

2. 데이터 검색

컴퓨터공학부

김은경

[학습 목표]

1. SQL 문장의 작성 규칙을 나열할 수 있다.
2. 데이터 검색의 세가지 유형을 설명할 수 있다.
3. 테이블에서 특정 열(속성)만 검색할 수 있다.
4. 열에 별칭(Alias)을 부여할 수 있다.
5. 검색 시 중복되는 행(튜플)을 제거할 수 있다.

1. SQL 언어 소개

★ SQL이란?

- 관계형 DBMS의 표준 언어

★ SQL문의 종류

분류	특징	해당 명령어
데이터 정의어 (DDL)	새로운 데이터의 정의 및 수정, 삭제를 위한 명령어	CREATE ALTER DROP RENAME TRUNCATE
데이터 조작어 (DML)	데이터의 검색, 삽입, 삭제, 수정을 위한 명령어	SELECT INSERT UPDATE DELETE
데이터 제어어 (DCL)	시스템 및 객체와 관련된 여러 가지 권한을 부여 및 철회하기 위한 명령어	GRANT REVOKE
트랜잭션 제어어	모든 미결정된 데이터 조작을 영구적으로 변경하거나 취소하여 트랜잭션을 완료하기 위한 명령어	COMMIT ROLLBACK SAVEPOINT

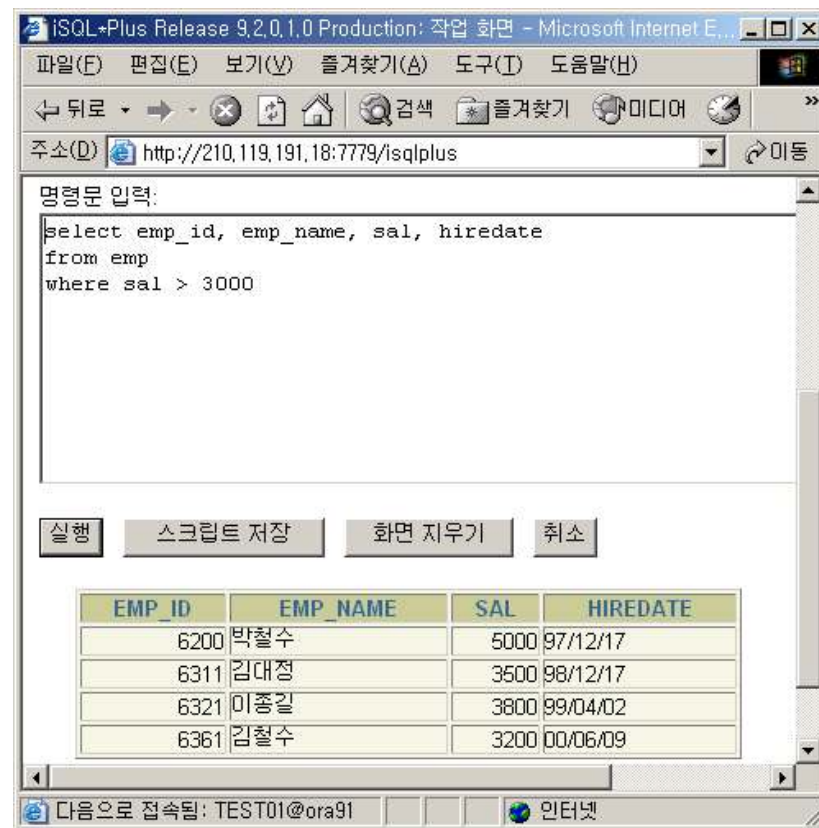
1. SQL 언어 소개

★ SQL 문장 작성 규칙

주요 요소	작성 규칙
대소문자	구분하지 않음. 단, 인용부호 안에서는 구분됨 (출력 시에는 모두 대문자로 표시됨)
한 문장의 길이	한 문장을 한 줄 이상에 걸쳐서 작성할 수 있음
예약어 단축	단축해서 사용하거나 줄을 나누어 쓸 수 없음
절(Claue)	가능하면 줄을 바꾸어 쓰는 것이 읽기에 좋다.
문자의 끝	세미콜론(;)으로 표시한다. (단, iSQL* Plus에서는 생략 가능)
가독성 (Readability)	탭(Tab)과 들여쓰기(Indent)를 적절하게 사용하는 것이 좋음
SQL* Plus	SQL 프롬프트 뒤에 SQL문 입력하고, 세미콜론 뒤에서 엔터키를 누르면 실행됨
iSQL* Plus	작업 화면의 "명령문 입력" 영역에 SQL문 입력하고, '실행' 버튼 클릭해서 실행시킴

1. SQL 언어 소개

[참고] SQL* Plus와 iSQL* Plus 화면 비교



1. SQL 언어 소개

★ SQL 문장 관련 용어 정의

- ① 예약어(Keyword) - 기능이 정해져 있는 독립적인 SQL 요소를 의미한다.
예) SELECT, FROM 등
- ② 절(Clause) - 하나의 예약어를 포함하는 SQL 문장의 한 부분을 의미한다.
예) SELECT id, ename
- ③ 문장(Statement) - 둘 이상의 절이 조합된 것을 뜻한다.
예) SELECT * FROM emp

1. SQL 언어 소개

★ SQL 문의 기본 출력 형태

- ① 기본 문자 출력 형태 : 대문자
- ② 기본 자리 맞춤
 - a) 날짜와 문자 데이터 : 왼쪽 정렬
 - b) 숫자 데이터 : 오른쪽 정렬

예) SELECT emp_id,
emp_name,
hiredate
FROM emp;

EMP_ID	EMP_NAME	HIREDATE
6200	박철수	17/12/17
6311	김대정	18/12/17
7489	민동규	19/02/20
7522	정종철	18/02/22
6321	이종길	19/04/02
6351	진대준	20/05/31
7910	이영희	20/09/01
6361	김철수	20/06/09
7878	백기수	20/12/09
7854	진영진	20/09/08
7872	이문정	21/01/12
7920	김마리아	20/12/03
7901	정동길	20/12/03
7933	김철수	21/01/23

14 rows selected.

2. 데이터 검색

★ 데이터 검색의 3가지 유형

① 셀렉션(Selection)

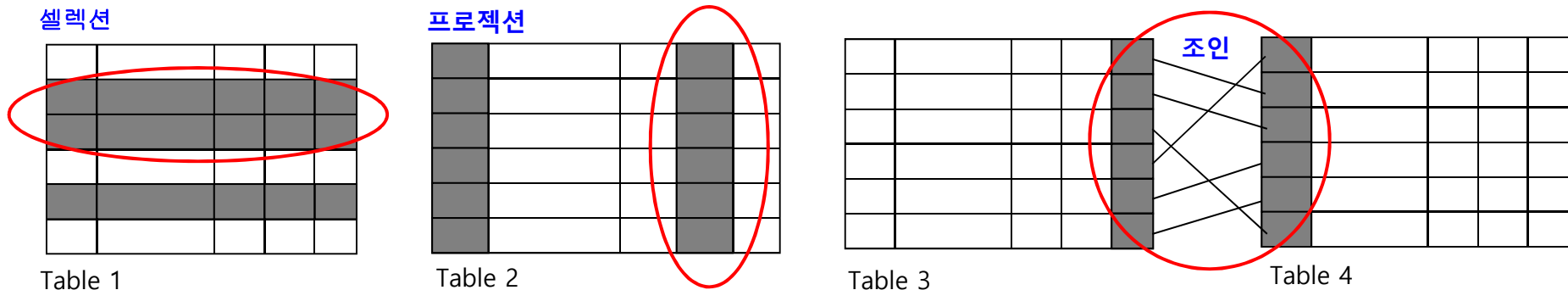
- 테이블에서 검색 조건이 만족되는 특정한 행(Row)을 검색하는 방법

② 프로젝션(Projection)

- 테이블에서 검색 조건이 만족되는 특정 열(Column)을 검색하는 방법

③ 조인(Join)

- 하나 이상의 테이블에 있는 특정 열을 서로 연관시켜, 하나 이상의 테이블에서 필요한 데이터를 검색하는 방법



2. 데이터 검색

★ 기본 SELECT 문의 형식

```
SELECT [DISTINCT] { * | 열이름 [별칭], ... }  
FROM   테이블_이름;
```

- 1) SELECT 절 : 출력시킬 하나 이상의 열을 나열한다.
- 2) DISTINCT : 열의 값이 중복되면 제거 시킬 것을 지시하는 예약어이다.
- 3) * : 모든 열을 선택함을 표시하는 와일드카드(wildcard) 문자이다.
- 4) 열이름 : 하나 이상의 선택할 열 이름을 기술한다.
- 5) 별칭(Alias) : 선택된 열에 부여할 다른 이름, 즉 별칭을 기술한다.
- 6) FROM 절 : 나열한 열을 포함하는 테이블 이름을 명시한다.

[참고] SQL문 작성 시 사용하는 기호들

- 수직바(|) : 둘 중 하나를 기술하라는 의미함
즉, '*' 나 열이름 가운데 하나를 기술해야 함
- 중괄호({ }) : 값의 리스트(list), 즉 여러 값을 나열함을 의미함
- 대괄호([]) : 생략 가능함(optional)을 의미함

2. 데이터 검색

★ 모든 열 검색하기

- 어떤 테이블에 있는 데이터의 모든 열을 검색하려면, 특정 열 이름을 나열하는 대신 '*'를 지정하면 됨
- 예)

```
SELECT *  
FROM dept;
```

DEPT_ID	DEPT_NAME	DEPT_LOC
100	관리부	서울
200	연구부	대전
300	홍무부	서울
400	영업부	인천
500	개발부	천안

- 예) 조건을 만족하는 데이터의 모든 열 검색하기

```
SELECT *  
FROM dept  
WHERE dept_id = 100;
```

DEPT_ID	DEPT_NAME	DEPT_LOC
100	관리부	서울

2. 데이터 검색

[참고] 테이블 구조 확인 명령어

- `describe` 또는 줄여서 `desc`라는 명령을 사용하면 테이블의 구조를 확인할 수 있음
- 예) `desc dept;`

Name	Null?	Type
DEPT_ID	NOT NULL	NUMBER(3)
DEPT_NAME		VARCHAR2(20)
DEPT_LOC		VARCHAR2(15)

2. 데이터 검색

★ 특정 열만 검색하기 : 프로젝션

- 한 테이블에 저장된 데이터의 특정 열만을 검색하려면, **검색을 원하는 열의 이름을 콤마로 구분하여 나열함**
- 예)

```
SELECT dept_name, dept_loc  
FROM dept;
```

DEPT_NAME	DEPT_LOC
관리부	서울
연구부	대전
총무부	서울
영업부	인천
개발부	천안

3. 검색 결과 가공 방법

★ 산술식을 활용한 검색 결과 가공

구분	설명
산술 연산자의 종류	사칙 연산자(+, -, *, /) 와 괄호 ()
산술식 (Arithmetic Expression)	열 이름과 숫자 상수 값 및 산술 연산자로 구성되는 수식
산술 연산자의 우선순위	괄호 > 곱셈과 나눗셈(*, /) > 덧셈과 뺄셈(+, -)
산술 연산자의 사용	SELECT 문의 FROM 절을 제외한 나머지 절에서 사용 가능

3. 검색 결과 가공 방법

✓ 산술식 활용 예

```
SELECT emp_name, sal, (sal+1000) * 12  
FROM emp;
```

EMP_NAME	SAL	(SAL+1000)*12
박철수	9000	120000
김대정	6200	86400
민동규	3700	56400
정종철	4520	66240
이종길	7000	96000
진대준	5850	82200
이영희	3200	50400
김철수	6500	90000
백기수	5000	72000
진영진	3500	54000
이문정	2500	42000
김마리아	2650	43800
정동길	4500	66000
김철수	2500	42000

14 rows selected.

3. 검색 결과 가공 방법

★ 널(Null)을 활용한 검색 결과 가공

구분	설명
Null이란?	이용할 수 없거나, 지정되지 않았거나, 알 수 없는 모든 값을 일컫는 예약어
Null의 특징	<ul style="list-style-type: none">- 숫자 0(Zero)이나 공백(Blank) 문자와는 다름- 특정 열에 대한 값이 없는 경우, 그 값은 Null이 됨
Null과 숫자 0의 차이점	출력 시에 Null은 공백으로 표시되고, 숫자 0은 0으로 표시됨
Null을 포함하는 산술식	<ul style="list-style-type: none">- Null 값을 포함하는 산술식은 무조건 Null로 계산됨- Null을 0으로 나누면 오류가 발생하고, 숫자로 나누면 결과값은 Null 또는 Unknown(알 수 없음)이 된다.

3. 검색 결과 가공 방법

✓ Null을 포함한 산술식 출력 예

```
SELECT emp_name, sal, bonus, sal+bonus
FROM emp;
```

EMP_NAME	SAL	BONUS	SAL+BONUS
박철수	9000		
김대정	6200		
민동규	3700	200	3900
정종철	4520	300	4820
이종길	7000	500	7500
진대준	5850		
이영희	3200		
김철수	6500		
백기수	5000		
진영진	3500	0	3500
이문정	2500		
김마리아	2650		
정동길	4500		
김철수	2500		

Null 값을 포함하는 산술식은 무조건 Null로 계산됨

14 rows selected.

3. 검색 결과 가공 방법

✓ NVL 함수를 활용해서 널 값 변환하기

→ 함수의 첫 번째 인수에 포함된 널 값을 두 번째 인수로 변환함

예)

```
SELECT emp_name, sal, bonus, sal + NVL(bonus, 0)
FROM emp;
```

EMP_NAME	SAL	BONUS	SAL+NVL(BONUS,0)
박철수	9000		9000
김대정	6200		6200
민동규	3700	200	3900
정종철	4520	300	4820
이종길	7000	500	7500
진대준	5850		5850
이영희	3200		3200
김철수	6500		6500
백기수	5000		5000
진영진	3500	0	3500
이문정	2500		2500
김마리아	2650		2650
정동길	4500		4500
김철수	2500		2500

14 rows selected.

3. 검색 결과 가공 방법

★ 열 별칭(alias)을 활용한 검색 결과 가공

구분	설명
열 별칭이란?	열 헤딩(Heading) 즉, 열의 제목에 부여하는 새로운 이름
열 별칭 정의 방법	<ul style="list-style-type: none">① 공백을 구분자(Separator)로 사용하여 열 이름 바로 뒤에 정의함② 열 이름과 별칭 사이에 예약어 'AS'를 기술하여 구분할 수 있음 예) SELECT sal AS salary, ...③ 별칭에 공백이나 특수 문자가 포함되거나, 대소문자를 구분하고 싶으면, 이중 인용부호(" ")로 묶어서 기술해야 함
열 별칭 사용의 장점	<ul style="list-style-type: none">① 열 헤딩을 열 이름이 아닌 보다 명확한 내용으로 수정할 수 있음② 산술식에 별칭을 부여하여 열 헤딩을 간소화 시킬 수 있음③ 열 헤딩에서 대소문자를 구분할 수 있음

3. 검색 결과 가공 방법

✓ 열 별칭 사용 예

```
SELECT dept_name Department, dept_loc as Location
FROM dept;
```

DEPARTMENT	LOCATION
관리부	서울
연구부	대전
총무부	서울
영업부	인천
개발부	천안

```
SELECT dept_name "Department", dept_loc as "Location"
FROM dept;
```

Department	Location
관리부	서울
연구부	대전
총무부	서울
영업부	인천
개발부	천안

3. 검색 결과 가공 방법

✓ 산술식에 열 별칭 사용 예

```
SELECT emp_name "Name", sal + NVL(bonus, 0) "Annual Salary"  
FROM emp;
```

Name	Annual Salary
박철수	9000
김대정	6200
민동규	3900
정종철	4820
이종길	7500
진대준	5850
이영희	3200
김철수	6500
백기수	5000
진영진	3500
이문정	2500
김마리아	2650
정동길	4500
김철수	2500

14 rows selected.

3. 검색 결과 가공 방법

★ 연결 연산자를 활용한 검색 결과 가공

구분	설명
연결 연산자란?	열이나 문자열(String)을 다른 열과 연결할 때 사용하는 연산자로, 두 개의 수직 바()로 표시함
연결 연산자 사용 방법	<ul style="list-style-type: none">① 연결할 열 이름이나 문자열 사이에 연결 연산자를 기술함② 하나의 연결된 문자 표현식을 만드는데 하나 이상의 연결 연산자를 사용해도 무방함③ 문자열은 반드시 단일 인용부호(' ')로 묶어서 표시해야 함 예) select '이름은' emp_name, ...
연결 연산자 사용의 장점	여러 개의 열이나 문자열을 연결하여 하나의 문자 표현식을 생성할 수 있음

3. 검색 결과 가공 방법

✓ 단일 연결 연산자 사용 예

```
SELECT emp_name || job as employee  
FROM emp;
```

EMPLOYEE
박철수대표이사
김대정부장
민동규세일즈
정종철세일즈
이종길부장
진대준부장
이영희경리
김철수부장
백기수연구직
진영진세일즈
이문정사무직
김마리아사무직
정동길연구직
김철수사무직

14 rows selected.

3. 검색 결과 가공 방법

✓ 다중 연결 연산자 사용 예

```
SELECT emp_name || '의 담당 업무는 ' || job as "사원과 담당 업무"  
FROM emp;
```

사원과 담당 업무
박철수의 담당 업무는 대표이사
김대정의 담당 업무는 부장
민동규의 담당 업무는 세일즈
정종철의 담당 업무는 세일즈
이종길의 담당 업무는 부장
진대준의 담당 업무는 부장
이영희의 담당 업무는 경리
김철수의 담당 업무는 부장
백기수의 담당 업무는 연구직
진영진의 담당 업무는 세일즈
이문정의 담당 업무는 사무직
김마리아의 담당 업무는 사무직
정동길의 담당 업무는 연구직
김철수의 담당 업무는 사무직

14 rows selected.

3. 검색 결과 가공 방법

★ 중복 행 제거 기능(Distinct)을 활용한 검색 결과 가공

구분	설명
중복 행 제거란?	검색된 결과에서 중복되는 내용을 제거하는 것을 의미함
Distinct란?	<ul style="list-style-type: none">- 검색 시에 중복되는 행을 제거하기 위해 사용하는 옵션을 나타내는 예약어- SELECTR 문에서 예약어 SELECT 뒤, 열 이름 앞에 기술해야 함
연결 연산자 사용의 장점	여러 개의 열이나 문자열을 연결하여 하나의 문자 표현식을 생성할 수 있음

3. 검색 결과 가공 방법

✓ 단일 중복 행의 제거 예

```
SELECT  DISTINCT dept_id  
FROM    emp;
```

DEPT_ID	
	100
	200
	300
	400

비교) SELECT dept_id
FROM emp;

DEPT_ID	
	200
	100
	400
	400
	400
	300
	300
	200
	200
	400
	100
	300
	200

3. 검색 결과 가공 방법

✓ 다중 중복 행의 제거 예

```
SELECT  DISTINCT dept_id, job
FROM    emp
ORDER BY dept_id;
```

두 열들의 값이 조합된 상태
에서 중복을 제거함

DEPT_ID	JOB
100	부장
100	사무직
200	대표이사
200	부장
200	사무직
200	연구직
300	경리
300	부장
300	사무직
400	부장
400	세일즈
	연구직

12 rows selected.



Q & A