

## 데이터베이스, 테이블 구조



## 데이터베이스, 테이블 다루기

쿼리	의미
CREATE DATABASE [데이터베이스 이름];	데이터베이스 만들기
SHOW DATABASES;	모든 데이터베이스 조회하기
USE [데이터베이스 이름];	데이터베이스 사용 선언하기
CREATE TABLE [테이블 이름] ( [컬럼 이름] [데이터 타입], [컬럼 이름] [데이터 타입], ... );	테이블 생성하기
ALTER TABLE [테이블 이름] RENAME [새로운 테이블 이름];	테이블 이름 변경하기
ALTER TABLE [테이블 이름] ADD COLUMN [컬럼 이름] [데이터 타입];	새로운 컬럼 추가하기
ALTER TABLE [테이블 이름] MODIFY COLUMN [컬럼 이름] [새로운 데이터 타입];	기존 컬럼 타입 변경하기
ALTER TABLE [테이블 이름] CHANGE COLUMN [컬럼 이름] [새로운 컬럼 이름] [새로운 데이터 타입];	기존 컬럼 이름과 타입 변경하기
ALTER TABLE [테이블 이름] DROP COLUMN [컬럼 이름];	컬럼 지우기
DROP DATABASE [데이터베이스 이름];	데이터베이스 지우기
TRUNCATE TABLE [테이블 이름];	테이블 값만 지우기
INSERT INTO [테이블 이름] ( [컬럼1 이름], [컬럼2 이름], [컬럼3 이름]) VALUES ((컬럼1 값, [컬럼2 값], [컬럼3 값]), ([컬럼1 값], [컬럼2 값], [컬럼3 값]), ([컬럼1 값], [컬럼2 값], [컬럼3 값]));	데이터 여러 개 삽입하기
DELETE FROM [테이블 이름] WHERE 조건식;	데이터 삭제하기
UPDATE [테이블 이름] SET [컬럼 이름] = [새 값] WHERE 조건식;	데이터 수정하기

## 6가지 쿼리 핵심 키워드

키워드	문법	의미
SELECT	SELECT [컬럼 이름]	가져올 데이터를 선택하는 키워드
FROM	FROM [테이블 이름]	데이터를 가져올 테이블을 지정하는 키워드
WHERE	WHERE 조건식	가져올 데이터의 조건을 지정해주는 키워드
GROUP BY	GROUP BY [컬럼 이름]	컬럼에서 동일한 값을 가지는 로우를 그룹화하는 키워드
HAVING	HAVING 조건식	가져올 데이터 그룹에 조건을 지정해주는 키워드
ORDER BY	ORDER BY [컬럼 이름]	가져올 데이터를 정렬해주는 키워드

## 6가지 쿼리 작성 순서, 실행 순서

<b>작성순서</b>	SELECT > FROM > WHERE > GROUP BY > HAVING > ORDER BY
<b>실행순서</b>	FROM > WHERE > GROUP BY > HAVING > SELECT > ORDER BY

## 쿼리 주요 키워드

키워드	문법	의미
AS	AS [컬럼 별명]	가져온 데이터에 별명을 지정하는 키워드
LIMIT	LIMIT [로우 수]	가져올 데이터의 로우 개수를 지정하는 키워드
DISTINCT	DISTINCT [컬럼 이름]	중복된 데이터는 제외하고 같은 값은 한 번만 가져오는 키워드

## 순위 함수

[합수 이름]() OVER (ORDER BY [컬럼 이름])	
합수	차이점
RANK	공동 순위가 있으면 다음 순서로 건너 뛴
DENSE_RANK	공동 순위가 있어도 다음 순위를 뛰어 넘지 않음
ROW_NUMBER	공동 순위를 무시함

## 그룹 함수

*Group이 아닌 컬럼 값 전체에도 사용 가능	
합수	의미
COUNT	그룹의 값 수를 세는 함수
SUM	그룹의 합을 계산하는 함수
AVG	그룹의 평균을 계산하는 함수
MIN	그룹의 최솟값을 반환하는 함수
MAX	그룹의 최댓값을 반환하는 함수

## 비교 연산자

연산자	활용	의미	예시
=	A = B	A와 B가 같다	1 = 1
!=	A != B	A와 B가 같지 않다	1 != 2
>	A > B	A가 B보다 크다	10 > 1
>=	A >= B	A가 B보다 크거나 작다	10 >= 10
<	A < B	A가 B보다 작다	10 < 100
<=	A <= B	A가 B보다 작거나 같다	10 <= 10

## 논리 연산자

연산자	활용	의미
AND	A AND B	A와 B 모두 True이면 True
OR	A OR B	A와 B 둘중 하나라도 True이면 True
NOT	NOT A	A가 아니면 True

## 기타 주요 연산자

연산자	활용	의미
BETWEEN	[컬럼 이름] BETWEEN A AND B	특정 범위 내의 데이터를 선택할 때 사용하는 연산자
IN	[컬럼 이름] IN (A, B, ..., C)	목록 내 포함되는 데이터를 선택할 때 사용하는 연산자
LIKE	[컬럼 이름] LIKE [검색할 문자열]	특정 문자열이 포함된 데이터를 선택하는 연산자
IS NULL	[컬럼 이름] IS NULL	데이터가 NULL인지 아닌지를 확인하는 연산자

## 자주 사용하는 문자형 데이터 함수

합수	활용 예시	설명
LOCATE	LOCATE("A", "ABC")	"ABC"에서 "A"는 몇 번째에 위치해 있는지 검색해 위치 반환
SUBSTRING	SUBSTRING("ABC", 2)	"ABC"에서 2번째 문자부터 반환
RIGHT	RIGHT("ABC", 1)	"ABC"에서 오른쪽에서 1번째 문자까지 반환
LEFT	LEFT("ABC", 1)	"ABC"에서 왼쪽에서 1번째 문자까지 반환
UPPER	UPPER("abc")	"abc"를 대문자로 바꿔 반환
LOWER	LOWER("ABC")	"ABC"를 소문자로 바꿔 반환
LENGTH	LENGTH("ABC")	"ABC"의 글자 수를 반환
CONCAT	CONCAT("ABC", "DEF")	"ABC" 문자열과 "DEF" 문자열을 합쳐 반환
REPLACE	REPLACE("ABC", "A", "Z")	"ABC"의 "A"를 "Z"로 바꿔 반환



## 자주 사용하는 숫자형 데이터 함수

함수	활용	설명
ABS	ABS(숫자)	숫자의 절댓값 반환
CEILING	CEILING(숫자)	숫자를 정수로 올림해서 반환
FLOOR	FLOOR(숫자)	숫자를 정수로 내림해서 반환
ROUND	ROUND(숫자, 자릿수)	숫자를 소수점 자릿수까지 반올림해서 반환
TRUNCATE	TRUNCATE(숫자, 자릿수)	숫자를 소수점 자릿수까지 버림 해서 반환
POWER	POWER(숫자A, 숫자B)	숫자A의 숫자B 제곱 반환
MOD	MOD(숫자A, 숫자B)	숫자A를 숫자B로 나눈 나머지 반환

## 자주 사용하는 날짜형 데이터 함수

함수	활용	설명
NOW	NOW()	현재 날짜와 시간 반환
CURRENT_DATE	CURRENT_DATE()	현재 날짜 반환
CURRENT_TIME	CURRENT_TIME()	현재 시간 반환
YEAR	YEAR(날짜)	날짜의 연도 반환
MONTH	MONTH(날짜)	날짜의 월 반환
MONTHNAME	MONTHNAME(날짜)	날짜의 월을 영어로 반환
DAYNAME	DAYNAME(날짜)	날짜의 요일을 영어로 반환
DAYOFMONTH	DAYOFMONTH(날짜)	날짜의 일 반환
DAYOFWEEK	DAYOFWEEK(날짜)	날짜의 요일을 숫자로 반환
WEEK	WEEK(날짜)	날짜가 해당 연도에 몇 번째 주인지 반환
HOURL	HOURL(시간)	시간의 시 반환
MINUTE	MINUTE(시간)	시간의 분 반환
SECOND	SECOND(시간)	시간의 초 반환
DATE_FORMAT	DATEFORMAT(날짜/시간, 형식)	날짜/시간의 형식을 형식으로 바꿔 반환
DATEDIFF	DATEDIFF(날짜1, 날짜2)	날짜1과 날짜2의 차이 반환 (날짜1 - 날짜2)
TIMEDIFF	TIMEDIFF(시간1, 시간2)	시간1과 시간2의 차이 반환 (시간1 - 시간2)

## 조건 만들기

조건 만들기	IF (조건식, 참일 때 값, 거짓일 때 값)	
여러 조건 한번에 만들기	<pre> CASE [컬럼 이름]   WHEN 조건값1 THEN 결과값1   WHEN 조건값2 THEN 결과값2   ELSE 결과값3 END </pre>	<pre> CASE   WHEN 조건식1 THEN 결과값1   WHEN 조건식2 THEN 결과값2   ELSE 결과값3 END </pre>

## 레이블 합치기

표현	의미	활용	설명
INNER JOIN		<pre> SELECT [컬럼 이름] FROM [테이블 A 이름] INNER JOIN [테이블 B 이름] ON [테이블 A 이름].[컬럼 A 이름] = [테이블 B 이름].[컬럼 B 이름]; </pre>	두 테이블 모두에 있는 값만 합치기
LEFT JOIN		<pre> SELECT [컬럼 이름] FROM [테이블 A 이름] LEFT JOIN [테이블 B 이름] ON [테이블 A 이름].[컬럼 A 이름] = [테이블 B 이름].[컬럼 B 이름]; </pre>	왼쪽 테이블에 있는 값만 합치기
RIGHT JOIN		<pre> SELECT [컬럼 이름] FROM [테이블 A 이름] RIGHT JOIN [테이블 B 이름] ON [테이블 A 이름].[컬럼 A 이름] = [테이블 B 이름].[컬럼 B 이름]; </pre>	오른쪽 테이블에 있는 값만 합치기
OUTER JOIN		<pre> SELECT [컬럼 이름] FROM [테이블 A 이름] LEFT JOIN [테이블 B 이름] ON [테이블 A 이름].[컬럼 A 이름] = [테이블 B 이름].[컬럼 B 이름] UNION SELECT [컬럼 이름] FROM [테이블 A 이름] RIGHT JOIN [테이블 B 이름] ON [테이블 A 이름].[컬럼 A 이름] = [테이블 B 이름].[컬럼 B 이름]; </pre>	두 테이블에 있는 모든 값 합치기
CROSS JOIN		<pre> SELECT [컬럼 이름] FROM [테이블 A 이름] CROSS JOIN [테이블 B 이름]; </pre>	두 테이블에 있는 모든 값을 각각 합치기
SELF JOIN		<pre> SELECT [컬럼 이름] FROM [테이블 A 이름] AS t1 INNER JOIN [테이블 A 이름] AS t2 ON t1.[컬럼 A 이름] = t2.[컬럼 A 이름]; </pre>	같은 테이블에 있는 값 합치기

## 함수 만들기

문법
<pre> CREATE FUNCTION [함수 이름] ( [입력값 이름] [데이터 타입], ...) RETURNS [결과값 데이터 타입]  BEGIN  DECLARE [임시값 이름] [데이터 타입]; SET [임시값 이름] = [입력값 이름]; 쿼리; RETURN 결과값  END </pre>

## 여러 테이블 한번에 다루기

표현	의미	활용
합집합		<pre> SELECT [컬럼 이름] FROM [테이블A 이름] UNION SELECT [컬럼 이름] FROM [테이블B 이름]; </pre>
중복 포함 합집합		<pre> SELECT [컬럼 이름] FROM [테이블A 이름] UNION ALL SELECT [컬럼 이름] FROM [테이블B 이름]; </pre>
교집합		<pre> SELECT [컬럼 이름] FROM [테이블A 이름] AS A INNER JOIN [테이블B 이름] AS B ON A.[컬럼1 이름] = B.[컬럼1 이름] AND ... AND A.[컬럼n 이름] = B.[컬럼n 이름]; </pre>
차집합		<pre> SELECT [컬럼 이름] FROM [테이블A 이름] AS A LEFT JOIN [테이블B 이름] AS B ON A.[컬럼1 이름] = B.[컬럼1 이름] AND ... AND A.[컬럼n 이름] = B.[컬럼n 이름] WHERE B.[컬럼 이름] IS NULL; </pre>

