
BI / OLAP - ANALIZA OCEN

ANALITYKA I EKSPLOMACJA DANYCH

Autor: Piotr Szczypior
Kontakt: 264468@student.pwr.edu.pl
Data: 16 stycznia 2026
Prowadzący: Prof. dr hab. inż. Henryk Maciejewski

1 Wstęp

1.1 Cel i zakres projektu

Niniejsze sprawozdanie dokumentuje proces projektowania i implementacji systemu Business Intelligence opartego o technologie Microsoft SQL Server Integration Services (SSIS) oraz SQL Server Analysis Services (SSAS). Głównym celem projektu było stworzenie kostki analitycznej umożliwiającej wielowymiarową analizę danych dotyczących ocen studentów z kierunku na Wydziale Elektronicznym Politechniki Wrocławskiej.

System został zaprojektowany w celu umożliwienia analizy wyników akademickich w różnych przekrojach, takich jak studenci, prowadzący zajęcia, rodzaje egzaminów oraz kursy. Implementacja obejmuje kompletny proces ETL (Extract, Transform, Load) realizowany w środowisku SSIS oraz wielowymiarową kostkę OLAP (Online Analytical Processing) zbudowaną w technologii SSAS.

Projekt pozwala na agregację i eksplorację danych akademickich, wspierając procesy podejmowania decyzji oraz identyfikację trendów w wynikach nauczania.

1.2 Źródła danych

Dane wejściowe do systemu zostały przygotowane w formacie CSV i obejmują następujące zbiory:

- `course_group.csv` – dane grup kursów
- `grades.csv` – dane z ocenami
- `students.csv` – dane o studentach
- `teacher_title.csv` – dane z tytułami nauczycieli
- `teachers.csv` – dane nauczycieli

2 ETL

W ramach realizacji systemu Business Intelligence opracowano proces ETL (Extract, Transform, Load), odpowiedzialny za pozyskanie, przetworzenie oraz załadowanie danych do hurtowni. Dane źródłowe w formacie CSV zostały poddane obróbce, obejmującej ekstrakcję z plików, walidację oraz transformację do struktury zgodnej z modelem analitycznym.

Implementacja procesu ETL obejmowała następujące etapy:

1. **Stworzenie schematu bazy danych** – utworzenie struktury relacyjnej bazy danych w środowisku Microsoft SQL Server.
2. **Załadowanie danych źródłowych** – wykorzystanie narzędzia SQL Server Integration Services (SSIS) do automatycznego importu danych z plików CSV do tabel pośrednich (staging tables) w bazie danych.
3. **Czyszczenie i walidacja danych** – identyfikacja oraz usunięcie niespójności, duplikatów i wartości odstających w danych źródłowych, a także standaryzacja formatów i uzupełnienie brakujących wartości.
4. **Transformacja i załadowanie do hurtowni** – przekształcenie danych z tabel pośrednich do docelowego modelu wymiarowego, obejmującego tabele wymiarów i tabele faktów, stanowiącego podstawę dla kostki analitycznej SSAS.

Wszystkie operacje ETL zostały zautomatyzowane w postaci pakietów SSIS, w celu zapewnienia powtarzalności procesu oraz możliwość regularnej aktualizacji danych w systemie.

2.1 Stworzenie tabel Staging

W pierwszym etapie procesu ETL utworzono tabele stagingowe, pełniące rolę warstwy pośredniej pomiędzy źródłowymi plikami CSV a docelową strukturą hurtowni danych. Tabele staging służą do bezpośredniego załadowania surowych danych z plików zewnętrznych, umożliwiając ich późniejsze czyszczenie, walidację oraz transformację bez ingerencji w dane źródłowe. Strukturę tabel stagingowych zaprojektowano w taki sposób, aby odzwierciedlała format danych wejściowych, jednocześnie umożliwiając elastyczne przetwarzanie w kolejnych etapach ETL.

Przykładowa struktura tabeli stagingowej dla danych o ocenach:

```
CREATE TABLE [stg_grades] (  
    [semester] tinyint,  
    [year] int,  
    [course] varchar(10),  
    [teacher_id] bigint,  
    [grade] real,  
    [exam] varchar(1),  
    [student_id] bigint  
);
```

2.2 Czyszczenie danych

Tabela Teacher

Kolumna `gender` w pliku `teachers.csv` przyjmowała wartości 0 i 1. W celu ujednolicenia reprezentacji danych dokonano konwersji na typ znakowy oraz mapowania wartości na oznaczenia M (Man) i F (Female).

```
ALTER TABLE stg_teachers  
ALTER COLUMN gender VARCHAR(1);
```

Następnie przeprowadzono aktualizację wartości w kolumnie `gender` według ustalonego schematu mapowania:

```
UPDATE stg_teachers  
SET gender = CASE  
    WHEN gender = '1' THEN 'M'  
    WHEN gender = '2' THEN 'K'  
    ELSE 'U'  
END;
```

W celu zachowania spójności danych uzupełniono brakujące wartości w kolumnie `institute`, przypisując oznaczenie UNK (unknown) dla rekordów z wartością NULL lub pustym ciągiem znaków:

```
UPDATE stg_teachers  
SET institute = 'UNK'  
WHERE institute IS NULL OR institute = '';
```

Analogicznie, dla kolumny `faculty` zastąpiono wartości NULL wartością domyślną -1, oznaczającą brak przypisania do wydziału:

```
UPDATE stg_teachers  
SET faculty = -1  
WHERE faculty IS NULL;
```

Tabela Grades

Usunięto rekordy zawierające nieprawidłowe wartości ocen (0 i 1), które nie mieszczą się w standardowej skali oceniania:

```
DELETE FROM stg_grades  
WHERE grade IN (0, 1);
```

Dla kolumny `exam` uzupełniono brakujące wartości domyślnym oznaczeniem 'T' (test):

```
UPDATE stg_grades  
SET exam = 'T'  
WHERE exam IS NULL OR exam = '';
```

W celu standaryzacji nazw kursów dokonano konwersji na wielkie litery:

```
UPDATE stg_grades  
SET course = UPPER(course);
```

Tabela Students

Uzupełniono brakujące wartości w kolumnie `sub_spec2` oznaczeniem 'NA' (not applicable):

```
UPDATE stg_students
SET sub_spec2 = 'NA'
WHERE sub_spec2 IS NULL OR sub_spec2 = '';
```

Tabela Course Group

Standaryzacja nazw kursów poprzez konwersję na wielkie litery:

```
UPDATE stg_course_group
SET course = UPPER(course);
```

2.3 Standaryzacja relacji między tabelami

W celu zapewnienia integralności referencyjnej wprowadzono rekord tytułu domyślnego dla nauczycieli bez przypisanego tytułu naukowego:

```
IF NOT EXISTS (SELECT 1 FROM stg_teacher_title WHERE title_id = 0)
BEGIN
    INSERT INTO stg_teacher_title (title_id, title_long, title)
    VALUES (0, 'Unknown Title', 'UNK');
END
```

Następnie uzupełniono tabelę `stg_teachers` o brakujące rekordy nauczycieli występujących w tabeli ocen, ale nieobecnych w tabeli nauczycieli. Rekordom tym przypisano wartości domyślne:

```
INSERT INTO stg_teachers (teacher_id, gender, faculty, institute, title_id)
SELECT g.teacher_id, 'U', NULL, 'UNK', 0
FROM stg_grades AS g
LEFT JOIN stg_teachers AS t ON g.teacher_id = t.teacher_id
WHERE t.teacher_id IS NULL
GROUP BY g.teacher_id;
```

Analogicznie uzupełniono tabelę `stg_course_group` o kursy występujące w ocenach, ale nieobecne w tabeli grup kursów. Przypisano im domyślną grupę o identyfikatorze 5:

```
INSERT INTO stg_course_group (course, course_group)
SELECT g.course, 5
FROM stg_grades AS g
LEFT JOIN stg_course_group AS cg ON g.course = cg.course
WHERE cg.course IS NULL
GROUP BY g.course;
```

2.4 Wzbogacenie danych o kolumny analityczne

W celu umożliwienia zaawansowanych analiz rozszerzono strukturę tabel o kolumny pochodne zawierające przetworzone wartości analityczne.

Tabela Grades

Dodano trzy nowe kolumny umożliwiające analizę w kontekście roku studiów oraz statusu zaliczenia:

```
ALTER TABLE stg_grades
ADD [semester_type] varchar(1),
    [study_year] int,
    [pass] bit;
```

Wypełniono kolumnę `semester_type` oznaczeniem typu semestru (W – zimowy, S – letni) na podstawie parzystości numeru semestru:

```
UPDATE stg_grades
SET semester_type = CASE
    WHEN semester % 2 = 1 THEN 'W'
    ELSE 'S'
END;
```

Obliczono rok studiów na podstawie numeru semestru:

```
UPDATE stg_grades
SET study_year = (semester + 1) / 2;
```

Dodano kolumnę binarną `pass` wskazującą status zaliczenia przedmiotu (1 – zaliczony, 0 – niezaliczony):

```
UPDATE stg_grades
SET pass = CASE
    WHEN grade >= 3.0 THEN 1
    WHEN grade = 2.0 THEN 0
END;
```

Tabela Course Group

Rozszerzono tabelę o kolumnę `course_type` identyfikującą typ zajęć na podstawie ostatniego znaku nazwy kursu:

```
ALTER TABLE stg_course_group
ADD course_type varchar(1);
```

Wypełniono kolumnę `course_type` według schematu mapowania typów zajęć (W – wykład, L – laboratorium, C – ćwiczenia, P – projekt, S – seminarium, E – egzamin, U – nieznany):

```
UPDATE stg_course_group
SET course_type = CASE
    WHEN course IS NULL THEN 'U'
    WHEN UPPER(RIGHT(course, 1)) NOT IN ('W', 'L', 'C', 'P', 'S', 'E')
    THEN 'U'
    ELSE UPPER(RIGHT(course, 1))
END;
```

2.5 Tworzenie tabel wymiarowych

Tabela Semester

Została utworzona nowa tabela wymiarowa `dim_semester` agregująca informacje o semestrach:

```
DROP TABLE IF EXISTS dim_semester;

CREATE TABLE [dim_semester] (
    [semester_id] bigint identity(1,1) primary key,
    [semester] int,
    [year] int,
    [semester_type] varchar(1),
    [study_year] int
);
```

załadowano dane do tabeli wymiarowej na podstawie unikalnych kombinacji wartości z tabeli `stg_grades`:

```
INSERT INTO dim_semester (semester, year, semester_type, study_year)
SELECT DISTINCT
    semester,
    year,
    semester_type,
    study_year
FROM stg_grades
ORDER BY year, semester;
```

Tabela Teacher

Pozostałe tabele wymiarów zostały utworzone na podstawie tabel pośrednich. Etap ten został przeprowadzony w celu standaryzacji nazw, zapewnienia odpowiedniej struktury tabel oraz utworzenia kluczy. Poniżej znajduje się przykładowy kod SQL dla tabeli wymiarowej `dim_teachers`.

```
DROP TABLE IF EXISTS dim_teachers;

CREATE TABLE dim_teachers (
    teacher_id BIGINT NOT NULL PRIMARY KEY,
    gender VARCHAR(1),
    faculty INT,
    institute VARCHAR(5),
    title_id INT
);
```

załadowano dane do tabeli wymiarowej na podstawie tabeli `stg_teachers`:

```
INSERT INTO dim_teachers (teacher_id, gender, faculty, institute, title_id)
SELECT teacher_id, gender, faculty, institute, title_id
FROM stg_teachers;
```

Podobny mechanizm został zastosowany dla tabel `stg_students`, `stg_course_group` oraz `stg_teacher_title`.

2.6 Tworzenie tabeli faktów

Utworzono tabelę faktów `fact_grades` będącą centralnym elementem modelu wymiarowego, przechowującą szczegółowe informacje o wystawionych ocenach wraz z kluczami obcymi do odpowiednich wymiarów:

```
DROP TABLE IF EXISTS fact_grades;

CREATE TABLE [fact_grades] (
    [grade_id] bigint identity(1,1) primary key,
    [student_id] bigint,
    [teacher_id] bigint,
    [semester_id] bigint,
    [course] varchar(50),
    [grade] decimal(3,1),
    [exam] varchar(1),
    [pass] bit,
);
```

załadowano dane do tabeli faktów na podstawie połączenia tabeli `stg_grades` z wymiarem `dim_semester`:

```
INSERT INTO fact_grades
(student_id, teacher_id, semester_id, course, grade, exam, pass)
SELECT
    g.student_id,
    g.teacher_id,
    s.semester_id,
    g.course,
    g.grade,
    g.exam,
    g.pass
FROM stg_grades AS g
JOIN dim_semester AS s
ON g.semester = s.semester AND g.year = s.year;
```

2.7 Dodatkowe tabele faktów

Tabela Teacher Workload

Utworzono dodatkowo nową tabelę `teacher_semester_workload` agregującą metryki obciążenia nauczycieli w poszczególnych semestrach, obejmujące liczbę wystawionych ocen, prowadzonych kursów, ocenianych studentów, liczbę egzaminów i testów oraz współczynnik zdawalności:

```
DROP TABLE IF EXISTS teacher_semester_workload;

CREATE TABLE [teacher_semester_workload] (
    teacher_id BIGINT NOT NULL,
    semester_id BIGINT NOT NULL,

    workload_tier VARCHAR(20),

    student_count int,
    exam_count int,
    test_count int,
    exam_test_ratio decimal(5,2),

    CONSTRAINT PK_student_workload_2 PRIMARY KEY (teacher_id, semester_id)
);
```

Załadowano dane do tabeli wymiarowej na podstawie agregacji danych z tabeli faktów `fact_grades`:

```
INSERT INTO [teacher_semester_workload] (teacher_id, semester_id,
    workload_tier, student_count, exam_count, test_count, exam_test_ratio)
SELECT
    teacher_id,
    semester_id,
    CASE
        WHEN COUNT(DISTINCT student_id) < 50 THEN 'light'
        WHEN COUNT(DISTINCT student_id) < 150 THEN 'normal'
        ELSE 'heavy'
    END,
    COUNT(DISTINCT student_id) AS student_count,
    SUM(CASE WHEN exam = 'E' THEN 1 ELSE 0 END) AS exam_count,
    SUM(CASE WHEN exam = 'T' THEN 1 ELSE 0 END) AS test_count,
    CASE
        WHEN SUM(CASE WHEN exam = 'T' THEN 1 ELSE 0 END) = 0 THEN NULL
        ELSE CAST(SUM(CASE WHEN exam = 'E' THEN 1 ELSE 0 END) AS DECIMAL(5,2)) /
            CAST(SUM(CASE WHEN exam = 'T' THEN 1 ELSE 0 END) AS DECIMAL(5,2))
    END AS exam_test_ratio
FROM [fact_grades]
GROUP BY teacher_id, semester_id;
```

Tabela Student Semester Workload

Podobnie została utworzona tabela `Student Semester Workload` agregująca metryki obciążenia studentów w poszczególnych semestrach.

```
DROP TABLE IF EXISTS [student_semester_workload];

CREATE TABLE [student_semester_workload] (
    [student_id] bigint,
    [semester_id] bigint,
    [grade_count] int,
    [course_count] int,
    [teacher_count] int,
    [exam_count] int,
    [test_count] int,
    [pass_rate] DECIMAL(5,2)

    CONSTRAINT PK_student_workload PRIMARY KEY (student_id, semester_id)
);
```

Załadowano dane do tabeli wymiarowej na podstawie agregacji danych z tabeli faktów fact_grades:

```
INSERT INTO student_semester_workload
SELECT
    student_id,
    semester_id,
    COUNT(*) as grade_count,
    COUNT(DISTINCT course) as course_count,
    COUNT(DISTINCT teacher_id) as teacher_count,
    SUM(CASE WHEN exam = 'E' THEN 1 ELSE 0 END) as exam_count,
    SUM(CASE WHEN exam = 'T' THEN 1 ELSE 0 END) as test_count,
    CAST(SUM(CAST(pass as int)) * 100.0 / COUNT(*) AS DECIMAL(5,2)) as pass_rate
FROM fact_grades
GROUP BY student_id, semester_id
ORDER BY student_id, semester_id;
```

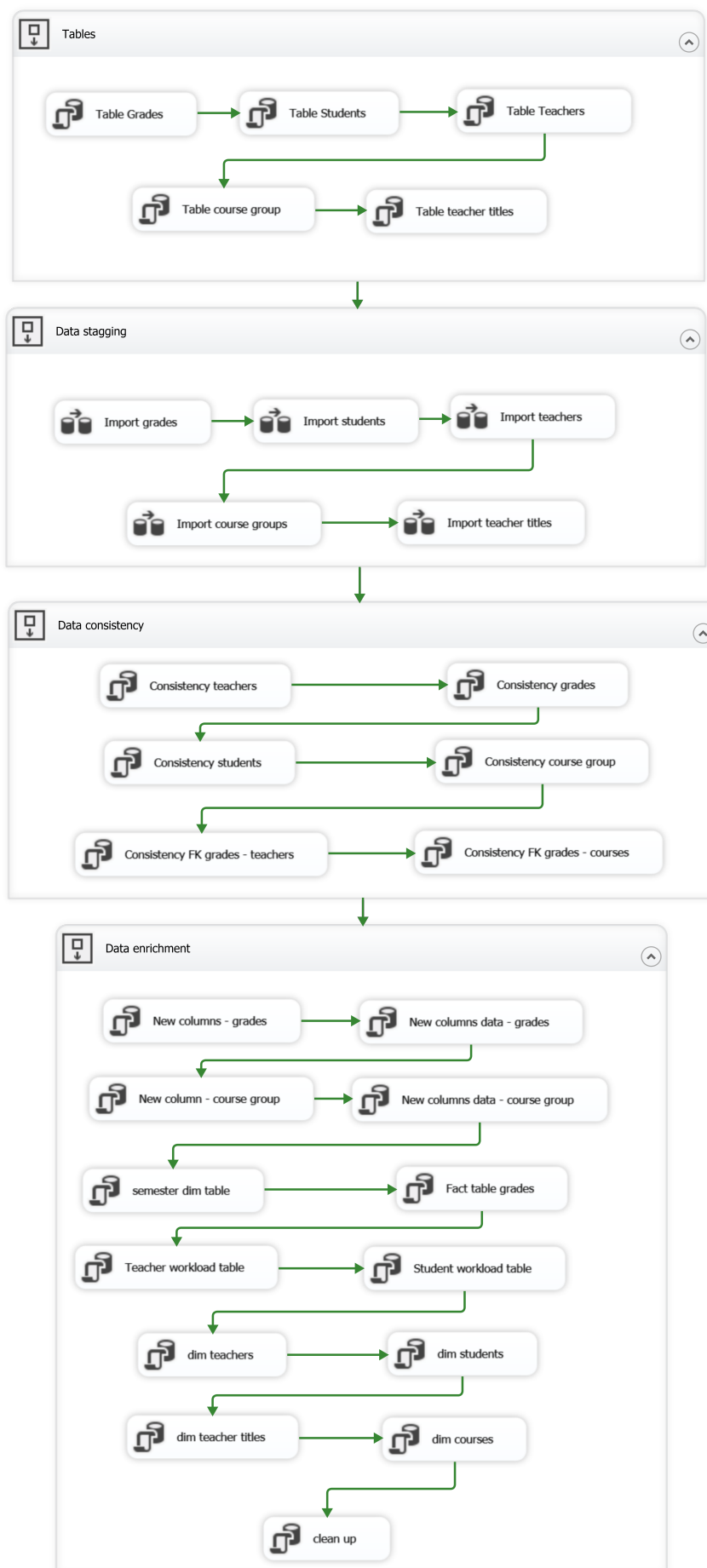
2.8 Usunięcie tabel pośrednich

Na zakończenie procesu ETL w narzędziu SSIS przeprowadzono czyszczenie bazy danych. Tabele pośrednie (staging) zostały usunięte, pozostawiając jedynie tabele wymiarowe oraz tabelę faktów stanowiące finalną strukturę hurtowni danych.

Przykładowe usunięcie tabeli staging:

```
DROP TABLE IF EXISTS stg_students;
```

2.9 Diagram procesu ETL



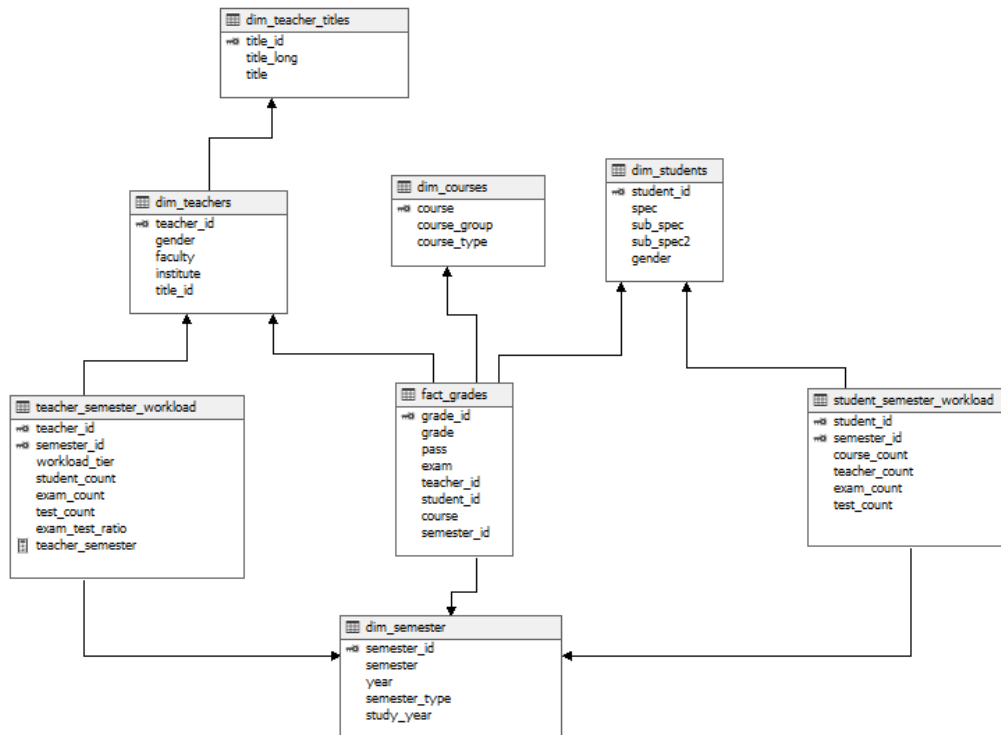
Rysunek 1: Diagram przepływu danych w procesie ETL zrealizowanym w narzędziu SSIS

3 SSAS

Po zakończeniu procesu ETL, dane zostały wykorzystane do budowy kostki OLAP przy pomocy narzędzia SQL Server Analysis Services (SSAS). Celem tego etapu było umożliwienie efektywnej analizy ocen w wielu przekrojach analitycznych, takich jak cechy studentów, prowadzących oraz kursów.

3.1 Źródło danych i widok

Po zdefiniowaniu połączenia został stworzony widok (Data Source View), który zawiera tabele faktów oraz tabele wymiarów. Widok ten stanowi podstawę dla kostki OLAP i definiuje relacje pomiędzy tabelami.



Rysunek 2: Schemat gwiazdy po transformacjach ETL - tabela faktów Fact_grades oraz tabele wymiarów

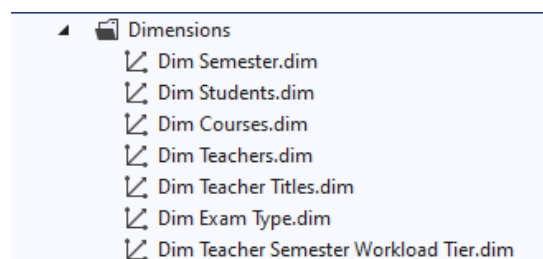
Schemat gwiazdy (rys. 2) składa się z centralnej tabeli faktów **Fact_grades** oraz dodatkowych tabel dotyczących obciążenia: **teacher_semester_workload** i **student_semester_workload**, wraz z otaczającymi je tabelami wymiarów.

3.2 Struktura kostki OLAP

Następnie została stworzona kostka OLAP, a przy pomocy kreatora zostały zdefiniowane wymiary oraz miary analityczne.

3.2.1 Wymiary

Wymiary zostały zdefiniowane poprzez kreator wymiarów **Dimension Wizard**. Każdy wymiar reprezentuje określony aspekt analizy i umożliwia grupowanie oraz filtrowanie danych według różnych kryteriów.



Rysunek 3: Zdefiniowane wymiary w kostce OLAP

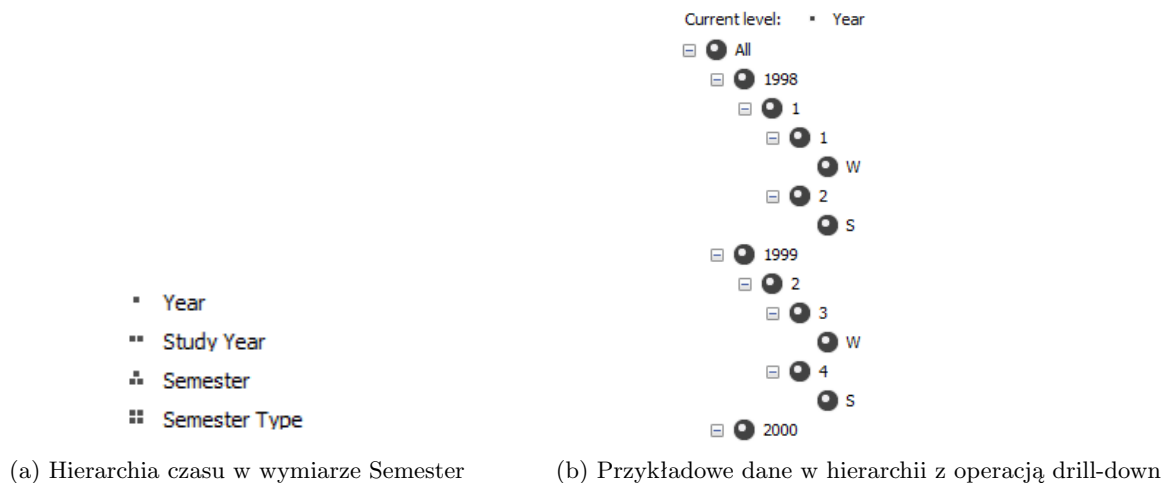
Poniżej znajduje się opis zdefiniowanych wymiarów (rys. 3):

- Wymiar `Dim Semester` - wymiar zbudowany na podstawie tabeli `dim_semester`
- Wymiar `Dim Students` - wymiar zbudowany na podstawie tabeli `dim_students`
- Wymiar `Dim Courses` - wymiar zbudowany na podstawie tabeli `dim_courses`
- Wymiar `Dim Teachers` - wymiar zbudowany na podstawie tabeli `dim_teachers`
- Wymiar `Dim Teacher Titles` - wymiar zbudowany na podstawie tabeli `dim_teacher_titles`
- Wymiar `Dim Exam Type` - wymiar zbudowany na podstawie kolumny `exam` z tabeli `fact_grades`
- Wymiar `Dim Teacher Semester Workload Tier` - wymiar zbudowany na podstawie kolumny `workload_tier` z tabeli `teacher_semester_workload`

3.2.2 Hierarchie w wymiarach

W wybranych wymiarach zostały zdefiniowane hierarchie, które umożliwiają wielopoziomową nawigację po danych. Hierarchie pozwalają na analizę danych na różnych poziomach szczegółowości.

Przykładem takiej hierarchii jest hierarchia zdefiniowana w wymiarze `Dim Semester`. Hierarchia ta umożliwia analizę danych w następujących poziomach przedstawionych na rysunku 4b. ilustruje ona działanie operacji drill-down, gdzie użytkownik może przejść od danych zagregowanych rocznych do bardziej szczegółowych semestralnych danych.



Rysunek 4: Hierarchia czasu w kostce OLAP

3.2.3 Grupy miar

Następnie wymiary zostały dodane do nowo utworzonej kostki OLAP. W zakładce `Cube Structure` zostały zdefiniowane tabele faktów jako nowe grupy miar (`Measure Group`). Każda grupa miar reprezentuje zestaw powiązanych ze sobą metryk analitycznych pochodzących z określonej tabeli faktów.

Measures	
	Student Analysis DB
	Student Semester Workload
	Teacher Semester Workload
	Fact Grades

Rysunek 5: Grupy miar zdefiniowane w kostce OLAP

Na rysunku 5 przedstawiono zdefiniowane grupy miar, które zawierają podstawowe metryki takie jak liczba ocen, suma punktów oraz liczba pozytywnych wyników.

3.2.4 Miary wyliczane

Oprócz podstawowych miar pochodzących bezpośrednio z tabel faktów, za pomocą funkcji `Calculated Member` w zakładce `Calculations` zostały dodane miary wyliczane w języku MDX (Multidimensional Expressions). Miary wyliczane zdefiniowane są na podstawie istniejących miar.

Average Grade

Miara wyliczająca średnią arytmetyczną ocen poprzez podzielenie sumy ocen przez liczbę ocen:

`[Measures].[Grade] / [Measures].[Fact Grades Count]`

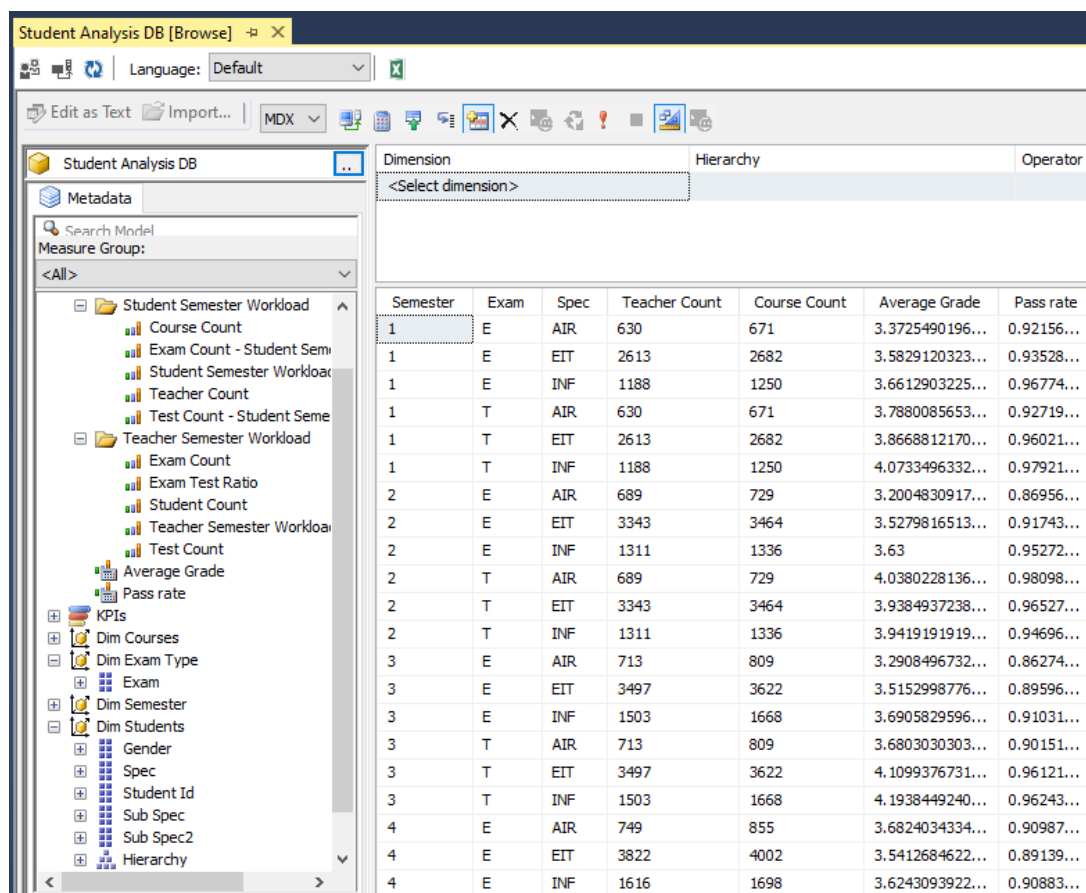
Pass Rate

Miara obliczająca procent pozytywnych ocen w stosunku do wszystkich przyznanych ocen:

`[Measures].[Pass] / [Measures].[Fact Grades Count]`

4 Analiza

Przykładowe analizy kostki OLAP zostały wykonane przy użyciu narzędzia Cube Browser dostępnego w SQL Server Management Studio oraz w tabelach przestawnych w programie Microsoft Excel. Poniżej przedstawiono wyniki analiz odpowiadających na kluczowe pytania biznesowe dotyczące wpływu różnych czynników na wyniki studentów i obciążenia studentów i nauczycieli.



Semester	Exam	Spec	Teacher Count	Course Count	Average Grade	Pass rate
1	E	AIR	630	671	3.3725490196...	0.92156...
1	E	EIT	2613	2682	3.5829120323...	0.93528...
1	E	INF	1188	1250	3.6612903225...	0.96774...
1	T	AIR	630	671	3.7880085653...	0.92719...
1	T	EIT	2613	2682	3.8668812170...	0.96021...
1	T	INF	1188	1250	4.0733496332...	0.97921...
2	E	AIR	689	729	3.2004830917...	0.86956...
2	E	EIT	3343	3464	3.5279816513...	0.91743...
2	E	INF	1311	1336	3.63	0.95272...
2	T	AIR	689	729	4.0380228136...	0.98098...
2	T	EIT	3343	3464	3.9384937238...	0.96527...
2	T	INF	1311	1336	3.9419191919...	0.94696...
3	E	AIR	713	809	3.2908496732...	0.86274...
3	E	EIT	3497	3622	3.5152998776...	0.89596...
3	E	INF	1503	1668	3.6905829596...	0.91031...
3	T	AIR	713	809	3.6803030303...	0.90151...
3	T	EIT	3497	3622	4.1099376731...	0.96121...
3	T	INF	1503	1668	4.1938449240...	0.96243...
4	E	AIR	749	855	3.6824034334...	0.90987...
4	E	EIT	3822	4002	3.5412684622...	0.89139...
4	E	INF	1616	1698	3.6243093922...	0.90883...

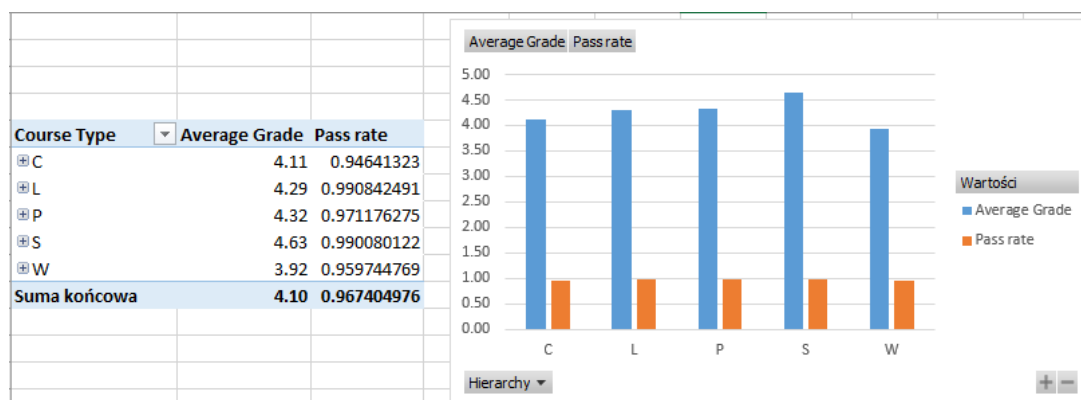
Rysunek 6: Przykładowa analiza w narzędziu Cube Browser

Na rysunku 6 przedstawiono interfejs narzędzia Cube Browser, w którym przeprowadzono przykładową analizę obciążenia studenta, zdawalności oraz średniej ocen. Analiza uwzględnia wymiary: semestr, forma zaliczenia oraz specjalność studenta.

4.1 Pytania analityczne

4.1.1 Wpływ typu kursu na średnią ocen i zdawalność

Pierwsza analiza dotyczy zależności między typem kursu a osiąganymi wynikami studentów. Na rysunku 7 przedstawiono średnią ocen oraz wskaźnik zdawalności (pass rate) w zależności od kategorii kursu.



Rysunek 7: Średnia ocen i zdawalność w zależności od typu kursu

4.1.2 Wpływ tytułu naukowego prowadzącego na wyniki studentów

Druga analiza bada zależność między tytułem naukowym prowadzącego a średnią ocen oraz zdawalnością. Wykorzystano hierarchię drill-down w wymiarze tytułów: od skróconego tytułu (title) do pełnej nazwy (title long). Wyniki przedstawiono na rysunku 8.

Titles	Average Grade	Pass rate	Fact Grades Count
doc.	4.03	0.931623932	117
doc. dr inż.	4.03	0.931623932	117
dr	4.03	0.963543267	29295
dr	3.96	0.957868304	3584
dr inż.	4.04	0.964334332	25711
dr hab.	3.66	0.902277737	2722
mgr	4.18	0.980655859	10856
prof.	4.04	0.960437114	9883
unk	4.35	0.985228472	12321
Suma końcowa	4.10	0.967404976	65194

Rysunek 8: Średnia ocen i zdawalność w zależności od tytułu prowadzącego (hierarchia drill-down)

4.1.3 Wpływ jednostki prowadzącego na wyniki studentów

Kolejna analiza sprawdza, czy jednostka organizacyjna prowadzącego (wydział, instytut) ma wpływ na wyniki studentów. Zastosowano hierarchię drill-down: wydział (faculty) → instytut (institute). Wyniki zaprezentowano na rysunku 9.

Faculty	Average Grade	Pass rate	Fact Grades Count
2	4.00	0.98	300
I-22	4.10	0.981481481	108
I-23	3.96	0.977777778	180
I-24	3.63	1	12
4	4.09	0.967096135	50450
8	4.53	0.991196271	1931
10	4.38	1	8
11	3.59	0.945381372	5218
21	4.04	0.948130277	1658
23	4.83	0.991457286	1990
Suma końcowa	4.08	0.966355292	61555

Rysunek 9: Średnia ocen i zdawalność w zależności od jednostki prowadzącego (hierarchia drill-down)

4.1.4 Zależność wyników od formy zaliczenia i roku studiów

Analiza przedstawiona na rysunku 10 bada zdawalność, średnią ocen oraz liczbę ocen w zależności od typu formy zaliczenia (egzamin lub kolokwium) oraz roku studiów studenta.

Exam/Study Year	Average Grade	Pass rate	Fact Grades Count
E	3.74	0.932613062	14439
1	3.55	0.931778929	3474
2	3.56	0.897216951	3162
3	3.75	0.931386447	3527
4	3.93	0.958999096	3317
5	4.25	0.965589155	959
T	4.21	0.977302729	50755
1	3.93	0.962100865	6702
2	4.05	0.957505632	9766
3	4.15	0.975787118	12514
4	4.30	0.989978098	15067
5	4.59	0.995675514	6706
Suma końcowa	4.10	0.967404976	65194

Rysunek 10: Zdawalność, średnia ocen i liczba ocen w zależności od formy zaliczenia i roku studiów

4.1.5 Wpływ obciążenia nauczyciela na wyniki studentów

Najważniejsza analiza dotyczy wpływu obciążenia dydaktycznego nauczyciela na wyniki studentów. Badano średnią ocen, zdawalność, liczbę studentów oraz liczbę egzaminów w zależności od semestru (zimowy W, letni S), poziomu obciążenia oraz tytułu prowadzącego. Wyniki zaprezentowano na rysunku 11.

Semester Type/Teacher Title	Average Grade	Pass rate	Exam Test Ratio	Student Count	Exam Count	Test Count
S	4.08	0.967268927	1059.72	25129	7310	22356
doc.	3.97	0.787878788	0	33	0	33
dr	4.00	0.959474008	431.59	10770	3382	9622
dr hab.	3.72	0.897941681	38.96	938	389	777
mgr	4.15	0.98046875	333.84	4096	1189	3931
prof.	4.04	0.968937446	161.05	4139	1500	3168
unk	4.27	0.987136564	94.28	5153	850	4825
W	4.12	0.967518577	882.23	29315	7129	28399
doc.	4.05	0.988095238	0	84	0	84
dr	4.06	0.96679148	308.12	12942	3251	13040
dr hab.	3.61	0.905526992	33.56	1296	608	948
mgr	4.20	0.980822873	98.84	4552	699	5037
prof.	4.04	0.95282838	383.96	4591	1946	3269
unk	4.41	0.983599157	57.75	5850	625	6021
Suma końcowa	4.10	0.967404976	1941.95	54444	14439	50755

Rysunek 11: Wpływ obciążenia nauczyciela na wyniki studentów w zależności od semestru i tytułu