Analityka i eksploracja danych - OLAP

prowadzący: dr hab. inż. Henryk Maciejewski

Jarosław Ciołek-Żelechowski(*218386*)

27.01.2020

1 Cel projektu

Celem projektu była wielowymiarowa analiza ocen uzyskanych przez studentów na przestrzeni lat - w szczególności analizy "ciekawych" zależności które miały wpływ na ocenę . Do jej wykonania skorzystaliśmy z narzędzi z rodziny MS SQL Server 2017:

- **Integration Services (SSIS)** - do przeprowadzenia pierwszej części projektu polegającego na wczytaniu i obróbce danych
- **Analysis Services (SSAS)** - do projektu i budowy wielowymiarowego modelu danych(kostki)

2 Integration Services - obróbka danych

Celem pierwszego etapu prac jest przygotowanie danych do załadowania do kostki. Po utworzeniu projektu *Integration Services* dane zostały zaczytane z plików *.csv* i wczytane do bazy danych *218386*. Następnie zostały usunięte wszystkie niespójności między danymi(klucze obce), a na koniec zostały utworzone nowe tabele, bazujące na już zebranych danych. Poszczególne etapy zostały szczegółowo opisane poniżej.

2.1 Zaczytanie Danych

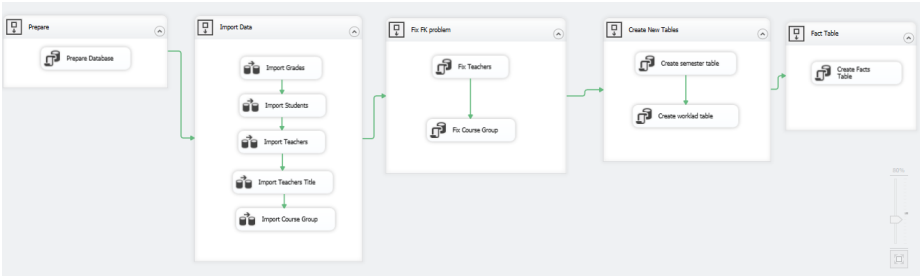
Do zaczytania danych z plików *.csv* wykorzystałem *Data Flow Taski*. Każdy plik posiada własny *Data Flow Task*, który wczytuje dane, obrabia je jeśli jest taka potrzeba, a następnie zapisuje do bazy danych.

Wykorzystane funkcje wewnątrz *Data Flow Tasków*:

- **Flat File Sorce** - wykorzystana do zdefiniowania pliku wejściowego,
- **Data Connector** - użyty do rozpoznania zawartości pliku i "przetłumaczeniu" go na poszczególne kolumny(o konkretnym typie danych),
- **Conditional Split** - wykorzystany do podziały danych(odsiania danych niepotrzebnych),
- **Derived Column** - użyte do stworzenia nowych kolumn, na podstawie informacji zawartych w innych miejscach,
- **DB Destination** - wykorzystane do zdefiniowania bazy danych, stworzenia odpowiedniej tabeli oraz zmapowaniu konkretnych kolumn.

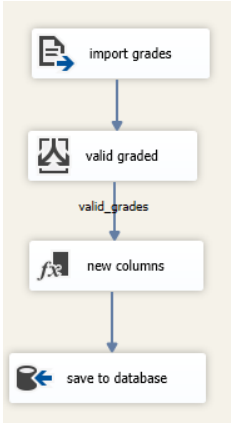
2.2 Struktura poszczególnych tasków

Poszczególne zadania rozbiłem na sekwencję, co można zobaczyć na rysunku poniżej. Pierwsza sekwencja jest odpowiedzialna za czyszczenie zawartości bazy danych(przy pomocy narzędzia *Execute SQL Task*) tak żeby nie powstały niepotrzebne duplikaty podczas kolejnych uruchomień. Struktura tego zapytania to polecenia *TRUNCATE* dla każdej z tabel.



Rysunek 1: Flow całego zadania ETL

2.2.1 Oceny



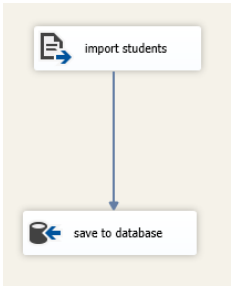
Rysunek 2: Flow Wczytania Ocen

- **Flat File Source** - wczytanie pliku *grades.csv*,
- **Conditional Split** - wybranie ocen $\geqslant 2$,
- **Derived Column** - zmiana pustych rekordów w kolumnie *exam* na *Z*(zaliczenie), oraz utworzenie kolumn *study_year* oraz *semestr_type*
- **DB Destination** - utworzenie tabeli *Grades*.

Derived Column Name	Derived Column	Expression
study_year	<add as new column>	semester % 2 == 0 ? semester / 2 : (semester / 2) + 1
semester_type	<add as new column>	semester % 2 == 0 ? "SUMMER" : "WINTER"
exam	Replace 'exam'	exam == "E" ? "E" : "Z"

Rysunek 3: *Derived Column* - TSQL

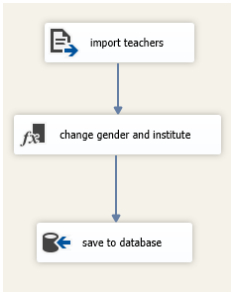
2.2.2 Studenci



Rysunek 4: Flow Wczytania Studentów

- **Flat File Source** - wczytanie pliku *students.csv*,
- **DB Destination** - utworzenie tabeli *Students*.

2.2.3 Nauczyciele



Rysunek 5: Flow Wczytania Nauczycieli

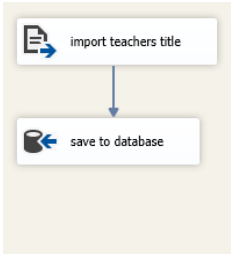
- **Flat File Source** - wczytanie pliku *teachers.csv*,
- **Derived Column** - zmiana zapisu płci z [1, 2] na [M, K](ujednolicenie względem Studentów) oraz zmiana pustych rekordów kolumny *institute* na 0,

- **DB Destination** - Utworzenie tabeli *Teachers*.

Derived Column Name	Derived Column	Expression
gender	Replace 'gender'	gender == "1" ? "M" : "K"
institute	Replace 'institute'	institute == "" ? "0" : institute

Rysunek 6: *Derived Column* - TSQL

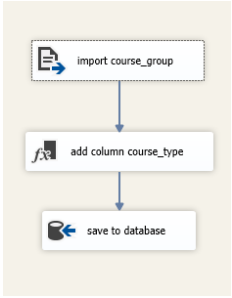
2.2.4 Tytuły Nauczycieli



Rysunek 7: Flow Wczytania Tytułów Nauczycieli

- **Flat File Source** - wczytanie pliku *teacher_title.csv*,
- **DB Destination** - utworzenie tabeli *Teacher_title*.

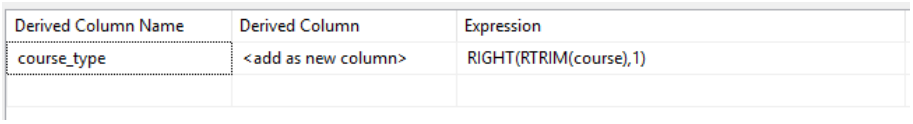
2.2.5 Kursy



Rysunek 8: Flow Wczytania Kursów

- **Flat File Source** - wczytanie pliku *course_group.csv*,

- **Derived Column** - dodanie nowej kolumny *course_type* przechowującej typ kursu,
- **DB Destination** - utworzenie tabeli *Course_group*.



Derived Column Name	Derived Column	Expression
course_type	<add as new column>	RIGHT(RTRIM(course),1)

Rysunek 9: *Derived Column* - TSQL

2.3 Usunięcie niespójności

Niespójności występowały pomiędzy tabelami *Grades* i *Teachers* oraz między tabelami *Grades* i *Course_group*. Zostały usuięte przez dwa *SQL Taski*.

```
INSERT INTO Teachers
SELECT g.teacher_id, 0, 0, 0, 0
FROM Grades as g
LEFT JOIN teachers AS t ON g.teacher_id = t.teacher_id
WHERE (t.teacher_id IS NULL)
GROUP BY g.teacher_id;
```

oraz

```
INSERT INTO Course_group
SELECT g.course, 0, 0
FROM Grades AS g
LEFT JOIN Course_group AS cg ON g.course = cg.course
WHERE (cg.course IS NULL)
GROUP BY g.course;
```

2.4 Stworzenie nowych tabel(w tym tabeli faktów)

Wszystkie opisane w tym punkcie zadania zostały zrealizowane przy pomocy *SQL Tasków*.

2.4.1 Semestr

Nowa tabela *Semester* zawiera informacje o numerze semestru, roku studiów oraz jego typie(zimowy czy letni).

```
SELECT DISTINCT semester, study_year, semester_type
INTO Semester
FROM Grades;
```

2.4.2 Workload

Nowa tabela *Workload* jest wyliczana jako ilość kursów przez danego prowadzącego w danym semestrze.

```
SELECT teacher_id, semester, COUNT(DISTINCT course) AS workload
INTO Workload
FROM Grades
GROUP BY semester, teacher_id;
```

2.4.3 Tabela Faktów

Do utworzenia nowej tabeli faktów *Grades_fact_table* potrzebne były trzy kroki. Pierwszy - utworzenie tymczasowej tabeli(*Grades_tmp*), która jest kopią tabeli *Grades*. Drugi - komenda tworząca nową tabelę wymiaru *Grades_desc* opisującą ocenę.Trzeci - tworzący nową tabelę faktów *Grades_fact_table* z kluczami obcymi do tabel wymiarów *Grades_desc*.

```
-- krok 1
SELECT IDENTITY (bigint, 1, 1) AS id, semester, year, course,
                                  teacher_id, grade, exam,
                                  student_id, study_year,
                                  semester_type
INTO Grades_tmp
FROM Grades;

-- krok 2
SELECT id, semester, year, grade, exam
INTO Grades_desc
FROM Grades_tmp;

-- krok 3
SELECT grade, id AS grade_desc_id, course, teacher_id,
                  student_id, semester
INTO Grades_fact_table
FROM Grades_tmp;
```

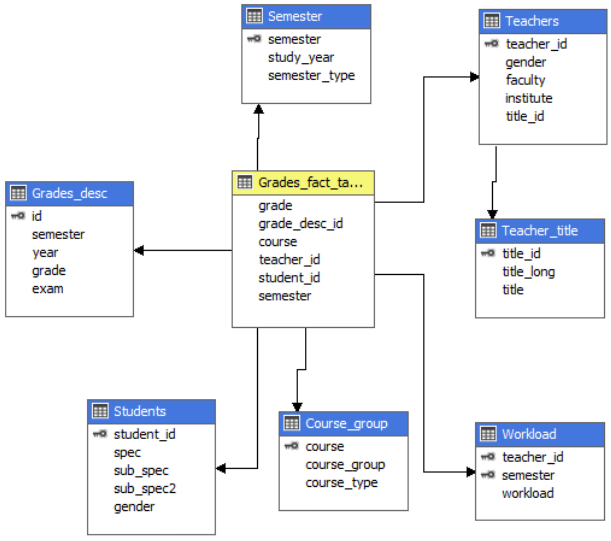
3 Analysis Services - budowa kostki

Celem drugiego etapu było stworzenie działającej kostki. W tym celu utworzyłem projekt *Analysis Services Project*, gdzie podałem jako źródło danych bazę danych stworzoną w części pierwszej.

3.1 Definicja relacji

Posiadając załadowaną bazę danych kolejnym krokiem było stworzenie jej widoku, czyli zdefiniowanie interesujących Nas relacji. Relacje zostały oczywiście zde-

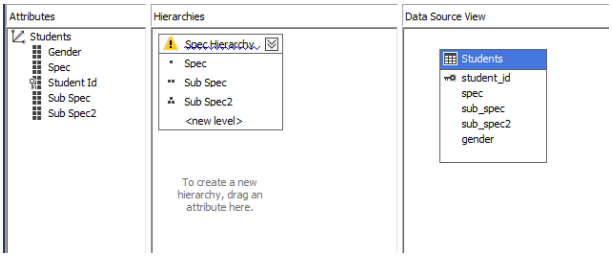
finiowane zgodnie z kluczami głównymi(zdefiniowanymi w tabelach wymiarów) i odpowiadającego im kluczą obcym(zdefiniowanymi w tabeli faktów). Wszystkie relacje widoczne są na zdjęciu poniżej.



Rysunek 10: Zdefiniowane relacje

3.2 Miary(nowa miara), wymiary i ich definicja

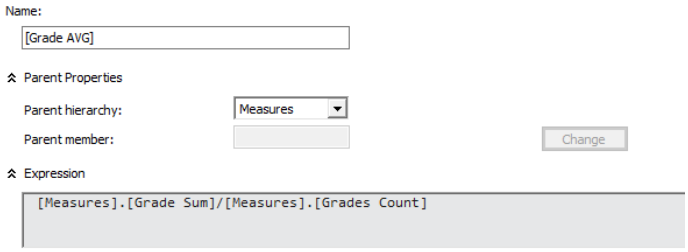
W dalszej części należy zdefiniować poszczególne wymiary. Wymiarami jest wszystko co nie jest kluczem głównym w tabelach wymiaru. Jedyną *ciekawostką* i interesującym elementem może być hierarchia specjalizacji, którą użyłem przy definicji wymiaru Studentów. Hierarchia jest po prostu agregacją kilku elementów.



Rysunek 11: Hierarchia Specjalizacji

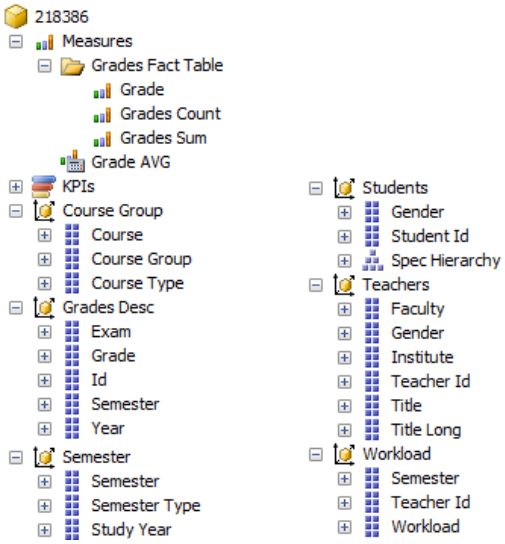
Na tym etapie dodałem również nową miarę - *Grade AVG*, którą wyliczyłem

z innych miar(Suma ocen / ilość).



Rysunek 12: Nowa Miara - *Grade AVG*

Całość prezentuje się następująco:



Rysunek 13: Zdefiniowane Miary i Wymiary

3.3 Prezentacja kostki

W tym punkcie chciałbym zaprezentować parę miar:

Gender	Grade AVG
K	4.16269455252918
M	4.10029617662897

Rysunek 14: Panie mają trochę lepszą średnią

Spec	Grade AVG
AIR	4.050030...
EIT	4.054185...
INF	4.247273...

Rysunek 15: Informatycy uczą się najlepiej!

Spec	Sub Spec	Sub Spec2	Grade AVG
AIR	ARK		4.013796...
AIR	ARR		3.758346...
AIR	ARS		4.177201...
EIT	EAE		4.195656...
EIT	EMA	EUE	4
EIT	EMS		3.910231...
EIT	EOT		4.226376...
EIT	ESA		4.015265...
EIT	ETA		3.897697...
EIT	EZI		4.114552...
EIT	TEL	TRU	4.208429...
EIT	TEL	TSC	3.801663...
EIT	TEL	TTD	3.927627...
EIT	TEL	TTP	4.270285...
EIT	TEL	TTR	3.907557...
EIT	TEL	TTS	3.882868...
INF	IMT		4.257006...
INF	INS	IIO	4.069972...
INF	INS	IMS	4.198302...
INF	INS	ISB	4.240639...
INF	INT	IBS	4.052877...
INF	INT	IRS	4.016009...
INF	ISK		4.452301...
INF	ISM		3.909010...

Rysunek 16: Ale jednak Ci z ISM mają najniższą średnią

Title Long	Grade AVG
	4.346035...
doc. dr in¾.	4.029914...
dr	3.958007...
dr hab.	4.050660...
dr hab. in¾.	3.624048...
dr in¾.	4.044747...
mgr	4.024469...
mgr in¾.	4.233254...
prof. dr hab.	4.027455...
prof. dr hab. in¾.	4.040045...
prof. dr in¾.	3.850909...
prof. nadz. dr hab. in¾.	3.965324...
prof. ndzw. dr hab. in¾.	4.328054...
prof. PWr dr hab. in¾.	4.255364...

Rysunek 17: Doktorzy dają gorsze oceny od magistrów i profesorów

Study Year	Grade AVG
1	3.802427...
2	3.933555...
3	4.066548...
4	4.233735...
5	4.544292...

Rysunek 18: Z roku na rok średnia rośnie!

Workload	Grade AVG
1	4.060488...
10	4.315909...
11	4.806358...
13	4
19	4.333333...
2	4.103421...
3	4.051499...
4	4.186798...
5	4.075084...
6	4.121134...
7	4.306049...
8	4.381054...
9	4.5625

Rysunek 19: od 1 do 6 *Workload* nie ma prawie żadnego wpływu na średnią wystawionych ocen. Między 7 a 11 oceny są ewidentnie lepsze, potem następuje jednak punkt przegięcia i wraz z wzrostem zapracowania średnia ocena spada