**02 실전용 SQL 미리 맛보기**

**테이블이(가) 표시된 사진

자동 생성된 설명02-1 데이터베이스 모델링**

DBMS의 조건

:여러 명이 공유하고 동시에 접근이 가능해야함

**테이블이(가) 표시된 사진

자동 생성된 설명02-2 데이터베이스 시작부터 끝까지**

**02-3 데이터베이스 객체**

테이블이(가) 표시된 사진

자동 생성된 설명

**3. SQL 기본 문법**

텍스트이(가) 표시된 사진

자동 생성된 설명- 데이터 베이스 만들기

1) if문과 결합해서 존재하면 삭제하는 문장

2) 데이터 베이스 새로 만들기

텍스트이(가) 표시된 사진

자동 생성된 설명

1) USE문은 데이터 베이스 선택하는 문장. 더블 클릭을 연상하면 됌

2) 테이블 만드는 과정

텍스트, 장치, 게이지, 측정기이(가) 표시된 사진

자동 생성된 설명1) 회원 테이블에 값을 입력

2) 첫번째 열인 순번은 자동입력이라 NULL로 표기

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명**SQL 기본 조회 명령어**

기본 구조 => 순서 중요

**USE문**

먼저 사용할 데이터 베이스를 지정.

더블 클릭 생각하면 됌

**Select문** : 가져올 데이터의 열을 의미. 1) \* 는 전체를 의미 2)별칭 ” ” or AS “ “를 써서 가능

**FROM문** : 테이블의 출처를 의미함.

**WHERE문** : 조건식 의미하며, 연산자 사용가능

- 연산자 : 숫자은 연속적, 문자는 불연속적

조건식이 숫자인 경우 ( 부등호, AND, OR, [NOT] BETWEEN ~ AND ~ )

조건식이 문자인 경우 ( AND, OR, [NOT] IN )

**LIKE 이용** : 정규식과 비슷한 개념. 문자열의 일부 글자로 검색

🎈 % : 특정 문자 검색

‘우%’ => 앞글자가 ‘우’이고 뒤에는 무엇이든(공백 포함) 허용

‘%우’ => 뒷글자가 ‘우’이고 앞에는 무엇이든(공백 포함) 허용

‘%우%’ => 사이에 ‘우’가 있고, 앞뒤에는 무엇이든(공백 포함) 허용

🎈 \_ : 한 글자씩 매치

‘\_ \_ 핑크’ : ex)블랙 핑크 처럼 글자 수는 맞춰서 무엇이든(공백 포함) 허용

**- 서브 쿼리** : SQL의 2개의 명령문을 하나로 합친 것

Q. 이름이 ‘에이핑크’인 회원의 키를 검색 + 그 키를 바탕으로 회원의 평균 키보다 큰 회원을 검색



어떻게 합쳐야 할까? 일단 SELECT와 FROM은 변하지 않아 영향 X

WHERE (1)SQL문이 164를 의미하니깐 그것을 통째로 넣어줌



**ORDER BY 문** : 결과 출력되는 순서를 조절함. 기본 값은 ASC(오름차순)이고 DESC(내림차순) 가능

- 예시 : ORDER문이 두개인 경우에는 첫번째 정렬 후, 값이 같으면 뒤에 정렬로 분류

(height 내림차순 후 height 값이 같으면 debut\_data 오름차순)

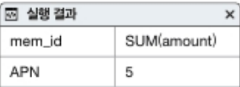
**LIMIT 문** : 출력하는 개수 제한

- 예시

1) 숫자가 한 개인 경우 : 위에서부터 개수만큼 제한(0~3까지와 동일) =>

2) 숫자가 두 개인 경우(a, b) : a번째부터 b개 만큼 개수제한 =>

**GROUP BY 문** : 그룹으로 묶어주는 역할



왜 필요할까? A, B마트에 갔는데 A 마트의 각각의 제품의 금액 말고 총액을 알고 싶음.

- 집계 함수 : GROUP문에서 쓰는 함수

테이블이(가) 표시된 사진

자동 생성된 설명

- 예시

1) 합계 이용 =>

2) 구매한 금액 (가격 \* 수량) =>

3) 전체 회원수 COUNT =>

4) NULL제외하고 COUNT =>

**HAVING 문** : GROUP의 조건문 (WHERE에서는 사용 불가) why? WHERE문이 GROUP보다 앞이라서

- 예시텍스트이(가) 표시된 사진

자동 생성된 설명 : 합계 조건문 =>

**데이터 구축 방법**

**INSERT 문** : 새로 회원을 테이블에 입력할 때

예시 1) 순서대로 넣을 때 =>

 2) 순서를 바꾸고 싶을 때 =>

테이블이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명📌유일키를 자동으로 하나씩 증가하게 할수 없을까? AUTO\_INCREMENT로 유일 키 한 개씩 증가, 입력할땐 null로 해야함.

텍스트이(가) 표시된 사진

자동 생성된 설명📌그렇다면 하나씩 말고 증가값을 조절할 수 있을까 또 시작점을 내가 조절 할 수 있을까?

ALTER문에서 시작값을 지정 가능하며,

SET문에서 증가값을 지정가능함.

테이블이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷, 문패이(가) 표시된 사진

자동 생성된 설명📌하나씩 입력하지 말고 다른 테이블을 한번에 입력할 수 있을까?

1) 다른 테이블 개수 조사

2) 다른 테이블 구조(열에 무엇이 있는지) 확인

3) 모든 열을 사용할 수 없으니깐 사용할 열만 따로 테이블(city\_popul) 만듦.



텍스트이(가) 표시된 사진

자동 생성된 설명4) 원래 테이블에 삽입

**텍스트이(가) 표시된 사진

자동 생성된 설명UPDATE 문** : 회원의 정보가 변경되어 수정할 때

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명📌WHERE가 없는 경우에는 모든 테이블을 변경하고 싶을 때

**텍스트이(가) 표시된 사진

자동 생성된 설명DELETE 문** : 회원의 정보를 삭제할 때

📌대용량으로 삭제할 때는?

텍스트이(가) 표시된 사진

자동 생성된 설명DELETE는 오래 걸리지만 조건식 가능, 테이블 구조 남아있음

DROP은 가장빠르며 테이블 구조도 지음.

TRUNCATE은 빠르게 가능하며 조건식 X, 테이블 구조 남아있음

4. SQL 고급문법

**데이터 형식**

**정수형** 📌 양수만 존재하게 바꿀수 없을까?

테이블이(가) 표시된 사진

자동 생성된 설명 => UNSIGNDE 예약어 사용

스크린샷, 텍스트, 라인이(가) 표시된 사진

자동 생성된 설명

**문자형**  📌언제 숫자로 잡고 문자로 잡아야할까?

CHAR는 고정길이 문자형 =>더하고 빼고 연산 의미있거나 크다/작다 처럼 순서 의 VARCHAR는 가변길이 문자형 미가 있는 경우에는 숫자 사용

|  |  |  |  |
| --- | --- | --- | --- |
| 데이터 형식 | | 바이트 수 | 사용하는 곳 |
| TEXT 형식 | TEXT | 1 ~ 65535 | 대본과 같은 데이터 |
| LONGTEXT | 1 ~ 4294967295 |
| BLOB 형식 | BLOB | 1 ~ 65535 | 이미지, 동영상 등의 데이터 |
| LONGBLOB | 1 ~ 4294967295 |

**대량의 데이터 형식**

**테이블이(가) 표시된 사진

자동 생성된 설명실수형**

**텍스트이(가) 표시된 사진

자동 생성된 설명날짜형**

**텍스트, 폰트, 라인, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명변수**

when? 없는 값을 만들 때

📌 변수는 LIMIT에는 사용할 수 없는데, 해결 방법은?

텍스트이(가) 표시된 사진

자동 생성된 설명

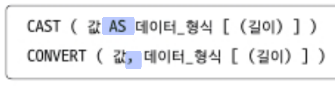
②은 실행시키지 않고 준비만 하는 구문, “?” 부분은 현재는 모르지만 나중에 값 채워짐

③은 USING으로 “?”에 @count를 변수의 값을 대입하는 것

**데이터 형 변환**

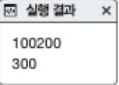
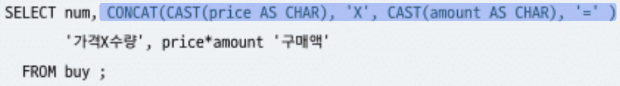
**명시적인 변환(강제변환)** -> 숫자 타입을 바꾸거나 문자열로 바꾸고 싶은 경우

📌 평균값이 실수값이 나왔을 때 정수형으로 바꾸고 싶으면??



📌 숫자를 문자로 바꾸고 싶은 경우??

=> 계산하지 않고 그대로 나타내기 위해 숫자를 정수로 바꿔서 출력

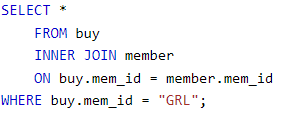
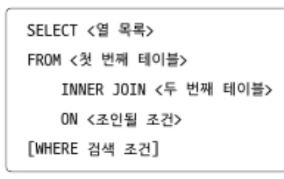


**암시적인 변환(자동 변환)** -> SQL에서는 숫자 우선

**조인** -> 두 개의 테이블을 서로 묶어서 하나의 결과로 만드는 것

💥💥 필수 조건 : 두 테이블이 일대다(PK -FK관계)로 연결되어 있어야 함. why? 연결 고리

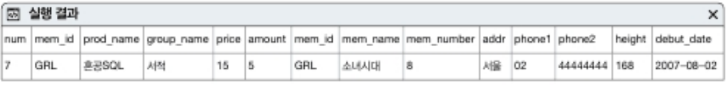
**내부 조인** -> 일대다 관계(무결정성)에서는 다쪽이 일쪽에 갈 수 X. 그래서 다 쪽에 일에 정보를 연결해줌



❓❓ 왜 mem\_id가 아니고 buy.mem\_id 또는 member.mem\_id를 써야 될까?

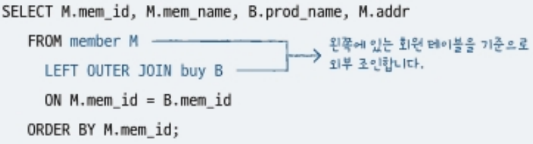
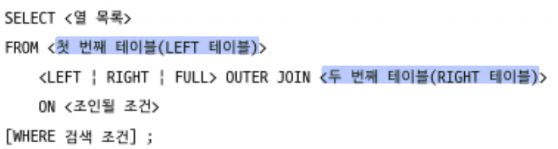
=> PK-FK관계로 이어진 열은 대부분 같은 이름을 써서 확실히 구분하기 위해 앞에 테이블. 을 붙임.

=>그래서 내부 조인을 쓸때는 구분 잘하기!! (별칭으로도 쓰기 가능)



두번째 테이블 내용 첫번째 테이블 내용

**외부 조인** -> 두 테이블을 조인할 때 한쪽 테이블에만 있어도 결과 추출 가능



테이블이(가) 표시된 사진

자동 생성된 설명

내부 조인과 외부 조인에서 결과가 여기서 차이가 남

내부 조인은 prod\_name 값이 null인 부분은 출력 X

외부 조인은 prod\_name 값이 null로 출력

**기타 조인**

1) 상호 조인 : 한쪽 테이블의 모든 행과 다른 쪽 테이블의 모든 행을 조인시키는 기능으로 주 용도는 테스트하기 위해 대용량의 데이터의 데이터를 생성할 때 사용

테이블이(가) 표시된 사진

자동 생성된 설명2) 자체 조인 : 자신이 자신과 조인함. 회사의 조직 관계에서 볼 수 있음.

텍스트이(가) 표시된 사진

자동 생성된 설명

SQL 프로그래밍 부분 (p 196 ~ p 213)

테이블 만들기 (p 216 ~ p 233)

제약조건 -> 데이터의 무결성을 지키기 위해 제한하는 조건

**기본키**

1) 데이터를 구분 할 수 있는 식별자이며, 중복 X, NULL X

2) 자동으로 클러스터형 인덱스가 생성

3) 테이블은 한 개의 기본 키만 가질 수 있음

생성하는 방법 3가지

1) 생성할 때, 해당 열에 PRIMARY KEY 붙임 2)생성할 때, 마지막 행에 PRIMARY KEY 추가

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명3) 수정할 때, ALTER TABLE문 사용

텍스트이(가) 표시된 사진

자동 생성된 설명

**외래 키**

1) 두 테이블 사이의 관계를 연결해주며, 무결성 보장해주는 역할

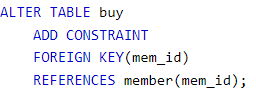
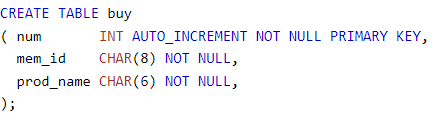
2) 반드시 다른 테이블의 기본 키와 연결됨(기본키 테이블을 기준 테이블, 외래 키 테이블을 참조 테이블)

생성 방법

1) 생성할 때, 마지막 행에 PK-FK관계를 맺어줌 텍스트이(가) 표시된 사진

자동 생성된 설명

2) 수정할 때, ALTER문을 이용해서 PK-FK관계를 맺어줌



텍스트, 도표, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명

📌 PK를 변경하는 경우에는 FK는 어떻게 동작할까?

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명=> foreign key 오류 발생(변경이나 삭제하는 경우)해 데이터의 무결성을 확보하게 됨.

📌 PK를 변경하는 경우에는 FK도 같이 변경하고 싶을 때는?

텍스트이(가) 표시된 사진

자동 생성된 설명

=> ALTER문에 ON ~ CASCADE을 이용하여 변경 및 삭제가 가능하도록 함.

=> 변경하는 경우에는 FK도 같이 변경

=> 삭제하는 경우에는 PK, FK 내용 같이 삭제

**기타 제약 조건**

1) **고유키** : 중복되지 않은 유일한 값 + NULL 허용 (NOT NULL 설정가능)

2) **체크 제약 조건** : 입력 되는 데이터를 제한하는 기능

텍스트, 장치, 측정기이(가) 표시된 사진

자동 생성된 설명생성할 때 :

수정할 때 :

(ALTER문)

3) **기본 값** : 입력하지 않아도 자동으로 입력될 값을 지정해 놓는 방법

생성할 때 :

텍스트이(가) 표시된 사진

자동 생성된 설명수정할 때 :

(ALTER문)

**뷰**

1) “바로가기 아이콘”을 의미하며, 뷰의 실체는 SELECT문이 됨.

도표, 라인, 스크린샷, 스케치이(가) 표시된 사진

자동 생성된 설명

**뷰 작동 방식**

📌뷰를 사용하는 이유?

1) 보안에 도움이 됌 -> 필요한 데이터만 접근 권한

2) 복잡한 SQL을 단순하게 만들 수 있음. ->

**뷰 생성/조회/삭제**

1) 생성 후 조회텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

2) 수정 후 조회 3) 삭제

텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

**뷰 정보확인**

1) 기존 뷰 정보 확인

2) 뷰의 소스 코드 확인

**뷰의 데이터 수정/삭제**

1) 수정 

2) 입력

📌만약에 167이하에 데이터만 보는 뷰인데 167이상인 데이터가 들어오면?

오류 X 입력됌 =>그래서 데이터의 불안정성이 생김

해결책은? WITH CHECK OPTION을 통해 데이터의 값을 제한함

3) 삭제

뷰가 참조한 테이블이 있어도 삭제 가능 단, 연결관계는 끊어짐

인덱스 : ‘찾아보기’개념으로,저장 공간을 활용하여 데이터베이스 테이블의 검색 속도를 향상시키기 위한 자료구조

장점

✔ 테이블을 조회하는 속도와 그에 따른 성능을 향상시킬 수 있다.

✔ 전반적인 시스템의 부하를 줄일 수 있다.

단점

✔ 인덱스를 관리하기 위해 DB의 약 10%에 해당하는 저장공간이 필요하다.

✔ 인덱스를 관리하기 위해 추가 작업이 필요하다.

✔ 인덱스를 잘못 사용할 경우 오히려 성능이 저하되는 역효과가 발생할 수 있다.

인덱스의 종류

클러스터형 인덱스 : 영어사전

✔ 테이블의 데이터를 정렬해서 저장되는 순서를 정의

✔ 테이블당 하나만 생성 가능

✔ PRIMARY KEY에 의해 자동으로 생성

✔ 리프 페이지 == 데이터페이지

보조인덱스(NON 클러스터형 인덱스) : 책 뒤의 찾아보기

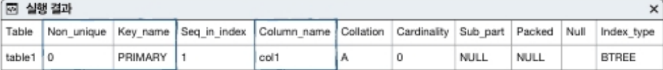
✔ 데이터와 인덱스(페이지 번호+#위치)를 각각 다른 위치에 저장해 순서의 변화 X

✔ 인덱스는 해당 데이터의 위치에 대한 포인터를 포함

✔ 테이블당 여러 개 생성 가능

✔ UNIQUE KEY에 의해 자동으로 생성

✔ 리프 페이지(데이터 주소) != 데이터 페이지(정렬되지 않은 데이터)

인덱스 정보 확인

Non\_unique : 고유하지 않아 즉 0(False)는 고유해 중복X, 1(True)는 고유하지않아 중복 O

Key\_name : 기본키인지, 고유키인지 알려줌 Column\_name : 해당하는 열

인덱스의 내부 작동

도표, 텍스트, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명클러스터형 인덱스와 보조 인덱스 모두 균형 트리(B-Tree)의 데이터의 구조를 갖음

데이터가 저장되는 공간을 노드라고 하고, MYSQL에서는 페이지라고 부름. 크기는 최소 16Kbyte.

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명클러스터 형 인덱스가 있는 경우 검색 방법

보조 인덱스가 있는 경우 검색 방법

텍스트, 도표, 평행, 평면도이(가) 표시된 사진

자동 생성된 설명

텍스트, 장치, 측정기, 스크린샷이(가) 표시된 사진

자동 생성된 설명리프 페이지만 있는 경우(인덱스가 없는 경우) 검색방법 => Full Table Scan

📌데이터 변경 작업(INSERT, UPDATE, DELETE)이 인덱스에서 오래 걸리는 이유? 페이지 분할

도표, 라인, 평면도, 스크린샷이(가) 표시된 사진

자동 생성된 설명1) 리프페이지의 빈공간이 있는 경우

=> 한 칸 아래로 데이터들을 이동시키고 그 위치에 데이터를 삽입함.

=> 비교적 빠른 시간에 처리 가능

텍스트, 도표, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명2) 리프페이지의 빈 공간이 없는 경우

=> 번호의 순서대로 데이터 분할작업함

텍스트, 도표, 스크린샷, 평면도이(가) 표시된 사진

자동 생성된 설명3) 중간 페이지의 공간이 없는 경우

=> 번호의 순서대로 데이터 분할작업함

인덱스 생성/삭제 문법

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명생성 삭제

인덱스를 효과적으로 사용하는 방법

✔ 인덱스는 열 단위에 생성하기( 2개 이상 묶는 것은 지양)

✔ WHERE 절에 자주 사용되는 인덱스 만들기(결국 인덱스는 WHERE절에 사용함)

✔ 데이터 중복이 낮은 곳(=카디널리티가 높은 값)에 인덱스 만들기