Normalization Brian Air database

Table of Contents

[Introduction 2](#_Toc388020296)

[Normalization 3](#_Toc388020297)

[Definitions of the normal forms 3](#_Toc388020298)

[1NF 3](#_Toc388020299)

[2NF 3](#_Toc388020300)

[3NF 3](#_Toc388020301)

[BCNF 3](#_Toc388020302)

[Tables to normalize 3](#_Toc388020303)

[\_weekday 3](#_Toc388020304)

[passenger\_factor 4](#_Toc388020305)

[city 4](#_Toc388020306)

[airplane 5](#_Toc388020307)

[passengers 5](#_Toc388020308)

[credit\_card 6](#_Toc388020309)

[Participates 6](#_Toc388020310)

[route 7](#_Toc388020311)

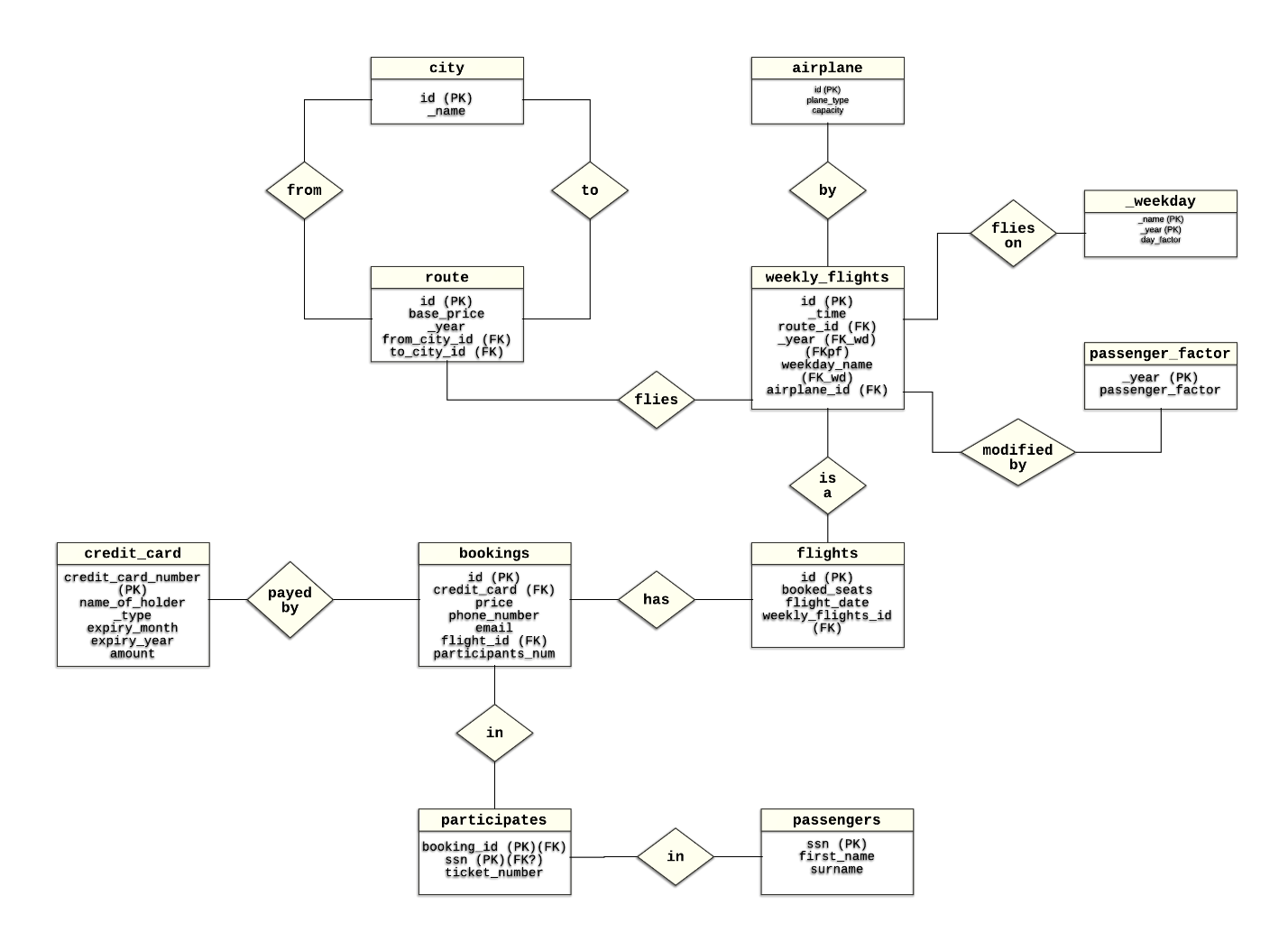
[weekly\_flights 7](#_Toc388020312)

[flights 8](#_Toc388020313)

[bookings 8](#_Toc388020314)

# Introduction

When the database tables where first constructed they had dependencies according to the diagram below [Har inte med multiplar då antalet element i lucidcharts tog slut (typ 5 kvar), du kanske kan skarva det Tobbe?]:



In this document the tables will be normalized step-by-step.

# Normalization

The normalization will be performed to the stage of BCNF one table at the time, starting with the most basic tables.

Each table will have its own sub title where the attributes are declared and explained. Assumptions regarding attribute dependencies will also be presented

An assumed requirement for the database is that it is supposed to be viable for several years to come.

## Definitions of the normal forms

### 1NF

The table does not contain any attribute consisting of non-atomic values.

### 2NF

Non-prime attributes cannot be functionally dependent on a part of a candidate key / every non-prime attribute is fully functionally dependent on every candidate key

### 3NF

Non-prime attributes cannot be functionally dependent on a set of attributes that is not a candidate key / non-prime attributes are not transitively dependent on any candidate key

### BCNF

Every determinant is a candidate key

# Tables to normalize

## \_weekday

Every tuple contains a day\_factor, which is used to calculate the price of a ticket.

#### Attributes

The \_weekday table contains three attributes: \_year, \_name and day\_factor. Where \_year and \_name specifies which year and what day the day\_factor applies to respectively. day\_factor is used to calculate the price of a ticket and may differ from day to day and year to year. To determine a specific day\_factor both \_name and \_year are needed.

#### Dependencies

Primary key: (\_name, \_year)

Functional dependencies: (\_name, \_year) 🡪 (day\_factor)

No other candidate keys except from the primary key

#### Normal form

1NF: None of the values are non-atomic, therefore \_weekday satisfies the requirements of 1NF.

2NF: The only functional dependency contains the whole primary key as a determinant, satisfies 2NF.

3NF: day\_factor is dependent on the primary key, satisfies 3NF.

BCNF: The determinant is a candidate key, satisfies BCNF.

## passenger\_factor

Every tuple contains a passenger\_factor, used to determine the price of a ticket.

#### Attributes

The table contains two attributes: \_year and passenger\_factor. The passenger\_factor may differ from year to year and is used to calculate the price of a ticket.

#### Dependencies

Primary key: \_year

Functional dependencies: \_year 🡪 passenger\_factor

No other candidate keys, except from the primary key.

#### Normal form

1NF: None of the values are non-atomic, therefore passenger\_factor satisfies the requirements of 1NF.

2NF: The only functional dependency contains the whole primary key as a determinant, satisfies 2NF.

3NF: passenger\_factor is dependent on the primary key, satisfies 3NF.

BCNF: The determinant is a candidate key, satisfies BCNF.

## city

Every tuple contains information about a city.

#### Attributes

The table contains two attributes: id and \_name. Where id is an integer representing the city and city is the name of the city.

#### Dependencies

Primary key: id

Functional dependencies: id 🡪 \_name

No other candidate keys except from the primary key.

#### Normal form

1NF: None of the values are non-atomic, therefore city satisfies the requirements of 1NF.

2NF: The only functional dependency contains the whole primary key as a determinant, satisfies 2NF.

3NF: city is dependent on the primary key, satisfies 3NF.

BCNF: The determinant is a candidate key, satisfies BCNF.

## 

## airplane

Every tuple in airplane contains information about a specific airplane in Brian Airs air fleet.

#### Attributes

The table contains three attributes: id, plane\_type and capacity. The id is an integer referring to a specific airplane in Brian Airs air fleet, plane\_type is the model of the aircraft and capacity is the amount of passengers this aircraft can hold.

#### Dependencies

Primary key: id (Since Brian Air can have several aircrafts of the same type)

Functional dependencies:

id 🡪 plane\_type, capacity

No other candidate key except for the primary key. id is a prime attribute, while plane\_type and capacity are non-prime attributes.

#### Normal form

1NF: None of the attributes are atomic values, airplane therefore satisfy 1NF

2NF: id is the whole primary key. Therefore airplane satisfy 2Nf

3NF: The non-prime attributes are referenced by only the primary key, no conflict with 3NF.

BCNF: id is the primary key, therefore airplane satisfy BCNF

## passengers

Every tuple in passengers contains information about the people taking part in a flight.

#### Attributes

The table contains three attributes: ssn, first\_name and surname. Where ssn is the social security number of the person, first\_name is the first name of the person and surname is the person’s surname.

#### Dependencies

Primary key: ssn (which is unique to every person, while first names and surnames can be shared)

Functional dependencies: ssn 🡪 first\_name, surname

No other candidate key except from the primary key. ssn is a prime attribute while first\_name and surname are non-prime attributes

#### Normal form

1NF: None of the attributes are atomic values, passengers therefore satisfy 1NF

2NF: ssn is the whole primary key and the only determinant. Therefore passengers satisfy 2NF

3NF: ssn is the whole primary key and the only determinant. Therefore passengers satisfy 3NF

BCNF: ssn is the whole primary key and the only determinant. Therefore passengers satisfy BCNF

## credit\_card

Every tuple in the table contains information about the credit card which is used to pay for a booking. There is only one tuple for e credit card even though it is used to pay for several bookings.

#### Attributes

The table contains six attributes: credit\_card\_number, name\_of\_holder, \_type, expiry\_month, expiry\_year and amount. credit\_card\_number is the unique number which identifies the credit card, name\_of\_holder is the first name and the surname of the person that registered the card, \_type is the name of the corporation issuing the card, expiry\_month is an integer representing the month the card expires, expiry\_year is an integer representing the year the card expires and amount is the sum of all purchases made by the card from Brian Air.

#### Dependencies

Primary key: credit\_card\_number

Functional dependencies: credit\_card\_number 🡪 name\_of\_holder, \_type, expiry\_month, expiry\_year, amount

There are no other candidate keys except for the primary key. credit\_card\_number is the only prime attribute.

#### Normal form

There is only functional dependency, in which the primary key is the determinant and all other attributes are non-prime and dependent of it. None of the attributes are non-atomic. Therefore the relation satisfies BCNF.

## Participates

Every tuple in participates describes which booking a certain passenger belongs to, and what ticket number it has for that flight.

#### Attributes

There are three attributes in the table: booking\_id, ssn and ticket number. booking\_id is a reference to which booking the passenger belong to, ssn is a reference to identify a passenger and ticket number is the ticket number issued for a flight.

#### Dependencies

Primary key:   
The combination of booking\_id and ssn is the primary key. (A ticket\_number is not generated for reservations)  
booking\_id cannot be a primary key alone, since several persons can be included in one booking, while ssn cannot be the primary key alone since a single person can have made several bookings.

Foreign keys: booking\_id references the id in the bookings table and ssn references ssn in the passengers table.

Functional dependencies: booking\_id, ssn 🡪 ticket\_number

There are no other candidate keys except for the primary key. booking\_id and ssn are prime attributes and ticket\_number is a non-prime attribute

#### Normal form

1NF: None of the attributes are non-atomic, satisfies 1NF.

2NF: The determinant of the only functional dependency is the whole primary key, satisfies 2NF.

3NF: The non-prime attribute is depending on the primary key, satisfies 3NF

BCNF: The only determinant is the primary key, satisfies BCNF.

## route

Every tuple in route defines a one way flight between two cities.

#### Attributes

The table consists of five attributes: id, base\_price, \_year, to\_city\_id and from\_city\_id. id is an integer identifying a unique route, base\_price is an integer used to calculate the price of a ticket, \_year is an integer representing the year the route is valid (base\_price may differ from year to year), to\_city\_id is the id of the city that is the destination of the flight, from\_city\_id is the id of the city from where the flight departs.

#### Dependencies

Primary key: id

Foreign keys: to\_city\_id and from\_city\_id are both references to id in the city table.

Functional dependencies: id 🡪 base\_price, \_year, to\_city\_id, from\_city\_id

There are no other candidate keys except from the primary key. id is a prime attribute while the other attributes are non-prime attributes.

#### Normal form

1NF: None of the attributes are non-atomic, satisfies 1NF.

2NF: The non-prime attributes are dependent on the whole primary key, satisfies 2NF.

3NF: The non-prime attributes are dependent on the primary key (candidate key), satisfies 3NF.

BCNF: The only determinant is the primary key, satisfies BCNF

## weekly\_flights

Every tuple in weekly flights contains information about one of the flights that are scheduled for every week.

#### Attributes

The table contains six attributes: id, \_time, route\_id, \_year, weekday\_name, airplane\_id.   
id is an integer that identifies a specific weekly\_flight, \_time is the time of day when the flight will depart, route\_id is an integer that identifies the departure and destination of the flight, \_year is which year the weekly\_flight flies (may only fly one specific year), weekday\_name is the day of the week when the flight goes, airplane\_id is an integer that specifies which airplane is taking the flight,

#### Dependencies

Primary key: id

Functional dependencies: id 🡪 \_time ,route\_id, \_year, weekday\_name, airplane\_id

There are no other candidate keys except from the primary key. id is a prime attribute while the others are non-prime attributes.

#### Normal form

1NF: All the attributes in weekly\_flights are atomic, belongs to 1NF.

2NF: All non-prime attributes are dependent on the whole candidate key, belongs to 2NF.

3NF: All non-prime attributes are dependent on the whole candidate key, belongs to 3NF.

BCNF: The only determinant is the primary key, therefore weekly\_flights belong to BCNF.

## flights

Every tuple in flights contain information about a specific flight flew by Brian Air.

#### Attributes

The table contains four attributes: id, booked\_seats, flight\_date, weekly\_flights\_id. id is an integer that identifies a unique flight, booked\_seats is an integer representing the number of seats on the flight that has been booked, flight\_date is the date of departure for the flight and weekly\_flights\_id is the id of which weekly flight the flight corresponds to.

#### Dependencies

Primary key: id

Functional dependencies: id 🡪 booked\_seats, flight\_date, weekly\_flights\_id

There are no other candidate keys except for the primary key. id is a prime attribute while booked\_seats, flight\_date and weekly\_flights\_id are non-prime attributes.

#### Normal form

1NF: none of the attributes are non-atomic, flights belong to 1NF.

2NF: no non-prime attribute is dependent on part of a candidate key, flights belong to 2NF.

3NF: no non-prime attribute is dependent on a determinant that is not a candidate key, flights belong to 3NF

BCNF: The determinant is the primary key, flights belong to BCNF.

## bookings

Every tuple in bookings contains information about a specific booking.

#### Attributes

The table contains seven attributes: id, credit\_card, price, phone\_number, email, flight\_id, participants\_num.  
id is an integer that identifies a unique booking, credit\_card is the credit card number of the card used to pay for the booking, price is the price of one ticket [ELLER ÄR DET TOTALPRISET!?!?] in the booking, phone\_number is the telephone number of the person that performed the booking, email is the e-mail address of the person that performed the booking, flight\_id is an integer that identifies which flight that booking is made for and participants\_num is an integer representing the number of people that is included in the booking.

#### Dependencies

Primary key: id

Functional dependencies: id 🡪 credit\_card, price, phone\_number, email, flight\_id, participants\_num

There are no other candidate keys except for the primary key. id is a prime attribute while all other attributes are non-prime attributes.

#### Normal form

1NF: There are no non-atomic attributes, therefore flights belong to 1NF.

2NF: No non-prime attribute is dependent on part of a candidate key, flights belong to 2NF.

3NF: The non-prime attributes are all dependent on the primary key, flights belong to 3NF.

BCNF: The determinant of the functional dependency is the primary key, flights belong to BCNF.