Documentation to the database

Airport maintenance

Authors: Ariela Šťastná (231151), Pavel Ostrý (230624)

Subject: Bezpečnost databázových systémů

Academic year: 2021/2022 winter semester

Links to repositories: <https://github.com/ArielaStastna/BPC-BDS-DB-project>

<https://github.com/OstryP/bds-db-design>

Application description

Coronavirus has put us into a non-easy situation. We could travel only in our imagination and usually the furthest distance we travelled was the local supermarket. We have already forgotten even how to travel, so let´s remind it and recollect together about the travelling passion, we used to take for granted. We have chosen to create a database of an airport. We created 12 tables with the most important tasks an airport handles with. Our aim is to ensure, that all people will get on the right flight – the flight they booked earlier and will have a pleasant journey and their luggage will get to the same place and won´t get lost. We also have to differentiate between various airlines, because one destination can be reached by many airlines, that can vary in quality, comfort or flight time. We also have to prevent that nobody is going to get hurt as long as all people have to undergo the security check and in case of any suspicions, they won´t be let on board, also we created another safety measure and created a no-fly list, a list of people suspicious from terroristic intentions. In addition, according to booking, we created a flight manifest, that consists of people that are present on the flight, including both passengers and crew. In case of emergency, these lists are very important, because they provide information about the real state and occupancy of the flight and help lifeguards to make an overview of missing people. Boarding passes are issued based on booking after the baggage check and serve as a pass on the flight board. Since we are creating an airport database and keeping the schedule is very important here, we use a lot of timestamp data types.   
The database is created in PostgreSQL and MySQL. The ERD diagram was created in MySQL Workbench.

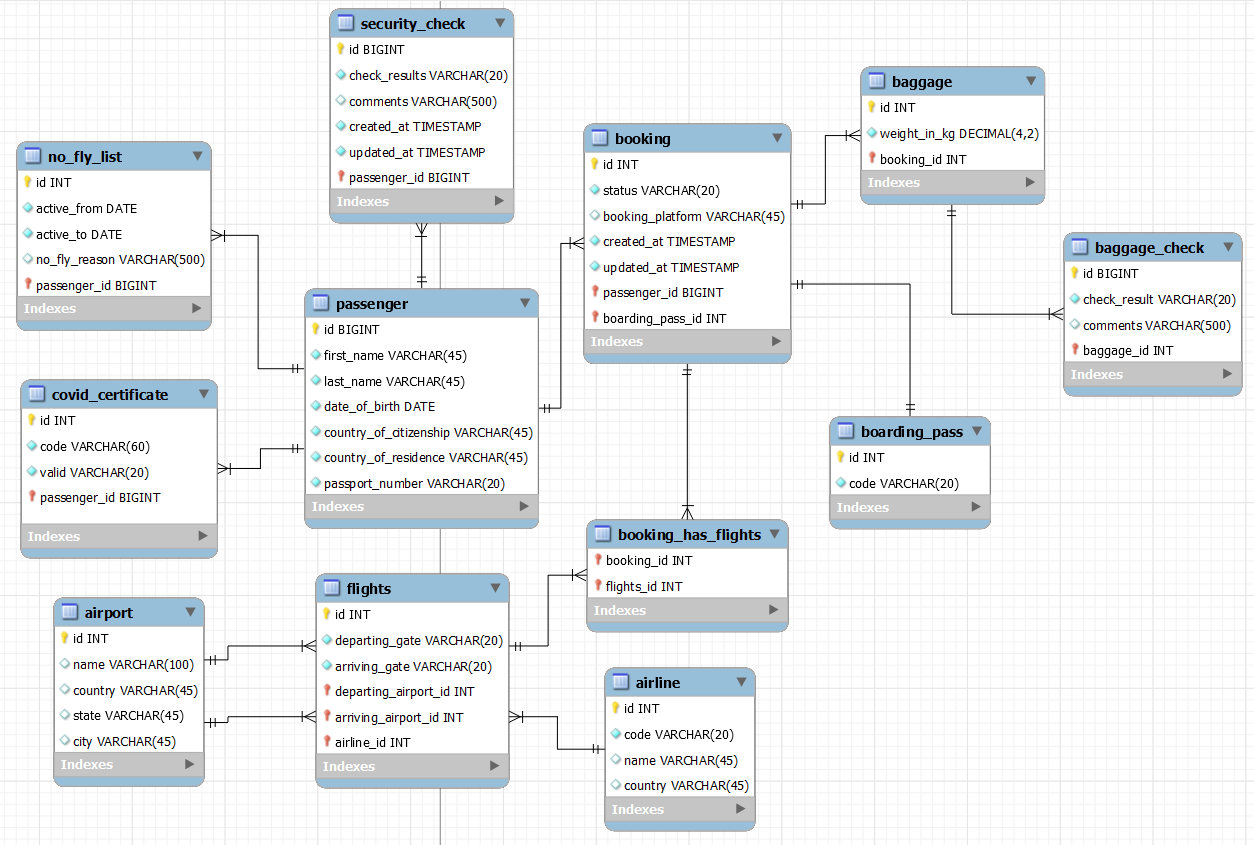


Table description

Passenger

Table Passenger is used for gathering all passengers that have a scheduled flight and servers for their identification. The table consists of passenger id, that is represented by columns id, first\_name, last\_name, date\_of\_birth, country\_of\_citizenship, country\_of\_residence, passport\_number. Id is the primary key in the table Passengers, used for differentiation of passengers. We chose the data type bigint, as long as the airport daily deals with thousands of passengers and we need range, big enough, to cover all of them. Column first\_name contains information about passenger´s first name and is a varchar (45) data type. Varchar is a data type for variable-size strings. We set the limit on 45, because we do not really expect any longer first names. Last\_name is the same case as first\_name, with the only difference, that it contains passenger´s last name. Also set on varchar to enable various string lengths, but maximally 45. Column date\_of\_birth represents the passenger´s date of birth and also serves as basic data about the passenger. Data type is, of course, date, that stores passenger´s day, month and year of birth. Country\_of\_citizenship informs about passenger´s citizenship and is a data type varchar, also with the limit 45 characters. Country\_of\_residence informs about passenger´s permanent residence and is represented by varchar, as long as names are always represented by a string. Passport\_number is used for authentication of passengers and passport numbers are represented by a string of characters and numbers. These strings can vary by lengths from state to state, that´s why we still need to use a varchar data type.

Airport

Table airport contains information about the final destination a passenger wants to take flight to. The table airport is in a one-to-many relationship with table flights. This means that one flight can end up only at one airport (e.g. flight Vienna – Paris Charles de Gaulle arrives at this certain airport), but many flights land in the same airport (Vienna airport offers flights to 141 destinations). The table consists of attributes id, name, state and city. Id is an identification number of each airport and it´s an int data type that represents a number sequence. It servers as a primary key. Name of the airport is important to enter, because big cities usually have more than one airport, so further specification is required. Data type is varchar (45) because the column is filled with strings. Column state is an information about the state the airport takes place in. Data type is also varchar because data are represented by a text string. Cities can have same names, so we can´t forget to enter the state, but also one state can have airports in different cities, so all these attributes are necessary. Column city reveals information about the city airport takes place in. Data type is varchar for text strings.

Airline

Table airline provides information about airlines that mediate flights. Airline is in an one-to-many relationship with flights. One exact flight is arranged by only one airline (e.q. flight FR8343 is a 9:25 Ryanair flight from London – Stansted to Porto), but airlines arrange many different flights (Ryanair offers flights to over 200 destinations). Primary key id serves for number identification of the airline, that´s why its data type is int. Code is a unified code, that substitutes the official airline name, can consists of letters, numbers and characters, so we chose varchar (example: Czech airlines have code IATA-064 or IATA-OK). Name is the airline name and a varchar data type – can consist of letters, numbers and characters. Country is the country of origin of the certain airline, also varchar for text strings.

Baggage

Table baggage connects passengers with their baggage via booking. Baggage is in one-to-many relationship with booking. One person is able through one booking book more baggage (for example father of the family books 223kg baggage on his name), but one baggage can have only one owner (the person who made the booking). Id is the primary key of this table and is an int data type. Weight\_in\_kg carries information about the weight of the baggage, that was weighed in the check-in table and the result is in decimal. Notation (4,2) indicates, that the number will have 2 digits before the decimal and 2 digits after the decimal. Booking\_id is a unique sequence of numbers generated while booking, that contains the reservation information and is a bigint data type and a foreign key.

Baggage\_check

Table baggage\_check refers to the security check that is made after giving away our baggage at the check-in desk. This security check is performed without us being present. Baggage\_check is in one-to-many relationship with baggage. One baggage goes on only one baggage check, but one baggage check has to screen a lot of baggage. The table consists of big integer id, which is also a primary key. Check\_result contains information, whether the baggage complies or not complies the security measures, the result is stated as varchar. Comments provide further information about the check result and is represented by a varchar data type. Baggage\_id is a foreign key.

Flights

Table flights contains the flight information about gates, airlines and airports. Flight table is in one-to-many relationship with airport and airline and in many-to-many relationship with booking. Id is the primary key that identifies flights. The data type is int because we are dealing with number representations. Departing\_gate and arriving\_gate are the gates where passengers embark and disembark. They are usually represented by a letter and a 2 digit number (e.g.D64), that´s why it´s a varchar. Departing\_airport\_id and arriving\_airport\_id are a number representation of departing and arriving airports, data type is int. Airline\_id is the identification number of the airline that takes care of the transportation of passengers, it´s an int data type. Departing\_airport\_id, arriving\_airport\_id and airline\_id are all foreign keys.

Booking

Table booking is in a many-to-many relationship with flights. You can have more flights on one booking (for example when travelling from Vienna to Dubai with a transfer in Warsaw) and also you can have many bookings from different people on the same flight. Id (int) is the table´s primary key. Flight\_id (varchar) is the unique identification sequence of the flight. Booking status is a varchar providing information about the actual state of our booking – can be for example waiting, cancelled, confirmed. Booking platforms are the mediators of bookings, such as kiwi.com, pelikan.cz, that take care of our flight bookings, they are a varchar data type. Timestamps created\_at and updated\_at inform about the time booking was done and updated (e.g. when someone buys an extra luggage the day before departure). Passenger\_id (bigint) is a foreign key.

Booking\_has\_flights

Table booking\_has\_flights serves as an intertable between booking and flights, due to their many-to-many relationship. It consists of 2 foreign keys: booking\_id (int) from the table booking and flight\_id (int) from the table flights.

Covid\_certificate

Table covid\_certificate is in one to many relationship with table passenger. One passenger can own more certificates, for example, a person who overcame coronavirus can also own a certificate about vaccination. But one document can be owned only by one certain person. The table´s primary key is an id represented by an integer data type. Code is a specific sequence that serves as a verification of person´s certificate, it´s a varchar. Column valid approves or declines the validity of the certificate. For example, if the person´s PCR test is older than 72 hours, the validity is evaluated as non-valid. The data type varchar gains result either valid or non-valid. To connect people with their certificate, we use the passenger\_id as a foreign key and a bigint data type.

No\_fly\_list

No\_fly\_list is a table listing people, suspicious from terroristic intentions, that can not be let on board. No\_fly\_list is in an one to many relationship with the table passenger. One passenger can be on the list only once, but the list consists of many passengers. Id is the number primary key of the table. Active\_from and active\_to is a date data type, that refers to the date since when and until when is the person on the no fly list. No\_fly\_reason is a varchar data type, that contains information about the case and the reason why to person figures in this list. Passenger\_id (bigint) is the foreign key of no\_fly\_list table.

Boarding\_pass

Table boarding\_pass is in a one-to-one relationship with booking table. Boarding passes are given out based on bookings. Primary key of the table is id, in the data type int. Code (varchar) of the boarding pass is a 6 digit alphanumeric code generated after booking. Booking\_id (int) is the foreign key of the table.

Security\_check

Security\_check is in one to many relationship with table passenger. All passengers have to undergo the security check, but there is only one common security check. Id (bigint) is the identification number of compartments performing the security check and is the table´s primary key. The check\_result can be either positive or negative based on the result of the scanning technology and body search, this information is signed into varchar data type. Comments contain information about the reason someone failed the security check, the reason why the person had to be double-checked (for example because the person forgot to take out their wallet from the pocket) and an information whether any belongings were taken away from the person (for example some sharp tools as nail files or manicure scissors). Created\_at and updated\_at are timestamps, containing information about the time the security check was done. Passenger\_id is the foreign key.

Third normal form  
We put our best effort to create a flexible database that considers every case by eliminating redundancy and dependency of data. The database is in the first normal form, because its elements are atomic (for example in the table passenger, we did not create column passenger\_info that would store all passenger information, but we created single elements such as first\_name, last\_name, date\_of\_birth, country\_of\_citizenship, country\_of\_residence and passport\_number) – there are examples of keeping the first normal form in every table. Database is in the second normal form because it follows the first normal form and we also removed partial dependencies by creating single tables and making every non-key attribute fully dependent on primary key. The second normal form would be broken if we created for example a table with passenger ids and passenger names (name can be determined as id) together with covid certificate id and code. To avoid breaking the second normal form, we created separate tables with full dependencies and did not mix data inappropriately. If we created only one table, where flights and airport would figure together and the primary key would be flight\_id, we would definitely break the third normal form, because airport id and name would be in a transitive relationship, that is in contradiction with the third normal form definition. To avoid this, we have to create separate table for airport, where id will be a primary key. We followed this rule together with following the second normal form, so the third normal form is also kept.

MySQL DML script

INSERT INTO `airline` VALUES (1,'OK','Czech Airlines','Czech Republic'),(2,'DL','Delta Air Lines','United States'),(3,'EW','Eurowings','Germany'),(4,'MU','China Eastern Airlines','China'),(5,'BA','British Airways','United Kingdom'); INSERT INTO `airport` VALUES (1,'Guangzhou Baiyun International Airport','China','Guangdong','Guangzhou'),(2,'Hartsfield–Jackson Atlanta International Airport','United States','Georgia','Atlanta'),(3,'Chengdu Shuangliu International Airport','China','Sichuan','Chengdu'),(4,'Dallas/Fort Worth International Airport','United States','Texas','Dallas'),(5,'Shenzhen Bao\'an International Airport','China','Guangdong','Shenzhen'),(6,'Chongqing Jiangbei International Airport','China','Chongqing','Yubei'),(7,'Beijing Capital International Airport','China','Beijing','Beijing'),(8,'Denver International Airport','United States','Colorado','Denver'),(9,'Kunming Changshui International Airport','China','Yunnan','Kunming'),(10,'Shanghai Hongqiao International Airport','China','Shanghai','Shanghai'),(11,'Xi\'an Xianyang International Airport','China','Shaanxi','Xianyang'),(12,'Tokyo Haneda Airport','China','Tokyo','Tokyo'),(13,'O\'Hare International Airport','United States','Illinois','Chicago'),(14,'Shanghai Pudong International Airport','China','Shanghai','Shanghai'),(15,'Los Angeles International Airport','United States','California','Los Angeles'),(16,'Indira Gandhi International Airport','India','Delhi','Delhi'),(17,'Hangzhou Xiaoshan International Airport','China','Zhejiang','Hangzhou'),(18,'Charlotte Douglas International Airport','United States','North Carolina','Charlotte'),(19,'Dubai International Airport','United Arab Emirates','Dubai','Dubai'),(20,'Istanbul Airport','Turkey','Istanbul','Istanbul'),(21,'Charles de Gaulle Airport','France','Île-de-France','Paris'),(22,'Heathrow Airport','United Kingdom','London','London'),(23,'Mexico City International Airport','Mexico','Mexico City','Mexico City'),(24,'Phoenix Sky Harbor International Airport','United States','Arizona','Phoenix'),(25,'Tan Son Nhat International Airport','Vietnam','Ho Chi Minh City','Ho Chi Minh City'),(26,'Orlando International Airport','United States','Florida','Orlando'),(27,'Zhengzhou Xinzheng International Airport','China','Henan','Zhengzhou'),(28,'Jeju International Airport','South Korea','Jeju Province','Jeju City'),(29,'Amsterdam Airport Schiphol','Netherlands','North Holland','Haarlemmermeer'),(30,'São Paulo/Guarulhos International Airport','Brazil','Guarulhos','Guarulhos'),(31,'Seattle–Tacoma International Airport','United States','Washington','SeaTac'),(32,'Nanjing Lukou International Airport','China','Jiangsu','Nanjing'),(33,'Sheremetyevo International Airport','Russia','Moscow Oblast','Khimki'),(34,'Changsha Huanghua International Airport','China','Hunan','Changsha'),(35,'Frankfurt Airport','Germany','Hesse','Frankfurt'),(36,'Miami International Airport','United States','Florida','Miami'),(37,'George Bush Intercontinental Airport','United States','Texas','Houston'),(38,'Gimpo International Airport','South Korea','Seoul','Seoul'),(39,'Madrid Barajas Airport','Spain','Madrid','Madrid'),(40,'Istanbul Sabiha Gökçen International Airport','Turkey','Istanbul','Istanbul'),(41,'Xiamen Gaoqi International Airport','China','Fujian','Xiamen'),(42,'Suvarnabhumi Airport','Thailand','Samut Prakan','Bang Phli'),(43,'John F. Kennedy International Airport','United States','New York','New York'),(44,'Guiyang Longdongbao International Airport','China','Guizhou','Guiyang'),(45,'Haikou Meilan International Airport','China','Hainan','Haikou'),(46,'Fort Lauderdale–Hollywood International Airport','United States','Florida','Broward County'),(47,'San Francisco International Airport','United States','California','San Mateo Country'),(48,'Moscow Domodedovo Airport','Russia','Moscow Oblast','Domodedovo'),(49,'Beijing Daxing International Airport','China','Beijing','Beijing'),(50,'Newark Liberty International Airport','United States','New Jersey','Newark');

INSERT INTO `baggage` VALUES (1,3.25,1),(2,2.13,1),(3,5.58,2),(4,3.07,2),(5,7.17,4);

INSERT INTO `baggage\_check` VALUES (1,'OK',NULL,1),(2,'OK',NULL,2),(3,'OK',NULL,3),(4,'PENDING','found fruit and vegetable',4),(5,'FAILED','not proper type of baggage',5);

INSERT INTO `boarding\_pass` VALUES (1,'lapsingw5vhj59i8h2xd'),(2,'6fea8nhpz0q3a5c82d5e'),(3,'qb5c6az8b5q3y6s5d8bb'),(4,'33agrtmhfcqb5c6az8b5'),(5,'6adcvbttrfasf58ecvbg'),(6,'saf5e8asc2ert9vfastg'),(7,'dalgfivclst58x6v2f4a');

INSERT INTO `booking` VALUES (1,'confirmed','booking.com','2021-02-07 23:00:00','2021-02-07 23:00:00',7,1),(2,'confirmed','booking.com','2021-06-23 19:59:30','2021-06-24 06:22:30',8,2),(3,'pending','booking.com','2021-11-01 21:35:27','2021-11-01 21:35:27',39,3),(4,'confirmed','booking.com','2021-08-02 23:22:04','2021-08-02 23:22:04',16,4),(5,'cancelled','booking.com','2021-08-09 10:21:30','2021-09-01 15:12:30',3,5),(6,'confirmed','booking.com','2021-12-01 22:25:27','2021-12-01 22:25:27',9,6),(7,'confirmed','booking.com','2021-09-03 09:22:04','2021-09-03 09:22:04',10,7); INSERT INTO `booking\_has\_flights` VALUES (1,1),(2,2),(4,3),(6,4),(7,5); INSERT INTO `covid\_certificate` VALUES (1,'URN:UVCI:01:CZ:EHSU2S4F8YYYQVKMW5DF7BDYGEACB9D2','YES',1),(2,'URN:UVCI:01:CZ:EHSU2SAFKLMSDNFNA5DF7BDYGEASASCVE','YES',6),(3,'URN:UVCI:03:CZ:ASDJKLNKMNASMNDKW5DF7BDYGEASPXIQ2','YES',7),(4,'URN:UVCI:11:CZ:EHSWASDMNWAMNIXKWJA7BDYGEASA3Q5','NO',13),(5,'URN:UVCI:05:CZ:EHSU2S4F8YYYQVKMW5DASDAWKJJSKWJS','YES',8);

INSERT INTO `flights` VALUES (1,'J44','A13',1,9,2),(2,'C23','B19',6,7,3),(3,'B1','A8',34,20,4),(4,'C5','A1',1,2,5),(5,'A7','A2',14,34,4);

INSERT INTO `no\_fly\_list` VALUES (1,'2020-03-15','2120-03-15','suspected member of terorist organization',43),(2,'2017-07-19','2117-07-19','member of terorist organization',28),(3,'2003-01-01','2006-01-01','insulting other passangers during flight',38),(4,'2005-11-12','2010-11-12','damaging airline seats during flight',40),(5,'2011-12-11','2051-12-11','dangerous criminal',46);

INSERT INTO `passenger` VALUES (1,'Pavel','Ostrý','2001-03-21','Czech Republic','Czech Republic','16851250'),(2,'Karolína','Kopřivová','1984-10-06','Czech Republic','Czech Republic','95723548'),(3,'Věra','Štarhová','1989-04-25','Czech Republic','Czech Republic','68745135'),(4,'Marie','Hrbáčová','1939-04-01','Czech Republic','Czech Republic','54862418'),(5,'Tereza','Kadlečková','1953-02-06','Czech Republic','Czech Republic','85245600'),(6,'Michael','Dohnal','1990-02-22','Czech Republic','Czech Republic','21557896'),(7,'Aleš','Mach','1955-08-19','Czech Republic','Czech Republic','78965412'),(8,'Pavel','Mach','1982-03-08','Czech Republic','Czech Republic','35412856'),(9,'Leona','Ottová','1996-11-22','Czech Republic','Czech Republic','32165485'),(10,'Elena','Juráková','1985-12-06','Czech Republic','Czech Republic','84563572'),(11,'Rostislav','Bartík','1979-07-15','Czech Republic','Czech Republic','55588866'),(12,'Helena','Štorkánová','2009-12-04','Czech Republic','Czech Republic','65488749'),(13,'Marie','Petrželová','1973-10-25','Czech Republic','Czech Republic','35445123'),(14,'Miroslav','Bošek','1978-09-13','Czech Republic','Germany','65485521'),(15,'Miroslav','Černý','1996-07-07','Czech Republic','Canada','33215789'),(16,'Jiřina','Hermanová','2007-06-29','Czech Republic','Brazil','54877862'),(17,'David','Novák','1997-04-20','Czech Republic','France','24158796'),(18,'František','Marek','1969-02-21','Czech Republic','Austria','95632157'),(19,'Monika','Pechanová','1997-12-14','Czech Republic','Estonia','22115588'),(20,'Michala','Jeriová','1990-07-05','Czech Republic','Tunisia','35125870'),(21,'Louis','Eusebio','1971-03-03','United States','United States','57803651'),(22,'Toya','Benally','1965-05-27','United States','United States','45122500'),(23,'Katie','Raymond','1989-04-12','United States','United States','32521356'),(24,'Christiana','Summerfield','1995-06-11','United States','United States','65488004'),(25,'Marry','Puccio','2003-12-21','United States','United States','54009573'),(26,'Larry','Crowley','2005-06-09','United States','United States','65248900'),(27,'James','Jones','1987-11-16','United States','Poland','20062048'),(28,'Ed','Jason','1963-07-03','United States','Tunisia','65002587'),(29,'Eugine','Conrad','1988-09-11','United States','Sweden','97060842'),(30,'John','Houston','1998-11-04','United States','Switzeland','03654028'),(31,'George','Savage','1983-11-05','United States','United Kingdom','60054826'),(32,'Elliot','Reyes','1982-08-22','United States','Belgium','89632175'),(33,'Neil','Harris','1958-10-28','United States','Brazil','00585200'),(34,'Edward','Bray','1974-01-23','United States','Austria','65090275'),(35,'Manuela','Nagel','1957-10-03','Germany','Germany','15589635'),(36,'Thomas','Fenstermacher','1996-05-05','Germany','Germany','32896540'),(37,'Uta','Farber','2006-07-13','Germany','Germany','59674005'),(38,'Sussane','Eggers','1954-02-22','Germany','Germany','68000528'),(39,'Marco','Friedman','1952-12-23','Germany','Germany','20080540'),(40,'Marcel','Gerste','1992-09-26','Germany','Germany','97080506'),(41,'Monika','Herzog','1969-03-21','Germany','Germany','56984000'),(42,'Swen','Seiler','1980-04-18','Germany','Czech Republic','00580548'),(43,'Philipp','Herman','1949-06-16','Germany','Poland','50487365'),(44,'Vanessa','Kaufmann','1963-08-02','Germany','Estonia','60154780'),(45,'Kristian','Voglet','1967-03-27','Germany','Finland','02459024'),(46,'Sabrina','Fuerst','1994-02-23','Germany','France','06900582'),(47,'Lena','Hertzog','1998-05-19','Germany','Spain','99085460'),(48,'Florian','Neudorf','1991-09-20','Germany','Sweden','30069087'),(49,'Leon','Gerste','1957-11-01','Germany','Portugal','30685420'),(50,'Ralph','Kastner','1966-02-12','Germany','Italy','56804569');

INSERT INTO `security\_check` VALUES (1,'OK',NULL,'2021-10-31 23:00:00','2021-10-31 23:00:00',7),(2,'OK',NULL,'2021-10-28 22:00:00','2021-10-28 22:00:00',8),(3,'OK',NULL,'2021-10-19 22:00:00','2021-10-19 22:00:00',16),(4,'OK',NULL,'2021-10-06 22:00:00','2021-10-06 22:00:00',9),(5,'OK',NULL,'2021-10-24 22:00:00','2021-10-24 22:00:00',10);

|  |
| --- |
|  |
|  |  |

PostgreSQL DDL script

|  |
| --- |
| CREATE TABLE passenger ( |
|  | id bigint NOT NULL, |
|  | first\_name varchar(45) NOT NULL, |
|  | last\_name varchar(45) NOT NULL, |
|  | date\_of\_birth date NOT NULL, |
|  | country\_of\_citizenship varchar(45) NOT NULL, |
|  | country\_of\_residence varchar(45) NOT NULL, |
|  | passport\_number varchar(20) NOT NULL, |
|  | PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE security\_check ( |
|  | id bigint NOT NULL, |
|  | check\_results varchar(20) NOT NULL, |
|  | comments varchar(500), |
|  | created\_at timestamp NOT NULL, |
|  | updated\_at timestamp NOT NULL, |
|  | passenger\_id bigint NOT NULL, |
|  | PRIMARY KEY (id), |
|  | FOREIGN KEY (passenger\_id) REFERENCES passenger (id) |
|  | ); |
|  |  |
|  | CREATE TABLE no\_fly\_list ( |
|  | id int NOT NULL, |
|  | active\_from date NOT NULL, |
|  | active\_to date NOT NULL, |
|  | no\_fly\_reason varchar(500), |
|  | passenger\_id bigint NOT NULL, |
|  | PRIMARY KEY (id), |
|  | FOREIGN KEY (passenger\_id) REFERENCES passenger (id) |
|  | ); |
|  |  |
|  | CREATE TABLE covid\_certificate ( |
|  | id int NOT NULL, |
|  | code varchar(60) NOT NULL, |
|  | valid varchar(20) NOT NULL, |
|  | passenger\_id bigint NOT NULL, |
|  | PRIMARY KEY (id), |
|  | FOREIGN KEY (passenger\_id) REFERENCES passenger (id) |
|  | ); |
|  |  |
|  | CREATE TABLE boarding\_pass ( |
|  | id int NOT NULL, |
|  | code varchar(20) NOT NULL, |
|  | PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE booking ( |
|  | id int NOT NULL, |
|  | status varchar(20) NOT NULL, |
|  | booking\_platform varchar(45), |
|  | created\_at timestamp NOT NULL, |
|  | updated\_at timestamp NOT NULL, |
|  | passenger\_id bigint NOT NULL, |
|  | boarding\_pass\_id int NOT NULL, |
|  | PRIMARY KEY (id), |
|  | FOREIGN KEY (boarding\_pass\_id) REFERENCES boarding\_pass (id), |
|  | FOREIGN KEY (passenger\_id) REFERENCES passenger (id) |
|  | ); |
|  |  |
|  | CREATE TABLE baggage ( |
|  | id int NOT NULL, |
|  | weight\_in\_kg decimal(4,2) NOT NULL, |
|  | booking\_id int NOT NULL, |
|  | PRIMARY KEY (id), |
|  | FOREIGN KEY (booking\_id) |
|  | REFERENCES booking (id) |
|  | ); |
|  |  |
|  | CREATE TABLE baggage\_check ( |
|  | id bigint NOT NULL, |
|  | check\_result varchar(20) NOT NULL, |
|  | comments varchar(500), |
|  | baggage\_id int NOT NULL, |
|  | PRIMARY KEY (id), |
|  | FOREIGN KEY (baggage\_id) REFERENCES baggage (id) |
|  | ); |
|  |  |
|  | CREATE TABLE airport ( |
|  | id int NOT NULL, |
|  | name varchar(100), |
|  | country varchar(45), |
|  | state varchar(45), |
|  | city varchar(45), |
|  | PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE airline ( |
|  | id int NOT NULL, |
|  | code varchar(20) NOT NULL, |
|  | name varchar(45), |
|  | country varchar(45), |
|  | PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE flights ( |
|  | id int NOT NULL, |
|  | departing\_gate varchar(20) NOT NULL, |
|  | arriving\_gate varchar(20) NOT NULL, |
|  | departing\_airport\_id int NOT NULL, |
|  | arriving\_airport\_id int NOT NULL, |
|  | airline\_id int NOT NULL, |
|  | PRIMARY KEY (id), |
|  | FOREIGN KEY (airline\_id) REFERENCES airline (id), |
|  | FOREIGN KEY (departing\_airport\_id) REFERENCES airport (id), |
|  | FOREIGN KEY (arriving\_airport\_id) REFERENCES airport (id) |
|  | ); |
|  |  |
|  | CREATE TABLE booking\_has\_flights ( |
|  | booking\_id int NOT NULL, |
|  | flights\_id int NOT NULL, |
|  | FOREIGN KEY (booking\_id) REFERENCES booking (id), |
|  | FOREIGN KEY (flights\_id) REFERENCES flights (id) |
|  | ); |

PostgreSQL DML script

|  |
| --- |
|  |
|  | INSERT INTO airline (id, code, name, country) VALUES (1,'OK','Czech Airlines','Czech Republic'),(2,'DL','Delta Air Lines','United States'),(3,'EW','Eurowings','Germany'),(4,'MU','China Eastern Airlines','China'),(5,'BA','British Airways','United Kingdom'); |
|  | INSERT INTO airport (id, name, country, state, city) VALUES (1,'Guangzhou Baiyun International Airport','China','Guangdong','Guangzhou'),(2,'Hartsfield–Jackson Atlanta International Airport','United States','Georgia','Atlanta'),(3,'Chengdu Shuangliu International Airport','China','Sichuan','Chengdu'),(4,'DallasFort Worth International Airport','United States','Texas','Dallas'),(5,'Shenzhen Bao an International Airport','China','Guangdong','Shenzhen'),(6,'Chongqing Jiangbei International Airport','China','Chongqing','Yubei'),(7,'Beijing Capital International Airport','China','Beijing','Beijing'),(8,'Denver International Airport','United States','Colorado','Denver'),(9,'Kunming Changshui International Airport','China','Yunnan','Kunming'),(10,'Shanghai Hongqiao International Airport','China','Shanghai','Shanghai'),(11,'Xian Xianyang International Airport','China','Shaanxi','Xianyang'),(12,'Tokyo Haneda Airport','China','Tokyo','Tokyo'),(13,'OHare International Airport','United States','Illinois','Chicago'),(14,'Shanghai Pudong International Airport','China','Shanghai','Shanghai'),(15,'Los Angeles International Airport','United States','California','Los Angeles'),(16,'Indira Gandhi International Airport','India','Delhi','Delhi'),(17,'Hangzhou Xiaoshan International Airport','China','Zhejiang','Hangzhou'),(18,'Charlotte Douglas International Airport','United States','North Carolina','Charlotte'),(19,'Dubai International Airport','United Arab Emirates','Dubai','Dubai'),(20,'Istanbul Airport','Turkey','Istanbul','Istanbul'),(21,'Charles de Gaulle Airport','France','Île-de-France','Paris'),(22,'Heathrow Airport','United Kingdom','London','London'),(23,'Mexico City International Airport','Mexico','Mexico City','Mexico City'),(24,'Phoenix Sky Harbor International Airport','United States','Arizona','Phoenix'),(25,'Tan Son Nhat International Airport','Vietnam','Ho Chi Minh City','Ho Chi Minh City'),(26,'Orlando International Airport','United States','Florida','Orlando'),(27,'Zhengzhou Xinzheng International Airport','China','Henan','Zhengzhou'),(28,'Jeju International Airport','South Korea','Jeju Province','Jeju City'),(29,'Amsterdam Airport Schiphol','Netherlands','North Holland','Haarlemmermeer'),(30,'São Paulo/Guarulhos International Airport','Brazil','Guarulhos','Guarulhos'),(31,'Seattle–Tacoma International Airport','United States','Washington','SeaTac'),(32,'Nanjing Lukou International Airport','China','Jiangsu','Nanjing'),(33,'Sheremetyevo International Airport','Russia','Moscow Oblast','Khimki'),(34,'Changsha Huanghua International Airport','China','Hunan','Changsha'),(35,'Frankfurt Airport','Germany','Hesse','Frankfurt'),(36,'Miami International Airport','United States','Florida','Miami'),(37,'George Bush Intercontinental Airport','United States','Texas','Houston'),(38,'Gimpo International Airport','South Korea','Seoul','Seoul'),(39,'Madrid Barajas Airport','Spain','Madrid','Madrid'),(40,'Istanbul Sabiha Gökçen International Airport','Turkey','Istanbul','Istanbul'),(41,'Xiamen Gaoqi International Airport','China','Fujian','Xiamen'),(42,'Suvarnabhumi Airport','Thailand','Samut Prakan','Bang Phli'),(43,'John F. Kennedy International Airport','United States','New York','New York'),(44,'Guiyang Longdongbao International Airport','China','Guizhou','Guiyang'),(45,'Haikou Meilan International Airport','China','Hainan','Haikou'),(46,'Fort Lauderdale–Hollywood International Airport','United States','Florida','Broward County'),(47,'San Francisco International Airport','United States','California','San Mateo Country'),(48,'Moscow Domodedovo Airport','Russia','Moscow Oblast','Domodedovo'),(49,'Beijing Daxing International Airport','China','Beijing','Beijing'),(50,'Newark Liberty International Airport','United States','New Jersey','Newark'); |
|  |  |
|  | INSERT INTO passenger (id, first\_name, last\_name, date\_of\_birth, country\_of\_citizenship, country\_of\_residence, passport\_number) VALUES (1,'Pavel','Ostrý','2001-03-21','Czech Republic','Czech Republic','16851250'),(2,'Karolína','Kopřivová','1984-10-06','Czech Republic','Czech Republic','95723548'),(3,'Věra','Štarhová','1989-04-25','Czech Republic','Czech Republic','68745135'),(4,'Marie','Hrbáčová','1939-04-01','Czech Republic','Czech Republic','54862418'),(5,'Tereza','Kadlečková','1953-02-06','Czech Republic','Czech Republic','85245600'),(6,'Michael','Dohnal','1990-02-22','Czech Republic','Czech Republic','21557896'),(7,'Aleš','Mach','1955-08-19','Czech Republic','Czech Republic','78965412'),(8,'Pavel','Mach','1982-03-08','Czech Republic','Czech Republic','35412856'),(9,'Leona','Ottová','1996-11-22','Czech Republic','Czech Republic','32165485'),(10,'Elena','Juráková','1985-12-06','Czech Republic','Czech Republic','84563572'),(11,'Rostislav','Bartík','1979-07-15','Czech Republic','Czech Republic','55588866'),(12,'Helena','Štorkánová','2009-12-04','Czech Republic','Czech Republic','65488749'),(13,'Marie','Petrželová','1973-10-25','Czech Republic','Czech Republic','35445123'),(14,'Miroslav','Bošek','1978-09-13','Czech Republic','Germany','65485521'),(15,'Miroslav','Černý','1996-07-07','Czech Republic','Canada','33215789'),(16,'Jiřina','Hermanová','2007-06-29','Czech Republic','Brazil','54877862'),(17,'David','Novák','1997-04-20','Czech Republic','France','24158796'),(18,'František','Marek','1969-02-21','Czech Republic','Austria','95632157'),(19,'Monika','Pechanová','1997-12-14','Czech Republic','Estonia','22115588'),(20,'Michala','Jeriová','1990-07-05','Czech Republic','Tunisia','35125870'),(21,'Louis','Eusebio','1971-03-03','United States','United States','57803651'),(22,'Toya','Benally','1965-05-27','United States','United States','45122500'),(23,'Katie','Raymond','1989-04-12','United States','United States','32521356'),(24,'Christiana','Summerfield','1995-06-11','United States','United States','65488004'),(25,'Marry','Puccio','2003-12-21','United States','United States','54009573'),(26,'Larry','Crowley','2005-06-09','United States','United States','65248900'),(27,'James','Jones','1987-11-16','United States','Poland','20062048'),(28,'Ed','Jason','1963-07-03','United States','Tunisia','65002587'),(29,'Eugine','Conrad','1988-09-11','United States','Sweden','97060842'),(30,'John','Houston','1998-11-04','United States','Switzeland','03654028'),(31,'George','Savage','1983-11-05','United States','United Kingdom','60054826'),(32,'Elliot','Reyes','1982-08-22','United States','Belgium','89632175'),(33,'Neil','Harris','1958-10-28','United States','Brazil','00585200'),(34,'Edward','Bray','1974-01-23','United States','Austria','65090275'),(35,'Manuela','Nagel','1957-10-03','Germany','Germany','15589635'),(36,'Thomas','Fenstermacher','1996-05-05','Germany','Germany','32896540'),(37,'Uta','Farber','2006-07-13','Germany','Germany','59674005'),(38,'Sussane','Eggers','1954-02-22','Germany','Germany','68000528'),(39,'Marco','Friedman','1952-12-23','Germany','Germany','20080540'),(40,'Marcel','Gerste','1992-09-26','Germany','Germany','97080506'),(41,'Monika','Herzog','1969-03-21','Germany','Germany','56984000'),(42,'Swen','Seiler','1980-04-18','Germany','Czech Republic','00580548'),(43,'Philipp','Herman','1949-06-16','Germany','Poland','50487365'),(44,'Vanessa','Kaufmann','1963-08-02','Germany','Estonia','60154780'),(45,'Kristian','Voglet','1967-03-27','Germany','Finland','02459024'),(46,'Sabrina','Fuerst','1994-02-23','Germany','France','06900582'),(47,'Lena','Hertzog','1998-05-19','Germany','Spain','99085460'),(48,'Florian','Neudorf','1991-09-20','Germany','Sweden','30069087'),(49,'Leon','Gerste','1957-11-01','Germany','Portugal','30685420'),(50,'Ralph','Kastner','1966-02-12','Germany','Italy','56804569'); |
|  |  |
|  | INSERT INTO boarding\_pass (id, code) VALUES (1,'lapsingw5vhj59i8h2xd'),(2,'6fea8nhpz0q3a5c82d5e'),(3,'qb5c6az8b5q3y6s5d8bb'),(4,'33agrtmhfcqb5c6az8b5'),(5,'6adcvbttrfasf58ecvbg'),(6,'saf5e8asc2ert9vfastg'),(7,'dalgfivclst58x6v2f4a'); |
|  |  |
|  | INSERT INTO booking (id, status, booking\_platform, created\_at, updated\_at, passenger\_id, boarding\_pass\_id) VALUES (1,'confirmed','booking.com','2021-02-07 23:00:00','2021-02-07 23:00:00',7,1),(2,'confirmed','booking.com','2021-06-23 19:59:30','2021-06-24 06:22:30',8,2),(3,'pending','booking.com','2021-11-01 21:35:27','2021-11-01 21:35:27',39,3),(4,'confirmed','booking.com','2021-08-02 23:22:04','2021-08-02 23:22:04',16,4),(5,'cancelled','booking.com','2021-08-09 10:21:30','2021-09-01 15:12:30',3,5),(6,'confirmed','booking.com','2021-12-01 22:25:27','2021-12-01 22:25:27',9,6),(7,'confirmed','booking.com','2021-09-03 09:22:04','2021-09-03 09:22:04',10,7); |
|  |  |
|  | INSERT INTO baggage (id, weight\_in\_kg, booking\_id) VALUES (1,3.25,1),(2,2.13,1),(3,5.58,2),(4,3.07,2),(5,7.17,4); |
|  |  |
|  | INSERT INTO baggage\_check (id, check\_result, comments, baggage\_id) VALUES (1,'OK',NULL,1),(2,'OK',NULL,2),(3,'OK',NULL,3),(4,'PENDING','found fruit and vegetable',4),(5,'FAILED','not proper type of baggage',5); |
|  |  |
|  | INSERT INTO flights (id, departing\_gate, arriving\_gate, departing\_airport\_id, arriving\_airport\_id, airline\_id) VALUES (1,'J44','A13',1,9,2),(2,'C23','B19',6,7,3),(3,'B1','A8',34,20,4),(4,'C5','A1',1,2,5),(5,'A7','A2',14,34,4); |
|  |  |
|  | INSERT INTO booking\_has\_flights (booking\_id, flights\_id) VALUES (1,1),(2,2),(4,3),(6,4),(7,5); |
|  |  |
|  | INSERT INTO covid\_certificate (id, code, valid, passenger\_id) VALUES (1,'URN:UVCI:01:CZ:EHSU2S4F8YYYQVKMW5DF7BDYGEACB9D2','YES',1),(2,'URN:UVCI:01:CZ:EHSU2SAFKLMSDNFNA5DF7BDYGEASASCVE','YES',6),(3,'URN:UVCI:03:CZ:ASDJKLNKMNASMNDKW5DF7BDYGEASPXIQ2','YES',7),(4,'URN:UVCI:11:CZ:EHSWASDMNWAMNIXKWJA7BDYGEASA3Q5','NO',13),(5,'URN:UVCI:05:CZ:EHSU2S4F8YYYQVKMW5DASDAWKJJSKWJS','YES',8); |
|  |  |
|  | INSERT INTO no\_fly\_list (id, active\_from, active\_to, no\_fly\_reason, passenger\_id) VALUES (1,'2020-03-15','2120-03-15','suspected member of terorist organization',43),(2,'2017-07-19','2117-07-19','member of terorist organization',28),(3,'2003-01-01','2006-01-01','insulting other passangers during flight',38),(4,'2005-11-12','2010-11-12','damaging airline seats during flight',40),(5,'2011-12-11','2051-12-11','dangerous criminal',46); |
|  |  |
|  | INSERT INTO security\_check (id, check\_results, comments, created\_at, updated\_at, passenger\_id) VALUES (1,'OK',NULL,'2021-10-31 23:00:00','2021-10-31 23:00:00',7),(2,'OK',NULL,'2021-10-28 22:00:00','2021-10-28 22:00:00',8),(3,'OK',NULL,'2021-10-19 22:00:00','2021-10-19 22:00:00',16),(4,'OK',NULL,'2021-10-06 22:00:00','2021-10-06 22:00:00',9),(5,'OK',NULL,'2021-10-24 22:00:00','2021-10-24 22:00:00',10); |
|  |  |
|  |  |
|  |  |

Obrázok, na ktorom je text

Automaticky generovaný popisProof that all the tables have been created in PostgreSQL – screenshot from pgAdm

Obrázok, na ktorom je stôl

Automaticky generovaný popis