



응용 SW 기초 활용 기술 part 2

데이터 조작하기



한국기술교육대학교
온라인평생교육원



학습목표



- 데이터의 검색
- 데이터의 삽입



- 테이블에 저장되어 있는 데이터를 원하는 조건에 따라 검색할 수 있다.
- 테이블에 데이터를 저장할 수 있다.



데이터의 검색

1 데이터의 검색

데이터베이스를 구축한 후, 데이터를 이용하여 의미 있는 정보를 얻기 위해 사용하는 기능

데이터 검색을 위해서는 SQL의 **SELECT** 문 이용

- SELECT 문은 테이블 내에 저장된 데이터를 검색하여 결과에 따라 하나 또는 그 이상의 결과집합을 돌려줌



데이터의 검색

1 데이터의 검색

1) 기본 검색

1 원하는 열의 정보만 검색

■ SELECT 열1, 열2, ...FROM 테이블명;

예 SELECT name,address FROM student;

■ 검색을 원하는 열의 이름을 적으며, 테이블에 저장된 열의 이름과 순서가 동일하지 않아도 됨

2 모든 열의 정보 검색

■ SELECT * FROM 테이블명;

예 SELECT * FROM student;

3 DISTINCT

■ 해당 열의 중복되는 데이터를 제거하여 한 번씩만 출력되도록 함

예 SELECT DISTINCT dep_code FROM student;



데이터의 검색

1 데이터의 검색

3) 조건 검색

테이블에 저장된 데이터를 원하는 조건으로 검색



WHERE 절 이용

테이블에 저장되어 있는 행들 중
조건과 일치하는 행만을 추출할 수 있음

SELECT 열1, 열2, ... FROM 테이블명 WHERE 조건;

- 연산자를 이용하여 조건을 제시
- 숫자, 문자, 날짜 값의 비교 가능



숫자 값은 그냥 입력해도 되지만, **문자와 날짜는 작은따옴표로 감싸주어야 함**

WHERE 절의 비교 조건을 위한 연산자

연산자	설명
=	같다
<>	같지 않다(SQL 버전에 따라 != 로 사용하기도 함)
>	크다
<	작다
>=	크거나 같다
<=	작거나 같다



SELECT std_code,name FROM student WHERE dep_code='061';



데이터의 검색

1 데이터의 검색

3) 조건 검색

1 LIKE를 이용한 패턴 검색

- 부분적으로 일치하는 데이터를 검색
- SELECT 열1, 열2, ... FROM 테이블명 WHERE 열 이름 LIKE 패턴;
 - % : 0개 이상의 문자를 의미하며, 문자의 개수와 상관없이 없음
 - _ : 하나의 문자를 의미
 - ➡ %와 _를 결합하여 사용 가능
- LIKE 다음 사용할 수 있는 패턴의 예

LIKE 패턴의 예	설명
LIKE 'a%'	a로 시작하는 문자열
LIKE '%a'	a로 끝나는 문자열
LIKE '%or%'	위치와 상관없이 or을 포함하는 문자열
LIKE '_r%'	문자열의 두 번째가 r인 문자열
LIKE 'a_%_ %'	a로 시작하여 최소 3개의 문자를 가지는 문자열
LIKE 'a%o'	a로 시작하여 o로 끝나는 문자열

예 SELECT std_code,name FROM student WHERE name LIKE '이%';

2 BETWEEN을 이용한 범위 검색

- SELECT 열1, 열2, ... FROM 테이블명 WHERE 열 이름 BETWEEN 값1 AND 값2;
 - 값1과 값2 사이의 값을 가지는 데이터 검색
- 예** SELECT std_code,name FROM student WHERE birthday BETWEEN '1995-01-01' AND '1995-03-31';



데이터의 검색

1 데이터의 검색

3) 조건 검색

3 NULL값을 이용한 검색

- 특정 열에 값이 입력되지 않아 NULL값을 가지는 데이터 검색

- SELECT 열1, 열2... FROM 테이블명 WHERE 열 이름 IS NULL;

예 SELECT std_code,name,birthday FROM student
WHERE birthday IS NULL;

- 특정 열에 값이 입력된 데이터 검색

- SELECT 열1, 열2... FROM 테이블명 WHERE 열 이름 IS NOT NULL;

예 SELECT std_code,name,birthday FROM student
WHERE birthday IS NOT NULL;

4 논리를 이용한 검색

AND

- 나열하는 조건들이 모두 만족할 경우 검색

- SELECT 열1, 열2, ... FROM 테이블명 WHERE 조건1
AND 조건2 AND 조건3 ...;

예 SELECT std_code,name,dep_code FROM student
WHERE name LIKE '이%' AND dep_code='062';

OR

- 나열하는 조건들 중 하나만 만족해도 검색

- SELECT 열1, 열2, ... FROM 테이블명 WHERE 조건1
OR 조건2 OR 조건3 ...;

예 SELECT std_code,name,dep_code FROM student
WHERE name LIKE '이%' OR dep_code='062';

NOT

- 조건이 만족하지 않을 경우 검색

- SELECT 열1, 열2, ... FROM 테이블명 WHERE NOT 조건;

예 SELECT std_code,name,dep_code FROM student
WHERE name NOT LIKE '이%';



데이터의 검색

1 데이터의 검색

3) 조건 검색

5 ORDER BY를 이용한 정렬 검색

- 검색된 데이터를 정렬하여 출력
- SELECT 열1, 열2, ... FROM 테이블명 ORDER BY 열 이름... ASC|DESC;
 - ASC : 오름차순 정렬
 - DESC : 내림차순 정렬

예 SELECT std_code,name,dep_code FROM student
WHERE std_code LIKE '17%'
ORDER BY dep_code DESC;

6 함수를 이용한 정렬 검색

- 결과값의 통계를 위해 함수 사용 가능
- SELECT 함수(열 이름) FROM 테이블명;
 - WHERE 절은 선택 사항
 - 집계 함수(Aggregate Function)

함수	설명
COUNT()	결과값의 개수
MIN()	결과값 중 최소값
MAX()	결과값 중 최대값
SUM()	결과값의 합계
AVG()	결과값의 평균

예 SELECT MAX(name) AS 학생이름 FROM student;
SELECT COUNT(name) AS 학생 수 FROM student WHERE std_code LIKE '17%';



데이터의 검색

1 데이터의 검색

3) 조건 검색

7

GROUP BY를 이용한 그룹별 검색

- 데이터 검색을 위한 SELECT문의 형식

SELECT DISTINCT 열 이름들 FROM 테이블명

WHERE 조건

GROUP BY 열 이름 HAVING 조건

ORDER BY 열 이름 ASC|DESC

8

테이블을 연결하여 검색

- SELECT 테이블.열 이름들 FROM 테이블명1, 테이블명2;
- 'WHERE 조건'은 선택사항



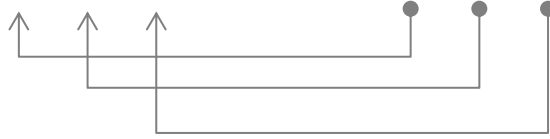
예 SELECT student.std_code, student.name, department.dep_name
FROM student, department;



데이터의 삽입

1 데이터의 삽입

INSERT INTO 테이블명 (열1, 열2, 열3, ...) VALUES (값1, 값2, 값3, ...);



- 데이터가 있는 열의 이름과 값만을 삽입

예 INSERT INTO student (std_code,name,address,dep_code)
VALUES('17061001','홍길동','충남 천안시','061');

INSERT INTO 테이블명 VALUES (값1, 값2, 값3);

- 모든 행의 값을 삽입
- 테이블에서 정의한 열의 순서와 입력하는 값의 순서가 일치해야 함

예 INSERT INTO department VALUES ('061','041-234-1111','정보통신공학과');

데이터의 검색

+ 기본 검색

- 'SELECT 열1, 열2, ...FROM 테이블명;' 또는 'SELECT * FROM 테이블명;'을 이용하여 테이블에 저장되어 있는 데이터를 검색할 수 있음

+ 조건 검색

- WHERE 절을 이용하여 검색할 데이터의 조건을 기술할 수 있으며, 조건의 기술을 위해 비교연산자, LIKE, BETWEEN...AND..., AND, OR, NOT, IS NULL, IS NOT NULL 등을 사용할 수 있음
- ORDER BY 절을 이용하여 특정 열을 기준으로 정렬할 수 있으며, GROUP BY 절을 이용하여 그룹별 검색을 할 수 있음
- 외래 키를 이용하여 연결된 테이블 간에 데이터를 연결하여 검색할 수 있음

데이터의 삽입

+ 데이터의 삽입

- 'INSERT INTO 테이블명 (열1, 열2, 열3, ...) VALUES (값1, 값2, 값3, ...);'을 이용하여 원하는 열의 값만을 삽입할 수 있음
- 'INSERT INTO 테이블명 VALUES (값1, 값2, 값3);'를 이용하여 열의 이름을 생략하고 모든 열의 값을 삽입할 수 있음



+ 데이터의 검색 관련 구문

기능	구문
원하는 열의 정보만 검색	SELECT 열1, 열2, ...FROM 테이블명;
모든 열의 정보 검색	SELECT * FROM 테이블명;
테이블에 저장된 데이터를 원하는 조건으로 검색	SELECT 열1, 열2, ... FROM 테이블명 WHERE 조건;