

1. SELECT 문의 기본 형식



SQL SELECT 문의 기능

프로젝션

테이블 1

선택

테이블 1

조인

테이블 1

테이블 2

기본 SELECT 문

- SELECT는 표시할 대상 열을 지정합니다.
- FROM은 대상 열을 포함하는 해당 테이블을 지정합니다.
- 구문

```
SELECT *|{[DISTINCT] column|expression [alias],...}  
FROM    table;
```

- SELECT 하나 이상의 열로 구성되는 목록입니다.
- * 모든 열을 선택합니다.
- DISTINCT 중복되는 열을 생략합니다.
- column|expression 지정한 열 또는 표현식을 선택합니다.
- alias 선택된 열에 다른 머리글 이름을 줍니다.
- FROM table 해당 열을 포함하는 테이블을 지정합니다.

열 선택

- 모든 열 선택

```
SELECT *  
FROM departments;
```

- 특정 열 선택

```
SELECT department_id, location_id  
FROM departments;
```

SQL 문 작성

- SQL 문은 대소문자를 구분하지 않습니다.
- SQL 문은 하나 이상의 줄에 입력할 수 있습니다.
- 키워드는 약어로 쓰거나 여러 줄에 나눠 쓸 수 없습니다.
- 절은 일반적으로 서로 다른 줄에 씁니다.
- 들여쓰기를 사용하면 SQL 문을 좀더 읽기 쉽게 작성할 수 있습니다.



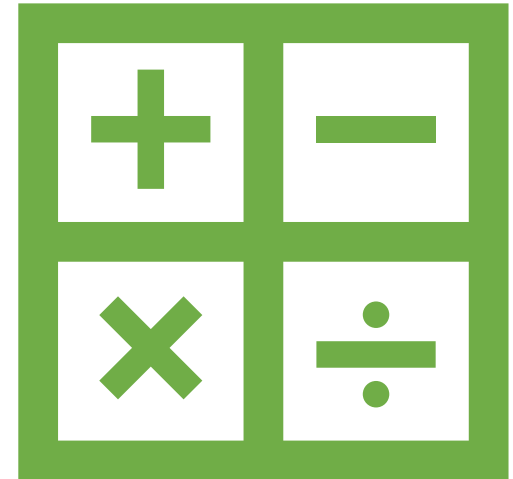
산술식

- 산술 연산자를 사용하여 숫자 및 날짜 데이터에 대한 표현식을 작성합니다.

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기

연산자 우선순위

- 곱셈과 나눗셈은 덧셈과 뺄셈보다 우선순위가 높습니다.
- 우선순위가 동일한 연산자는 왼쪽에서 오른쪽으로 계산됩니다.
- 괄호는 우선적으로 계산되며 명령문을 명확히 나타내는 역할을 합니다.



산술 연산자 사용

- 산술연산자 사용

```
SELECT last_name, salary, salary + 300  
FROM employees;
```

- 연산자 우선순위 및 괄호 사용

```
SELECT last_name, salary, 12*salary+100  
FROM employees;
```

```
SELECT last_name, salary, 12*(salary+100)  
FROM employees;
```


널 값 정의

- 널 값의 의미 :
- 알 수 없는 값, 사용할 수 없는 값, 할당할 수 없는 값, 적용할 수 없는 값을 의미합니다.
- 널은 0 또는 공백과 다릅니다.
- 한 행의 특정 열에 데이터 값이 없으면 그 값을 널이라고 하거나 널을 포함한다고 합니다.
- 널 값을 포함하는 산술식은 널로 평가됩니다.

널 값 정의

- EMPLOYEES 테이블의 COMMISSION_PCT 열의 경우 영업 과장과 영업 직원만이 커미션을 받을 수 있습니다.
- 영업직원이 아닌 직원들은 커미션을 받을 수 없는데, 이 경우 널을 사용합니다.

```
SELECT last_name, job_id, salary, commission_pct  
FROM employees;
```

- 커미션이 널인 직원은 다음과 같은 산술식의 결과가 널입니다.

```
SELECT last_name, 12*salary*commission_pct  
FROM employees;
```

→ 산술식에 널값이 포함되면 계산 결과가 널이 됩니다.

열 별칭 정의

- 열 별칭:
- 열 머리글의 이름을 변경합니다.
- 계산식에 대한 열머리글을 지정할때 유용합니다.
- 열 이름 바로 뒤에 사용합니다. 열 이름과 별칭 사이에 선택적으로 AS 키워드를 사용할 수 있습니다.
- 공백 또는 특수 문자가 있거나 대소문자를 구분할 경우 큰 따옴표를 사용합니다.

열 별칭 사용

- 모든 사원의 이름과 커미션 비율을 표시합니다.
 - 선택 사항인 AS 키워드가 열 별칭 이름 앞에 사용되었으며 열 별칭인 name 및 comm이 소문자로 되어 있지만 결과에서는 열 머리글이 대문자로 표시됩니다.

```
SELECT last_name AS name, commission_pct comm  
FROM employees;
```

- 모든 사원의 이름과 연봉을 표시합니다.
 - Annual Salary에는 공백이 포함되어 있으므로 큰 따옴표로 묶여 있습니다.

```
SELECT last_name "Name", salary*12 "Annual Salary"  
FROM employees;
```

연결 연산자

- 연결 연산자:
- 열 또는 문자열을 다른 열에 연결합니다.
- 두 개의 세로선(||)으로 표시합니다.
- 문자식에 해당하는 결과 열을 생성합니다.
- 사원의 이름과 업무 코드가 결합되어 하나의 열로 출력됩니다

```
SELECT last_name||job_id AS "Employees"  
FROM   employees;
```

리터럴 문자열

- 리터럴은 SELECT 목록에 포함된 문자, 숫자 또는 날짜입니다.
- 날짜 및 문자 리터럴 값은 작은 따옴표로 묶어야 합니다.
- 각 문자열은 각 행(row)이 반환될 때마다 한 번씩 출력됩니다.
- 예제는 모든 사원의 전체이름과 업무코드를 함께 출력합니다.
- 결과에는 이름과 성 사이에 띄워쓰기를, 이름과 업무코드 사이엔 공백과 콜론(:)을 출력합니다.

```
SELECT first_name||' '||last_name||' : '|| job_id AS "Employees"  
FROM employees;
```

대체 인용(q) 연산자

- 따옴표 자체를 리터럴로 사용할 경우 다음과 같이 작성해야 합니다.

```
SELECT last_name || '''s Salary : ''' || salary AS "Salary Report "  
FROM employees;
```

- q 연산자를 사용할 경우 [],{}와 같은 사용자 정의 따옴표 구분자로 사용됩니다.
- 따옴표 구분자 사이의 문자열은 리터럴 문자열로 해석됩니다
- q 연산자를 사용하면 가독성 및 사용성이 증가합니다.

```
SELECT last_name || q['s Salary : '] || salary AS "Salary Report "  
FROM employees;
```

중복 행 제거

- 질의는 기본적으로 중복 행을 포함한 모든 행을 표시합니다.

```
SELECT department_id  
FROM employees;
```

- SELECT 절에서 DISTINCT 키워드를 사용하여 중복 행을 제거합니다.

```
SELECT DISTINCT department_id  
FROM employees;
```


여러 열을 지정하여 중복 행 제거

- DISTINCT 수식자 뒤에 여러 열을 지정할 수 있습니다.
- DISTINCT 수식자는 선택된 모든 열에 영향을 주며 결과에는 모든 고유한 열 조합이 나타납니다.

```
SELECT DISTINCT department_id, job_id  
FROM employees;
```

- 키워드 DISTINCT 대신 동의어인 키워드 UNIQUE를 지정할 수도 있습니다.

```
SELECT UNIQUE department_id, job_id  
FROM employees;
```

Thank you 😊