

Airport Database Design and SQL

Posted on August 24, 2019

An airport database is created to store information about airport locations and aeroplanes. The information on flights, passengers, their bookings, and baggage has been stored in this database. Moreover, payment details have been stored in this database. The airport database has been created to generate reports based on the stored information. Such reports can be used for effective decision-making.

We have created a database using MYSQL. Different views have been created. We have structured SQL commands using Data Definition Language (DDL) and Data manipulation language (DML).

Data Definition Language (DDL):

Create Database:

```
CREATE DATABASE airport_database
```

Create Tables:

```
CREATE TABLE AIRPORT
```

```
(
```

```
airport_id int AUTO_INCREMENT,
```

```
airport_name varchar (100),
```

```
airport_location varchar (255),
```

```
PRIMARY KEY (airport_id)
```

```
);
```

```
CREATE TABLE checkin
```

```
(
```

```
checkin_id int AUTO_INCREMENT,
```

```
Seat_number int,
```

```
row_number int,  
  
empty_seats int,  
  
PRIMARY KEY (checkin_id)  
  
);  
  
CREATE TABLE baggage  
  
(  
  
Baggage_id INT,  
  
Baggage_Type VARCHAR (15),  
  
Baggage_weight INT,  
  
Baggage_number INT,  
  
PRIMARY KEY (Baggage_id)  
  
);  
  
CREATE TABLE AIRPLANE  
  
(  
  
Airplane_id INT AUTO_INCREMENT,  
  
Producer VARCHAR (20),  
  
Type INT,  
  
PRIMARY KEY (Airplane_id)  
  
);  
  
CREATE TABLE FLIGHT  
  
(  
  
Flight_number INT,  
  
Arrival_Time Time,  
  
Arrival_Date Date,
```

```
Departure_Time Time,  
Departure_Date Date,  
Destination VARCHAR (45),  
Airplane_id INT,  
Departure_City VARCHAR (45),  
Arrival_City VARCHAR (45),  
PRIMARY KEY (Flight_number),  
FOREIGN KEY (Airplane_id) REFERENCES Airplane(Airplane_id)  
);  
  
CREATE TABLE Passenger  
(  
    Passenger_id INT AUTO_INCREMENT,  
    First_name VARCHAR (45),  
    Last_name VARCHAR (45),  
    State VARCHAR (45),  
    City VARCHAR (45),  
    Street VARCHAR (45),  
    Age INT,  
    phone_number INT ,  
    Email VARCHAR (45),  
    PRIMARY KEY (Passenger_id)  
);  
  
CREATE TABLE booking(  
    Booking_id INT AUTO_INCREMENT,
```

```
Passanger_id INT,  
  
Flight_number INT,  
  
Seat_number INT,  
  
Baggage VARCHAR (45),  
  
Class VARCHAR (15),  
  
PRIMARY KEY (Booking_id),  
  
FOREIGN KEY(Passanger_id) REFERENCES passenger(Passenger_id),  
  
FOREIGN KEY(Flight_number) REFERENCES flight(Flight_number)  
  
);
```

```
CREATE TABLE Payment  
  
(  
  
Payment_id INT AUTO_INCREMENT,  
  
Booking_id INT,  
  
Payment_amount DATE,  
  
Payment_date TIMESTAMP,  
  
PRIMARY KEY(Payment_id),  
  
FOREIGN KEY(Booking_id) REFERENCES booking(booking_id)  
  
);
```

```
CREATE TABLE Flight_class  
  
(  
  
class_id INT AUTO_INCREMENT,  
  
class_type VARCHAR (15),  
  
flight_id int,  
  
PRIMARY key (class_id),
```

FOREIGN KEY (flight_id) REFERENCES flight(Flight_number)

);

Data Manipulation Languages (DML):

Populate Database:

Table Airplane:

```
INSERT INTO `airplane` (`Airplane_id`, `Producer`, `Type`) VALUES ('1', 'Airbus', '14'), ('2', 'Boeing', '5');
```

```
INSERT INTO `airplane` (`Airplane_id`, `Producer`, `Type`) VALUES ('3', 'Bombardian', '10'), ('4', 'Embraer', '9'), ('5', 'Tupolav', '6');
```

Airport:

```
INSERT INTO `airport` (`airport_id`, `airport_name`, `aiport_location`) VALUES ('1', 'Hartsfield-Jackson Atlanta International Airport', 'Atlanta'), ('2', 'Los Angeles International Airport', ''), ('3', 'O'Hare International Airport', 'Chicago'), ('4', 'Dallas/Fort Worth International Airport', 'Dallas/Fort Worth'), ('5', 'John F. Kennedy International Airport', 'New York'), ('6', 'Denver International Airport', 'Denver'), ('7', 'San Francisco International Airport', 'San Francisco'), ('8', 'McCarran International Airport', 'Las Vegas'), ('9', 'Charlotte Douglas International Airport', 'Charlotte'), ('10', 'Seattle-Tacoma International Airport', 'Seattle/Tacoma');
```

Baggage:

```
INSERT INTO `baggage` (`Baggage_id`, `Baggage_Type`, `Baggage_weight`, `Baggage_number`) VALUES ('1', 'Carry-on', '35', '1'), ('2', 'Carry-on', '25', '2'), ('9', 'checked-in', '30', '16'), ('10', 'checked', '50', '2'), ('3', 'checked', '40', '3'), ('4', 'carry-on', '22', '5'), ('5', 'carry-on', '23', '10'), ('6', 'carry-on', '24', '8'), ('7', 'checked-in', '42', '20'), ('8', 'checked-in', '41', '12');
```

Passenger:

```
INSERT INTO `passenger` (`Passenger_id`, `First_name`, `Last_name`, `State`, `City`, `Street`, `Age`, `phone_number`, `Email`) VALUES ('1', 'john', 'hanz', 'Alaska', 'Anchorage', 'abcd', '20', '23145638', 'john@abc.com'), ('2', 'peter', 'dave', 'Arizona', 'Phoenix', 'efg', '23', '36852149', 'peter@abc.com'), ('3', 'daren', 'James', 'California', 'Los Angeles', 'street1', '24', '32658974',
```

```
'daren@abc.com'), ('4', 'sammy', 'benz', 'Colorado', 'Denver', 'street2', '32', '32145236',
'sammy@abc.com'), ('5', 'steve', 'george', 'Florida', 'Miami', 'street4', '36', '36524789',
'steve@abc.com'), ('6', 'james', 'bind', 'Georgia', 'Atlanta', 'street5', '32', '56985214',
'james@abc.com'), ('7', 'george', 'Michael', 'Illinois', 'Springfield', 'street1', '35', '36541278',
'george@abc.com'), ('8', 'william', 'jobs', 'District of Columbia', 'Washington', 'street2', '45',
'36985214', 'william@abc.com'), ('9', 'johnson', 'will', 'Iowa', 'Davenport', 'street7', '65', '65478236',
'johnson@abc.com'), ('10', 'john', 'steve', 'Florida', 'Orlando', 'street10', '36', '65498256',
'john@abc.com');
```

Flight:

```
INSERT INTO `flight` (`Flight_number`, `Arrival_Time`, `Arrival_Date`, `Departure_Time`,
`Departure_Date`, `Destination`, `Airplane_id`, `Departure_City`, `Arrival_City`) VALUES ('1',
'02:00:00', '2018-04-03', '05:00:00', '2018-04-03', 'Orlando', '1', 'Atlanta', 'Chicago'), ('2', '04:00:00',
'2018-04-05', '11:00:00', '2018-04-04', 'Miami', '2', 'Los angeles', 'Orlando'), ('3', '02:23:56',
'2018-04-03', '03:00:00', '2018-04-03', 'indianapolis', '2', 'chicago', 'springfield'), ('4', '01:00:00',
'2018-04-09', '05:00:00', '2018-04-09', 'Miami', '3', 'Dover', 'Washington'), ('5', '03:30:00',
'2018-04-12', '04:00:00', '2018-04-12', 'Hazard', '3', 'sprinfield', 'indianapolis'), ('6', '12:00:00',
'2018-04-13', '01:00:00', '2018-04-14', 'Boston', '4', 'freeport', 'baltimon'), ('7', '06:00:00',
'2018-04-19', '07:00:00', '2018-04-20', 'Pensacola', '1', 'Beverly hills', 'denver'), ('8', '08:00:00',
'2018-04-18', '09:00:00', '2018-04-18', 'St. Louis', '2', 'Dulux', 'bilothi'), ('9', '10:00:00', '2018-04-23',
'11:00:00', '2018-04-23', 'New Orleans', '5', 'miami', 'ORLANDO'), ('10', '09:00:00', '2018-04-26',
'12:00:00', '2018-04-28', 'Logan', '5', 'Autin', 'Camden');
```

Flight Class:

```
INSERT INTO `flight_class` (`class_id`, `class_type`, `flight_id`) VALUES ('1', 'first class', '1'), ('2',
'business class', '1'), ('3', 'economy class', '1'), ('4', 'First class', '2'), ('5', 'business class', '2'), ('6',
'economy class', '2'), ('7', 'first class', '3'), ('8', 'business class', '3'), ('9', 'economy class', '3'), ('10',
'economy class', '4');
```

Checkin:

```
INSERT INTO `checkin` (`checkin_id`, `Seat_number`, `row_number`, `empty_seats`) VALUES ('1', '1',
'1', '5'), ('2', '4', '2', '4'), ('3', '3', '3', '6'), ('4', '4', '1', '5'), ('5', '12', '2', '6'), ('6', '13', '1', '7'), ('7',
'15', '2', '5'), ('8', '16', '2', '6'), ('9', '17', '3', '3'), ('10', '19', '2', '4');
```

Booking:

```
INSERT INTO `booking` (`Booking_id`, `Passanger_id`, `Flight_number`, `Seat_number`, `Baggage`,
```

```
`Class`) VALUES ('1', '1', '1', '12', 'carry-on', 'First class'), ('2', '2', '1', '13', 'checked-in', 'economy class'), ('3', '3', '2', '10', 'carry-on', 'first class'), ('4', '4', '2', '5', 'carry-on', 'business class'), ('5', '5', '3', '9', 'carry-on', 'business class'), ('6', '6', '3', '6', 'carry-on', 'business class'), ('7', '7', '4', '5', 'carry-on', 'business class'), ('8', '8', '4', '6', 'carry-on', 'business class'), ('9', '9', '5', '9', 'carry-on', 'economy class'), ('10', '10', '7', '2', 'carry-on', 'business class');
```

Payment:

```
ALTER TABLE payment
```

```
ALTER COLUMN Payment_amount INT
```

```
INSERT INTO `payment` (`Payment_id`, `Booking_id`, `Payment_date`, `Payment_amount`) VALUES ('1', '2', '2018-04-03 03:00:00', '100'), ('2', '1', '2018-04-03 05:08:16', '150'), ('3', '3', '2018-04-04 10:00:00', '100'), ('4', '4', '2018-04-05 03:00:00', '120'), ('5', '5', '2018-04-08 06:00:00', '130'), ('6', '6', '2018-04-10 11:00:00', '100'), ('7', '7', '2018-04-07 12:00:00', '130'), ('8', '8', '2018-04-20 04:00:00', '100'), ('9', '9', '2018-04-24 05:00:00', '110'), ('10', '10', '2018-04-14 00:00:00', '120');
```

Test Cases:

Test Case # 1:

Retrieve aeroplanes and airport information. The information contains the airplane's name and type as well as its airport name and location.

Expected Result:

The information will be retrieved containing the airplane name and type, as well as its airport name and location.

Actual Results:

As Expected.

+ Options

 		name	type	airport_name	aiport_location
☐  Edit  Copy  Delete		Airbus	14	Los Angeles International Airport	Los angeles
☐  Edit  Copy  Delete		Boeing	5	O'Hare International Airport	Chicago
☐  Edit  Copy  Delete		Bombardian	10	Hartsfield–Jackson Atlanta International Airport	Atlanta
☐  Edit  Copy  Delete		Embraer	9	Dallas/Fort Worth International Airport	Dallas/Fort Worth
☐  Edit  Copy  Delete		Tupolav	6	Los Angeles International Airport	Los angeles

Test Pass/Fail:

Pass.

Test Case # 2:

Retrieve airport information located in New York and Chicago.

Expected Result:

The information will be retrieved and include details of airports situated in New York and Chicago.

Actual Result:

As Expected.

+ Options

				airport_name	airport_location
☐	Edit	Copy	Delete	O'Hare International Airport	Chicago
☐	Edit	Copy	Delete	John F. Kennedy International Airport	New York

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Export

Test Pass/Fail:

Pass.

Test Case # 3:

Retrieve information related to passengers and their flight bookings.

Expected Result:

The information will be generated by containing data about passengers and their flight bookings.

Actual Result:

+ Options

		Booking_id	First_name	Class	Seat_number	Payment_date	Payment_amount
☐	Edit Copy Delete	2	peter	economy class	13	2018-04-03 03:00:00	100
☐	Edit Copy Delete	1	john	First class	12	2018-04-03 05:08:16	150
☐	Edit Copy Delete	3	daren	first class	10	2018-04-04 10:00:00	100
☐	Edit Copy Delete	4	sammy	business class	5	2018-04-05 03:00:00	120
☐	Edit Copy Delete	5	steve	business class	9	2018-04-08 06:00:00	130
☐	Edit Copy Delete	6	james	business class	6	2018-04-10 11:00:00	100
☐	Edit Copy Delete	7	george	business class	5	2018-04-07 12:00:00	130
☐	Edit Copy Delete	8	william	business class	6	2018-04-20 04:00:00	100
☐	Edit Copy Delete	9	johnson	economy class	9	2018-04-24 05:00:00	110
☐	Edit Copy Delete	10	john	business class	2	2018-04-14 00:00:00	120

☐ Check all With selected: Edit Copy Delete Export

Test Pass/Fail:

Pass.

Test case #4:

The information retrieved contained the Airbus flight details.

Expected Result:

The information will contain the details about the Airbus flights.

Actual Result:

As expected.

+ Options

		Flight_number	name	class
☐	Edit Copy Delete	1	Airbus	first class
☐	Edit Copy Delete	7	Airbus	first class
☐	Edit Copy Delete	1	Airbus	business class
☐	Edit Copy Delete	7	Airbus	business class
☐	Edit Copy Delete	1	Airbus	economy class
☐	Edit Copy Delete	7	Airbus	economy class

☐ Check all With selected: Edit Copy Delete Export

Test Pass/Fail:

Pass.

Test Case # 5:

The information retrieved contains passenger, booking and baggage details.

Expected Result:

The information will contain details about the passenger, booking, and baggage.

Actual Result:

As expected.

+ Options

		Booking_id	First_name	Seat_number	Class	Baggage_Type	Baggage_number	Baggage_weight
☐	Edit Copy Delete	1	john	12	First class	Carry-on	1	35
☐	Edit Copy Delete	2	peter	13	economy class	checked	3	40
☐	Edit Copy Delete	3	daren	10	first class	Carry-on	2	25
☐	Edit Copy Delete	4	sammy	5	business class	carry-on	8	24
☐	Edit Copy Delete	5	steve	9	business class	carry-on	10	23
☐	Edit Copy Delete	6	james	6	business class	carry-on	8	24
☐	Edit Copy Delete	7	george	5	business class	checked-in	20	42
☐	Edit Copy Delete	8	william	6	business class	Carry-on	1	35
☐	Edit Copy Delete	9	johnson	9	economy class	Carry-on	1	35
☐	Edit Copy Delete	10	john	2	business class	Carry-on	1	35

☐ Check all With selected: Edit Copy Delete Export

Test Pass/Fail:

Pass.

Conclusion:

We have created the [airport database](#) using MYSQL. We have implemented Queries using DDL and DML. Multiple queries have been created and executed. The results have been analyzed. The actual results are according to the expected results, and hence, we have created and linked tables by following the relationship rules. We have tried to normalize the database structure to avoid complex and unexpected results of SQL queries. We have constructed efficient queries to generate multiple views so that we can manipulate those views in future according to the changing requirements.

References

DML, DDL, DCL and TCL Statements in SQL with Examples. (2018). Retrieved from tech-recipes website: <https://www.tech-recipes.com/rx/55356/dml-ddl-dcl-and-tcl-statements-in-sql-with-examples/>

SQL FOREIGN KEY Constraint. (2018). Retrieved from w3schools website: https://www.w3schools.com/sql/sql_foreignkey.asp