Database Report – Airport Management System

The database that I chose to design for this project was one of an airport management system that manages employees of the airport and flights in operation. The tables in this Database incorporate the airports themselves, the employees of the subsequent airports, the flights incorporated by the airports, the airlines that operate the flights, the airline employees, the tickets sold by the airlines to the passengers, and the passengers of the flight.

The airport table contains the primary key airport id, and others airport name, location of the airport which marks down to the city id and county id.

The airport employee table is for the employees directly hired by the airport, which consists of the employee's PPSN (primary key) that identifies them, the id of the airport (foreign key, in reference to airport id in the airport table) which they work in, their name, age, employee type (which is the sector they belong to) and their position (specific job title).

The flight table is comprised of the primary key flight id, and others including the destination, departure time, arrival time, status (whether it's on time, delayed, or cancelled), number of stops, flight type (whether it's a direct, zero stops, or connecting flight, one or more stops), and foreign keys origin, which references airport id in the airport table, and airline id which references airline id in the table airline.

The airline table which includes the primary key airline id that identifies the airline, and the name of the airline.

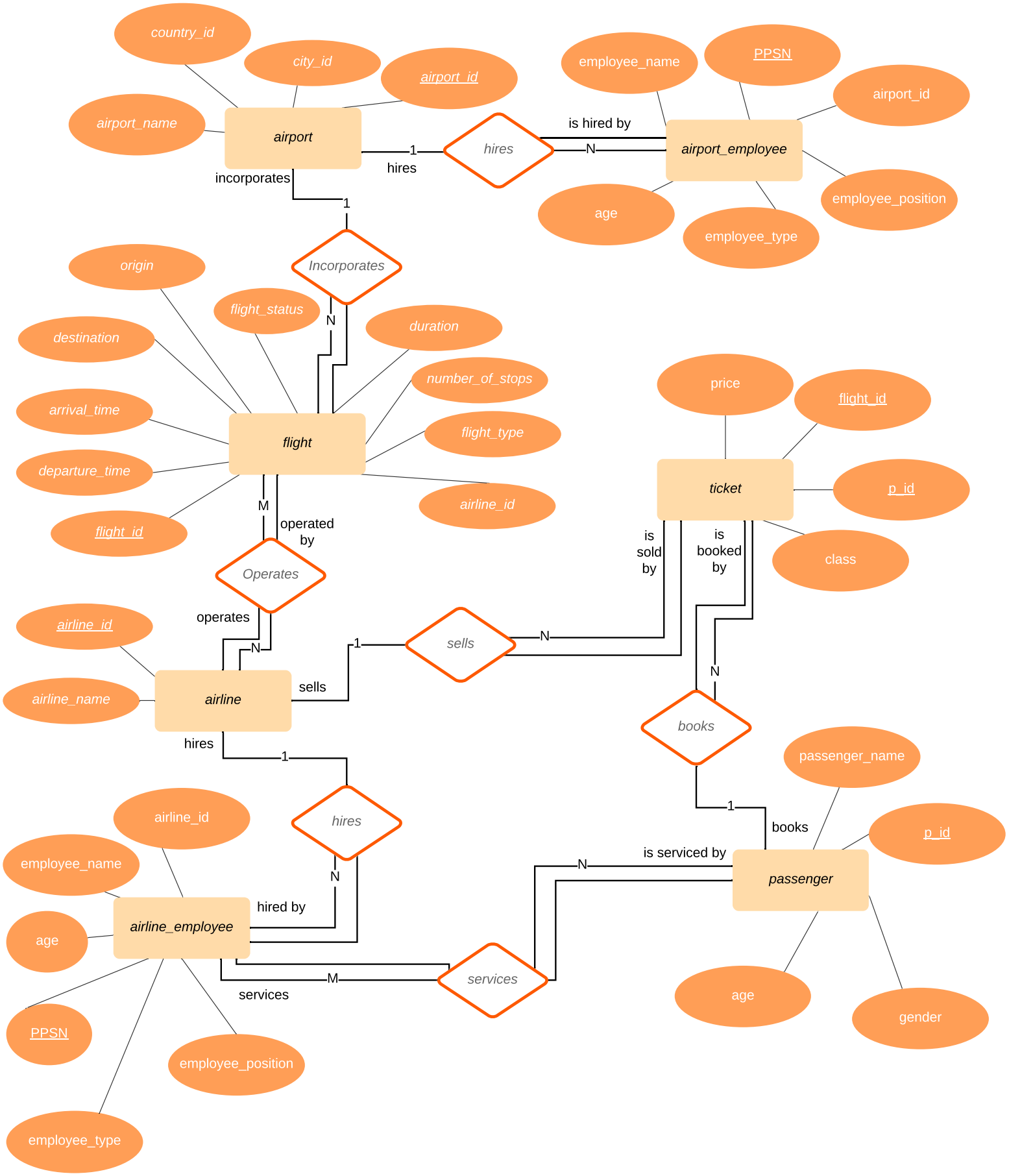
The airline employee table is for the employees of a specific airline, which contains the employee's PPSN (primary key) that identifies them, the id of the airline that recruited them, their name, age, job type, job position, and foreign key airline id of the airline table which identifies the airline they are involved in.

The ticket table which consists of the flight id and p_id (passenger id) which are combined as the primary key of the table that reference the flight id in the flight table and p_id in the passenger table consecutively. The table also contains the price and class of the ticket.

The passenger table which consists of the primary key p_id (passenger id, or passport identification number of the passenger that uniquely identifies them), the name, age, and gender of the passenger also.

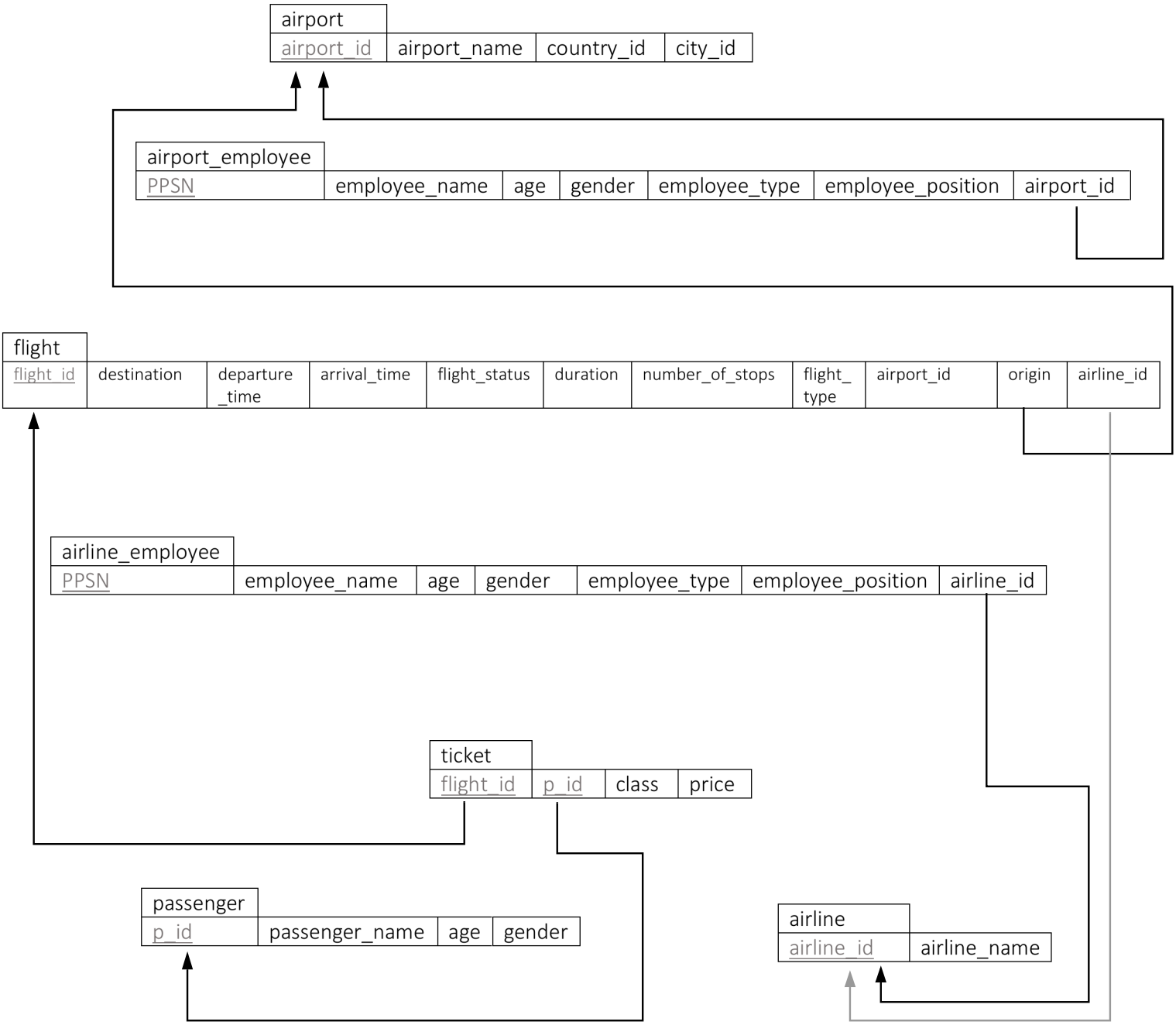
Basic Airport Management Relational Diagram

Caroline Liu | November 20, 2019

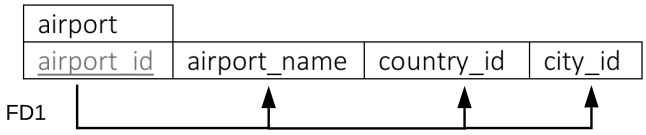


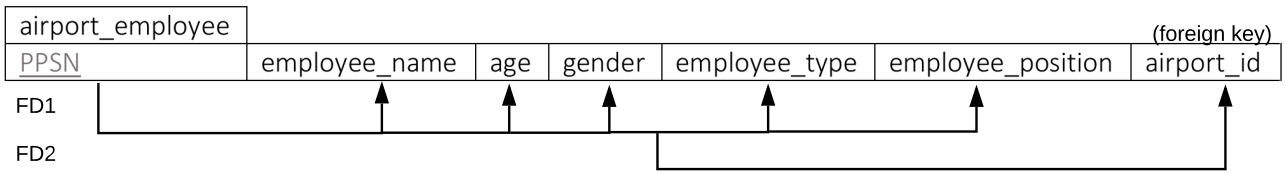
Mapping to Relational Schema

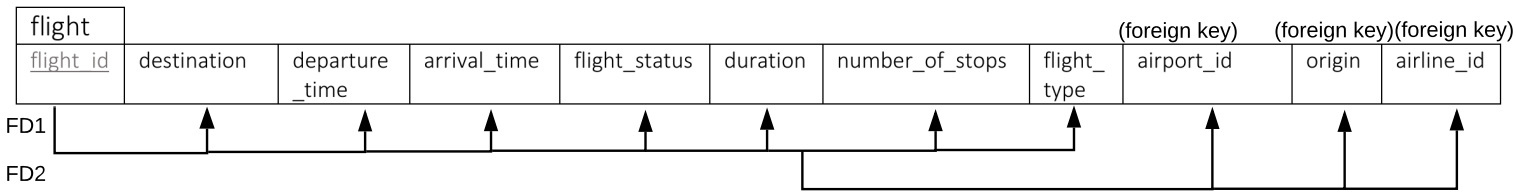
The mapping from the entity relationship diagram to an outline relational schema is shown below:

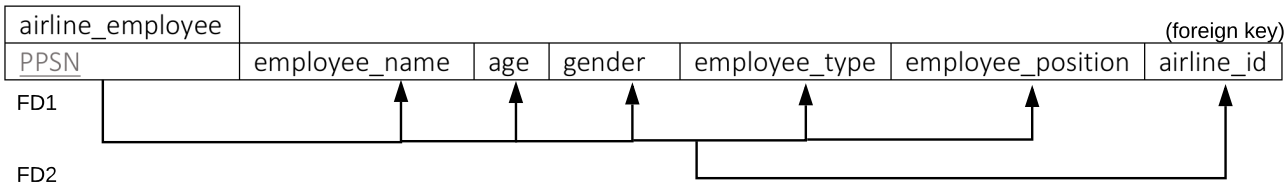


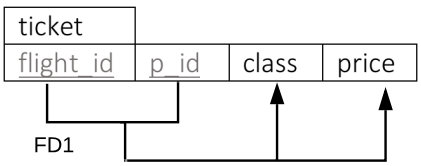
Functional Dependency

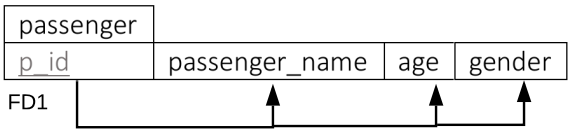


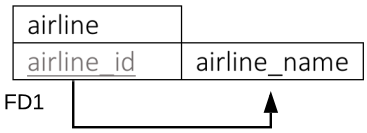












Normalisation

The subsequent tables were generally normalised.

Semantic Constraints

On the defining of some specific tables, CHECKS were used to maintain data integrity of the database. This is seen in the following tables:

- In the flight table, the status of the flight checks that only three states are possible: on time, delayed, or cancelled:
  CONSTRAINT check_status
  CHECK(flight_status IN ('On Time', 'Delayed', 'Cancelled'))
- In the flight table also, the type of the flight is restricted to either direct or connecting, and is checked upon declaration:
  CONSTRAINT check_type
  CHECK(flight_type IN('Direct', 'Connecting'))
- The PPSN number of the airport employee must be 9 digits long and it is enforced by two constraints, that it should be of max length 9 and less than 100000000:
  PPSN INT(9) NOT NULL,
  CONSTRAINT check_ppsn CHECK (PPSN < 1000000000)
- Similarly, in the airline employee table, the PPSN of the employee must not be more than 9 digits, this is enforced also by two restrictions, that its max length is 9 in defining the PPSN and that its value should not be greater than 1000000000:
  PPSN INT(9) NOT NULL,
  CONSTRAINT check_ppsn CHECK (PPSN < 1000000000)
- In the flight table, origin and airline_id are set foreign key constraints referencing the key in the foreign table on DELETE CASCADE:
  origin VARCHAR(10) NOT NULL REFERENCES airport(airport_id) ON DELETE CASCADE,
  airline_id VARCHAR(3) NOT NULL REFERENCES airline(airline_id) ON DELETE C ASCADE,

Every table in the database has its primary key set to NOT NULL, this is so that the subsequent primary keys of each table can be uniquely identified and thus avoiding integrity constraint violations, hence introducing an entity integrity constraint. Foreign keys are defined as NOT NULL as well, referencing the attribute in the table in which they are the primary key on the DELETE CASCADE, specifying a referential integrity.

Other constraints included are NOT NULL for most entities and primary and foreign key constraints similar to that of point 5 above.

Triggers

The following triggers are included in the database:

Upgrade_class updates the class of the passenger before insertion to the database.

```
CREATE TRIGGER upgrade_class
BEFORE INSERT ON ticket
FOR EACH ROW
IF new.p_id = 'IRL12345678'
  THEN SET new.class = 'First Class';
END IF;
```

Flight_change trigger for changing the flight number.

```
CREATE TRIGGER flight_change
BEFORE INSERT ON ticket
FOR EACH ROW
UPDATE passenger
SET passenger.flight_id = 'RYR321'
WHERE passenger.flight_id = 'RYR123';
```

Updates

```
-- Find an airport in a specific country e.g. Ireland
SELECT country_id, city_id FROM airport WHERE country_id = 'IRL';

-- Find an airport not in a specific country e.g. Germany
SELECT country_id, city_id FROM airport WHERE country_id <> 'GER';

-- Find passengers on a certain flight
SELECT passenger_name, class, price FROM (ticket INNER JOIN passenger USING(p_id)
 INNER JOIN flight USING(flight_id)) WHERE flight_id = 'RYR123';
```

Security

Security within a database is vital, there are two main ways to handle database security, of which include security policies and access control. Integrity handles the unintended corruption of data, whilst security is maintaining the overall database not to abort.

For this database I have given users roles and they are granted certain privileges in their given row. In the example given, where the normal employee role is able to view only targeted information about the employees in the given airport only, other sensitive information is hidden from the employee. It is thought that a normal employee should not have access to sensitive data of other co-workers. In this scenario only the manager is allowed to update, read and write the view airport_staff.

```
CREATE VIEW airport_staff AS

SELECT employee_name, employee_type, employee_position

FROM airport_employee

WHERE airport_id = 'CDG';
```

The manager is identified by the password cdadminman, while other staff are granted the role of cdg_employee identified by the password cdgstaff.

```
CREATE ROLE cdg_manager IDENTIFIED BY cdgadminman;

CREATE ROLE cdg_employee IDENTIFIED BY cdgstaff;
```

The manager and the normal staff are all granted certain privileges on the view created by using the GRANT WITH GRANT OPTION:

```
GRANT  WRITE, UPDATE, READ ON airport_employee TO to cdg_manager WITH GRANT OPTION;

GRANT SELECT ON airport_staff TO cdg_employee WITH GRANT OPTION;
```

If a certain manager is thought to have been abusing his privileges, they can be revoked.

```
REVOKE CREATE TABLE FROM cdg_manager;
```

The security policy above further enhances the security of the system, as only certain administrative users of the system are granted privileges to alter the data. Also noting the fact that for both the normal and administrative users, the sensitive data is initially hidden and that they can only alter certain attributes of the created views.

Appendix

```
-- Creating tables
CREATE TABLE airport(airport_name VARCHAR(50) NOT NULL,
                airport_id  VARCHAR(3) NOT NULL,
                country_id   VARCHAR(10) NOT NULL,
                city_id     VARCHAR(10) NOT NULL,
                CONSTRAINT pk_airport
                PRIMARY KEY (airport_id));

CREATE TABLE airline(airline_id VARCHAR(3) NOT NULL,
                airline_name VARCHAR(50) NOT NULL,
                CONSTRAINT pk_airline
                PRIMARY KEY (airline_id));

CREATE TABLE flight(flight_id VARCHAR(10) NOT NULL,
                origin VARCHAR(10) REFERENCES airport(airport_id) ON DELETE CASCADE,
                destination VARCHAR(10) NOT NULL,
                departure_time VARCHAR(10) NOT NULL,
                arrival_time VARCHAR(10) NOT NULL,
                flight_status   VARCHAR(10) NOT NULL,
                duration VARCHAR(10) NOT NULL,
                number_of_stops INT,
                flight_type VARCHAR(10) NOT NULL,
                airline_id VARCHAR(3) REFERENCES airline(airline_id) ON DELETE CASCADE
,
                CONSTRAINT pk_flight
                PRIMARY KEY (flight_id),
                CONSTRAINT check_status
```

```
        CHECK(flight_status IN ('On Time', 'Delayed', 'Cancelled')),
        CONSTRAINT check_type
        CHECK(flight_type IN('Direct', 'Connecting')));

CREATE TABLE airport_employee(PPSN INT(9) NOT NULL,
            employee_name VARCHAR(50) NOT NULL,
            age INT(2) NOT NULL,
            gender VARCHAR(10) NOT NULL,
            employee_type VARCHAR(20) NOT NULL,
            employee_position VARCHAR(30) NOT NULL,
            airport_id VARCHAR(3) REFERENCES airport(airport_id) ON DELETE CA
SCADE,
            CONSTRAINT pk_employee
            PRIMARY KEY (PPSN),
            CONSTRAINT check_ppsn
            CHECK (PPSN < 1000000000));

CREATE TABLE airline_employee(PPSN INT(9) NOT NULL,
            employee_name VARCHAR(50) NOT NULL,
            age INT(2) NOT NULL,
            gender VARCHAR(10) NOT NULL,
            employee_type VARCHAR(20) NOT NULL,
            employee_position VARCHAR(20) NOT NULL,
            airline_id VARCHAR(3) REFERENCES airline(airline_id) ON DELETE CAS
CADE,
            CONSTRAINT pk_employee
            PRIMARY KEY (PPSN),
            CONSTRAINT check_ppsn
            CHECK (PPSN < 1000000000));
```

```
CREATE TABLE passenger(p_id VARCHAR(11) NOT NULL,
                    passenger_name VARCHAR(50) NOT NULL,
                    age INT(2) NOT NULL,
                    gender VARCHAR(10) NOT NULL,
                    CONSTRAINT pk_passenger
                    PRIMARY KEY (p_id));


CREATE TABLE ticket(flight_id VARCHAR(10) REFERENCES flight(flight_id) ON DELETE C
ASCADE,
                price INT NOT NULL,
                class VARCHAR(20) NOT NULL,
                p_id VARCHAR(11) REFERENCES passenger(p_id) ON DELETE CASCADE,
                CONSTRAINT pk_ticket
                PRIMARY KEY (flight_id, p_id));


-- View to display employees of a specific airport e.g. CDG
CREATE VIEW airport_staff AS
SELECT employee_name, employee_type, employee_position
FROM airport_employee
WHERE airport_id = 'CDG';


CREATE ROLE cdg_manager IDENTIFIED BY cdgadminman;
CREATE ROLE cdg_employee IDENTIFIED BY cdgstaff;
GRANT  WRITE, UPDATE, READ ON airport_employee TO to cdg_manager WITH GRAN
T OPTION;
GRANT SELECT ON airport_staff TO cdg_employee WITH GRANT OPTION;
-- If manager is abusing his privileges, the below is to revoke them
REVOKE CREATE TABLE FROM cdg_manager;
```

```
-- Find an airport in a specific country e.g. Ireland

SELECT country_id, city_id FROM airport WHERE country_id = 'IRL';


-- Find an airport not in a specific country e.g. Germany

SELECT country_id, city_id FROM airport WHERE country_id <> 'GER';


-- Find passengers on a certain flight

SELECT passenger_name, class, price FROM (ticket INNER JOIN passenger USING(p_id)
 INNER JOIN flight USING(flight_id)) WHERE flight_id = 'RYR123';


-- Trigger to upgrade class of certain passenger

CREATE TRIGGER upgrade_class

AFTER INSERT ON ticket

FOR EACH ROW

UPDATE ticket

SET NEW.class = 'First Class'

WHERE NEW.p_id = 'IRL12345678';


-- Trigger for changing the flight number

CREATE TRIGGER flight_change

BEFORE INSERT ON ticket

FOR EACH ROW

UPDATE passenger

SET passenger.flight_id = 'RYR321'

WHERE passenger.flight_id = 'RYR123';


-- Inserting values of airport

INSERT INTO airport VALUES('Dublin Airport', 'DUB', 'IRL', 'IED001');

INSERT INTO airport VALUES('Amsterdam Airport Schiphol', 'AMS', 'NED', 'NLA001');

INSERT INTO airport VALUES('Birmingham Airport', 'BHX', 'ENG', 'UKB001');
```

```
INSERT INTO airport VALUES('Heathrow Airport', 'LHR', 'ENG', 'UKL001');

INSERT INTO airport VALUES('Dubai International Airport', 'DXB', 'UAE', 'UAED001');

INSERT INTO airport VALUES('Frankfurt Airport', 'FRA', 'GER', 'DEF001');

INSERT INTO airport VALUES('Charles de Gaulle Airport', 'CDG', 'FRA', 'FRAP001');

INSERT INTO airport VALUES('John F. Kennedy International Airport', 'JFK', 'USA', 'USA N001');


-- Inserting values into airline

INSERT INTO airline VALUES('RYR', 'Ryanair');

INSERT INTO airline VALUES('VRN', 'Virgin America');

INSERT INTO airline VALUES('AFR', 'Air France');

INSERT INTO airline VALUES('KLM', 'KLM');

INSERT INTO airline VALUES('QTR', 'Qatar');

INSERT INTO airline VALUES('BAW', 'British Airways');

INSERT INTO airline VALUES('DLH', 'Lufthansa');

INSERT INTO airline VALUES('ETD', 'Etihad Airways');

INSERT INTO airline VALUES('UAE', 'Emirates Airlines');

INSERT INTO airline VALUES('EIN', 'Aer Lingus');


-- Inserting values into flight

INSERT INTO flight VALUES('RYR123', 'DUB', 'BHX', '12:34', '13:20', 'On Time', '00:46', 0 , 'Direct', 'RYR');

INSERT INTO flight VALUES('VRN877', 'LHR', 'JFK', '08:00', '11:15', 'On Time', '08:10', 0,  'Direct', 'VRN');

INSERT INTO flight VALUES('AFR098', 'CDG', 'AMS', '20:50', '22:05', 'On Time', '01:20', 0, 'Direct', 'AFR');

INSERT INTO flight VALUES('UAE161', 'DXB', 'DUB', '14:35', '18:55', 'Delayed', '08:15', 0 , 'Direct', 'KLM');

INSERT INTO flight VALUES('DLH776', 'LHR', 'FRA', '06:30', '09:05', 'On Time', '01:35', 1,  'Connecting', 'DLH');

INSERT INTO flight VALUES('ETD646', 'CDG', 'DXB', '13:30', '23:25', 'Delayed', '06:45', 0 , 'Direct', 'ETD');
```

```
INSERT INTO flight VALUES('RYR321', 'DUB', 'BHX', '18:34', '19:20', 'On Time', '00:46', 0
, 'Direct', 'RYR');


-- Inserting values into airport employee

INSERT INTO airport_employee VALUES(198700703, 'Daniel Marks', 45, 'Male', 'Admin
istrative', 'PR specialist', 'DUB');

INSERT INTO airport_employee VALUES(231897198, 'Frank Brown', 34, 'Male', 'Sales',
'Sales Managar', 'DXB');

INSERT INTO airport_employee VALUES(342627678, 'Anne Francesca', 23, 'Female', 'A
dministrative', 'Receptionist', 'CDG');

INSERT INTO airport_employee VALUES(432489434, 'Regan Mathews', 27, 'Female', 'S
ales', 'Sales Agent', 'LHR');

INSERT INTO airport_employee VALUES(432924777, 'Fortran Porta', 43, 'Male', 'Maint
enance', 'Aircraft Maintenance', 'QTR');

INSERT INTO airport_employee VALUES(343728498, 'Paparica Dew', 24, 'Female', 'Ser
vices', 'Passenger Service Agent', 'BHX');

INSERT INTO airport_employee VALUES(879964343, 'John Federal', 43, 'Male', 'Groun
d Support', 'Aviation Mechanic', 'JFK');

INSERT INTO airport_employee VALUES(123874366, 'Abd al Qadir Matek', 32, 'Male', '
Operations', 'Operations Agent', 'DXB');

INSERT INTO airport_employee VALUES(157783429, 'Lyron Dymek', 57, 'Male', 'Admin
istrative', 'Reservation Sales Agent', 'LHR');

INSERT INTO airport_employee VALUES(342477898, 'Michael Dryer', 43, 'Male', 'Coor
dinate', 'Flight Dispatcher', 'CDG');

INSERT INTO airport_employee VALUES(342627678, 'Dull Day', 23, 'Female', 'Administ
rative', 'Airport Manager', 'CDG');


-- Inserting values into airline employee

INSERT INTO airline_employee VALUES(123674832, 'Emma Harry', 32, 'Female', 'Cabi
n Crew', 'Flight Attendant', 'RYR');

INSERT INTO airline_employee VALUES(467321873, 'Wendy Gardiner', 29, 'Female', 'C
abin Crew', 'Manager', 'VRN');

INSERT INTO airline_employee VALUES(342343243, 'Mario Speedwagon', 45, 'Male', '
Pilot', 'Pilot in command', 'KLM');
```

```
INSERT INTO airline_employee VALUES(634242423, 'Petey Cruier', 54, 'Male', 'Cabin C
rew', 'Flight Attendant', 'QTR');

INSERT INTO airline_employee VALUES(123456654, 'Anna Sthesia', 43, 'Female', 'Cabi
n Crew', 'Flight Attendant', 'BAW');

INSERT INTO airline_employee VALUES(164535545, 'Paul Molive', 42, 'Male', 'Cabin Cr
ew', 'Manager', 'DLH');

INSERT INTO airline_employee VALUES(234564566, 'Anna Mull', 37, 'Female', 'Pilot', '
Second Officer', 'AFR');

INSERT INTO airline_employee VALUES(823434243, 'Gail Forcewind', 34, 'Male', 'Cabi
n Crew', 'Flight Attendant', 'UAE');

INSERT INTO airline_employee VALUES(923432432, 'Paige Turner', 67, 'Female',  'Pilot
', 'First Officer','EIN');

INSERT INTO airline_employee VALUES(342432424, 'Bob Frapples', 29, 'Male', 'Cabin
Crew', 'Flight Attendant', 'EIN');


-- Inserting values into passenger

INSERT INTO passenger VALUES('IRL12345678', 'Taylor Murphy', 24, 'Male');

INSERT INTO passenger VALUES('FRA09876542', 'Cecile Martin', 60, 'Female');

INSERT INTO passenger VALUES('IRL76767777', 'Aoife O Toole', 34, 'Female');

INSERT INTO passenger VALUES('POL34561899', 'Emanuel Oklahoma', 23, 'Male');

INSERT INTO passenger VALUES('RUS09232317', 'Alexandera Bykov', 54, 'Male');

INSERT INTO passenger VALUES('USA43434333', 'Donal Trumpet', 67, 'Male');

INSERT INTO passenger VALUES('QAT09867778', 'Muhamad Palsha', 47, 'Male');

INSERT INTO passenger VALUES('DEC80766333', 'Frederick Sederick', 56, 'Male');

INSERT INTO passenger VALUES('ITA34247899', 'Monet Aneeta', 34, 'Female');

INSERT INTO passenger VALUES('OMA34234236', 'Beri Berida', 36, 'Female');


-- Inserting values into ticket

INSERT INTO ticket VALUES('RYR123', 123, 'Economy Class', 'IRL12345678');

INSERT INTO ticket VALUES('VRN877', 1002, 'First Class', 'FRA09876542');

INSERT INTO ticket VALUES('AFR098', 456, 'First Class', 'IRL76767777');

INSERT INTO ticket VALUES('UAE161', 877, 'Economy Class', 'RUS09232317');
```

```
INSERT INTO ticket VALUES('ETD646', 238, 'Economy Class', 'DEC80766333');
INSERT INTO ticket VALUES('DLH776', 78, 'Economy Class', 'OMA34234236');
INSERT INTO ticket VALUES('UAE161', 987, 'Economy Class', 'USA43434333');
INSERT INTO ticket VALUES('RYR123', 123, 'Economy Class', 'ITA34247899');
INSERT INTO ticket VALUES('VRN877', 567, 'Economy Class', 'POL34561899');
INSERT INTO ticket VALUES('UAE161', 1238, 'First Class', 'FRA09876542');
```