

SQL 문의 개요

■ SQL(Structured Query Language, 구조화된 질의 언어)

- 데이터베이스에서 사용되는 일종의 공통 언어
- NCITS(국제표준화위원회)에서 ANSI/ISO SQL이라는 명칭의 SQL 표준을 관리하고 있음
- 1992년에 제정된 ANSI-92 SQL과 1999년에 제정된 ANSI-99 SQL을 대부분의 DBMS 회사에서 SQL 표준으로 사용하고 있음
- 각 회사는 ANSI-92/99 SQL의 표준을 준수하면서도 자신의 제품 특성을 반영한 SQL에 별도의 이름을 붙임
- MySQL에서는 그냥 SQL, 오라클에서는 PL/SQL, SQL Server에서는 Transact SQL(T-SQL) 사용

SELECT 문

– 기본 검색문

- SELECT 키워드와 함께 검색하고 싶은 속성의 이름을 나열
- FROM 키워드와 함께 검색하고 싶은 속성이 있는 테이블의 이름을 나열
- 검색 결과는 테이블 형태로 반환됨

```
SELECT [ ALL | DISTINCT ] 속성_리스트
FROM   테이블_리스트;
```

- ALL
 - 결과 테이블이 레코드의 중복을 허용하도록 지정, 생략 가능
- DISTINCT
 - 결과 테이블이 레코드의 중복을 허용하지 않도록 지정

SELECT 문

고객 테이블에서 고객아이디, 고객이름, 등급 속성을 검색해보자.

결과 테이블

	고객아이디	고객이름	등급
1	apple	정소화	gold
2	banana	김선우	vip
3	carrot	고명석	gold
4	orange	김용욱	silver
5	melon	성원웅	gold
6	peach	오형준	silver
7	pear	채광주	silver

SELECT 문

고객 테이블에 존재하는 모든 속성을 검색해보자.

결과 테이블

	고객아이디	고객이름	나이	등급	직업	적립금
1	apple	정소화	20	gold	학생	1000
2	banana	김선우	25	vip	간호사	2500
3	carrot	고명석	28	gold	교사	4500
4	orange	김용욱	22	silver	학생	0
5	melon	성원웅	35	gold	회사원	5000
6	peach	오형준	(null)	silver	의사	300
7	pear	채광주	31	silver	회사원	500

SELECT 문

제품 테이블에서 제조업체를 검색해보자.

결과 테이블에서 제조업체가 중복됨

결과 테이블

제조업체
1 대한식품
2 민국푸드
3 한빛제과
4 한빛제과
5 대한식품
6 민국푸드
7 한빛제과

5/55

SELECT 문

제품 테이블에서 제조업체를 검색하되, ALL 키워드를 사용해보자.

결과 테이블에서 제조업체가 중복됨

결과 테이블

제조업체
1 대한식품
2 민국푸드
3 한빛제과
4 한빛제과
5 대한식품
6 민국푸드
7 한빛제과

6/55

SELECT 문

제품 테이블에서 제조업체 속성을 중복 없이 검색해보자.

결과 테이블에서 제조업체가 한 번씩만 나타남

결과 테이블

제조업체
1 대한식품
2 민국푸드
3 한빛제과

7/55

SELECT 문

- 기본 검색

- AS 키워드를 이용해 결과 테이블에서 속성의 이름을 바꾸어 출력 가능
 - 새로운 이름에 공백이 포함되어 있으면 큰따옴표나 작은따옴표로 묶어주어야 함
 - » 오라클에서는 큰따옴표, MS SQL 서버에서는 작은따옴표 사용
- AS 키워드는 생략 가능

8/55

SELECT 문

제품 테이블에서 제품명과 단가를 검색하되, 단가를 가격이라는 새 이름으로 출력해보자.

결과 테이블

	제품명	가격
1	그냥만두	4500
2	매운폴면	5500
3	콩떡파이	2600
4	맛난초콜릿	2500
5	얼큰라면	1200
6	통통우동	1550
7	달콤비스킷	1500

9/55

SELECT 문

– 산술식을 이용한 검색

- SELECT 키워드와 함께 산술식 제시
 - 산술식: 속성의 이름과 +, -, *, / 등의 산술 연산자와 상수로 구성
- 속성의 값이 실제로 변경되는 것은 아니고 결과 테이블에서만 계산된 결과 값이 출력됨

10/55

SELECT 문

제품 테이블에서 제품명과 단가 속성을 검색하되, 단가에 500원을 더해 '조정 단가'라는 새 이름으로 출력해보자.

결과 테이블

	제품명	조정 단가
1	그냥만두	5000
2	매운폴면	6000
3	콩떡파이	3100
4	맛난초콜릿	3000
5	얼큰라면	1700
6	통통우동	2050
7	달콤비스킷	2000

11/55

SELECT 문

– 조건 검색

- 조건을 만족하는 데이터만 검색

```
SELECT  [ ALL | DISTINCT ] 속성_리스트
FROM    테이블_리스트
[ WHERE 조건 ];
```

- WHERE 키워드와 함께 비교 연산자와 논리 연산자를 이용한 검색 조건 제시
 - 숫자뿐만 아니라 문자나 날짜 값을 비교하는 것도 가능
 - » 예) 'A' < 'C'
 - » 예) '2019-12-01' < '2019-12-02'
 - 조건에서 문자나 날짜 값은 작은따옴표로 묶어서 표현

12/55

SELECT 문

표 7-2 비교 연산자

연산자	의미
=	같다.
< >	다르다.
<	작다.
>	크다.
<=	작거나 같다.
>=	크거나 같다.

표 7-3 논리 연산자

연산자	의미
AND	모든 조건을 만족해야 검색한다.
OR	여러 조건 중 한 가지만 만족해도 검색한다.
NOT	조건을 만족하지 않는 것만 검색한다.

SELECT 문

제품 테이블에서 한빛제과가 제조한 제품의 제품명, 재고량, 단가를 검색해보자.

결과 테이블

	제품명	재고량	단가
1	공작파이	3600	2600
2	맛난초콜릿	1250	2500
3	달콤비스킷	1650	1500

SELECT 문

주문 테이블에서 apple 고객이 15개 이상 주문한 주문제품, 수량, 주문일자를 검색해보자.

결과 테이블

주문제품	수량	주문일자
1 p03		22 19/03/15

SELECT 문

주문 테이블에서 apple 고객이 주문했거나 15개 이상 주문된 제품의 주문제품, 수량, 주문일자, 주문고객을 검색해보자.

결과 테이블

	주문제품	수량	주문일자	주문고객
1	p03	10	19/01/01	apple
2	p06	45	19/01/11	banana
3	p06	36	19/02/20	melon
4	p01	19	19/03/02	banana
5	p03	22	19/03/15	apple
6	p02	50	19/04/10	pear
7	p04	15	19/04/11	banana
8	p03	20	19/05/22	carrot

SELECT 문

제품 테이블에서 단가가 2,000원 이상이면서 3,000원 이하인 제품의 제품명, 단가, 제조업체를 검색해보자.

결과 테이블

	제품명	단가	제조업체
1	콩떡파이	2600	한빛제과
2	맛난초콜릿	2500	한빛제과

17/55

SELECT 문

– LIKE를 이용한 검색

- LIKE 키워드를 이용해 부분적으로 일치하는 데이터를 검색
- 문자열을 이용하는 조건에만 LIKE 키워드 사용 가능

표 7-4 LIKE 키워드와 함께 사용할 수 있는 기호

기호	설명
%	0개 이상의 문자 (문자의 내용과 개수는 상관 없음)
_	1개의 문자 (문자의 내용은 상관 없음)

18/55

SELECT 문

표 7-5 LIKE 키워드의 사용 예

사용 예	설명
LIKE '데이터%'	데이터로 시작하는 문자열 (데이터로 시작하기만 하면 길이는 상관 없음)
LIKE '%데이터'	데이터로 끝나는 문자열 (데이터로 끝나기만 하면 길이는 상관 없음)
LIKE '%데이터%'	데이터가 포함된 문자열
LIKE '데이터 _ _ _'	데이터로 시작하는 6자 길이의 문자열
LIKE '_ _ 한%'	세 번째 글자가 '한'인 문자열

19/55

SELECT 문

고객 테이블에서 성이 김 씨인 고객의 고객이름, 나이, 등급, 적립금을 검색해보자.

결과 테이블

	고객이름	나이	등급	적립금
1	김선우	25	vip	2500
2	김용욱	22	silver	0

20/55

SELECT 문

고객 테이블에서 고객아이디가 5자인 고객의 고객아이디, 고객이름, 등급을 검색해보자.

결과 테이블

	고객아이디	고객이름	등급
1	apple	정소화	gold
2	melon	성원용	gold
3	peach	오형준	silver

21/55

SELECT 문

– NULL을 이용한 검색

- IS NULL 키워드를 이용해 특정 속성의 값이 널 값인지를 비교
- IS NOT NULL 키워드를 이용해 특정 속성의 값이 널 값이 아닌지를 비교
- 검색 조건에서 널 값은 다른 값과 크기를 비교하면 결과가 모두 거짓이 됨

22/55

SELECT 문

고객 테이블에서 나이가 아직 입력되지 않은 고객의 고객이름을 검색해보자.

결과 테이블

고객이름
1 오형준

23/55

SELECT 문

고객 테이블에서 나이가 이미 입력된 고객의 고객이름을 검색해보자.

결과 테이블

고객이름
1 정소화
2 김선우
3 고명석
4 김용욱
5 성원용
6 채광주

24/55

SELECT 문

– 정렬 검색

- ORDER BY 키워드를 이용해 결과 테이블 내용을 사용자가 원하는 순서로 출력
- ORDER BY 키워드와 함께 정렬 기준이 되는 속성과 정렬 방식을 지정
 - 오름차순(기본): ASC / 내림차순: DESC
 - 널 값은 오름차순에서는 맨 마지막에 출력되고, 내림차순에서는 맨 먼저 출력됨
 - 여러 기준에 따라 정렬하려면 정렬 기준이 되는 속성을 차례대로 제시

```
SELECT [ ALL | DISTINCT ] 속성_리스트
FROM   테이블_리스트
[ WHERE 조건 ]
[ ORDER BY 속성_리스트 [ ASC | DESC ] ];
```

SELECT 문

고객 테이블에서 고객이름, 등급, 나이를 검색하되, 나이를 기준으로 내림차순 정렬해보자.

결과 테이블

	고객이름	등급	나이
1	오형준	silver	(null)
2	성원용	gold	35
3	채광주	silver	31
4	고명석	gold	28
5	김선우	vip	25
6	김용욱	silver	22
7	정소화	gold	20

SELECT 문

주문 테이블에서 수량이 10개 이상인 주문의 주문고객, 주문제품, 수량, 주문일자를 검색해보자. 단, 주문제품을 기준으로 오름차순 정렬하고, 동일 제품은 수량을 기준으로 내림차순 정렬해보자.

결과 테이블

	주문고객	주문제품	수량	주문일자
1	banana	p01	19	19/03/02
2	pear	p02	50	19/04/10
3	apple	p03	22	19/03/15
4	carrot	p03	20	19/05/22
5	apple	p03	10	19/01/01
6	banana	p04	15	19/04/11
7	banana	p06	45	19/01/11
8	melon	p06	36	19/02/20

SELECT 문

– 집계 함수를 이용한 검색

- 특정 속성 값을 통계적으로 계산한 결과를 검색하기 위해 집계 함수를 이용
 - 집계 함수(aggregate function)
 - » 열 함수(column function)라고도 함
 - » 개수, 합계, 평균, 최댓값, 최솟값의 계산 기능을 제공
- 집계 함수 사용 시 주의 사항
 - 집계 함수는 널인 속성 값은 제외하고 계산함
 - 집계 함수는 WHERE 절에서는 사용할 수 없고, SELECT 절이나 HAVING 절에서만 사용 가능

SELECT 문

표 7-6 집계 함수

함수	의미	사용 가능한 속성의 타입
COUNT	속성 값의 개수	모든 데이터
MAX	속성 값의 최댓값	
MIN	속성 값의 최솟값	
SUM	속성 값의 합계	숫자 데이터
AVG	속성 값의 평균	

SELECT 문

제품 테이블에서 모든 제품의 단가 평균을 검색해보자.

결과 테이블

	AVG(단가)
1	2764.285714285714285714285714285714

SELECT 문

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짜면	2500	5500	민국푸드
p03	콩떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

AVG(단가)
2764

그림 7-7 모든 제품의 평균 단가를 계산하는 과정 : 제품 테이블

SELECT 문

한빛제과에서 제조한 제품의 재고량 합계를 제품 테이블에서 검색해보자.

결과 테이블

	재고량 합계
1	6500

SELECT 문

고객 테이블에 고객이 몇 명 등록되어 있는지 검색해보자.

▶▶ ❶ 고객아이디 속성을 이용해 계산하는 경우

결과 테이블

	고객수
1	7

❷ 나이 속성을 이용해 계산하는 경우

결과 테이블

	고객수
1	6

❸ *를 이용해 계산하는 경우

결과 테이블

	고객수
1	7

33/55

SELECT 문

널인 속성 값은 제외하고 개수 계산

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
pear	채광주	31	silver	회사원	500
peach	오형준	NULL	silver	의사	300

COUNT (고객아이디)

7

COUNT (나이)

6

그림 7-8 고객의 수를 계산하는 과정 : 고객 테이블

34/55

SELECT 문

정확한 개수를 계산하기 위해서는 보통 기본키 속성이나 *를 주로 이용

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
pear	채광주	31	silver	회사원	500
peach	오형준	NULL	silver	의사	300

그림 7-9 COUNT(*)로 개수를 계산하는 과정 : 고객 테이블

35/55

SELECT 문

제품 테이블에서 제조업체의 수를 검색해보자.

결과 테이블

	제조업체 수
1	3

DISTINCT 키워드를 이용해 중복을 없애고 서로 다른 제조업체의 개수만 계산

36/55

SELECT 문

- 그룹별 검색

```
SELECT  [ ALL | DISTINCT ] 속성_리스트
FROM    테이블_리스트
[ WHERE 조건 ]
[ GROUP BY 속성_리스트 [ HAVING 조건 ] ]
[ ORDER BY 속성_리스트 [ ASC | DESC ] ];
```

- GROUP BY 키워드를 이용해 특정 속성의 값이 같은 튜플을 모아 그룹을 만들고, 그룹별로 검색
 - GROUP BY 키워드와 함께 그룹을 나누는 기준이 되는 속성을 지정
- HAVING 키워드를 함께 이용해 그룹에 대한 조건을 작성
- 그룹을 나누는 기준이 되는 속성을 SELECT 절에도 작성하는 것이 좋음

SELECT 문

주문 테이블에서 주문제품별 수량의 합계를 검색해보자.

결과 테이블

	주문제품	총주문수량
1	p03	52
2	p02	58
3	p06	81
4	p04	15
5	p01	24

SELECT 문

주문제품	수량
p03	10
p01	5
p06	45
p02	8
p06	36
p01	19
p03	22
p02	50
p04	15
p03	20

주문제품	총주문수량
1 p03	52
2 p02	58
3 p06	81
4 p04	15
5 p01	24

동일 제품을 주문한 튜플을 모아 그룹으로 만들고, 그룹별로 수량의 합계를 계산

그림 7-10 주문제품별 수량의 합계를 계산하는 과정

SELECT 문

제품 테이블에서 제조업체별로 제조한 제품의 개수와 제품 중 가장 비싼 단가를 검색하되, 제품의 개수는 제품수라는 이름으로 출력하고 가장 비싼 단가는 최고가라는 이름으로 출력해 보자.

결과 테이블

	제조업체	제품수	최고가
1	대한식품	2	4500
2	민국푸드	2	5500
3	한빛제과	3	2600

SELECT 문

제품 테이블에서 제품을 3개 이상 제조한 제조업체별로 제품의 개수와, 제품 중 가장 비싼 단가를 검색해보자.

결과 테이블

	제조업체	제품수	최고가
1	한빛제과	3	2600

집계 함수를 이용한 조건은 WHERE 절에는 작성할 수 없고 HAVING 절에서 작성

41/55

SELECT 문

고객 테이블에서 적립금 평균이 1,000원 이상인 등급에 대해 등급별 고객수와 적립금 평균을 검색해보자.

결과 테이블

	등급	고객수	평균적립금
1	gold	3	3500
2	vip	1	2500

42/55

SELECT 문

주문 테이블에서 각 주문고객이 주문한 제품의 총주문수량을 주문제품별로 검색해보자.

결과 테이블

	주문제품	주문고객	총주문수량
1	p02	carrot	8
2	p01	banana	19
3	p06	melon	36
4	p03	apple	32
5	p01	melon	5
6	p02	pear	50
7	p03	carrot	20
8	p06	banana	45
9	p04	banana	15

← 하나의 그룹

43/55

SELECT 문

– 여러 테이블에 대한 조인 검색

- 조인 검색: 여러 개의 테이블을 연결하여 데이터를 검색하는 것
- 조인 속성: 조인 검색을 위해 테이블을 연결해주는 속성
 - 연결하려는 테이블 간에 조인 속성의 이름은 달라도 되지만 도메인은 같아야 함
 - 일반적으로 외래키를 조인 속성으로 이용함
- FROM 절에 검색에 필요한 모든 테이블을 나열
- WHERE 절에 조인 속성의 값이 같아야 함을 의미하는 조인 조건을 제시
- 같은 이름의 속성이 서로 다른 테이블에 존재할 수 있기 때문에 속성 이름 앞에 해당 속성이 소속된 테이블의 이름을 표시
 - 예) 주문.주문고객

44/55

SELECT 문

주문 테이블					제품 테이블				
주문번호	주문고객	주문제품	수량	배송지	주문일자	제품번호	제품명	재고량	단가
o01	apple	p03	10	서울시 마포구	2019-01-01	p01	그냥만두	5000	4500
o02	melon	p01	5	인천시 계양구	2019-01-10	p02	매운쫄면	2500	5500
o03	banana	p06	45	경기도 부천시	2019-01-11	p03	콩떡파이	3600	2600
o04	carrot	p02	8	부산서 금정구	2019-02-01	p04	맛난초콜릿	1250	2500
o05	melon	p06	36	경기도 용인시	2019-02-20	p05	얼큰라면	2200	1200
o06	banana	p01	19	충청북도 보은군	2019-03-02	p06	통통우동	1000	1550
o07	apple	p03	22	서울시 영등포구	2019-03-15	p07	달콤비스킷	1650	1500
o08	pear	p02	50	강원도 춘천시	2019-04-10				
o09	banana	p04	15	전라남도 목포시	2019-04-11				
o10	carrot	p03	20	경기도 안양시	2019-05-22				

그림 7-11 2개의 테이블을 이용한 조인 검색 예 : 주문과 제품 테이블

45/55

SELECT 문

판매 데이터베이스에서 banana 고객이 주문한 제품의 이름을 검색해보자.

결과 테이블	제품명
1	그냥만두
2	맛난초콜릿
3	통통우동

46/55

SELECT 문

판매 데이터베이스에서 나이가 30세 이상인 고객이 주문한 제품의 주문제품과 주문일자를 검색해보자.

결과 테이블	주문제품	주문일자
1	p01	19/01/10
2	p06	19/02/20
3	p02	19/04/10

47/55

SELECT 문

```
SELECT  주문제품, 주문일자
FROM    고객 c, 주문 o
WHERE   c.나이 >= 30 AND o.주문고객 = c.고객아이디;
```

테이블의 이름을 대신하는 단순한 별명을 제시하여
질의문을 작성하는 것도 좋다.

이름이 같은 속성이 없다면
테이블 이름 없이 속성 이름으로만 작성해도 된다.

48/55

SELECT 문

판매 데이터베이스에서 고명석 고객이 주문한 제품의 제품명을 검색해보자.

결과 테이블

	제품명
1	매운쫄면
2	통짜파이

49/55

UPDATE 문

- 테이블에 저장된 튜플에서 특정 속성의 값을 수정

```
UPDATE 테이블_이름
SET 속성_이름1 = 값1, 속성_이름2 = 값2, ...
[WHERE 조건];
```

- SET 키워드 다음에 속성 값을 어떻게 수정할 것인지를 지정
- WHERE 절에 제시된 조건을 만족하는 레코드들에 대해서만 속성 값을 수정

• WHERE 절을 생략하면 테이블에 존재하는 모든 레코드를 대상으로 수정

50/55

UPDATE 문

제품 테이블에서 제품번호가 p03인 제품의 제품명을 통큰파이로 수정해보자.

	제품번호	제품명	재고량	단가	제조업체
1	p01	그냥만두	5000	4500	대한식품
2	p02	매운쫄면	2500	5500	민국푸드
3	p03	통큰파이	3600	2600	한빛제과
4	p04	맛난초콜릿	1250	2500	한빛제과
5	p05	얼큰라면	2200	1200	대한식품
6	p06	통통우동	1000	1550	민국푸드
7	p07	달콤비스킷	1650	1500	한빛제과

51/55

UPDATE 문

제품 테이블에 있는 모든 제품의 단가를 10% 인상해보자. 그런 다음 제품 테이블의 모든 내용을 검색하여 인상 내용을 확인해보자.

	제품번호	제품명	재고량	단가	제조업체
1	p01	그냥만두	5000	4950	대한식품
2	p02	매운쫄면	2500	6050	민국푸드
3	p03	통큰파이	3600	2860	한빛제과
4	p04	맛난초콜릿	1250	2750	한빛제과
5	p05	얼큰라면	2200	1320	대한식품
6	p06	통통우동	1000	1705	민국푸드
7	p07	달콤비스킷	1650	1650	한빛제과

52/55

DELETE 문

- 테이블에 저장된 데이터를 삭제

```
DELETE
FROM 테이블_이름
[WHERE 조건];
```

- WHERE 절에 제시한 조건을 만족하는 레코드만 삭제

- WHERE 절을 생략하면 테이블에 존재하는 모든 레코드를 삭제해 빈 테이블이 됨

DELETE 문

주문 테이블에서 주문일자가 2019년 5월 22일인 주문내역을 삭제해보자. 그런 다음 주문 테이블의 모든 내용을 검색하여 삭제 여부를 확인해보자.

결과 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
1 o01	apple	p03	5	서울시 마포구	19/01/01
2 o02	melon	p01	5	인천시 계양구	19/01/10
3 o03	banana	p06	45	경기도 부천시	19/01/11
4 o04	carrot	p02	8	부산시 금정구	19/02/01
5 o05	melon	p06	36	경기도 용인시	19/02/20
6 o06	banana	p01	19	충청북도 보은군	19/03/02
7 o07	apple	p03	5	서울시 영등포구	19/03/15
8 o08	pear	p02	50	강원도 춘천시	19/04/10
9 o09	banana	p04	15	전라남도 목포시	19/04/11