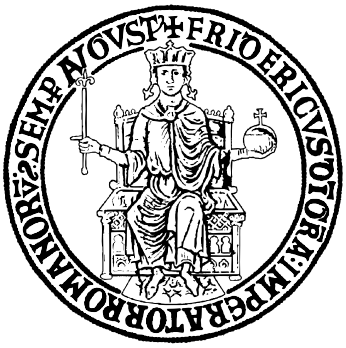
"Federico II" University of Naples



Gestione dei Sistemi Aerospaziali per la Difesa

Database Systems

Flight Management System

Authors:
Gabriele Carinci
Ugo Galleri
Matteo Ingrosso
Claudio Raciti

Supervisor:
Prof. Ing. Antonio
Picariello, P.h.D.

June 2018

Chapter 1

Introduction

This project is meant to represent a data base system capable of tracking and managing a high traffic volume on behalf of the F.M.S. (*Flight Management System*). This System will compute the process behind the flights' schedules in order to make the Airline companies operations easier when operating their assets.

The data base will also be updated in *"near real time"*, by the pilots through their F.M.C. (*Flight Management Computer*) and *Datalink*, in order to send to the Airline the flight status (on time, late or canceled). Furthermore, it will let passengers gain the data they need to travel.

1.1 Data specifications

The specifications that follows resulted from the customer interview and the analysis of the procedures which go on inside the sample reality.

"We want to create a database for a Flight Management System (FMS). A flight can be registered by an authorized airline, which schedules it in advance, indicating:

- The *flight identification number* which, associated with the identification code of the airline, represents the flight code;
- The type of aircraft that will perform that flight;
- The crew, identifying the names and roles (pilot, co-pilot, flight assistants, etc.) of the crew;
- Airports, expected dates and times of take-off and landing, both for the departure and arrival node, as well as for any intermediate layover.

The time must be indicated according to the UTC reference to the Greenwich meridian. The airline can change or cancel the flight until it is taken over by the pilot: in that moment, the pilot becomes the only person who is authorized to modify the flight data or cancel it.

When the pilot takes charge of the aircraft, he must trace the flight assigned to him on the basis of the identification code, and confirm the flight data entered by the company, or he may modify, in case of necessity, the timetable, the aircraft and the composition of the aircraft 's crew. In addition, the IFF identification code required for radar surveys will be associated with the flight data.

The pilot must also indicate, for each intermediate stop and for the airport of destination, the alternative airports to be reached in case of need (for example due to congestion of the primary airport or for particular emergency reasons).

The on-board personnel must respect rest intervals between one flight and another (defined in 8 hours for domestic and continental flights, and 24 hours for intercontinental flights), therefore each crew member can not be assigned to other flights before the expiry of the expected range.

In addition, the *Air Traffic Control unit*, that is the one in charge of a *Flight Information Region*, must be able to carry out a series of checks on each flight, in particular:

- They must trace all the data related to the airline that has scheduled the flight and in particular:
  - the company name;
  - the identification code;
  - the ATM authorization for carrying on the flight;
  - references for administrative and emergency contacts;
  - employees and related qualifications;
- They may be able to verify that, in case the aircraft is actually used for a particular flight that is different from that envisaged, its class is equal to or higher than that of the intended aircraft;
- They must be able to check which classes of aircraft each pilot is enabled to pilot;
- They have to check whether the airports indicated do not present elements of incompatibility with a specific flight, in particular with regard to:

  - The category of aircrafts that are able to take off from and land on those airport's runways;
  - The kind of traffic (international / national) which can be managed.

Finally, after the checks listed above, they must authorize a pilot, who has taken charge of an airplane, to flight. Only after this authorization, a flight will become "active" and can be managed by the flight control that will assist the pilot during the different phases of the flight, until landing at the destination airport. Once the landing operations have been completed, the pilot can release the flight that can be "stored" by the airline in a historical archive."

When the whole system goes at operating speed, in one-year time the database will be saving the quantity of information that follows:

- About 500.000 flights per year;
- Around 45.000 airports (mostly of them already exists, but a little percentage of the number may vary);
- An average of 4 or 5 runways per airport;
- About 1.500 types of aircraft;
- Around 5.000 airlines;
- There are 310 *Flight Information Regions* in the world, so at least the same number of *Flight Information Centers* (the ATC unit meant in the specifications);
- Around 10.000 cities;
- About 3.000.000 people works in airlines.

1.2 Operational specifications

The fundamental operations needed over database are:

- Insert of a new flight (n-times a day);
- Upgrade of flight (n-times);
- Insert of a new airline company (1 time a month);

- Insert of new aircraft (1 time a week);
- Insert of a new airport (1 time a month);
- Insert of new Personnel (1 time a month);
- Upgrade of an airline company (1 time a month);
- Upgrade of available airport (1 time a day);
- Upgrade of Personnel (1 time a month);
- Upgrade of new aircraft (1 time a month);
- Views of flight status for Atc unit (n-times a day);
- Views of flight status for Passenger (n-times a day);
- Views of flight status Airline Company (n-times a day);
- Views of flight status for Airport (n-times a day).

1.3 Technological constraints and security policies

The standard application uses a classical architecture three-tier: at the presentation level there are user interfaces for the application (web or desktop); at the application level, instead, there are object software which realize a closed logical application; at the end, in the data level, there are the DBMS and its managed information. DBMS is installed on a server that is made up of 6GB RAM e 4HDD, which create 1TB of available mass memory.

There are 4 categories of user, as follows, that will manage the database system:

- DBA, it owns all the privileges available on the database schema;
- Pilot, each pilot shall receive all the flight's information from a web application and is also able to update them;
- Airline, each airline company can schedule, follow and manage its own flight from planning to landing with a web application;
- ATC, it can update and monitoring flight status in order to inform passengers and the flight network exploiting a web application.

Chapter 2

Database design

Once all the requirements have been fetched, they have to be analyzed in order to proceed with the conceptual design, then the logical one and, finally, the physical one.

All the entity-relationship graphs were made using *LucidChart* on internet.

2.1 Specifications analysis

From the analysis of the specifications we have gathered all the words needed to describe the database, with a description for each of them. These data are showed in Table 2.1.

Furthermore, we have divided the requirements into categories, in order to have a clearer vision for understanding what is required by the design of the *Flight Management System* application.

General information

We want to create a database for a *Flight Management System* (FMS). Due to its structure, it is intended to be used, firstly, by the manager of the facility and, then, by all the actors of an aviation environment. The time shall be indicated according to the UTC reference to the Greenwich meridian.

Information on flights

A flight can be registered by an authorized airline, which schedules it in advance. We need to know: its flight code (that is made up of the flight ID number plus

the airline's ID); the type of aircraft; its crew, with all the names and roles; arrival and departure airports, its stopovers and all the alternates; the expected dates and times of take off and landing; the IFF code; its activity; its status and the ATM authorization.

One flight shall have only one responsible and all the ended flights are saved into a historical archive.

Information on airports

About the airports we want to save: both the ICAO and IATA code; its name and location; all the runways; the kind of traffic it is allowed to receive and the services it has.

Information on aircrafts

For each aircraft it is required to know: its model and manufacturer; its category and dimensions; its ICAO code; the number of passengers and the MTOW.

Information on airline

Each company is indicated by: its name and ID code; some references for administrative and emergency contacts; the aircrafts owned and the employees hired; its main hubs; its web site and the home country. Then, each airline schedules and can change or cancel its flights.

Information on personnel/crew

All the data related to people involved in flights have to be saved and we mean: their SSN, name and surname; date and place of birth; their qualifications and the airline in which they are employed; their last flight's time, in order to respect rest intervals (8 hrs for domestic flights, 24 hrs for intercontinental ones). The types of crew we can encounter are: pilot flying; pilot monitoring; flight engineer; flight attendant.

The pilot flying takes over the flight and he is the only one authorized to modify the flight. He shall trace his flight and confirm all the data inserted. Furthermore, he communicates the alternate airports.

Information on ATC unit

It is described by the ICAO region of responsibility and it has a main role in authorizing flights, tracing all the data related, verifying that the aircraft planned to use

is right for that flight, checking pilots' qualifications and the correct match among aircraft's category and airports.

(For the project they are meant to be the Flight Information Center).

Information on cities

It is needed a set to carry the information of cities, like their name, country and even the ICAO region within they are located.

Table 2.1: Glossary of the terms.

WORD	***DESCRIPTION***	***LINKS***
Flight	A scheduled transport using an aircraft, that is owned by an airline, between two or more airports.	*Airport, Airline, Aircraft, Crew, Scheduling, Performing, ATC unit, Taking charge of*
Status	The status is used for knowing the actual state of the flight, in particular if it is landed and stored or not.	*Aircraft*
ATM authorization	Furthermore, the ATM authorization is the one that the ATC unit checks before authorizing the current flight.	*Aircraft*
Airport	They are places that provide runways, facilities and services for aircrafts to operate.	*Flight, City, Aircraft, Airline*
Airline	A company that owns aircraft, hires crew members and plans flights as its aim.	*Personnel, Aircraft, City, Airport, Flight*
Personnel	People hired by airlines for working as pilots or crew members. The pilot flying, monitoring and the flight engineer are related to the aircraft they are enabled to fly.	*Airline, Flight, City*
Aircraft	Means of transportation that can fly.	*Airline, Flight, Airport*

Continues in the next page

Continues from the previous page

WORD	***DESCRIPTION***	***LINKS***
City	Urban areas where people lives or were born, where companies stays and where airports are.	*Airport, Personnel, Airline*
Crew	The part of the whole personnel set that is actually flying.	*Personnel, Flight*
Scheduling	The act of planning a flight in advance.	*Flight*
Performing	The act of carry out something such as duties and tasks. It is used for aircraft and pilots that perform a flight.	*Flight, Crew*
Associating	The act of linking two or more things by shared characteristics.	
Pilot flying	The part of the crew that is in charge of the aircraft and the flight.	*Flight, Crew, Personnel, Airline, City*
Pilot monitoring	The part of the crew that is employed as the second pilot.	*Flight, Crew, Personnel, Airline, City*
Flight engineer	The part of the crew that helps the pilots.	*Flight, Crew, Personnel, Airline, City*
Flight attendant	The part of the crew that ensure the safety and comfort of passengers aboard commercial flights. They have to be more than one and one of them is the chief of the cabin crew	*Flight, Crew, Personnel, Airline, City*
Intermediate node, Stopover	An airport where an aircraft temporary stops its flight.	*City, Airport, Flight*
Authorizing	The act of allowing someone to do something, such as the ATC or the airline that authorize a flight.	*ATC unit, Flight*
Taking charge of	The act of becoming responsible for something.	*Pilot flying, Flight*

Continues in the next page

Continues from the previous page

WORD	***DESCRIPTION***	***LINKS***
Managing	Having the capability for controlling and changing something.	*ATC unit, Flight*
ATC unit	A unit that provides Air Traffic Control services. It checks that the flight has all the requirements for taking off.	*Flight, Autorizing, City*

Ends from the previous page

2.2 Conceptual design

Carrier design

This project has 5 main entities. These entities are FLIGHT, PERSONNEL, AIRCRAFT, AIRPORT and AIRLINE respectively representing: the whole flight volume in the Flight Management System, the totality of the aircrafts operated in order to transport passengers and goods, the departure and arrival airport, all the personnel hired and employed by the companies for flight duties and the companies that own the aircrafts.

The carrier design is shown in Figure 2.1.

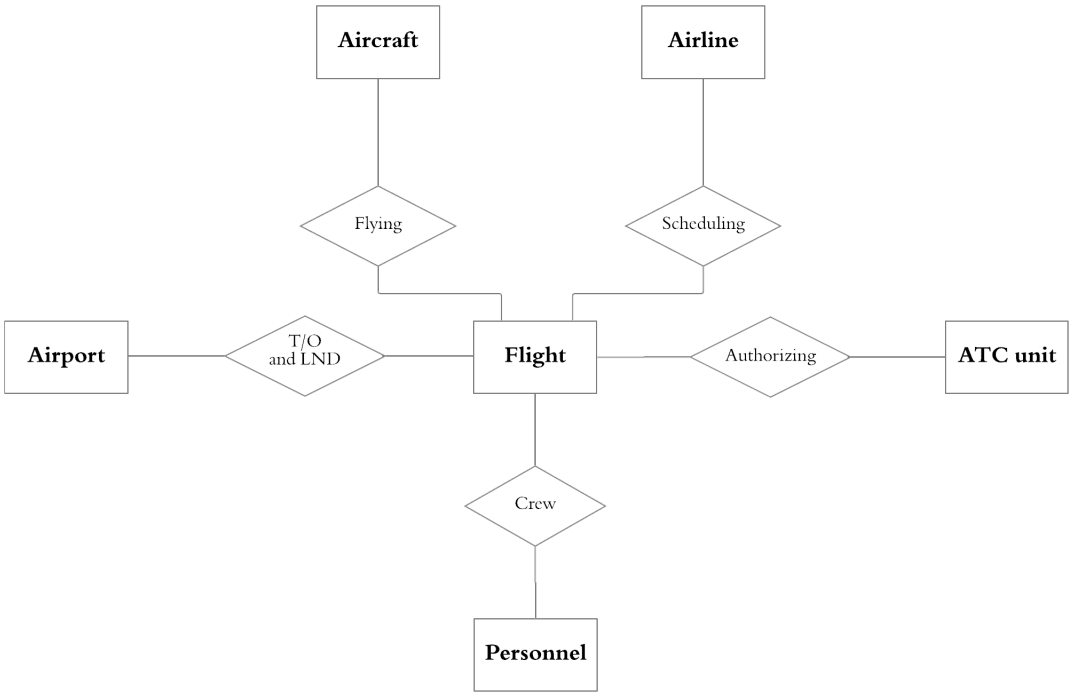


Figure 2.1: Carrier design.

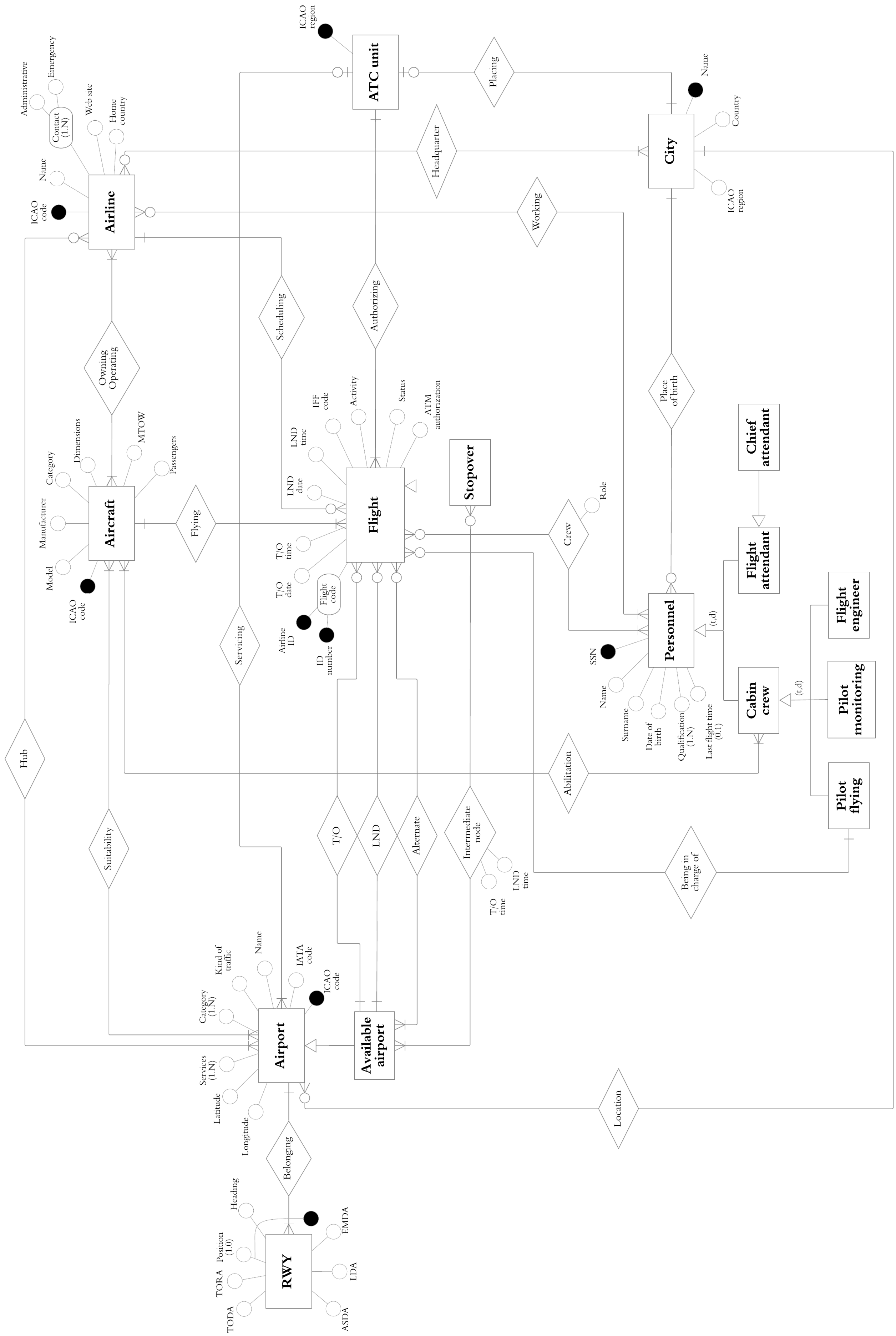


Figure 2.2: EER design.

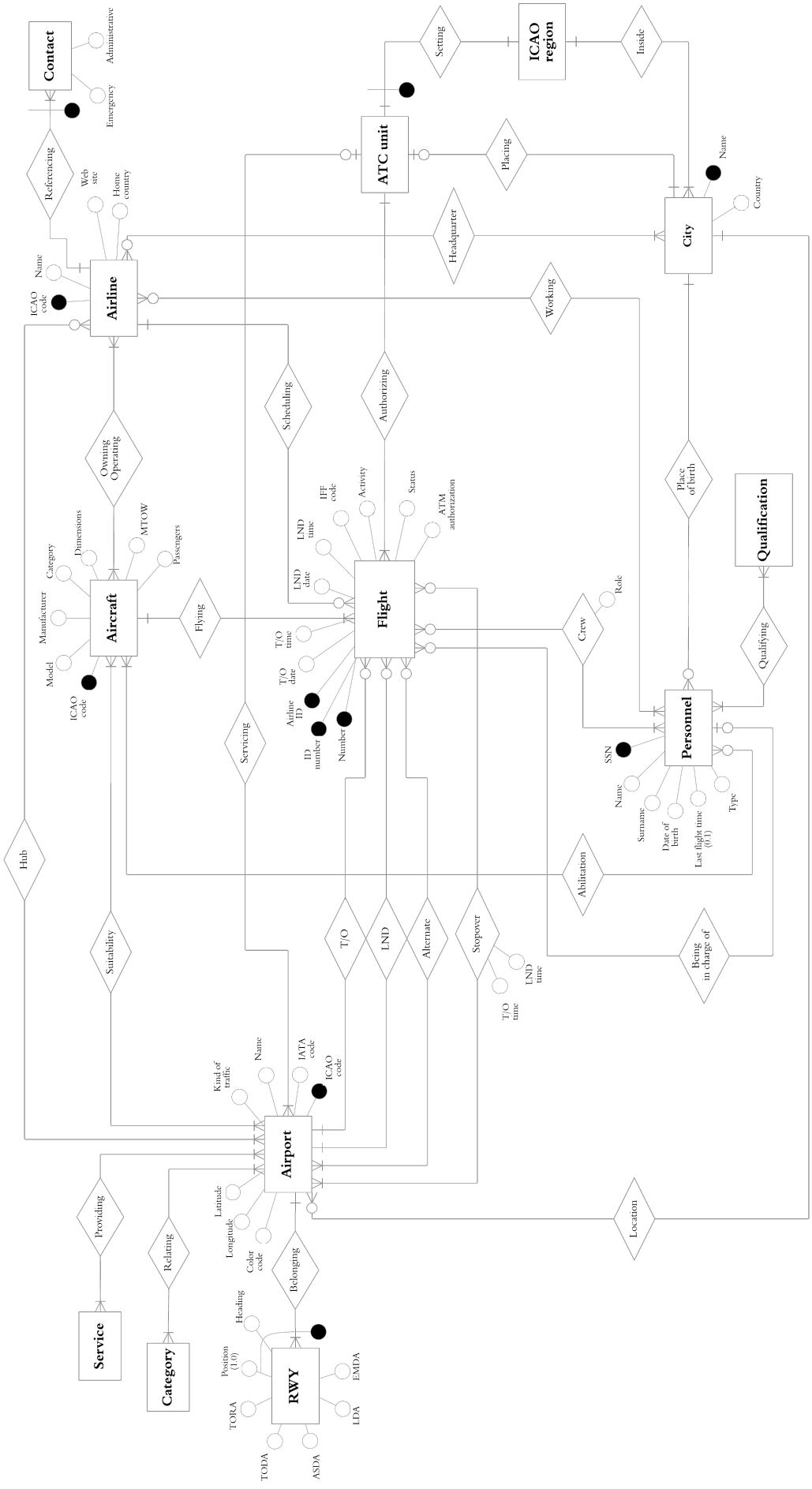


Figure 2.3: EER design transformed.

Complete EER design

The entities called FLIGHT and PERSONNEL are bond by the mean of the CREW association that stands for the crew on duty on a flight. FLIGHT and AIRCRAFT are coupled through the FLYING association that embodies the type of aircraft operated during a flight. FLIGHT and AIRPORT are each other related to using T/O, LND and ALTERNATE associations that respectively represent the departure, arrival and alternate airport scheduled during a flight. AIRCRAFT and AIRPORT are linked by SUITABILITY association expressing the compatibility of the airport structures with a variety of different types of aircraft. AIRLINE and FLIGHT are associated by the means of SCHEDULING. AIRLINE and PERSONNEL are bond by the mean of WORKING expressing the relationship between the companies and the crews operating the aircrafts. AIRLINE and PERSONNEL are linked by OWING/OPERATING association representing the companies ownership over the aircrafts. AIRLINE and AIRPORT are each other related to using HUB association. The EER design is shown in Figure 2.2.

Looking closer is possible to notice that the scheme is not over yet as we have to manage the remaining entities and associations. In particular, the PERSONNEL entity has a large number of specializations such as PILOT, PILOT FLYING, PILOT MONITORING, FLIGHT ENGINEER, FLIGHT ATTENDANT and CHIEF ATTENDANT as generalization. AIRPORT has the entity AVAILABLE AIRPORT and FLIGHT has the entity STOPOVER as well as generalization.

2.3 Logical and relational design

The next step is the logical design. It provides the definition of the Database's relational scheme, with all the sets of relations and constraints. It splits in two phases:

1. The transformation of the conceptual design into a simplified scheme, it is obtained removing both composite and multi-value attributes and solving the generalizations;

2. The translation of the simplified scheme into the relational scheme.

Transformation phase

Referring to the scheme, the transformation only involves the suppression of PERSONNEL, AIRPORT and FLIGHT generalizations.

Applying solution number 2, the subsidiary entities are removed and those derived attributes are added to the parent entity. Our choice has been made due to the type of the generalization, that is total and disjoint. So, this solution can be applied without any other reason to PERSONNEL and AIRPORT generalizations.

Regarding to the subsidiary entity STOPOVER, we have decided to transform it and the association INTERMEDIATE NODE into a single association, named STOPOVER, in order to avoid data redundancy.

The transformed design is shown in Figure 2.3.

Translation phase

Generally, the translation phase requires the following actions:

1. Each entity turns into a relation whose attributes are the entity's attributes and whose primary key is the identifier of the related entity;

2. Many-to-many associations, as CREW, SUITABILITY, RELATING, PROVIDING, HUB, ABILITATION, OWNING, HEADQUARTER, WORKING, QUALIFYING, become relations with the identifiers of the related entities as primary keys and with its own attributes, if it has them;

3. One-to-many associations, as PLACE OF BIRTH, LOCATION, BEING IN CHARGE OF, T/O, LND, AUTHORIZING, REFERENCING, INSIDE, ALTERNATE, FLYING, SCHEDULING disappear and, simultaneously, identifiers from the many side are renamed and added to the set of attributes of the one side. The added attributes become foreign keys, so they embody the reference to the other relation;

4. One-to-one associations, as PLACING AND SETTING, have been translated including their identifiers into ATC UNIT relation;

5. Due to the high volume of data, we decided to add a counter in FLIGHT in order to help users to manage the flights inserted in and operated by the system.

The whole relational scheme follows:

FLIGHTS (*Counter, AirlineID, IDnumber, T/Odate, T/Otime, LNDdate, LNDtime, IFFcode, Activity, Status, ATMauthorization, Responsible*:PERSONNEL, *Aircraft*:AIRCRAFTS, *Airline*:AIRLINES, *Departure*:AIRPORTS, *Arrival*:AIRPORTS, *Authorization*:ATC UNITS)

STOPOVERS (*Flight*:FLIGHTS, *Airport*:AIRPORTS, *T/Otime, LNDtime*)

AIRCRAFTS (*ICAOcode, Model, Manufacturer, Category, Dimension, MTOW, Passengers*)

CREWS (*Flight*:FLIGHTS, *Person*:PERSONNEL, *Role*)

OWNINGS (*Aircraft*:AIRCRAFTS, *Airline*:AIRLINES)

AIRPORTS (*ICAOcode, IATAcode, Name, KindOfTraffic, Latitude, Longitude, ColorCode, ATCunit*:ATC UNITS, *City*:CITIES)

CATEGORIES (*Category*)

RELATINGS (*Airport*:AIRPORTS, *Category*:CATEGORIES)

SERVICES (*Service*)

PROVIDINGS (*Airport*:AIRPORTS, *Service*:SERVICES)

RWYS (*Heading, Position, Airport*:AIRPORTS, *TORA, TODA, ASDA, LDA, EMDA*)

ALTERNATES (*Flight*:FLIGHTS, *Airport*:AIRPORTS)

HUBS (*Airport*:AIRPORTS, *Airline*:AIRLINES)

SUITABILITIES (*Airport*:AIRPORTS, *Aircraft*:AIRCRAFTS)

PERSONNEL (*SSN, Name, Surname, DateOfBirth, PlaceOfBirth*:CITIES, *LastFlightTime, Type*)

WORKINGS (*Airline*:AIRLINES, *Person*:PERSONNEL)

ABILITATIONS (*Aircraft*:AIRCRAFTS, *Person*:PERSONNEL)

QUALIFICATIONS (*Qualification*)

QUALIFYING (*Person*:PERSONNEL, *Qualification*:QUALIFICATIONS)

Cities (*Name, Country, ICAOregion*:ICAO Regions)

Headquarters (*Airline*:Airlines, *City*:Cities)

ICAO Regions (*ICAOregion*)

ATC Units (*ICAOregion*:ICAO Regions, *City*:Cities)

Airlines (*IDcode, Name, WebSite, HomeCountry*)

Contacts (*Airline*:Airlines, *Administrative, Emergency*)

2.4 Physical design

The physical design of a database is composed of:

- Data dimension of a database, global and for each object, in terms of storage sizing needed on the HDD;
- Creation of the database;
- Definition of security policies and creation of users and roles;
- Creation of objects and definition of constraints;
- Creation of tuples.

The steps needed in the physical project, with this specific operational condition, are described below:

- Install ORACLE 11g;
- Server set up with 4CPU 3.00GHz, 6GB Ram, HDD with 500GB;
- Server OS: Windows 2003 Advanced Server.

Table 2.2: Table ABILITATIONS sizing.

Attribute	Type	Byte
Person	VARCHAR2	25
Aircraft	CHAR	6

Table 2.3: Table ARICRAFTS sizing.

Attribute	Type	Byte
ICAOcode	CHAR	6
Model	VARCHAR2	10
Manufacturer	VARCHAR2	50
Category	VARCHAR2	20
Dimension	VARCHAR2	10
MTOW	NUMBER	5
Passengers	NUMBER	5

Physical sizing of the database

It is possible to make an evaluation of the memory space capabilities of the database exploiting data information and the attributes' types used in the implementation of each table. The memory space, in terms of bytes, is:

- NUMBER(x): ((x/2)+2) byte;
- DATE: 7 byte;
- CHAR(x): x byte;
- VARCHAR (x): 0...x byte.

An evaluation of the space memory required follows. In each table the worst case is considered.

Table 2.4: Table AIRLINES sizing.

Attribute	**Type**	**Byte**
IDcode	VARCHAR2	15
Name	VARCHAR2	30
Website	VARCHAR2	30
HomeCountry	VARCHAR2	30

Table 2.5: Table AIRPORTS sizing.

Attribute	**Type**	**Byte**
ICAOcode	CHAR	4
IATAcode	CHAR	3
Name	VARCHAR2	50
KindOftraffic	VARCHAR2	20
Latitude	VARCHAR2	20
Longitude	VARCHAR2	20
ColorCode	VARCHAR2	10
ATCunit	VARCHAR2	30
City	VARCHAR2	15

Table 2.6: Table ALTERNATES sizing.

Attribute	**Type**	**Byte**
Flight	NUMBER	5
Airport	CHAR	4

Table 2.7: Table ATC UNITS sizing.

Attribute	**Type**	**Byte**
ICAOregion	VARCHAR2	15
City	VARCHAR2	15

Table 2.8: Table CATEGORIES sizing.

Attribute	**Type**	**Byte**
Category	VARCHAR2	30

Table 2.9: Table Cities sizing.

Attribute	Type	Byte
Name	VARCHAR2	15
Country	VARCHAR2	30
ICAOregion	VARCHAR2	15

Table 2.10: Table Crews sizing.

Attribute	Type	Byte
Airline	VARCHAR2	15
Administrative	VARCHAR2	15
Emergency	VARCHAR2	15

Table 2.11: Table Crews sizing.

Attribute	Type	Byte
Flight	NUMBER	5
Person	VARCHAR2	25
Role	VARCHAR2	20

Table 2.12: Table Flights sizing.

Attribute	Type	Byte
Counter	NUMBER	
AirlineID	VARCHAR	15
IDnumber	CHAR	4
TakeOffDate	TIMESTAMP	6
LNDdate	TIMESTAMP	6
IFFcode	CHAR	4
Activity	VARCHAR2	3
Status	VARCHAR2	20
ATMauthorization	VARCHAR2	10
Responsible	VARCHAR2	25
Aircraft	CHAR	6
Airline	VARCHAR2	30
Departure	CHAR	4
Arrival	CHAR	4
Authorization	VARCHAR2	15

Table 2.13: Table HEADQUARTERS sizing.

Attribute	Type	Byte
Airline	VARCHAR2	15
City	VARCHAR2	15

Table 2.14: Table HUBS sizing.

Attribute	Type	Byte
Airport	CHAR	4
Airline	VARCHAR2	15

Table 2.15: Table ICAO REGIONS sizing.

Attribute	Type	Byte
ICAOregion	VARCHAR2	15

Table 2.16: Table OWNINGS sizing.

Attribute	Type	Byte
Aircraft	CHAR	6
Airline	VARCHAR2	15

Table 2.17: Table PERSONNEL sizing.

Attribute	Type	Byte
SSN	VARCHAR2	25
Name	VARCHAR2	15
Surname	VARCHAR2	15
DateOfBirth	DATE	7
PlaceOfBirth	VARCHAR2	15
LastFlightTime	TIMESTAMP	6
Type	VARCHAR2	20

Table 2.18: Table PROVIDINGS sizing.

Attribute	Type	Byte
Airport	CHAR	4
Service	VARCHAR2	30

Table 2.19: Table QUALIFICATIONS sizing.

Attribute	**Type**	**Byte**
Qualification	VARCHAR2	30

Table 2.20: Table QUALIFYING sizing.

Attribute	**Type**	**Byte**
Person	VARCHAR2	25
Qualification	VARCHAR2	30

Table 2.21: Table RELATINGS sizing.

Attribute	**Type**	**Byte**
Airport	CHAR	4
Category	VARCHAR2	30

Table 2.22: Table RWYS sizing.

Attribute	**Type**	**Byte**
Heading	NUMBER	5
Position	CHAR	1
Airport	CHAR	4
TORA	NUMBER	5
TODA	NUMBER	5
ASDA	NUMBER	5
LDA	NUMBER	5
EMDA	NUMBER	5

Table 2.23: Table SERVICES sizing.

Attribute	**Type**	**Byte**
Service	VARCHAR2	30

Table 2.24: Table STOPOVERS sizing.

Attribute	**Type**	**Byte**
Flight	NUMBER	5
Airport	CHAR	4
TakeOffTime	TIMESTAMP	6
LNDtime	TIMESTAMP	6

Table 2.25: Table SUITABILITIES sizing.

Attribute	**Type**	**Byte**
Airport	CHAR	4
Aircraft	CHAR	6

Table 2.26: Table WORKINGS sizing.

Attribute	**Type**	**Byte**
Airline	VARCHAR2	15
Person	VARCHAR2	25

Chapter 3

DB configuration

3.1 DB creation/SQL development

Tablespace creation

The database was programmed on *SQL developer*.

Inside the architecture of an ORACLE database it is possible to distinguish between the logical and physical structure.

Physical structures are determined by those OS' files that make up the database (i.e. data file, control file, redo log file). On the contrary, logical structures describe logical areas of memorization (tablespace), where the objects can be saved.

The code for creating the tablespace follows:

```
1 CREATE TABLESPACE fms_db DATAFILE 'fms_db.dbf' SIZE 200M;
```

User creations and security

Regarding to specifications and security policies, we have decided to create a DBA user, that owns the database, and three roles: these are related to pilots, airline companies and ATC units and have different privileges on the database.

The script for creating the user and the roles follows:

```
1 CREATE USER brug_dba DEFAULT TABLESPACE fms_db identified
    by 12345;
2 GRANT DBA, CONNECT, RESOURCE, UNLIMITED TABLESPACE TO
    brug_dba;
3
4 CREATE ROLE Pilot;
```

```
GRANT CONNECT TO pilot;
GRANT SELECT, DELETE, UPDATE ON brug_dba.FLIGHTS TO pilot;
GRANT SELECT, INSERT, DELETE, UPDATE ON brug_dba.
    ALTERNATES TO pilot;
GRANT SELECT, INSERT, DELETE, UPDATE ON brug_dba.CREWS TO
    pilot;

CREATE ROLE Airline_company;
GRANT CONNECT TO Airline_company;
GRANT SELECT, INSERT, DELETE, UPDATE ON brug_dba.FLIGHTS
    TO Airline_company;
GRANT SELECT, UPDATE ON brug_dba.AIRCRAFTS TO
    Airline_company;
GRANT SELECT, ALTER ON brug_dba.FLIGHTS_Counter_seq TO
    Airline_company;

CREATE ROLE ATC_unit;
GRANT CONNECT TO ATC_unit;
GRANT SELECT, UPDATE ON brug_dba.FLIGHTS TO ATC_unit;
GRANT SELECT ON brug_dba.AIRLINES TO ATC_unit;
GRANT SELECT ON brug_dba.CONCTACTS TO ATC_unit;
GRANT SELECT ON brug_dba.WORKINGS TO ATC_unit;
GRANT SELECT ON brug_dba.OWNINGS TO ATC_unit;
GRANT SELECT ON brug_dba.ABILITATIONS TO ATC_unit;
GRANT SELECT, UPDATE ON brug_dba.AIRPORTS TO ATC_unit;
```

Objects creation

The script below is used for the creation of all the objects of the database. A counter sequence is also added. The creation of integrity constraints is obtained by using ALTER TABLE.

```
CREATE TABLE FLIGHTS(
    Counter number primary key,
    AirlineID varchar(15) NOT NULL,
    IDnumber char(4) NOT NULL,
    TakeOffDate TIMESTAMP WITH TIME ZONE,
    LNDdate TIMESTAMP WITH TIME ZONE,
    IFFcode char(4) NOT NULL,
```

```
    Activity varchar(3),
    Status varchar(20),
    ATMauthorization varchar(10) NOT NULL,
    Responsible varchar(25),
    Aircraft char(6),
    Airline varchar(30),
    Departure char(4) NOT NULL,
    Arrival char(4) NOT NULL,
    Authorization varchar(15)
);
CREATE SEQUENCE FLIGHTS_Counter_seq MINVALUE 1 INCREMENT
   BY 1 START WITH 1 NOCYCLE;
CREATE TABLE AIRCRAFTS(
    ICAOcode char(6) primary key,
    Model varchar(10) NOT NULL,
    Manufacturer varchar(50),
    Category varchar(20) NOT NULL,
    Dimensions varchar(10) NOT NULL,
    MTOW int NOT NULL,
    Passengers int
);
CREATE TABLE AIRPORTS(
    ICAOcode char(4) primary key,
    IATAcode char(3) UNIQUE,
    Name varchar(50) UNIQUE NOT NULL,
    KindOfTraffic varchar(20),
    Latitude varchar(20),
    Longitude varchar(20),
    ColorCode varchar(10) NOT NULL,
    ATCunit varchar(30),
    City varchar(15)
);
CREATE TABLE CATEGORIES(
    Category varchar(30) primary key
);
CREATE TABLE SERVICES(
    Service varchar(30) primary key
);
```

```
CREATE TABLE RWYs(
    Heading int,
    Position char(1),
    Airport char(4),
    TORA int NOT NULL,
    TODA int NOT NULL,
    ASDA int NOT NULL,
    LDA int NOT NULL,
    EMDA int NOT NULL,
    foreign key (Airport) REFERENCES AIRPORTS(ICAOcode)
    on delete CASCADE,
    primary key (Heading, Position, Airport)
);
CREATE TABLE ICAO_REGIONS(
    ICAOregion varchar(15) primary key
);
CREATE TABLE CITIES(
    Name varchar(15) primary key,
    Country varchar(30),
    ICAOregion varchar(15),
    foreign key (ICAOregion) REFERENCES ICAO_REGIONS(
   ICAOregion)
);
CREATE TABLE PERSONNEL(
    SSN varchar(25) primary key,
    Name varchar(15) NOT NULL,
    Surname varchar(15) NOT NULL,
    DateofBirth date NOT NULL,
    PlaceofBirth varchar(15) NOT NULL,
    Lastflighttime TIMESTAMP,
    Type varchar(20),
    foreign key (PlaceOfBirth) REFERENCES CITIES(Name)
);
CREATE TABLE STOPOVERS(
    Flight number,
    Airport char(4),
    TakeOffTime TIMESTAMP,
    LNDtime TIMESTAMP,
```

```
    primary key (Flight, Airport),
    foreign key (Flight) REFERENCES FLIGHTS(Counter),
    foreign key (Airport) REFERENCES AIRPORTS(ICAOcode)
);
CREATE TABLE CREWS(
    Flight number,
    Person varchar(25),
    Role varchar(20),
    primary key (Flight, Person),
    foreign key (Flight) REFERENCES FLIGHTS(Counter),
    foreign key (Person) REFERENCES PERSONNEL(SSN)
);
CREATE TABLE AIRLINES(
    IDcode varchar(15) primary key,
    Name varchar(30) UNIQUE NOT NULL,
    Website varchar(30),
    HomeCountry varchar(30)
);
CREATE TABLE OWNINGS(
    Aircraft char(6),
    Airline varchar(15),
    primary key (Aircraft, Airline),
    foreign key (Aircraft) REFERENCES AIRCRAFTS(ICAOcode),
    foreign key (Airline) REFERENCES AIRLINES(IDcode)
);
CREATE TABLE RELATINGS(
    Airport char(4),
    Category varchar(30),
    primary key (Airport, Category),
    foreign key (Airport) REFERENCES AIRPORTS(ICAOcode),
    foreign key (Category) REFERENCES CATEGORIES(Category)
);
CREATE TABLE PROVIDINGS(
    Airport char(4),
    Service varchar(30),
    primary key (Airport, Service),
    foreign key (Airport) REFERENCES AIRPORTS(ICAOcode),
    foreign key (Service) REFERENCES SERVICES(Service)
```

```
);
CREATE TABLE ALTERNATES(
    Flight number,
    Airport char(4),
    primary key (Flight, Airport),
    foreign key (Airport) REFERENCES AIRPORTS(ICAOcode),
    foreign key (Flight) REFERENCES FLIGHTS(counter)
);
CREATE TABLE  HUBS(
    Airport char(4),
    Airline varchar(15),
    primary key (Airport, Airline),
    foreign key (Airport) REFERENCES AIRPORTS(ICAOcode),
    foreign key (Airline) REFERENCES AIRLINES(IDcode)
);
CREATE TABLE  SUITABILITIES(
    Airport char(4),
    Aircraft char(6),
    primary key (Airport, Aircraft),
    foreign key (Airport) REFERENCES AIRPORTS(ICAOcode),
    foreign key (Aircraft) REFERENCES AIRCRAFTS(ICAOcode)
);
CREATE TABLE  WORKINGS(
    Person varchar(25),
    Airline varchar(15),
    primary key (Person, Airline),
    foreign key (Airline) REFERENCES AIRLINES(IDcode),
    foreign key (Person) REFERENCES PERSONNEL(SSN)
);
CREATE TABLE  ABILITATIONS(
    Person varchar(25),
    Aircraft char(6),
    primary key (Person, Aircraft),
    foreign key (Aircraft) REFERENCES AIRCRAFTS(ICAOcode),
    foreign key (Person) REFERENCES PERSONNEL(SSN)
);
CREATE TABLE QUALIFICATIONS(
    Qualification varchar(30) primary key
```

```
158 );
159 CREATE TABLE QUALIFYING(
160     Person varchar(25),
161     Qualification varchar(30),
162     primary key (Person, Qualification),
163     foreign key (Person) REFERENCES PERSONNEL(SSN),
164     foreign key (Qualification) REFERENCES QUALIFICATIONS(
    Qualification)
165 );
166 CREATE TABLE HEADQUARTERS(
167     Airline varchar(15),
168     City varchar(15),
169     primary key (Airline, City),
170     foreign key (Airline) REFERENCES AIRLINES(IDcode),
171     foreign key (City) REFERENCES CITIES(Name)
172 );
173 CREATE TABLE ATC_UNITS(
174     ICAOregion varchar(15) primary key,
175     City varchar(15),
176     foreign key (ICAOregion) REFERENCES ICAO_REGIONS(
    ICAOregion),
177     foreign key (City) REFERENCES CITIES(Name)
178 );
179 CREATE TABLE CONTACTS(
180     Airline varchar(15) primary key,
181     Administrative varchar(15) UNIQUE NOT NULL,
182     Emergency varchar(15) UNIQUE NOT NULL,
183     foreign key (Airline) REFERENCES AIRLINES(IDcode)
184 );
185
186 ALTER TABLE FLIGHTS
187 ADD CONSTRAINT FK_FLI_PER foreign key (Responsible)
188 REFERENCES PERSONNEL(SSN);
189
190 ALTER TABLE FLIGHTS
191 ADD CONSTRAINT FK_FLI_ACFT foreign key (Aircraft)
192 REFERENCES AIRCRAFTS(ICAOcode);
193
```

```
194 ALTER TABLE FLIGHTS
195 ADD CONSTRAINT FK_FLI_ALN foreign key (Airline) REFERENCES
      AIRLINES(Name);
196
197 ALTER TABLE FLIGHTS
198 ADD CONSTRAINT FK_FLI_ALNID foreign key (AirlineID)
     REFERENCES AIRLINES(IDcode);
199
200 ALTER TABLE FLIGHTS
201 ADD CONSTRAINT FK_FLI_DEP foreign key (Departure)
202 REFERENCES AIRPORTS(ICAOcode);
203
204 ALTER TABLE FLIGHTS
205 ADD CONSTRAINT FK_FLI_ARR foreign key (Arrival)
206 REFERENCES AIRPORTS(ICAOcode);
207
208 ALTER TABLE FLIGHTS
209 ADD CONSTRAINT FK_FLI_AUT foreign key (Authorization)
210 REFERENCES ATC_UNITS(ICAOregion);
211
212 ALTER TABLE AIRPORTS
213 ADD CONSTRAINT FK_APT_ATC foreign key (ATCunit)
214 REFERENCES ATC_UNITS(ICAOregion)
215 on delete SET NULL;
216
217 ALTER TABLE AIRPORTS
218 ADD CONSTRAINT FK_APT_CIT foreign key (City)
219 REFERENCES CITIES(Name)
220 on delete SET NULL;
```

Data insert

The data insert is provided by the SQL statement INSERT, or using automatic import procedures.