5

Logical Data Warehouse Design

Exercises

5.1 Consider the data warehouse of a telephone provider given in Ex. 3.1. Draw a star schema diagram for the data warehouse.

Answer A star schema is shown in Fig. 5.1.

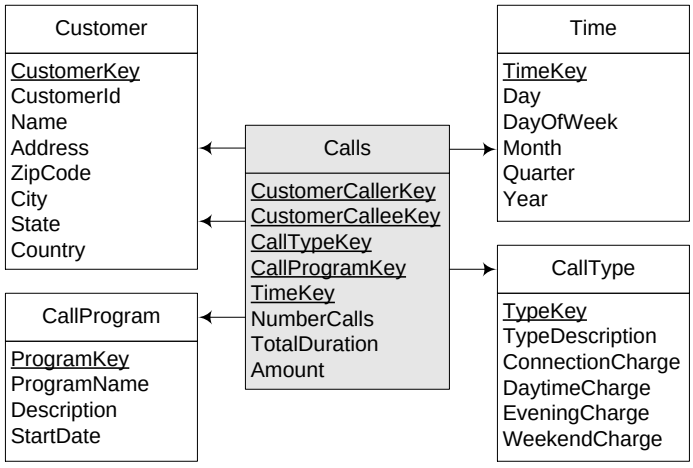

Fig. 5.1. A star schema for the data warehouse of Ex. 3.1

5.2 For the star schema obtained in the previous exercise, write in SQL the following queries.

(a) List the total amount collected by each call program in 2012.

```
SELECT    ProgramName, SUM(Amount)
```

```
FROM      Calls C, Time T, CallProgram P
WHERE     C.TimeKey = T.TimeKey AND T.Year = 2012 AND
          C.CallProgramKey = P.ProgramKey
GROUP BY ProgramName
```

(b) List the total duration of calls made by customers from Brussels in 2012.

```
SELECT    SUM(TotalDuration)
FROM      Calls C, Time T, Customer U
WHERE     C.TimeKey = T.TimeKey AND T.Year = 2012 AND
          C.CustomerFromKey = U.CustomerKey AND
          C.City = 'Brussels'
```

(c) List the total number of weekend calls made by customers from Brussels to customers in Antwerp in 2012.

```
SELECT    SUM(NumberCalls)
FROM      Calls C, Time T, Customer F, Customer To
WHERE     C.TimeKey = T.TimeKey AND T.Year = 2012 AND
          ( T.DayOfWeek = 'Saturday' OR
          T.DayOfWeek = 'Saturday') AND
          C.CustomerFromKey = F.CustomerKey AND
          C.CustomerToKey = To.CustomerKey AND
          F.City = 'Brussels' AND To.City = 'Antwerp'
```

(d) List the total duration of international calls started by customers in Belgium in 2012.

```
SELECT    SUM(TotalDuration)
FROM      Calls C, Time T, Customer F, Customer To
WHERE     C.TimeKey = T.TimeKey AND T.Year = 2012 AND
          C.CustomerFromKey = F.CustomerKey AND
          C.CustomerToKey = To.CustomerKey AND
          F.Country = 'Belgium' AND To.Country <> 'Belgium'
```

(e) List the total amount collected from customers in Brussels who are enrolled in the corporate program in 2012.

```
SELECT    SUM(Amount)
FROM      Calls C, Time T, Customer U, CallProgram P
WHERE     C.TimeKey = T.TimeKey AND T.Year = 2012 AND
          C.CustomerFromKey = U.CustomerKey AND
          C.CallProgramKey = P.CallProgramKey AND
          C.City = 'Brussels' AND P.ProgramName = 'Corporate'
```

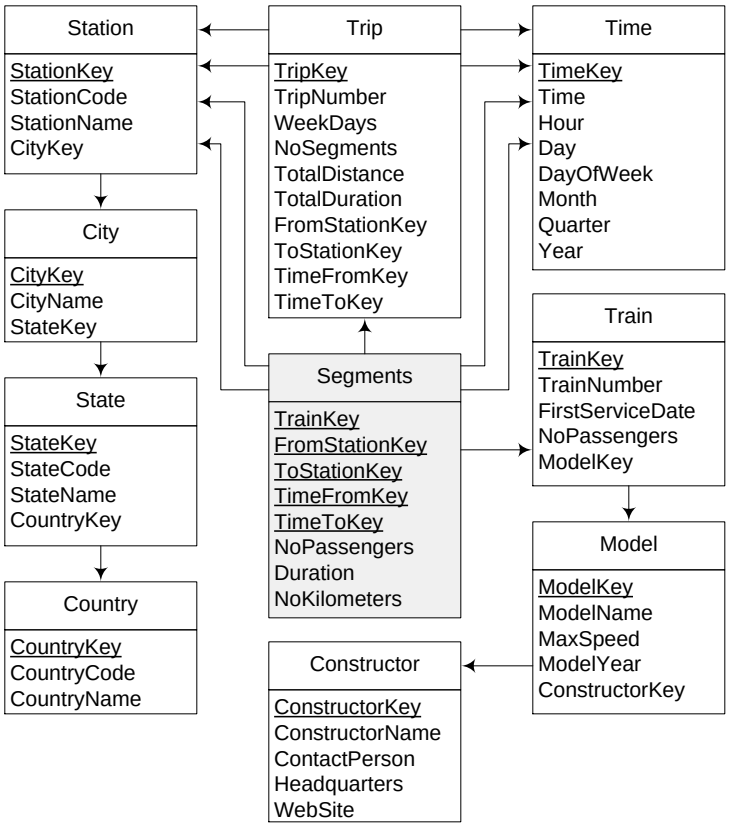

Fig. 5.2. A snowflake schema for the data warehouse of Ex. 3.2

5.3 Consider the data warehouse of the train application given in Ex. 3.2. Draw a snowflake schema diagram for the data warehouse with hierarchies for the train and station dimensions.

Answer A snowflake schema is shown in Fig. 5.2.

5.4 For the snowflake schema obtained in the previous exercise, write in SQL the following queries.

(a) List the total number of kilometers made by Alstom trains during 2012 departing from French or Belgian stations.

```
SELECT SUM(NoKilometers)
FROM   Segments F, Time T, Train TR, Model M,
       Constructor C, Station S, City CI, State ST, Country CO
WHERE  F.TimeFromKey = T.TimeKey AND T.Year = '2012' AND
       F.TrainKey = TR.TrainKey AND
       TR.ModelKey = M.ModelKey AND
```

```
        M.ConstructorKey = C.ConstructorKey AND
        C.ConstructorName = 'Alstom' AND
        F.FromStationKey = S.StationKey AND
        S.CityKey = CI.CityKey AND
        CI.StateKey = ST.StateKey AND
        ST.CountryKey = CO.CountryKey AND
        ( CO.CountryName = 'France' OR
        CO.CountryName = 'Belgium' )
```

(b) List the total duration of international trips during 2012, that is, trips departing from a station located in a country and arriving at a station located in another country.

```
SELECT SUM(Duration)
FROM   Segments F, Time T, Station A1, City C1, State S1,
       Station A2, City C2, State S2
WHERE  F.TimeFromKey = T.TimeKey AND T.Year = '2012' AND
       F.FromStationKey = A1.StationKey AND
       A1.CityKey = C1.CityKey AND
       C1.StateKey = S1.StateKey AND
       F.ToStationKey = A2.StationKey AND
       A2.CityKey = C2.CityKey AND
       C2.StateKey = S2.StateKey AND
       S1.CountryKey <> S2.CountryKey
```

(c) List the total number of trains that departed from or arrived at Paris during July 2012.

```
SELECT COUNT(*)
FROM   Segments F, Time T, Trip TR, Station S, City C
WHERE  F.TimeFromKey = T.TimeKey AND
       T.Month = 'July' AND T.Year = '2012' AND
       ( F.FromStationKey = S.StationKey OR
       F.ToStationKey = S.StationKey ) AND
       S.CityKey = C.CityKey AND
       C.CityName = 'Paris'
```

(d) List the average duration of train segments in Belgium in 2012.

```
SELECT SUM(Duration)
FROM   Segments F, Time T, Station A1, City C1, State S1,
       Country CO1, Station A2, City C2, State S2, Country CO2
WHERE  F.TimeFromKey = T.TimeKey AND T.Year = '2012' AND
       F.FromStationKey = A1.StationKey AND
       A1.CityKey = C1.CityKey AND
       C1.StateKey = S1.StateKey AND
       S1.CountryKey = CO1.CountryKey AND
```

```
CO1.CountryName = 'Belgium' AND
F.ToStationKey = A2.StationKey AND
A2.CityKey = C2.CityKey AND
C2.StateKey = S2.StateKey AND
S2.CountryKey = CO2.CountryKey AND
CO2.CountryName = 'Belgium'
```

(e) For each trip, list the average number of passengers per segment, that means, take all the segments of each trip, and average the number of passengers.

```
SELECT    T.TripNumber, AVG(NoPassengers)
FROM      Segments F, Trip T
WHERE     F.TimeFromKey = T.TripKey
GROUP BY T.TripNumber
```

5.5 Consider the university data warehouse described in Ex. 3.3. Draw a constellation schema for the data warehouse taking into account the different granularities of the time dimension.

Answer A constellation schema is shown in Fig. 5.3.

5.6 For the constellation schema obtained in the previous exercise, write in SQL the following queries.

(a) List by department the total number of teaching hours during the academic year 2012–2013.

```
SELECT    DepartmentName, SUM(NoHours)
FROM      Teaching T, AcademicSemester S, Department D
WHERE     T.SemesterKey = S.SemesterKey AND
          T.DepartmentKey = D.DepartmentKey
          AcademicYear = '2012-2013'
GROUP BY DepartmentName
```

(b) List by department the total amount of research projects during the calendar year 2012.

```
SELECT    DepartmentName, SUM(Amount)
FROM      Research R, Professor P, Department D, Time T
WHERE     R.ProfessorKey = P.ProfessorKey AND
          P.DepartmentKey = D.DepartmentKey AND
          R.StartDateKey = T.TimeKey AND Year = '2012'
GROUP BY DepartmentName
```

(c) List by department the total number of professors involved in research projects during the calendar year 2012.

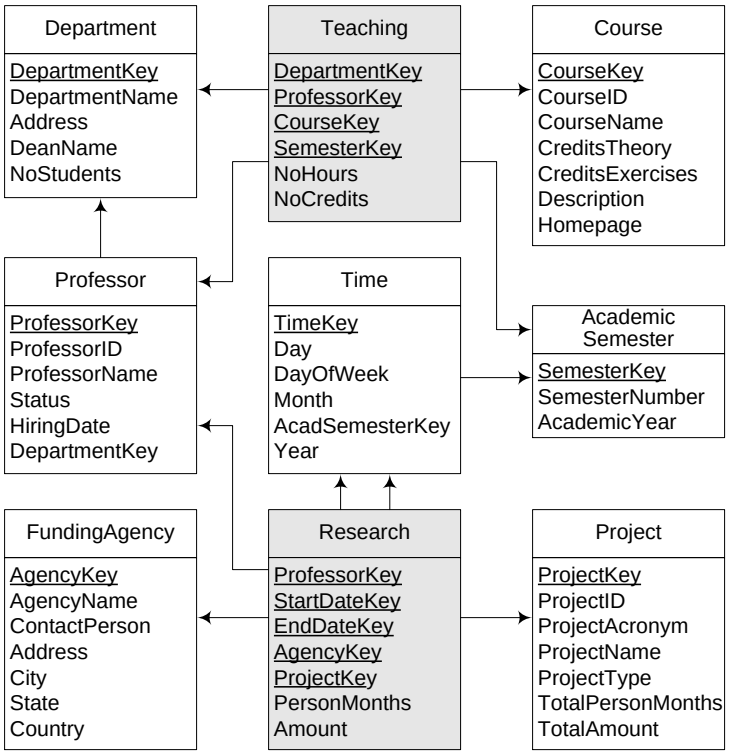

Fig. 5.3. A constellation schema for the data warehouse in Ex. 3.3

```
SELECT    DepartmentName, COUNT(ProfessorID)
FROM      Research R, Professor P, Department D, Time T
WHERE     R.ProfessorKey = P.ProfessorKey AND
          P.DepartmentKey = D.DepartmentKey AND
          R.StartDateKey = T.TimeKey AND Year = '2012'
GROUP BY DepartmentName
```

(d) List by department the total number of courses delivered during the academic year 2012–2013.

```
SELECT    DepartmentName, COUNT(CourseID)
FROM      Teaching T, AcademicSemester S, Department D
WHERE     T.SemesterKey = S.SemesterKey AND
          T.DepartmentKey = D.DepartmentKey
          AcademicYear = '2012-2013'
GROUP BY DepartmentName
```

(e) List by department and funding agency, the total number of projects started in 2012.

```
SELECT    DepartmentName, AgencyName, COUNT(ProfessorID)
FROM      Research R, Professor P, Department D,
          FundingAgency F, Time T
WHERE     R.ProfessorKey = P.ProfessorKey AND
          P.DepartmentKey = D.DepartmentKey AND
          R.StartDateKey = T.TimeKey AND Year = '2012' AND
          R.AgencyKey = F.AgencyKey
GROUP BY DepartmentName, AgencyName
```

5.7 Translate into the relational model the MultiDim schema given in Fig. 5.4.

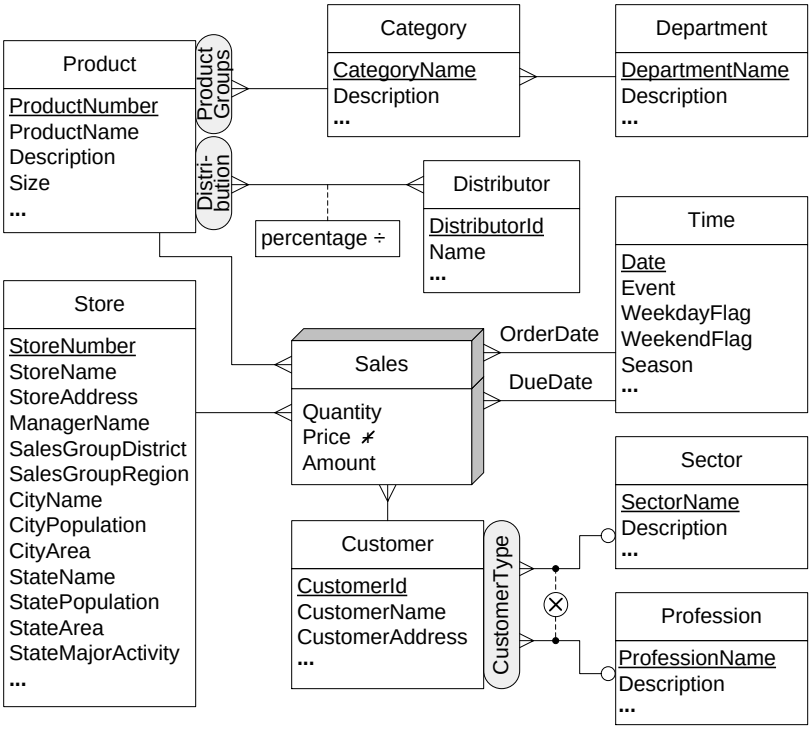

Fig. 5.4. A conceptual schema of a sales data warehouse

Answer The logical schema is given in Fig. 5.5.

5.8 Translate the MultiDim schema obtained for the French horse race application in Ex. 4.5 into the relational model.

Answer A constellation schema for this application is given in Fig. 5.6.

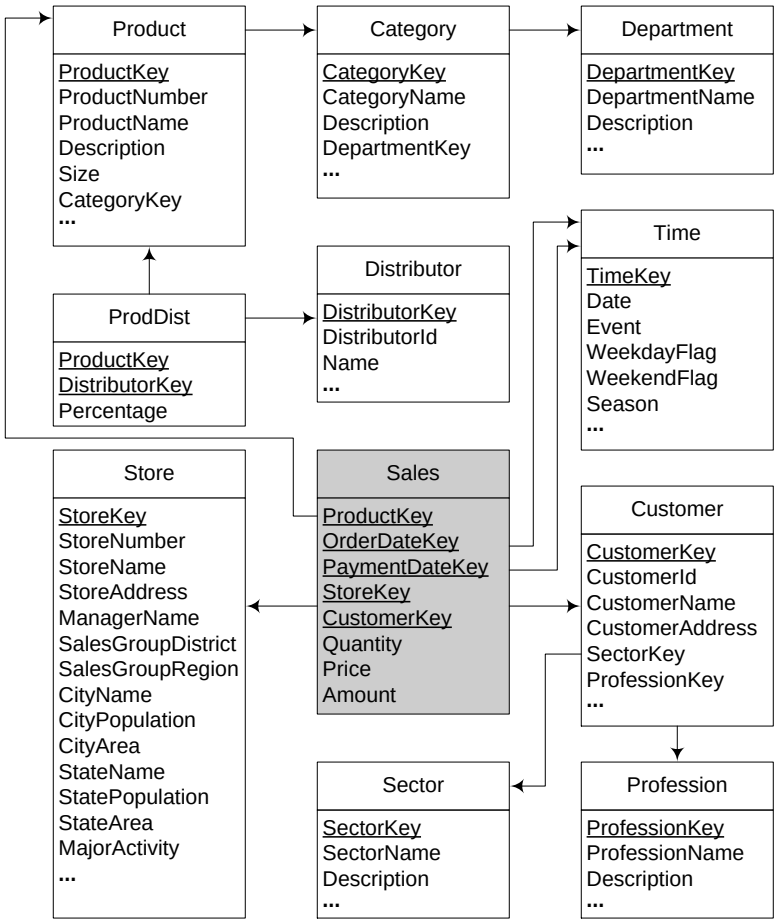

Fig. 5.5. A logical translation of the conceptual schema in Fig. 5.4

5.9 Translate the MultiDim schema obtained for the Formula One application in Ex. 4.7 into the relational model.

Answer A snowflake schema for this application is given in Fig. 5.7.

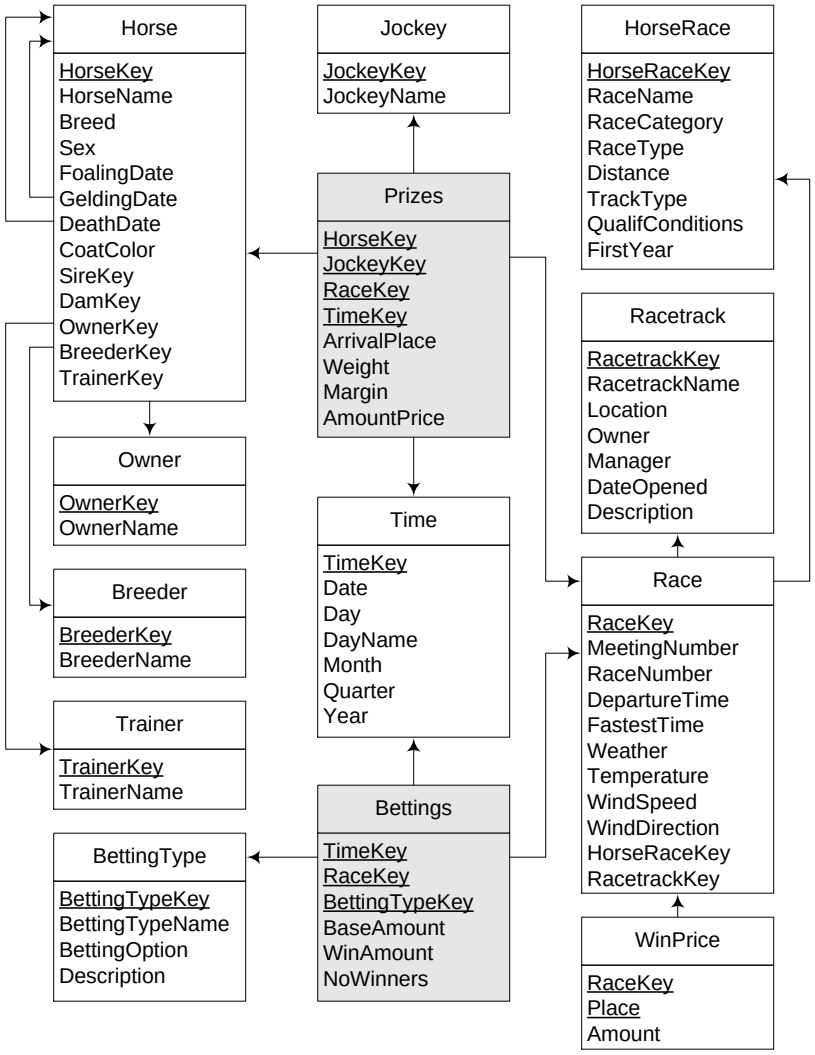

Fig. 5.6. Constellation schema of the French horse racing data warehouse in Ex. 4.5

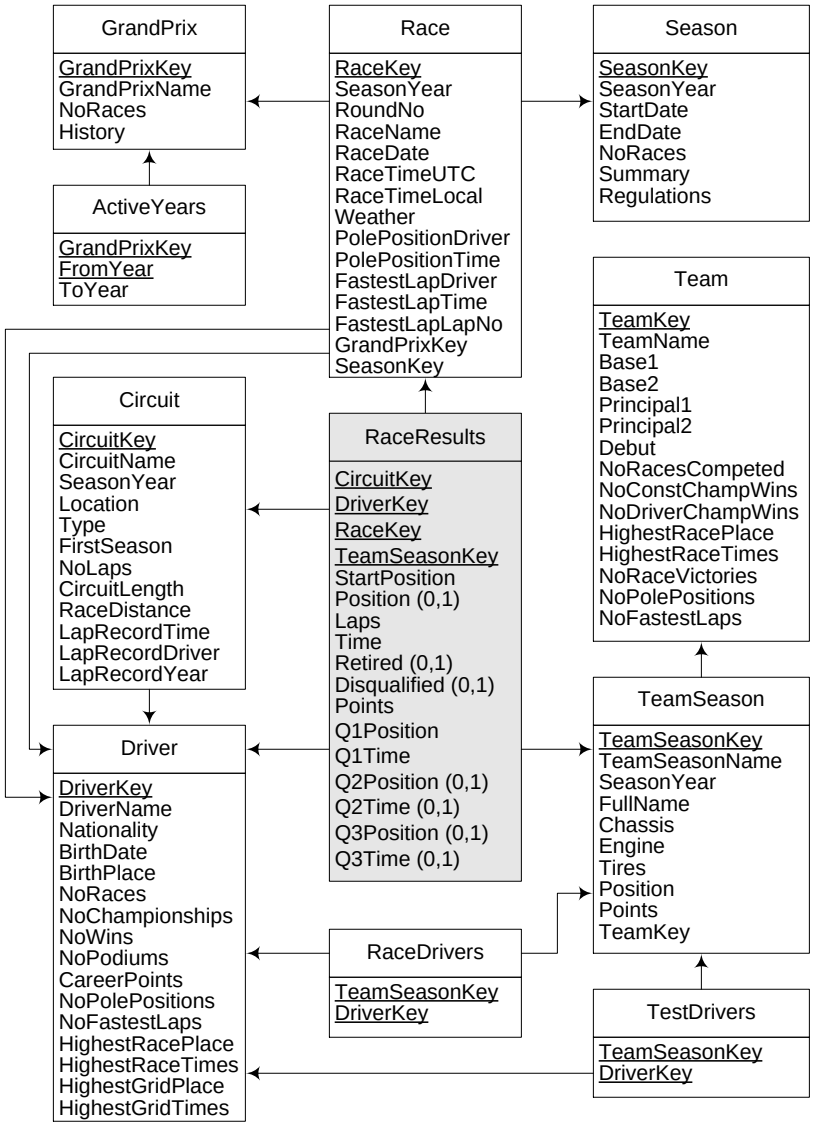

Fig. 5.7. Snowflake schema of the Formula One data warehouse in Ex. 4.7

5.10 The Research and Innovative Technology Administration (RITA)[1] coordinates the U.S. Department of Transportation's (DOT) research programs. It collects several statistics about many kinds of transportation means. The information is published at the following URL: http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=261

There is a set of tables T_T100I_Segment_All_Carrier_XXXX, one by year, ranging from 1990 up till now. These tables report statistics about flight segments between airports summarized by month. This information includes the scheduled and actually departured flights, the number of seats sold, the freight transported, and the distance traveled, among other ones. The schema and description of these tables is given in Table 5.1. A set of lookup tables given in Table 5.2 include information about airports, carriers, time, and other ones. The schemas of these lookup tables are composed of just two columns called Code and Description. The mentioned web site describes all tables in detail.

From the information above, construct an appropriate data warehouse schema. Analize the input data and motivate the choice of your schema.

Answer We have imported all the input CSV files in database tables using the import utility provided by SQL Server. Since the fact data is split into tables Fact_1990, Fact_1991, ..., Fact_2013, we created a view that performs the union of these tables as follows

```
CREATE VIEW Fact AS (
SELECT * FROM Fact_1990 UNION
SELECT * FROM Fact_1991 UNION
...
SELECT * FROM Fact_2013 )
```

In the tables Fact_XXXX the following attributes may be null:

Origin_Country, Dest_Country, Unique_Carrier, Airline_ID
Unique_Carrier_Name, Unique_Carrier_Entity, Carrier_Name,
Carrier_Group_New, Region

Records with either Origin_Country or Dest_Country null are obviously erroneous but this can be easily corrected since in both cases the Origin_Country_Name or Dest_Country_Name is not null and contains the values Berlin or Czechoslovakia. Since all other attributes that may be null refer to carriers, we study next data about carriers.

The attributes pertaining to carriers are the following

Unique_Carrier, Airline_ID, Unique_Carrier_Name, Unique_Carrier_Entity,
Region, Carrier, Carrier_Name, Carrier_Group, Carrier_Group_New

[1] http://www.transtats.bts.gov/

Summaries	
DepScheduled	Departures scheduled
DepPerformed	Departures performed
Payload	Available payload (pounds)
Seats	Available seats
Passengers	Non-stop segment passengers transported
Freight	Non-stop segment freight transported (pounds)
Mail	Non-stop segment mail transported (pounds)
Distance	Distance between airports (miles)
RampTime	Ramp to ramp time (minutes)
AirTime	Airborne time (minutes)
Carrier	
UniqueCarrier	Unique carrier code. When the same code has been used by multiple carriers, a numeric suffix is used for earlier users, for example, PA, PA(1), PA(2). Use this field for analysis across a range of years.
AirlineID	An identification number assigned by US DOT to identify a unique airline (carrier). A unique airline (carrier) is defined as one holding and reporting under the same DOT certificate regardless of its code, name, or holding company/corporation.
UniqueCarrierName	Unique carrier name. When the same name has been used by multiple carriers, a numeric suffix is used for earlier users, for example, Air Caribbean, Air Caribbean (1).
UniqCarrierEntity	Unique entity for a carrier's operation region.
CarrierRegion	Carrier's operation region. Carriers report data by operation region
Carrier	Code assigned by IATA and commonly used to identify a carrier. As the same code may have been assigned to different carriers over time, the code is not always unique. For analysis, use the unique carrier code.
CarrierName	Carrier name
CarrierGroup	Carrier group code. Used in legacy analysis
CarrierGroupNew	Carrier group new

Table 5.1. Attributes of the tables T_T100I_Segment_All_Carrier_XXXX

Origin	
OriginAirportID	Origin airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused.
OriginAirportSeqID	Origin airport, Airport Sequence ID. An identification number assigned by US DOT to identify a unique airport at a given point of time. Airport attributes, such as airport name or coordinates, may change over time.
OriginCityMarketID	Origin airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market.
Origin	Origin airport
OriginCityName	Origin city
OriginCountry	Origin airport, country
OriginCountryName	Origin airport, country name
OriginWAC	Origin airport, world area code
Destination	
DestAirportID	Destination airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused.
DestAirportSeqID	Destination airport, Airport Sequence ID. An identification number assigned by US DOT to identify a unique airport at a given point of time. Airport attributes, such as airport name or coordinates, may change over time.
DestCityMarketID	Destination airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market.
Dest	Destination airport
DestCityName	Destination city
DestCountry	Destination airport, country
DestCountryName	Destination airport, country name
DestWAC	Destination airport, world area code

Table 5.1. Attributes of the tables T_T100I_Segment_All_Carrier_XXXX (cont.)

Aircraft	
AircraftGroup	Aircraft group
AircraftType	Aircraft type
AircraftConfig	Aircraft configuration
Time Period	
Year	Year
Quarter	Quarter
Month	Month
Other	
DistanceGroup	Distance intervals, every 500 Miles, for flight segment
Class	Service Class

Table 5.1. Attributes of the tables T_T100I_Segment_All_Carrier_XXXX (cont.)

L_STRCRAFT_CONFIG	L_CITY_MARKET_ID
L_STRCRAFT_GROUP	L_COUNTRY_CODE
L_STRCRAFT_TYPE	L_DISTANCE_GROUP_500
L_STRLINE_ID	L_MONTHS
L_STRPORT	L_QUARTERS
L_STRPORT_ID	L_REGION
L_STRPORT_SEQ_ID	L_SERVICE_CLASS
L_CARRIER_GROUP	L_UNIQUE_CARRIER_ENTITIES
L_CARRIER_GROUP_NEW	L_UNIQUE_CARRIERS
L_CARRIER_HISTORY	L_WORLD_AREA_CODES

Table 5.2. Lookup tables for the table T_T100I_Segment_All_Carrier_XXXX

There are many records for which only the carrier name is known among all these attributes, such as follows

```
NULL NULL NULL NULL NULL BEQ NULL 0 NULL
NULL NULL NULL NULL NULL EG  NULL 0 NULL
NULL NULL NULL NULL NULL SU  NULL 0 NULL
```

For this reason we exluded those records. These are the only ones with Unique_Carrier as null.

Thus, data about carriers can be obtained from the query below.

```
SELECT DISTINCT Unique_Carrier, Airline_ID, Unique_Carrier_Name,
        Unique_Carrier_Entity, Region, Carrier,
        Carrier_Name, Carrier_Group, Carrier_Group_New
FROM    Fact
WHERE Unique_Carrier IS NOT NULL
ORDER BY Unique_Carrier
```

As stated in the documentation, carriers report data by operation region (such as Atlantic, Domestic, International, ...). Attribute Unique_Carrier_Entity is unique for a combination of carrier and region. Therefore, the following query allows to obtain all the information pertaining to carriers without the region information.

```
SELECT DISTINCT Unique_Carrier, Airline_ID, Unique_Carrier_Name,
        Carrier, Carrier_Name, Carrier_Group, Carrier_Group_New
FROM    Fact
WHERE Unique_Carrier IS NOT NULL
ORDER BY Unique_Carrier
```

There is a functional dependency Unique_Carrier $\rightarrow$ Airline_ID. Indeed, the following query returns an empty answer.

```
SELECT *
FROM    Fact F1
WHERE EXISTS (
        SELECT *
        FROM    Fact F2
        WHERE F1.Unique_Carrier = F2.Unique_Carrier AND
                F1.Airline_ID <> F2.Airline_ID )
```

Further, attribute Airline_ID can be removed since the lookup table L_Airline_ID does not add any other information that is not in the attributes Carrier and Carrier_Name.

Therefore, the information about carriers is obtained as follows

```
SELECT DISTINCT Unique_Carrier, Unique_Carrier_Name,
        Carrier, Carrier_Name, Carrier_Group, Carrier_Group_New
FROM    Fact
WHERE Unique_Carrier IS NOT NULL
ORDER BY Unique_Carrier
```

A carrier may have several carrier groups and carrier groups new as illustrated by the records below which are in the answer of the above query.

```
9E Endeavor Air Inc. 9E Pinnacle Airlines Inc. 1 6
9E Endeavor Air Inc. 9E Pinnacle Airlines Inc. 2 2
```

Therefore, this will induce nonstrict hierarchies between carriers and carrier groups. Finally, we have chosen to keep only the carrier group new since the documentation states that carrier group should only be used for legacy analysis.

Carrier information without carrier groups can be obtained by the following query.

```
SELECT DISTINCT Unique_Carrier, Unique_Carrier_Name,
        Carrier, Carrier_Name
FROM    Fact
WHERE Unique_Carrier IS NOT NULL
ORDER BY Unique_Carrier
```

Contrary to what it is said in the documentation, the attribute Unique_Carrier is not unique for carriers since the following query gives less answers that the previous query.

```
SELECT DISTINCT Unique_Carrier
FROM    Fact
WHERE Unique_Carrier IS NOT NULL
ORDER BY Unique_Carrier
```

What is unique is the combination of Unique_Carrier, Carrier, and Carrier_Name

```
SELECT DISTINCT Unique_Carrier, Carrier, Carrier_Name
FROM    Fact
WHERE Unique_Carrier IS NOT NULL
ORDER BY Unique_Carrier
```

since the query above has the same number of answers as the query asking for the four attributes before. For example, the following records belong to the answer of the last query.

```
ADB AUQ Antonov Company
ADB ADB Antonov Company
ADB AUQ Antonov Design Bureau
```

We can see that for the same unique carrier there are two values of carrier and two values of carrier names.

The next group of attributes of the fact table pertains to origin and destination airports. The information about all airports can be obtained by the following query

```
SELECT DISTINCT
    Origin_City_Market_ID AS City_Market_ID, Origin AS Airport_Code,
    Origin_City_Name AS City_Name, Origin_Country AS Airport_Code,
    Origin_Country_Name AS Country_Name, Origin_WAC AS WAC
FROM Fact
UNION
SELECT DISTINCT
    Dest_Airport_Seq_ID AS Airport_Seq_ID,
    Dest_City_Market_ID AS City_Market_ID, Dest AS AirportCode,
    Dest_City_Name AS City_Name, DEST_Country AS Country_Code,
    Dest_Country_Name AS Country_Name, Dest_WAC AS WAC
FROM FACT
```

The attribute Airport_Seq_ID is a key of the data obtained by the above query, as the number of distinct values of the attribute is the same as the number of answers of the above query.

Finally, the remaining attributes of the fact table pertain to dimensions such as AircraftGroup, AircraftType, AircraftConfig, etc., and the value of such dimensions is given in the lookup tables.

Therefore, the conceptual schema of the data warehouse is given in Fig. 5.8 and the logical schema is given in Fig. 5.9.

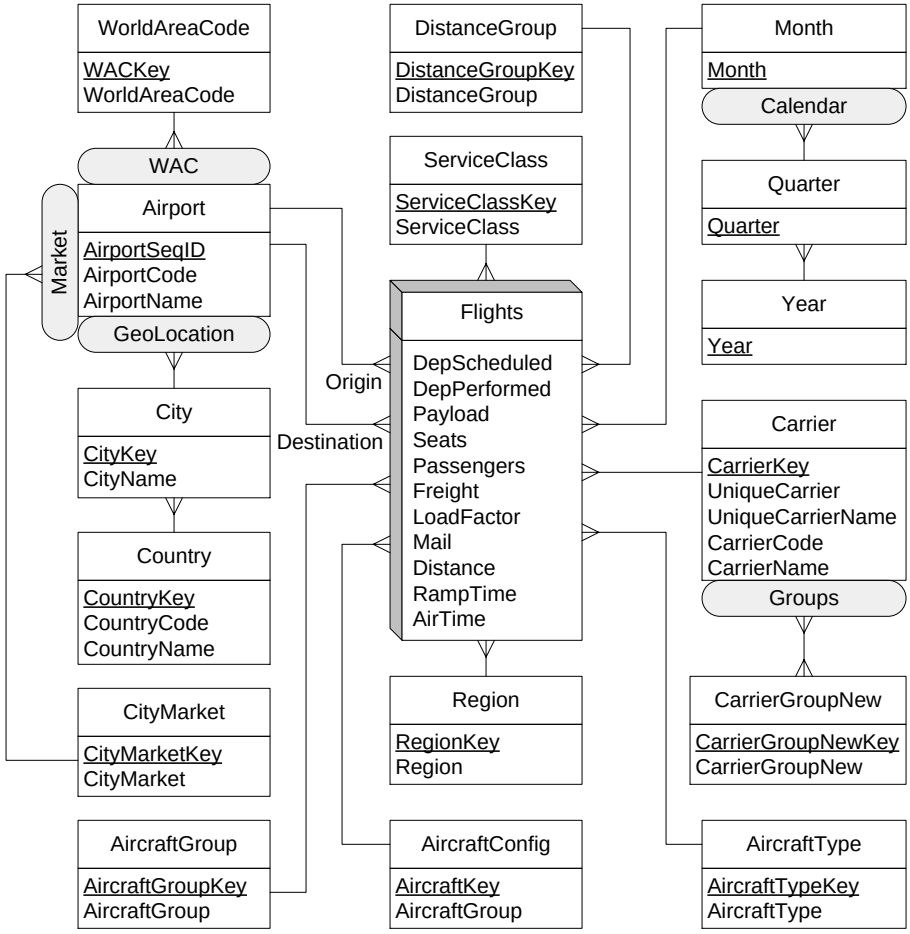

Fig. 5.8. Conceptual schema of the data warehouse for the air carrier example

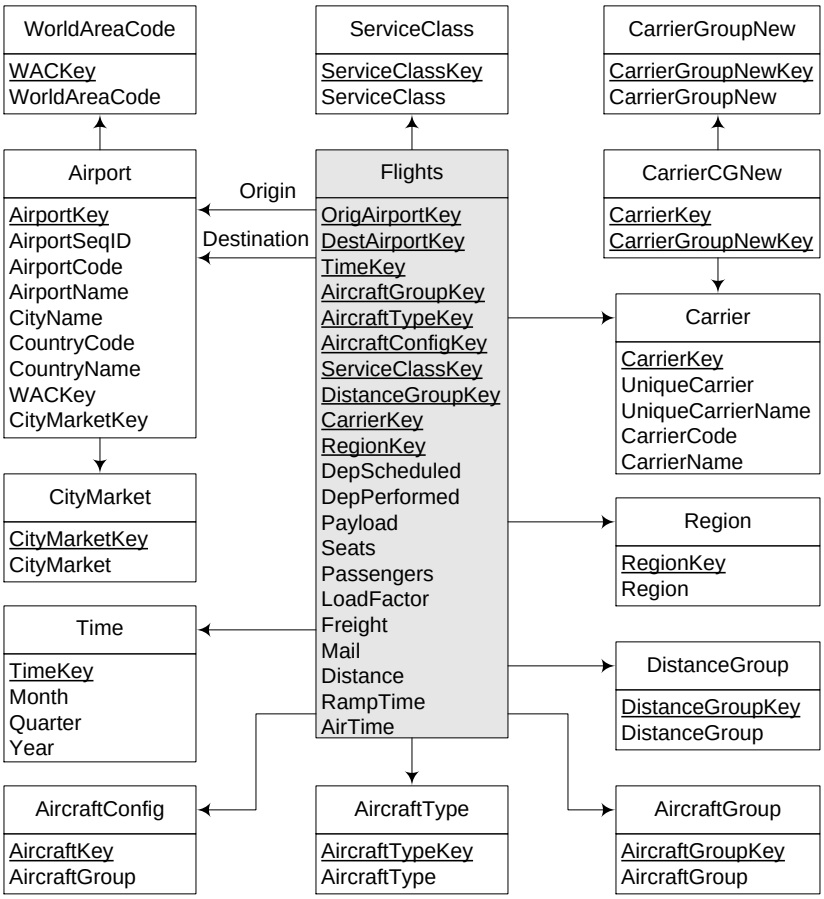

Fig. 5.9. Logical schema of the data warehouse for the air carrier example