20241129

#기본 SELECT 문법

SELECT \* FROM customers;

SELECT \* FROM customers;

SELECT customerName, country FROM customer WHERE country=’USA’;

#숫자는 그냥 작성/글자는 무조건’’로 묶어준다.

#다르다DB에서는 NOT(!)개념

SELECT customerName, country FROM customer WHERE NOT country=’USA’;

SELECT customerNumber, credlimit FROM customers’

WHERE …’

#조건을 여러 개 사용할 때

#AND OR XOR; 참: 1, 거짓: 0

같으면 0 다르면 1=XOR;

AND이고

OR이거나

글자중에 일부를 포함한 경우 조회

WHERE contry LIKE ‘USA’

A로 시작한다=LIKE ‘a%’;=\*개념

#정렬 ORDER BY country(ASC)오름차순/(DESE)내림차순;

#특정수만큼만 조회

LIMIT 3;//3개의 행만

# LIMIT 데이터의 수 OFSET 몇번째부터 몇 개를 뽑을건지

Ex) LIMIT 3 OFFSET 0;

Ex) LIMIT 0,3;

무한스크롤//웹페이지 참조(인스타개념??) 몇 개 정해져있음

나라 알파벳순으로 저장(원래는 기본키가 디폴트)

#특정 조건에 맞는 데이터를 조회

#SELECT

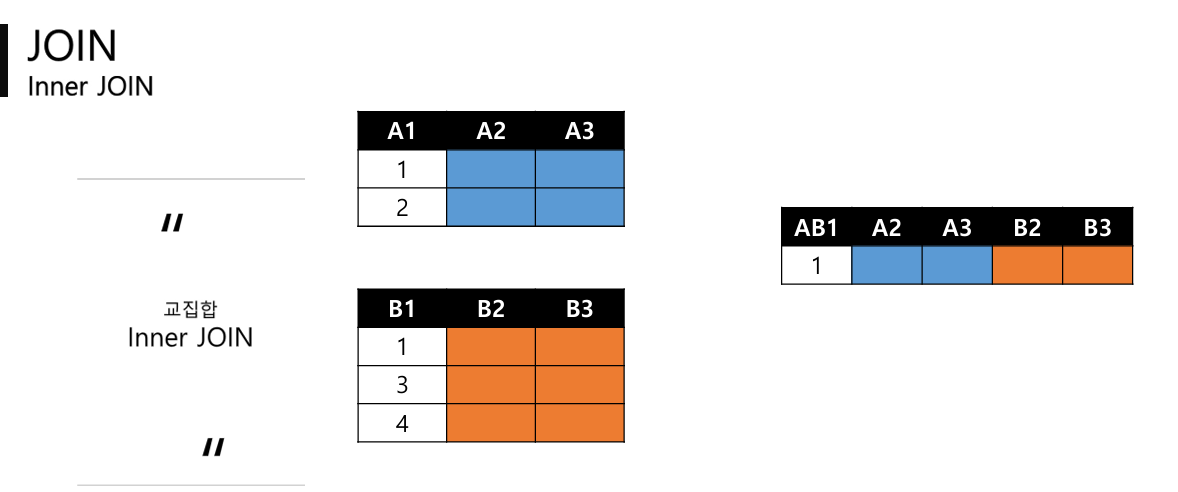
DB표

SELECT하면 메모리를 내가 원하는 거만 지정해서 뽑기 때문에

더 시간이 적게 걸린다.

이런 애들이 있다…

#SELECT에서 제일 중요한거 JOIN



이너조인은 구매내역 X 안나옴

아우터조인은 구매내역이없어도 회원정보 나옴

DB에서 순서의 속성은 전혀 상관이 없다.

SELECT 속성… FROM 테이블1 INNER JOIN 테이블2 ON 테이블1.속성= 테이블2.속성

Left조인

SELECT 속성… FROM 테이블1 LEFT JOIN 테이블2 ON 테이블1.속성= 테이블2.속성

빨간색이 외래키

노란색 참조키

개체 무결성

참조 무결성

도메인 무결성

보통 JOIN할 때는 외래캐 가지고 조인한다.

한땀한땀 실행하고 다음줄은 추가

전체를 말고 잘라서 봐라

두개의 테이블이 하나로 합쳐진다(같은애들끼리…)

외래키도 속성이다.

조인했을 경우 중복이 되었기 때문에 두개의 테이블이 같은 속성이 생기는 일이 발생

#런던에서 주문한 고객들의 주문번호와 고객 이름조회

#customers의 city, orders의 orderNumber, customers의 customerName

//조인부터 해라

//SELECT customer.orderNumber

#주문한 제품의 수량이 35개 이상인 주문의 주문 번호와 고객 이름

#terriotory가 APAC에서 근무하는 직원이 담당하는 고객이름과 고객번호 목록 조회

#주문 상태가 Shipped인 주문에 포함된 제품의 이름과 주문한 고객의 이름 조회

WHERE 가로

FROM 세로

서브쿼리 Left JOIN가 제일 좋다.

#내장함수

#집계함수

SELECT SUM(quantiyOrdered) FROM orderdetails;

도시별로 주문한 고객들의 수와 도시 이름

주문한 제품별 제품의 이름과 고객의 수

Territory별 근무하는 직원이 담당하는 고객의 수

도시별로 주문한 고객들의 수와 도시 이름

주문한 제품별 제품의 이름과 고객의 수

Territory별 근무하는 직원이 담당하는 고객의 수

#도시별로 주문한 고객들의 수와 도시 이름

SELECT cityname, COUNT(customerNumber) FROM customers group by city;

#주문한 제품별 제품의 이름과 고객의 수

SELECT products, COUNT(customerNumber) FROM customers group by products;

#Territory별 근무하는 직원이 담당하는 고객의 수

SELECT terriotory, COUNT(customerNumber) FROM customers group by customerNumber;

#도시별로 주문한 고객들의 수와 도시 이름

SELECT city, COUNT(customerNumber) FROM customers group by city;

#주문한 제품별 제품의 이름과 고객의 수

SELECT products, COUNT(customerNumber) FROM customers group by products;

#Territory별 근무하는 직원이 담당하는 고객의 수

SELECT Territory, COUNT(customerNumber) FROM customers group by customerNumber;

#중복제거 DISITINCT 속성 앞에

Group by가 있을 때만 having 사용 가능

HAVING COUNT(customers.customerNumber)>30;

데이트포멧함수