

```
1  UPDATE flights SET duration = 430 WHERE origin = 'New York' AND destination =  
   'London';  
2  UPDATE flights SET duration = duration + 15 WHERE origin = 'New York' AND  
   destination = 'London';  
3
```

```
1  SELECT * FROM flights;
2  SELECT origin, destination FROM flights;
3  SELECT * FROM flights WHERE id = 3;
4  SELECT * FROM flights WHERE origin = 'New York';
5  SELECT * FROM flights WHERE duration > 500;
6  SELECT * FROM flights WHERE destination = 'Paris' AND duration > 500;
7  SELECT * FROM flights WHERE destination = 'Paris' OR duration > 500;
8  SELECT COUNT(*) FROM flights;
9  SELECT AVG(duration) FROM flights;
10
```

```
1  import os
2
3  from sqlalchemy import create_engine
4  from sqlalchemy.orm import scoped_session, sessionmaker
5
6  engine = create_engine(os.getenv("DATABASE_URL"))
7  db = scoped_session(sessionmaker(bind=engine))
8
9  def main():
10
11     # List all flights.
12     flights = db.execute("SELECT id, origin, destination, duration FROM
13     flights").fetchall()
14     for flight in flights:
15         print(f"Flight {flight.id}: {flight.origin} to {flight.destination},
16         {flight.duration} minutes.")
17
18     # Prompt user to choose a flight.
19     flight_id = int(input("\nFlight ID: "))
20     flight = db.execute("SELECT origin, destination, duration FROM flights WHERE id
21     = :id",
22                        {"id": flight_id}).fetchone()
23
24     # Make sure flight is valid.
25     if flight is None:
26         print("Error: No such flight.")
27         return
28
29     # List passengers.
30     passengers = db.execute("SELECT name FROM passengers WHERE flight_id =
31     :flight_id",
32                            {"flight_id": flight_id}).fetchall()
33     print("\nPassengers:")
34     for passenger in passengers:
35         print(passenger.name)
36     if len(passengers) == 0:
37         print("No passengers.")
38
39 if __name__ == "__main__":
40     main()
```

```
1  import os
2
3  from sqlalchemy import create_engine
4  from sqlalchemy.orm import scoped_session, sessionmaker
5
6  engine = create_engine(os.getenv("DATABASE_URL"))
7  db = scoped_session(sessionmaker(bind=engine))
8
9  def main():
10     flights = db.execute("SELECT origin, destination, duration FROM
        flights").fetchall()
11     for flight in flights:
12         print(f"{flight.origin} to {flight.destination}, {flight.duration} minutes.")
13
14  if __name__ == "__main__":
15     main()
16
```

```
1  CREATE TABLE passengers (  
2      id SERIAL PRIMARY KEY,  
3      name VARCHAR NOT NULL,  
4      flight_id INTEGER REFERENCES flights  
5  );  
6  
7  INSERT INTO passengers (name, flight_id) VALUES ('Alice', 1);  
8  INSERT INTO passengers (name, flight_id) VALUES ('Bob', 1);  
9  INSERT INTO passengers (name, flight_id) VALUES ('Charlie', 2);  
10 INSERT INTO passengers (name, flight_id) VALUES ('Dave', 2);  
11 INSERT INTO passengers (name, flight_id) VALUES ('Erin', 4);  
12 INSERT INTO passengers (name, flight_id) VALUES ('Frank', 6);  
13 INSERT INTO passengers (name, flight_id) VALUES ('Grace', 6);  
14  
15  
16 SELECT origin, destination, name FROM flights INNER JOIN passengers ON  
passengers.flight_id = flights.id;  
17 SELECT origin, destination, name FROM flights JOIN passengers ON  
passengers.flight_id = flights.id;  
18 SELECT origin, destination, name FROM flights LEFT OUTER JOIN passengers ON  
passengers.flight_id = flights.id;  
19 SELECT origin, destination, name FROM flights RIGHT OUTER JOIN passengers ON  
passengers.flight_id = flights.id;  
20
```

```
1  INSERT INTO flights (origin, destination, duration) VALUES ('New York', 'London',
2  415);
3  INSERT INTO flights (origin, destination, duration) VALUES ('Shanghai', 'Paris', 760);
4  INSERT INTO flights (origin, destination, duration) VALUES ('Istanbul', 'Tokyo', 700);
5  INSERT INTO flights (origin, destination, duration) VALUES ('New York', 'Paris', 435);
6  INSERT INTO flights (origin, destination, duration) VALUES ('Moscow', 'Paris', 245);
7  INSERT INTO flights (origin, destination, duration) VALUES ('Lima', 'New York', 455);
```

```
1  CREATE INDEX duration_index ON flights (duration) ;
2
```

```
1  import csv
2  import os
3
4  from sqlalchemy import create_engine
5  from sqlalchemy.orm import scoped_session, sessionmaker
6
7  engine = create_engine(os.getenv("DATABASE_URL"))
8  db = scoped_session(sessionmaker(bind=engine))
9
10 def main():
11     f = open("flights.csv")
12     reader = csv.reader(f)
13     for origin, destination, duration in reader:
14         db.execute("INSERT INTO flights (origin, destination, duration) VALUES
15             (:origin, :destination, :duration)",
16                 {"origin": origin, "destination": destination, "duration":
17                     duration})
18         print(f"Added flight from {origin} to {destination} lasting {duration}
19             minutes.")
20     db.commit()
21
22 if __name__ == "__main__":
23     main()
```



---

1	Paris, New York, 540
2	Tokyo, Shanghai, 185
3	Seoul, Mexico City, 825
4	Mexico City, Lima, 350
5	Hong Kong, Shanghai, 130
6	

```
1  DELETE FROM countries WHERE destination = 'Tokyo';
2
```

```
1  CREATE TABLE flights (  
2      id SERIAL PRIMARY KEY,  
3      origin VARCHAR NOT NULL,  
4      destination VARCHAR NOT NULL,  
5      duration INTEGER NOT NULL  
6  );  
7
```

```
1  SELECT * FROM flights LIMIT 2;
2  SELECT * FROM flights ORDER BY duration ASC;
3  SELECT * FROM flights ORDER BY duration DESC;
4  SELECT origin, COUNT(*) FROM flights GROUP BY origin;
5  SELECT origin, COUNT(*) FROM flights GROUP BY origin HAVING COUNT(*) > 1;
6
```

```
1  from flask import Flask, render_template, request, session
2  from flask_session import Session
3
4  app = Flask(__name__)
5
6  app.config["SESSION_PERMANENT"] = False
7  app.config["SESSION_TYPE"] = "filesystem"
8  Session(app)
9
10 notes = []
11
12 @app.route("/", methods=["GET", "POST"])
13 def index():
14     if request.method == "POST":
15         note = request.form.get("note")
16         notes.append(note)
17
18     return render_template("index.html", notes=notes)
19
```

```
1  {% extends "layout.html" %}
2
3  {% block heading %}
4      Second Page
5  {% endblock %}
6
7  {% block body %}
8      <p>
9          Donec lobortis dapibus magna, quis facilisis velit malesuada a. Nunc iaculis
          augue nulla, sed sodales dui volutpat lacinia. Vestibulum ante ipsum primis
          in faucibus orci luctus et ultrices posuere cubilia Curae; Nullam non lorem
          eget augue auctor feugiat at ut justo. Pellentesque habitant morbi tristique
          senectus et netus et malesuada fames ac turpis egestas. Aenean nec lectus
          eget diam tempor aliquam sit amet sed augue.
10     </p>
11
12     <a href="{ { url_for('index') } }">Go back</a>
13 {% endblock %}
14
```

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>My Website!</title>
5      <link rel="stylesheet" href="{ url_for('static', filename='styles.css') }">
6    </head>
7    <body>
8      <h1>{% block heading %}{% endblock %}</h1>
9
10     {% block body %}
11     {% endblock %}
12   </body>
13 </html>
14
```

```
1  {% extends "layout.html" %}
2
3  {% block heading %}
4      Notes
5  {% endblock %}
6
7  {% block body %}
8
9      <ul>
10         {% for note in notes %}
11             <li>{{ note }}</li>
12         {% endfor %}
13     </ul>
14
15     <form action="{{ url_for('index') }}" method="post">
16         <input type="text" name="note" placeholder="Enter Note Here">
17         <button>Add Note</button>
18     </form>
19 {% endblock %}
20
```



```
1  h1 {  
2      color: red;  
3  }  
4  
5  p {  
6      font-family: sans-serif;  
7      font-size: 18px;  
8  }  
9
```

```
1  import os
2
3  from flask import Flask, render_template, request
4  from sqlalchemy import create_engine
5  from sqlalchemy.orm import scoped_session, sessionmaker
6
7  app = Flask(__name__)
8
9  engine = create_engine(os.getenv("DATABASE_URL"))
10 db = scoped_session(sessionmaker(bind=engine))
11
12 @app.route("/")
13 def index():
14     flights = db.execute("SELECT * FROM flights").fetchall()
15     return render_template("index.html", flights=flights)
16
17 @app.route("/book", methods=["POST"])
18 def book():
19     """Book a flight."""
20
21     # Get form information.
22     name = request.form.get("name")
23     try:
24         flight_id = int(request.form.get("flight_id"))
25     except ValueError:
26         return render_template("error.html", message="Invalid flight number.")
27
28     # Make sure the flight exists.
29     if db.execute("SELECT * FROM flights WHERE id = :id", {"id":
30 flight_id}).rowcount == 0:
31         return render_template("error.html", message="No such flight with that id.")
32     db.execute("INSERT INTO passengers (name, flight_id) VALUES (:name, :flight_id)",
33 {"name": name, "flight_id": flight_id})
34     db.commit()
35     return render_template("success.html")
36
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Success!
5  {% endblock %}
6
7  {% block body %}
8      <h1>Success!</h1>
9
10     You have successfully booked your flight.
11 {% endblock %}
12
```

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>{% block title %}{% endblock %}</title>
5      <link rel="stylesheet"
6        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/css/bootstrap.min
7        .css"
8        integrity="sha384-Zug+QiDoJOrZ5t4lssLdxGhVrurbmBWopoEl+M6BdEfwnCJZtKx1lKgxUyJq
9        13dy" crossorigin="anonymous">
10     </head>
11     <body>
12       <div class="container">
13         {% block body %}
14         {% endblock %}
15       </div>
16     </body>
17   </html>
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Flights
5  {% endblock %}
6
7  {% block body %}
8      <h1>Book a Flight</h1>
9
10     <form action="{{ url_for('book') }}" method="post">
11
12         <div class="form-group">
13             <select class="form-control" name="flight_id">
14                 {% for flight in flights %}
15                     <option value="{{ flight.id }}">{{ flight.origin }} to {{
16                         flight.destination }}</option>
17                 {% endfor %}
18             </select>
19         </div>
20
21         <div class="form-group">
22             <input class="form-control" name="name" placeholder="Passenger Name">
23         </div>
24
25         <div class="form-group">
26             <button class="btn btn-primary">Book Flight</button>
27         </div>
28     </form>
29 {% endblock %}
30
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Error
5  {% endblock %}
6
7  {% block body %}
8      <h1>Error</h1>
9
10     {{ message }}
11 {% endblock %}
12
```

```
1  import os
2
3  from flask import Flask, render_template, request
4  from sqlalchemy import create_engine
5  from sqlalchemy.orm import scoped_session, sessionmaker
6
7  app = Flask(__name__)
8
9  engine = create_engine(os.getenv("DATABASE_URL"))
10 db = scoped_session(sessionmaker(bind=engine))
11
12 @app.route("/")
13 def index():
14     flights = db.execute("SELECT * FROM flights").fetchall()
15     return render_template("index.html", flights=flights)
16
17 @app.route("/book", methods=["POST"])
18 def book():
19     """Book a flight."""
20
21     # Get form information.
22     name = request.form.get("name")
23     try:
24         flight_id = int(request.form.get("flight_id"))
25     except ValueError:
26         return render_template("error.html", message="Invalid flight number.")
27
28     # Make sure flight exists.
29     if db.execute("SELECT * FROM flights WHERE id = :id", {"id":
30 flight_id}).rowcount == 0:
31         return render_template("error.html", message="No such flight with that id.")
32     db.execute("INSERT INTO passengers (name, flight_id) VALUES (:name, :flight_id)",
33 {"name": name, "flight_id": flight_id})
34     db.commit()
35     return render_template("success.html")
36
37 @app.route("/flights")
38 def flights():
39     """Lists all flights."""
40     flights = db.execute("SELECT * FROM flights").fetchall()
41     return render_template("flights.html", flights=flights)
42
43 @app.route("/flights/<int:flight_id>")
44 def flight(flight_id):
45     """Lists details about a single flight."""
46
47     # Make sure flight exists.
48     flight = db.execute("SELECT * FROM flights WHERE id = :id", {"id":
49 flight_id}).fetchone()
50     if flight is None:
51         return render_template("error.html", message="No such flight.")
52
53     # Get all passengers.
54     passengers = db.execute("SELECT name FROM passengers WHERE flight_id =
55 :flight_id",
56 {"flight_id": flight_id}).fetchall()
57     return render_template("flight.html", flight=flight, passengers=passengers)
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Success!
5  {% endblock %}
6
7  {% block body %}
8      <h1>Success!</h1>
9
10     You have successfully booked your flight.
11 {% endblock %}
12
```



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>{% block title %}{% endblock %}</title>
5      <link rel="stylesheet"
6        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/css/bootstrap.min
7        .css"
8        integrity="sha384-Zug+QiDoJOrZ5t4lssLdxGhVrurbmBWopoEl+M6BdEfwnCJZtKXi1KgxUyJq
9        13dy" crossorigin="anonymous">
10    </head>
11    <body>
12      <div class="container">
13        {% block body %}
14        {% endblock %}
15      </div>
16    </body>
17  </html>
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Flights
5  {% endblock %}
6
7  {% block body %}
8      <h1>Book a Flight</h1>
9
10     <form action="{{ url_for('book') }}" method="post">
11
12         <div class="form-group">
13             <select class="form-control" name="flight_id">
14                 {% for flight in flights %}
15                     <option value="{{ flight.id }}">{{ flight.origin }} to {{
16                         flight.destination }}</option>
17                 {% endfor %}
18             </select>
19         </div>
20
21         <div class="form-group">
22             <input class="form-control" name="name" placeholder="Passenger Name">
23         </div>
24
25         <div class="form-group">
26             <button class="btn btn-primary">Book Flight</button>
27         </div>
28     </form>
29 {% endblock %}
30
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Flights
5  {% endblock %}
6
7  {% block body %}
8      <h1>All Flights</h1>
9
10     <ul>
11         {% for flight in flights %}
12             <li>
13                 <a href="{% url_for('flight', flight_id=flight.id) %}">
14                     {{ flight.origin }} to {{ flight.destination }}
15                 </a>
16             </li>
17         {% endfor %}
18     </ul>
19
20 {% endblock %}
21
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Flight
5  {% endblock %}
6
7  {% block body %}
8      <h1>Flight Details</h1>
9
10     <ul>
11         <li>Origin: {{ flight.origin }}</li>
12         <li>Destination: {{ flight.destination }}</li>
13         <li>Duration: {{ flight.duration }} minutes</li>
14     </ul>
15
16     <h2>Passengers</h2>
17     <ul>
18         {% for passenger in passengers %}
19             <li>{{ passenger.name }}</li>
20         {% else %}
21             <li>No passengers.</li>
22         {% endfor %}
23     </ul>
24
25 {% endblock %}
26
```

```
1  {% extends "layout.html" %}
2
3  {% block title %}
4      Error
5  {% endblock %}
6
7  {% block body %}
8      <h1>Error</h1>
9
10     {{ message }}
11 {% endblock %}
12
```