

No. date / /

title

한빛 영어 노트

혼자 공부하는 SQL

혼자 공부하며 함께 만드는

혼공 용어 노트

목차

가나다순

고유 보조 인덱스 Unique Secondary Index	20	데이터베이스 관리 시스템	
고유 키 제약조건 UNIQUE Constraint	17	Database Management System, DBMS	07
관계 연산자 relational operator	10	데이터베이스 모델링 database modeling	08
관계형 데이터베이스 관리 시스템		동적 SQL 문	15
Relational DataBase Management System, RDBMS	07	문자형	12
구조화 질의어 Structured Query Language, SQL	07	변수 variable	13
균형 트리 Balanced tree, B-tree	19	별칭 alias	11
그래픽 사용자 인터페이스		보조 인덱스 Secondary Index	19
Graphical User Interface, GUI	24	뷰 view	17
기본 키 - 외래 키 Primary key - Foreign key	16	상호 조인 cross join	14
기본 키 제약조건 PRIMARY KEY Constraint	16	스크립트 모드	23
기본값 정의 DEFAULT	17	스크립트 언어(인터프리트 언어)	
날짜형	13	interpreted language	23
내부 조인 inner join	14	스키마 schema	09
널 Null	09	스토어드 프로시저 stored procedure	21
널 허용 안 함 Not Null, NN	10	스토어드 함수 stored function	22
노드 Node	19	실수형	13
논리 연산자 logical operation	10	실행 계획 Execution Plan	20
단순 보조 인덱스 Simple Secondary Index	20	열 column	09
대화형 모드	23	예약어 reserved word	09
데이터 형식 data type	09	외래 키 제약조건 FOREIGN KEY Constraint	17
데이터베이스 Database, DB	07	외부 라이브러리	23
		외부 조인 outer join	14

위젯 widget	24	커밋 commit	24
인덱스 index	18	커서 cursor	22
인덱스 검색 Index Scan	20	컴파일 언어 compiled language	23
일대다 관계 one to many	13	클러스터형 인덱스 Clustered Index	18
입력 매개변수	21	테이블 table	16
자체 조인 self join	14	트리거 trigger	22
정수형	12	페이지 Page	19
제약조건 constraint	16	페이지 분할	19
조인 join	13	폭포수 모델 waterfall model	08
집계 함수 aggregate function	11	프로젝트 project	08
체크 제약조건 CHECK Constraint	17	행 row	09
출력 매개변수	21	형 변환 type conversion	13

목차

ABC 순

aggregate function 집계 함수	11	Database Management System, DBMS	
alias 별칭	11	데이터베이스 관리 시스템	07
ALTER TABLE	16	database modeling 데이터베이스 모델링	08
ALTER VIEW	18	DEFAULT 기본값 정의	17
ANALYZE TABLE	20	DELETE	12
AUTO_INCREMENT	11	DELIMITER	21
Balanced tree, B-tree 균형 트리	19	DESCRIBE	16
BEGIN ~ END	21	DISTINCT	11
CALL	21	DROP INDEX	20
CASE 문	14	DROP VIEW	18
CHECK Constraint 체크 제약조건	17	endOfRow	22
Clustered Index 클러스터형 인덱스	18	Execution Plan 실행 계획	20
column 열	09	fetchone()	24
commit 커밋	24	FOREIGN KEY Constraint 외래 키 제약조건	17
compiled language 컴파일 언어	23	Graphical User Interface, GUI	
constraint 제약조건	16	그래픽 사용자 인터페이스	24
CREATE INDEX	20	GROUP BY	11
CREATE TABLE	16	HAVING	11
CREATE VIEW	18	IDLE: Integrated Development and Learning	
cross join 상호 조인	14	Environment	23
cursor 커서	22	IF 문	14
data type 데이터 형식	09	IF ~ ELSE 문	14
Database, DB 데이터베이스	07	import pymysql	24

index 인덱스	18	project 프로젝트	08
Index Scan 인덱스 검색	20	PyMySQL	23
inner join 내부 조인	14	Relational DataBase Management System,	
INSERT	11	RDBMS 관계형 데이터베이스 관리 시스템	07
INSERT INTO ~ SELECT	12	relational operator 관계 연산자	10
interpreted language 스크립트 언어(인터프리트 언어)	23	reserved word 예약어	09
join 조인	13	RETURN	22
LIKE	10	RETURNS	22
LIMIT	11	root	08
logical operation 논리 연산자	10	row 행	09
MySQL	07	schema 스키마	09
MySQL Workbench	08	Secondary Index 보조 인덱스	19
Node 노드	19	SELECT ~ FROM ~ WHERE	10
Not Null, NN 널 허용 안 함	10	self join 자체 조인	14
Null 널	09	SHOW INDEX	20
ON DELETE CASCADE	17	SHOW TABLE STATUS	20
ON UPDATE CASCADE	17	Simple Secondary Index 단순 보조 인덱스	20
one to many 일대다 관계	13	stored function 스토어드 함수	22
ORDER BY	10	stored procedure 스토어드 프로시저	21
outer join 외부 조인	14	Structured Query Language, SQL	
Page 페이지	19	구조화 질의어	07
PRIMARY KEY Constraint 기본 키 제약조건	16	table 테이블	16
Primary key - Foreign key	16	tkinter	24

목차

trigger 트리거	22	variable 변수	13
type conversion 형 변환	13	view 뷰	17
UNIQUE Constraint 고유 키 제약조건	17	waterfall model 폭포수 모델	08
Unique Secondary Index 고유 보조 인덱스	20	WHILE 문	15
UNSIGNED	12	widget 위젯	24
UPDATE	12	WITH CHECK OPTION	18
USE	10		

01장 [✓]

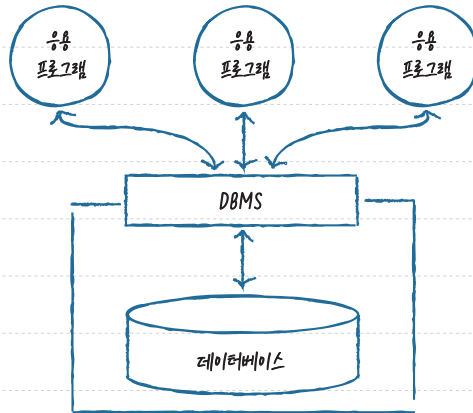
□ 데이터베이스 Database, DB [01장 024쪽]

데이터의 집합. 데이터의 저장소

□ 데이터베이스 Database Management System, DBMS [01장 025쪽]

관리 시스템

대량의 데이터베이스를 효율적으로 관리하고 운영하기 위한 시스템 또는 소프트웨어. 대표적으로 MySQL, 오라클(Oracle), SQL 서버(Server), MariaDB 등이 있다.



□ 관계형 데이터베이스 관리 시스템 Relational DataBase Management System, RDBMS [01장 030쪽]

테이블이라는 최소 단위로 이루어진 관계형 DBMS

□ 구조화 질의어 Structured Query Language, SQL [01장 031쪽]

RDBMS에서 데이터를 구축, 관리하고 활용하기 위해 사용되는 언어. 대부분의 DBMS 회사에서는 국제표준화기구에서 지정한 '표준 SQL'을 준수한다.

□ MySQL 오라클 사에서 제공하는 데이터베이스 관리 소프트웨어. 대용량의 데 [01장 034쪽]

이터를 관리하고 운영하는 기능을 제공한다.

□ root MySQL의 모든 권한이 있는 관리자 이름. MySQL을 설치할 때 기본적으로 생성된다. [01장 042쪽]

□ MySQL Workbench MySQL 서버에 접속해 SQL 문을 입력/실행할 수 있도록 돕는 클라이언트 도구. [01장 046쪽]

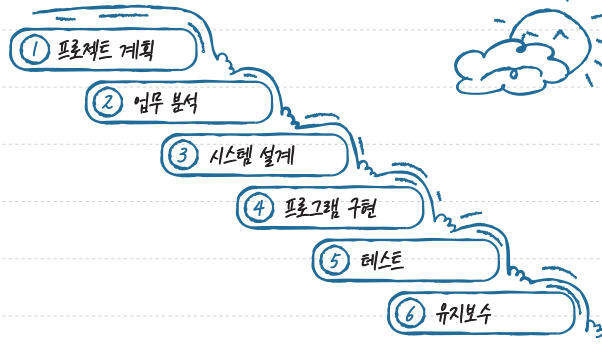
02장

□ 프로젝트 project [02장 057쪽]
현실 세계의 업무를 컴퓨터 시스템으로 옮겨 놓는 일련의 과정

□ 데이터베이스 모델링 database modeling [02장 059쪽]
데이터베이스 구축 전 테이블의 구조를 미리 설계하는 과정. 폭포수 모델의 업무 분석과 시스템 설계 단계에 해당한다.



□ 폭포수 모델 waterfall model [02장 058쪽]
소프트웨어 개발 절차 중 하나로, 위에서 아래로 폭포가 떨어지듯 개발 단계가 진행되기 때문에 붙여진 이름이다.



□ 데이터 형식	data type [02장 061쪽]
	테이블의 열에 저장될 데이터의 형식. 문자형(CHAR), 정수형(INT), 날짜형 (DATE) 등이 있다.
□ 열	column [02장 061쪽]
	테이블의 세로. 테이블은 여러 개의 열로 구성되며, 각 열을 구분하기 위해 열 이름을 사용한다.
□ 행	row [02장 061쪽]
	테이블의 가로. 실질적인 데이터(행 데이터)를 의미한다.
□ 스키마	schema [02장 069쪽]
	MySQL 안에 들어 있는 데이터베이스
□ 예약어	reserved word [02장 072쪽]
	기준에 약속된 SQL. SELECT, FROM, WHERE 등이 있다. 쿼리 창에서 입력하면 자동으로 파란색으로 표시된다.
□ 널	Null [02장 072쪽]
	비어 있는 값

□ 널 허용 안 함

Not Null, NN

[02장 072쪽]

빈 값을 허용하지 않는, 즉 반드시 입력해야 하는 값

03장

□ USE

사용할 데이터베이스를 선택하는 구문으로, 한 번 지정하면 계속 유지된다. MySQL 워크벤치를 재시작하거나 쿼리 창을 새로 열면 다시 USE를 실행해야 한다. [03장 111쪽]

□ SELECT ~
FROM ~
WHERE

가장 기본적인 SQL 문. SELECT 다음에는 열 이름이, FROM 다음에는 테이블 이름이, WHERE 다음에는 다양한 조건식이 나온다. [03장 112쪽]

□ 관계 연산자

relational operator [03장 117쪽]
WHERE 절에서 숫자로 표현된 데이터 범위의 크다/작다/같다 등을 지정하는 기호. >, <, >=, <= 등이 있다.

□ 논리 연산자

logical operation [03장 117쪽]
WHERE 절에서 2가지 이상의 조건을 지정할 때 사용. AND, OR 등으로 참/거짓을 판별한다.

□ LIKE

문자열의 일부 글자를 검색할 때 사용. 모두 허용할 때는 %를, 하나로 지정할 때는 _(언더 바)를 사용한다. [03장 119쪽]

□ ORDER BY

결과가 출력되는 순서를 조절한다. 오름차순인 ASC(Ascending)와 내림차순인 DESC(Descending) 중 선택할 수 있다. [03장 125쪽]

□ LIMIT	출력하는 개수를 제한하며, 주로 ORDER BY와 함께 사용한다.	[03장 128쪽]														
□ DISTINCT	조회된 결과에서 중복된 데이터를 1개만 남긴다. DISTINCT를 열 이름 앞에 붙여 사용한다.	[03장 129쪽]														
□ GROUP BY	데이터를 그룹으로 묶어주는 기능을 한다. 주로 집계 함수와 함께 사용된다.	[03장 130쪽]														
□ 별칭	alias SELECT 문에서 실제 열 이름 대신 출력되도록 설정하는 문자	[03장 132쪽]														
□ 집계 함수	aggregate function 데이터를 그룹화(grouping)해주는 기능을 한다. 주로 GROUP BY와 함께 사용한다.	[03장 132쪽]														
<div style="border: 1px solid #ccc; border-radius: 15px; padding: 10px; margin: 10px 0;"> <p style="margin: 0;">그것이 알고싶다 집계 함수 종류</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #0056b3; color: white;"> <th style="padding: 5px;">함수명</th> <th style="padding: 5px;">설명</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">SUM()</td> <td style="padding: 5px;">합계를 구한다.</td> </tr> <tr> <td style="padding: 5px;">AVG()</td> <td style="padding: 5px;">평균을 구한다.</td> </tr> <tr> <td style="padding: 5px;">MIN()</td> <td style="padding: 5px;">최소값을 구한다.</td> </tr> <tr> <td style="padding: 5px;">MAX()</td> <td style="padding: 5px;">최대값을 구한다.</td> </tr> <tr> <td style="padding: 5px;">COUNT()</td> <td style="padding: 5px;">행의 개수를 센다.</td> </tr> <tr> <td style="padding: 5px;">COUNT(DISTINCT)</td> <td style="padding: 5px;">행의 개수를 센다(중복은 1개만 인정).</td> </tr> </tbody> </table> </div>			함수명	설명	SUM()	합계를 구한다.	AVG()	평균을 구한다.	MIN()	최소값을 구한다.	MAX()	최대값을 구한다.	COUNT()	행의 개수를 센다.	COUNT(DISTINCT)	행의 개수를 센다(중복은 1개만 인정).
함수명	설명															
SUM()	합계를 구한다.															
AVG()	평균을 구한다.															
MIN()	최소값을 구한다.															
MAX()	최대값을 구한다.															
COUNT()	행의 개수를 센다.															
COUNT(DISTINCT)	행의 개수를 센다(중복은 1개만 인정).															
□ HAVING	집계 함수와 관련된 조건을 제한하며, 반드시 GROUP BY 절 다음에 나와야 한다.	[03장 134쪽]														
□ INSERT	테이블에 데이터를 입력(삽입)하는 기본적인 SQL 문	[03장 141쪽]														
□ AUTO_INCREMENT	열을 정의할 때 1부터 증가하는 값을 자동으로 입력해준다. 해당 열은 반드시 PRIMARY KEY로 지정해야 한다.	[03장 142쪽]														

□ **INSERT INTO ~ SELECT** 다른 테이블의 데이터를 가져와 한 번에 입력한다. 단, SELECT 문의 열 개수는 INSERT를 할 테이블의 열 개수와 같아야 한다. [03장 144쪽]

□ **UPDATE** 기존에 입력되어 있는 값을 수정하는 명령. 주로 WHERE 절과 함께 사용한다. WHERE 절을 생략하면 테이블의 모든 행의 값이 변경되므로 주의해야 한다. [03장 146쪽]

□ **DELETE** 행 단위로 데이터를 삭제한다. WHERE 절을 생략하면 전체 행 데이터가 삭제되므로 주의해야 한다. [03장 149쪽]

그것이 알고싶다 테이블 삭제

- DELETE: 행 단위로 삭제하고 빈 테이블은 남긴다. 데이터 양이 많을 경우 시간이 오래 걸릴 수 있다.
- DROP: 테이블 자체를 삭제한다.
- TRUNCATE: WHERE 문을 사용할 수 없어 조건 없이 전체 행을 삭제하고 빈 테이블은 남긴다. 대량의 행 데이터를 모두 삭제할 때는 DELETE보다 TRUNCATE가 효율적이다.

04장

□ **정수형** 소수점이 없는 숫자이며, TINYINT, SMALLINT, INT, BIGINT 등이 있다. [04장 159쪽]

□ **UNSIGNED** 정수형에 붙이면 범위가 0부터 지정된다. [04장 161쪽]

□ **문자형** 글자를 저장하기 위해 사용하며, 입력할 최대 글자의 개수를 지정해야 한다. 고정길이 문자형인 CHAR(Character)와 가변길이 문자형인 VARCHAR (Variable Character)가 있다. [04장 162쪽]

□ 실수형 소수점이 있는 숫자를 저장할 때 사용한다. 소수점 아래 7자리까지 [04장 166쪽]
표현되는 FLOAT와 소수점 아래 15자리까지 표현되는 DOUBLE이 있다.

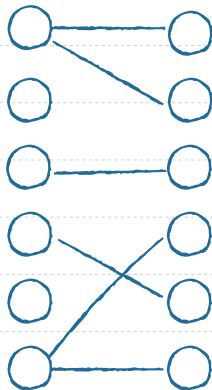
□ 날짜형 날짜를 저장하는 DATE, 시간을 저장하는 TIME, 날짜와 시간을 둘 [04장 166쪽]
다 저장하는 DATETIME이 있다.

□ 변수 **variable** [04장 166쪽]
MySQL에서는 변수 이름 앞에 @를 붙이고 SET 문으로 변수에 값을 대입한다.

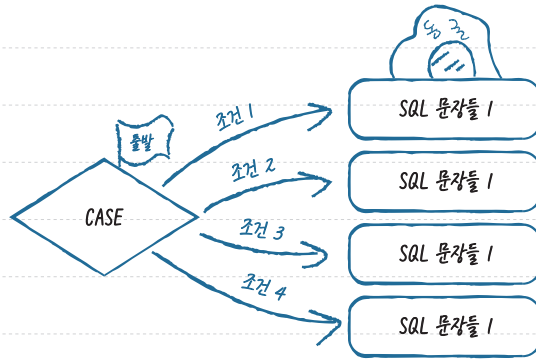
□ 형 변환 **type conversion** [04장 169쪽]
문자형, 정수형 등의 데이터 형식을 다른 형태로 변경하는 것. 직접 함수를 사용해
서 변환하는 명시적 변환-explicit conversion과 별도의 지시 없이 자연스럽게 변환되는
암시적 변환-implicit conversion이 있다. 명시적 변환에는 CAST(), CONVERT() 등을
사용한다.

□ 조인 **join** [04장 175쪽]
두 개의 테이블을 서로 묶어서 하나의 결과를 만들어 내는 것

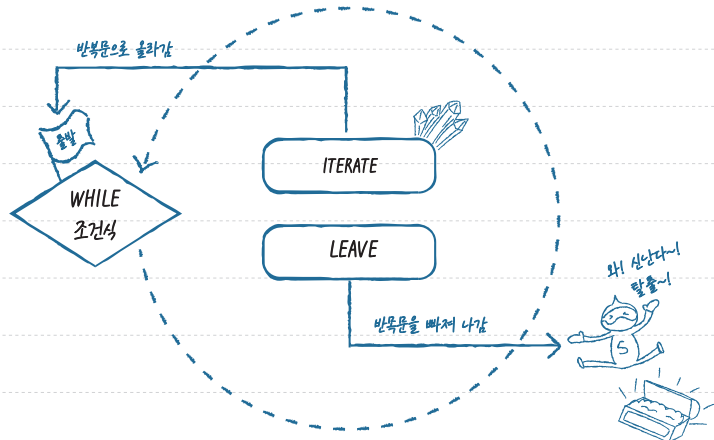
□ 일대다 관계 **one to many** [04장 176쪽]
두 테이블이 조인할 경우 한쪽 테이블에는 하나의 값만 존재해야 하지만, 연결된
다른 테이블에는 여러 개의 값이 존재할 수 있는 관계를 말한다. 주로 기본 키(PK)
와 외래 키(FK) 관계로 맺어져 있어 'PK-FK 관계'라고 부르기도 한다.



<p>□ 내부 조인</p>	<p>inner join [04장 177쪽]</p> <p>두 테이블을 조인할 때, 두 테이블에 모두 데이터가 있어야 결과가 추출된다. 조인 중에 가장 많이 사용되며, 일반적으로 조인이라고 하면 바로 이 내부 조인을 뜻한다.</p>
<p>□ 외부 조인</p>	<p>outer join [04장 184쪽]</p> <p>두 테이블을 조인할 때, 필요한 데이터가 한쪽 테이블에만 있어도 결과를 추출할 수 있다.</p>
<p>□ 상호 조인</p>	<p>cross join [04장 186쪽]</p> <p>한쪽 테이블의 모든 행과 다른 쪽 테이블의 모든 행을 조인시키는 기능. 따라서 상호 조인 결과의 전체 행 개수는 두 테이블의 각 행 개수를 곱한 개수가 된다.</p>
<p>□ 자체 조인</p>	<p>self join [04장 189쪽]</p> <p>자신이 자신과 조인된다는 의미로, 하나의 테이블에 서로 다른 별칭을 붙여 사용한다.</p>
<p>□ IF 문</p>	<p>다양한 조건을 처리할 때 사용한다. 조건식이 참이라면 결과가 실행 [04장 197쪽] 되지만, 그렇지 않으면 실행되지 않는다.</p> 
<p>□ IF ~ ELSE 문</p>	<p>조건식이 참일 때와 거짓일 때 각각 다른 부분을 실행한다. [04장 198쪽]</p>
<p>□ CASE 문</p>	<p>2가지 이상의 여러 가지 조건에 따라 처리가 가능하며, '다중 분기'라 [04장 201쪽] 고 부른다.</p>



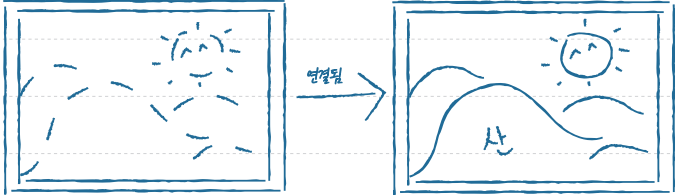
□ **WHILE 문** 조건식이 참인 동안에는 계속 같은 내용을 반복한다. 지정한 레이블 [04장 206쪽]로 가서 계속 실행하는 ITERATE 문과 지정한 레이블을 빠져나가는 LEAVE 문을 함께 사용할 수 있다.



□ **동적 SQL 문** 미리 SQL 문을 준비한 후에 필요할 때 실행하는 것이다. PREPARE [04장 210쪽]는 SQL 문을 준비만 해놓고, EXECUTE는 준비한 SQL 문을 실행한다. 실행 후에는 DEALLOCATE PREPARE로 문장을 해제해야 한다.

05장

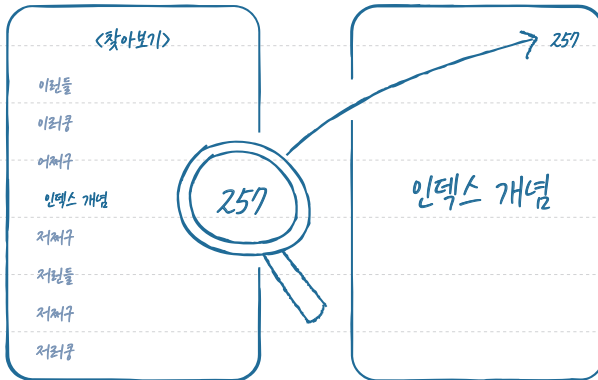
□ 테이블	table	[05장 216쪽]
	표 형태로 구성된 2차원 구조로, 열(column)과 행(row)으로 구성되어 있다.	
□ CREATE TABLE	테이블을 생성하는 SQL로, 테이블 이름, 열 이름, 데이터 형식 등을 지정한다.	[05장 219쪽]
□ 기본 키 - 외래 키	Primary key - Foreign key	[05장 223쪽]
	두 테이블이 일대다로 연결되는 관계로 기본 키-외래 키 관계를 맺으면 오류가 없는 데이터가 된다. 기본 키가 있는 테이블을 기준 테이블, 외래 키가 있는 테이블을 참조 테이블이라고 한다.	
□ 제약조건	constraint	[05장 235쪽]
	데이터의 무결성, 즉 데이터에 결함을 없애기 위해 제한하는 조건.	
□ 기본 키 제약조건	PRIMARY KEY Constraint	[05장 236쪽]
	많은 행 데이터 중 특정 데이터를 구분할 수 있는 식별자, 기본 키(Primary Key)를 지정하여 중복된 데이터가 입력되지 않도록 제한하는 제약조건이다.	
□ DESCRIBE	테이블 또는 뷰의 정보를 조회하는 SQL 문. 줄여서 DESC라고 써도 된다.	[05장 238쪽]
□ ALTER TABLE	이미 만들어진 테이블을 수정하는 SQL 문	[05장 238쪽]

□ 외래 키	FOREIGN KEY Constraint	[05장 239쪽]
제약조건	외래 키로 두 테이블 사이를 기본 키-외래 키 관계로 연결하여 무결한(오류가 없는) 데이터로 만드는 제약조건이다.	
□ ON UPDATE CASCADE	기준 테이블의 기본 키를 변경하면 참조 테이블의 외래 키도 자동으로 변경되는 기능	[05장 245쪽]
□ ON DELETE CASCADE	기준 테이블의 기본 키를 삭제하면 참조 테이블의 외래 키도 자동으로 삭제되는 기능	[05장 245쪽]
□ 고유 키	UNIQUE Constraint	[05장 247쪽]
제약조건	중복되지 않는 유일한 값을 입력해야 하는 제약조건이다. 기본 키와는 다르게 NULL 값을 허용하며 여러 개를 설정해도 된다.	
□ 체크 제약조건	CHECK Constraint	[05장 248쪽]
	입력되는 데이터를 점검하는 제약조건이다. 해당 열의 정의 뒤에 CHECK(조건)을 추가하여 사용한다.	
□ 기본값 정의	DEFAULT	[05장 250쪽]
	값을 입력하지 않았을 때 자동으로 입력될 값을 미리 지정해 놓는 제약조건이다.	
□ 뷰	view	[05장 256쪽]
	가상의 테이블. 테이블처럼 실제 데이터를 가지고 있지는 않고, SELECT 문으로 만들어져 있기 때문에 뷰에 접근하는 순간 SELECT가 실행되면서 그 결과가 화면에 출력되는 방식이다.	
		
	뷰(실체 없음)	테이블(실체 있음)

□ CREATE VIEW	뷰를 생성하는 SQL 문	[05장 257쪽]
□ ALTER VIEW	뷰를 수정하는 SQL 문	[05장 262쪽]
□ DROP VIEW	뷰를 삭제하는 SQL 문	[05장 262쪽]
□ WITH CHECK OPTION	뷰에 설정된 범위에 따라 그에 해당되는 값만 입력되도록 지정하는 SQL 문	[05장 267쪽]

06장 [✓]

□ 인덱스	index	[06장 275쪽]
	테이블에서 데이터를 빠르게 조회할 수 있도록 도와주는 도구. 책 뒤의 '찾아보기'와 비슷한 개념이다.	



□ 클러스터형 인덱스	Clustered Index	[06장 282쪽]
	기본 키(Primary Key)로 지정하면 자동으로 생성되고 테이블당 1개만 만들 수 있다. 해당 열을 기준으로 자동 정렬된다.	

□ 보조 인덱스

Secondary Index

[06장 285쪽]

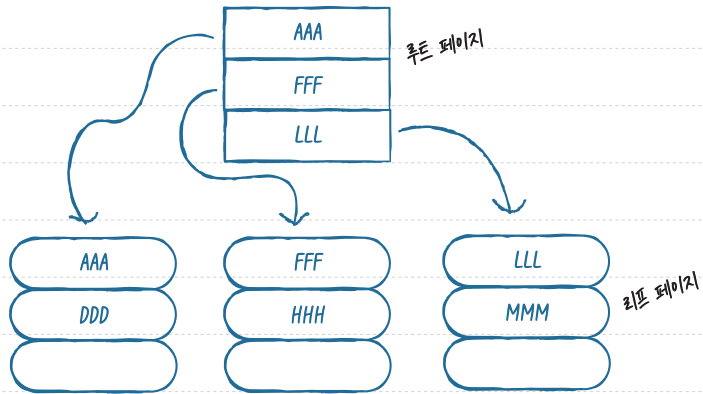
고유 키(Unique)로 지정하면 자동으로 생성되며, 테이블당 여러 개를 만들 수 있다. 자동 정렬은 되지 않는다.

□ 균형 트리

Balanced tree, B-tree

[06장 291쪽]

자료 구조에서 범용적으로 사용되는 데이터 구조. 클러스터형 인덱스와 보조 인덱스 모두 내부적으로 이 균형 트리로 만들어진다.



□ 노드

Node

[06장 292쪽]

균형 트리 구조에서 데이터가 저장되는 공간. 가장 상위부터 루트 노드, 중간 노드, 리프 노드로 구분한다.

□ 페이지

Page

[06장 292쪽]

MySQL에서는 노드를 페이지라고 부른다. 16Kbyte(16384byte) 크기를 가진 최소한의 저장 단위로, 데이터를 1건만 입력해도 1개 페이지가 필요하다.

□ 페이지 분할

데이터 입력 시 입력할 페이지에 공간이 없는 경우 새로운 페이지를 준비해서 데이터를 나누는 작업. 페이지 분할이 일어날수록 MySQL이 느려지기 때문에 성능에 나쁜 영향을 줄 수 있다.

□ 인덱스 검색	Index Scan [06장 304쪽]
	클러스터형 또는 보조 인덱스를 이용해서 데이터를 검색하는 것. 속도는 인덱스를 사용하지 않았을 때보다 빠르고, 클러스터형 인덱스가 보조 인덱스보다 조금 더 빠르다.
□ CREATE INDEX	인덱스를 직접 생성하는 구문. 보조 인덱스가 생성된다. [06장 308쪽]
□ DROP INDEX	CREATE INDEX 문으로 생성한 인덱스를 제거한다. 기본 키, 고유 키로 자동 생성된 인덱스는 DROP INDEX로 제거하지 못한다. [06장 309쪽]
□ SHOW INDEX	테이블에 어떤 인덱스가 설정되어 있는지 확인하는 SQL 문 [06장 311쪽]
□ SHOW TABLE STATUS	테이블에 생성된 인덱스의 크기를 확인하는 SQL 문 [06장 311쪽]
□ 단순 보조 인덱스	Simple Secondary Index [06장 312쪽] 중복을 허용하는 보조 인덱스. CREATE INDEX 문으로 생성한다.
□ 고유 보조 인덱스	Unique Secondary Index [06장 312쪽] 중복을 허용하지 않는 보조 인덱스. CREATE UNIQUE INDEX 문으로 생성한다.
□ ANALYZE TABLE	인덱스를 생성한 후 ANALYZE TABLE 문을 실행해서 테이블을 분석/처리해줘야 실제로 적용된다. [06장 313쪽]
□ 실행 계획	Execution Plan [06장 317쪽] MySQL 워크벤치에서 SQL 실행 시 인덱스의 사용 여부를 확인할 수 있다.

그것이 알고싶다 인덱스를 효과적으로 사용하는 방법

1. 인덱스는 열 단위에 생성한다.
2. WHERE 절에서 사용되는 열에 인덱스를 만들어야 한다.
3. WHERE 절에 사용되더라도 자주 사용해야 가치가 있다.
4. 데이터의 중복이 높은 열은 인덱스를 만들어도 별 효과가 없다.
5. 클러스터형 인덱스는 테이블당 하나만 생성할 수 있다.
6. 사용하지 않는 인덱스는 제거한다.

07장

□ 스토어드 프로시저	stored procedure MySQL에서 제공하는 프로그래밍 기능. 쿼리 문의 집합으로 어떤 동작을 일괄 처리하기 위한 용도로 사용한다. SQL을 묶는 개념 외에 연산식, 조건문, 반복문 등을 사용할 수도 있다.	[07장 331쪽]
□ BEGIN ~ END	스토어드 프로시저는 BEGIN ~ END 사이에 코드를 구현한다.	[07장 331쪽]
□ DELIMITER	MySQL에서는 기본적으로 세미콜론(;)을 구분자로 사용하는데, 스토어드 프로시저 안에 있는 SQL의 끝에도 세미콜론을 사용하면 혼동될 수 있으므로 구분자를 \$\$, ##, %%, && 등으로 바꿔 사용하는 것이 좋다.	[07장 332쪽]
□ CALL	작성된 스토어드 프로시저를 실행(호출)하는 명령문	[07장 332쪽]
□ 입력 매개변수	스토어드 프로시저에 값을 전달한다. 형식은 IN 구문을 사용한다.	[07장 334쪽]
□ 출력 매개변수	스토어드 프로시저에서 처리된 결과를 돌려받는다. 형식은 OUT 구문을 사용한다.	[07장 335쪽]

□ 스토어드 함수 **stored function** [07장 346쪽]

MySQL에서 제공하는 다양한 내장 함수 외에 사용자가 직접 만들어서 사용하는 함수

□ RETURNS 스토어드 함수에서 반환할 값의 데이터 형식을 미리 지정할 때 사용 [07장 346쪽]

한다.

□ RETURN 스토어드 함수에서 값을 반환할 때 사용한다. [07장 346쪽]

□ 커서 **cursor** [07장 350쪽]

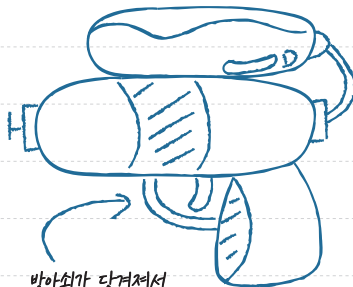
테이블의 모든 행을 한 행씩 처리할 때 사용하는 방식. DECLARE로 선언하며, 그 내용은 SELECT 문이다.

□ endOfRow 행의 끝을 파악하기 위한 변수로 endOfRow를 준비한다. 행의 끝 [07장 352쪽]

에 다다르면 TRUE 값인지 체크하는 방식을 사용한다.

□ 트리거 **trigger** [07장 360쪽]

테이블에서 DML 문(Insert, Update, Delete 등)의 이벤트가 발생할 때 자동으로 실행되는 프로그래밍 기능. 사용자가 추가 작업을 잊어버리는 실수를 방지해준다.



방아쇠가 당겨져서
트리거 실행

08장

- **외부 라이브러리** 파이썬에서 제공하지 않는 기능을 외부 개발자가 만들어서 제공하는 [08장 372쪽]
기능. 파이썬을 설치한 후에 필요한 라이브러리를 추가로 설치하면 된다.

- **PyMySQL** 파이썬을 MySQL과 연결시켜주는 대표적인 외부 라이브러리 [08장 372쪽]

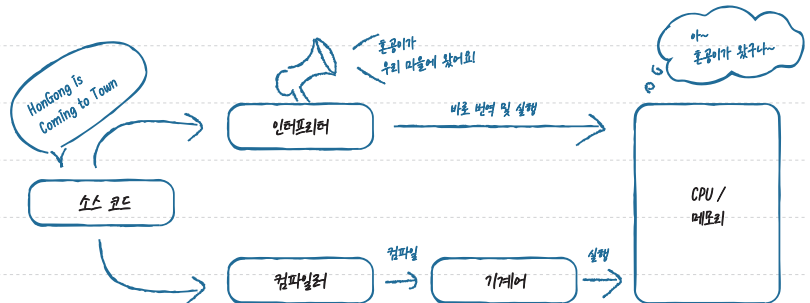
- **IDLE** **Integrated Development and Learning Environment** [08장 377쪽]
파이썬을 개발하는 통합 개발 환경

- **대화형 모드** 소스 코드 한 줄을 입력하고 [Enter] 키를 누르면 바로 실행하는 방식 [08장 377쪽]

- **스크립트 모드** 여러 줄을 코딩한 후 한 번에 실행하는 방식 [08장 378쪽]

- **컴파일 언어** **compiled language** [08장 378쪽]
소스 코드를 실행 가능한 기계어로 일괄 번역한 후에 번역이 완료된 파일(*.exe, *.class 등)이 실행되는 언어. 대표적인 컴파일 언어로 C, C++, 자바 등이 있다.

- **스크립트 언어 (인터프리트 언어)** **interpreted language** [08장 378쪽]
소스 코드를 한 줄씩 읽어 바로 실행되는 언어. 따라서 별도의 실행 파일이 생성되지 않는다. 대표적인 스크립트 언어로 파이썬, 자바스크립트, 펄 등이 있다.

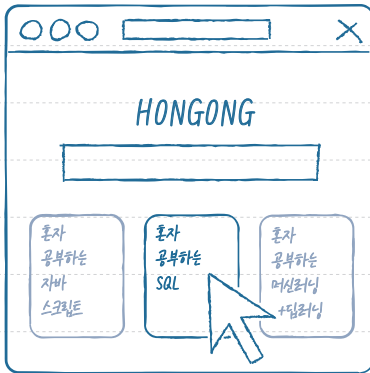


□ **import pymysql** SQL과 파이썬을 연동하기 위해 사용하는 명령 [08장 385쪽]

□ **커밋** **commit** [08장 387쪽]
변경한 내용을 확실하게 저장하는 것. 커밋 전의 데이터는 데이터베이스에 임시 저장된 상태이다.

□ **fetchone()** 파이썬에서 SELECT 문으로 조회한 데이터를 한 행씩 가져오기 위해 사용하는 함수. 모든 행에 한꺼번에 접근할 때는 fetchall()을 사용한다. [08장 389쪽]

□ **그래픽 사용자 인터페이스** **Graphical User Interface, GUI** [08장 395쪽]
그래픽으로 화면을 제공하여 사용자가 편리하게 작업할 수 있는 환경. 파이썬을 통해 윈도우에 출력되는 GUI 응용 프로그램을 작성할 수 있다.



□ **tkinter** 파이썬에서 GUI 관련 모듈을 제공해주는 표준 윈도우 라이브러리. 윈도우 화면이 필요할 때는 꼭 써줘야 한다. [08장 396쪽]

□ **위젯** **widget** [08장 397쪽]
윈도우에 나오는 라벨, 버튼, 프레임, 리스트 박스 등을 통합해서 지칭하는 용어

혼자
공부하는
사람들을 위한
용어 노트

