형 SQL
트리거	데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL



단원종합문제

[단답형] **28** 데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL 문은 무엇인지 쓰시오.

해설	프로시저 일련의 쿼리를 마치 하나의 함수처럼 실행하기 위한 쿼리의 집합
사용자 정의함수	일련의 SQL 처리를 수행하고, 수행 결과를 단일 값으로 반환할 수 있는 절차형 SQL
트리거	데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL

[단답형] **30** 다음 괄호 () 안에 알맞은 용어를 영어로 쓰시오.

통계 정보가 없는 상태에서 사전 등록된 규칙에 따라 질의실행 계획을 선택하는 옵티マイ저인 (①)와/과 통계정보로부터 모든 접근 경로를 고려한 질의실행 계획을 선택하는 옵티マイ저인 (②)이/가 있다.

①

②

해설	규칙기반 옵티マイ저(RBO) 통계 정보가 없는 상태에서 사전 등록된 규칙에 따라 질의 실행 계획을 선택하는 옵티マイ저
	비용기반 옵티マイ저(CBO) 통계 정보로부터 모든 접근 경로를 고려한 질의실행 계획을 선택하는 옵티マイ저

[약술형] **29** DB 쿼리에 사용하는 힌트에 대해서 서술하시오.

해설 SQL 성능 개선의 핵심 부분으로 옵티マイ저의 실행 계획을 원하는 대로 변경할 수 있게 한다.

정답

01. 트랜잭션을 메모리에 영구적으로 저장하는 제어어 02. ① 200, ② 3, ③ 1 03. ALTER TABLE 사원 DROP 생년월일;
04. CREATE TABLE 사람 (이름 VARCHAR(10), 생년월일 CHAR(8) NOT NULL);
05. CREATE VIEW 학생부 AS SELECT 학번, 이름 FROM 학생 WHERE 전공 = '전산';
06. CREATE INDEX 학번인덱스 ON 학생(학번);
07. CREATE TABLE 교수 (교번 VARCHAR(10) PRIMARY KEY, 이름 VARCHAR(10), 임용일 VARCHAR(8) NOT NULL);
08. ALTER TABLE 성적 ADD 학점 VARCHAR(4) NOT NULL
09. SELECT 학번, 이름 FROM 학생 WHERE 학과 = '전산과';
10. SELECT 부서, SUM(급여) AS 급여합계 FROM 급여 GROUP BY 부서;
11. SELECT 부서, SUM(급여) AS 급여합계 FROM 급여 GROUP BY 부서 HAVING 급여합계 >= 6000;
12. SELECT 이름, 과목, 학점 FROM 성적 ORDER BY 학점 DESC, 이름 ASC;
13. SELECT A.책번호, A.책명, B.가격 FROM 도서 A JOIN 도서가격 B ON A.책번호 = B.책번호;
14. SELECT A.책번호, A.책명, B.책번호, B.가격 FROM 도서 A FULL OUTER JOIN 도서가격 B ON A.책번호 = B.책번호;
15. SELECT MAX(가격) FROM 도서가격 WHERE 책번호 IN (SELECT 책번호 FROM 도서 WHERE 책명 = '자료구조');
16. SELECT (SELECT MAX(가격) FROM 도서 A WHERE A.책번호 = B.책번호 AND 책명 = '자료구조') AS 가격 FROM 도서가격 B;
17. INSERT INTO 학생 (학번, 성명, 학년, 수강과목) VALUES (3000, '장길산', 3, '수학');
18. UPDATE 도서 SET 수강과목 = '영어' WHERE 학번 = 1000;
19. DELETE FROM 도서 WHERE 학번 = 3000; 20. GRANT SELECT ON 사원 TO 홍길동;
21. REVOKE UPDATE ON 학생 FROM 장길산;
22. SELECT 이름, 연봉 RANK() OVER(ORDER BY 연봉 DESC) 순위 FROM 사원;
23. SELECT 이름, 연봉, FIRST_VALUE(이름) OVER (ORDER BY 연봉 DESC) A FROM 사원;
24. SELECT 이름, 연봉, RATIO_TO_REPORT(연봉) OVER (ORDER BY 연봉 DESC) A FROM 사원;
25. UPDATE EMPLOYEE SET SALARY = 250 WHERE AGE <= 25; 포기
26. REVOKE SELECT ON 학생 FROM 홍길동; 27. 사용자 정의 함수 28. 트리거 29. SQL 성능 개선의 핵심 부분으로 옵티マイ저의 실행 계획을 원하는 대로 변경할 수 있게 한다. 30. ① RBO(Rule Based Optimizer), ② CBO(Cost Based Optimizer)