

2024학년도 1학기 그룹스터디

데이터베이스 3주차

MaveDB (한아림, 한별하, 이경화)



SQL의 개요

SQL

관계형 데이터베이스 표준언어

- 가장 많이 사용된다
- MySQL등 거의 모든 관계형 DBMS가 지원



SQL의 기능별 분류

데이터 정의어 (DDL)

- Data Definition Language
- DB 구조 정의
- DB 객체 생성, 수정, 삭제

데이터 정의어 (DDL)

- Data Manipulation Language
- DB 데이터 관리
- 입력, 수정, 삭제, 검색

데이터 제어어 (DCL)

- Data Control Language
- DB 관리 및 통제
- DB 백업, 복원
- 사용자 등록, 권한 관리

MYSQL

오픈소스 RDBMS

- 가장 많이 사용된다
- 누구나 다운로드하여 사용가능한 데이터베이스 관리 시스템
- 무료
- <http://www.mysql.com>에서 다운로드 가능



SQL 데이터 조작문

SELECT문

- 행검색을 위해 사용한다
- 테이블로부터 데이터 검색 시에는 SELECT 문 사용

SELECT문의 형식

- SELECT절
- FROM절
- WHERE절 (조건 검색)
- ORDER BY절 (순서화 검색)

```
①SELECT [DISTINCT|ALL] 열_리스트           ⑤
②FROM   테이블_리스트                     ①
        [WHERE 검색_조건식]                 ②
        [GROUP BY 그룹_기준열_리스트]       ③
        [HAVING 그룹_조건식]                ④
        [ORDER BY { 정렬_기준열 [ASC|DESC][.] }+ ]; ⑥
```

```
SELECT  ①[DISTINCT|ALL] 열_리스트
FROM     테이블_리스트
[WHERE  ③검색_조건식] ;②
```

```
SELECT [DISTINCT|ALL] 열_리스트
FROM   테이블_리스트
[ORDER BY { ①정렬_기준열 ②[ASC|DESC][.]}+ ] ;
```

SQL 기본 집계 함수

집계 함수

- 각 열에 대한 기본 통계 결과를 반환하는 함수
- 실제 테이블 저장 값이 아닌 행의 개수 또는 특정 열의 평균을 구하는 질의가 필요할 경우 사용

집계 함수	기능
COUNT(열_이름)	특정 열 값의 개수 또는 행의 개수
MAX(열_이름)	특정 열 값 중에서 최댓값
MIN(열_이름)	특정 열 값 중에서 최솟값
SUM(열_이름)	특정 열 값의 합계
AVG(열_이름)	특정 열 값의 평균

그룹 함수

GROUP BY절

- 그룹화 검색에 사용

HAVING절

- 그룹화 조건 검색에 사용

```
SELECT [DISTINCT|ALL] 열_리스트❷  
FROM 테이블_리스트  
[GROUP BY 그룹_기준열_리스트]❶  
[HAVING 그룹_조건식] ;❸
```

```
SELECT 학년, COUNT(*) AS '학년별 학생수'  
FROM 학생  
GROUP BY 학년  
HAVING COUNT(*) >= 2;
```


NULL값 검색

IS NULL

- NULL 연산에 사용

IS NOT NULL

- NULL 비교 검색에 사용

부 질의문을 이용한 검색

부 질의문

- 내부 질의
- 외부 질의
- 중첩 부 질의문

```
SELECT 이름  
FROM 학생  
WHERE 학번 IN ( SELECT 학번  
                FROM 수강  
                WHERE 과목번호='c002' ) ;
```

```
SELECT 이름  
FROM 학생  
WHERE 학번 IN ( SELECT 학번  
                FROM 수강  
                WHERE 과목번호 = ( SELECT 과목번호  
                                   FROM 과목  
                                   WHERE 이름='정보보호' ) ) ;
```

IN 연산자

- OR 연산과 유사

```
SELECT 이름  
FROM 학생  
WHERE 학번 IN ( 's001', 's003', 's004' ) ;
```

EXISTS 연산자

- 부 질의문의 실행 결과로 반환되는 행이 존재하는지 확인하는 연산자

```
SELECT 이름  
FROM 학생  
WHERE EXISTS ( SELECT *  
               FROM 수강  
               WHERE 수강.학번=학생.학번 AND 과목번호='c002' ) ;
```

조인 검색

JOIN

- 둘 이상의 테이블로부터 연관된 행들의 결합을 통해 검색결과 생성

CROSS JOIN

- ‘조인_조건식’ 없이 이루어진 조인

EQUI JOIN

- 조인_조건식’에 ‘=’ 연산자를 사용하는 동등 조건에 의한 조인

[조인 명세 방법1]①

```
SELECT 열_리스트  
FROM   조인테이블_리스트  
WHERE  ③<조인_조건식> AND <검색_조건식>
```

[조인 명세 방법2]②

```
SELECT 열_리스트  
FROM   테이블이름1 { [INNER] JOIN 테이블이름2 ON ③<조인_조건식> }+  
WHERE  <검색_조건식>
```

행 수정, 삽입, 삭제 문

INSERT문

- 행 삽입에 사용

```
INSERT INTO 테이블_이름 [(열_리스트)]❶  
VALUES (열_값_리스트) ;
```

UPDATE문

- 행 수정에 사용

```
UPDATE 테이블_이름❶  
SET 열_이름 = 산술식 [{.열_이름=산술식}*]❷  
[WHERE 수정_조건식] ;❸
```

DELETE문

- 행 삭제에 사용

```
DELETE FROM 테이블_이름❶  
[WHERE 삭제_조건식] ;❷
```

THANK
YOU

www.reallygreatsite.com

