

Sprawozdanie AiED - BI/OLAP

Leszek Błażewski - 241264

12.12.2021

Spis treści

1	Cel laboratoriów	3
2	Proces ETL	3
2.1	Połączenie z plikami CSV oraz bazą danych	4
2.2	Struktura procesu - Control Flow	4
2.2.1	Tworzenie i usuwanie tabel w bazie danych	5
2.2.2	Importowanie kursów i semestrów - Data flow	7
2.2.3	Import teachers data - Sequence container	8
2.2.4	Import students - Data flow	8
2.2.5	Import grades - Data flow	9
2.2.6	Create missing teachers - Execute SQL task	10
2.2.7	Calculate workload - Execute SQL task	10
3	OLAP	11
3.1	Data Source View	11
3.1.1	Named calculations	12
3.2	Kostka wielowymiarowa	14
3.2.1	Miary i wymiary kostki	14
3.2.2	Hierarchie atrybutów	15
3.2.2.1	Hierarchia wymiaru Students	15
3.2.2.2	Hierarchia wymiaru Courses	16
3.2.2.3	Hierarchia wymiaru Teachers	16
3.2.2.4	Hierarchia wymiaru Semesters	16
4	Analiza danych	17
4.1	Kto ocenia surowiej?	17
4.2	Czy tytuł ma wpływ na wystawianą ocenę?	17
4.3	Obciążenie oraz wydział nauczyciela	18
4.4	Oceny w zależności od płci i kierunku	19
4.5	Oceny w zależności od typu zajęć i formy zaliczenia	20
4.6	Zapytania MDX	20
4.6.1	Średnia ocen w zależności od typu semestru	20
4.6.2	Średnia ocen w zależności od grupy kursu	20

1 Cel laboratoriów

Celem pierwszej części laboratorium było zapoznanie z technologiami umożliwiającymi budowę środowiska OLAP, które pozwala na wielowymiarową analizę faktów w funkcji wymiarów. Cały proces został zrealizowany przy pomocy dwóch modułów narzędzia MS SQL Server 2019:

- Integration Services (SSIS) - realizacja procesu ETL.
- Analysis Services (SSAS) - projektowanie i wykonanie wielowymiarowego modelu danych w postaci kostki.

Zapoznanie z technologiami odbyło się na przykładzie wielowymiarowej analizy ocen wystawianych studentom na Wydziale Elektroniki, w funkcji wszystkich dostępnych w danych atrybutów (zmiennych) mogących mieć związek na wystawiane oceny – takich jak np. typ kursu, semestr studiów, kierunek, specjalność, atrybuty opisujące studenta oraz inne.

Zadanie składało się z dwóch głównych etapów: procesu ETL (pkt. 1,2,3) oraz budowy wielowymiarowej kostki (pkt. 4,5) w skład, których wchodziły następujące zadania:

1. Załadowanie danych o studentach z dziekanatu z plików CSV.
2. Transformacja i czyszczenie wraz z likwidacją niespójności w załadowanych danych oraz relacjach pomiędzy nimi.
3. Umieszczenie przetworzonych danych w bazie MS SQL Server.
4. Określenie relacji pomiędzy tabelami (fakty - wymiary).
5. Zdefiniowanie i transformacja zmiennych, które będą przedmiotem analizy wraz z odpowiednimi wymiarami.

W sekcji 2 opisano pierwszą część procesu związaną z transformacją i ładowaniem danych - ETL, natomiast w rozdziale 3 przedstawiono drugą część związaną z budową relacji oraz tworzeniem kostki wielowymiarowej - OLAP. Tam też znajduje się wynikowa struktura tabel, która odpowiada transformacji danych w początkowych etapach. Ostatni etap obejmował wykorzystanie utworzonej kostki do przeprowadzenia analiz wielowymiarowych, których rezultaty przedstawione zostały w rozdziale 4.

2 Proces ETL

W tym podrozdziale opisano zadania wraz z wykorzystanymi metodami, które pozwoliły na realizację całego procesu ETL. Projekt został wykonany w oparciu o szablony wygenerowany przy pomocy modułu SSIS.

2.1 Połączenie z plikami CSV oraz bazą danych

Wejściem procesu ETL są pliki CSV, w których znajdują się analizowane dane, natomiast wyjściem procesu jest baza danych, w której umieszczone zostaną wszystkie przetransformowane wyniki.

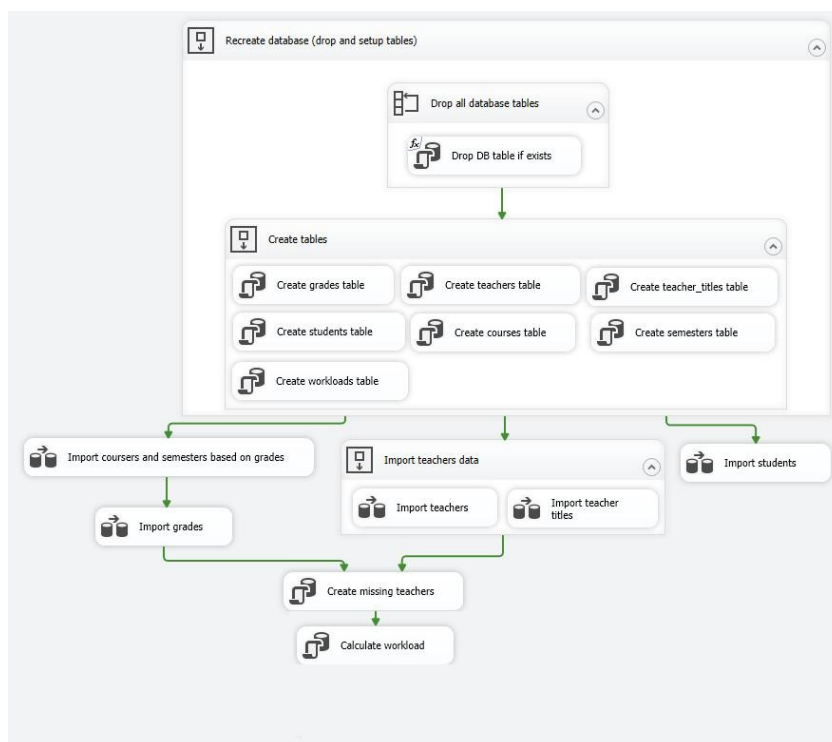
W celu połączenie z zewnętrznymi zasobami utworzono sześć odpowiednich połączeń w sekcji *Connection Managers*, gdzie pliki CSV przetwarzane były z wykorzystaniem typu *Flat File* natomiast połączenie z bazą odbywało się przy pomocy Managera typu *Ole DB*. W ustawieniach połączeń z plikami tekstowymi zdefiniowano również odpowiednie typy dla każdej z kolumn z źródłowych plików CSV.



Rysunek 1: Sekcja Connection Managers

2.2 Struktura procesu - Control Flow

Poniżej przedstawiono strukturę kolejnych kroków, które odpowiadają za realizację całego procesu ETL. Na schemacie widać zależności pomiędzy kolejnymi zadaniami, które pozwalają na odpowiednią transformację danych. W całym przebiegu zawarto również zadania odpowiedzialne za tworzenie oraz usuwanie tabel z bazy danych tak aby zapewnić pełną obsługę z poziomu procesu ETL i brak potrzeby bezpośredniej ingerencji w silnik bazodanowy.



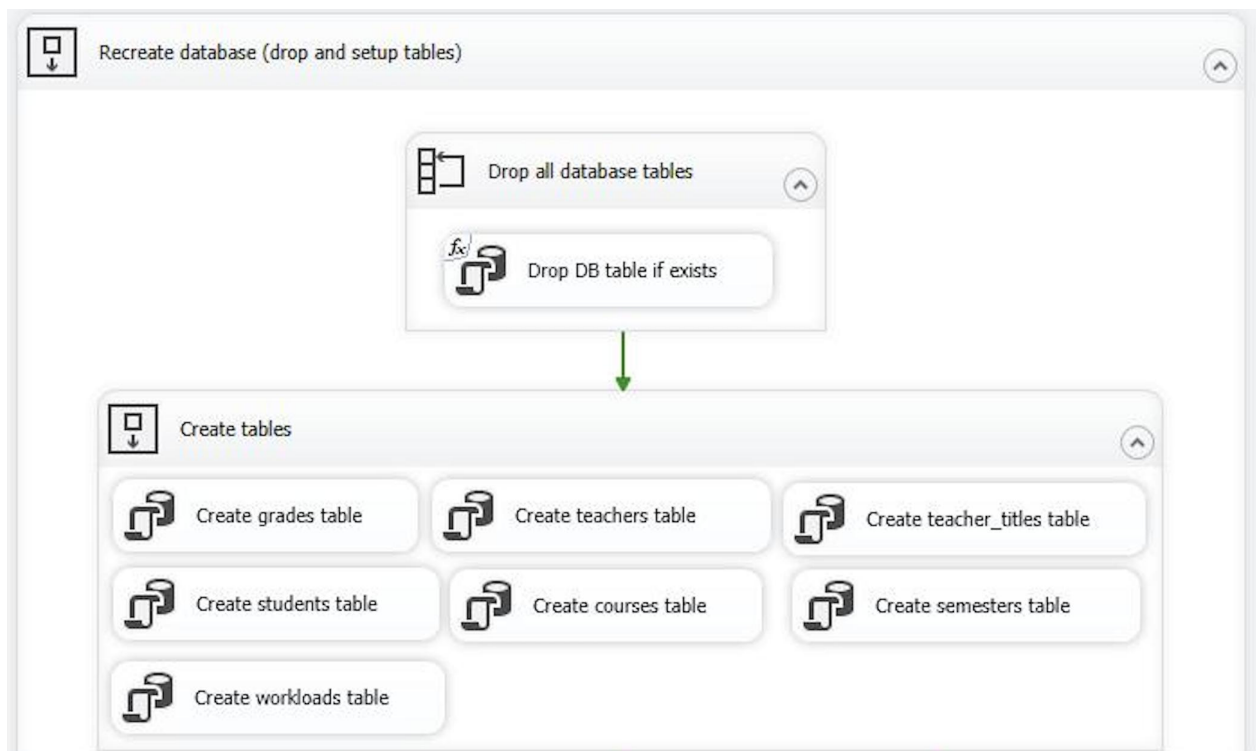
Rysunek 2: Przebieg procesu ETL - Control flow

Cały proces został zbudowany z wykorzystaniem następujących bloków:

1. *Sequence container* - pozwala na grupowanie zadań z tej samej domeny i wykonywanie ich wedle żądania.
2. *Foreach loop container* - wykonywanie parametryzowanych zadań dla każdego z elementów zdefiniowanej kolekcji.
3. *Data flow task* - transformacja danych złożona z wielu zadań.
4. *Execute SQL task* - wykonanie kwerendy na bazie danych.

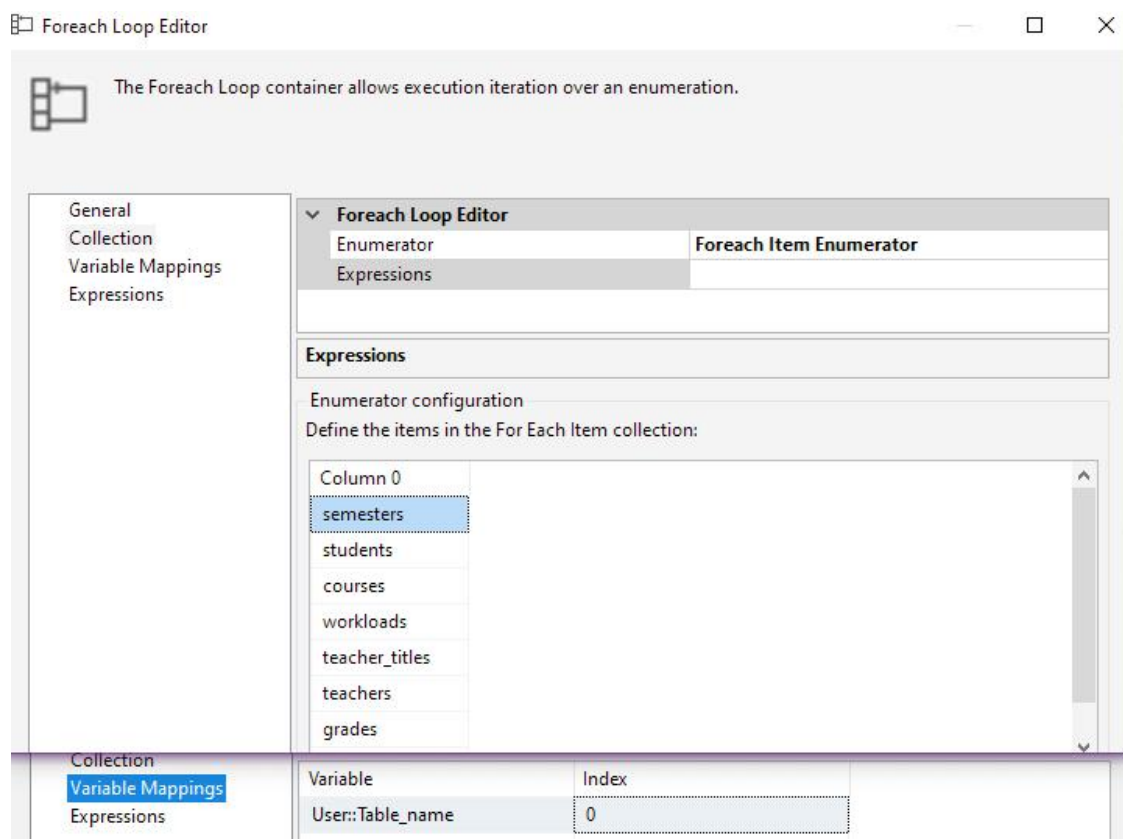
2.2.1 Tworzenie i usuwanie tabel w bazie danych

Aby uniknąć potrzeby manualnego tworzenia tabel w bazie danych oraz duplikacji danych w bazie podczas wielokrotnego uruchamiania pakietu, zdefiniowano zadania, które usuwają tabele jeśli istnieją oraz tworzą nowe z odpowiednimi typami danych dla każdej z kolumn.

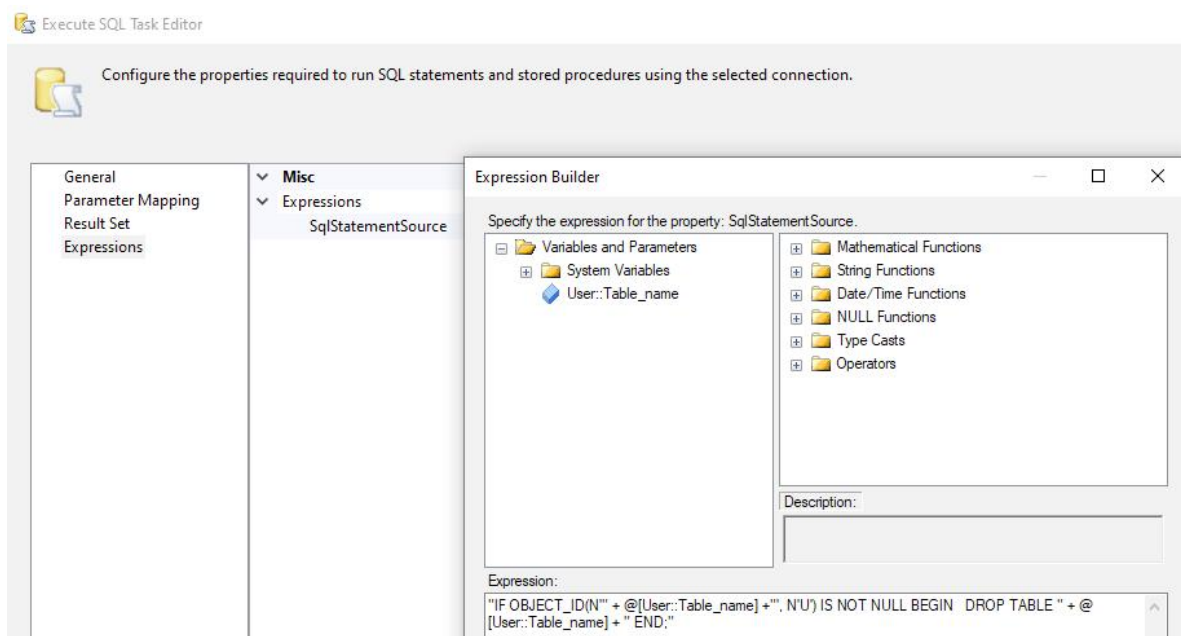


Rysunek 3: Tworzenie i usuwanie tabel w bazie danych

Czyszczenie bazy danych zostało zrealizowane przy pomocy *Foreach loop container*, który pozwolił na zdefiniowanie nazw tabel w kolekcji i następnie w trakcie każdej z iteracji przekazanie parametru *User::Table_name* do zadania *Drop DB table if exist*, który na bazie otrzymanej danej usuwał daną tabelę.

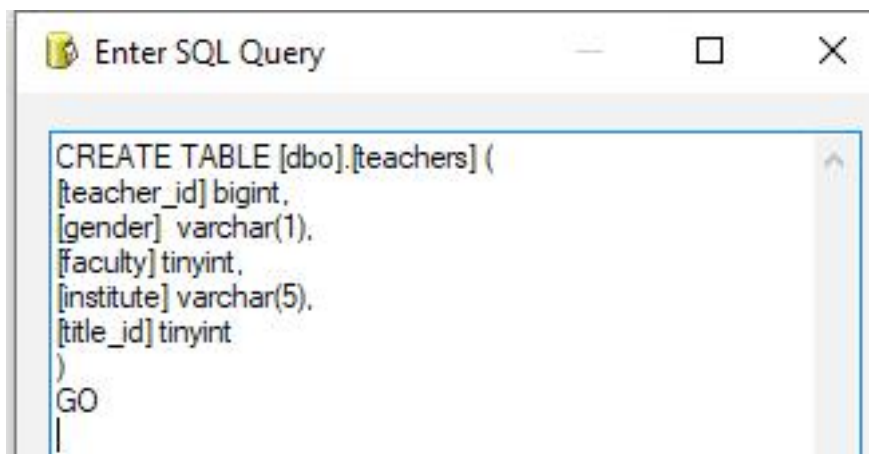


Rysunek 4: Definicja kontenera foreach loop container



Rysunek 5: Dynamiczne budowanie kwerendy SQL z wykorzystaniem parametru

Za tworzenie każdej z tabel odpowiadały zadania *Create table*, które określały również poprawny typ kolumn w bazie. Przykład jednego z nich przedstawiono na rysunku 6.

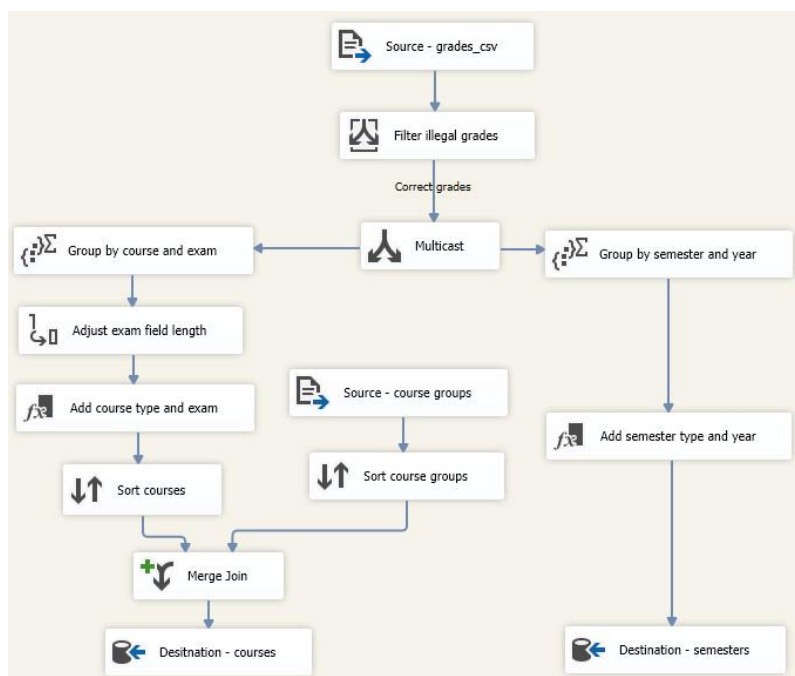


Rysunek 6: Tworzenie tabeli z nauczycielami

2.2.2 Importowanie kursów i semestrów - Data flow

Zadanie transformuje dane o ocenach z pliku Grades CSV do dwóch tabel *Semesters* i *Courses* tak aby w tabeli faktów w późniejszym etapie zostały tylko oceny oraz klucze obce do reszty wymiarów.

Dodatkowo w procesie filtrujemy niepoprawne oceny, transformujemy dane do poprawnych formatów oraz dokonujemy odpowiedniej agregacji wraz z wyliczeniem dodatkowych wartości.



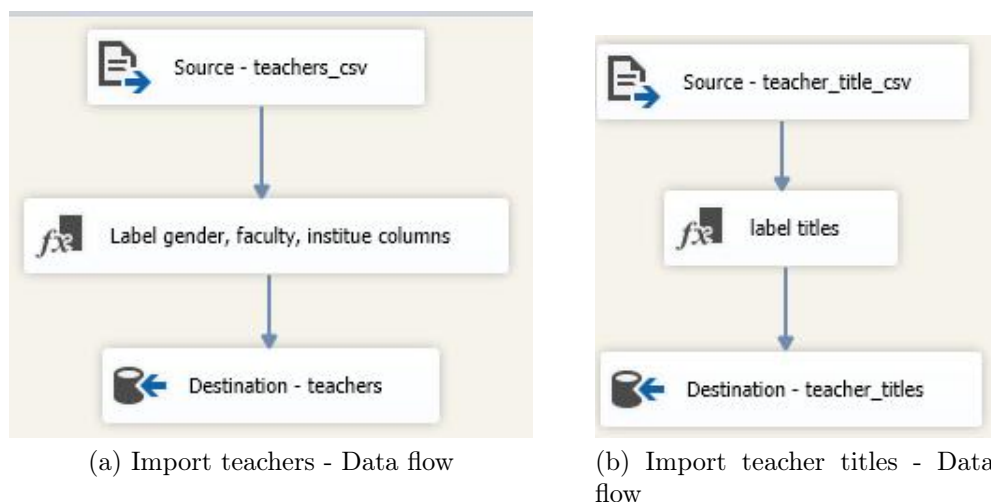
Rysunek 7: Importowanie danych dotyczących semestrów i kursów

Zadanie *Add course type and exam* dodaje nową wartość typ kursu (L,W,S itp.) oraz naprawia kodowanie egzaminu poprzez zakodowanie braku zmiennej jako *PASSED* natomiast wartość *E* została zastąpiona przez *EXAM*. Natomiast w zadaniu *Add semester type and year* wyliczamy na bazie semestru rok studiów oraz typ - *SUMMER* lub *WINTER*.

Dane pochodzące z pliku *course_group.csv* zostały uwzględnione w tabeli *Courses*, ponieważ zawierały one wyłącznie jedną daną w postaci grupy kursów.

2.2.3 Import teachers data - Sequence container

Kontener odpowiada za ładowanie danych dotyczących nauczycieli do bazy danych.



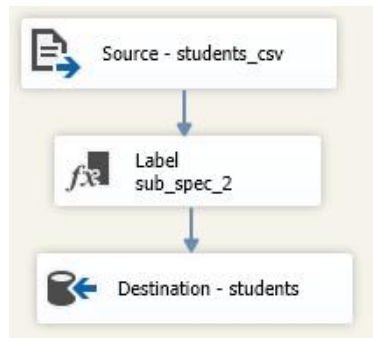
Rysunek 8: Zadania Data flow wchodzące w skład kontenera Import teachers data

W zadaniu *Import teachers* dokonujemy kodowania pól *faculty* i *institute* jako *NULL*, jeśli nie istnieją w danych wejściowych oraz uspójniamy kodowanie płci zastępując je literami *M* oraz *K*.

Natomiast w zadaniu *Import teacher titles* naprawiamy kodowanie kolumny *title_long* oraz dodajemy wartości *NULL* dla tych, które nie mają wypełnionego pola *title*.

2.2.4 Import students - Data flow

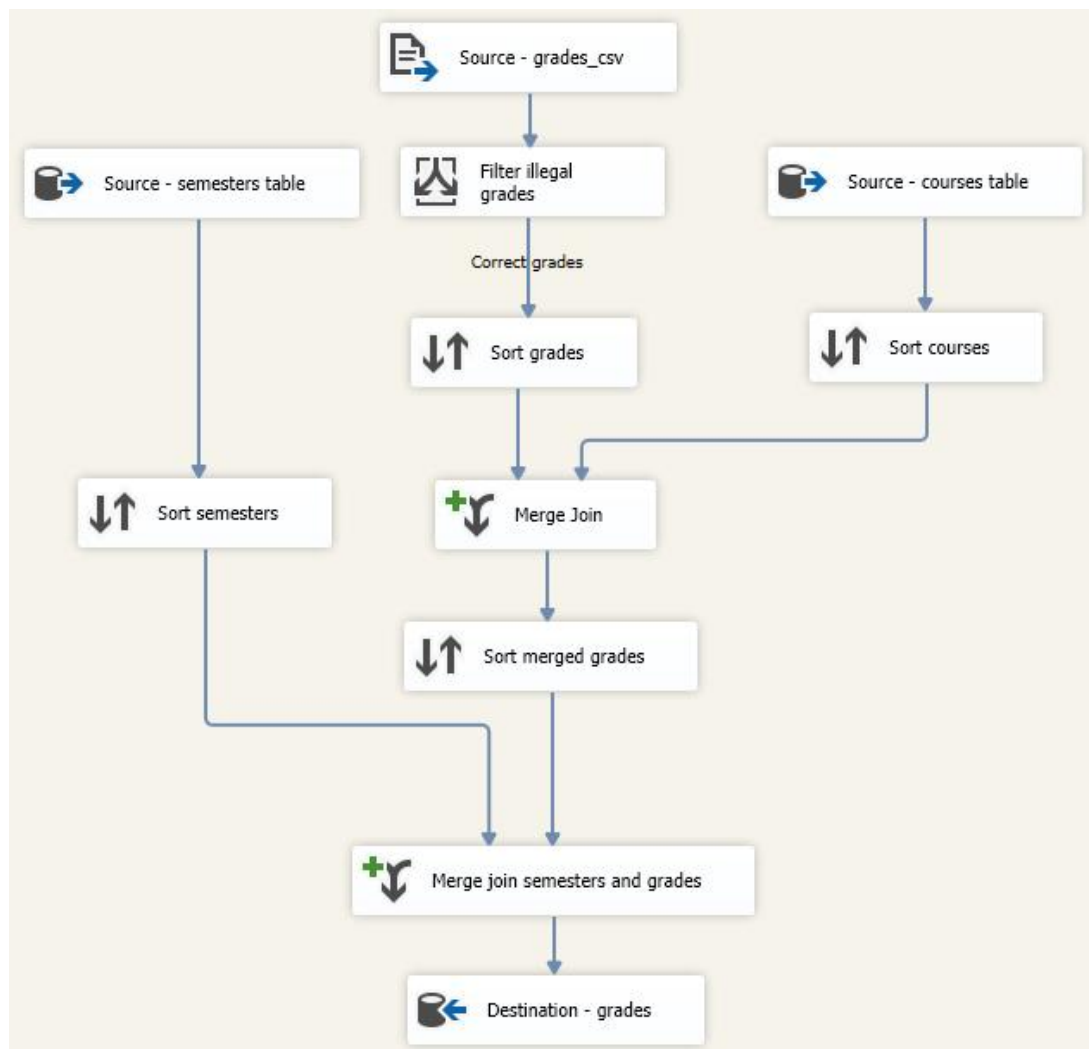
W zadaniu wczytujemy dane studentów z pliku *students.csv*, dodajemy wartości *NULL* dla kolumny *sub_spec2* gdy jest ona pusta i całość ładujemy do tabeli *Students* w bazie danych.



Rysunek 9: Importowanie danych dotyczących studentów

2.2.5 Import grades - Data flow

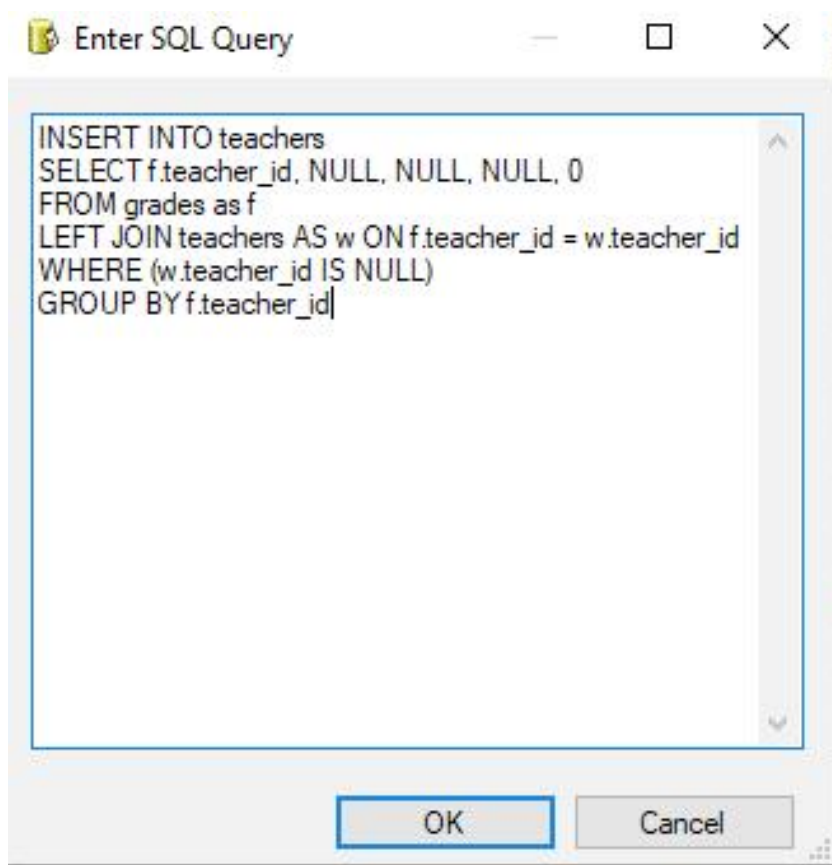
Po wczytaniu wymiarów, do których odwoływać się będzie tabela faktów, ładujemy do bazy danych oceny, ustawiając odpowiednio wartości kluczy obcych dla każdego z rekordów.



Rysunek 10: Importowanie danych dotyczących ocen

2.2.6 Create missing teachers - Execute SQL task

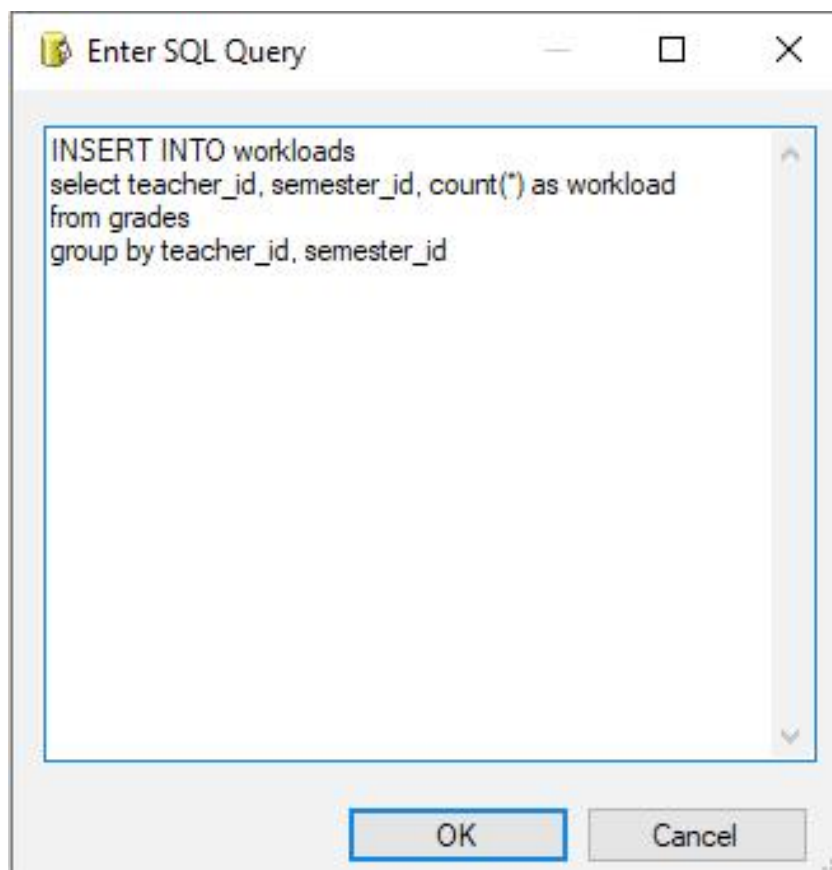
Zadanie jest odpowiedzialne za naprawienie relacji ocena - nauczyciel tak aby zachować spójność danych pochodzących z tabeli faktów. Relacje zostały uspołnione wykorzystując metodę zaproponowaną w instrukcji laboratoryjnej, to znaczy dla każdej oceny, która posiadała przypisany *teacher_id* nieznajdujący się w tabeli *Teachers* utworzono nowy rekord z wartościami *NULL*. Dla kolumny *title_id* dodano wartość 0, ponieważ odpowiada ona rekordowi z tabeli *Teacher titles*, który zarówno pole *title* jak i *title_long* posiada ustawione na *NULL*.



Rysunek 11: Naprawa relacji ocena nauczyciel

2.2.7 Calculate workload - Execute SQL task

Zadanie wylicza obciążenie dla każdego prowadzącego definiowane jako ilość wystawionych ocen w danym semestrze i roku. Dane zostały już odpowiednio zagregowane w związku z czym obciążenie może zostać wyliczone wyłącznie na bazie tabeli *Grades* przy pomocy kwerendy SQL.



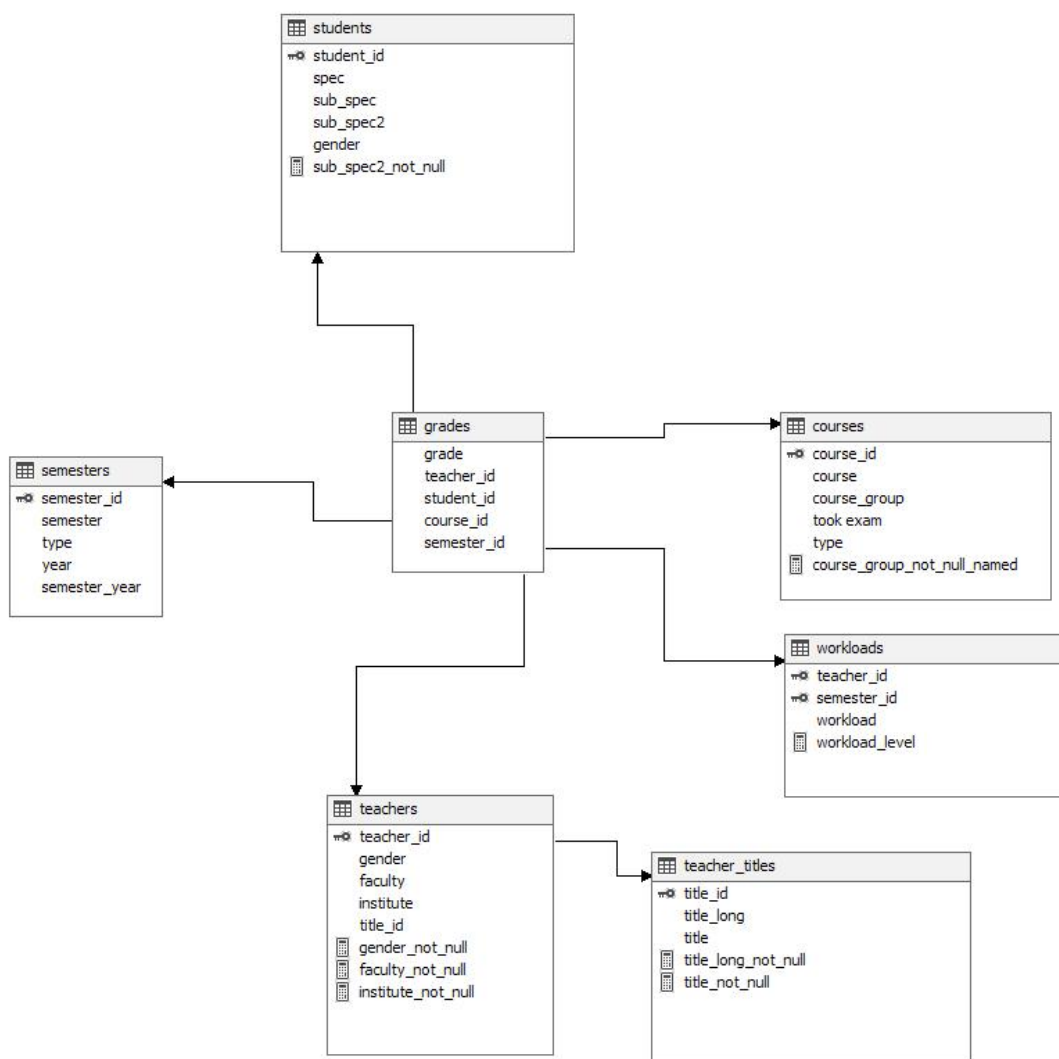
Rysunek 12: Wyliczenie obciążenia dla nauczycieli

3 OLAP

Po poprawnym załadowaniu danych do bazy danych, utworzono projekt w oparciu o szablon SSAS, który odpowiadał za utworzenie wielowymiarowej kostki. W sekcji tej zaprezentowano końcowe relacje pomiędzy obiektami, wyliczone kalkulacje, kostkę oraz dodane w niej miary i hierarchie.

3.1 Data Source View

W widoku bazy danych zdefiniowano relacje pomiędzy tabelami wymiarów i tabelą faktów na bazie odpowiednich kluczy.



Rysunek 13: Widok bazy danych wraz z relacjami

3.1.1 Named calculations

W bazie danych brakujące wartości zapisywane były z wykorzystaniem wartości *NULL*, która nie jest czytelna dla końcowego użytkownika, dlatego w widoku dodano również kolumny typu *Named Calculation*, które pozwoliły na bardziej czytelne kodowanie danych. Wszystkie z kalkulek kończące się frazą *not_null* mają analogiczną postać do tej przedstawionej na przykładzie 14.

The screenshot shows a dialog box titled "Edit Named Calculation". It has three main sections: "Column name:", "Description:", and "Expression:". The "Column name:" field contains the text "institute_not_null". The "Description:" field is empty. The "Expression:" field contains a SQL CASE statement: `CASE WHEN institute IS NULL THEN 'unknown' ELSE CAST(institute as nvarchar) END`. At the bottom, there are three buttons: "OK", "Cancel", and "Help".

Rysunek 14: Kodowanie wartości *NULL* w widoku bazy danych

Dodatkowo utworzono kalkulację *course_group_not_null_named*, która kodowała kody kursów z wartości liczbowych na tekstowe zgodnie z reprezentacją z instrukcji.

The screenshot shows a dialog box titled "Edit Named Calculation". It has three main sections: "Column name:", "Description:", and "Expression:". The "Column name:" field contains the text "course_group_not_null_named". The "Description:" field is empty. The "Expression:" field contains a SQL CASE statement: `CASE WHEN course_group IS NULL THEN 'unknown' WHEN course_group = 1 THEN 'faculty courses' WHEN course_group = 2 THEN 'sports' WHEN course_group = 3 THEN 'languages' WHEN course_group = 4 THEN 'humanities' ELSE CAST(course_group AS nvarchar) END`. At the bottom, there are three buttons: "OK", "Cancel", and "Help".

Rysunek 15: Kodowanie wartości dla grupy kursu w widoku bazy danych

Ostatnią z dodatkowych kolumn jest *workload_level*, która agreguje obciążenie prowadzącego aby końcowa analiza była bardziej czytelna dla użytkownika. Przedziały dla wartości zostały dobrane eksperymentalnie.

Column name: workload_level

Description:

Expression:

```

CASE
  WHEN workload < 30 THEN 'low'
  WHEN workload >= 30 and workload < 60 THEN 'neutral'
  WHEN workload >= 60 and workload < 100 THEN 'medium'
  WHEN workload >= 100 THEN 'high'
END

```

OK Cancel Help

Rysunek 16: Przedziały wartości dla obciążenia prowadzących

Jako alternatywę dla zaprezentowanego grupowania obciążenia prowadzących wykorzystano również *Property* na atrybucie wymiaru *Workloads* o nazwie *Workload*. Ustawienie wartości *DiscretizationBucketCount* oraz *DiscretizationMethod* pozwoliło na automatyczną agregację danych w przedziały wybrane na bazie danego algorytmu. Na rysunku 17 przedstawione zastosowane ustawienia, natomiast w tabeli 4 zobaczyć można wynik przeprowadzonej przez narzędzie dynamicznej dyskretyzacji.

Properties

Workload DimensionAttribute

DiscretizationBucketCount 4

DiscretizationMethod Automatic

Rysunek 17: Automatyczna dyskretyzacja na bazie parametrów

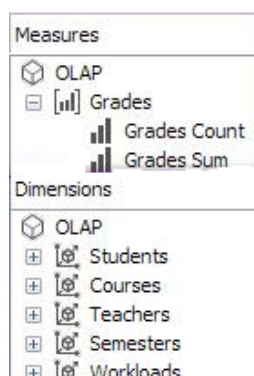
3.2 Kostka wielowymiarowa

Kolejnym krokiem po utworzeniu widoku bazy danych było zdefiniowanie kostki w której tabelą faktów była tabela *Grades*, natomiast pozostałe wykorzystane zostały jako wymiary.

3.2.1 Miary i wymiary kostki

W kostce utworzono dwa wymiary *Grades Count* oraz *Grades Sum*, które bezpośrednio zależały od tabeli faktów oraz jeden wymiar [*Grades average*], który był wyliczany na bazie

poprzednich. Miary *Grades Count* oraz *Grades Sum* zostały ukryte w wynikowej kostce. Dodatkowo w wymiarach ukryto klucze obce do pozostałych tabel oraz parametry nieistotne podczas końcowej analizy.



Rysunek 18: Miary i wymiary w kostce

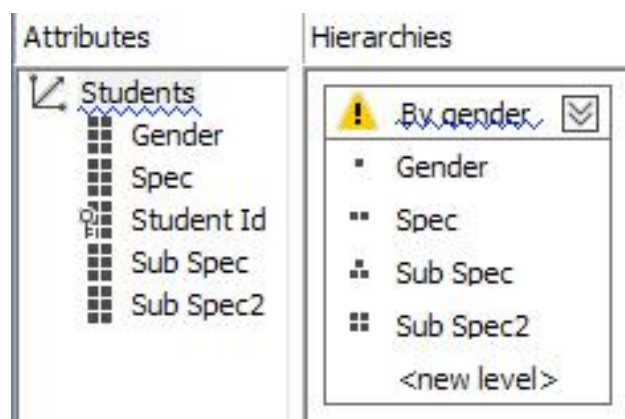
Miara *[Grades average]* została utworzona z wykorzystaniem mechanizmu *Calculated Member*, gdzie w polu expression umieszczono następujące wyrażenie:

`[Measures].[Grades Sum]/[Measures].[Grades Count]`

3.2.2 Hierarchie atrybutów

Aby umożliwić łatwiejszą analizę ocen utworzono zgodnie z wymaganiami z instrukcji hierarchie, które pozwalają na szybsze generowanie wybranych zestawów danych.

3.2.2.1 Hierarchia wymiaru Students



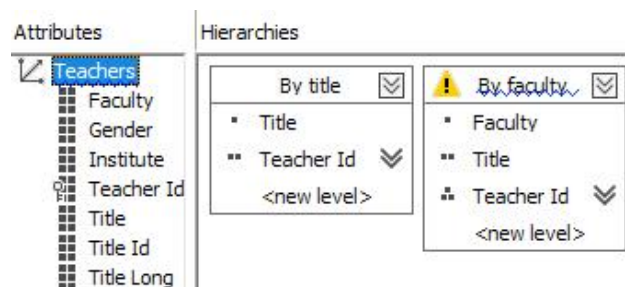
Rysunek 19: Hierarchia wymiaru Students

3.2.2.2 Hierarchia wymiaru Courses



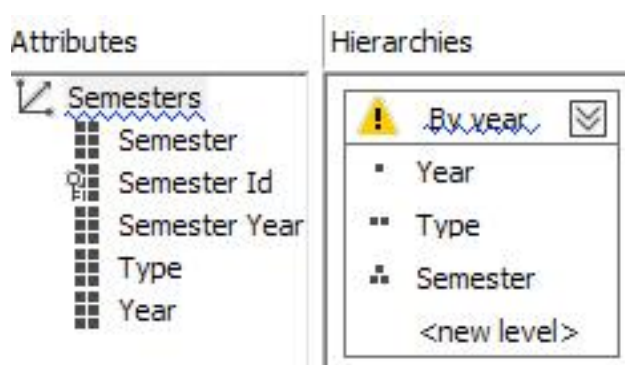
Rysunek 20: Hierarchia wymiaru Courses

3.2.2.3 Hierarchia wymiaru Teachers



Rysunek 21: Hierarchia wymiaru Teachers

3.2.2.4 Hierarchia wymiaru Semesters



Rysunek 22: Hierarchia wymiaru Semesters

4 Analiza danych

Ostatni etap polegał na znalezieniu zależności pomiędzy ocenami a wymiarami w funkcji, których analizować możemy dane. Analiza przeprowadzona została po podłączeniu kostki do narzędzia Excel.

4.1 Kto ocenia surowiej?

Z tabeli 1 wynika, że wykładowcy wystawiają oceny statystycznie wyższe dla obu płci. Dodatkowo płć żeńska uzyskuje lepsze oceny zarówno od płci męskiej jak i żeńskiej. Fakt ten potwierdza również średnia ocen dla nauczycieli, których płci nie znamy (w głównej mierze są to rekordy nauczycieli wygenerowane dla ocen, które powstały w procesie ETL podczas naprawy relacji ocena brakujący nauczyciel).

Płeć nauczycieli i studentów	Grades average
Nauczycielki (K)	4.195
Studentki (K)	4.362
Studenci (M)	4.189
Nauczyciele (M)	4.131
Studentki (K)	4.166
Studenci (M)	4.130
unknown	4.357
Studentki (K)	4.805
Studenci (M)	4.342

Tablica 1: Ocenę wystawiane przez wykładowców studentkom i studentom

4.2 Czy tytuł ma wpływ na wystawianą ocenę?

Z tabeli 2 wynika, że statystycznie najgorsze oceny wystawiają wykładowcy z tytułem **dr hab. inż.**, natomiast najlepsze **prof. ndzw. dr hab. inż.**.

Tytuł prowadzącego/prowadzącej	Grades average
doc. dr inż.	4.179
dr	3.946
dr hab.	4.069
dr hab. inż.	3.835
dr inż.	4.108
mgr	4.086
mgr inż.	4.266
prof. dr hab.	4.119
prof. dr hab. inż.	4.085
prof. dr inż.	3.988
prof. nadz. dr hab. inż.	3.923
prof. ndzw. dr hab. inż.	4.328
prof. PWr dr hab. inż.	4.202
unknown	4.347

Tablica 2: Ocenę wystawiane w zależności od tytułu naukowego

4.3 Obciążenie oraz wydział nauczyciela

Analizując obciążenie nauczycieli ze względu na dany wydział, zauważyć możemy, że najwięcej ocen wystawiają nauczyciele z wydziału **11** oraz **4** oraz, że oceny wystawiane na wydziale **11** są statystycznie znacznie gorsze od tych z wydziału **4**. Wywnioskować również można, że nauczyciele z małym obciążeniem z wydziału **8** oraz **23** wystawiają znacznie lepsze oceny niż wykładowcy również z niewielkim obciążeniem z wydziałów **2** i **21**.

Obciążenie i wydział	Grades average
high	4.026
11	3.564
4	4.089
medium	4.097
11	3.701
2	3.782
4	4.152
unknown	4.188
8	4.457
neutral	4.223
11	3.651
4	4.236
21	4.332
unknown	4.534
8	4.580
23	4.844
low	4.262
11	3.754
2	4.102
21	4.129
4	4.158
10	4.375
unknown	4.417
8	4.560
23	4.854

Tablica 3: Oceny wystawiane w zależności od obciążenia nauczycieli w danym wydziale na bazie *workload_level*

Ta sama analiza z wykorzystaniem mechanizmu opisanego na rysunku 17 przyniosła nieco inne rezultaty ze względu na automatyczną dyskretyzację pola *workload*, która bardziej uwidacznia różnice w obciążeniu nauczycieli na danych wydziałach.

Obciążenie i wydział	Grades average
1 - 23	4.257
10	4.375
11	3.752
2	4.102
21	4.136
23	4.854
4	4.129
8	4.560
unknown	4.412
179 - 554	3.883
11	3.456
4	3.924
24 - 88	4.181
11	3.678
2	3.782
21	4.203
23	4.848
4	4.200
8	4.549
unknown	4.462
89 - 178	4.100
11	3.607
4	4.177

Tablica 4: Oceny wystawiane w zależności od obciążenia nauczycieli w danym wydziale na bazie automatycznej dyskretyzacji pola workload

4.4 Oceny w zależności od płci i kierunku

Z analizy wywnioskować możemy, że statystycznie lepsze oceny osiągają studenci kierunku **INF** od studentów **AIR** oraz **EIT**.

Płeć i kierunek	Grades average
K	4.215
AIR	4.035
EIT	4.210
INF	4.682
M	4.145
AIR	4.113
EIT	4.102
INF	4.263

Tablica 5: Oceny w zależności od płci i kierunku studentów

4.5 Oceny w zależności od typu zajęć i formy zaliczenia

Z tabeli 6 wywnioskować możemy, że najlepsze oceny można było osiągnąć z seminarium, natomiast zaliczenie poprzez egzamin skutkowało gorszą średnią ocen zarówno na ćwiczeniach jak i wykładach.

Forma zajęć i zaliczenie	Grades average
C	4.217
EXAM	4.141
PASSED	4.233
L	4.316
P	4.393
S	4.659
W	3.978
EXAM	3.839
PASSED	4.070

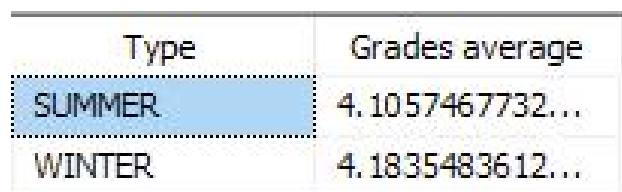
Tablica 6: Oceny w zależności od formy kursu i typu zaliczenia

4.6 Zapytania MDX

Zawarte poniżej przykładowe zapytania *MDX*, służyły głównie przetestowaniu składni języka, ponieważ Excel umożliwiał łatwiejszą analizę wybranych wymiarów.

4.6.1 Średnia ocen w zależności od typu semestru

```
SELECT
  {[Measures].[Grades average]} ON COLUMNS,
  {[Semesters].[Type].&[SUMMER], [Semesters].[Type].&[WINTER]} ON ROWS
FROM [OLAP]
```



Type	Grades average
SUMMER	4.1057467732...
WINTER	4.1835483612...

Rysunek 23: Średnia ocen w zależności od typu semestru

4.6.2 Średnia ocen w zależności od grupy kursu

```
SELECT
  {[Measures].[Grades average]} ON COLUMNS,
  NON EMPTY {([Courses].[Course Group].[Course Group].ALLMEMBERS)} ON ROWS
FROM [OLAP]
```

Course Group	Grades average
faculty courses	4.0035874817...
humanities	4.6426793861...
languages	4.1345197592...
sports	4.8535262206...
unknown	4.4376502191...

Rysunek 24: Średnia ocen w zależności od grupy kursu