

1. Create tables

- Print 5 rows from each
- Print columns and types from each

```
✓ [8] for row in cursor.execute("SELECT * FROM flights LIMIT 5;"): print(row)
0s

('12:10', 'MUC', '19:38', 'SFO', 'UA 907 ', '763', '1', 16, 28, 16.47, None)
('10:35', 'TPE', '8:06', 'SFO', 'UA 830 ', '777', '1', 13, 31, 13.52, None)
('15:15', 'SYD', '9:38', 'SFO', 'UA 870 ', '744', 'Non-Stop', 13, 23, 13.38, None)
('20:00', 'AKL', '11:15', 'SFO', 'NZ 008 ', '772', 'Non-Stop', 12, 15, 12.25, None)
('15:50', 'MUC', '19:00', 'SFO', 'LH 458 ', '346', 'Non-Stop', 12, 10, 12.17, None)

✓ [9] for row in cursor.execute("SELECT * FROM aircrafts LIMIT 5;"): print(row)
0s

('737', 124)
('733', 124)
('319', 124)
('73G', 124)
('735', 104)

✓ [10] for row in cursor.execute("PRAGMA table_info('flights');"): print(row)
0s

(0, 'Depart_Time', 'TIME', 0, None, 0)
(1, 'Origin', 'TEXT', 0, None, 0)
(2, 'Arrival_Time', 'TIME', 0, None, 0)
(3, 'Destination', 'TEXT', 0, None, 0)
(4, 'Flight', 'VARCHAR', 0, None, 0)
(5, 'Aircraft', 'VARCHAR', 0, None, 0)
(6, 'Stops', 'VARCHAR', 0, None, 0)
(7, 'hours', 'INT', 0, None, 0)
(8, 'minutes', 'INT', 0, None, 0)
(9, 'duration', 'FLOAT', 0, None, 0)
(10, 'codeshare', 'INT', 0, None, 0)

✓ [11] for row in cursor.execute("PRAGMA table_info('aircrafts');"): print(row)
0s

(0, 'Aircraft', 'VARCHAR', 0, None, 0)
(1, 'Seats', 'INT', 0, None, 0)
```

2. Check columns with distinct values

- Print total number of rows in each
- Print number of distinct values from flights

✓
0s

```
[55] q = '''SELECT COUNT(*)  
      FROM flights;'''  
      cursor.execute(q)  
      result = cursor.fetchall()  
      print(result[-1][-1])
```

893

✓
0s



```
q = '''SELECT COUNT(*)  
      FROM aircrafts;'''  
      cursor.execute(q)  
      result = cursor.fetchall()  
      print(result[-1][-1])
```

☐→ 34

✓
0s

```
[60] q = '''SELECT COUNT(DISTINCT(Depart_Time))  
      FROM flights;'''  
      cursor.execute(q)  
      result = cursor.fetchall()  
      print(result[-1][-1])
```

286

✓
0s

```
[61] q = '''SELECT COUNT(DISTINCT(Origin))  
      FROM flights;'''  
      cursor.execute(q)  
      result = cursor.fetchall()  
      print(result[-1][-1])
```

71

✓
0s

```
[62] q = '''SELECT COUNT(DISTINCT(Arrival_Time))
      FROM flights;'''
      cursor.execute(q)
      result = cursor.fetchall()
      print(result[-1][-1])
```

302

✓
0s

```
[63] q = '''SELECT COUNT(DISTINCT(Destination))
      FROM flights;'''
      cursor.execute(q)
      result = cursor.fetchall()
      print(result[-1][-1])
      q = '''SELECT COUNT(DISTINCT(Flight))
      FROM flights;'''
      cursor.execute(q)
      result = cursor.fetchall()
      print(result[-1][-1])
      q = '''SELECT COUNT(DISTINCT(Aircraft))
      FROM flights;'''
      cursor.execute(q)
      result = cursor.fetchall()
      print(result[-1][-1])
      q = '''SELECT COUNT(DISTINCT(Stops))
      FROM flights;'''
      cursor.execute(q)
      result = cursor.fetchall()
      print(result[-1][-1])
      q = '''SELECT COUNT(DISTINCT(hours))
      FROM flights;'''
      cursor.execute(q)
      result = cursor.fetchall()
      print(result[-1][-1])
      q = '''SELECT COUNT(DISTINCT(minutes))
      FROM flights;'''
```

✓
0s



```
cursor.execute(q)
result = cursor.fetchall()
print(result[-1][-1])
q = '''SELECT COUNT(DISTINCT(duration))
FROM flights;'''
cursor.execute(q)
result = cursor.fetchall()
print(result[-1][-1])
q = '''SELECT COUNT(DISTINCT(codeshare))
FROM flights;'''
cursor.execute(q)
result = cursor.fetchall()
print(result[-1][-1])
```



```
1
869
32
2
15
56
186
1
```

3. Drop tables and create new ones with primary key
 - a. Compute number of passengers flying from DEN-SFO non-stop

```
[64] q = "DROP TABLE IF EXISTS flights;"
      cursor.execute(q) ;
      con.commit();
      q = "DROP TABLE IF EXISTS aircrafts;"
      cursor.execute(q) ;
      con.commit();

[65] q = """CREATE TABLE IF NOT EXISTS flights
      (Depart_Time TIME, Origin TEXT, Arrival_Time TIME, Destination TEXT, Flight VARCHAR, Aircraft VARCHAR, Stops VARCHAR, hours INT, minutes INT, duration FLO
      PRIMARY KEY(Depart_Time, Flight) );"""
      cursor.execute(q) ;
      con.commit();

[66] q = """CREATE TABLE IF NOT EXISTS aircrafts
      (Aircraft VARCHAR, Seats INT,
      PRIMARY KEY(Aircraft, Seats) );"""
      cursor.execute(q) ;
      con.commit();

[76] df = pd.read_csv('flights-2 (1).csv')
      df.to_sql('flights', con, if_exists='append', index = False)

[78] df = pd.read_csv('aircrafts.csv')
      df.to_sql('aircrafts', con, if_exists='append', index = False)

[79] q = """SELECT SUM(Seats)
      FROM flights LEFT JOIN aircrafts ON flights.Aircraft = aircrafts.Aircraft
      WHERE flights.Origin = 'DEN' AND flights.Destination = 'SFO' AND flights.Stops = 'Non-Stop';"""
      for row in cursor.execute(q): print(row)

      (6292,)
```

4. Find max number of passengers arriving in SFO from DEN or SAN on airplane models starting with '7'

```
[80] q = """SELECT MAX(Seats)
      FROM flights LEFT JOIN aircrafts ON flights.Aircraft = aircrafts.Aircraft
      WHERE flights.Aircraft >= 700 AND flights.Aircraft <= 799 AND flights.Destination = 'SFO' AND flights.Origin = 'DEN' OR flights.Origin = 'SAN';"""
      for row in cursor.execute(q): print(row)

      (182,)
```