

Advanced Data Management

Technologies

*Data Warehouse Project*

Fanti Coelho Lima Rachel

Rachel.FantiCoelhoLima@stud-inf.unibz.it

pORRAS galindo johan leandro

jOHANlEANDRO.PORRASGALINDO@stud-inf.unibz.it

Table of Contents

[ADMT PROJECT 2020 1](#_Toc60923814)

[1 Domain Analysis and Description 2](#_Toc60923815)

[1.1 Overview of the Company and the app 2](#_Toc60923816)

[1.2 Management decisions 2](#_Toc60923817)

[1.3 Motivations for the development of a DW 3](#_Toc60923818)

[1.4 Business Processes to Model 3](#_Toc60923819)

[Sales Process 3](#_Toc60923820)

[Tutoring Process 4](#_Toc60923821)

[Granularity 5](#_Toc60923822)

[2 Conceptual Design 5](#_Toc60923823)

[2.1 Data Warehouse Bus Architecture 5](#_Toc60923824)

[2.2. Dimensional Fact model 6](#_Toc60923825)

[Sales DFM 7](#_Toc60923826)

[Tutoring DFM 11](#_Toc60923827)

[3 Logical Design 12](#_Toc60923828)

[Sales Star Schema 15](#_Toc60923829)

[Tutoring Star Schema 16](#_Toc60923830)

[4 Implementation 16](#_Toc60923831)

[4.1 SQL Scripts 17](#_Toc60923832)

[Course 17](#_Toc60923833)

[Tutor 17](#_Toc60923834)

[Customer 17](#_Toc60923835)

[User Account 18](#_Toc60923836)

[Date 18](#_Toc60923837)

[Fact Tutoring 19](#_Toc60923838)

[Promotion 19](#_Toc60923839)

[Fact Sales 20](#_Toc60923840)

[4.2 Data Extraction, Transformation and Load (ETL) 20](#_Toc60923841)

[Dimensions 21](#_Toc60923842)

[5 Querying 22](#_Toc60923843)

[5.1 GROUP BY extensions: ROLLUP, CUBE and GROUPING SETS. 23](#_Toc60923844)

[5.2 Ranking query using NTILE, RANK or DENSE RANK functions 27](#_Toc60923845)

[5.3 Windowing query using the windowing clause 32](#_Toc60923846)

[5.4 Period-to-period comparison query 34](#_Toc60923847)

[6 Data analisys tool 36](#_Toc60923848)

[6.1 Data analysis tool selection and set up 36](#_Toc60923849)

[Adding data sources 36](#_Toc60923850)

[Dashboard creation 38](#_Toc60923851)

[7 Advanced Data Management 40](#_Toc60923852)

[References 40](#_Toc60923853)

# ADMT PROJECT 2020

*The following pages contain the business case analysis of the implementation of a data warehouse for the company Cambly Inc.*

1 Domain Analysis and Description



### 1.1 Overview of the Company and the app

Cambly is an online platform founded in 2012 that provides on-demand access to tutors via video chat. Cambly tutors are native speakers from the United States, Canada, UK and Australia, etc. The platform allows users to connect with the suitable tutor in under 5 seconds. Users can then practice English conversation, take an English course, or prepare for the IELTS or TOEFL exams.

Every class on Cambly is recorded, and video recordings are available on-demand for the respective users. Cambly provides in-chat translations that allow users to type in their native language in case they get stuck in conversation. Users can book classes according to their schedule with the specific tutors as well. After 10 hours of private English classes on Cambly, every student also gets their own certificate. Cambly provides plans depending on the user's weekly agenda.

On the platform, registered students can video chat with English tutors from different regions with different accents. They browse tutor profiles to find the ones with relevant experience and interests. ​There are available two different services for a monthly fee: tutoring and interactive language courses.

### 1.2 Management decisions

Profit comes from the subscriptions. As it is a business based on a service that has essentially fixed costs (courses, tutors, and bandwidth), management must try to attract as much paying users as possible, without running out of bandwidth and available tutors. ​

The most significant decisions have to do with: ​

* Subscriptions pricing (calibrated according to the different economic regions)
* Promotion to win a constant (possibly increasing) number of paying users and to localize old ones, to ensure a healthy ecosystem and high level of service delivering​
* User satisfaction (improvements, fidelity, platform engagement, client retention etc)
* Bandwidth utilization (how much, in which regions, optimizations based for example on service utilization in different time zones)
* Incentives to increase the number of tutors and to retain the best ones
* Development of new courses and new ideas that attract and retain customers

Management wants to better understand: ​

* What convinces the typical trial users to buy a plan
* The typical user behaviour based on the several dimensions and attributes
* The sales behaviour through different channels
* The sales​ behaviour acrost time
* Identify most valuable customers (Profiling)
* What types of courses attract the most

To support the analysis and future business decisions the business wanted to use the data collected through the web site or the apps (IOS/Android) for analysis. Based on this demand and other similar ones previously, the board of directors decided to evaluate the option of developing a data warehouse.

### 1.3 Motivations for the development of a DW

A DW allow the professionals of the company understand better your market and business, propitiating data-based decision making and increasing the competitive advantage. It supports the managers to better understand the points mentioned previously, identifying users’ needs and new opportunities, allowing the company to respond more quickly to changes.

Moreover, a completer and more consistent store of data and systems with more scalability are required due to the large amount of data involved, the growing increase in users and the continuous evolution of the system. Furthermore, this technology allows the company to host new applications and technologies without redesigning the whole system.

### 1.4 Business Processes to Model

The managers decided to develop a data warehouse, according to the bottom-up approach, allowing the DW being incrementally built by iteratively creating several data marts. Each data mart is based on a set of facts that are linked to specific process. In the first data marts, Sales and Tutoring process were chosen to be modelled and analysed priority.

#### Sales Process

Students can choose between different subscriptions that differ in length (monthly, quarterly, annually), duration of each video chat (15, 30 or 60 minutes per day) and frequency of video chat (2, 3 or 5 days per week). Subscription prices can range from 40€/month to 271€/month. Modeling the sales process the business could potentially answer questions such as:

* What is the average spending per student? ​
* What is the profit per period per student? ​
* Which teachers collect the most? ​
* What is the percentage of sales to natural persons and companies (juridical persons)? ​
* What is the percentage between adults and children? ​
* In which country and continent are there more sales? (countries of origin) ​
* What are the price ranges for services in the main countries and continents? ​
* At what time of the year are there more sales? ​
* Is there a need to create specific price actions for segments of the public (companies, adult, kids, teenagers)? ​
* What is the age group of the main customers? ​
* What is the level of proficiency of the main customers? ​
* From which mean (web site, ads, referral. etc.) the sale was made?

#### Tutoring Process

There are tutors available 24 hours, 7 days in the week. The students can see each tutor’s class schedule and their available time. They can speak with as many tutors as they'd like up until your minutes run out. ​

There are 300,000 students of all ages spread across the world, such as Brazil, Saudi Arabia, China, South Korea, Japan, Russia and Turkey. ​There are more than 3,000 tutors registered who have a wide range of education and professions. ​

The students can search for interesting topics to read or discuss with the tutors if they want. They can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. Modeling the tutoring process the business could potentially answer questions such as:

* What are the top 5 most popular courses chosen by the students? Which kind of material they look for? ​
* What are the top 5 best tutors? ​
* Which is the profile of the best evaluated tutors? ​
* How many tutors are selected by students per year? ​
* How many days a week on average are attended by students? ​
* What is the number of Tutoring sessions per week by User and Teacher? ​
* What is the average duration of Tutoring a session by User and Teacher​?
* At which hours, Tutoring are most requested? ​
* What percentage of classes had connection problems? ​
* How many classes occur simultaneously in the app? ​
* What is the average amount of time a teacher and the students stay on Cambly? For what reasons do users leave the app? ​
* How many days a week on average are offered by teachers? And by the best rated? ​
* There were tutors available to attend students at the requested time and online?

#### Granularity

After analyzing the frequency of the transactional data available and when comparing against the target questions to be answered by our models and data warehouse, we identify that the granularity had to be at the most detailed level.

For the sales data, the individual transaction holds value to our analyzes that if it gets summarized could lost value to the business. Similar situation is found with the Tutoring transactions on which each tutoring session is relevant to determine the user engagement, frequency of the sessions, tutor time etc.

|  |  |
| --- | --- |
| Sales | Sales per customer per tutor per course per promotion per day |
| Tutoring | Tutoring per customer per tutor per course per day |

2 Conceptual Design

### 2.1 Data Warehouse Bus Architecture

The Business Intelligence Competence Center created to develop the data warehouse chose the “Data warehouse Bus Architecture” as a standard interface, since it supports the incremental development of a DW.

The Bus Matrix was used to document the bus Architecture, where the rows represent business processes (to be translated into DMs) and the columns represent a suite of standardized, common, and shared dimensions.

**Bus Matrix**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Date | Customer | Student | Tutor | Course | Promotion | … |
| Sales | X | X |  | X | X |  |  |
| Tutoring | X | X | X |  |  | X |  |
| … |  |  |  |  |  |  |  |

The table 1 - Bus Matrix representing the Sales and Tutoring process and the dimensions.

After that, a conceptual design for the DW was designed using a mixed approach of supply-driven and demand-driven, where requirement and data source analysis were done at the same time. Therefore, the conceptual schema for the model was derived from the reconciled layer.

### 2.2. Dimensional Fact model

A graphical conceptual model for DWs, the **Dimensional Fact model** (DFM), was conceived.

According to (MATTEO GOLFARELLI, 1998) the representation of reality built using the DFM is called dimensional scheme and consists of a set of fact schemes whose basic elements are facts, measures, dimensions, and hierarchies. In the following an intuitive description of these concepts is given:

**Facts** are the concepts on which data mart end users base their decision-making process. In our case study we will focus on the company's main processes and core business: sales and tutoring.

Compatible fact schemes may be overlapped to relate and compare data. In our case, for instance, it can be necessary to analyze how many courses sold where concluded or who were the students and the age group that sold the most.

**The measures** are continuously valued (typically numerical) attributes which describe the fact from different points of view. For example, each sale is measured by the gross sales amount, in dollar**.**

**Dimensions** represent the attributes related to the fact, based on which the facts can be analyzed, sliced and diced, or rolled through different hierarchies. For instance, in our sales fact scheme the dimensions are date, customer, tutor, course, and promotion.

**Conformed dimensions**, identical or strict mathematical subsets of the most granular, detailed dimension are necessary to the point of view of the uniformity of the logical model and that of the flexibility of OLAP operators. This allows the users roll-up dimensions conform to the base-level dimension.

**Hierarchies** are made up of discrete dimension attributes linked by -to-one relationships and determine how facts may be aggregated and selected significantly for the decision-making process. The dimension in which a hierarchy is rooted defines its finest aggregation granularity; the other dimension attributes define progressively coarser granularities (MATTEO GOLFARELLI, 1998). For example, a hierarchy on the course dimension include the attributes course, course type, course subcategory, course category and institution.

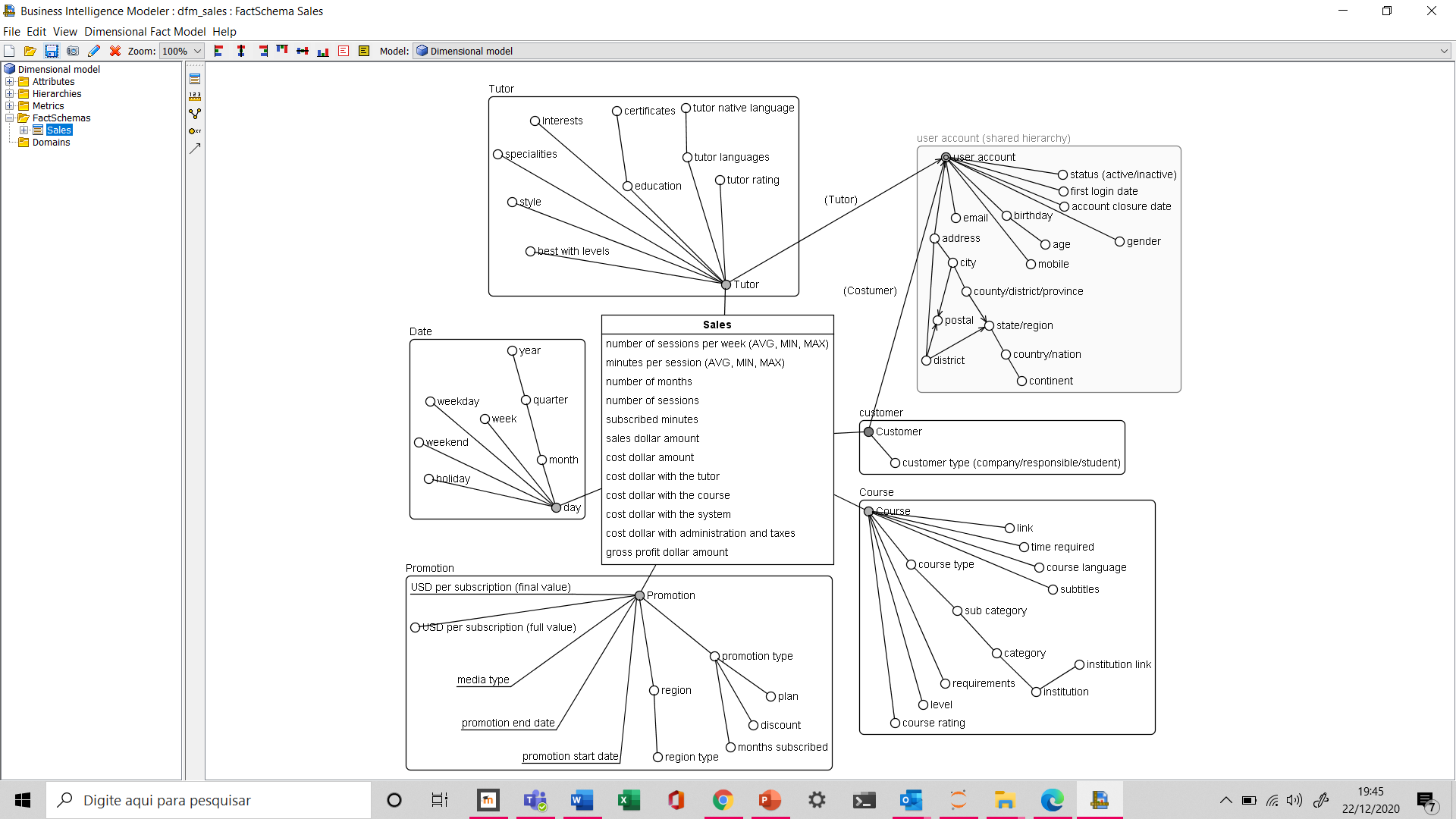
Hierarchies may also include non-dimension attributes (descriptive attributes). A non-dimension attribute contains additional information about a dimension attribute of the hierarchy and is connected by a -to-one relationship (e.g., phone number); unlike dimension attributes, it cannot be used for aggregation (MATTEO GOLFARELLI, 1998).

Other features which are represented on the fact schemes are the additivity of fact attributes along dimensions, the optionality of dimension attributes and the existence of non-dimension attributes.

A measure is additive on a dimension if its values can be aggregated along the corresponding hierarchy by the sum operator. Since this is the most frequent case, to simplify the graphic notation in the DFM, only the exceptions will be specified.

In the following we will discuss the graphic representation of the concepts introduced above with reference to the fact schemes sales and tutoring.

### Sales DFM



**Sales fact**

In the sale scheme, each primary instance describes a customer’s subscription (sale) to one course adopting one promotion on a certain date, with one tutor. No promotion' should be considered as a particular case of promotion.

**Measures**

Measures of the sales fact:

* number of sessions per week;
* minutes per session;
* number of months;
* number of sessions (number of sessions per week \* weeks per month\* number of months)
* minutes subscribed (minutes per session \* total sessions contracted);
* salles dollar amount (already with the discounts);
* cost dollar amount (including costs with the tutor, the course, the system maintenance, administrative costs and taxes);
* gross profit dollar amount (salles – cost).

**Additivity**

As we wrote before, a measure is additive if its values can be aggregated along the corresponding hierarchy by the sum of operator.

In our conceptual scheme almost all measures of our sales fact are additive.

However, there are two non-additive measures, that can be misinterpreted if analyzed in an aggregate way. These kinds of measures can still be aggregated by using operators such as average, maximum, minimum.

Non additive measures:

* number of sessions per week;
* minutes per session;

For aggregate analysis we can use the columns number of sessions and minutes subscribed.

**Dimensions**

In the following we will present the dimensional attributes for each dimension.

**Tutor:**

Attributes related to tutor:

* tutor average rating;
* languages that they are proficient in;
* native language;
* education (graduation, master’s, doctorate, proficiency exams);
* certificates (graduation, master’s, doctorate, IELTS, TOEFL, PTE Academic, CAE, etc);
* interests: science and technology, language and culture, world news, entertainment and lifestyle, business, sports, literature;
* Specialties: accent coaching, business, friendly conversation, grammar instructions, tests preparatory (e.g. IELTS, TOEFL);
* style: kind and patient, fun and gregarious, scholarly and knowledgeable;
* best with levels (beginner, elementary, conversational, intermediate, advanced).

**Customer:**

Attributes related to customer:

* customer type: a company, a responsible person for one or more kids or a student;

**User account (for tutor and customer)**

Attributes related to user account:

* status (active/inactive) - when more than 1 month without activity;
* first login date;
* account closure date – only when the user delete his account from the system;
* gender;
* birthday and age;
* mobile; email;
* address, city or district, postal, county/district/province, state/region, country/nation, continent.

**Course:**

Attributes related to course:

* course rating;
* link to the course;
* time required to do the course;
* course language and subtitles;
* course type, sub category, category, institution that elaborated the course and link for the institution
* course requirements;
* course level.

**Promotion:**

Attributes related to promotion:

* type (e.g. “buy a quarter plan and get more 2 months free”);
* plan subscribed in that transition (month, quarter, year);
* percentage of discount;
* number of months subscribed (plan + free month);
* type of region of the promotion (region, country, continent);
* region (e.g. Asia, Italy);
* promotion start and end date;
* media used to advertise the promotion;
* USD per subscription (full value) – without the discounts;
* USD per subscription (final value): USD per subscription (full value) \* (1-percentage of discount)

Attributes related to promotion: name, type (e.g. “buy a quarter plan and get more 2 months free”), plan subscribed in that transition (month, quarter, year), percentage of discount, number of month subscribed (plan + free month), region type of the promotion (region, country, continent), region (e.g. Asia, Italy), promotion start and end date, media used, number of users that accepted the promotion, USD amount subscribed in each promotion (without the discounts), USD discounted values, promotion USD Net subscribed (USD amount subscribed – discounts).

**Date:**

Attributes related to date:

* day, month, quarter, year;
* week, weekday, weekend and holyday.

**Hierarchies:**

Hierarchies are used to represent relationships between dimensional attributes and non dimensional attributes (descriptive attributes). A hierarchy is a directed tree, nodes are the attributes and arcs model the associations between the attributes (many-to-one when dimensional attribute or one to one when non-dimensional).

For example, a hierarchy on the date dimension include the dimensional attributes day, month, quarter, year.

The descriptive attributes store additional information about dimensional attributes. Usually, they are not used for aggregation and are always leave node. (e.g, promotion star date, promotion end date and media used to advertise the promotion).

**Shared hierarchies**

In our Fact Sales there is an example of shared hierarchy. Customer and tutor have the same user account hierarchy structure, with attributes such as user name, birthday, mobile, email, address, etc. Note that our notation is a little different from the notation used in the classroom because of the system used.

**Multiple Arcs**

A multiple arc models a many-to-many association between two dimensional attributes (and not many-to-one).

In our case there are some cases of many to many association.

**Optional arcs**

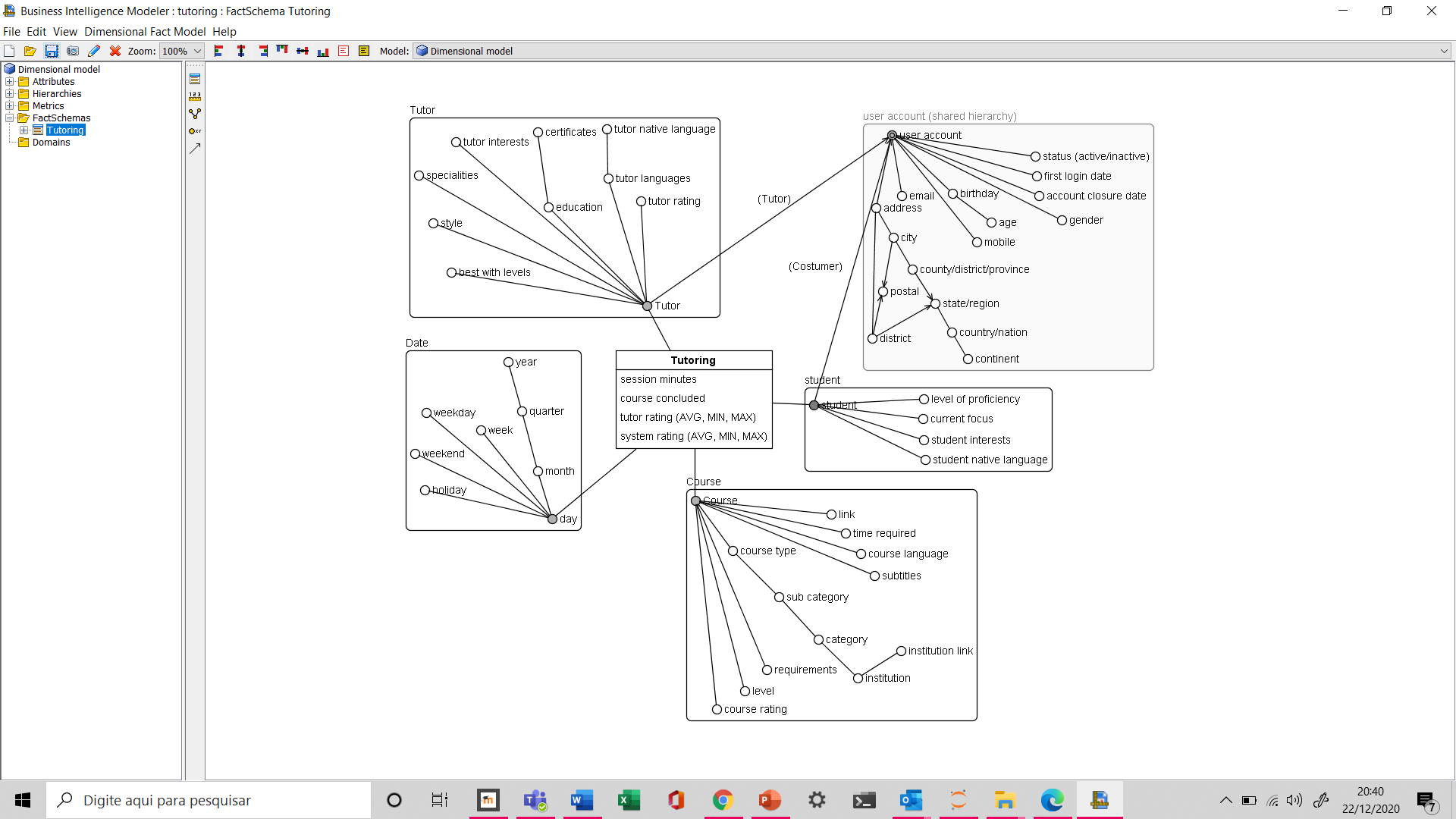
There are some attributes that are optional. In the tutor dimension, the attribute of certificates is optional since tutors can include or not certificates that they have from the recognized institutions.

The attribute account closure date should be written only if a customer cancel their account and left the system.

**Convergence**

The user account hierarchy is incomplete (or ragged) for the city branch, as different nations have different ways of dividing the country into administrative units. Large countries, such as the U.S. have a very thorough subdivision, while e.g. a microstate in contrast is very flat in this regard. There is convergence between the sales district on one and the city and county attributes on the other side, for they both determine the state/region attribute (when existing) and country.

### Tutoring DFM



We will be more succinct in explaining this scheme, since some dimensions are repeated or have similar characteristics to those mentioned in the sales scheme. Thus, we will focus only on the main aspects and complementary explanation of this view.

**Tutoring Fact**

In the tutoring scheme, each primary instance describes one course’s session with one student and one tutor in a date.

**Measures of the tutoring fact**

* session minutes
* course concluded
* tutor rating
* system rating

**Additivity**

The tutor rating and system rating cannot be aggregate as a sum. These are non additive measures and can still be aggregated by using operators such as average, maximum, minimum.

**Dimensions**

**Student:**

Attributes related to student:

* level of proficiency: beginner, elementary, conversational, intermediate, advanced
* current focus: exam preparation, professional development, academic course, foreign travel, self improvement, other
* Interests: science and technology, language and culture, world news, entertainment and lifestyle, business, sports, literature
* Native language

**Course:** already descript in the sales scheme

**Tutor:** already descript in the sales scheme

**Date:** already descript in the sales scheme

## 3 Logical Design



The logical design transforms the conceptual schema for a DM into a logical schema.

The star schema and snowflake schema are the most common types of logical schema. A star schema is a dimensional model with fully denormalised hierarchies, while a snowflake schema is a dimensional model with fully normalised hierarchies.

Cambly experts of the center of intelligence decided to use the start schema because although there are more data redundancy and the hierarchies are hidden, this solution simplifies the structure and the queries, reducing the number of tables and joins between tables and has a better performance. The user account data was in a separate table, not being together with the client and tutor dimension.

**Dealing with non-hierarchical data**

One of the main complexities in dimensional modelling is dealing with non-hierarchically structured data. Dimensional models assume an underlying hierarchical structure and therefore exclude data that is naturally non-hierarchical. Therefore, it is important to evaluate how to handle the stored data in the form of many-to-many relationships.

|  |  |
| --- | --- |
| Many-to-many relationships | |
| Dimension | Attributes |
| Tutor | Tutor language  Tutor education  Certificates  Interests  Specialties  Teaches for levels |
| Course | Course subtitles  Course requirements |
| Student | Students interests |

The first analysis carried out was to identify whether these information are significative important for decision making and try to simplify.

**Tutor - Tutor language and Tutor - Teaches for levels:**

We will create bridge tables to represent the n to n relationship between tutor and the language taught by teachers. This table will inform the priority of teachers for each language (first language, second, third, etc.) and the level at which they are able to teach (beginner, elementary, conversational, intermediate, advanced).

The detailed information of these data is necessary since with this data we can evaluate if there are sufficient tutors to cover the student’s interest in languages, according to the required level, and act to avoid a lack of teachers who have this qualification.

**Tutor - Tutor interests, Tutor – Specialties and Student – Student interests:** these data will not be included at first because they are not relevant for analysis. A bridge table can be included in the future if it is necessary.

Creating the data warehouse in stages can greatly reduce company costs and time for the team evolved, avoiding monstrous tables, giving time for the professionals evaluate the real need for some fields.

**Tutor – education and Tutor - certificates:** we created 2 columns as a solution. One with the higher level of the tutor education (graduation, master’s or doctorate) and the second informing if the tutor has a proficiency certificate in an institution recognized. The complete information can be relevant for the student that look for the teacher in the system, but it does not have to be so detailed for business analysis. It is also interesting to know if the level of education obtained is in the language taught, but currently there is no such level of detail.

In practice, some attributes can have several possibilities of values, which makes the extraction of conclusive analyses hard. To address this issue, we can think of a summary capability solution, as for example creating groups of the attributes. In this case, we transformed the relation n to n in 2 groups (columns). Another solution could have been a mini-dimension table with some subsets, however, probably in this case would be necessary also a bridge table.

**Course - Course subtitles, Course – Course requirements:** We will create bridge tables for course requirements and subtitles as these data can be relevant to identify the need for other courses and languages.

All bridge tables will be included initially only in the tutoring scheme and will not be included in the sales scheme as they are not so relevant for sales analysis.

**Time-dependent Relationships (Historical)**

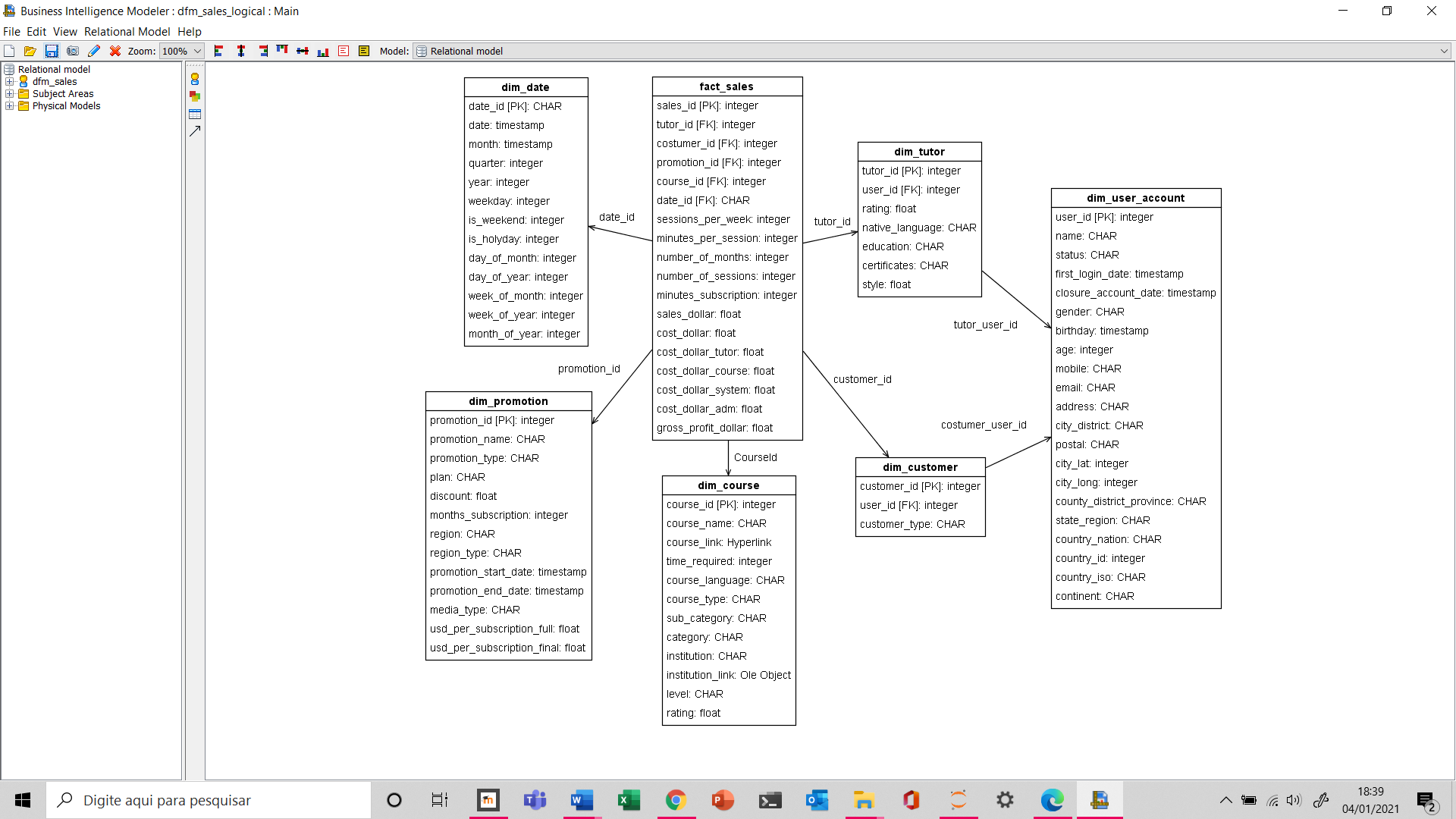
A special problem which occurs very commonly in data warehousing is that the values of the attributes and the relations can vary over the time. Consequently, it is necessary to evaluate how to deal with these changes. Some alternatives are overwriting the old values that change in the dimension tables or creating aslowly changing dimension. In the latter, a new row is created each time the underlying component entity changes some important characteristic. The purpose of this is to record the state of the dimension entity at the time each transaction took place.

For the attributes of customer, tutor and student, the old values will be overwritten. In the course dimension the attributes link for the course, time required, subtitles and requirements are overwritten too for the new values.

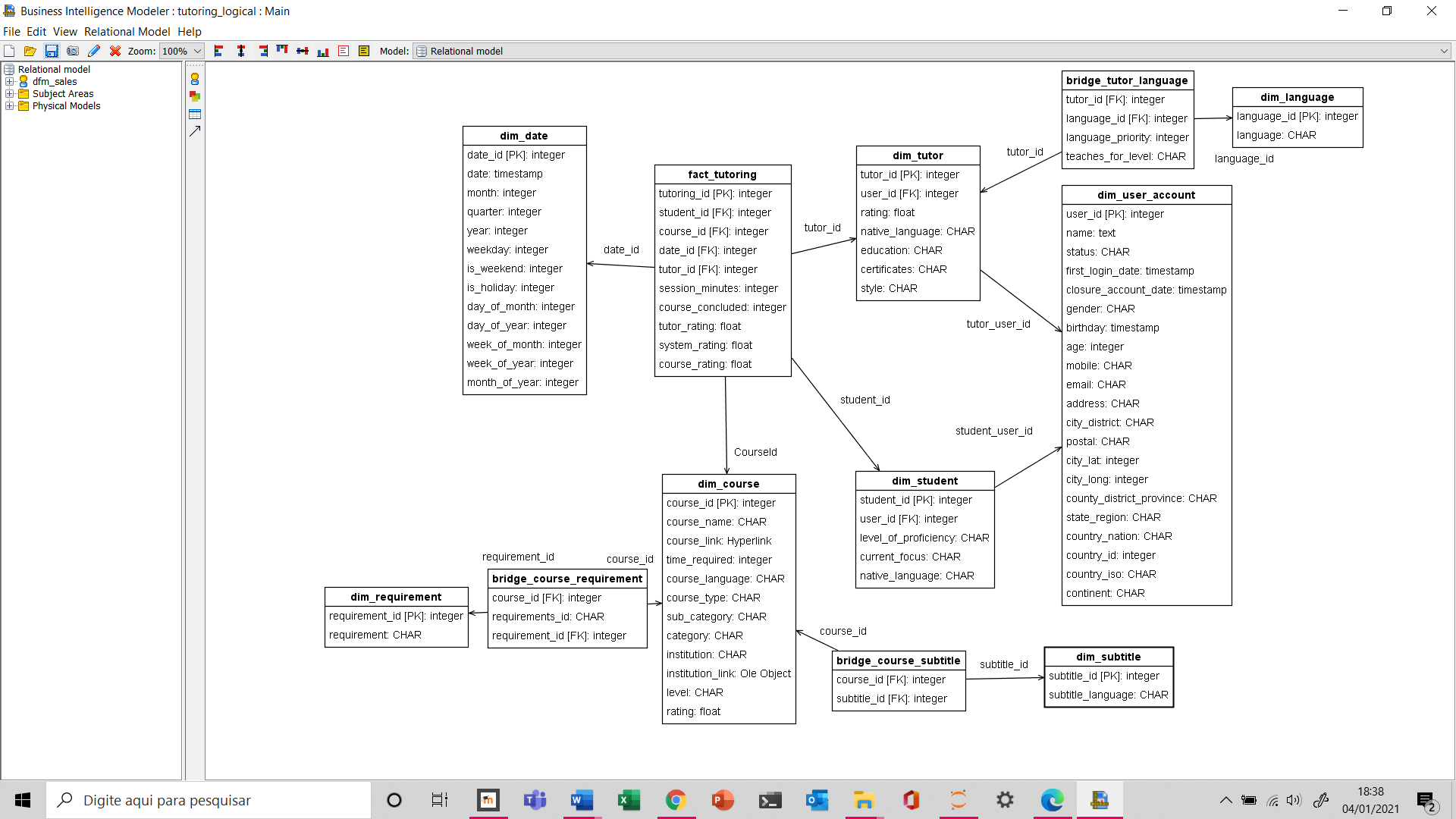
For promotion and the other attributes related to course, a new row will be added with the new version of the attributes.

In the following, the star schemas for sales and tutoring are depicted.

### Sales Star Schema



### Tutoring Star Schema



The abbreviations (FK) and (PK) after the attributes stand for foreign key and primary key, respectively.

Although a star schema was chosen, the last schema depicted below looks like a snowflake schema. This is due to the bridge tables, that were used to map multiple arcs. As multiple arcs could be ”unfold”, this would considerably increase the size of the dimension tables.

1. Implementation

To implement and exemplify the queries a more simplified database will be generated. The customer and student dimensions will be unified, and we will not use bridge tables.

### 4.1 SQL Scripts

To achieve the goal proposed in the Logical design chapter, the following SQL Scripts were implemented for each of the dimensions and facts proposed and explained on earlier chapters:

#### Course

CREATE TABLE public.dim\_course

(

course\_id BIGSERIAL

, course\_name VARCHAR(132)

, course\_type VARCHAR(14)

, time\_required NUMERIC(97, -1)

, course\_language VARCHAR(22)

, course\_subtitles VARCHAR(271)

, course\_skills VARCHAR(148)

, course\_rating BIGINT

, category VARCHAR(32)

, sub\_category VARCHAR(40)

, course\_level VARCHAR(115)

)

;

CREATE INDEX idx\_dim\_course\_lookup ON public.dim\_course(course\_id)

;

CREATE INDEX idx\_dim\_course\_tk ON public.dim\_course(course\_id)

;

ALTER TABLE public.dim\_course ADD PRIMARY KEY (course\_id)

;

#### Tutor

CREATE TABLE public.dim\_tutor

(

tutor\_id INTEGER PRIMARY KEY

, user\_id BIGINT

, country\_id VARCHAR(2)

, first\_teaching\_language VARCHAR(17)

, first\_teaching\_level BIGINT

, second\_teaching\_language VARCHAR(18)

, second\_teaching\_level BIGINT

, total\_sessions BIGINT

, signed\_up\_date BIGINT

)

;

CREATE INDEX idx\_dim\_tutor\_lookup ON public.dim\_tutor(tutor\_id)

;

CREATE INDEX idx\_dim\_tutor\_tk ON public.dim\_tutor(tutor\_id)

;

ALTER TABLE public.dim\_tutor ADD PRIMARY KEY (tutor\_id)

;

ALTER TABLE public.dim\_tutor ADD CONSTRAINT user\_id FOREIGN KEY (user\_id) REFERENCES public.dim\_user\_account (user\_id)

#### Customer

CREATE TABLE public.dim\_customer

(

customer\_id BIGSERIAL

, user\_id BIGSERIAL

, customer\_type VARCHAR (14)

, CONSTRAINT "CostumerId" PRIMARY KEY (customer\_id)

, CONSTRAINT user\_id FOREIGN KEY (user\_id)

REFERENCES public.dim\_user\_account (user\_id) MATCH SIMPLE

)

;

CREATE INDEX idx\_dim\_customer\_lookup ON public.dim\_customer(customer\_id)

;

CREATE INDEX idx\_dim\_customer\_tk ON public.dim\_customer(customer\_id)

;

#### User Account

CREATE TABLE public.dim\_user\_account

(

user\_id BIGSERIAL

, name TEXT

, gender TEXT

, age DOUBLE PRECISION

, mobile TEXT

, email TEXT

, address TEXT

, postal\_zip TEXT

, city TEXT

, city\_lat DOUBLE PRECISION

, city\_long DOUBLE PRECISION

, state TEXT

, country TEXT

, country\_id DOUBLE PRECISION

, country\_iso\_2 TEXT

, sub\_region TEXT

, continent TEXT

, bithday\_date BIGINT

)

;

CREATE INDEX idx\_dim\_ user\_account\_lookup ON public.dim\_ user\_account(user\_id

;

CREATE INDEX idx\_dim\_user\_account\_tk ON public.dim\_user\_account(user\_id)

;

ALTER TABLE public. dim\_user\_account ADD PRIMARY KEY (user\_id)

;

#### Date

CREATE TABLE public.dim\_date

(

date\_id BIGSERIAL

, date TIMESTAMP

, month\_year TIMESTAMP

, quarter NUMERIC(18, 2)

, year DOUBLE PRECISION

, day DOUBLE PRECISION

, month DOUBLE PRECISION

, day\_of\_week DOUBLE PRECISION

, week\_of\_month DOUBLE PRECISION

, day\_of\_year DOUBLE PRECISION

, week\_of\_year DOUBLE PRECISION

, is\_weekend TEXT

, is\_holiday TEXT

, name\_holiday TEXT

)

;

CREATE INDEX idx\_dim\_date\_lookup ON public.dim\_date (date\_id)

;

CREATE INDEX idx\_dim\_date\_tk ON public.dim\_date(date\_id)

;

ALTER TABLE public.dim\_date ADD PRIMARY KEY (date\_id)

;

#### Fact Tutoring

CREATE TABLE public.fact\_tutoring

(

tutoring\_id BIGINT

, tutor\_id BIGINT PRECISION

, customer\_id BIGINT PRECISION

, course\_id BIGINT PRECISION

, date\_id BIGINT PRECISION

, session\_minutes DOUBLE PRECISION

, course\_concluded INTEGER PRECISION

, tutor\_rating DOUBLE PRECISION

, system\_rating DOUBLE PRECISION

)

CREATE INDEX idx\_ fact\_tutoring\_lookup ON public. fact\_tutoring (tutoring\_id)

;

CREATE INDEX idx\_fact\_tutoring\_tk ON public.fact\_tutoring (tutoring\_id)

;

ALTER TABLE public.fact\_tutoring ADD PRIMARY KEY (tutoring\_id)

;

ALTER TABLE public.fact\_tutoring ADD CONSTRAINT tutorfk FOREIGN KEY (tutor\_id) REFERENCES public.dim\_tutor (tutor\_id)

;

ALTER TABLE public.fact\_tutoring ADD CONSTRAINT customerfk FOREIGN KEY (user\_id) REFERENCES public.dim\_user\_account (user\_id)

;

ALTER TABLE public.fact\_tutoring ADD CONSTRAINT coursefk FOREIGN KEY (course\_id) REFERENCES public.dim\_course (course\_id)

;

ALTER TABLE public.fact\_tutoring ADD CONSTRAINT datefk FOREIGN KEY (date\_id) REFERENCES public.dim\_date (date\_id)

#### Promotion

CREATE TABLE public.dim\_promotion

(

promotion\_id BIGSERIAL

, promotion\_name VARCHAR (100)

, promotion\_type VARCHAR (100)

, discount DOUBLE PRECISION

, month\_subscribed INTEGER

, region VARCHAR (50)

, region\_type VARCHAR (50)

, promotion\_start\_date TIMESTAMP

, promotion\_end\_date TIMESTAMP

, media\_type CHAR (50)

, usd\_per\_subscription\_full DOUBLE PRECISION

, usd\_per\_subscription\_final DOUBLE PRECISION

CONSTRAINT "PromotionId" PRIMARY KEY (promotion\_id)

)

;

CREATE INDEX idx\_dim\_promotion\_lookup ON public.dim\_promotion(promotion\_id)

;

CREATE INDEX idx\_dim\_promotion\_tk ON public.dim\_promotion(promotion\_id)

#### Fact Sales

CREATE TABLE public.fact\_sales

(

sales\_id BIGINT

, date\_id BIGINT PRECISION

, customer\_id BIGINT PRECISION

, promotion\_id BIGINT PRECISION

, course\_id BIGINT PRECISION

, tutor\_id BIGINT PRECISION

, minutes\_subscribed INTEGER

, sales\_dollar DOUBLE PRECISION

, cost\_dollar DOUBLE PRECISION

, gross\_profit\_dollar\_amount DOUBLE PRECISION

, CONSTRAINT sales\_id PRIMARY KEY (sales\_id)

, CONSTRAINT costumer\_id FOREIGN KEY (customer\_id)

REFERENCES public.dim\_customer (customer\_id) MATCH SIMPLE

, CONSTRAINT course\_id FOREIGN KEY (course\_id)

REFERENCES public.dim\_course (course\_id) MATCH SIMPLE

, CONSTRAINT date\_id FOREIGN KEY (date\_id)

REFERENCES public.dim\_date (date\_id) MATCH SIMPLE

, CONSTRAINT promotion\_id FOREIGN KEY (promotion\_id)

REFERENCES public. dim\_promotion (promotion\_id) MATCH SIMPLE

, CONSTRAINT tutor\_id FOREIGN KEY (tutor\_id)

REFERENCES public. dim\_tutor (tutor\_id) MATCH SIMPLE

;

CREATE INDEX idx\_fact\_sales \_lookup ON public.fact\_sales (sales\_id)

;

CREATE INDEX idx\_fact\_sales\_tk ON public.fact\_sales (sales\_id)

;

### 4.2 Data Extraction, Transformation and Load (ETL)

Much of the data was generated, based on data obtained from several different sources.

To populate our data warehouse, business data was provided in the form of RAW csv and xlsx files:

* Course\_catalog.csv
* User\_account.csv
* Customer.csv
* Promotion.csv
* Tutor.csv
* Date.csv
* Sales.csv
* Tutoring.csv

To properly ingest this data into our data warehouse we opted to create ETL transformation pipelines with the Pentaho Data integration Tool. With Pentaho Data Integration (PDI), organizations can access data from complex and heterogeneous sources and blend it with existing relational data to produce high-quality, ready-to-analyze information — all without writing a line of code. A rich graphical user interface paired with a powerful multithreaded transformation engine offers high-performance ETL (extract, transform and load) capabilities that cover all data integration needs, including big data ingestion and processing (Hitachi, 2020). This provided a great interface to load and manipulate the data to fit the needs of the logical design proposed.

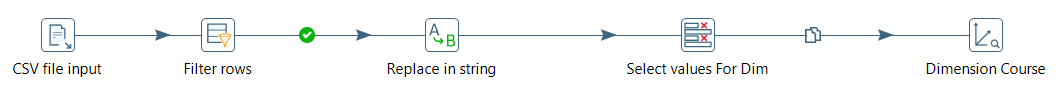
#### Dimensions

The typical ETL workflow that we used was to

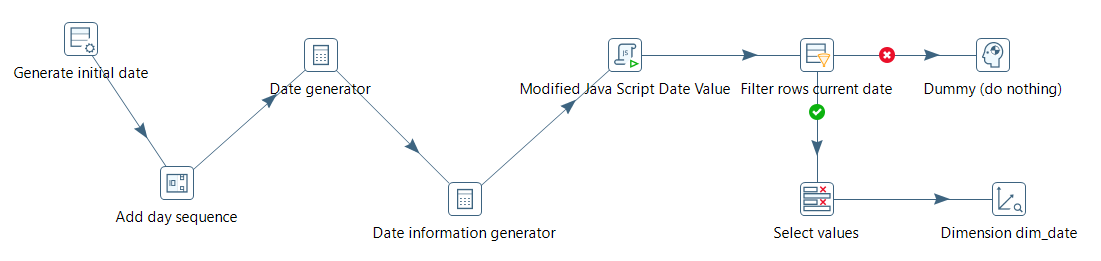
* Import the raw data
* Filter unnecessary rows
* Modify data if needed (Replacing wrong characters i.e)
* Loading data to the Dimension table on the data warehouse
* Optionally if the dimension had a relationship with an existing other dimension Join/Merge operation were added.

This above-mentioned implementation led to the creation of the following Pentaho data integration pipelines:

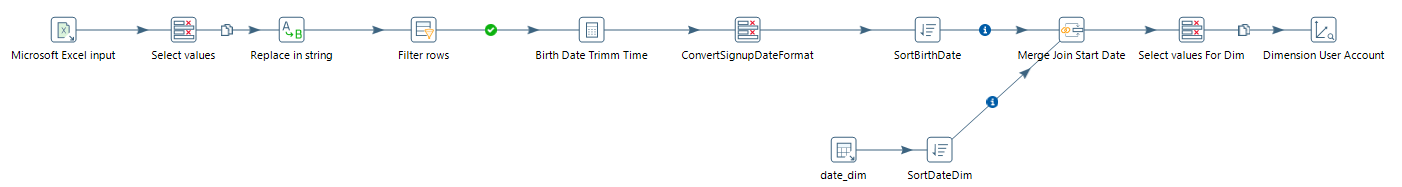
* Course dimension



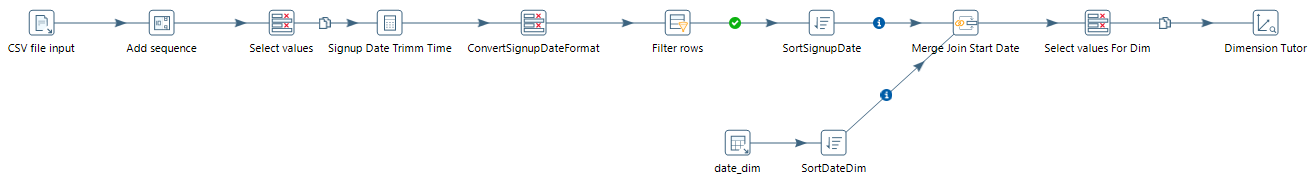
* Date dimension



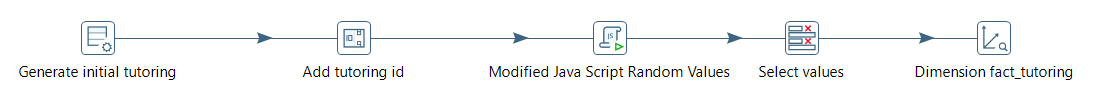
* User Account dimension



* Tutor dimension



* Fact tutoring



Another part of the data, such as promotion, customer, and sales, after having their tables created according to the script, were directly imported in PgAdmin.

To do this, it was important to validate that:

* all data were in the same data type format informed in the script.
* there were no duplicates, neither null in the primary key fields.
* the foreign keys did in fact exist in the source tables.
* it was necessary to remove character formatting settings to distinguish thousands.
* observe the proper format for importing dates.

5 Querying



In this chapter we will describe queries performed with the following functions:

* GROUP BY extensions: ROLLUP, CUBE, GROUPING SETS
* Analytic functions: Ranking and percentiles, Reporting and Windowing

### 5.1 GROUP BY extensions: ROLLUP, CUBE and GROUPING SETS.

The area responsible for the courses monthly extracts relevant information from the system to better understand the total sales for each category, subcategory, and level of the courses.

**ROLLUP**

|  |
| --- |
| **Natural language** |
| Evaluate for the last month, the sales amount and the minutes subscribed, according to the category, subcategory and level of the courses.  Filters: month (11/2020), categories (‘Language Learning”)  Computes and combines the 3 groupings:   * category, subcategory, and level of the courses * category, subcategory * category |
|  |
| **SQL codes** |
| select case when grouping (category)=1 then '\*' else category end,  case when grouping (sub\_category)=1 then '\*' else sub\_category end,  case when grouping (course\_level)=1 then '\*' else course\_level end,  sum (sales\_dollar) as sales\_dollar, sum (minutes\_subscribed) as minutes\_subscribed,  grouping (category, sub\_category, course\_level) as csl  from fact\_sales join dim\_course using (course\_id) join dim\_date using (date\_id)  where year = 2020  and month in (11)  and category in ('language learning')  group by category, rollup (sub\_category, course\_level) |
|  |
| **Results** |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **category** | **sub\_category** | **course\_level** | **sales\_dollar** | **minutes\_subscribed** | **csl** | | Language Learning | Learning English | Advanced Level | 27000 | 9000 | 0 | | Language Learning | Learning English | Beginner Level | 72810 | 24270 | 0 | | Language Learning | Learning English | Intermediate Level | 337290 | 112430 | 0 | | Language Learning | Learning English | not-mentioned | 126540 | 42180 | 0 | | Language Learning | Learning English | \* | 563640 | 187880 | 1 | | Language Learning | Other Languages | Advanced Level | 10260 | 3420 | 0 | | Language Learning | Other Languages | Beginner Level | 198360 | 66120 | 0 | | Language Learning | Other Languages | Intermediate Level | 120420 | 40140 | 0 | | Language Learning | Other Languages | not-mentioned | 88740 | 29580 | 0 | | Language Learning | Other Languages | \* | 417780 | 139260 | 1 | | Language Learning | \* | \* | 981420 | 327140 | 3 | |

Another relevant analysis is to understand what are the medias that sell more in each country over time. Knowing the relationship of each culture to each type of media and the trend of their use over time can generate much more sales.

**CUBE**

|  |
| --- |
| **Natural language** |
| Evaluate the sales amount and the minutes\_subscribed, according to the sales month, customer country and media type used for the promotion.  Filtering: month (08, 09 and 10/2020), media type (‘Facebook’, ‘Instagram’ and ‘Google’), countries (Brazil, Colombia, and Italy)  To calculate all possible combinations of aggregations. |
|  |
| **SQL codes** |
| select case when grouping (month)=1 then '0' else month end,  case when grouping (country)=1 then '\*'else country end,  case when grouping (media\_type)=1 then '\*'else media\_type end,  sum(sales\_dollar) as sales\_dollar, sum (minutes\_subscribed) as minutes\_subscribed,  grouping (month, media\_type, country) as MMC  from fact\_sales join dim\_promotion using (promotion\_id) join dim\_date using (date\_id)join dim\_customer using (customer\_id) join dim\_user\_account using (user\_id)  where year=2020  and month in (8,9,10)  and media\_type in ('Facebook', 'Instagram’, 'Google')  and country in ('Brazil', 'Colombia', 'Italy')  group by cube (month, country, media\_type)  order by month, media\_type, country |
|  |
| **Results** |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **month** | **country** | **media\_type** | **sales\_dollar** | **minutes\_subscribed** | **mmc** | | 0 | \* | \* | 146730 | 48910 | 7 | | 0 | Brazil | \* | 124050 | 41350 | 6 | | 0 | Colombia | \* | 7740 | 2580 | 6 | | 0 | Italy | \* | 14940 | 4980 | 6 | | 0 | \* | Facebook | 39960 | 13320 | 5 | | 0 | Brazil | Facebook | 35280 | 11760 | 4 | | 0 | Italy | Facebook | 4680 | 1560 | 4 | | 0 | \* | Google | 88350 | 29450 | 5 | | 0 | Brazil | Google | 73050 | 24350 | 4 | | 0 | Colombia | Google | 5040 | 1680 | 4 | | 0 | Italy | Google | 10260 | 3420 | 4 | | 0 | \* | Instagram | 18420 | 6140 | 5 | | 0 | Brazil | Instagram | 15720 | 5240 | 4 | | 0 | Colombia | Instagram | 2700 | 900 | 4 | | 8 | \* | \* | 62790 | 20930 | 3 | | 8 | Brazil | \* | 54150 | 18050 | 2 | | 8 | Colombia | \* | 2700 | 900 | 2 | | 8 | Italy | \* | 5940 | 1980 | 2 | | 8 | \* | Facebook | 19800 | 6600 | 1 | | 8 | Brazil | Facebook | 16560 | 5520 | 0 | | 8 | Italy | Facebook | 3240 | 1080 | 0 | | 8 | \* | Google | 33450 | 11150 | 1 | | 8 | Brazil | Google | 30750 | 10250 | 0 | | 8 | Italy | Google | 2700 | 900 | 0 | | 8 | \* | Instagram | 9540 | 3180 | 1 | | 8 | Brazil | Instagram | 6840 | 2280 | 0 | | 8 | Colombia | Instagram | 2700 | 900 | 0 | | 9 | \* | \* | 34980 | 11660 | 3 | | 9 | Brazil | \* | 25980 | 8660 | 2 | | 9 | Italy | \* | 9000 | 3000 | 2 | | 9 | \* | Facebook | 10440 | 3480 | 1 | | 9 | Brazil | Facebook | 9000 | 3000 | 0 | | 9 | Italy | Facebook | 1440 | 480 | 0 | | 9 | \* | Google | 19980 | 6660 | 1 | | 9 | Brazil | Google | 12420 | 4140 | 0 | | 9 | Italy | Google | 7560 | 2520 | 0 | | 9 | \* | Instagram | 4560 | 1520 | 1 | | 9 | Brazil | Instagram | 4560 | 1520 | 0 | | 10 | \* | \* | 48960 | 16320 | 3 | | 10 | Brazil | \* | 43920 | 14640 | 2 | | 10 | Colombia | \* | 5040 | 1680 | 2 | | 10 | \* | Facebook | 9720 | 3240 | 1 | | 10 | Brazil | Facebook | 9720 | 3240 | 0 | | 10 | \* | Google | 34920 | 11640 | 1 | | 10 | Brazil | Google | 29880 | 9960 | 0 | | 10 | Colombia | Google | 5040 | 1680 | 0 | | 10 | \* | Instagram | 4320 | 1440 | 1 | | 10 | Brazil | Instagram | 4320 | 1440 | 0 | |  |  |  |  |  |  | |  |  |  |  |  |  | |

**GROUPING SETTINGS**

The managers also want to know the number of students per course and level over time. Understanding which courses are most in-demand and whether they are well evaluated is also an analysis relevant to the business strategy.

|  |
| --- |
| **Natural language** |
| Evaluate for each month, course and level, the number of students per course, the session minutes of each student, and the system evaluation by students.  Filters: month (second quarter of 2020 – 04, 05 and 06/2020), only courses with more than 1 student  Computes and combines the following grouping sets:   * month, course name, course level * month, course name * month |
|  |
| **SQL codes** |
| select  case when grouping (month)=1 then 0 else month end,  case when grouping (course\_name)=1 then '\*' else course\_name end,  case when grouping (course\_level)=1 then '\*' else course\_level end,  sum (count\_students) as count\_students, sum (session\_minutes) as session\_minutes, avg (system\_rating) as avg\_system\_rating,  grouping (month, course\_name, course\_level) as MCL  from (select month, course\_name, course\_level, count (customer\_id)as count\_students,  sum (session\_minutes) as session\_minutes, avg (system\_rating) as system\_rating  from (select month, course\_name, course\_level, customer\_id, sum (session\_minutes) as session\_minutes, avg (system\_rating) as system\_rating  from fact\_tutoring join dim\_date using (date\_id)join dim\_course using (course\_id)  where year in (2019)  and month in (4,5,6)  group by month, course\_name, course\_level, customer\_id) as query1  group by month, course\_name, course\_level) as query2    where count\_students>1  group by grouping sets ((month, course\_name, course\_level),  (month, course\_name),  (month)) |
|  |
| **Results** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **month** | **course\_name** | **course\_level** | **count\_ students** | **session\_ minutes** | **avg\_system\_ rating** | **mcl** | | 4 | Business Russian Communication. Part 2 | Intermediate Level | 2 | 115 | 2.459101163 | 0 | | 4 | Business Russian Communication. Part 2 | \* | 2 | 115 | 2.459101163 | 1 | | 4 | Towards language universals through lexical semantics: introduction to lexical and semantic typology | Advanced Level | 2 | 166 | 1.717053691 | 0 | | 4 | Towards language universals through lexical semantics: introduction to lexical and semantic typology | \* | 2 | 166 | 1.717053691 | 1 | | 4 | \* | \* | 4 | 281 | 2.088077427 | 3 | | 5 | Build Your Professional ePortfolio in English | not-mentioned | 2 | 165 | 4.723412004 | 0 | | 5 | Build Your Professional ePortfolio in English | \* | 2 | 165 | 4.723412004 | 1 | | 5 | Chino bÃƒÂ¡sico: CÃƒÂ³mo dar una primera impresiÃƒÂ³n positiva | not-mentioned | 2 | 206 | 3.451554294 | 0 | | 5 | Chino bÃƒÂ¡sico: CÃƒÂ³mo dar una primera impresiÃƒÂ³n positiva | \* | 2 | 206 | 3.451554294 | 1 | | 5 | \* | \* | 4 | 371 | 4.087483149 | 3 | | 6 | Blended Language Learning: Design and Practice for Teachers | Intermediate Level | 2 | 190 | 2.918705009 | 0 | | 6 | Blended Language Learning: Design and Practice for Teachers | \* | 2 | 190 | 2.918705009 | 1 | | 6 | English for Science, Technology, Engineering, and Mathematics | Beginner Level | 3 | 307 | 2.010617201 | 0 | | 6 | English for Science, Technology, Engineering, and Mathematics | \* | 3 | 307 | 2.010617201 | 1 | | 6 | Perfect Tenses and Modals | Intermediate Level | 2 | 274 | 1.363491317 | 0 | | 6 | Perfect Tenses and Modals | \* | 2 | 274 | 1.363491317 | 1 | | 6 | \* | \* | 7 | 771 | 2.097604509 | 3 | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |

### 5.2 Ranking query using NTILE, RANK or DENSE RANK functions

Knowing where the main customers are, and which countries have a larger share of sales is essential for greater knowledge of the market and to outline strategies.

Based on this information, directors can prioritize actions to better understand the local market, evaluate the growth potential of each region, define promotion strategies, develop courses according to local demand. It is also a way of classifying countries to prioritize the commercial area.

**RANK 1**

|  |
| --- |
| **Natural language** |
| Ranking of the continents with more sales (dollar amount) over the past 3 years (2018-2020) |
|  |
| **SQL codes** |
| SELECT continent, SUM (sales\_dollar) AS sales\_dollar, SUM (minutes\_subscribed) AS minutes\_subscribed, RANK () OVER (ORDER BY SUM (sales\_dollar) DESC)  FROM fact\_sales JOIN dim\_customer USING (customer\_id) JOIN dim\_user\_account USING (user\_id) JOIN dim\_date USING (date\_id)  WHERE year IN (2018, 2019, 2020)  GROUP BY (continent) |
|  |
| **Results** |
| Graphic developed in GNUPLOT |

**DENSE RANK AND RANK**

|  |
| --- |
| **Natural language** |
| Ranking of the countries with more sales (dollar amount) per continent over the past 3 years (2018-2020)  and in the same table dense ranking of the continents with more sales over the past 3 years (2018-2020)  The difference between RANK and DENSE RANK is that the last, leaves no gaps in the ranking sequence when there are ties. |
|  |
| **SQL codes** |
| Select continent, country, sales\_dollar, minutes\_subscribed, dense\_rank () over (order by sales\_by\_continent desc) as denserank\_sales\_by\_continent, rank\_sales\_by\_country\_per\_continent  From  (select continent, country, sum (sales\_dollar) as sales\_dollar, sum (minutes\_subscribed) as minutes\_subscribed,  Sum (sum (sales\_dollar)) over (partition by continent) as sales\_by\_continent,  Rank () over (partition by continent order by sum (sales\_dollar) desc) as rank\_sales\_by\_country\_per\_continent  From fact\_sales join dim\_customer using (customer\_id) join dim\_user\_account using (user\_id) join dim\_date using (date\_id)  Where year in (2018, 2019, 2020)  Group by (continent, country)) as query1  Order by denserank\_sales\_by\_continent, rank\_sales\_by\_country\_per\_continent |
|  |
| **Results** |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **continent** | **country** | **sales\_dollar** | **minutes\_subscribed** | **denserank\_sales\_by\_continent** | **rank\_sales\_by\_country\_per\_continent** | | Asia | China | 6641910 | 2213970 | 1 | 1 | | Asia | India | 3619320 | 1206440 | 1 | 2 | | Asia | Japan | 1533570 | 511190 | 1 | 3 | | Asia | Turkey | 906120 | 302040 | 1 | 4 | | Asia | Indonesia | 672660 | 224220 | 1 | 5 | | Asia | Taiwan | 558900 | 186300 | 1 | 6 | | Asia | Iran | 540570 | 180190 | 1 | 7 | | Asia | Vietnam | 474960 | 158320 | 1 | 8 | | Asia | Malaysia | 455370 | 151790 | 1 | 9 | | Asia | Philippines | 354840 | 118280 | 1 | 10 | | Asia | Uzbekistan | 316920 | 105640 | 1 | 11 | | Asia | Saudi Arabia | 256560 | 85520 | 1 | 12 | | Asia | United Arab Emirates | 246030 | 82010 | 1 | 13 | | Asia | Kazakhstan | 229470 | 76490 | 1 | 14 | | Asia | Mongolia | 225480 | 75160 | 1 | 15 | | Asia | Pakistan | 208920 | 69640 | 1 | 16 | | Asia | Syria | 172350 | 57450 | 1 | 17 | | Asia | Oman | 149340 | 49780 | 1 | 18 | | Asia | Laos | 125670 | 41890 | 1 | 19 | | Asia | Iraq | 111750 | 37250 | 1 | 20 | | Asia | Israel | 107640 | 35880 | 1 | 21 | | Asia | Jordan | 102030 | 34010 | 1 | 22 | | Asia | Lebanon | 97560 | 32520 | 1 | 23 | | Asia | Bangladesh | 90360 | 30120 | 1 | 24 | | Asia | Georgia | 88740 | 29580 | 1 | 25 | | Asia | Afghanistan | 61020 | 20340 | 1 | 26 | | Asia | Kyrgyzstan | 50850 | 16950 | 1 | 27 | | Asia | Thailand | 34560 | 11520 | 1 | 28 | | Asia | Qatar | 23460 | 7820 | 1 | 29 | | Americas | United States | 5735310 | 1911770 | 2 | 1 | | Americas | Brazil | 3410460 | 1136820 | 2 | 2 | | Americas | Mexico | 1540200 | 513400 | 2 | 3 | | Americas | Venezuela | 638640 | 212880 | 2 | 4 | | Americas | Argentina | 520620 | 173540 | 2 | 5 | | Americas | Canada | 494520 | 164840 | 2 | 6 | | Americas | Peru | 350040 | 116680 | 2 | 7 | | Americas | Ecuador | 334080 | 111360 | 2 | 8 | | Americas | Colombia | 305280 | 101760 | 2 | 9 | | Americas | Chile | 210660 | 70220 | 2 | 10 | | Americas | Barbados | 199890 | 66630 | 2 | 11 | | Americas | Paraguay | 115470 | 38490 | 2 | 12 | | Americas | Nicaragua | 109920 | 36640 | 2 | 13 | | Americas | Guyana | 73530 | 24510 | 2 | 14 | | Americas | Jamaica | 70380 | 23460 | 2 | 15 | | Americas | Haiti | 63900 | 21300 | 2 | 16 | | Americas | Cuba | 61620 | 20540 | 2 | 17 | | Americas | Guadeloupe | 34260 | 11420 | 2 | 18 | | Americas | Uruguay | 30960 | 10320 | 2 | 19 | | Americas | Trinidad And Tobago | 27120 | 9040 | 2 | 20 | | Europe | Russia | 1116270 | 372090 | 3 | 1 | | Europe | Italy | 779730 | 259910 | 3 | 2 | | Europe | United Kingdom | 662700 | 220900 | 3 | 3 | | Europe | France | 552390 | 184130 | 3 | 4 | | Europe | Germany | 450000 | 150000 | 3 | 5 | | Europe | Spain | 366240 | 122080 | 3 | 6 | | Europe | Poland | 227250 | 75750 | 3 | 7 | | Europe | Sweden | 202950 | 67650 | 3 | 8 | | Europe | Greece | 158220 | 52740 | 3 | 9 | | Europe | Latvia | 140100 | 46700 | 3 | 10 | | Europe | Portugal | 89910 | 29970 | 3 | 11 | | Europe | Ukraine | 79680 | 26560 | 3 | 12 | | Europe | Romania | 75540 | 25180 | 3 | 13 | | Europe | Albania | 70350 | 23450 | 3 | 14 | | Europe | Ireland | 55440 | 18480 | 3 | 15 | | Europe | Netherlands | 36600 | 12200 | 3 | 16 | | Europe | Serbia | 35100 | 11700 | 3 | 17 | | Europe | Norway | 31950 | 10650 | 3 | 18 | | Europe | Austria | 30600 | 10200 | 3 | 19 | | Europe | Bulgaria | 21600 | 7200 | 3 | 20 | | Europe | Belgium | 21000 | 7000 | 3 | 21 | | Africa | Nigeria | 1063830 | 354610 | 4 | 1 | | Africa | South Africa | 442950 | 147650 | 4 | 2 | | Africa | Egypt | 358590 | 119530 | 4 | 3 | | Africa | Cameroon | 347190 | 115730 | 4 | 4 | | Africa | Mali | 255300 | 85100 | 4 | 5 | | Africa | Morocco | 248520 | 82840 | 4 | 6 | | Africa | Angola | 233190 | 77730 | 4 | 7 | | Africa | Tanzania | 210870 | 70290 | 4 | 8 | | Africa | Algeria | 148380 | 49460 | 4 | 9 | | Africa | Ethiopia | 147330 | 49110 | 4 | 10 | | Africa | Ghana | 133830 | 44610 | 4 | 11 | | Africa | South Sudan | 133620 | 44540 | 4 | 12 | | Africa | Liberia | 128430 | 42810 | 4 | 13 | | Africa | Burundi | 121080 | 40360 | 4 | 14 | | Africa | Zambia | 110880 | 36960 | 4 | 15 | | Africa | Tunisia | 104580 | 34860 | 4 | 16 | | Africa | Kenya | 84240 | 28080 | 4 | 17 | | Africa | Senegal | 78480 | 26160 | 4 | 18 | | Africa | Gabon | 59850 | 19950 | 4 | 19 | | Africa | Uganda | 18900 | 6300 | 4 | 20 | | Oceania | Tuvalu | 185460 | 61820 | 5 | 1 | | Oceania | New Zealand | 98460 | 32820 | 5 | 2 | | Oceania | Australia | 97770 | 32590 | 5 | 3 | | Oceania | Guam | 59760 | 19920 | 5 | 4 | | Oceania | French Polynesia | 28800 | 9600 | 5 | 5 | |

**NTILE**

|  |
| --- |
| **Natural language** |
| Order the countries according to sales dollar amount, from highest to lowest, in the last 3 years (2018-2020) and partition the list of countries into 4 parts. |
|  |
| **SQL codes** |
| SELECT country, SUM (sales\_dollar) as sales\_dollar, SUM (minutes\_subscribed) as minutes\_subscribed, NTILE (4) OVER (ORDER BY SUM (sales\_dollar) DESC)  FROM fact\_sales JOIN dim\_customer USING (customer\_id) JOIN dim\_user\_account USING (user\_id) JOIN dim\_date USING (date\_id)  WHERE year IN (2018, 2019, 2020)  GROUP BY (country) |
|  |
| **Results** |
| |  |  |  |  | | --- | --- | --- | --- | | **country** | **sales\_dollar** | **minutes\_subscribed** | **ntile** | | China | 6641910 | 2213970 | 1 | | United States | 5735310 | 1911770 | 1 | | India | 3619320 | 1206440 | 1 | | Brazil | 3410460 | 1136820 | 1 | | Mexico | 1540200 | 513400 | 1 | | Japan | 1533570 | 511190 | 1 | | Russia | 1116270 | 372090 | 1 | | Nigeria | 1063830 | 354610 | 1 | | Turkey | 906120 | 302040 | 1 | | Italy | 779730 | 259910 | 1 | | Indonesia | 672660 | 224220 | 1 | | United Kingdom | 662700 | 220900 | 1 | | Venezuela | 638640 | 212880 | 1 | | Taiwan | 558900 | 186300 | 1 | | France | 552390 | 184130 | 1 | | Iran | 540570 | 180190 | 1 | | Argentina | 520620 | 173540 | 1 | | Canada | 494520 | 164840 | 1 | | Vietnam | 474960 | 158320 | 1 | | Malaysia | 455370 | 151790 | 1 | | Germany | 450000 | 150000 | 1 | | South Africa | 442950 | 147650 | 1 | | Spain | 366240 | 122080 | 1 | | Egypt | 358590 | 119530 | 1 | | Philippines | 354840 | 118280 | 2 | | Peru | 350040 | 116680 | 2 | | Cameroon | 347190 | 115730 | 2 | | Ecuador | 334080 | 111360 | 2 | | Uzbekistan | 316920 | 105640 | 2 | | Colombia | 305280 | 101760 | 2 | | Saudi Arabia | 256560 | 85520 | 2 | | Mali | 255300 | 85100 | 2 | | Morocco | 248520 | 82840 | 2 | | United Arab Emirates | 246030 | 82010 | 2 | | Angola | 233190 | 77730 | 2 | | Kazakhstan | 229470 | 76490 | 2 | | Poland | 227250 | 75750 | 2 | | Mongolia | 225480 | 75160 | 2 | | Tanzania | 210870 | 70290 | 2 | | Chile | 210660 | 70220 | 2 | | Pakistan | 208920 | 69640 | 2 | | Sweden | 202950 | 67650 | 2 | | Barbados | 199890 | 66630 | 2 | | Tuvalu | 185460 | 61820 | 2 | | Syria | 172350 | 57450 | 2 | | Greece | 158220 | 52740 | 2 | | Oman | 149340 | 49780 | 2 | | Algeria | 148380 | 49460 | 2 | | Ethiopia | 147330 | 49110 | 3 | | Latvia | 140100 | 46700 | 3 | | Ghana | 133830 | 44610 | 3 | | South Sudan | 133620 | 44540 | 3 | | Liberia | 128430 | 42810 | 3 | | Laos | 125670 | 41890 | 3 | | Burundi | 121080 | 40360 | 3 | | Paraguay | 115470 | 38490 | 3 | | Iraq | 111750 | 37250 | 3 | | Zambia | 110880 | 36960 | 3 | | Nicaragua | 109920 | 36640 | 3 | | Israel | 107640 | 35880 | 3 | | Tunisia | 104580 | 34860 | 3 | | Jordan | 102030 | 34010 | 3 | | New Zealand | 98460 | 32820 | 3 | | Australia | 97770 | 32590 | 3 | | Lebanon | 97560 | 32520 | 3 | | Bangladesh | 90360 | 30120 | 3 | | Portugal | 89910 | 29970 | 3 | | Georgia | 88740 | 29580 | 3 | | Kenya | 84240 | 28080 | 3 | | Ukraine | 79680 | 26560 | 3 | | Senegal | 78480 | 26160 | 3 | | Romania | 75540 | 25180 | 3 | | Guyana | 73530 | 24510 | 4 | | Jamaica | 70380 | 23460 | 4 | | Albania | 70350 | 23450 | 4 | | Haiti | 63900 | 21300 | 4 | | Cuba | 61620 | 20540 | 4 | | Afghanistan | 61020 | 20340 | 4 | | Gabon | 59850 | 19950 | 4 | | Guam | 59760 | 19920 | 4 | | Ireland | 55440 | 18480 | 4 | | Kyrgyzstan | 50850 | 16950 | 4 | | Netherlands | 36600 | 12200 | 4 | | Serbia | 35100 | 11700 | 4 | | Thailand | 34560 | 11520 | 4 | | Guadeloupe | 34260 | 11420 | 4 | | Norway | 31950 | 10650 | 4 | | Uruguay | 30960 | 10320 | 4 | | Austria | 30600 | 10200 | 4 | | French Polynesia | 28800 | 9600 | 4 | | Trinidad And Tobago | 27120 | 9040 | 4 | | Qatar | 23460 | 7820 | 4 | | Bulgaria | 21600 | 7200 | 4 | | Belgium | 21000 | 7000 | 4 | | Uganda | 18900 | 6300 | 4 | |

### 5.3 Windowing query using the windowing clause

As the subscription can be assigned for one month, one quarter, one semester or one year, it is interesting to evaluate sales in a time window. Thus, we will evaluate the average sales, considering 3 months forward and 3 months back to understand the sales trend over the months and years. Moving averages are commonly used with time series to smooth short fluctuations and highlight long-term trends.

**CENTERED 7 MONTH MOVING AVERAGE**

|  |
| --- |
| **Natural language** |
| The centered 7 month moving average of sales during the last 4 years. |
|  |
| **SQL codes** |
| SELECT year, month, SUM (sales\_dollar) AS sales\_dollar, SUM (minutes\_subscribed) AS minutes\_subscribed,  AVG (SUM (sales\_dollar))  OVER (ORDER BY year DESC, month DESC ROWS BETWEEN 3 PRECEDING AND 3 FOLLOWING) AS avg\_sales\_dollar  FROM fact\_sales JOIN dim\_date USING (date\_id)  GROUP BY year, month |
|  |
| **Results** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **year** | **month** | **sales\_dollar** | **minutes\_subscribed** | **avg\_sales\_dollar** | | 2020 | 11 | 981,420 | 327,140 | 1,132,493 | | 2020 | 10 | 1,018,740 | 339,580 | 1,216,152 | | 2020 | 9 | 1,213,470 | 404,490 | 1,207,870 | | 2020 | 8 | 1,316,340 | 438,780 | 1,218,660 | | 2020 | 7 | 1,550,790 | 516,930 | 1,208,023 | | 2020 | 6 | 1,166,460 | 388,820 | 1,239,189 | | 2020 | 5 | 1,283,400 | 427,800 | 1,251,703 | | 2020 | 4 | 906,960 | 302,320 | 1,225,213 | | 2020 | 3 | 1,236,900 | 412,300 | 1,167,090 | | 2020 | 2 | 1,301,070 | 433,690 | 1,218,056 | | 2020 | 1 | 1,130,910 | 376,970 | 1,208,824 | | 2019 | 12 | 1,143,930 | 381,310 | 1,260,454 | | 2019 | 11 | 1,523,220 | 507,740 | 1,242,291 | | 2019 | 10 | 1,218,780 | 406,260 | 1,252,684 | | 2019 | 9 | 1,268,370 | 422,790 | 1,270,633 | | 2019 | 8 | 1,109,760 | 369,920 | 1,298,833 | | 2019 | 7 | 1,373,820 | 457,940 | 1,256,396 | | 2019 | 6 | 1,256,550 | 418,850 | 1,244,546 | | 2019 | 5 | 1,341,330 | 447,110 | 1,235,717 | | 2019 | 4 | 1,226,160 | 408,720 | 1,257,609 | | 2019 | 3 | 1,135,830 | 378,610 | 1,228,461 | | 2019 | 2 | 1,206,570 | 402,190 | 1,238,614 | | 2019 | 1 | 1,263,000 | 421,000 | 1,203,476 | | 2018 | 12 | 1,169,790 | 389,930 | 1,214,606 | | 2018 | 11 | 1,327,620 | 442,540 | 1,241,263 | | 2018 | 10 | 1,095,360 | 365,120 | 1,279,011 | | 2018 | 9 | 1,304,070 | 434,690 | 1,262,044 | | 2018 | 8 | 1,322,430 | 440,810 | 1,285,736 | | 2018 | 7 | 1,470,810 | 490,270 | 1,246,603 | | 2018 | 6 | 1,144,230 | 381,410 | 1,271,721 | | 2018 | 5 | 1,335,630 | 445,210 | 1,250,293 | | 2018 | 4 | 1,053,690 | 351,230 | 1,213,521 | | 2018 | 3 | 1,271,190 | 423,730 | 1,142,259 | | 2018 | 2 | 1,154,070 | 384,690 | 1,154,207 | | 2018 | 1 | 1,065,030 | 355,010 | 1,127,361 | | 2017 | 12 | 971,970 | 323,990 | 1,150,033 | | 2017 | 11 | 1,227,870 | 409,290 | 1,106,267 | | 2017 | 10 | 1,147,710 | 382,570 | 1,152,489 | | 2017 | 9 | 1,212,390 | 404,130 | 1,154,430 | | 2017 | 8 | 964,830 | 321,610 | 1,229,906 | | 2017 | 7 | 1,477,620 | 492,540 | 1,205,366 | | 2017 | 6 | 1,078,620 | 359,540 | 1,192,551 | | 2017 | 5 | 1,500,300 | 500,100 | 1,201,890 | | 2017 | 4 | 1,056,090 | 352,030 | 1,235,970 | | 2017 | 3 | 1,058,010 | 352,670 | 1,192,294 | | 2017 | 2 | 1,277,760 | 425,920 | 1,206,686 | | 2017 | 1 | 1,203,390 | 401,130 | 1,169,974 | | 2016 | 12 | 1,171,890 | 390,630 | 1,188,955 | | 2016 | 11 | 1,179,360 | 393,120 | 1,215,144 | | 2016 | 10 | 1,243,320 | 414,440 | 1,199,490 |     Graphic developed in GNUPLOT |

### 5.4 Period-to-period comparison query

Another relevant information for managers is the comparison of the use of the system over the years.

The number of accesses and the access time per day, week and month are relevant for understanding the need for more technological resources (e.g., increasing broadband), monitoring the number of customers and attractiveness of courses.

Therefore, period-to-period comparison query will be used. comparing values across time periods.

**PERIOD-TO-PERIOD COMPARISON QUERY**

|  |
| --- |
| **Natural language** |
| For each week of the year, we report the total session time of students in the current year, the previous year and 2 years ago.  The same is done for the number of users. For each week of the year, we report the total amount of session for students in the current year, the previous year and 2 years ago. |
|  |
| **SQL codes** |
| Select week\_of\_year, session\_minutes as session\_min\_2020, session\_minutes\_1yearbefore as session\_min\_2019, session\_minutes\_2yearsbefore as session\_min\_2018,  count\_stud as count\_stud\_2020, count\_stud\_1yearbefore as count\_stud\_2019, count\_stud\_2yearsbefore as count\_stud\_2018  from (Select week\_of\_year, year,  sum (session\_minutes) as session\_minutes,  Lag(sum(session\_minutes),1) over (order by week\_of\_year,year) as session\_minutes\_1yearbefore,  Lag(sum(session\_minutes),2) over (order by week\_of\_year,year) as session\_minutes\_2yearsbefore,  count (customer\_id) as count\_stud,  Lag(count(customer\_id),1) over (order by week\_of\_year,year) as count\_stud\_1yearbefore,  Lag(count(customer\_id),2) over (order by week\_of\_year,year) as count\_stud\_2yearsbefore  From (select customer\_id, week\_of\_year, year, sum (session\_minutes) as session\_minutes  From fact\_tutoring join dim\_date using (date\_id)  Where year in (3017, 2018, 2019, 2020)  Group by customer\_id, week\_of\_year, year) as query1  Group by week\_of\_year,year) as query2  where year=2020  and week\_of\_year>=40  and week\_of\_year<=50 |

|  |
| --- |
| **Results** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | week\_of\_year | session\_min\_2020 | session\_min\_2019 | session\_min\_2018 | count\_stud\_2020 | count\_stud\_2019 | count\_stud\_2018 | | 40 | 247 | 551 | 47 | 2 | 5 | 1 | | 41 | 449 | 485 | 121 | 4 | 6 | 1 | | 42 | 312 | 681 | 244 | 5 | 6 | 3 | | 43 | 119 | 173 | 123 | 4 | 3 | 1 | | 44 | 419 | 185 | 408 | 4 | 3 | 4 | | 45 | 477 | 691 | 603 | 4 | 8 | 4 | | 46 | 501 | 111 | 336 | 5 | 2 | 2 | | 47 | 728 | 570 | 394 | 6 | 5 | 4 | | 48 | 596 | 526 | 728 | 6 | 4 | 6 | | 49 | 671 | 753 | 15 | 6 | 9 | 1 | | 50 | 576 | 761 | 150 | 6 | 8 | 2 |       Graphics developed in GNUPLOT |

6 Data analisys tool

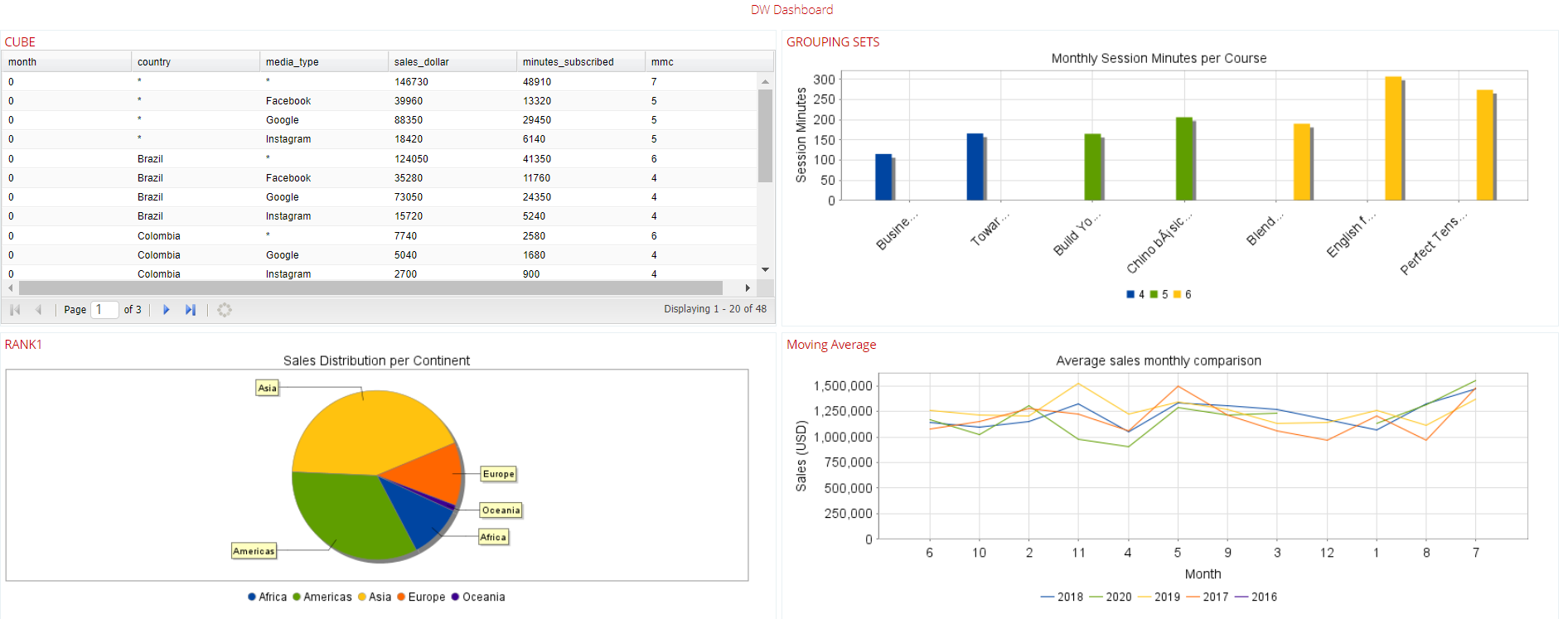


In order to provide valuable insight to the business and support decision making, a real time dashboard creation was proposed, this dashboard should be dynamic not static so change would be visible over time without the need to recreate the dashboard, it also should include most relevant graphics and tables from the queries created in the chapter 5.

Also, it is important to mentioned that as our data warehouse is published within the UNI BZ VPN, so it was preferred to use a on premises tool as oppose to a cloud-based tool.

### 6.1 Data analysis tool selection and set up

As we were already familiar with the Pentaho tool and as this tool could be run as a stand-alone server that would run locally on the selected machine, we chose to use the business analytics module in order to achieve the goal of creating a dynamic dashboard. It was easy to install in one of the workstations we use and that is connected to the UNIBZ VPN, facilitating data accessibility and the sharing capabilities.



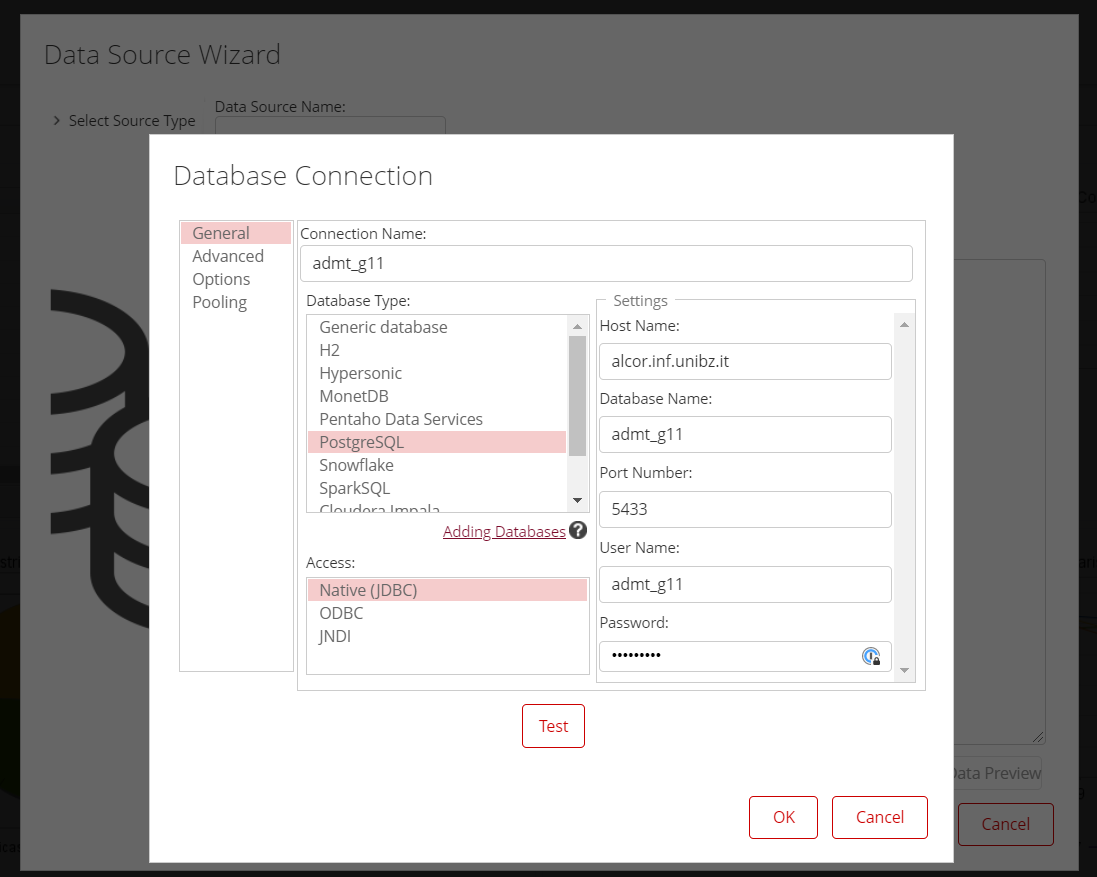
Pentaho Data warehouse Realtime Dashboard.

To create the desire dashboard two processes should be performed on Pentaho, set up of all the data sources required and creation of the dashboard template linking the data to the graphs and tables.

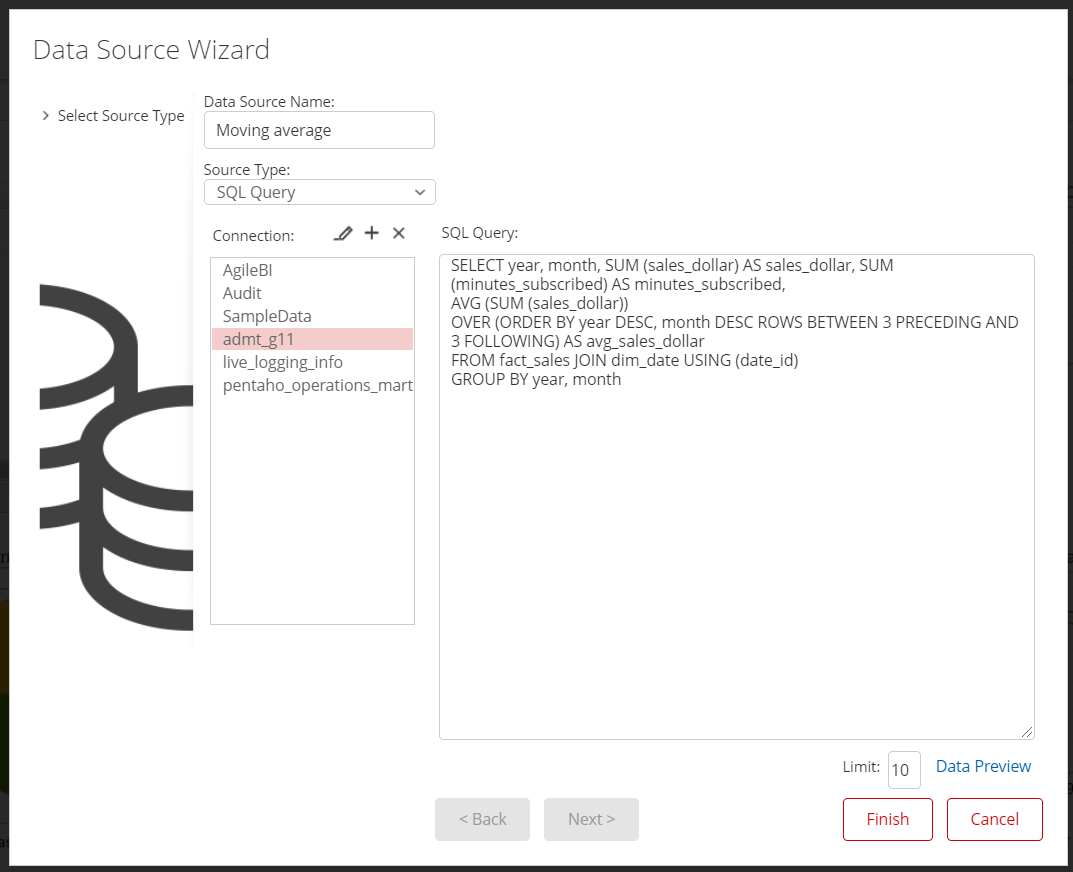
#### Adding data sources

Before we could create the actual dashboard, the proper data sources should be linked to our Pentaho server. To do this we need to follow the next steps:

* Ensure that the system is successfully logged into the VPN
* Check that the database is up and accessible through PG Admin
* Add a new database connection to our data warehouse specifying the connection details provided on class:



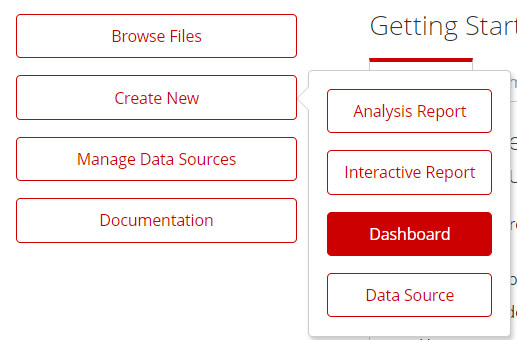
* Once the database connection is created and tested new data sources can be added, in our case we used the source type “SQL Query” then we specified the desired query on the right give this query source a name and then tested the query with the data preview button:



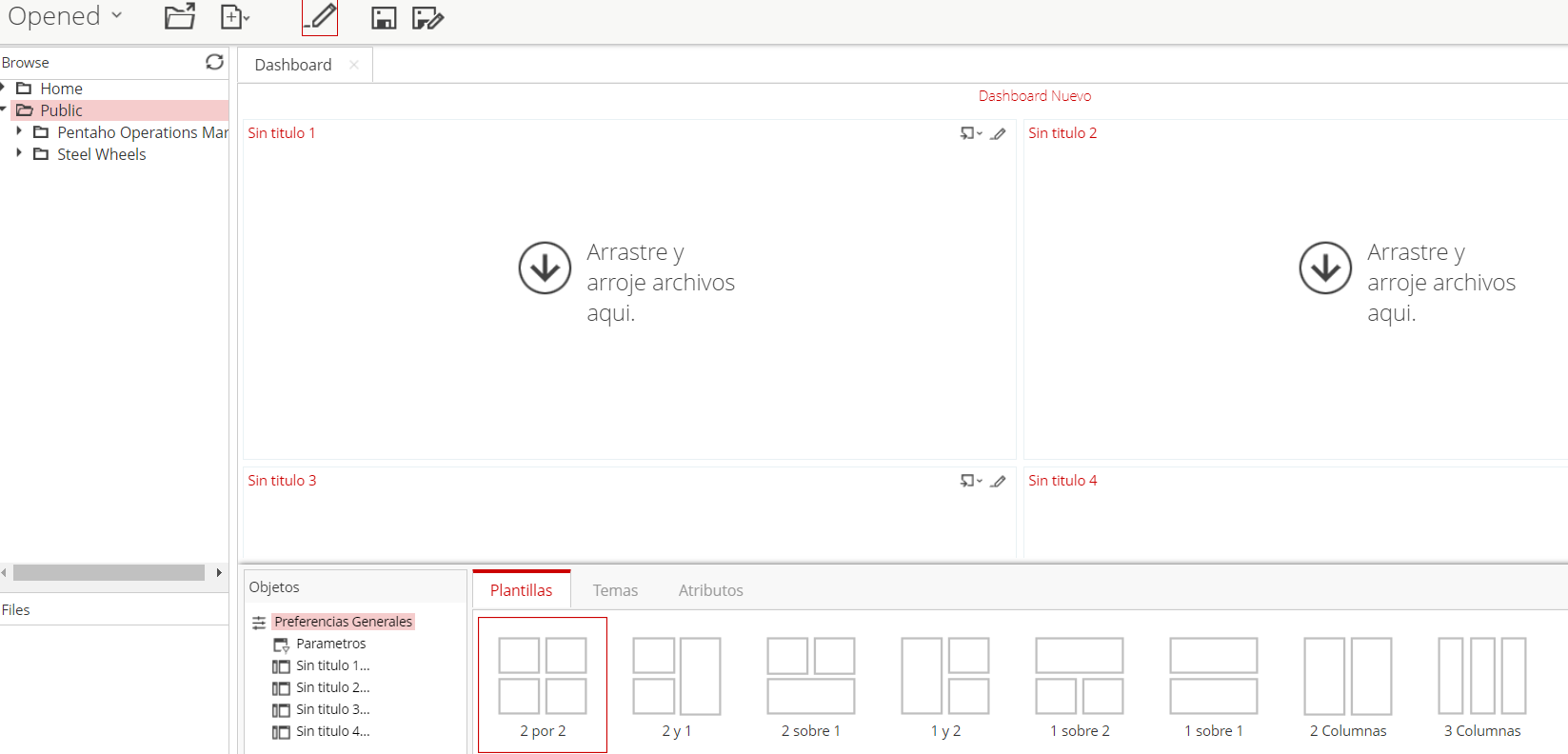
#### Dashboard creation

To create the dashboard the following steps must be performed:

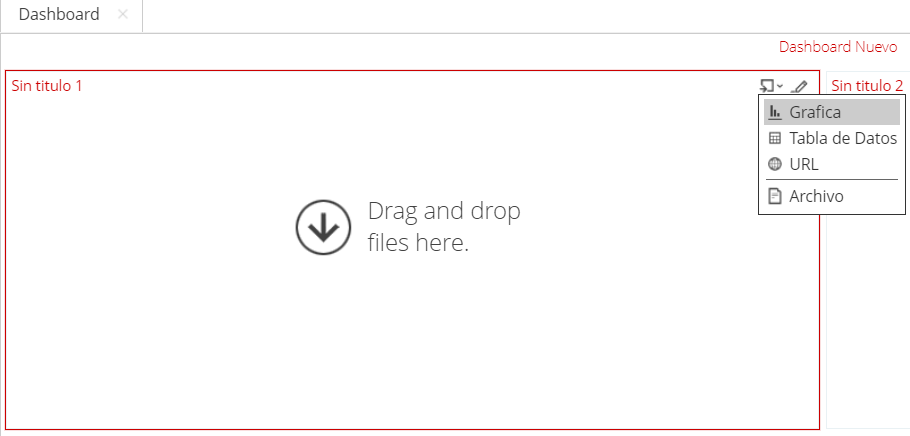
* Select Create New/ Dashboard:



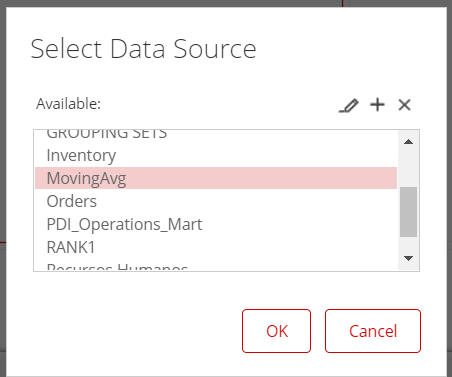
* This will take us to a blank dashboard where we can choose the layout (We chose 2 x 2)



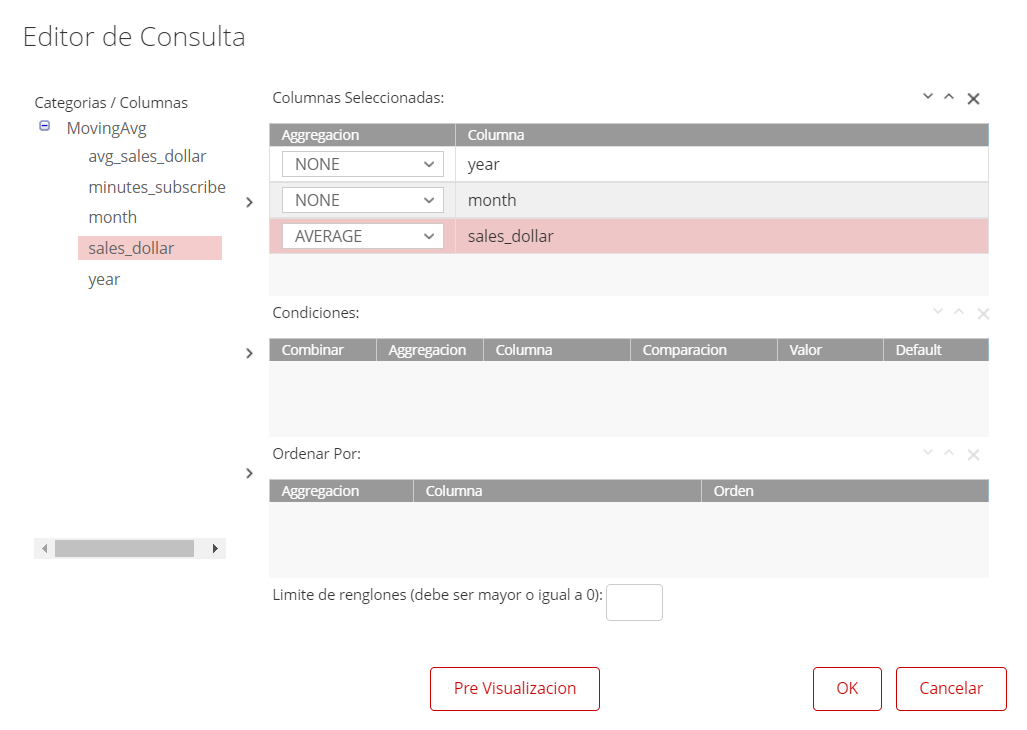
* Now for each slot of the layout we need to define the type of element we want to display, it could either a graph or a document:



* Next the data source must be selected, this would be the right SQL Query source added previously.



* Later the tool will show us the columns available from the query that we can select to use in our element:



* Finally, we can select the type of graph and assign the selected columns to the right axis as well as specifying the graph title and axis names:



# References

Dignos, A. (2020). *Lecture Notes.*

Hitachi. (2020, 12 26). *www.hitachivantara.com*. Retrieved from www.hitachivantara.com: https://www.hitachivantara.com/en-us/pdf/datasheet/pentaho-business-analytics-platform-datasheet.pdf

Koṫlawī, A. Y. (n.d.). *Akhlāq-uṣ-Ṣāliḥīn.* Karachi, Pakistan: Maktaba-tul-Madīnaĥ.

MAHLKNECHT , G., DIGNOS, A., & KOZMINA, N. (2019). MODELING AND QUERYING FACTS WITH PERIOD TIMESTAMPS IN DATA WAREHOUSES. *Int. J. Appl. Math. Comput. Sci*.

MATTEO GOLFARELLI, D. M. (1998). THE DIMENSIONAL FACT MODEL: A CONCEPTUAL MODEL FOR DATA WAREHOUSES. *International Journal of Cooperative Information Systems*.

Moody, D. L., & Kortink, M. A. (n.d.). From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design.

Moody, D. L., & Kortink, M. A. (n.d.). From ER Models to Dimensional Models Part II: Advanced Design Issues.

Prenner, J. A. (2014). *Advanced Data Management Technologies – Project Module 1 – Data Warehouse.*

Site. (n.d.). *http://gnuplot.sourceforge.net/.*