

**Select
SQL**



레코드 검색

Select

❖ 이해하기

- 저장되어 있는 데이터를 출력하기 위한 문법

❖ 구조

- SELECT [출력할 필드명]
- FROM [테이블명]
- WHERE [조건]

❖ 사용 예

SELECT dept_name
FROM department
WHERE dept_id=1

1	SELECT	dept_name
2	FROM	department
3	WHERE	dept_id=1
4		
<		
	dept_name	
1	컴퓨터 과학부	

distinct

❖ 이해하기

- 데이터 출력 시 중복 데이터 제거

❖ 사용 예

```
SELECT          year
FROM            student
```

```
SELECT  DISTINCT  year
FROM    student
```

1	SELECT	year
2	FROM	student
3		

	year
1	3
2	3
3	4

1	SELECT	DISTINCT	year
2	FROM		student
3			

	year
1	3
2	4

산술식

❖ 이해하기

- 필드 출력 부분에 산술 연산을 처리할 수 있음.

❖ 사용 예

```
SELECT      2000+year  
FROM        student
```

1	SELECT	2000+year
2	FROM	student
3		

	2000+year
1	2003
2	2003
3	2004

실습

모든 레코드 출력

데이터베이스 구조 데이터 보기 Pragma 수정 **SQL 실행**



SQL 1

```
1  SELECT *
2  FROM department
3
```

*를 이용하여
모든 레코드 출력

	dept_id	dept_name	office
1	1	컴퓨터 과학부	302호
2	2	전기전자 공학부	403호
3	3	데이터 싸이언스	303호

선택 레코드 출력

데이터베이스 구조

데이터 보기

Pragma 수정

SQL 실행



SQL 1

```
1 SELECT *
2 FROM department
3 WHERE dept_id=3
<
```

조건절을 이용하여
특정 레코드 출력

	dept_id	dept_name	office
1	3	데이터 싸이언스	303호

선택 레코드 출력

데이터베이스 구조

데이터 보기

Pragma 수정

SQL 실행



SQL 1



```
1  SELECT *
2  FROM   department
3  WHERE  dept_id < 3
```

<

비교연산을 통한
레코드 출력

	dept_id	dept_name	office
1	1	컴퓨터 과학부	302호
2	2	전기전자 공학부	403호

Student 테이블 레코드 추가

```
INSERT INTO student VALUES('1292001', '900424', 3, '1');
```

```
INSERT INTO student VALUES('1292002', '900305', 3, '2');
```

```
INSERT INTO student VALUES('1292003', '991021', 1, '3');
```

```
INSERT INTO student VALUES('1292004', '930504', 4, '1');
```

```
INSERT INTO student VALUES('1292005', '970105', 2, '2');
```

```
INSERT INTO student VALUES('1292006', '961101', 2, '3');
```

```
INSERT INTO student VALUES('1292007', '920214', 3, '1');
```

```
INSERT INTO student VALUES('1292008', '960305', 2, '2');
```

```
INSERT INTO student VALUES('1292009', '931224', 4, '3');
```

```
INSERT INTO student VALUES('1292010', '980824', 1, '1');
```

```
INSERT INTO student VALUES('1292011', '970625', 1, '2');
```

```
INSERT INTO student VALUES('1292012', '940721', 2, '3');
```

중복 데이터 제거

데이터베이스 구조 데이터 보기 Pragma 수정 **SQL 실행**

SQL 1 ✕

```
1  SELECT year
2  FROM student
3  WHERE year>2
```

	year
1	3
2	3
3	4
4	4
5	3
6	4

2학년 이상인 학생들의
학년 출력

중복 데이터 제거

데이터베이스 구조 데이터 보기 Pragma 수정 **SQL 실행**



SQL 1

```
1  SELECT  DISTINCT  year
2  FROM    student
3  WHERE   year>2
```

	year
1	3
2	4

중복된 데이터 제거

산술 연산을 통한 데이터 출력

데이터베이스 구조 데이터 보기 Pragma 수정 **SQL 실행**

SQL 1 ✕

```
1  SELECT  stu_id, year+1
2  FROM    student
3  WHERE   year<4
```

	stu_id	year+1
1	1292001	4
2	1292002	4
3	1292005	3
4	1292006	3
5	1292007	4
6	1292008	3
7	1292010	2

4학년 미만인 학생들의
학년+1 출력

Quiz

Quiz

	stu_id	resident_id	year	dept_id
1	1292002	900305	3	2
2	1292005	970105	2	2
3	1292008	960305	2	2
4	1292011	970625	1	2



Quiz

	stu_id	resident_id	year	dept_id
1	1292003	991021	4	3
2	1292005	970105	2	2
3	1292006	961101	2	3
4	1292008	960305	2	2
5	1292010	980824	1	1
6	1292011	970625	1	2



조인

조인

❖ 이해하기

- 두 개의 테이블을 연결하여 결과를 출력

❖ 사용 예

```
SELECT      *  
FROM        student, department  
WHERE       student.dept_id = department.dept_id
```

```
SELECT      *  
FROM        student as std, department as dept  
WHERE       std.dept_id = dept.dept_id
```

컬럼명 변경

❖ 이해하기

- 출력 결과 중 컬럼명을 별도의 지정된 이름으로 출력

❖ 사용 예

```
SELECT      std.stu_id as '학번', dept.dept_name as 학과
FROM        student as std, department as dept
WHERE       std.dept_id = dept.dept_id
```

논리연산

❖ 이해하기

- 2개 이상의 조건을 설정할 경우 사용

❖ 사용 예

```
SELECT      std.stu_id as '학번', dept.dept_name as 학과
FROM        student as std, department as dept
WHERE       std.dept_id = dept.dept_id
AND         dept.dept_id=2
```

```
SELECT      std.stu_id as '학번', dept.dept_name as 학과
FROM        student as std, department as dept
WHERE       std.dept_id = dept.dept_id
AND         dept.dept_id=2
OR          dept.dept_id=1
```

실습

조인

```

1  SELECT  *
2  FROM    student, department
3  WHERE   student.dept_id = department.dept_id
4

```

	stu_id	resident_id	year	dept_id	dept_id	dept_name	office
1	1292001	900424	3	1	1	컴퓨터 과학부	302호
2	1292002	900305	3	2	2	전기전자 공학부	403호
3	1292003	991021	4	3	3	데이터 사이언스	303호
4	1292004	930504	4	1	1	컴퓨터 과학부	302호
5	1292005	970105	2	2	2	전기전자 공학부	403호
6	1292006	961101	2	3	3	데이터 사이언스	303호
-	-----	-----	-	.	.	-----	----

테이블 별칭

```

1  SELECT  *
2  FROM    student as std, department as dept
3  WHERE   std.dept_id = dept.dept_id
4

```

	stu_id	resident_id	year	dept_id	dept_id	dept_name	office	
1	1292001	900424	3	1	1	컴퓨터 과학부	302호	
2	1292002	900305	3	2	2	전기전자 공학부	403호	
3	1292003	991021	4	3	3	데이터 사이언스	303호	
4	1292004	930504	4	1	1	컴퓨터 과학부	302호	
5	1292005	970105	2	2	2	전기전자 공학부	403호	
6	1292006	961101	2	3	3	데이터 사이언스	303호	

컬럼명 변경

```
1  SELECT  std.stu_id as '학번', dept.dept_name as 학과
2  FROM    student as std, department as dept
3  WHERE   std.dept_id = dept.dept_id
4
```

	학번	학과
1	1292001	컴퓨터 과학부
2	1292002	전기전자 공학부
3	1292003	데이터 사이언스
4	1292004	컴퓨터 과학부
5	1292005	전기전자 공학부
6	1292006	데이터 사이언스

논리연산

```
1 SELECT std.stu_id as '학번', dept.dept_name as 학과
2 FROM student as std, department as dept
3 WHERE std.dept_id = dept.dept_id
4 AND dept.dept_id=2
```

	학번	학과
1	1292002	전기전자 공학부
2	1292005	전기전자 공학부
3	1292008	전기전자 공학부
4	1292011	전기전자 공학부

논리연산

```
1  SELECT  std.stu_id as '학번', dept.dept_name as 학과
2  FROM    student as std, department as dept
3  WHERE   std.dept_id = dept.dept_id
4  AND     dept.dept_id=2
5  OR      dept.dept_id=1
```

	학번	학과
1	1292002	전기전자 공학부
2	1292005	전기전자 공학부
3	1292008	전기전자 공학부
4	1292011	전기전자 공학부
5	1292001	컴퓨터 과학부
6	1292002	컴퓨터 과학부

중복 데이터 발생

```
1  SELECT  std.stu_id as '학번'
2  FROM    student as std, department as dept
3  WHERE   std.dept_id = dept.dept_id
4  AND     dept.dept_id=2
5  OR      dept.dept_id=1
```

	학번
1	1292002
2	1292005
3	1292008
4	1292011
5	1292001
6	1292002
7	1292003

중복 데이터 발생

```
1  SELECT  std.stu_id as '학번'
2  FROM    student as std, department as dept
3  WHERE   std.dept_id = dept.dept_id
4  AND     dept.dept_id=2
5  OR      dept.dept_id=1
```

	학번
1	1292002
2	1292005
3	1292008
4	1292011
5	1292001
6	1292002
7	1292003

```
SELECT  DISTINCT std.stu_id as '학번'
FROM    student as std, department as dept
WHERE   std.dept_id = dept.dept_id
AND     dept.dept_id=2
OR      dept.dept_id=1
```

Quiz

Quiz

	stu_id	resident_id	dept_id+1
1	1292005	970105	3
2	1292008	960305	3
3	1292011	970625	3



Quiz

	학번	학과	호수
1	1292001	컴퓨터 과학부	302호
2	1292002	전기전자 공학부	403호
3	1292003	데이터 사이언스	303호
4	1292004	컴퓨터 과학부	302호
5	1292005	전기전자 공학부	403호
6	1292006	데이터 사이언스	303호

