Design of a Data Warehouse

Perna Mirco
Pizzuto Salvatore Leonardo

A.A. 2024/2025

Index

1 Introduction

The project involves designing a Data Warehouse for the analysis of healthcare-related information in the state of Massachusetts, United States of America, to meet the needs of decision-making processes. Each chapter describes the steps taken to design and develop the required Data Warehouse.

2 Creation and population of the schema

2.1 Analysis of the provided data

The data were provided in CSV format, which consists of simple text files containing lists of values. To enable a more accurate analysis, a preliminary database was created to reflect the original structure of the data. DBeaver, an SQL client, was used to facilitate the import of the files into the MySQL database. DBeaver offers a dedicated tool for importing data from CSV files. It is represented in Figure 1.

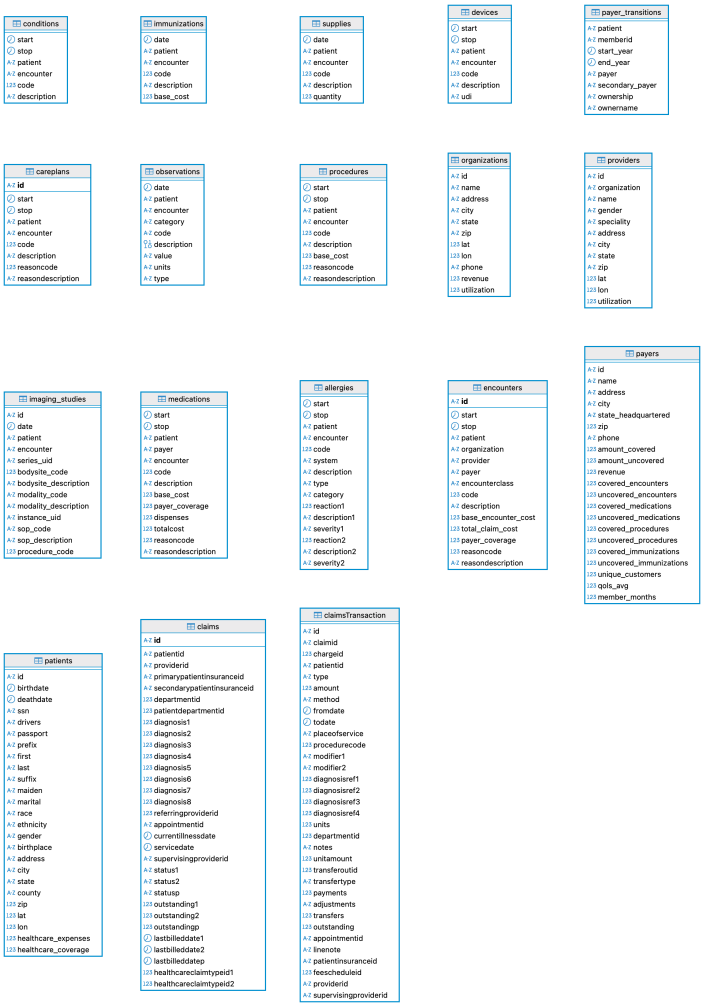

Figure 1: CSV files represented as tables.

However, the dates contained in the various CSV files do not match the format required by the DBeaver tool. To solve this issue, we decided to implement a Python program that converts the date format to the one required by the tool. The following is an example of the implemented code.

```
from datetime import datetime
import csv

with open('procedures.csv', newline='') as infile, open('dati_csv_cleaned/
    procedures_cleaned.csv', 'w', newline='') as outfile:
    reader = csv.DictReader(infile)
    fieldnames = reader.fieldnames
    writer = csv.DictWriter(outfile, fieldnames=fieldnames)
    writer.writeheader()

    for row in reader:
        if row['START']:
            dt = datetime.fromisoformat(row['START'].replace('Z', '+00:00'))
            row['START'] = dt.strftime('%Y-%m-%d %H:%M:%S')
        if row['STOP']:
            dt = datetime.fromisoformat(row['STOP'].replace('Z', '+00:00'))
            row['STOP'] = dt.strftime('%Y-%m-%d %H:%M:%S')
        writer.writerow(row)
```

Once the data was loaded into the database, a series of queries was executed to gain a deeper understanding of the information contained in the files, to identify primary keys and referential constraints among the data and to evaluate the data in terms of accuracy, completeness and consistency. The analysis showed that the majority of entities are related to the entity 'Encounter'. This will be more clearly observable following the normalization process, that contains all the information about a patient's visit to a specific hospital, including the reason for the visit and the doctor in charge. After completion of this initial analysis, it was decided to normalize the database to ensure data integrity, reduce redundancy, and improve the efficiency of the analytical operations required. The results obtained are shown in Figure 2.

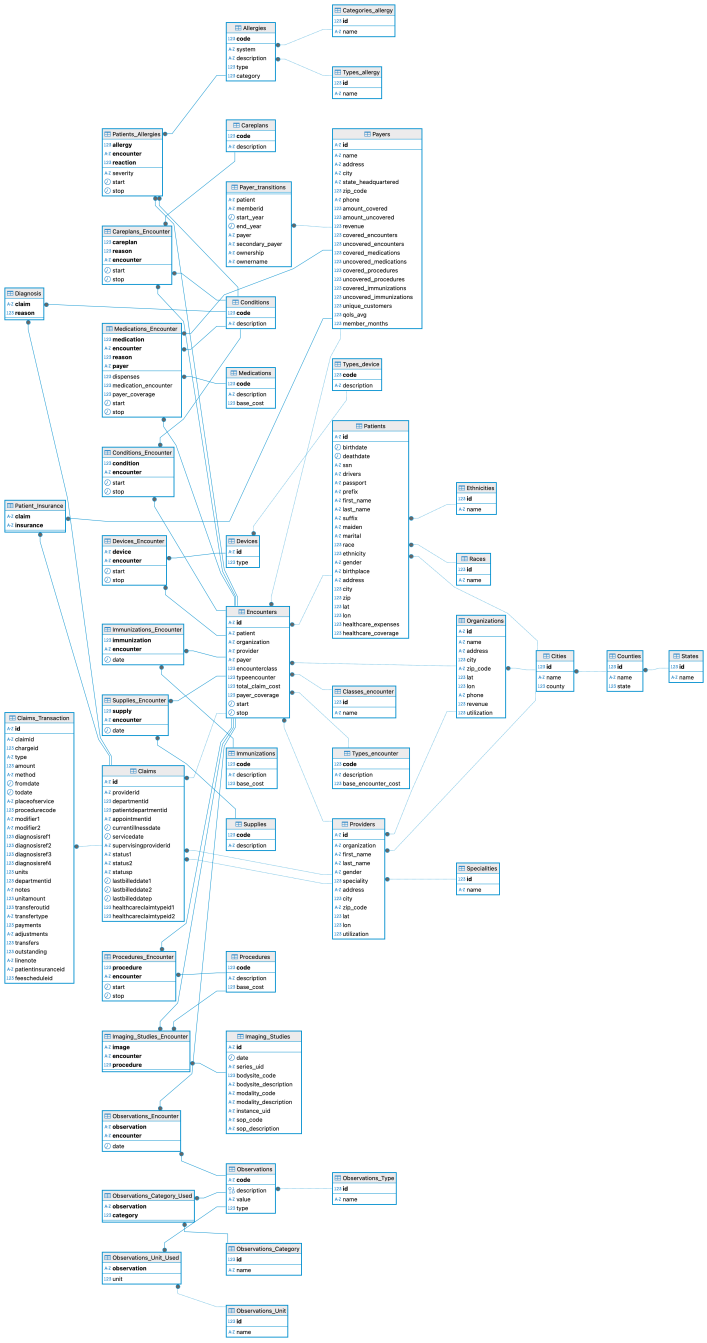

Figure 2: Normalized database.

To conclude this first step, the E-R diagram was created, which plays a significant role in the following steps, especially in the definition of facts. It is shown in Figure 3. Note that the attributes are not shown for readability reasons, but they can be read from the normalized database shown in Figure 2.

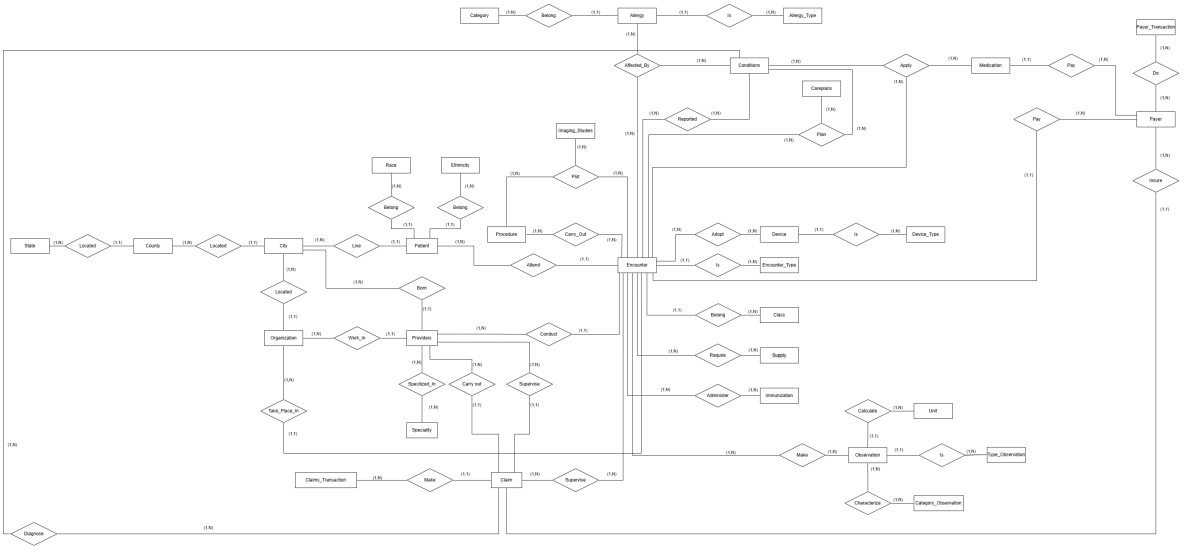

Figure 3: E/R diagram

2.2 Queries

1. What is the number of visits made to each hospital facility in the last five years, broken down by gender and age group?
2. What are the most used supplies each year, broken down by county and hospital facility?
3. What is the number of respiratory therapies recorded in various hospitals, broken down by gender and age group?
4. What is the average age of patients affected by acute bronchitis or prediabetes in each county recorded in the last five years?
5. What is the average number of treatment days required to complete a therapy plan for various diseases in each hospital over the years?
6. What is the cost of medical procedures broken down by year for the various counties and hospital facilities?
7. What are the most common allergies, grouped by patient age group, and county where they were recorded?
8. What is the incidence of the various side effects associated with each allergy, broken down by patient age group?
9. How many individuals, in the age group between 10 and 50 years, were vaccinated against HPV between the years 2015 and 2019, in the counties of Bristol, Essex, and Middlesex, grouped by sex, county, and year of administration?
10. What is the number of visits performed by each doctor, broken down by patient age group, gender, race, and, in addition, county and year of the conducted visit?

3 Cleaning Data and ETL Process

In order to use the previously discussed normalized database, an ETL service has been implemented, i.e. a process of data extraction, transformation, and loading. During this process, data cleaning was also performed, which involved handling null values and inconsistent data.
Below are some stages of the ETL process, implemented using the MySQL dialect.

3.1 Patients

During the processing of the 'Patient' entity, an issue was encountered in handling null values in the fields 'drivers', 'passport', 'prefix','suffix','maiden' and 'marital'. To address this issue, it was decided to replace the null values with standard default values, as shown in the table below.

Value	**Fields**
-	• marital
NO	• prefix

UNKNOWN	• drivers • maiden • passport • suffix

Table 1: Values used to replace null values in the fields.

From the 'Patients' table of the non-normalized database, we also extracted information related to the 'States', 'Counties', 'Cities', 'Races', and 'Ethnicities' tables. To avoid repeated queries to the non-normalized database, it was decided to use the temporary table 'patients_not_cleaned'.

```
CREATE TEMPORARY TABLE patients_not_cleaned AS
SELECT * FROM BigData.patients;

INSERT INTO BigData_CleanedData.States (Name)
SELECT State
FROM patients_not_cleaned
group by State;

INSERT INTO BigData_CleanedData.Counties (Name, State)
SELECT REPLACE(p.County, ' County', ''), s.id
FROM patients_not_cleaned p
INNER JOIN BigData_CleanedData.States s
        ON s.Name = p.State
group by s.id, p.county;

INSERT INTO BigData_CleanedData.Cities (Name, County)
SELECT p.city,c.id
FROM patients_not_cleaned p
INNER JOIN BigData_CleanedData.States s
        ON s.Name = p.State
INNER JOIN BigData_CleanedData.Counties c
        ON  c.Name = REPLACE(p.County, ' County', '')
group by c.id, p.city;

INSERT INTO BigData_CleanedData.Races (Name)
SELECT p.race
FROM patients_not_cleaned p
GROUP BY p.race;

INSERT INTO BigData_CleanedData.Ethnicities (Name)
SELECT p.ethnicity
FROM patients_not_cleaned p
GROUP BY p.ethnicity;

INSERT INTO BigData_CleanedData.Patients
(id, birthdate, deathdate, ssn, drivers, passport, prefix, first_name,
    last_name, suffix, maiden, marital, race, ethnicity, gender, birthplace,
    address, city, zip, lat, lon, healthcare_expenses, healthcare_coverage)
SELECT p.id,
       birthdate,
       deathdate,
       ssn,
       IFNULL(NULLIF(drivers, ''), 'UNKNOWN') as drivers,
       IFNULL(NULLIF(passport, ''), 'UNKNOWN') as passport,
       IFNULL(NULLIF(prefix, ''), 'NO') as prefix, `first`, `last`,
```

```
        IFNULL(NULLIF(suffix, ''), 'UNKNOWN') as suffix,
        IFNULL(NULLIF(maiden, ''), 'UNKNOWN') as maiden,
        IFNULL(NULLIF(marital, ''), '-') as marital,
        r.id as race,
        e.id as ethnicity,
        gender,
        birthplace,
        address,
        ci.id as city,
        zip,
        lat,
        lon,
        healthcare_expenses,
        healthcare_coverage
FROM patients_not_cleaned p
INNER JOIN BigData_CleanedData.States s
        ON s.Name = p.State
INNER JOIN BigData_CleanedData.Counties c
        ON c.Name = REPLACE(p.County, ' County', '') and c.state = s.id
INNER JOIN BigData_CleanedData.Cities ci
        on ci.Name = p.city  and ci.county = c.id
INNER JOIN BigData_CleanedData.Races r
        ON r.Name = p.race
INNER JOIN BigData_CleanedData.Ethnicities e
        ON e.Name = p.Ethnicity;
```

3.2 Organizations

During the processing of the 'Organizations' entity, inconsistencies were found in the 'city' field. Many entries were written in uppercase, some referred to neighborhoods of specific cities, and others contained abbreviations. To address these issues, we performed the following steps:

1. Created temporary table 'organizations_not_cleaned'
2. Created the table 'temp_massachusetts_cities' to map cities to counties.
3. Filtering of cities and counties already inserted during the 'Patients' processing.
4. Insertion of counties, cities, and organizations.

```
CREATE TEMPORARY TABLE organizations_not_cleaned AS
SELECT * FROM BigData.Organizations ;

CREATE TEMPORARY TABLE temp_massachusetts_cities (
    city VARCHAR(100),
    county VARCHAR(100)
);

INSERT INTO temp_massachusetts_cities (city, county) VALUES
  ('Alford', 'Berkshire'),
  ('Aquinnah', 'Dukes'),
  ('Abington', 'Plymouth'),
  ('Acton', 'Middlesex'),
  ('Acushnet', 'Bristol'),
  ('Adams', 'Berkshire'),
  ('Agawam', 'Hampden'),
  ('Amesbury', 'Essex'),
  ('Amherst', 'Hampshire'),
  ('Andover', 'Essex'),
  ('Arlington', 'Middlesex'),
  ('Ashburnham', 'Worcester'),
```

```
    ('Ashby', 'Middlesex'),
    ('Ashfield', 'Franklin'),
    ('Ashland', 'Middlesex'),
    ('Athol', 'Worcester'),
    ('Attleboro', 'Bristol'),
    ('Auburn', 'Worcester'),
    ('Avon', 'Norfolk'),
    ('Ayer', 'Middlesex'),
    ('Barnstable', 'Barnstable'),
    ('Barre', 'Worcester'),
    ('Becket', 'Berkshire'),
    ('Bedford', 'Middlesex'),
    ('Belchertown', 'Hampshire'),
    ('Bellingham', 'Norfolk'),
    ('Belmont', 'Middlesex'),
    ('Berkley', 'Bristol'),
    ('Berlin', 'Worcester'),
    ('Bernardston', 'Franklin'),
    ('Beverly', 'Essex'),
    ('Billerica', 'Middlesex'),
    ('Blackstone', 'Worcester'),
    ('Blandford', 'Hampden'),
    ('Bolton', 'Worcester'),
    ('Boston', 'Suffolk'),
    ('Boxborough', 'Middlesex'),
    ('Boxford', 'Essex'),
    ('Boylston', 'Worcester'),
    ('Braintree', 'Norfolk'),
    ('Brewster', 'Barnstable'),
    ('Bridgewater', 'Plymouth'),
    ('Brimfield', 'Hampden'),
    ('Brockton', 'Plymouth'),
    ('Brookfield', 'Worcester'),
    ('Brookline', 'Norfolk'),
    ('Buckland', 'Franklin'),
    ('Burlington', 'Middlesex'),
    ('Cambridge', 'Middlesex'),
    ('Canton', 'Norfolk'),
    ('Carlisle', 'Middlesex'),
    ('Carver', 'Plymouth'),
    ('Charlemont', 'Franklin'),
    ('Charlton', 'Worcester'),
    ('Chatham', 'Barnstable'),
    ('Chelmsford', 'Middlesex'),
    ('Chelsea', 'Suffolk'),
    ('Cheshire', 'Berkshire'),
    ('Chester', 'Hampden'),
    ('Chesterfield', 'Hampshire'),
    ('Chicopee', 'Hampden'),
    ('Chilmark', 'Dukes'),
    ('Clarksburg', 'Berkshire'),
    ('Clinton', 'Worcester'),
    ('Cohasset', 'Norfolk'),
    ('Colrain', 'Franklin'),
    ('Concord', 'Middlesex'),
    ('Conway', 'Franklin'),
    ('Cummington', 'Hampshire'),
    ('Dalton', 'Berkshire'),
    ('Danvers', 'Essex'),
    ('Dartmouth', 'Bristol'),
    ('Dedham', 'Norfolk'),
    ('Deerfield', 'Franklin'),
```

```
    ('Dennis', 'Barnstable'),
    ('Dighton', 'Bristol'),
    ('Douglas', 'Worcester'),
    ('Dover', 'Norfolk'),
    ('Dracut', 'Middlesex'),
    ('Dudley', 'Worcester'),
    ('Dunstable', 'Middlesex'),
    ('Duxbury', 'Plymouth'),
    ('East Bridgewater', 'Plymouth'),
    ('East Brookfield', 'Worcester'),
    ('East Longmeadow', 'Hampden'),
    ('Eastham', 'Barnstable'),
    ('Easthampton', 'Hampshire'),
    ('Easton', 'Bristol'),
    ('Edgartown', 'Dukes'),
    ('Egremont', 'Berkshire'),
    ('Erving', 'Franklin'),
    ('Essex', 'Essex'),
    ('Everett', 'Middlesex'),
    ('Fairhaven', 'Bristol'),
    ('Fall River', 'Bristol'),
    ('Falmouth', 'Barnstable'),
    ('Fitchburg', 'Worcester'),
    ('Florida', 'Berkshire'),
    ('Foxborough', 'Norfolk'),
    ('Framingham', 'Middlesex'),
    ('Franklin', 'Norfolk'),
    ('Freetown', 'Bristol'),
    ('Gardner', 'Worcester'),
    ('Georgetown', 'Essex'),
    ('Gill', 'Franklin'),
    ('Gloucester', 'Essex'),
    ('Goshen', 'Hampshire'),
    ('Gosnold', 'Dukes'),
    ('Grafton', 'Worcester'),
    ('Granby', 'Hampshire'),
    ('Granville', 'Hampden'),
    ('Great Barrington', 'Berkshire'),
    ('Greenfield', 'Franklin'),
    ('Groton', 'Middlesex'),
    ('Groveland', 'Essex'),
    ('Hadley', 'Hampshire'),
    ('Halifax', 'Plymouth'),
    ('Hamilton', 'Essex'),
    ('Hampden', 'Hampden'),
    ('Hancock', 'Berkshire'),
    ('Hanover', 'Plymouth'),
    ('Hanson', 'Plymouth'),
    ('Hardwick', 'Worcester'),
    ('Harvard', 'Worcester'),
    ('Harwich', 'Barnstable'),
    ('Hatfield', 'Hampshire'),
    ('Haverhill', 'Essex'),
    ('Hawley', 'Franklin'),
    ('Heath', 'Franklin'),
    ('Hingham', 'Plymouth'),
    ('Hinsdale', 'Berkshire'),
    ('Holbrook', 'Norfolk'),
    ('Holden', 'Worcester'),
    ('Holland', 'Hampden'),
    ('Holliston', 'Middlesex'),
    ('Holyoke', 'Hampden'),
```

```
    ('Hopedale', 'Worcester'),
    ('Hopkinton', 'Middlesex'),
    ('Hubbardston', 'Worcester'),
    ('Hudson', 'Middlesex'),
    ('Hull', 'Plymouth'),
    ('Huntington', 'Hampshire'),
    ('Ipswich', 'Essex'),
    ('Indian Orchard','Hampden'),
    ('Kingston', 'Plymouth'),
    ('Lakeville', 'Plymouth'),
    ('Lancaster', 'Worcester'),
    ('Lanesborough', 'Berkshire'),
    ('Lawrence', 'Essex'),
    ('Lee', 'Berkshire'),
    ('Leicester', 'Worcester'),
    ('Lenox', 'Berkshire'),
    ('Leominster', 'Worcester'),
    ('Leverett', 'Franklin'),
    ('Lexington', 'Middlesex'),
    ('Leyden', 'Franklin'),
    ('Lincoln', 'Middlesex'),
    ('Littleton', 'Middlesex'),
    ('Longmeadow', 'Hampden'),
    ('Lowell', 'Middlesex'),
    ('Ludlow', 'Hampden'),
    ('Lunenburg', 'Worcester'),
    ('Lynn', 'Essex'),
    ('Lynnfield', 'Essex'),
    ('Malden', 'Middlesex'),
    ('Manchester-by-the-Sea', 'Essex'),
    ('Mansfield', 'Bristol'),
    ('Marblehead', 'Essex'),
    ('Marion', 'Plymouth'),
    ('Marlborough', 'Middlesex'),
    ('Marshfield', 'Plymouth'),
    ('Mashpee', 'Barnstable'),
    ('Mattapoisett', 'Plymouth'),
    ('Maynard', 'Middlesex'),
    ('Medfield', 'Norfolk'),
    ('Medford', 'Middlesex'),
    ('Medway', 'Norfolk'),
    ('Melrose', 'Middlesex'),
    ('Mendon', 'Worcester'),
    ('Merrimac', 'Essex'),
    ('Methuen', 'Essex'),
    ('Middleborough', 'Plymouth'),
    ('Middlefield', 'Hampshire'),
    ('Middleton', 'Essex'),
    ('Milford', 'Worcester'),
    ('Millbury', 'Worcester'),
    ('Millis', 'Norfolk'),
    ('Millville', 'Worcester'),
    ('Milton', 'Norfolk'),
    ('Monroe', 'Franklin'),
    ('Monson', 'Hampden'),
    ('Montague', 'Franklin'),
    ('Monterey', 'Berkshire'),
    ('Montgomery', 'Hampden'),
    ('Mount Washington', 'Berkshire'),
    ('Nahant', 'Essex'),
    ('Nantucket', 'Nantucket'),
    ('Natick', 'Middlesex'),
```

```
    ('Needham', 'Norfolk'),
    ('New Ashford', 'Berkshire'),
    ('New Bedford', 'Bristol'),
    ('New Braintree', 'Worcester'),
    ('New Marlborough', 'Berkshire'),
    ('New Salem', 'Franklin'),
    ('Newbury', 'Essex'),
    ('Newburyport', 'Essex'),
    ('Newton', 'Middlesex'),
    ('Norfolk', 'Norfolk'),
    ('North Adams', 'Berkshire'),
    ('Northampton', 'Hampshire'),
    ('North Andover', 'Essex'),
    ('North Attleborough', 'Bristol'),
    ('Northborough', 'Worcester'),
    ('Northbridge', 'Worcester'),
    ('North Brookfield', 'Worcester'),
    ('Northfield', 'Franklin'),
    ('Norton', 'Bristol'),
    ('Norwell', 'Plymouth'),
    ('Norwood', 'Norfolk'),
    ('Oak Bluffs', 'Dukes'),
    ('Oakham', 'Worcester'),
    ('Orange', 'Franklin'),
    ('Orleans', 'Barnstable'),
    ('Otis', 'Berkshire'),
    ('Oxford', 'Worcester'),
    ('Palmer', 'Hampden'),
    ('Paxton', 'Worcester'),
    ('Peabody', 'Essex'),
    ('Pelham', 'Hampshire'),
    ('Pembroke', 'Plymouth'),
    ('Pepperell', 'Middlesex'),
    ('Peru', 'Berkshire'),
    ('Petersham', 'Worcester'),
    ('Phillipston', 'Worcester'),
    ('Pittsfield', 'Berkshire'),
    ('Plainfield', 'Hampshire'),
    ('Plainville', 'Norfolk'),
    ('Plymouth', 'Plymouth'),
    ('Plympton', 'Plymouth'),
    ('Princeton', 'Worcester'),
    ('Provincetown', 'Barnstable'),
    ('Quincy', 'Norfolk'),
    ('Randolph', 'Norfolk'),
    ('Raynham', 'Bristol'),
    ('Reading', 'Middlesex'),
    ('Rehoboth', 'Bristol'),
    ('Revere', 'Suffolk'),
    ('Richmond', 'Berkshire'),
    ('Rochester', 'Plymouth'),
    ('Rockland', 'Plymouth'),
    ('Rockport', 'Essex'),
    ('Rowe', 'Franklin'),
    ('Rowley', 'Essex'),
    ('Royalston', 'Worcester'),
    ('Russell', 'Hampden'),
    ('Rutland', 'Worcester'),
    ('Salem', 'Essex'),
    ('Salisbury', 'Essex'),
    ('Sandisfield', 'Berkshire'),
    ('Sandwich', 'Barnstable'),
```

```
    ('Saugus', 'Essex'),
    ('Savoy', 'Berkshire'),
    ('Scituate', 'Plymouth'),
    ('Seekonk', 'Bristol'),
    ('Sharon', 'Norfolk'),
    ('Sheffield', 'Berkshire'),
    ('Shelburne', 'Franklin'),
    ('Sherborn', 'Middlesex'),
    ('Shirley', 'Middlesex'),
    ('Shrewsbury', 'Worcester'),
    ('Shutesbury', 'Franklin'),
    ('Somerset', 'Bristol'),
    ('Somerville', 'Middlesex'),
    ('Southampton', 'Hampshire'),
    ('Southborough', 'Worcester'),
    ('Southbridge', 'Worcester'),
    ('South Hadley', 'Hampshire'),
    ('Southwick', 'Hampden'),
    ('Spencer', 'Worcester'),
    ('Springfield', 'Hampden'),
    ('Sterling', 'Worcester'),
    ('Stockbridge', 'Berkshire'),
    ('Stoneham', 'Middlesex'),
    ('Stoughton', 'Norfolk'),
    ('Stow', 'Middlesex'),
    ('Sturbridge', 'Worcester'),
    ('Sudbury', 'Middlesex'),
    ('Sunderland', 'Franklin'),
    ('Sutton', 'Worcester'),
    ('Swampscott', 'Essex'),
    ('Swansea', 'Bristol'),
    ('Taunton', 'Bristol'),
    ('Templeton', 'Worcester'),
    ('Tewksbury', 'Middlesex'),
    ('Tisbury', 'Dukes'),
    ('Tolland', 'Hampden'),
    ('Topsfield', 'Essex'),
    ('Townsend', 'Middlesex'),
    ('Truro', 'Barnstable'),
    ('Tyngsborough', 'Middlesex'),
    ('Tyringham', 'Berkshire'),
    ('Upton', 'Worcester'),
    ('Uxbridge', 'Worcester'),
    ('Wakefield', 'Middlesex'),
    ('Wales', 'Hampden'),
    ('Walpole', 'Norfolk'),
    ('Waltham', 'Middlesex'),
    ('Ware', 'Hampshire'),
    ('Wareham', 'Plymouth'),
    ('Warren', 'Worcester'),
    ('Warwick', 'Franklin'),
    ('Washington', 'Berkshire'),
    ('Watertown', 'Middlesex'),
    ('Wayland', 'Middlesex'),
    ('Webster', 'Worcester'),
    ('Wellesley', 'Norfolk'),
    ('Wellfleet', 'Barnstable'),
    ('Wendell', 'Franklin'),
    ('Wenham', 'Essex'),
    ('Westborough', 'Worcester'),
    ('West Boylston', 'Worcester'),
    ('West Bridgewater', 'Plymouth'),
```

```
  ('West Brookfield', 'Worcester'),
  ('Westfield', 'Hampden'),
  ('Westford', 'Middlesex'),
  ('Westhampton', 'Hampshire'),
  ('Westminster', 'Worcester'),
  ('West Newbury', 'Essex'),
  ('Westport', 'Bristol'),
  ('West Springfield', 'Hampden'),
  ('West Stockbridge', 'Berkshire'),
  ('West Tisbury', 'Dukes'),
  ('Westwood', 'Norfolk'),
  ('Weymouth', 'Norfolk'),
  ('Whately', 'Franklin'),
  ('Whitman', 'Plymouth'),
  ('Wilbraham', 'Hampden'),
  ('Williamsburg', 'Hampshire'),
  ('Williamstown', 'Berkshire'),
  ('Wilmington', 'Middlesex'),
  ('Winchendon', 'Worcester'),
  ('Winchester', 'Middlesex'),
  ('Windsor', 'Berkshire'),
  ('Winthrop', 'Suffolk'),
  ('Woburn', 'Middlesex'),
  ('Worcester', 'Worcester'),
  ('Worthington', 'Hampshire'),
  ('Wrentham', 'Norfolk'),
  ('Yarmouth', 'Barnstable'),
  ('Shelburne Falls','Franklin'),
  ('North Falmouth','Barnstable'),
  ('East Wareham','Plymouth'),
  ('South Attleboro','Bristol'),
  ('West Wareham','Plymouth');

CREATE TEMPORARY TABLE temp_city_abbreviations (
    short_name VARCHAR(100),
    full_name VARCHAR(100)
);

INSERT INTO temp_city_abbreviations (short_name, full_name) VALUES
  ('W CONCORD', 'Concord'),
  ('INDIAN ORCHARD', 'Springfield'),
  ('MARSTONS MILLS', 'Barnstable'),
  ('NORTH DARTMOUTH', 'Dartmouth'),
  ('SOUTH DARTMOUTH', 'Dartmouth'),
  ('EAST BOSTON', 'Boston'),
  ('FEEDING HILLS', 'Agawam'),
  ('JAMAICA PLAIN', 'Boston'),
  ('SOUTH WEYMOUTH', 'Weymouth'),
  ('BRIGHTON', 'Boston'),
  ('E SANDWICH', 'Sandwich'),
  ('EAST WAREHAM', 'Wareham'),
  ('DORCHESTER', 'Boston'),
  ('SOUTH ATTLEBORO', 'Attleboro'),
  ('MIDDLEBORO', 'Middleborough'),
  ('WABAN', 'Newton'),
  ('HYANNIS', 'Barnstable'),
  ('BRADFORD', 'Haverhill'),
  ('WELLESLEY HILLS', 'Wellesley'),
  ('FOXBORO', 'Foxborough'),
  ('POCASSET', 'Bourne'),
  ('WESTBORO', 'Westborough'),
  ('SAGAMORE BEACH', 'Bourne'),
```

```
  ('SHELBURNE FLS', 'Shelburne Falls'),
  ('ROSLINDALE', 'Boston'),
  ('WEST WAREHAM', 'Wareham'),
  ('NO FALMOUTH', 'Falmouth'),
  ('W SPRINGFIELD', 'Springfield'),
  ('N ADAMS', 'Adams'),
  ('S ATTLEBORO', 'Attleboro'),
  ('FLORENCE', 'Northampton'),
  ('TEATICKET', 'Falmouth'),
  ('PITTSFILED', 'Pittsfield'),
  ('CEDARVILLE', 'Plymouth'),
  ('WEST ROXBURY', 'Boston'),
  ('FISKDALE', 'Sturbridge'),
  ('CENTERVILLE', 'Barnstable'),
  ('BALDWINVILLE', 'Templeton'),
  ('HARWICH PORT', 'Harwich'),
  ('Leeds', 'Northampton'),
  ('NORTH ATTLEBORO', 'North Attleborough'),
  ('S YARMOUTH', 'South Yarmouth'),
  ('SOUTHBORO', 'Southborough'),
  ('HYDE PARK', 'Boston'),
  ('DENNIS PORT', 'Dennis'),
  ('EAST HARWICH', 'Harwich'),
  ('MARLBORO', 'Marlborough'),
  ('W WAREHAM', 'Wareham'),
  ('E WAREHAM', 'Wareham'),
  ('SHELBURNE FALLS', 'Shelburne Falls');

CREATE TEMPORARY TABLE cities_to_insert AS
        SELECT CONCAT(UCASE(SUBSTRING(n.city, 1, 1)), LOWER(SUBSTRING(n.city,
            2))) AS city
        FROM(
        SELECT CASE
                WHEN t.short_name is null  THEN o.city
                ELSE t.full_name
            END as city
        FROM organizations_not_cleaned o
        LEFT JOIN temp_city_abbreviations t
        ON o.city = t.short_name
        ) as n
        where CONCAT(UCASE(SUBSTRING(n.city, 1, 1)), LOWER(SUBSTRING(n.city,
            2))) not in
                        (
                        SELECT c.name FROM BigData_CleanedData.Cities c
                        );



CREATE TEMPORARY TABLE cities_and_county_to_insert AS
SELECT c.city, mc.county
FROM cities_to_insert c
INNER JOIN temp_massachusetts_cities mc
        on mc.city = c.city
group by c.city, mc.county


INSERT INTO BigData_CleanedData.Counties (Name, state)
SELECT t.county , 1
FROM cities_and_county_to_insert t
LEFT JOIN  BigData_CleanedData.Counties c
        ON t.county = c.name
WHERE c.name is null
```

```
GROUP  BY t.county;


INSERT INTO BigData_CleanedData.Cities (Name, county)
SELECT t.city , c.id
FROM cities_and_county_to_insert t
INNER JOIN  BigData_CleanedData.Counties c
        ON t.county = c.name
LEFT JOIN BigData_CleanedData.Cities ci
        ON t.city = ci.name
where ci.name is null
GROUP  BY t.city,c.id;

INSERT INTO BigData_CleanedData.Organizations
(id, name, address, city, zip_code, lat, lon, phone, revenue, utilization)
SELECT o.id,
       o.name,
       address,
       c.id as city,
       zip,
       lat,
       lon,
       phone,
       revenue,
       utilization
FROM BigData.organizations o
LEFT JOIN temp_city_abbreviations t
        ON  o.city = t.short_name
INNER JOIN BigData_CleanedData.Cities c
        ON c.Name =  o.city or  c.Name  = t.full_name
GROUP BY o.id, o.name, address, c.id, zip, lat, lon, phone, revenue,
    utilization;
```

4 Data Warehouse design

4.1 Design choices

For the design of the data marts, the analysis of the E-R schema (Section 2.1) and the query list (Section 2.2) has led to the identification of 7 facts, which divide the query list as represented in Table 1.

Identified Fact	**Query List**
Administer	• How many individuals, in the age group between 10 and 50 years, were vaccinated against HPV between the years 2015 and 2019, in the counties of Bristol, Essex, and Middlesex, grouped by sex, county, and year of administration?
Affected_By	• What are the most common allergies, grouped by patient age group, and county where they were recorded? • What is the incidence of the various side effects associated with each allergy, broken down by patient age group?
Carry_Out	• What is the cost of medical procedures broken down by year for the various counties and hospital facilities?
Conduct	• What is the number of visits performed by each doctor, broken down by patient age group, gender, race, and, in addition, county and year of the conducted visit?
Plan	• What is the number of respiratory therapies recorded in various hospitals, broken down by gender and age group? • What is the average age of patients affected by acute bronchitis or prediabetes in each county recorded in the last five years? • What is the average number of treatment days required to complete a therapy plan for various diseases in each hospital over the years?
Require	• What are the most used supplies each year, broken down by county and hospital facility?
Take_Place_In	• What is the number of visits made to each hospital facility in the last five years, broken down by gender and age group?

Table 2: Identified facts and associated queries.

4.2 Data mart

For each previously identified fact, the corresponding data mart design schemes follow. To obtain a fact schema, we need to perform the following steps:

1. Construction of the attribute tree
2. Pruning and grafting of the attribute tree
3. Definition of dimensions
4. Definition of measures
5. Creation of the fact schema

The following paragraphs illustrate how the steps are applied to obtain a fact schema.

4.3 Administer

The N-to-N relationship between Immunization and Encounter was reified by creating the entity Administer, Figure 4. The attribute tree was then constructed, as shown in Figure 5.

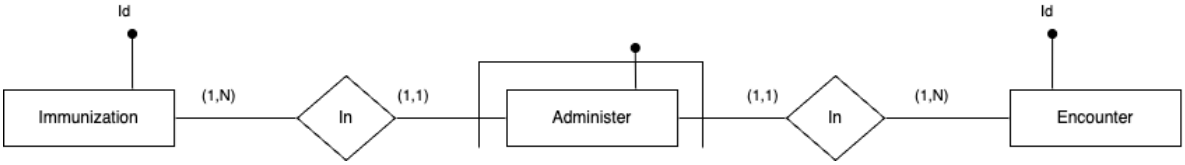

Figure 4: Reification of the "Administer" relationship

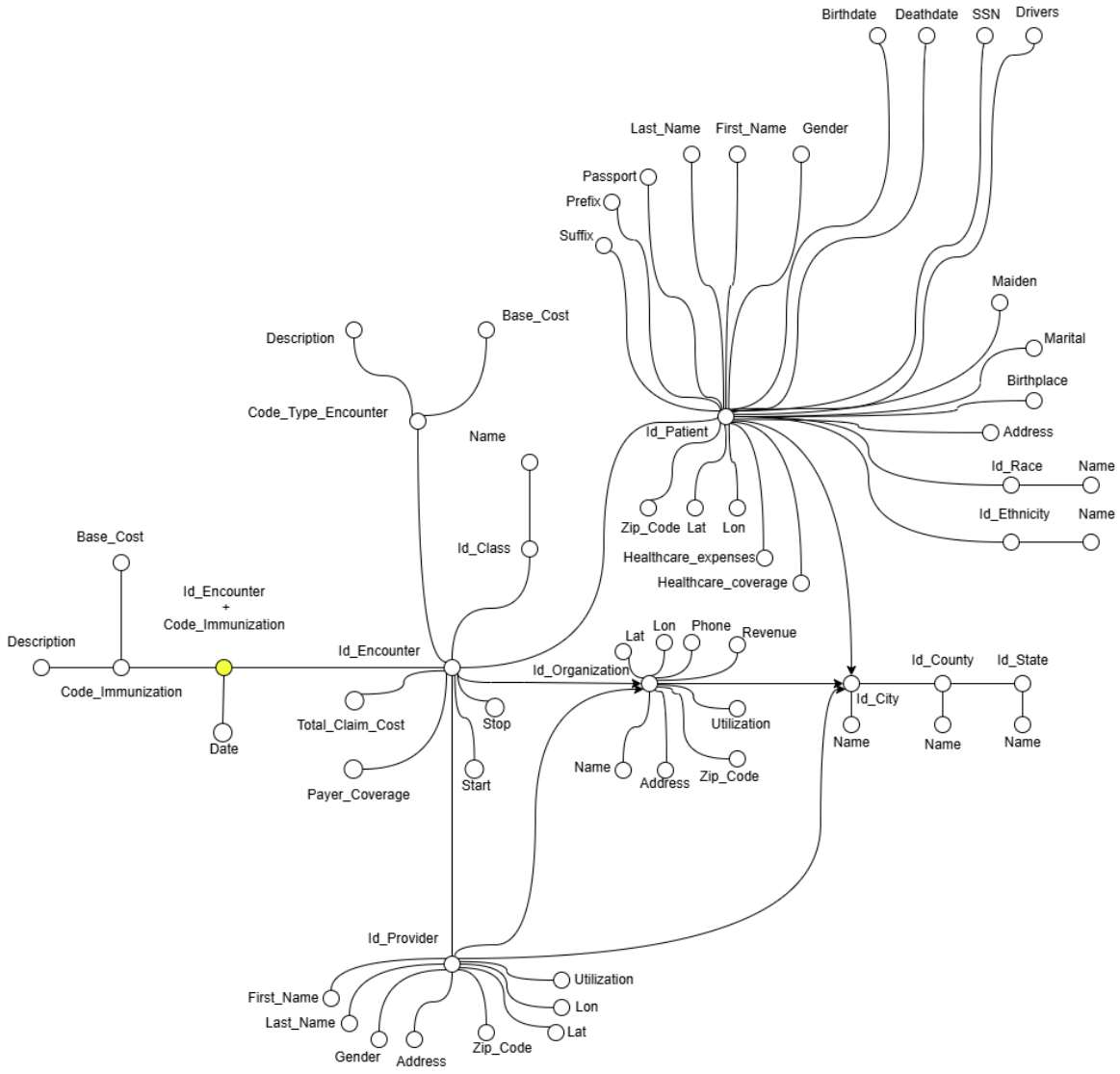

Figure 5: Complete attribute tree of the "Administer" fact

For the construction of the final attribute tree of the fact "Administer", we pruned:

- "Base_Cost" linked to "Code_Immunization"
- "Code_Type_Encounter" and its subtree
- "Id_Provider" and its subtree
- "Id_Class" and its subtree
- "Total_Claim_Cost", "Payer_Coverage", "Start" and "Stop" from "Id_Encounter"
- All attributes associated with "Id_Patient", except for "Gender", "Birthdate", and "Id_Race" along with its subtree
- All attributes associated with "Id_Organization", except for "Id_City" and its subtree

We then applied the grafting operation to the following:

- "Gender", "Birthdate", and "Id_Race" of "Id_Patient"
- "Id_City" of "Id_Organization"
- "Name" attributes, from "Id_County" and "Id_State" have been renamed in "County_Name" and "State_Name", before being grafted

The result is shown in the following figure:

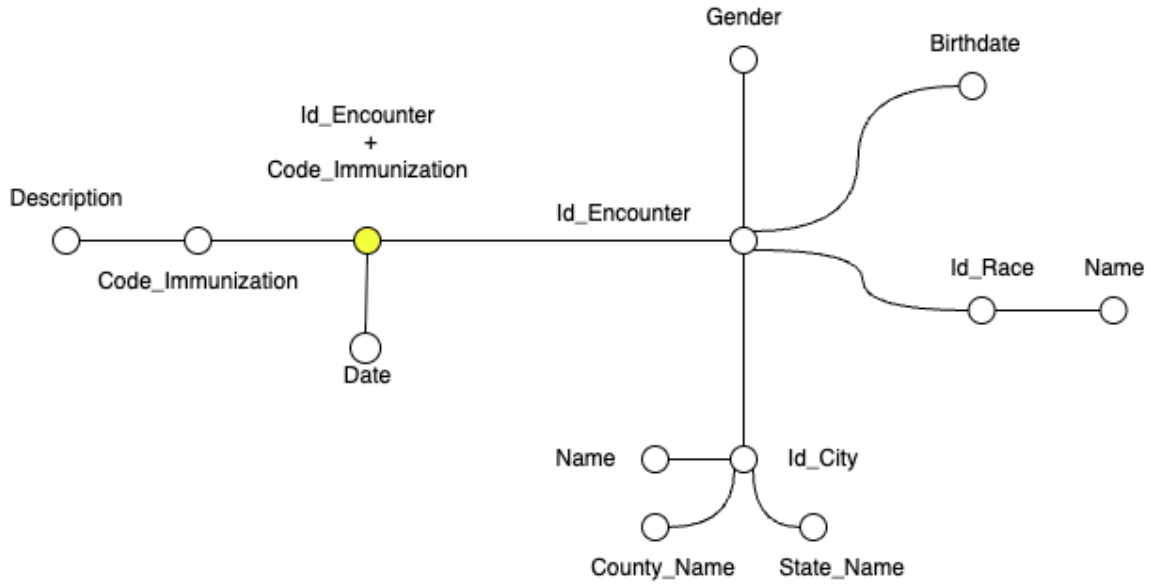

Figure 6: Pruned and grafted attribute tree of the "Administer" fact

Based on the final attribute tree, the fact schema was then constructed (Figure 7). The selected dimensions are: Immunization, Encounter and Date. The chosen measure is Amount_Immunization, obtained by aggregating the attributes from the tree (see Glossary of Measures 4.10). Finally, a date hierarchy was added to facilitate data aggregation during query operations. In the end, the star schema was built (Figure 8).

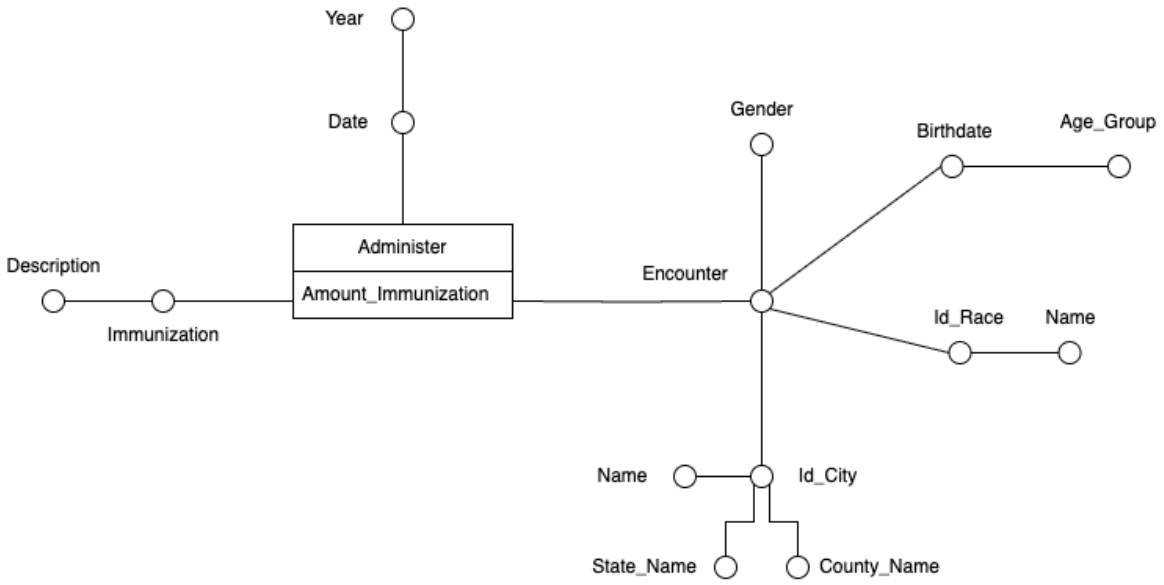

Figure 7: Fact schema of the "Administer" fact

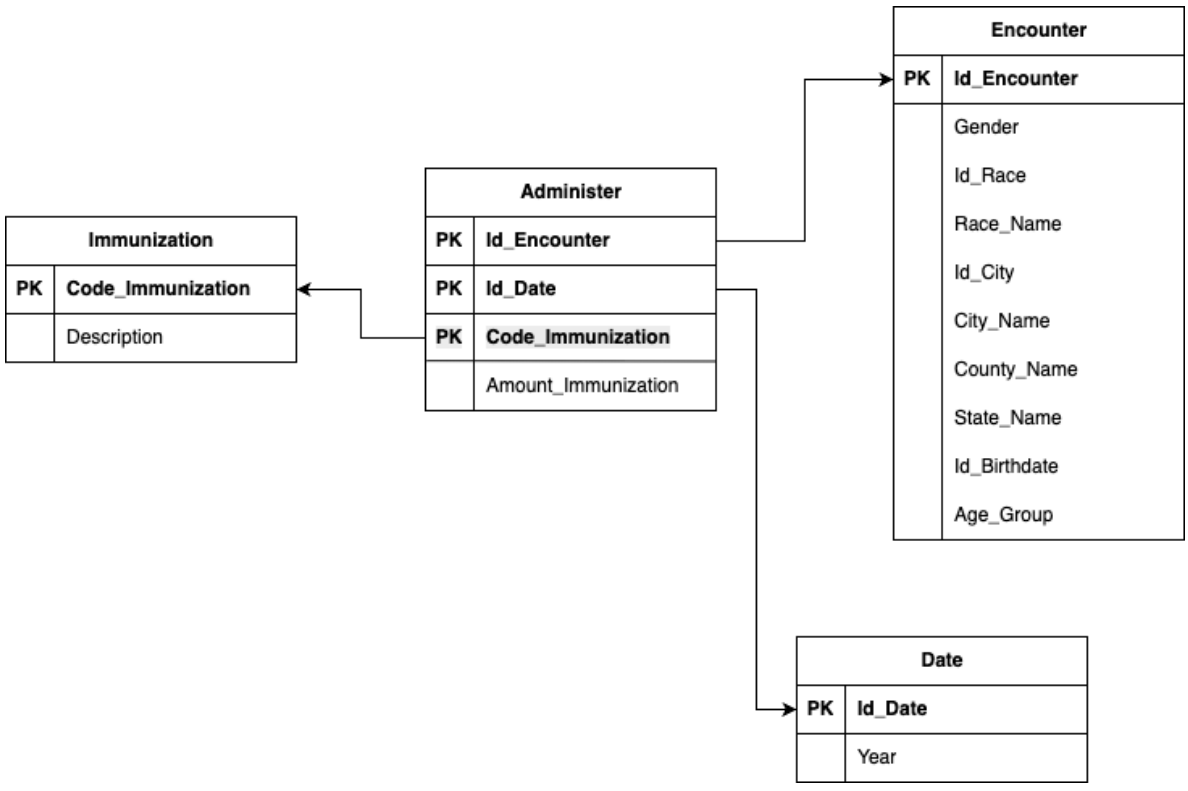

Figure 8: Star schema of the "Administer" fact

4.4 Affected_By

The N-to-N relationship between Conditions, Allergy and Encounter was reified by creating the entity Affected_By, Figure 9. The attribute tree was then constructed, as shown in Figure 10.

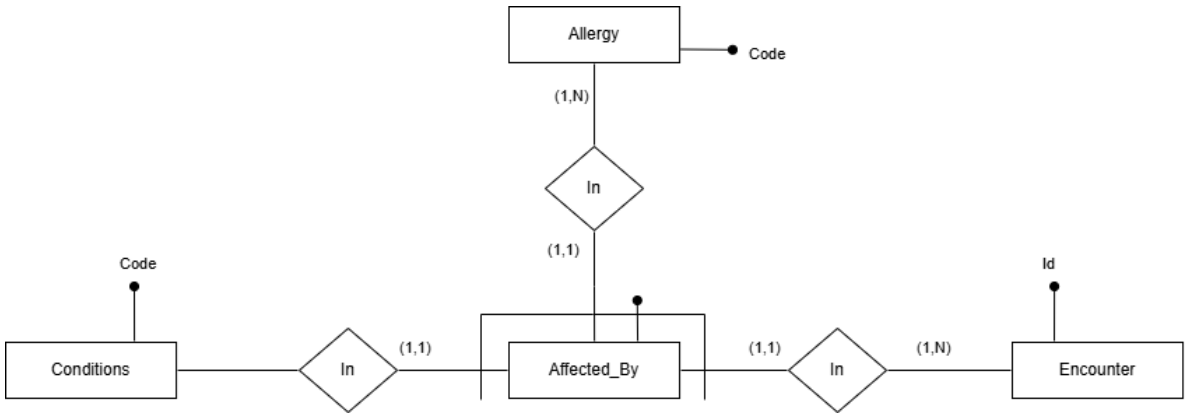

Figure 9: Reification of the "Affected_by" relationship

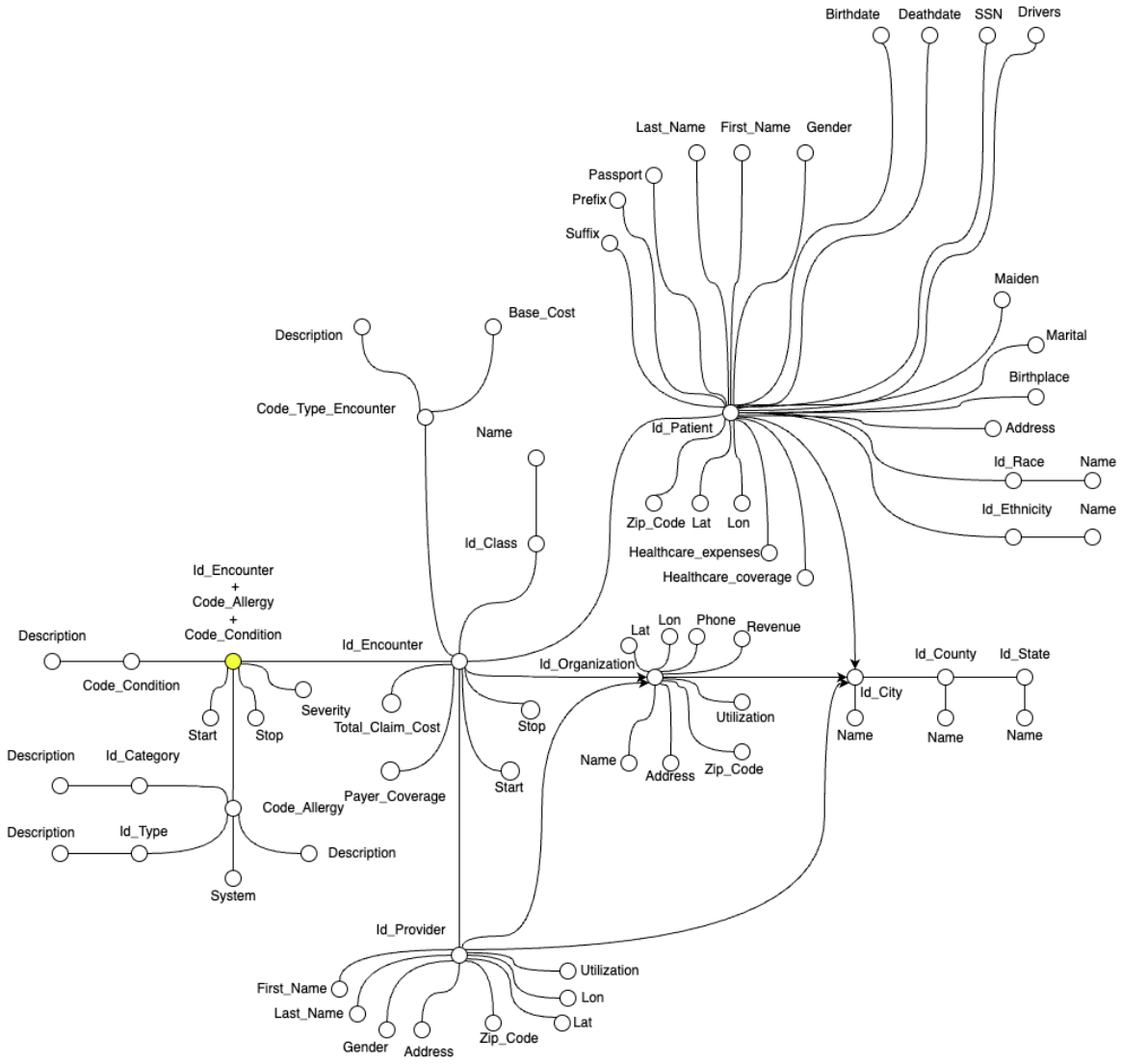

Figure 10: Complete attribute tree of the "Affected_By" fact

For the construction of the final attribute tree of the fact "Affected_By", we pruned:

- "Start", "Stop" and "Severity" from the starting node
- All attributes linked to "Code_Allergy", excluding "Description"
- "Code_Type_Encounter" and its subtree
- "Id_Class" and its subtree
- All attributes associated with "Id_Patient", except for "Gender", "Birthdate", and "Id_Race" along with its subtree
- All attributes associated with "Id_Organization", except for "Id_City" and its subtree

We then applied the grafting operation to the following:

- "Gender", "Birthdate", and "Id_Race" of "Id_Patient"
- "Id_City" of "Id_Organization"
- "Name" attributes, from "Id_County" and "Id_State" have been renamed in "County_Name" and "State_Name", before being grafted

The result is shown in the following figure:

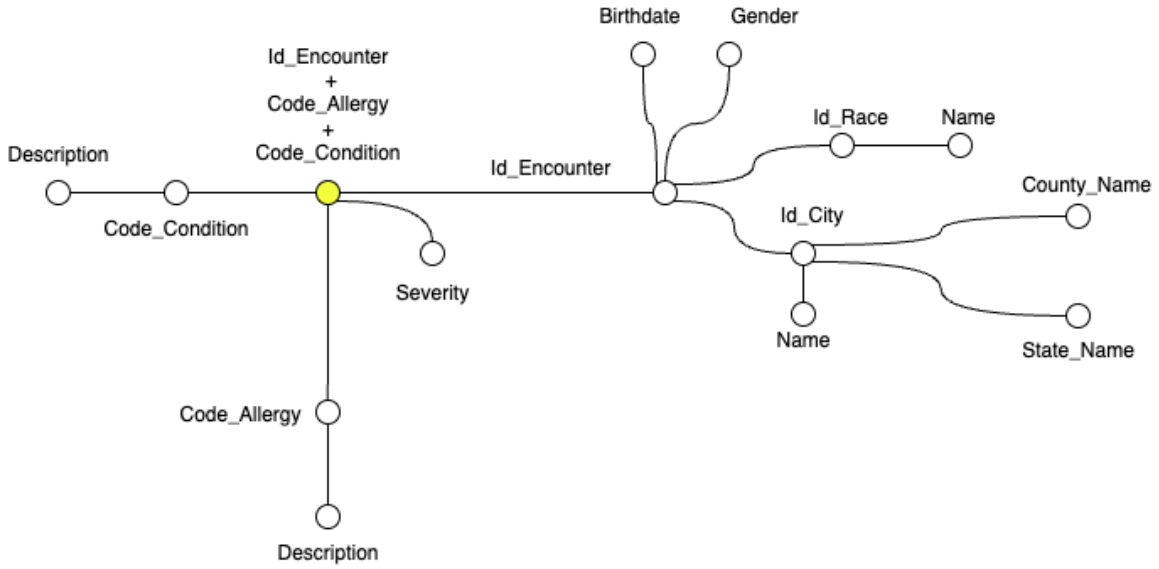

Figure 11: Pruned and grafted attribute tree of the "Affected_By" fact

Based on the final attribute tree, the fact schema was constructed (Figure 12). The selected dimensions are: Condition, Encounter and Allergy. The chosen measure are Amount_Condition and Amount_Allergy, obtained by aggregating the attributes from the tree (see Glossary of Measures 4.10). In the end, the star schema was built (Figure 13).

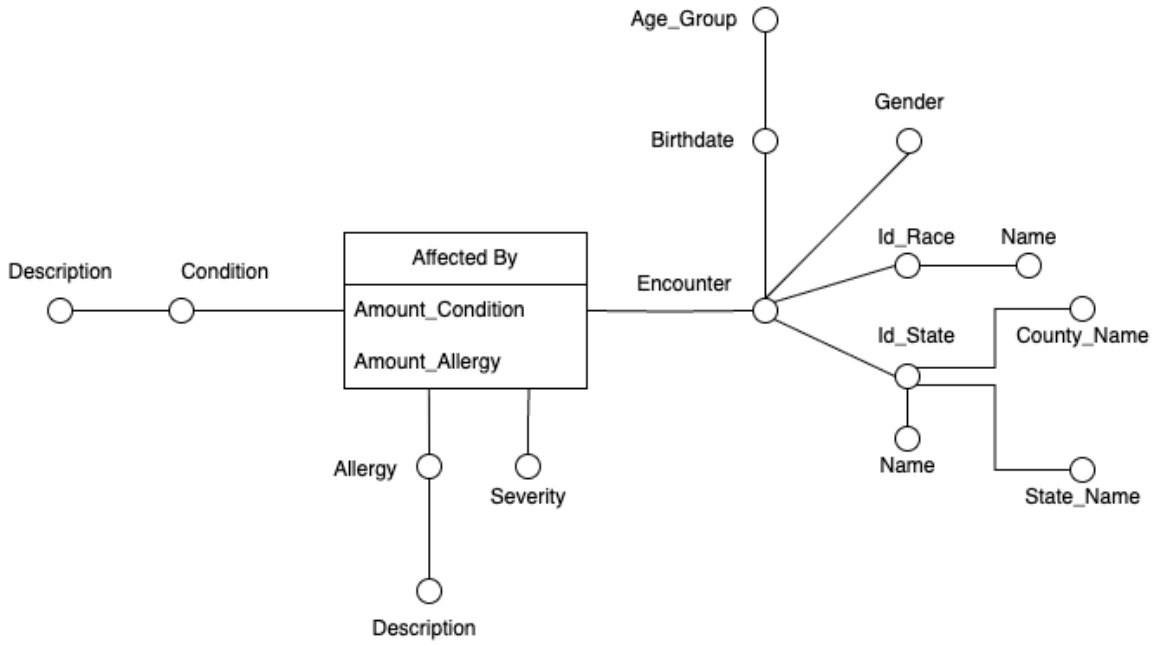

Figure 12: Fact schema of the "Affected_By" fact

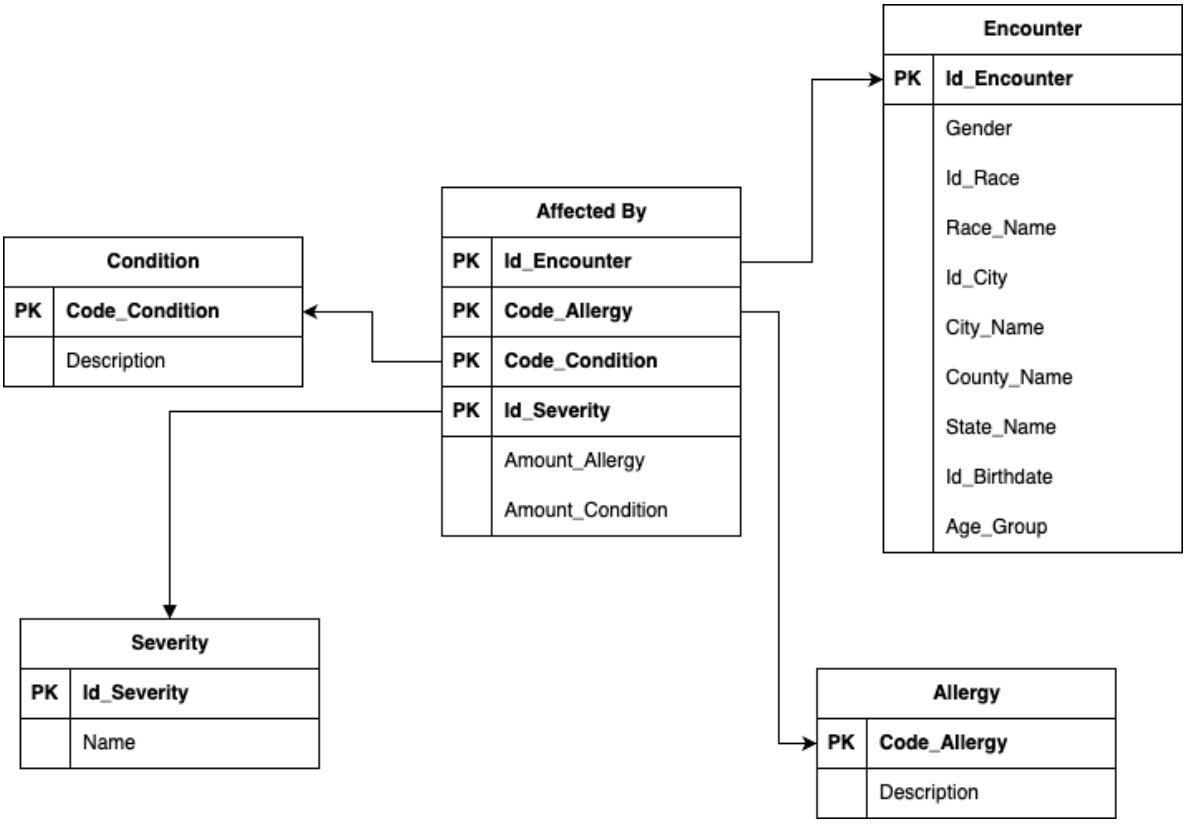

Figure 13: Star schema of the "Affected_By" fact

4.5 Carry_Out

The N-to-N relationship between Procedure and Encounter was reified by creating the entity Carry_Out, Figure 14. The attribute tree was then constructed, as shown in Figure 15.

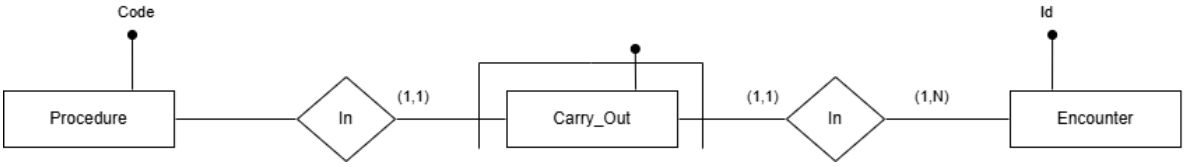

Figure 14: Reification of the "Carry_out" relationship

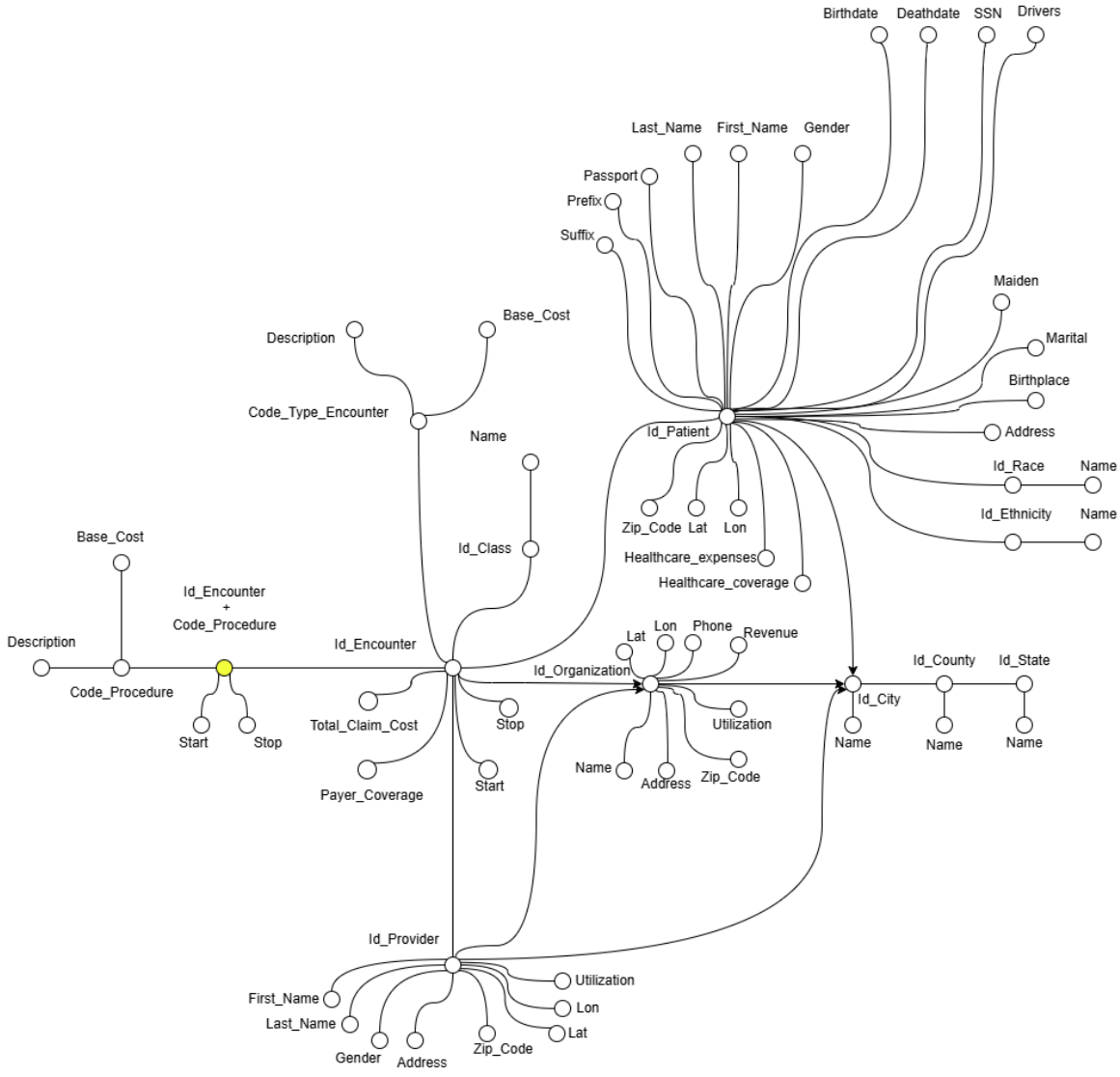

Figure 15: Complete attribute tree of the "Carry_Out" fact

For the construction of the final attribute tree of the fact "Carry_Out", we pruned:

- "Stop" from the starting node
- All attributes linked to "Id_Encounter", excluding "Id_Organization" and its subtree
- All attributes linked to "Id_Organization", excluding "Name" and "Id_City"
- "Name" from "Id_City"
- "Id_State" and its subtree

We then applied the grafting operation to the following:

- "Id_County" of "Id_City"

The result is shown in the following figure:

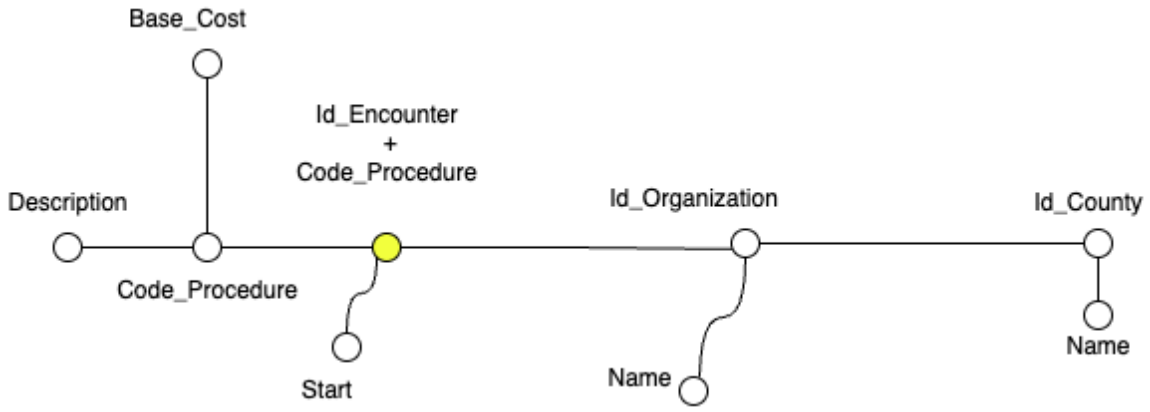

Figure 16: Pruned and grafted attribute tree of the "Carry_out" fact

Based on the final attribute tree, the fact schema was constructed (Figure 17). The selected dimensions are: Organization, Procedure and Start. The chosen measure is Total_Cost, obtained by aggregating the attributes from the tree (see Glossary of Measures 4.10). Finally, a start hierarchy was added to facilitate data aggregation during query operations. In the end, the star schema was built (Figure 18).

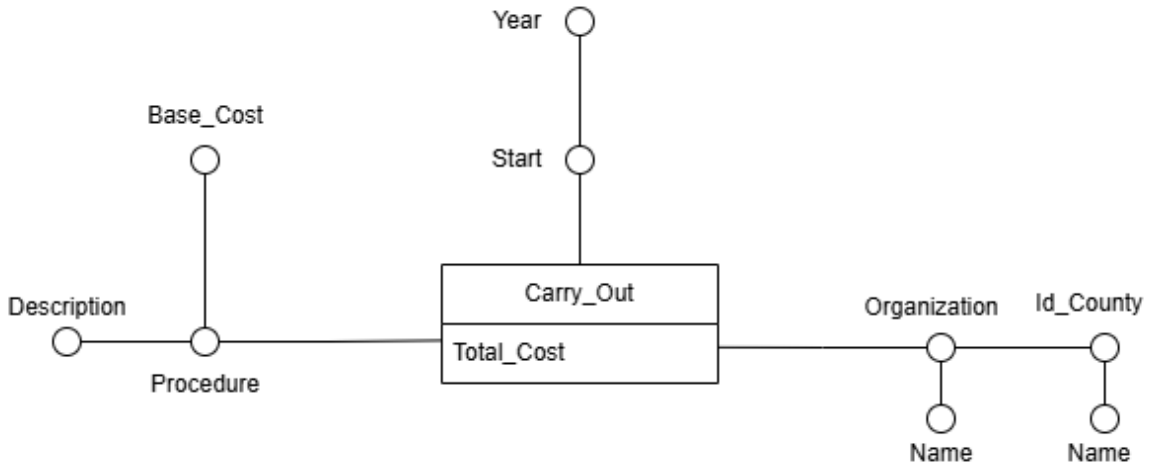

Figure 17: Fact schema of the "Carry_Out" fact

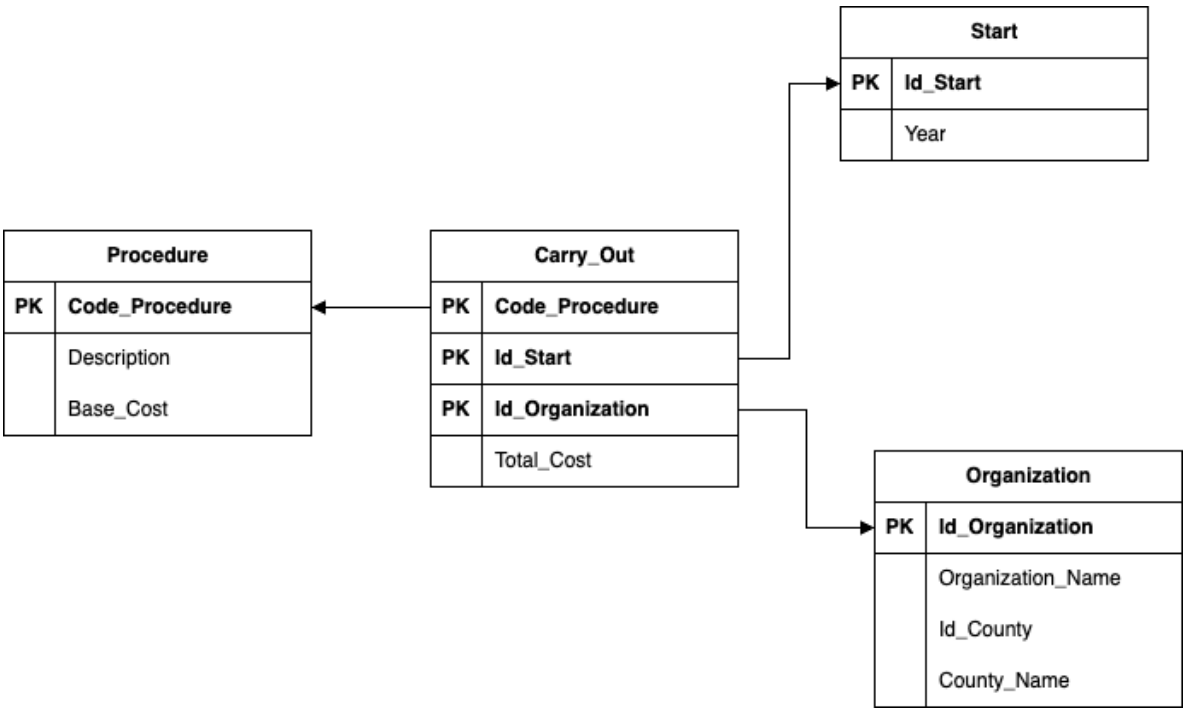

Figure 18: Star schema of the "Carry_Out" fact

4.6 Conduct

The N-to-N relationship between Providers and Encounter was reified by creating the entity Conduct, Figure 19. The attribute tree was then constructed, as shown in Figure 20.

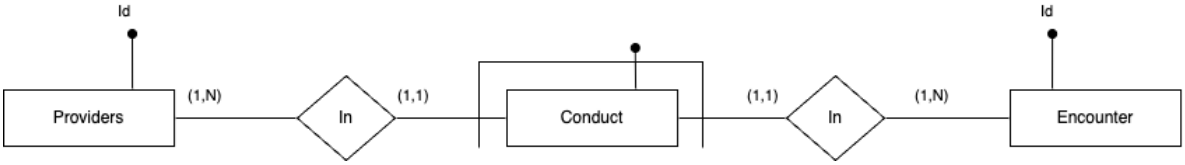

Figure 19: Reification of the "Conduct" relationship

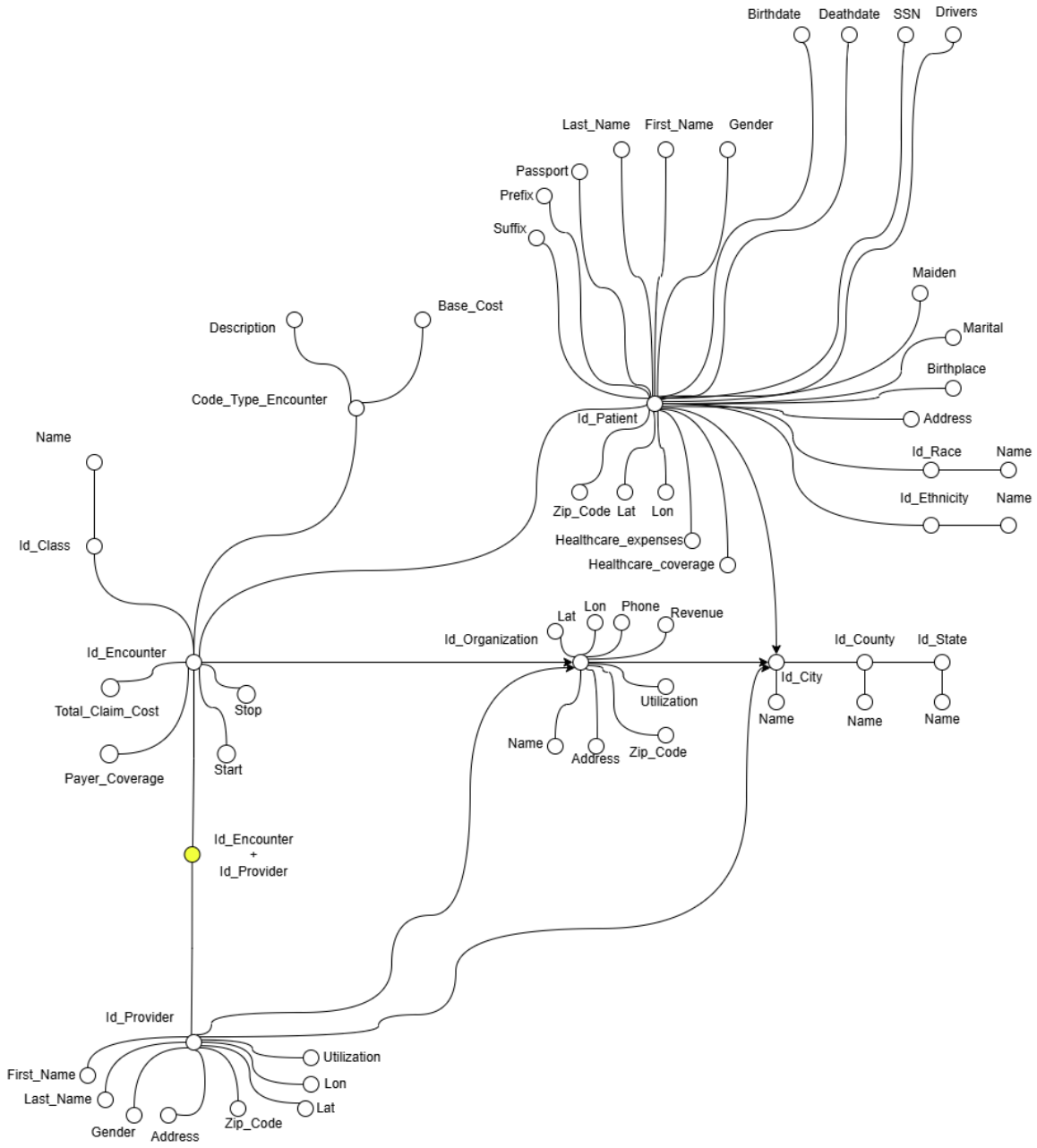

Figure 20: Complete attribute tree of the "Conduct" fact

For the construction of the final attribute tree of the fact "Conduct", we pruned:

- All attributes linked to "Id_Provider", except for "First_Name" and "Last_Name"
- All attributes linked to "Id_Encounter", excluding "Start", "Id_Organization" and its subtree and "Id_Patient" with its subtree
- All attributes of "Id_Patient", except for "Gender", "Birthdate", "Id_Race" and its subtree
- All attributes of "Id_Organization", excluding "Id_City" and its subtree
- "Name" from "Id_City"
- "Id_State" and its subtree

We then applied the grafting operation to the following:

- "Id_County" of "Id_City" and then of "Id_Organization"

The result is shown in the following figure:

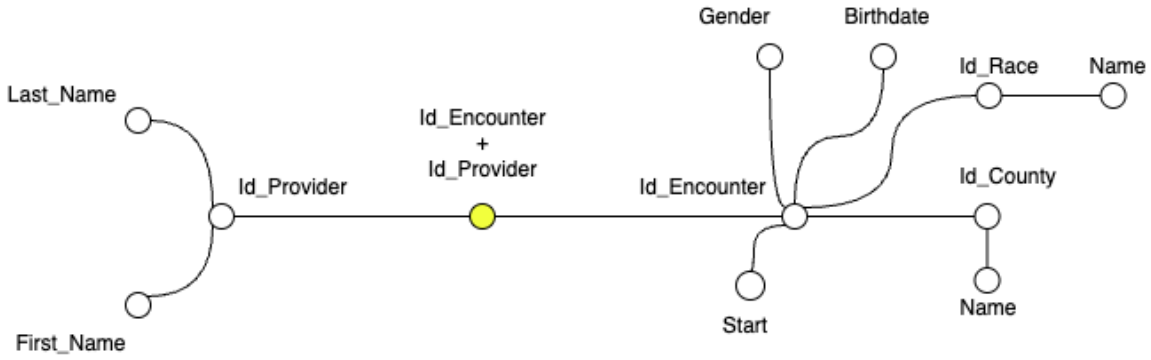

Figure 21: Pruned and grafted attribute tree of the "Conduct" fact

Based on the final attribute tree, the fact schema was constructed (Figure 21). The selected dimensions are: Provider and Encounter. The chosen measure is Amount_Visit, obtained by aggregating the attributes from the tree (see Glossary of Measures 4.10). Finally, a start hierarchy and a birthdate hierarchy were added to facilitate data aggregation during query operations. In the end, the star schema was built (Figure 22).

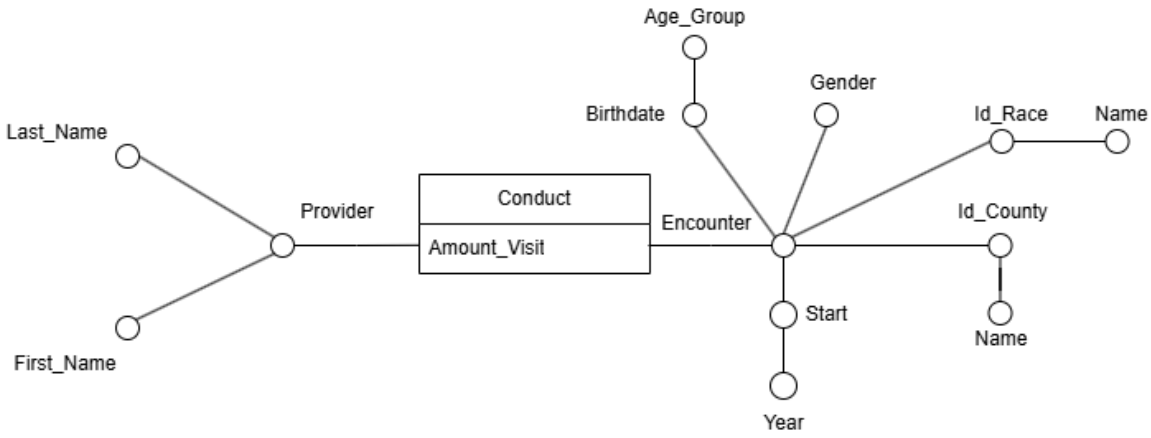

Figure 22: Fact schema of the "Conduct" fact

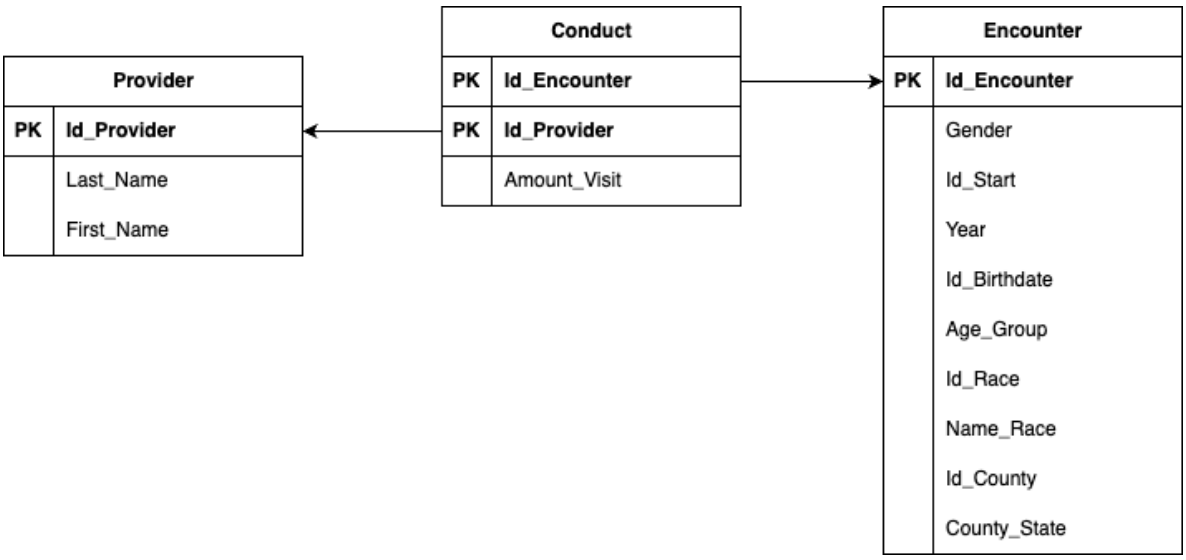

Figure 23: Star schema of the "Conduct" fact

4.7 Plan

The N-to-N relationship between Conditions, Careplans, and Encounter was reified by creating the entity Plan, Figure 24. The attribute tree was then constructed, as shown in Figure 25.

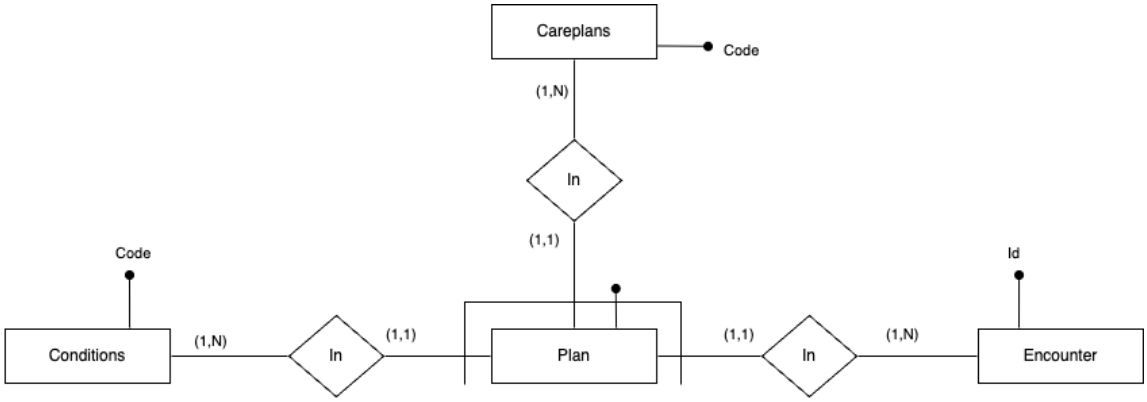

Figure 24: Reification of the "Plan" relationship

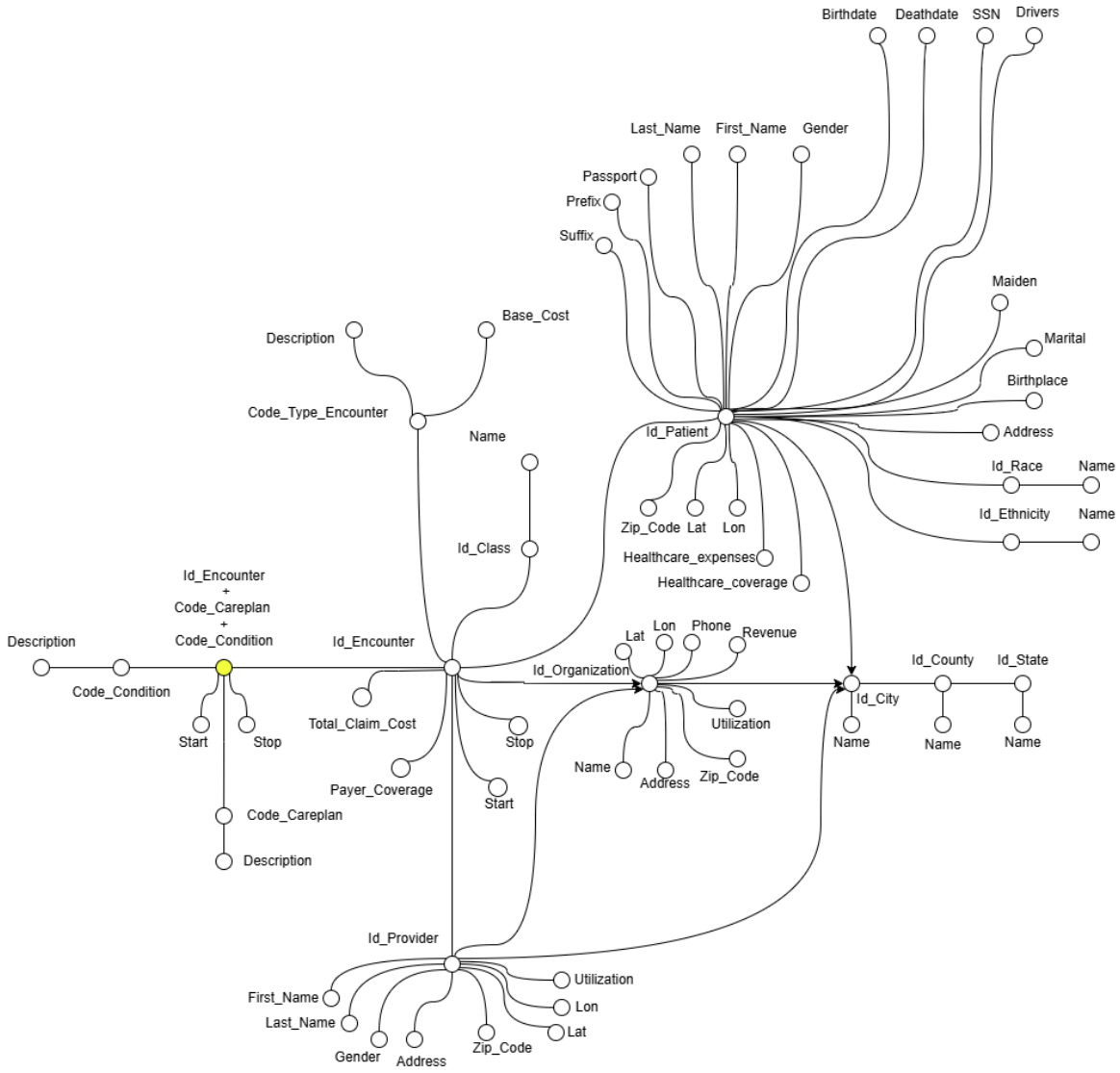

Figure 25: Complete attribute tree of the "Plan" fact

For the construction of the final attribute tree of the fact "Plan", we pruned:

- "Stop" from the starting node
- All attributes linked to "Id_Encounter", excluding "Id_Organization" and its subtree and "Id_Patient" and its subtree
- All attributes of "Id_Patient", except for "Gender" and "Birthdate"
- All attributes linked to "Id_Organization", excluding "Name" and "Id_City"

We then applied the grafting operation to the following:

- "Gender" and "Birthdate" of "Id_Patient"
- "Name" attributes, from "Id_County" and "Id_State" have been renamed in "County_Name" and "State_Name", before being grafted

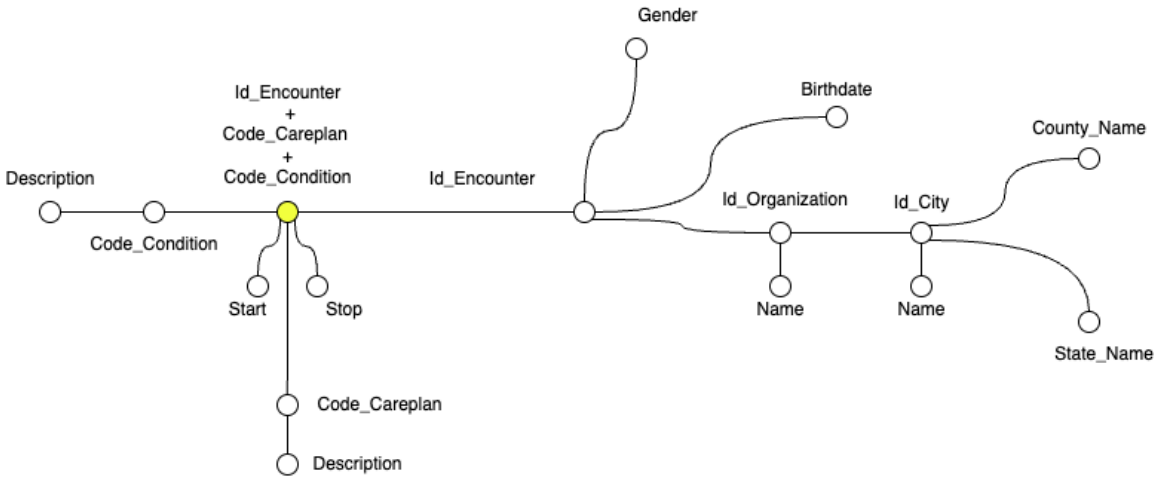

Figure 26: Pruned and grafted attribute tree of the "Plan" fact

Based on the final attribute tree, the fact schema was constructed (Figure 27). The selected dimensions are: Careplan, Condition, Encounter and Start. The chosen measure are Average_Age, Average_Days_Of_Hospitalization and Amount_Therapies, obtained by aggregating the attributes from the tree (see Glossary of Measures 4.10). Finally, several hierarchies were added to facilitate data aggregation during query operations. In the end, the star schema was built (Figure 28).

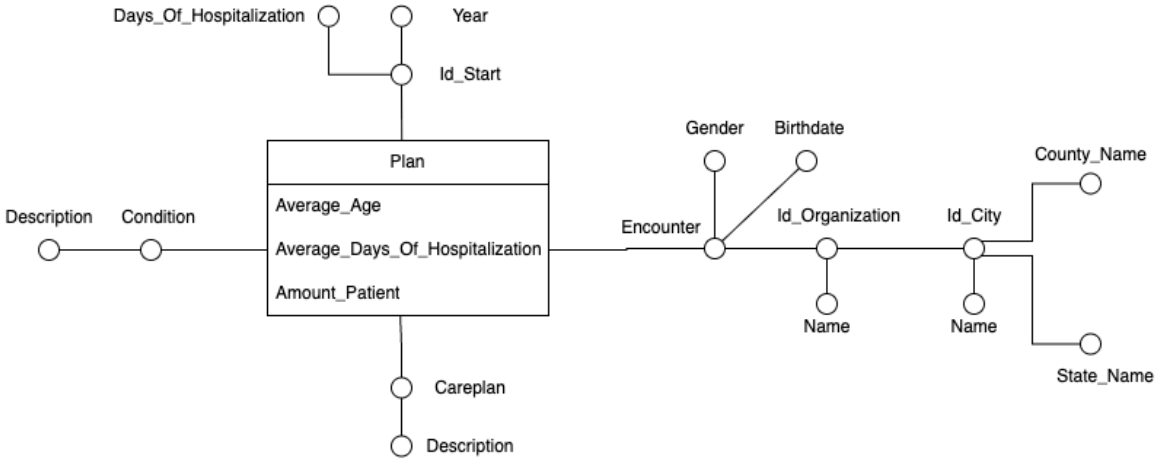

Figure 27: Fact schema of the "Plan" fact

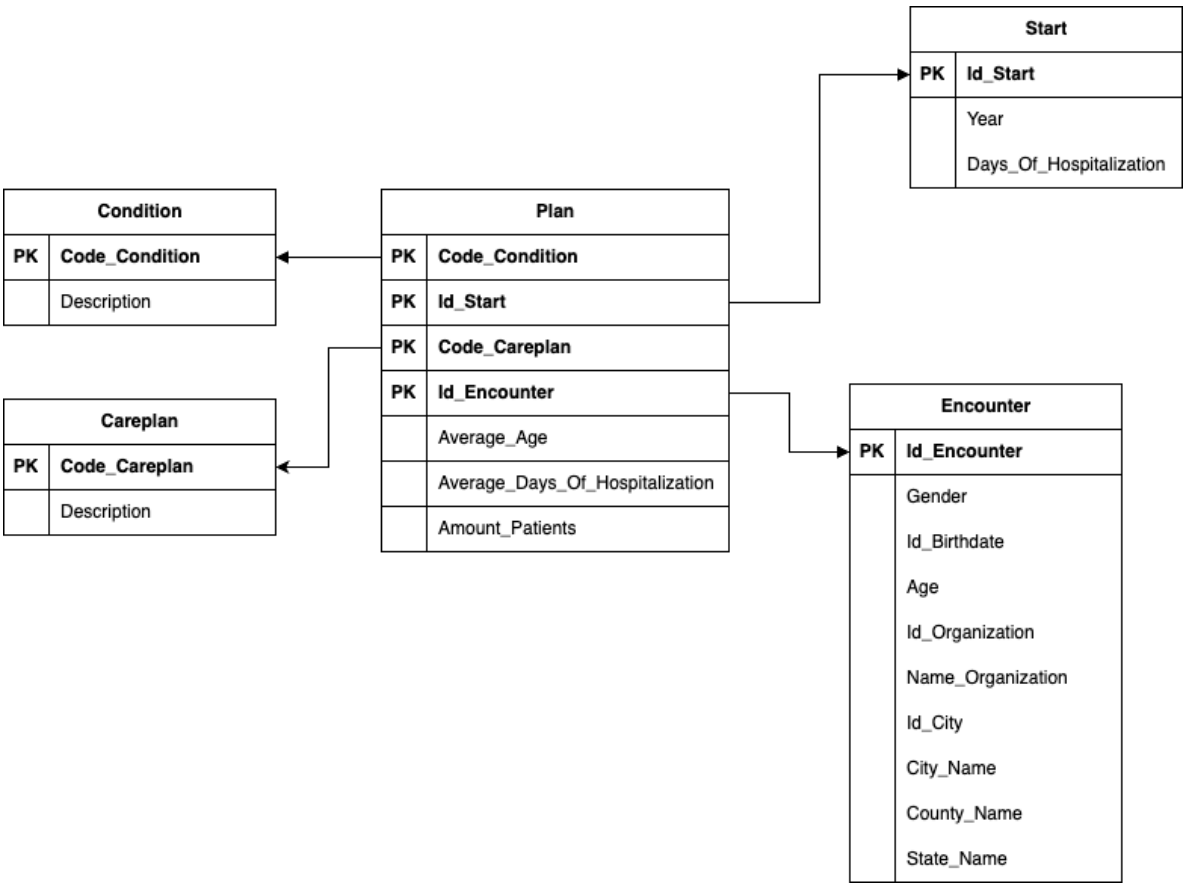

Figure 28: Star schema of the "Plan" fact

4.8 Require

The N-to-N relationship between Supply and Encounter was reified by creating the entity Require, Figure 29. The attribute tree was then constructed, as shown in Figure 30.

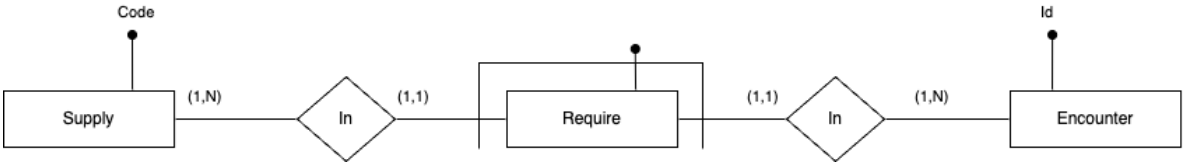

Figure 29: Reification of the "Require" relationship

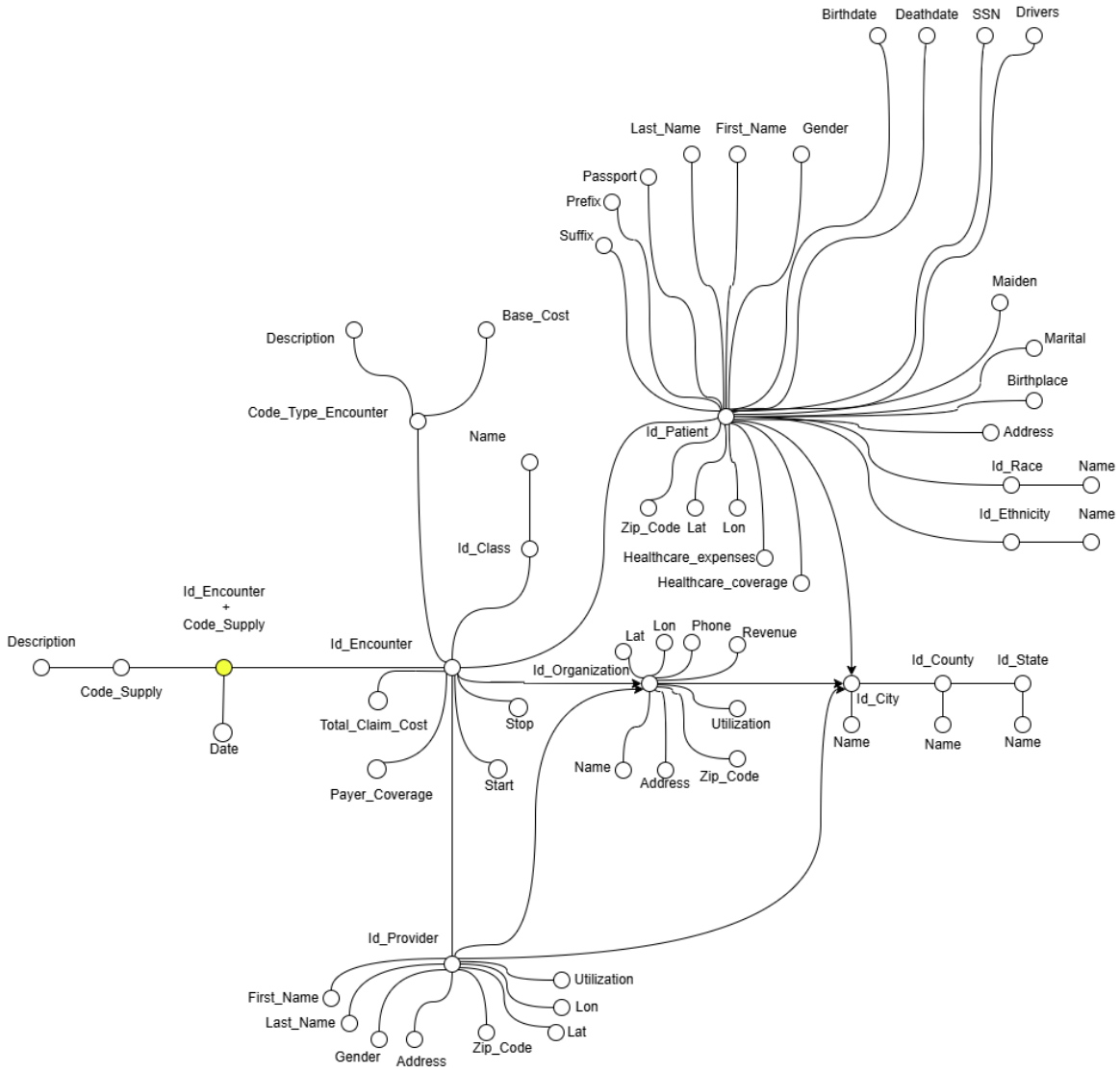

Figure 30: Complete attribute tree of the "Require" fact

For the construction of the final attribute tree of the fact "Require", we pruned:

- All attributes linked to "Id_Encounter", except for "Id_Organization"
- All attributes of "Id_Organization", except for "Id_City"
- "Name" from "Id_City"
- "Id_State" and its subtree

We then applied the grafting operation to the following:

- "Id_County" of "Id_City"

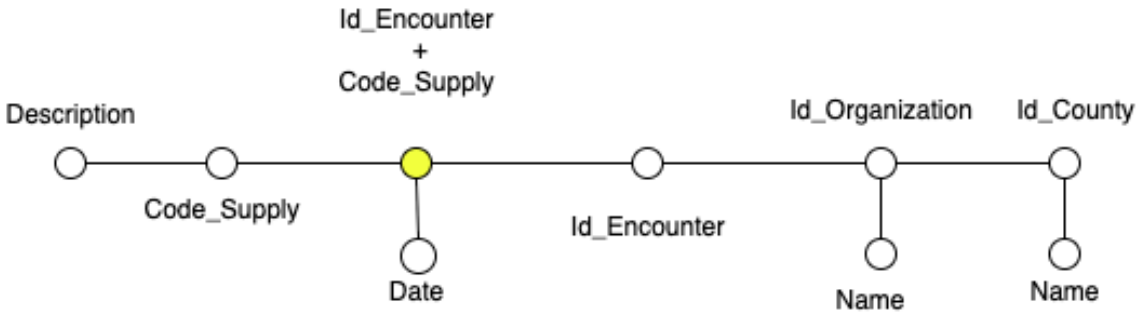

Figure 31: Pruned and grafted attribute tree of the "Require" fact

Based on the final attribute tree, the fact schema was constructed (Figure 32). The selected dimensions are: Date, Encounter and Supply. The chosen measure is Amount_Supply, obtained by aggregating the attributes from the tree (see Glossary of Measures 4.10). Finally, a Date hierarchy was added to facilitate data aggregation during query operations. In the end, the star schema was built (Figure 33).

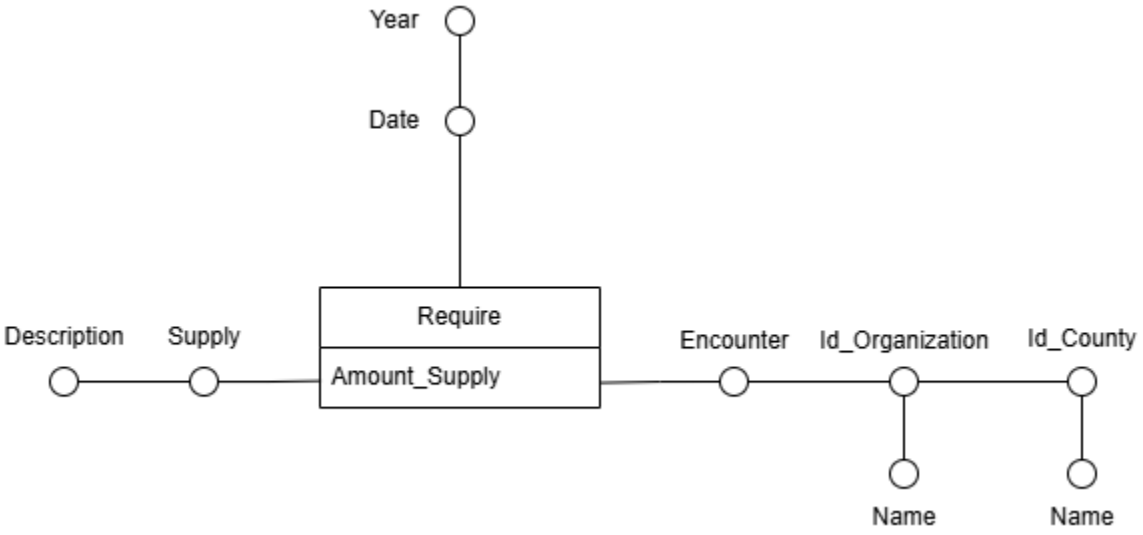

Figure 32: Fact schema of the "Require" fact

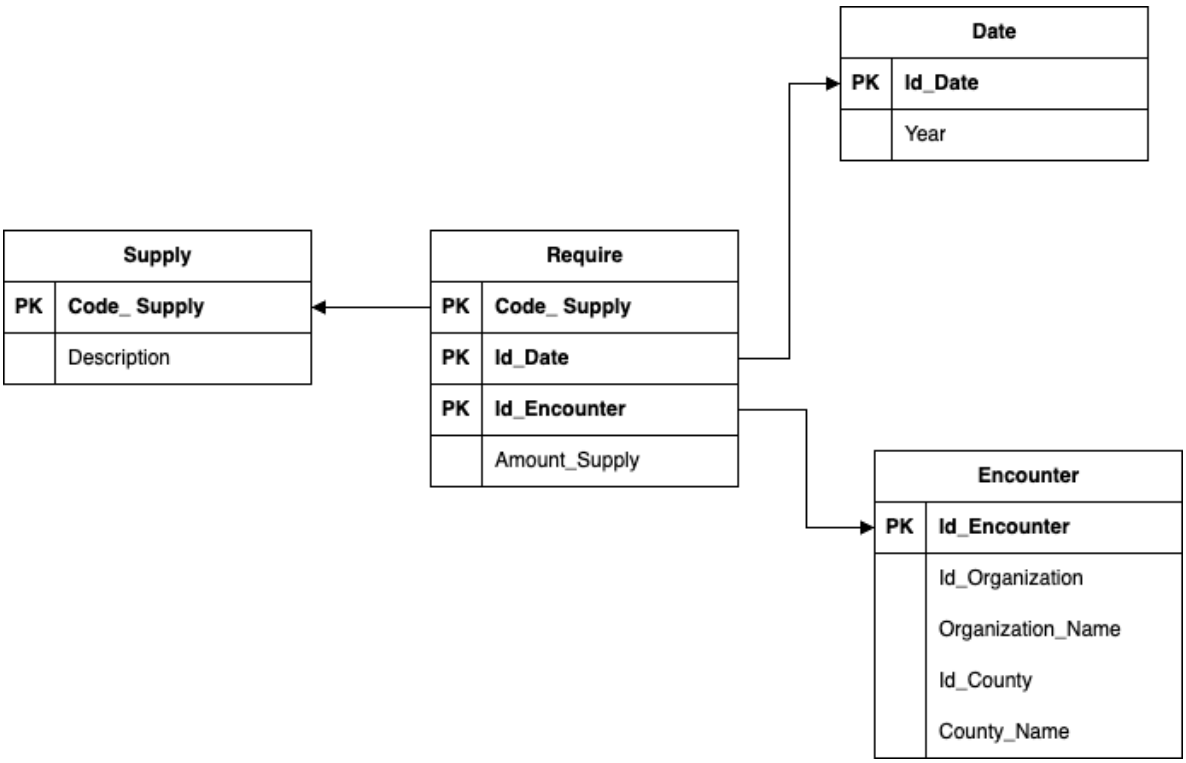

Figure 33: Star schema of the "Require" fact

4.9 Take_Place_In

The N-to-N relationship between Organization and Encounter was reified by creating the entity Take_Place_In, Figure 34. The attribute tree was then constructed, as shown in Figure 35.

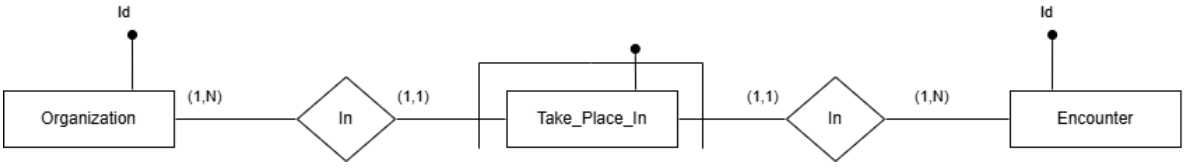

Figure 34: Reification of the "Take_Place_In" relationship

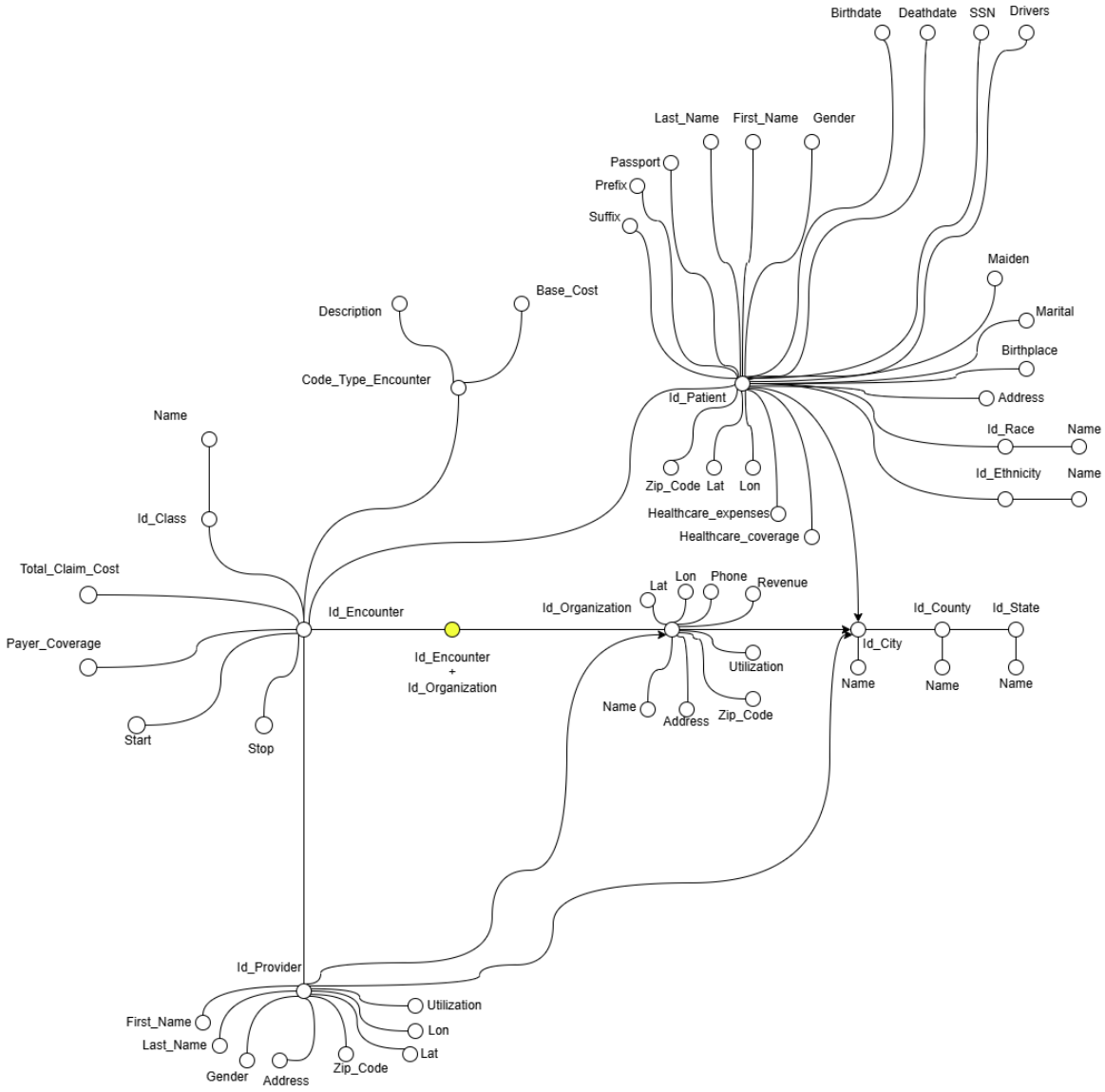

Figure 35: Complete attribute tree of the "Take_Place_In" fact

For the construction of the final attribute tree of the fact "Take_Place_In", we pruned:

- All attributes linked to "Id_Encounter", except for "Id_Patient" and "Start"
- All attributes of "Id_Organization", except for "Name"

We then applied the grafting operation to the following:

- "Birthdate" and "Gender" of "Id_Patient"

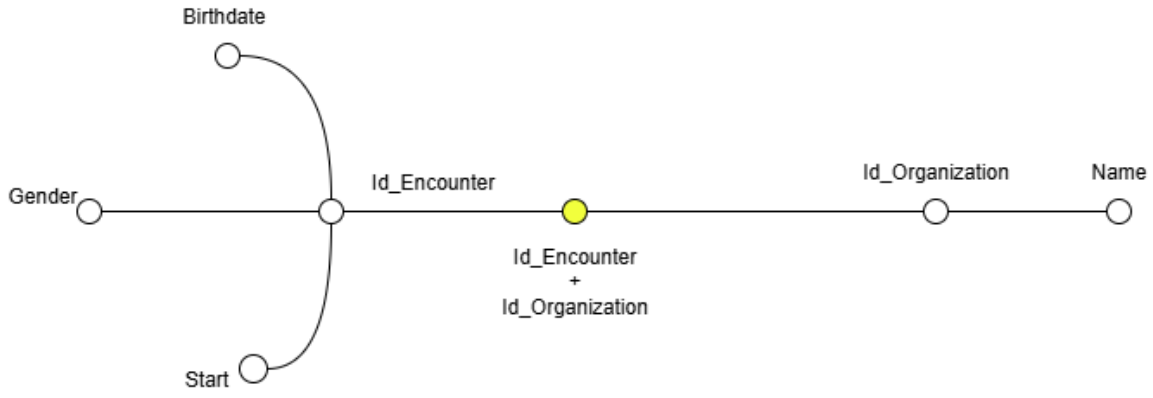

Figure 36: Pruned and grafted attribute tree of the "Take_Place_In" fact

Based on the final attribute tree, the fact schema was constructed (Figure 37). The selected dimensions are: Encounter and Organization. The chosen measure is Amount_Visit, obtained by aggregating the attributes from the tree (see Glossary of Measures 4.10). Finally, several hierarchy was added to facilitate data aggregation during query operations. In the end, the star schema was built (Figure 38).

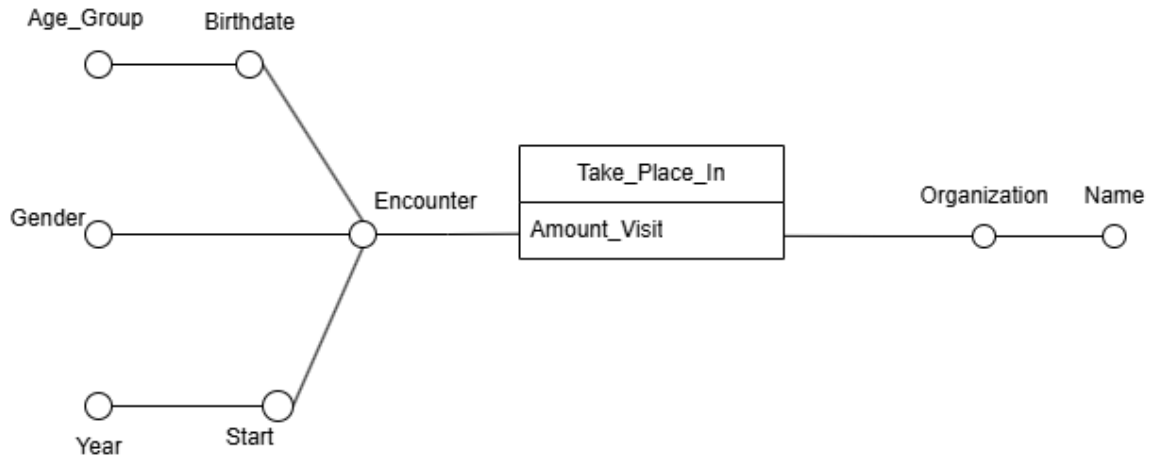

Figure 37: Fact schema of the "Take_Place_In" fact

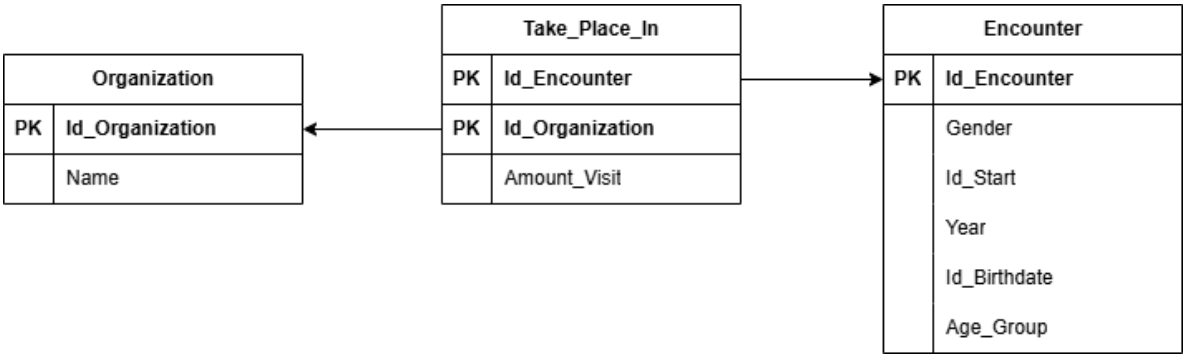

Figure 38: Star schema of the "Take_Place_In" fact

4.10 Glossary of Measures

Data mart	**Measures**
Administer	• `Amount_Immunization` ← `COUNT(Encounter.Id_Encounter)`
Affected_By	• `Amount_Allergy` ← `COUNT(Encounter.Id_Encounter) GROUP BY Allergy.Code_Allergy` • `Amount_Condition` ← `COUNT(Encounter.Id_Encounter) GROUP BY Condition.Code_Condition`
Carry_Out	• `Total_Cost` ← `SUM(Procedure.Base_Cost)`
Conduct	• `Amount_Visit` ← `COUNT(Encounter.Id_Encounter)`
Plan	• `Amount_Patient` ← `COUNT(Encounter.Id_Encounter)` • `Average_Days_Of_Hospitalization` ← `AVG(Start.Days_Of_Hospitalization)` • `Average_Age` ← `AVG(Encounter.Age)`
Require	• `Amount_Supply` ← `COUNT(Encounter.Id_Encounter)`
Take_Place_In	• `Amount_Visit` ← `COUNT(Encounter.Id_Encounter)`

Table 3: Facts identified for design and associated queries.

5 Use of the Data Warehouse

The Data Marts identified during the conceptual design phase were implemented. The software used for graphical representation is Tableau Software, a visual analytics platform that empowers individuals and organizations to understand and utilize data for problem-solving.

5.1 Administer

- **How many individuals, in the age group between 10 and 50 years, were vaccinated against HPV between the years 2015 and 2019, in the counties of Bristol, Essex, and Middlesex, grouped by sex, county, and year of administration?**

The selected dimensions are: Id_Date, County_Name, and Gender. The chosen measure is Amount_Immunization. Figure 39 shows the histogram resulting from the query.

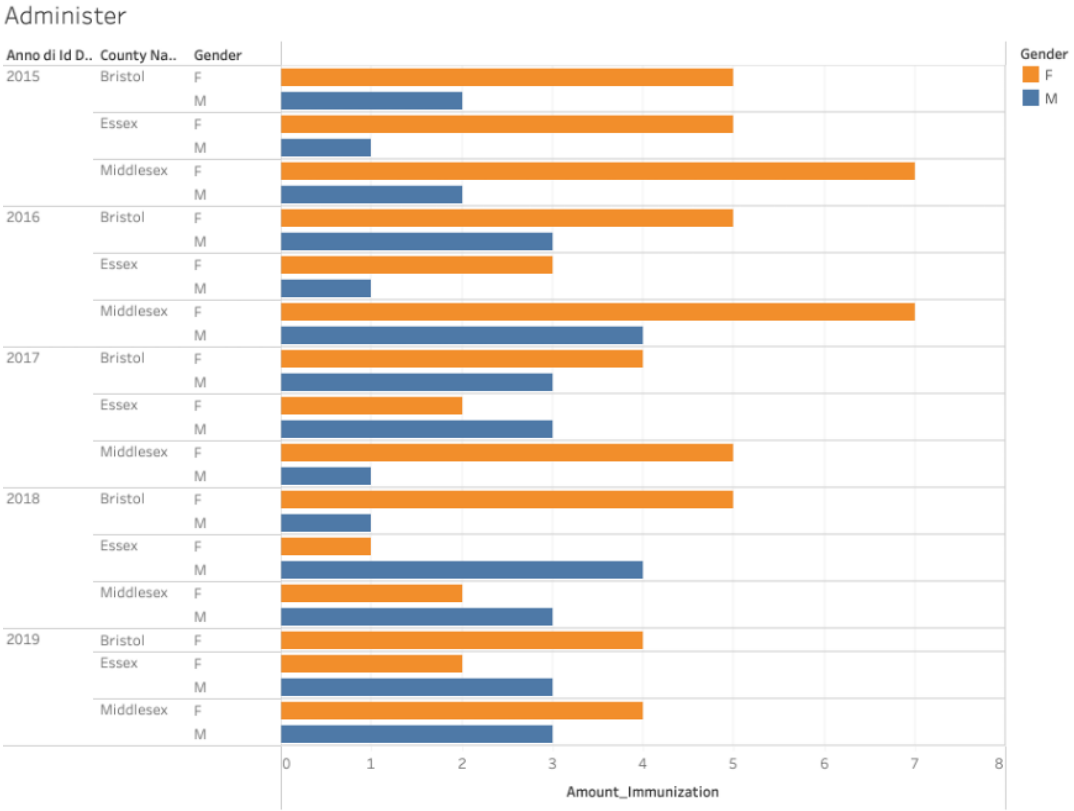

Figure 39: "Administer" query analyzed on tableau

5.2 Affected_By

- **What are the most common allergies, grouped by patient age group, and county where they were recorded?**

The selected dimensions are: Description of the allergy, County_Name and Age_Group. The chosen measure is Amount_Allergy. Figure 40 shows the histogram resulting from the query.

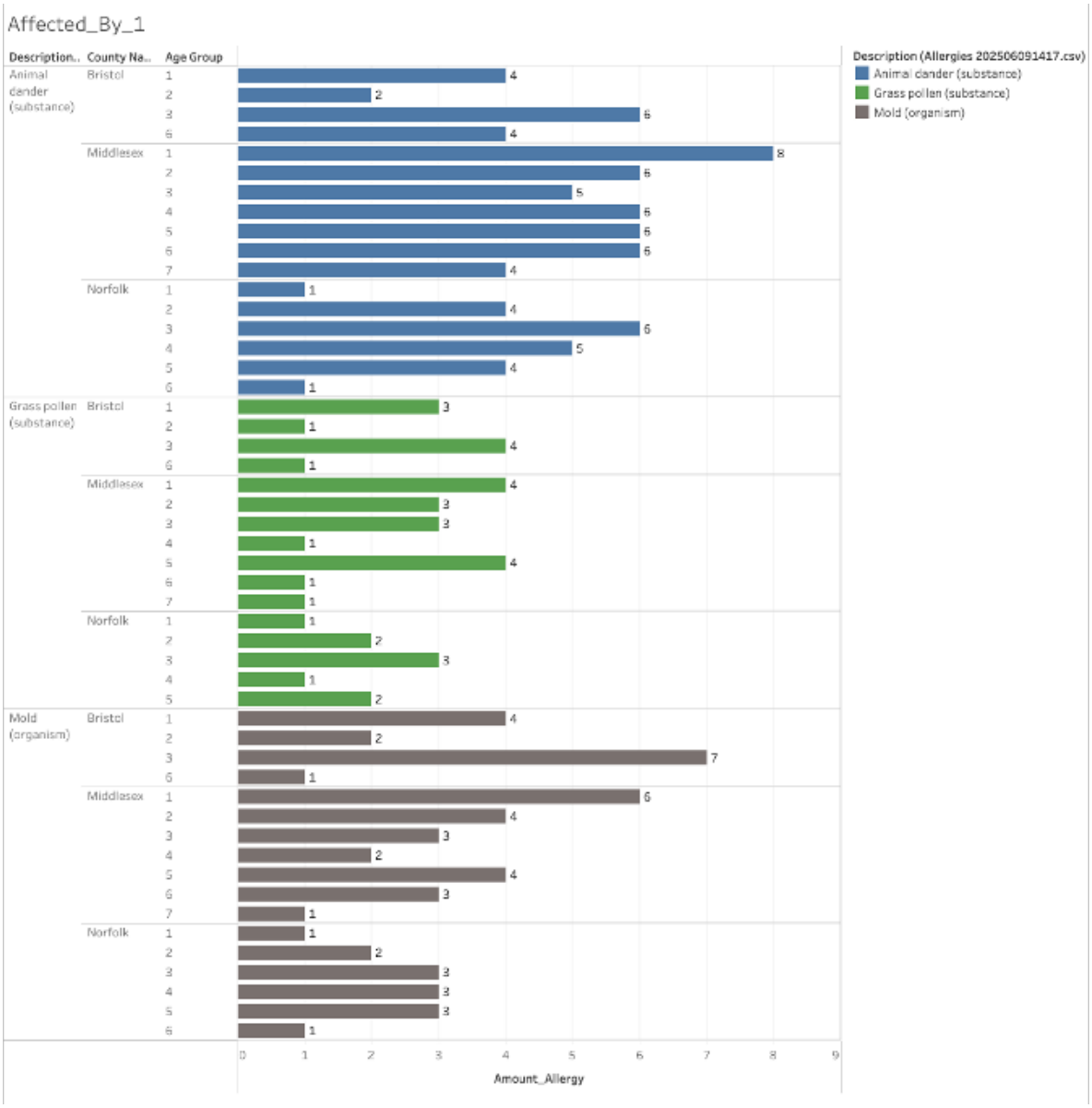

Figure 40: First "Affected_By" query analyzed on tableau

- **What is the incidence of the various side effects associated with each allergy, broken down by patient age group?**

The selected dimensions are: Description (of the side effect), Description (of the allergen), and Age_Group. The chosen measure is Amount_Conditions. Figure 41 shows the histogram resulting from the query. Note that only two allergens are plotted, because of readability issues.

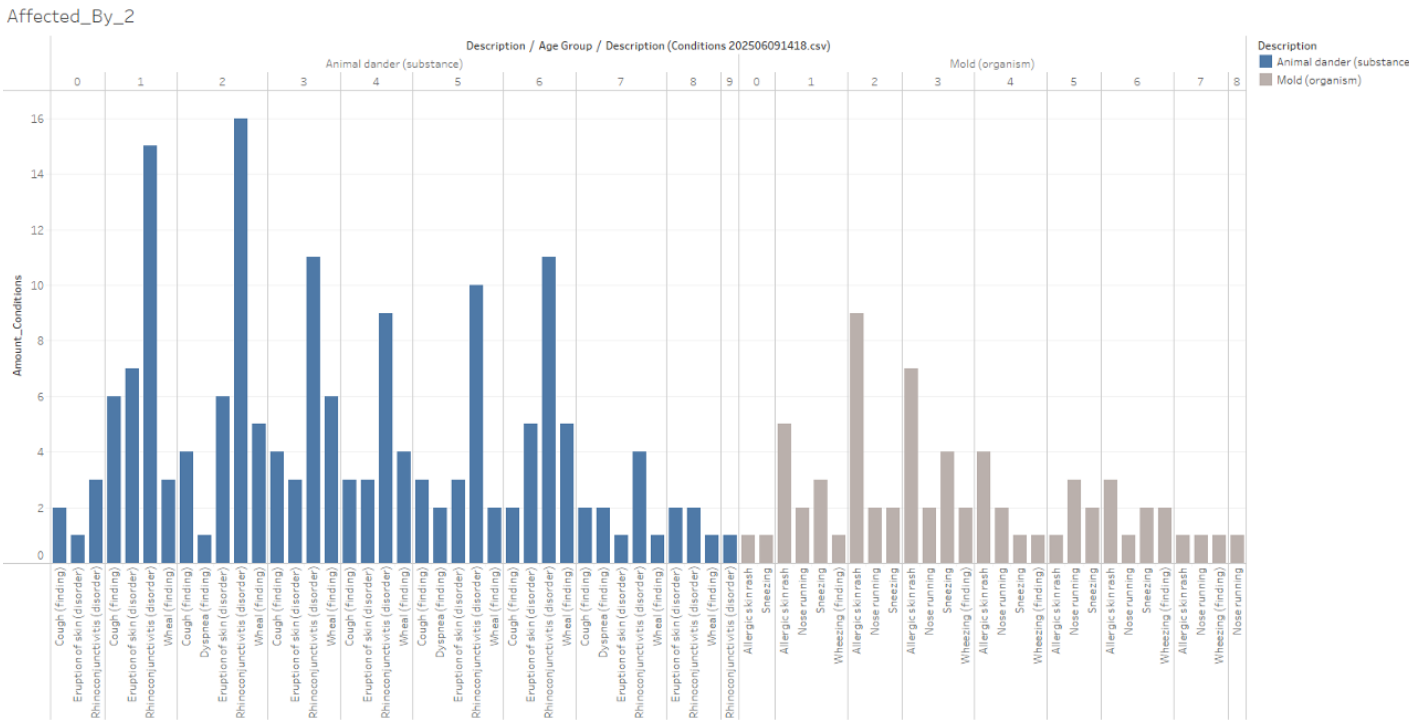

Figure 41: Second "Affected_By" query analyzed on tableau

5.3 Plan

- **What is the number of respiratory therapies recorded in various hospitals, broken down by gender and age group?**

The selected dimensions are: Organization_Name, Gender and Age_Group. The chosen measure is Amount_Patients. Figure 42 shows the histogram resulting from the query. Note that only four Organizations are plotted, because of readability issues.

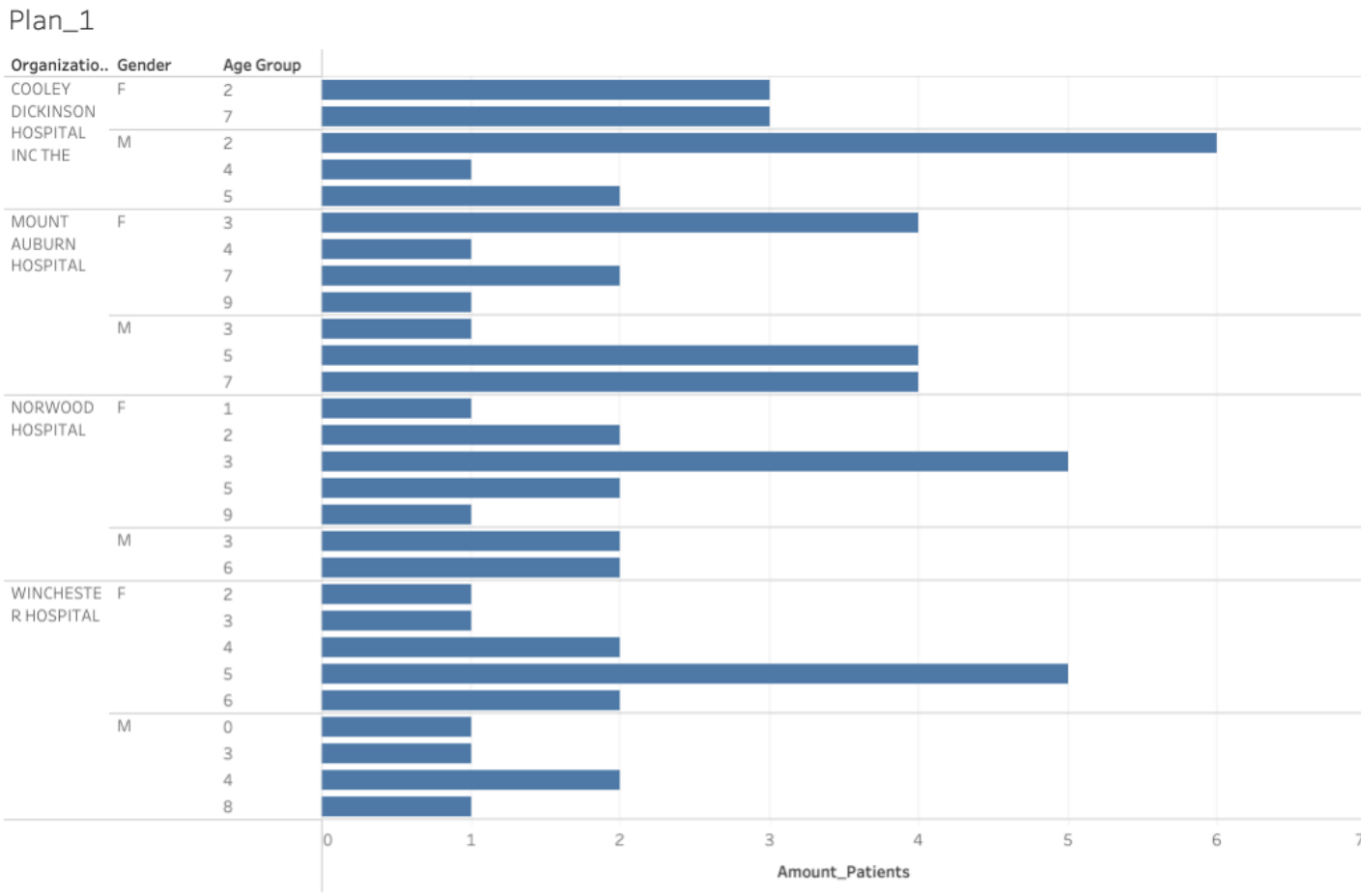

Figure 42: First "Plan" query analyzed on tableau

- **What is the average age of patients affected by acute bronchitis or prediabetes in each county recorded in the last five years?**

The selected dimensions are: Id_Start, Description of the disease and County_Name. The chosen measure is Average_Age_Of_Patients. Figure 42 shows the histogram resulting from the query.

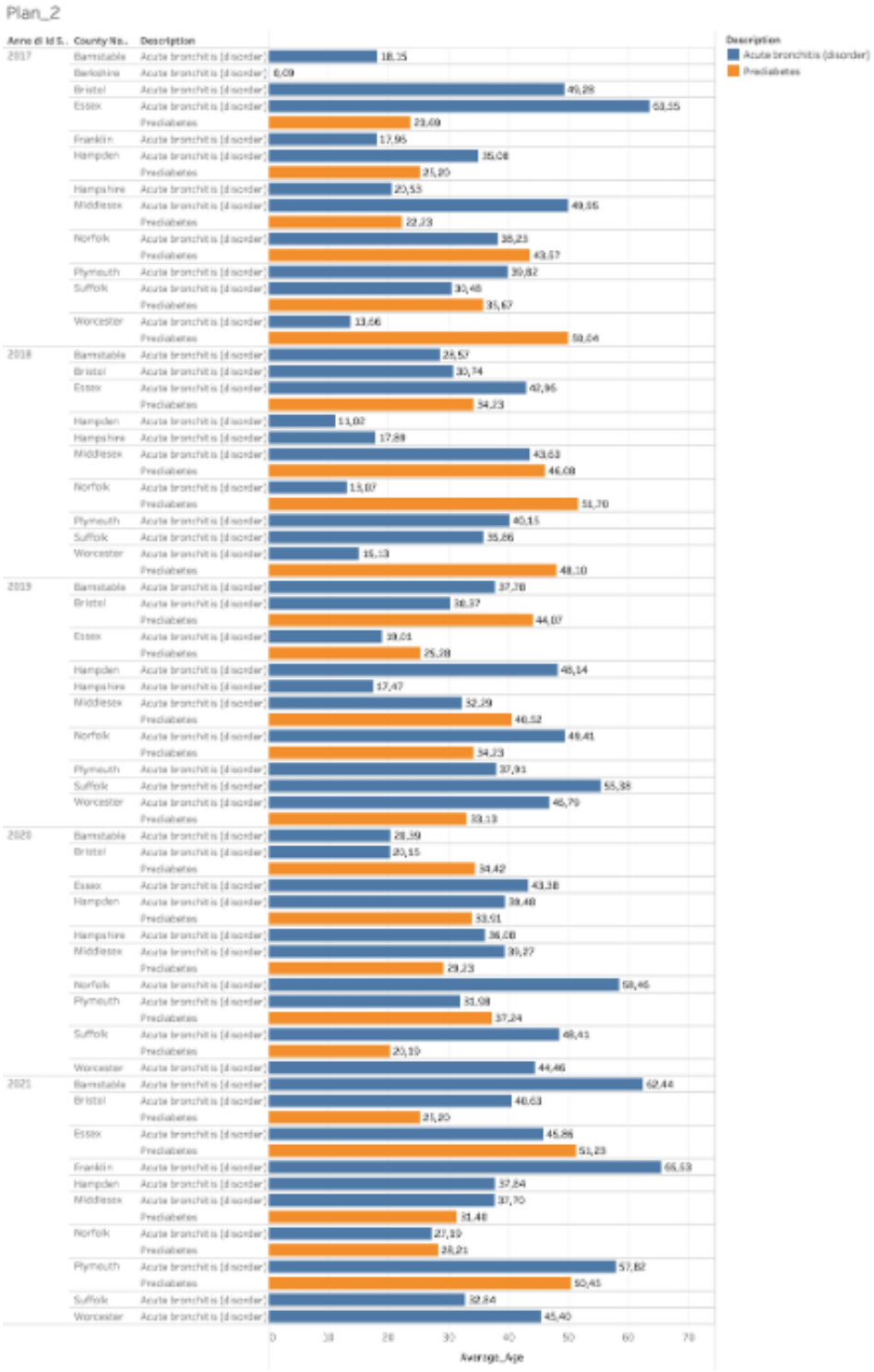

Figure 43: Second "Plan" query analyzed on tableau

- **What is the average number of treatment days required to complete a therapy plan for various diseases in each hospital over the years?**

The selected dimensions are: Description of the careplan, Description of the disease and Year of start of the plan. The chosen measure is Average_Days_Of_Hospitalization. Figure 43 shows the histogram resulting from the query.

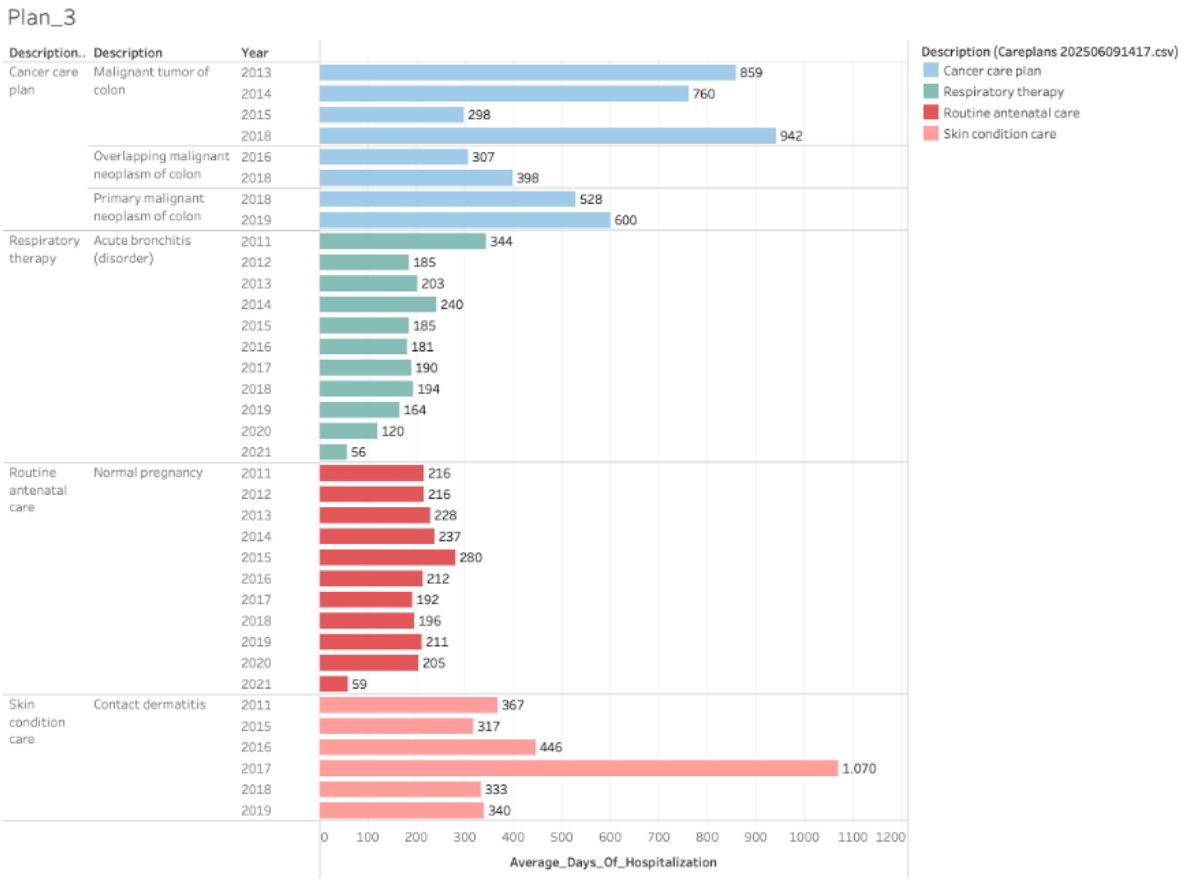

Figure 44: Third "Plan" query analyzed on tableau