

지금까지 테이블, 로우, 컬럼 생성 방법과 `Session` 클래스를 사용해 데이터를 추가하는 방법을 배웠다. 다음 절에서는 데이터베이스를 만들고 CRUD를 처리하는 방법을 살펴본다.

## 6.2 데이터베이스 생성

`SQLModel`에서는 `SQLAlchemy` 엔진을 사용해서 데이터베이스를 연결한다. `SQLAlchemy` 엔진은 `create_engine()` 메서드를 사용해서 만들며 `SQLModel` 라이브러리에서 임포트한다.

`create_engine()` 메서드는 데이터베이스 URL을 인수로 사용한다. 데이터베이스 URL은 `sqlite:///database.db` 또는 `sqlite:///database.sqlite`와 같은 형식이다. `create_engine()`은 `echo`를 선택적 인수로 지정할 수 있다. `True`로 설정하면 실행된 SQL 명령을 출력한다.

`create_engine()` 메서드만으로는 데이터베이스 파일을 만들 수 없다. `SQLModel.metadata.create_all(engine)` 메서드를 사용해서 `create_engine()` 메서드의 인스턴스를 호출해야 한다.

---

```
database_file = "database.db"
engine = create_engine(database_file, echo=True)
SQLModel.metadata.create_all(engine)
```

---

`create_all()` 메서드는 데이터베이스뿐만 아니라 테이블도 생성한다. 중요한 점은 데이터베이스 연결 파일(`connection.py`)에서 테이블 파일을 임포트해야 한다는 것이다.

이벤트 플래너 애플리케이션에서는 이벤트에 대한 CRUD 처리를 수행한다. `database` 폴더에 다음과 같이 파일을 생성하자. 이 파일에 데이터베이스 연결을 위한 데이터를 설정할 것이다.

---

```
(venv)$ touch database/connection.py
```

---

데이터베이스 연결을 위한 파일을 생성했으니 다음 단계를 따라 애플리케이션을 데이터베이스와 연동하는 함수를 작성하자.