

## 추가, 삭제, 갱신, 조회 (2)

### 여러 행을 INSERT하기

**Person** 테이블에 행을 추가하자. **VALUES** 절에 여러 행의 값을 적어서 한 번에 삽입할 수 있다.

```
INSERT INTO Person (Name, Birthday)
VALUES ('박소진', '1986-05-21'), ('김아영', '1992-11-06');
```

### 열(column) 값의 자동 증가(AUTOINCREMENT)

테이블을 생성할 때 **ID** 컬럼에 대하여 **AUTOINCREMENT** 속성을 설정했으므로, 위와 같이 **ID** 열을 생략하고 **INSERT** 하면 자동으로 값이 채워진다.<sup>1</sup>

### 널(NULL)

**Person** 테이블을 정의할 때 **Birthday** 컬럼은 널 값을 허용하도록 했으므로, 다음과 같이 실행하더라도 문제가 없다.

```
INSERT INTO Person (Name) VALUES ('민아');
```

이제 테이블을 조회해보면 다음과 비슷한 결과를 얻을 수 있을 것이다.

ID	Name	Birthday
1	헤리	1994-06-09
2	박소진	1986-05-21
3	김아영	1992-11-06
4	민아	NULL

### 원하는 열만 조회하기

**SELECT** 절에 컬럼명을 지정하여 조회할 수 있다.

```
SELECT Name FROM Person;
```

Name
헤리

Name
박소진
김아영
민아

## 원하는 순서로 조회하기

특정 컬럼을 기준으로 행을 정렬하여 조회할 수 있다.<sup>2</sup>

다음 문장은 `Person` 테이블의 `Name` 컬럼을 나열하되, `Name` 을 기준으로 오름차순으로 정렬한다.

```
SELECT Name FROM Person ORDER BY Name;
```

Name
김아영
민아
박소진
헤리

`DESC` 키워드를 추가하여 역순으로 조회할 수 있다.

```
SELECT Name FROM Person ORDER BY Name DESC;
```

Name
헤리
박소진
민아
김아영

`ORDER BY` 절을 이용해 정렬했다고 해서 데이터가 저장된 순서가 영구적으로 바뀌는 것은 아니다. `SELECT` 를 할 때만 정렬해서 보여준 것으로 생각하면 된다.

## SELECT ... WHERE

`Name` 이 '박소진'인 행을 찾아보자.

```
SELECT * FROM Person WHERE Name = '박소진';
```

실행결과는 다음과 같다.

ID	Name	Birthday
2	박소진	1986-05-21

이번에는 **Brithday** 가 널인 행을 제외하고 조회해보자.

```
SELECT * FROM Person
WHERE Birthday IS NOT NULL
```

실행결과는 다음과 같다.

ID	Name	Birthday
1	헤리	1994-06-09
2	박소진	1986-05-21
3	김아영	1992-11-06

## UPDATE ... WHERE

**UPDATE** 또는 **DELETE** 문에도 **WHERE** 절을 사용할 수 있다. **Name** 이 '박소진'인 행에 대하여, **Name** 을 '소진'으로 바꾸어보자.

```
UPDATE Person SET Name = '소진' WHERE Name = '박소진';
```

## LIKE

**WHERE** 절에 **=** 대신 **LIKE** 를 사용하여, 패턴과 일치하는 문자열을 찾을 수 있다.

```
SELECT * FROM Person WHERE Birthday LIKE '1986%';
```

ID	Name	Birthday
2	소진	1986-05-21