Data Warehousing Assigment - Part 2 Report

Abbas Al-Eryani	Geo Jolly	Alberto Parravicini	Almas Zhanibek
000439330	000439156	000437917	000439431

1 Notes about the files

- "Package.dtsx" is the **ETL** package. For completeness, the entire *integration* project is provided, in the "Assignment2" folder.
- The **ETL** phase doesn't use any external connection or flat files beside *source* and *target*, i.e. the provided **OLTP** and **OLAP** databases. Note that the **OLAP** database has to conform with what is contained in the next section for the script to work.
- The cube is contained in the "CubeFinal" folder.

2 Notes about the Data Warehouse Schema

As starting point of the second assignment, we have been provided with an *SQL script* to create the tables of the data warehouse. However, the script had some issues that had to be fixed before starting the **ETL procedure**:

- Tables *actor, address, category, city, country, film, inventory, language, rental* were set to have **auto-incrementing keys**: however, these tables do not use *surrogate keys*, and their keys are the same of the oltp database. As such, it was necessary to remove the *auto-increment* setting.
- Table *date* was not set to have an *auto-incrementing* key. Given that the tuples of this table have to be loaded from multiple sources, having an auto-incrementing key is definitely a good idea to simplify the data-warehouse management. We decided to modify the table and add an *auto-incrementing* key.
- The cyclic *foreign key* constraint between the tables *store* and *staff* (relatively to the store manager, and the store where he works) proved to troublesome to handle. Indeed, it is not possible to add a new store without previously having added its manager, and viceversa it's not possible to add a manager without adding his store beforehand.
  To deal with this problem during the **ETL** p, it was necessary to temporarily disable these two *foreign key* constraints. More details in the specific section.

3 ETL

This section includes pictures of the steps of the ETL procedure, with comments added where needed.

Note that for brevity some data-flows are not reported, in case the data-flow simply transfers data from the **OLTP** database to the data warehouse.

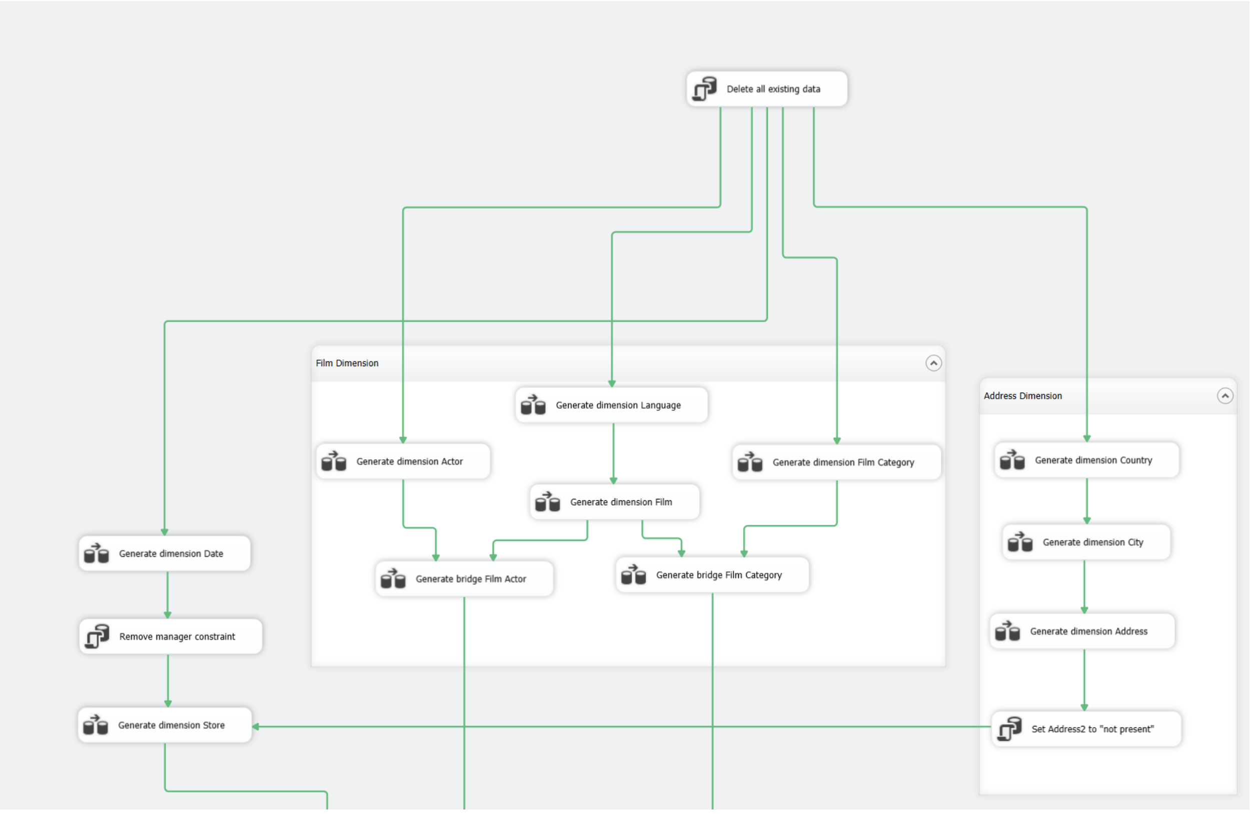

Delete all existing data
Film Dimension
Generate dimension Language
Generate dimension Actor
Generate dimension Film Category
Generate dimension Film
Generate bridge Film Actor
Generate bridge Film Category
Address Dimension
Generate dimension Country
Generate dimension City
Generate dimension Address
Set Address2 to "not present"
Generate dimension Date
Remove manager constraint
Generate dimension Store

Figure 1: This phase loads the data for *film* and *address*, and the related tables. At the beginning, all data are erased: if the data-warehouse is already empty, this part is redundant; we added it so that we could more easily test the *ETL* script.
Before generating the dimensions *store*, the *foreign key* constraint *[FK_store_staff]* is removed.

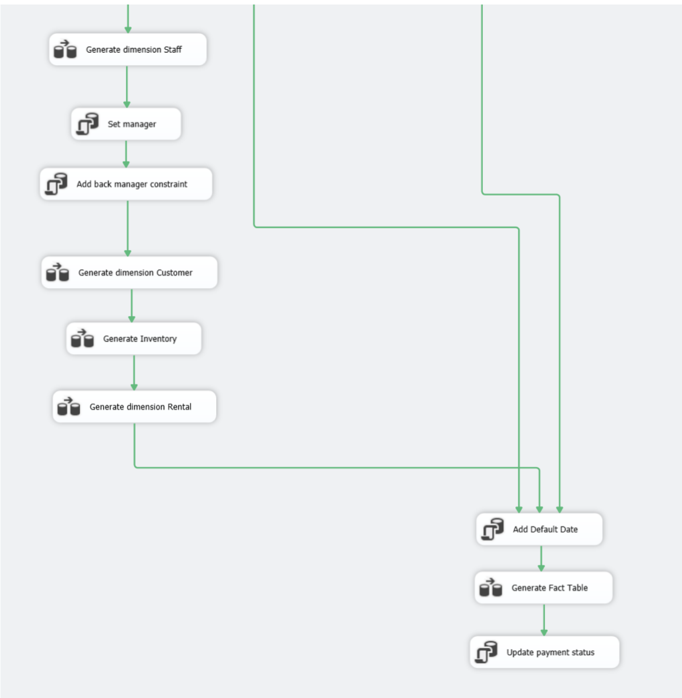
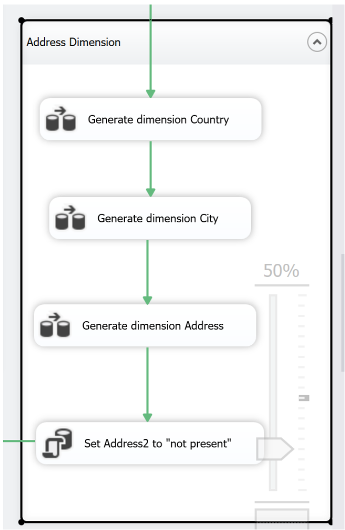
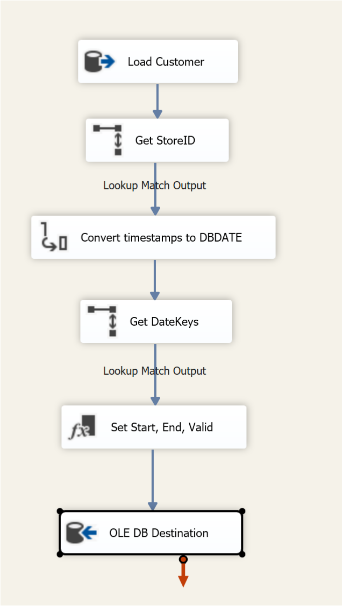


Figure 2: This phase loads the data of the other dimensions, and populates the fact table.
After populating the table *staff*, the constraint *[FK_store_staff]* is restored. Before populating the fact table, the default date *1900-01-01* is added to the database (see the specific section for more details). As last step, rentals without payment have the attribute *payment_status* set to *false*.

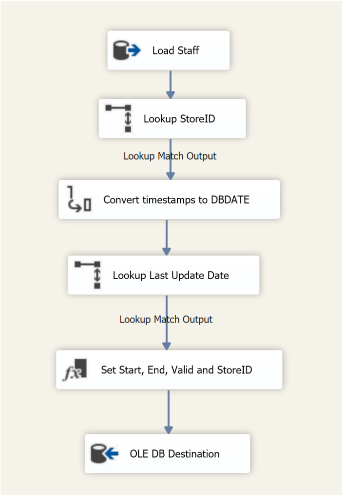
- **Note**: the data-flow for the film dimensions is trivial and is not reported.


Address Dimension
Generate dimension Country
Generate dimension City
50%
Generate dimension Address
Set Address2 to "not present"

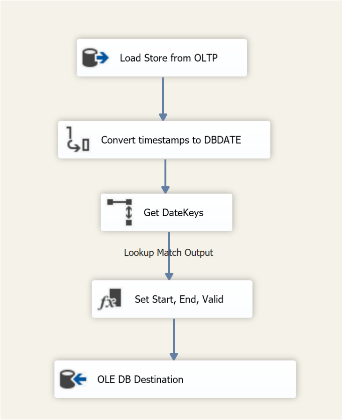
(a) Loading of *address*. At the end, empty values of *address2* are set to *not_present*.



(b) Loading of *customer*



(a) Loading of *staff*



(b) Loading of *store*

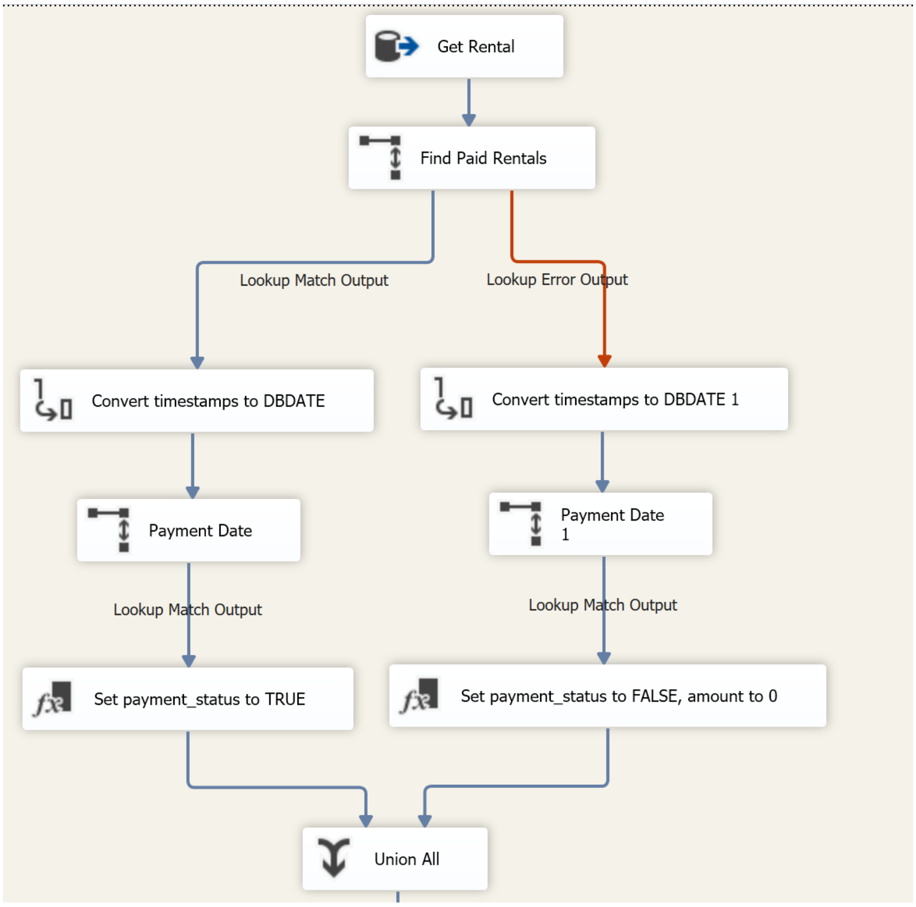


Figure 5: Loading of the *fact table*

- First of all, data are loaded from the *rental* table; then a lookup is done on the *payment* table, by comparing **rental_id**.
- If a match is found, the rental was paid. In this case, the values of *customer, staff* are updated to be the one in the *payment* table. In fact, the customer and the staff who performed a rental could be different from the ones who perform the payment. The data warehouse could have been designed to hold both couples, where available, but in our case only one couple can be kept.
- If no match was found, the rental was not paid: in this case, *amount* is set to 0, and the customer and staff who performed the rental are stored (as it is not possible to have *null* values).
- If no match was found, the *payment date* is temporarily set to the *rental date*. Later, it is substituted with the default value *1900-01-01*, as it's not possible to use *null* values.

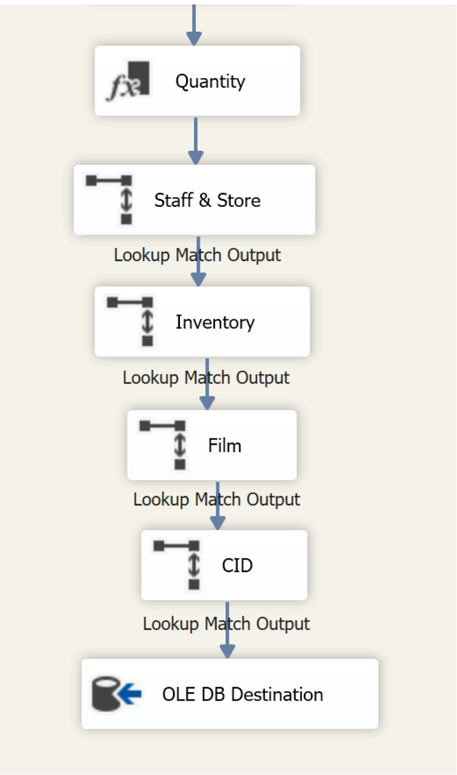


Figure 6: Loading of the *fact table*

- From the description, it is not clear what the *quantity* measure is.
  From the **OLTP** schema it is possible to infer that each payment refers to just one copy of a movie, so we assumed that *quantity* is simply used to count the number of facts. As such, it is set to 1 for all tuples.
- Then, all the keys of the fact table are loaded.

4 Data-Cube

This section includes an explanation about the structure of the *data-cube*, with pictures and comments where needed.

4.1 Overall structure

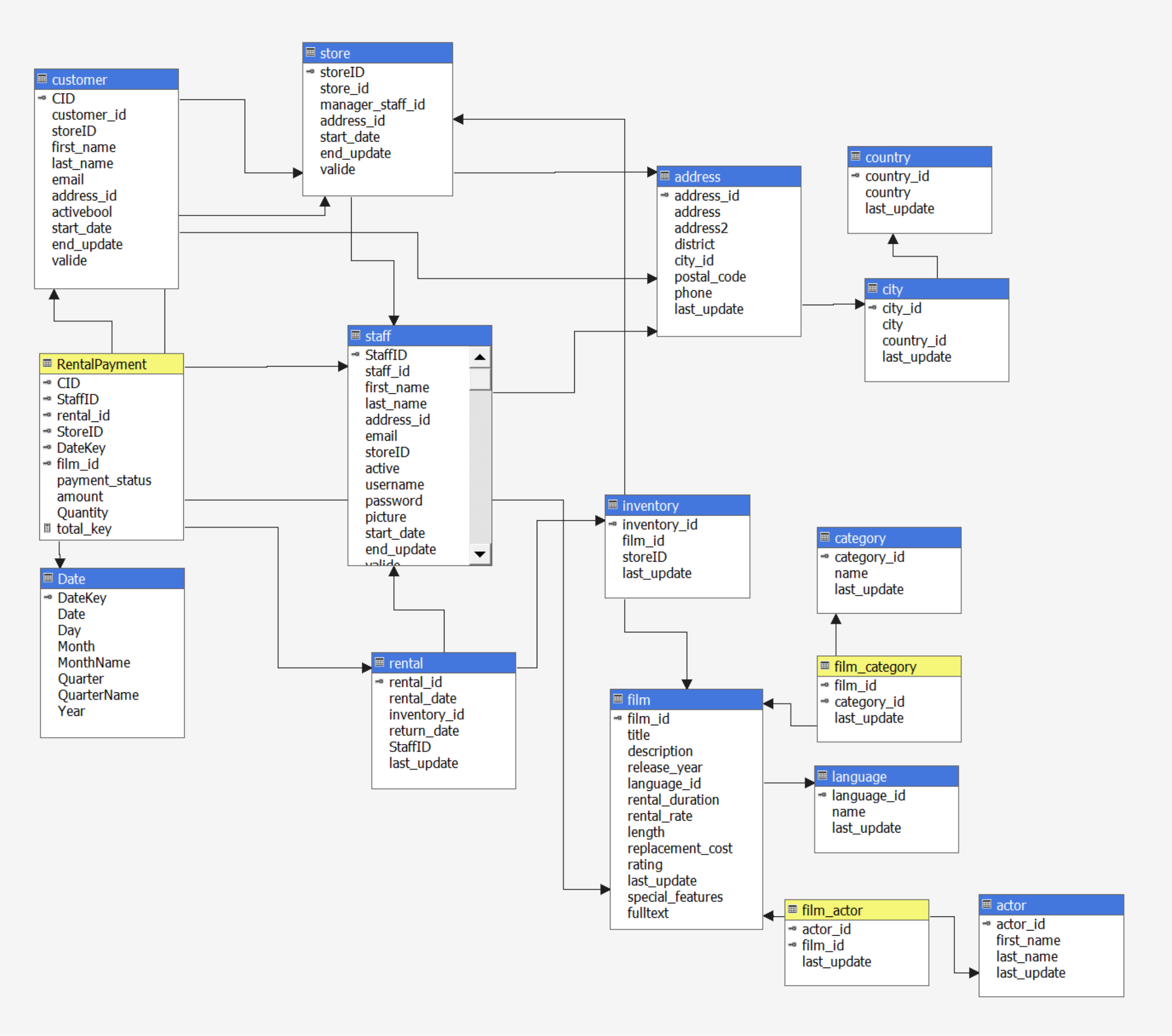
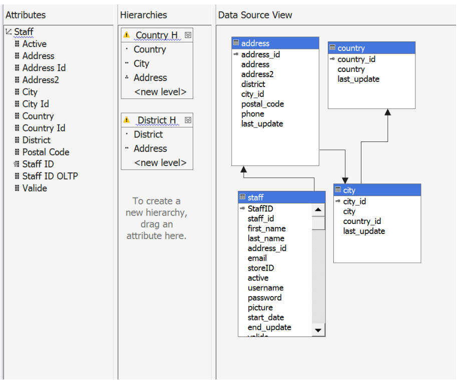
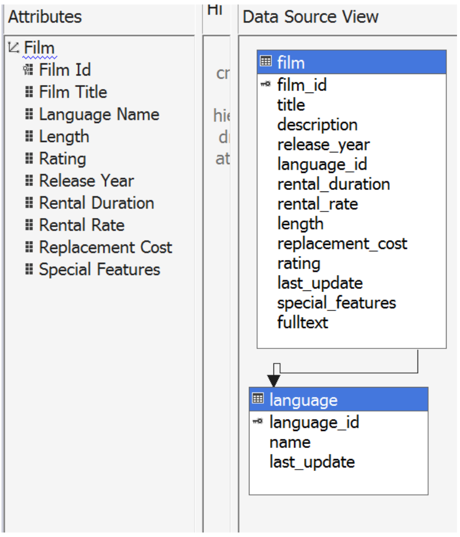


Figure 7: Loading of the *Data-Cube*. In yellow, the *fact table* and the *bridge tables*. *total_key* is the overall key of the *fact table*, and is used by the *payment status* dimension.

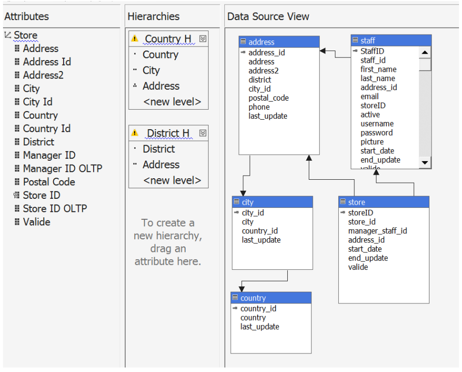
4.2 *Dimensions*



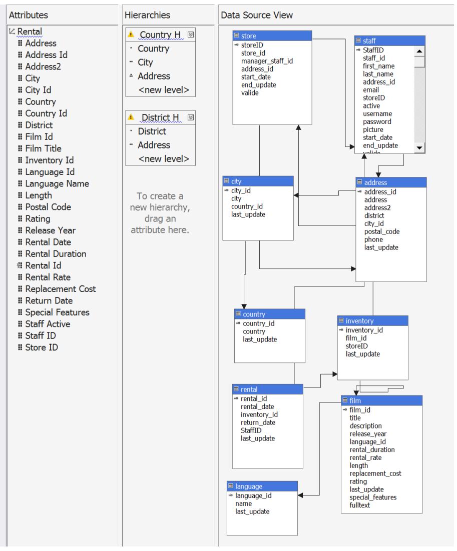
(a) *Staff* dimension



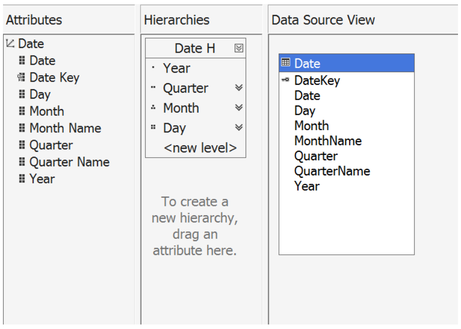
(b) *Film* dimension



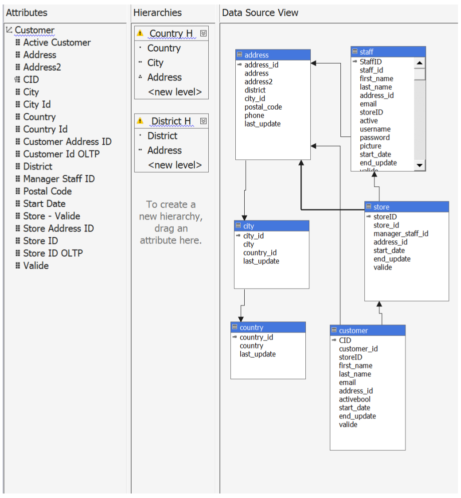
(a) *Store* dimension



(b) *Rental* dimension



(a) *Date* dimension



(b) *Customer* dimension

• Not shown, the supporting *Category* and *Actor* dimensions, connected to *Film* through bridge tables. Also not shown, the *Payment Status* dimension, which allows to aggregate distinguishing whether a rental was paid or not.

4.3 *Attribute Relations*

What follows are two examples of *attribute relationships* structures. Other dimensions follow a similar pattern, and are not shown for brevity.

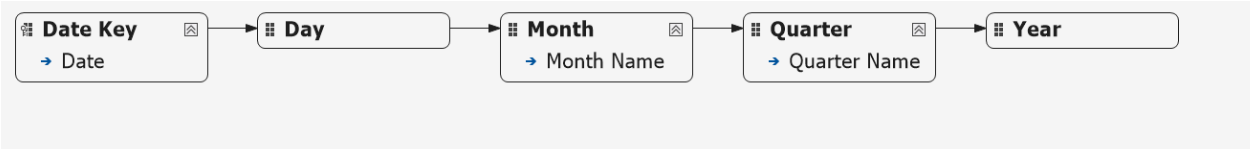


Figure 11: Attribute relationship of the dimension **Date**. Note that the hierarchy is built on the numerical values, but when displaying, the textual names for month and semester are used. Also, level of the hierarchy use composite keys: as an example, *month* has both *month* and *year* as key.

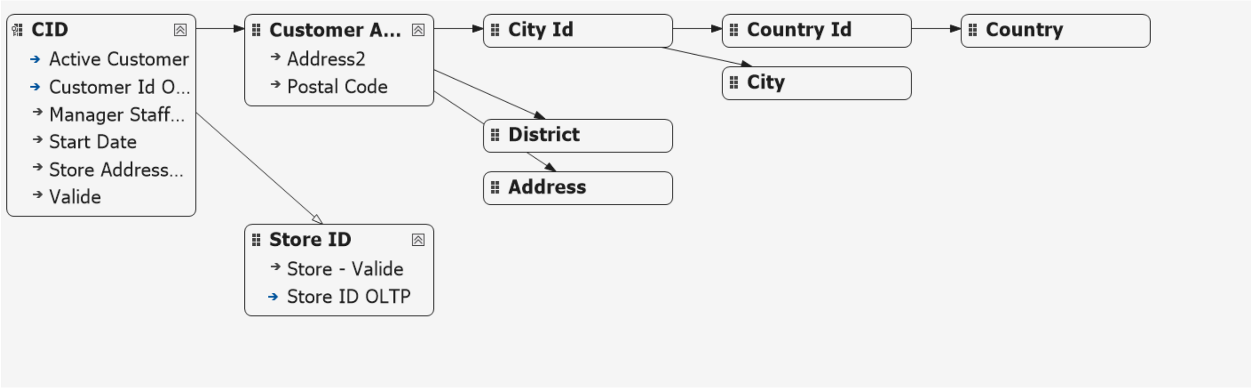
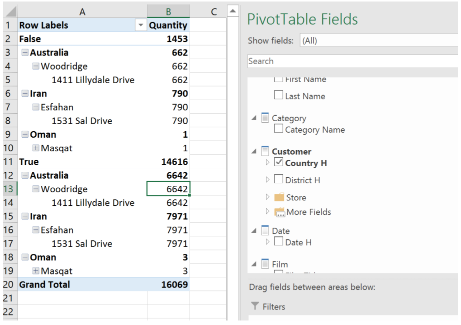
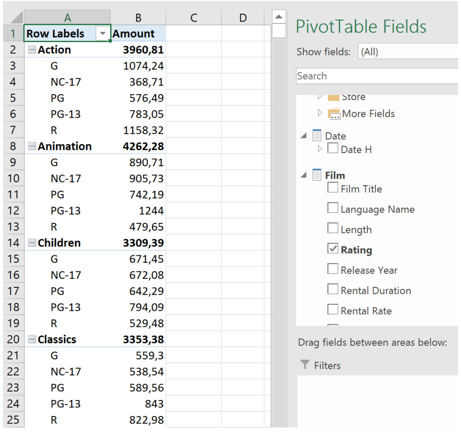


Figure 12: Attribute relationship of the dimension **Customer**. Dimensions *Rental*, *Store* and *Staff* have a similar structure. Note that the *valide* attribute is kept so that it is possible to aggregate only using the values that are currently valid.



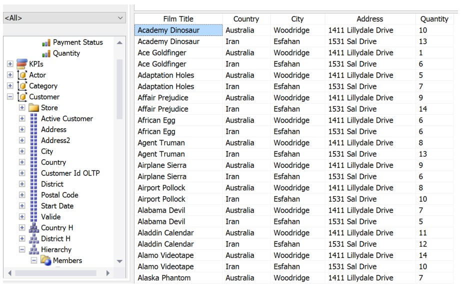
	A	B
1	**Row Labels**	**Quantity**
2	**False**	**1453**
3	⊟**Australia**	**662**
4	⊟Woodridge	662
5	1411 Lillydale Drive	662
6	⊟**Iran**	**790**
7	⊟Esfahan	790
8	1531 Sal Drive	790
9	⊟**Oman**	**1**
10	⊞Masqat	1
11	**True**	**14616**
12	⊟**Australia**	**6642**
13	⊟Woodridge	6642
14	1411 Lillydale Drive	6642
15	⊟**Iran**	**7971**
16	⊟Esfahan	7971
17	1531 Sal Drive	7971
18	⊟**Oman**	**3**
19	⊞Masqat	3
20	**Grand Total**	**16069**
21		
22		

(a) Number of rentals, divided by *paid* and *not paid*, and by *country, city, address*.



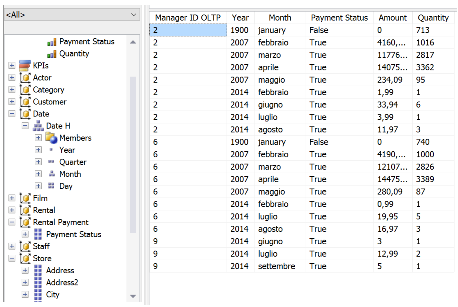
	A	B
1	**Row Labels**	**Amount**
2	⊟**Action**	**3960,81**
3	G	1074,24
4	NC-17	368,71
5	PG	576,49
6	PG-13	783,05
7	R	1158,32
8	⊟**Animation**	**4262,28**
9	G	890,71
10	NC-17	905,73
11	PG	742,19
12	PG-13	1244
13	R	479,65
14	⊟**Children**	**3309,39**
15	G	671,45
16	NC-17	672,08
17	PG	642,29
18	PG-13	794,09
19	R	529,48
20	⊟**Classics**	**3353,38**
21	G	559,3
22	NC-17	538,54
23	PG	589,56
24	PG-13	843
25	R	822,98

(b) Rental income divided by *film category* and *film rating*.



Film Title	Country	City	Address	Quantity
Academy Dinosaur	Australia	Woodridge	1411 Lillydale Drive	10
Academy Dinosaur	Iran	Esfahan	1531 Sal Drive	13
Ace Goldfinger	Australia	Woodridge	1411 Lillydale Drive	1
Ace Goldfinger	Iran	Esfahan	1531 Sal Drive	6
Adaptation Holes	Australia	Woodridge	1411 Lillydale Drive	5
Adaptation Holes	Iran	Esfahan	1531 Sal Drive	7
Affair Prejudice	Australia	Woodridge	1411 Lillydale Drive	9
Affair Prejudice	Iran	Esfahan	1531 Sal Drive	14
African Egg	Australia	Woodridge	1411 Lillydale Drive	6
African Egg	Iran	Esfahan	1531 Sal Drive	6
Agent Truman	Australia	Woodridge	1411 Lillydale Drive	8
Agent Truman	Iran	Esfahan	1531 Sal Drive	13
Airplane Sierra	Australia	Woodridge	1411 Lillydale Drive	9
Airplane Sierra	Iran	Esfahan	1531 Sal Drive	6
Airport Pollock	Australia	Woodridge	1411 Lillydale Drive	8
Airport Pollock	Iran	Esfahan	1531 Sal Drive	10
Alabama Devil	Australia	Woodridge	1411 Lillydale Drive	7
Alabama Devil	Iran	Esfahan	1531 Sal Drive	5
Aladdin Calendar	Australia	Woodridge	1411 Lillydale Drive	11
Aladdin Calendar	Iran	Esfahan	1531 Sal Drive	12
Alamo Videotape	Australia	Woodridge	1411 Lillydale Drive	14
Alamo Videotape	Iran	Esfahan	1531 Sal Drive	10
Alaska Phantom	Australia	Woodridge	1411 Lillydale Drive	7

(a) Number of rentals, divided by *film title*, and by *customer geographical information*.



Manager ID OLTP	Year	Month	Payment Status	Amount	Quantity
2	1900	january	False	0	713
2	2007	febbraio	True	4160,...	1016
2	2007	marzo	True	11776...	2817
2	2007	aprile	True	14075...	3362
2	2007	maggio	True	234,09	95
2	2014	febbraio	True	1,99	1
2	2014	giugno	True	33,94	6
2	2014	luglio	True	3,99	1
2	2014	agosto	True	11,97	3
6	1900	january	False	0	740
6	2007	febbraio	True	4190,...	1000
6	2007	marzo	True	12107...	2826
6	2007	aprile	True	14475...	3389
6	2007	maggio	True	280,09	87
6	2014	febbraio	True	0,99	1
6	2014	luglio	True	19,95	5
6	2014	agosto	True	16,97	3
9	2014	giugno	True	3	1
9	2014	luglio	True	12,99	2
9	2014	settembre	True	5	1

(b) Rental income and quantity divided by *manager_id*, *year and month of payment* and *payment_status*.