데이터조회

자바 강의실

목차

1. SELECT 문

- SELECT문
  - 관계 데이터베이스에서 정보를 검색하는 SQL문
  - 관계 데이터베이스에서 가장 자주 사용됨
  - 여러 가지 질의들의 결과를 보이기 위해서 그림의 관계 데이터베이스 상태를 사용함

1. SELECT 문

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	JOB_ID	MANAGER_ID	SALARY	DEPARTMENTS_ID
2106	김창섭	대리	1077	2500000	2
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
1003	조민희	과장	4377	3000000	2
3427	최종철	사원	3011	1500000	3
1365	김상원	사원	3426	1500000	1
4377	이성래	사장	^	5000000	2

DEPARTMENTS

DEPARTMENTS_ID	DEPARTMENTS_NAME	LOCATION_ID
1	영업	8
2	기획	10
3	개발	9
4	총무	7

[그림] 관계 데이터베이스 상태

1. SELECT 문

CATEGORY

```
CREATE TABLE CATEGORY (
    KIND NUMBER(2) PRIMARY KEY,
    NAME VARCHAR2(100) NOT NULL,
    DETAIL VARCHAR2(1000) NULL
);

INSERT INTO CATEGORY
(KIND, NAME, DETAIL) VALUES
(10, '의류', '모든의료제품');
```

[그림] 카테고리관련 테이블 생성 및 샘플데이터

1. SELECT 문

PRODUCT

```
CREATE TABLE PRODUCT (
    PSEQ NUMBER(5) PRIMARY KEY,
    NAME VARCHAR2(100) NOT NULL,
    KIND NUMBER(2) NOT NULL,
    PRICE1 NUMBER(10) NOT NULL,
    PRICE2 NUMBER(10) NOT NULL,
    CONTENT VARCHAR2(1000) NULL,
    USEYN CHAR(1) DEFAULT 'y',
    BESTYN CHAR(1) DEFAULT 'n',
    REGDATE DATE DEFAULT SYSDATE
);

INSERT INTO PRODUCT
(PSEQ, NAME, KIND, PRICE1, PRICE2, COTENT)
VALUES (1, '청바지', '10', 50000, 40000, '블랙진 청바지');
```

[그림] 상품관련 테이블 생성 및 샘플데이터

1. SELECT 문

- 기본적인 SQL 질의
  - SELECT절과 FROM절만 필수적인 절이고, 나머지는 선택 사항

```
SELECT      [DISTINCT] 애트리뷰트(들)                    (1)
FROM        릴레이션(들)                                 (2)
[WHERE      조건                                         (3)
                  [중첩 질의] ]                          (4)
[GROUP BY   애트리뷰트(들)]                              (5)
[ HAVING          조건]                                  (6)
[ORDER BY   애트리뷰트(들) [ASC | DESC] ];               (7)
```

[그림] SELECT문의 형식

1. SELECT 문

- 기본적인 SQL 질의
  - SELECT절
    - 질의 결과를 포함 하려는 애트리뷰트들의 리스트 열거
    - DISTINCT 절을 사용해서 중복 제거
  - FROM절
    - 질의에서 필요로 하는 릴레이션들의 리스트 열거
  - WHERE절
    - 관계 대수의 셀렉션(Select) 연산의 조건에 해당
  - 중첩질의
    - 다른 질의의 WHERE절에 포함된 SELECT문
  - GROUP BY절
    - 동일한 값을 갖는 튜플들을 한 그룹으로 묶는다
  - HAVING절
    - 튜플들의 그룹이 만족해야 하는 조건
  - ORDER BY절
    - 결과 튜플들의 정렬 순서

1. SELECT 문

- **별칭**(alias)
  - **서로 다른 릴레이션에 동일한 이름을 가진 애트리뷰트가 속해 있을 때 애트리뷰트의 이름을 구분하는 방법**

```
SELECT

E.DEPARTMENTS_ID AS DEPTNO

FROM EMPLOYEES E, DEPARTMENTS D
```

1. SELECT 문

- 릴레이션의 모든 애트리뷰트나 일부들을 검색

예: *를 사용하여 모든 애트리뷰트들 검색

질의 : 전체 부서의 모든 애트리뷰트들을 검색하라.

```
SELECT      *
FROM        DEPARTMENTS;
```

1. SELECT 문

- 릴레이션의 모든 애트리뷰트나 일부들을 검색

예: 원하는 애트리뷰트들의 이름을 열거

질의 : 모든 부서의 부서번호와 부서이름을 검색하라.

```
SELECT      DEPARTMENTS_ID, DEPARTMENT_NAME
FROM        DEPARTMENTS;
```

DEPARTMENTS_ID	DEPARTMENT_NAME
1	영업
2	기획
3	개발
4	총무

- 상이한 값들을 검색

예: DISTINCT절을 사용하지 않을 때

질의 : 모든 사원의 직급을 검색하라.

```
SELECT      JOB_ID
FROM        EMPLOYEES;
```

JOB_ID
대리
과장
부장
과장
사원
사원
사장

- 상이한 값들을 검색

예: DISTINCT절을 사용할 때

질의 : 모든 사원의 직급을 검색하라.

```
SELECT      DISTINCT    JOB_ID
FROM        EMPLOYEES;
```

JOB_ID
대리
과장
부장
사원
사장

2. 조건에 맞는 자료 조회

- **특정한 투플들의 검색**

예: WHERE절을 사용하여 검색 조건을 명시

질의 : 2번 부서에 근무하는 사원들에 관한 모든 정보를 검색하라.

```
SELECT      *
FROM        EMPLOYEES
WHERE       DEPARTMENTS_ID = 2;
```

EMPLOYEE_ID	FIRST_NAME	JOB_ID	MANAGER_ID	SALARY	DEPARTMENTS_ID
2106	김창섭	대리	1077	2500000	2
1003	조민희	과장	4377	3000000	2
4377	이성래	사장	^	5000000	2

2. 조건에 맞는 자료 조회

- **문자열 비교**

예: %를 사용하여 문자열 비교

질의 : 이씨 성을 가진 사원들의 이름, 직급, 소속부서 정보를 검색하라.

```
SELECT          FIRST_NAME, JOB_ID, DEPARTMENTS_ID
FROM            EMPLOYEES
WHERE           FIRST_NAME LIKE  '이%';
```

FIRST_NAME	JOB_ID	DEPARTMENTS_ID
이수민	부장	3
이성래	사장	2

2. 조건에 맞는 자료 조회

- 다수의 검색 조건
  - WHERE 절은 여러 조건들이 부울 연산자로 결합된 조건을 포함할 수 있다.
  - 부서이름은 단일 값을 갖기 때문에 WHERE절을 만족하는 튜플은 하나도 없다. 아래와 같은 질의는 잘못되었음

```
SELECT      FLOOR
FROM        DEPARTMENTS
WHERE       DEPARTMENT_NAME = '영업'  AND DEPARTMENT_NAME =
```

[표]
연산자들의 우선순위

연산자	우선순위
비교연산자	1
NOT	2
AND	2
OR	3

2. 조건에 맞는 자료 조회

예: 부울 연산자를 사용한 프레디키드

질의 : 직급이 과장이면서 1번 부서에 근무하는 사원들의 이름과 급여를 검색하라.

```
SELECT          FIRST_NAME, SALARY
FROM            EMPLOYEES
WHERE           JOB_ID = '과장'  AND DEPARTMENTS_ID = 1;
```

FIRST_NAME	SALARY
박영권	3000000

2. 조건에 맞는 자료 조회

- **부정 검색 조건**

예: 부정 연산자

질의 : 직급이 과장이면서 1번 부서에 속하지 않은 사원들의 이름과 급여를 검색하라.

```
SELECT      FIRST_NAME, SALARY
FROM        EMPLOYEES
WHERE       JOB_ID = '과장'  AND DEPARTMENTS_ID <> 1;
```

FIRST_NAME	SALARY
조민희	3000000

2. 조건에 맞는 자료 조회

- 범위를 사용한 검색

예: 범위 연산자

질의 : 급여가 3000000원 이상이고, 4500000원 이하인 사원들의 이름, 직급, 급여를 검색하라.

```
SELECT      FIRST_NAME, JOB_ID, SALARY
FROM        EMPLOYEES
WHERE       SALARY BETWEEN 3000000 AND 4500000;
```

BETWEEN은 양쪽의 경계값을 포함하므로 이 질의는 아래의 질의와 동일

```
SELECT      FIRST_NAME, JOB_ID, SALARY
FROM        EMPLOYEES
WHERE       SALARY >= 3000000 AND SALARY <= 4500000;
```

FIRST_NAME	JOB_ID	SALARY
박영권	과장	3000000
이수민	부장	4000000
조민희	과장	3000000

2. 조건에 맞는 자료 조회

- **리스트를 사용한 검색**
  - IN은 리스트 내의 값과 비교
  - 예제는 DEPARTMENTS_ID 값이 (1, 3)에 속하는가를 검사

예: IN

질의 : 1번 이나 3번부서에 소속된 사원들에 관한 모든 정보를 검색하라.

```
SELECT          *
FROM            EMPLOYEES
WHERE           DEPARTMENTS_ID IN (1, 3);
```

EMPLOYEE_ID	FIRST_NAME	JOB_ID	MANAGER_ID	SALARY	DEPARTMENTS_ID
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
3427	최종철	사원	3011	1500000	3
1365	김상원	사원	3426	1500000	1

2. 조건에 맞는 자료 조회

- SELECT절에서 산술 연산자(+, -, *, /) 사용

예: 산술 연산자

질의 : 직급이 과장인 사원들에 대하여 이름과, 현재의 급여, 급여가 10% 인상됐을때의 값을 검색하라.

```
SELECT      FIRST_NAME, SALARY, SALARY * 1.1
FROM        EMPLOYEES
WHERE       JOB_ID = '과장' ;
```

FIRST_NAME	SALARY	SALARY * 1.1
박영권	3000000	3300000
조민희	3000000	3300000

2. 조건에 맞는 자료 조회

- 널값
  - 널값을 포함한 다른 값과 널값을 +, - 등을 사용하여 연산하면 결과는 널이 됨
  - COUNT(*)를 제외한 집단 함수들은 널값을 무시함
    - 목록출력시 환경설정에서 개수를 변경한다(최대 500 개)
  - 어떤 애트리뷰트에 들어 있는 값이 널인가 비교하기 위해서 'DEPARTMENTS_ID=NULL'처럼 나타내면 안됨

```
SELECT      FIRST_NAME, FIRST_NAME
FROM        EMPLOYEES
WHERE       DEPARTMENTS_ID = NULL;
```

2. 조건에 맞는 자료 조회

- 널값(계속)
  - 다음과 같은 비교 결과는 모두 거짓
    - NULL > 300
    - NULL = 300
    - NULL <> 300
    - NULL = NULL
    - NULL <> NULL
  - 올바른 표현

```
SELECT      FIRST_NAME, FIRST_NAME
FROM        EMPLOYEES
WHERE       DEPARTMENTS_ID IS NULL;
```

2. 조건에 맞는 자료 조회

표 unknown에 대한 OR 연산

	true	false	unknown
true	true	true	true
false	true	false	unknown
unknown	true	unknown	unknown

표 unknown에 대한 AND연산

	true	false	unknown
true	true	false	unknown
false	false	false	false
unknown	unknown	false	unknown

표 unknown에 대한 NOT연산

true	false
false	true
unknown	unknown

3. 정렬과 중복 행 제거

- ORDER BY절
  - 사용자가 SELECT문에서 질의 결과의 순서를 명시하지 않으면 DBMS가 투플들을 검색한 임의의 순서대로 사용자에게 제시됨
  - ORDER BY절에서 하나 이상의 애트리뷰트를 사용하여 검색 결과를 정렬할 수 있음
  - ORDER BY절은 SELECT문에서 가장 마지막에 사용되는 절
  - 디폴트 정렬 순서는 오름차순(ASC)
  - DESC를 지정하여 정렬 순서를 내림차순으로 지정할 수 있음
  - 널값은 오름차순에서는 가장 마지막에 나타나고, 내림차순에서는 가장 앞에 나타남

3. 정렬과 중복 행 제거

예: ORDER BY

질의 : 2번 부서에 근무하는 사원들의 이름, 직급, 급여를 검색하여 급여의 오름차순으로 정렬하라.

```
SELECT      FIRST_NAME, JOB_ID, SALARY
FROM        EMPLOYEES
WHERE       DEPARTMENTS_ID = 2
ORDER BY    SALARY;
```

FIRST_NAME	JOB_ID	SALARY
김창섭	대리	2500000
조민희	과장	3000000
이성래	사장	5000000

4. 그룹 함수

- 집단 함수
  - 데이터베이스에서 검색된 여러 투플들의 집단에 적용되는 함수
  - 각 집단 함수는 한 릴레이션의 한 개의 애트리뷰트에 적용되어 단일 값을 반환함
  - SELECT절과 HAVING절에만 나타날 수 있음
  - COUNT(*)를 제외하고는 모든 집단 함수들이 널값을 제거한 후 남아 있는 값들에 대해서 집단 함수의 값을 구함
  - COUNT(*)는 결과 릴레이션의 모든 행들의 총 개수를 구하는 반면에 COUNT(애트리뷰트)는 해당 애트리뷰트에서 널값이 아닌 값들의 개수를 구함
  - 키워드 DISTINCT가 집단 함수 앞에 사용되면 집단 함수가 적용되기 전에 먼저 중복을 제거함

4. 그룹 함수

예: MAX

질의 : 모든 사원들의 평균 급여와 최대 급여를 검색하라.

```
SELECT      AVG(SALARY),MAX(SALARY)
FROM        EMPLOYEES;
```

AVG(SALARY)	MAX(SALARY)
2928571	5000000

[표] 그룹함수의 기능

집단함수	기능
COUNT	튜플이나 값들의 개수
SUM	값들의 합
AVG	값들의 평균값
MAX	값들의 최대값
MIN	값들의 최소값

4. 그룹 함수

- **그룹화**
  - GROUP BY절에 사용된 애트리뷰트에 동일한 값을 갖는 투플들이 각각 하나의 그룹으로 묶임
  - 이때 사용된 애트리뷰트를 그룹화 애트리뷰트(grouping attribute)라고 함
  - 각 그룹에 대하여 결과 릴레이션에 하나의 투플이 생성됨
  - SELECT절에는 각 그룹마다 하나의 값을 갖는 애트리뷰트, 집단함수, 그룹화에 사용된 애트리뷰트들만 나타날 수 있음
  - 다음 질의는 그룹화를 하지 않은 채 EMPLOYEES 릴레이션의 모든 투플에 대해서 사원번호와 모든 사원들의 평균 급여를 검색하므로 잘못됨

```
SELECT      EMPLOYEE_ID,AVG(SALARY)
FROM        EMPLOYEES;
```

4. 그룹 함수

예: 그룹화

질의 : 모든 사원들에 대해서 사원들이 속한 부서번호별로 그룹화하고, 각 부서마다 부서번호, 평균급여, 최대급여를 검색하라.

```
SELECT          DEPARTMENTS_ID, AVG(SALARY),MAX(SALARY)
FROM            EMPLOYEES
GROUP BY        DEPARTMENTS_ID;
```

4. 그룹 함수

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	JOB_ID	MANAGER_ID	SALARY	DEPARTMENTS_ID
1365	김상원	사원	3426	1500000	1
3426	박영권	과장	4377	3000000	1
2106	김창섭	대리	1077	2500000	2
1003	조민희	과장	4377	3000000	2
4377	이성래	사장	^	5000000	2
3427	최종철	사원	3011	1500000	3
3011	이수민	부장	4377	4000000	3

GROUP BY

DEPARTMENTS_ID	AVG(SALARY)	MAX(SALARY)
1	2250000	3000000
2	3500000	5000000
3	2750000	4000000

4. 그룹 함수

- HAVING
  - 어떤 조건을 만족하는 그룹들에 대해서만 집단 함수를 적용할 수 있음
  - 각 그룹마다 하나의 값을 갖는 애트리뷰트를 사용하여 각 그룹이 만족해야 하는 조건을 명시함
  - HAVING절은 그룹화 애트리뷰트에 같은 값을 갖는 투플들의 그룹에 대한 조건을 나타내고, 이 조건을 만족하는 그룹들만 질의 결과에 나타남
  - HAVING절에 나타나는 애트리뷰트는 반드시 GROUP BY절에 나타나거나 집단 함수에 포함되어야 함

예: HAVING

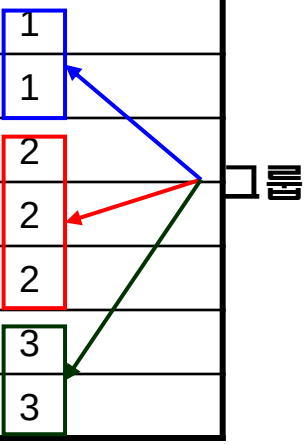
질의 : 모든 사원들에 대해서 사원들이 속한 부서번호별로 그룹화하고, 평균 급여가 2500000 이상인 부서에 대해서 부서번호, 평균급여, 최대급여를 검색하라.

```
SELECT      DEPARTMENTS_ID, AVG(SALARY),MAX(SALARY)
FROM        EMPLOYEES
GROUP BY    DEPARTMENTS_ID
HAVING      AVG(SALARY) >= 25000000;
```

4. 그룹 함수

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	JOB_ID	MANAGER_ID	SALARY	DEPARTMENTS_ID
1365	김상원	사원	3426	1500000	1
3426	박영권	과장	4377	3000000	1
2106	김창섭	대리	1077	2500000	2
1003	조민희	과장	4377	3000000	2
4377	이성래	사장	^	5000000	2
3427	최종철	사원	3011	1500000	3
3011	이수민	부장	4377	4000000	3



GROUP BY

DEPARTMENTS_ID	AVG(SALARY)	MAX(SALARY)
1	2250000	3000000
2	3500000	5000000
3	2750000	4000000

HAVING

DEPARTMENTS_ID	AVG(SALARY)	MAX(SALARY)
2	3500000	5000000
3	2750000	4000000