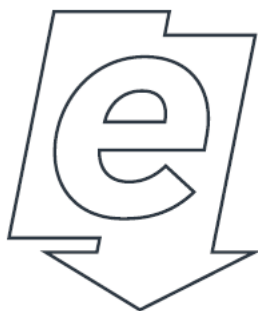


**Sprawozdanie z projektu**  
**Wykorzystanie MS SQL i pakietów MS Business**  
**Intelligence do budowy aplikacji**

Warszawa 11.05.2024

Tomasz Kowalski  
Informatyka Stosowana



**Wydział**  
**Elektryczny**

POLITECHNIKA WARSZAWSKA

## Spis treści

1	Omówienie	2
2	Dane wejściowe	2
3	Baza danych	3
4	Skrypt SSIS	4
5	Przenoszenia do docelowych tabel	4
6	Pobranie ocen	4
7	Walidacja danych wejściowych	6
7.1	Błędy w tmp_uczniowie	6
7.2	Błędy w tmp_oceny	7
8	Wnioski i rozwój	7

## Spis rysunków

1	Diagram bazy danych	3
2	Skrypt SSIS	4
3	Logi błędów dla sprawdzenia tabeli tmp_uczniowie	6
4	Logi błędów dla sprawdzenia tabeli tmp_oceny	7

## Spis tabel

1	Dane wejściowe uczniów	2
2	Dane wejściowe uczniów	2

## Listings

1	Zwracany JSON	4
---	---------------	---

# 1 Omówienie

Zadanie, które wybrałem polega na wgraniu danych z odpowiednich plików do bazy danych i wygenerowanie pliku JSON odpowiadającemu wprowadzonemu danym. Wgrywane dane są sprawdzane pod kątem różnego rodzaju błędów i odpowiednio wstawiane do reszty tabel w bazie danych. W całym projekcie korzystałem tylko z MS SQL oraz pakietu SSIS (SQL Server Integration Service) które umożliwiają szybkie wgrywanie i wykonywanie instrukcji z SQL poprzez uruchomienie pliku skryptowego. Tworzenie skryptu SSIS jest w miarę proste. Jest to język graficzny i dość intuicyjny. Jednak to jest samo wywołanie. Każda funkcja albo procedura trzeba samemu najpierw napisać żeby móc z niej skorzystać.

## 2 Dane wejściowe

Dane wejściowe są zapisane w plikach txt. Jednak mogą być łatwo utworzone po wyeksportowaniu arkusza kalkulacyjnego z Excela do pliku txt. Jeden plik zawiera informacje o uczniach: imię, nazwisko, numer legitymacji, datę urodzenia. Przykładowe poprawne dane przedstawione są w tabeli poniżej.

imię	nazwisko	nr_legitymacji	data_ur
Tomasz	Kowalski	313565	28.01.2009
Jan	Nowak	302648	20-05-2009
Piotr	Lewandowski	310013	20090411

Tabela 1: Dane wejściowe uczniów.

Kolejne wczytywane dane dotyczą ocen końcowych za rok szkolny. Jeden plik może zawierać dane z tylko jednego roku szkolnego. W pliku nie może być dwóch ocen dla tego samego ucznia z tego samego przedmiotu. Jest również informacja o nauczycielu który wystawił konkretną ocenę. Przykładowe poprawne dane przedstawione są w tabeli poniżej.

ocena	przedmiot	rok_sz	nr_legitymacji	imię_n	nazwisko_n
5	Przyroda	22/23	313565	Anna	Graba
4	Przyroda	22/23	302648	Anna	Graba
6	Przyroda	22/23	310013	Aleksandra	Wróblewska
6	WF	22/23	313565	Piotr	Januchta
4	WF	22/23	302648	Piotr	Januchta
4	WF	22/23	310013	Piotr	Januchta
5	Matematyka	22/23	313565	Tadeusz	Mikołajuk
3	Matematyka	22/23	302648	Tadeusz	Mikołajuk
2	Historia	22/23	310013	Piotr	Borowik

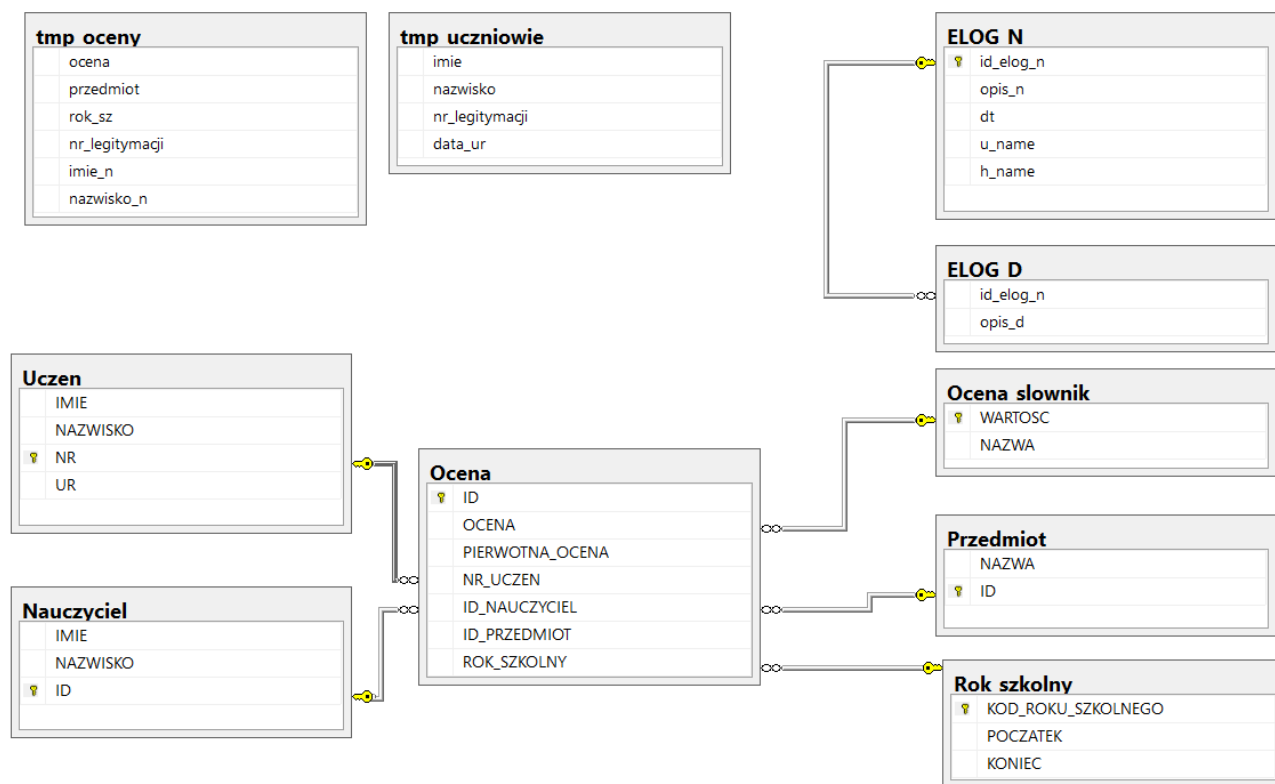
Tabela 2: Dane wejściowe uczniów.

### 3 Baza danych

Baza składa się z 10 tabel. Można łatwo podzielić na 4 grupy:

- Tabele tymczasowe
- Tabele błędów
- Tabele słownikowe
- Tabele docelowe

Do tabeli tymczasowych wgrywane są bezpośrednio dane z plików txt. Dlatego są odwzorowaniem powyższych tabel z danymi. Tabele błędów służą do logowania błędów, które są wykrywane podczas walidacji tabel tymczasowych. Słowniki służą do pomocy i żeby nie powielać danych. Nauczyciel uczy wielu uczniów, dlatego jak raz trafi do bazy, to dalej przekazywana jest tylko relacja z oceną zamiast pełnych danych. Każda ocena ma też odwzorowanie słowne i ten słownik jest wypełniony i do niego dane nie dochodzą. Główna tabela docelowa jest Ocena i Uczniowie, gdzie trafiają dane po obróbce i sprawdzeniu poprawności. Poniżej znajduje się diagram całej bazy danych.

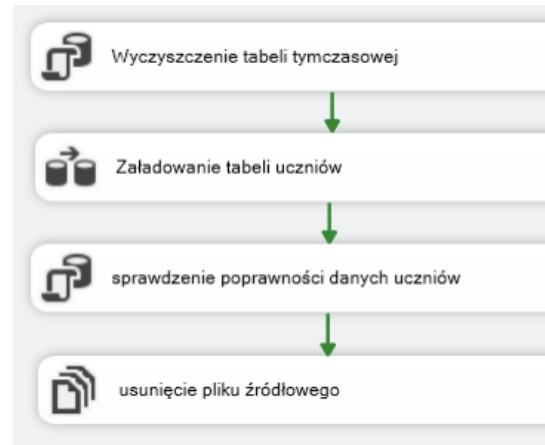


Rysunek 1: Diagram bazy danych

## 4 Skrypt SSIS

Skrypt jest prosty w budowie bo proces jaki wykonuje jest liniowy wiec nie zawiera jakies skomplikowanej logiki. Przechadzac po kolei przez wszystkie kroki skryptu. Na nastepuje usuniecie wszystkich wartosci z tabeli tymczasowej (tmp\_oceny lub tmp\_uczniowie). Kolejnym krokiem jest zaciagniecie danych z pliku i wpisanie ich do tabeli tmp\_\*. Następnie tabela tmp\_\* podlega sprawdzeniu poprawności. Jeśli wystepia jakies błędy to skrypt je zaloguje w odpowiednich tabelach oraz zwróci -1. Na koniec plik źródłowy jest usuwany. Obok widnieje zdjęcie obrazujące opisane kroki. Skrypt do dodawania ocen i uczniów ma taka sama konstrukcje tylko musza być dwa osobne bo każdy z nich odpowiada za odpowiedni plik źródłowy i table tymczasowa. Dlatego powyżej korzystałem z aliasu tmp\_\*.

Skrypt do ocen dodatkowo po sprawdzeniu poprawności, jeśli wszystko jest poprawne, następuje wgranie danych do tabel docelowych oraz zasilenie słowników. Teraz ostatnim etapem jest wyciągnięcie tych danych w kolejna napisana procedura.



Rysunek 2: Skrypt SSIS

## 5 Przenoszenia do docelowych tabel

Przenoszenie do tabel słownikowych nie jest skomplikowane.

Następuje sprawdzenie czy rekord który chcemy wstawić już istnieje w bazie, jeśli tak to go nie dodajemy jeśli nie istnieje to dodajemy nowy. Taka procedura odbywa się dla tabel: Uczeń, Nauczyciel, Przedmiot, Rok\_szkolny.

Dodawanie rekordów do tabeli Ocena jest nieco bardziej skomplikowane. Musimy znaleźć odwołania do wszystkich słowników aby poprawnie wstawić relacje z uczniem, nauczycielem, przedmiotem i rokiem szkolnym. Oraz sprawdzamy czy taki uczeń nie ma już oceny z takiego przedmiotu. Jeśli ma to pierwsza ocena zostaje przepisana na atrybut pierwotna ocena zaś nowa trafia do wartości ocena. W przypadku ponownej poprawki pierwotna ocena się nie zmienia a aktualizuje się tylko pole ocena.

## 6 Pobranie ocen

Napisałem procedurę która tworzy nam JSON dla danego roku szkolnego. Procedura za parametr przyjmuje wartość roku szkolnego np: 22/23 i generuje obiekt typu JSON który zawiera informacje o roku szkolnym, uczniach oraz ich ocenach z tego roku. Taki obiekt może pomóc w automatyzacji robienia świadectw dla uczniów bo wszystko jest uschematyzowane. Poniżej znajduje się wygenerowany json dla poprawnych danych z 2 rozdziału. Oczywiście zwracani są tylko Ci uczniowie którzy mają oceny w danym roku szkolnym.

Listing 1: Zwracany JSON

```
1 {
2   "rok_szkolny": "22/23",
3   "data_roz poczenia": "01-09-2022",
4   "data_zakonczenia": "30-06-2023",
5   "uczniowie": [
6     {
7       "uczen": "Jan Nowak",
8       "data_urodzenia": "20-05-2009",
9       "nr_legitymacji": "302648",
10      "oceny": [
11        {
12          "ocena": "4",
13          "ocena_sloownie": "Dobry",
14          "pierwotna_ocena": "null",
15          "przedmiot": "Przyroda",
```

```

16         "nauczyciel": "Anna Graba"
17     },
18     {
19         "ocena": "4",
20         "ocena_slownie": "Dobry",
21         "pierwotna_ocena": "null",
22         "przedmiot": "WF",
23         "nauczyciel": "Piotr Januchta"
24     },
25     {
26         "ocena": "3",
27         "ocena_slownie": "Dostateczny",
28         "pierwotna_ocena": "null",
29         "przedmiot": "Matematyka",
30         "nauczyciel": "Tadeusz Miko ajuk"
31     }
32 ]
33 },
34 {
35     "uczen": "Piotr Lewandowski",
36     "data_urodzenia": "11-04-2009",
37     "nr_legitymacji": "310013",
38     "oceny": [
39         {
40             "ocena": "6",
41             "ocena_slownie": "Celuj cy",
42             "pierwotna_ocena": "null",
43             "przedmiot": "Przyroda",
44             "nauczyciel": "Aleksandra Wr blewska"
45         },
46         {
47             "ocena": "2",
48             "ocena_slownie": "Dopuszczaj cy",
49             "pierwotna_ocena": "null",
50             "przedmiot": "Historia",
51             "nauczyciel": "Piotr Borowik"
52         },
53         {
54             "ocena": "4",
55             "ocena_slownie": "Dobry",
56             "pierwotna_ocena": "null",
57             "przedmiot": "WF",
58             "nauczyciel": "Piotr Januchta"
59         }
60     ]
61 },
62 {
63     "uczen": "Tomasz Kowalski",
64     "data_urodzenia": "28-01-2009",
65     "nr_legitymacji": "313565",
66     "oceny": [
67         {
68             "ocena": "5",
69             "ocena_slownie": "Bardzo dobry",
70             "pierwotna_ocena": "null",
71             "przedmiot": "Przyroda",

```

```

72         "nauczyciel": "Anna Graba"
73     },
74     {
75         "ocena": "6",
76         "ocena_sloownie": "Celuj cy",
77         "pierwotna_ocena": "null",
78         "przedmiot": "WF",
79         "nauczyciel": "Piotr Januchta"
80     },
81     {
82         "ocena": "5",
83         "ocena_sloownie": "Bardzo dobry",
84         "pierwotna_ocena": "null",
85         "przedmiot": "Matematyka",
86         "nauczyciel": "Tadeusz Miko ajuk"
87     }
88 ]
89 }
90 ]
91 }

```

## 7 Walidacja danych wejściowych

Każdy s plików jest walidowany po innym katem. Dlatego w skrypcie powstało kilka różnych funkcji pomocniczych które ułatwiły dalsza prace. Funkcje pomocnicze: rok\_szkolny\_check, txt2ocena, txt2data

### 7.1 Błędy w tmp\_uczniowie

- Brak danych w tabeli
- Powtarzający się numer legitymacji w tabeli
- Dodanie ucznia z takim numerem legitymacji który już istnieje w docelowej tabeli
- Nie można przekonwertowC daty urodzenia

Na poniższym zdjeciu możemy sprawdzić jakie błędy pojawiły się i w jaki sposób są logowane. Na potrzeby prezentacji wstawiłem dane z błędami żeby wszyscy możliwe błędy się zalogowały. Logi są na tyle czytelne żeby dało się łatwo znaleźć błędy w pliku źródłowym.

	id_elog...	opis_n	dt	u_na...	h_name
1	34	Błąd w procedurze: Uczeń_n_process / Plik z listą uczniów PUSTY!	2024-05-11 20:42:26.090	dbo	LAPTOP-CG651VMM
2	35	Błąd w procedurze: Uczeń_n_process / Numery Legitymacji się powtarzają	2024-05-11 20:42:39.257	dbo	LAPTOP-CG651VMM
3	36	Błąd w procedurze: Uczeń_n_process / Numery Legitymacji się powtarzają w tabeli Uczeń	2024-05-11 20:42:39.260	dbo	LAPTOP-CG651VMM
4	37	Błąd w procedurze: Uczeń_n_process / Błąd w konwersji daty	2024-05-11 20:42:39.273	dbo	LAPTOP-CG651VMM

	id_elog...	opis_d
1	34	0 wierszy w tmp_uczniowie
2	35	Nr legitymacji: 310013 / Ile razy: 2
3	36	Nr legitymacji: 302648 / Ile razy w Uczeń: 1
4	36	Nr legitymacji: 310013 / Ile razy w Uczeń: 2
5	37	Nr legitymacji z błędną konwersją daty: 302648
6	37	Nr legitymacji z błędną konwersją daty: 310013

Rysunek 3: Logi błędów dla sprawdzenia tabeli tmp\_uczniowie

## 7.2 Błędy w tmp\_oceny

- Brak danych w tabeli
- Nie można przekowertować oceny
- Różny rok szkolny w ocenach
- Niepoprawny rok szkolny
- Ocena należy do ucznia którego nie ma w bazie
- Wstawiamy jednemu uczniowi więcej niż jedną ocenę z tego samego przedmiotu

Oceny mogą być wpisywane z przecinkiem jednak do docelowych tabel są spłaszczane. Bardzo często oceny są wpisywane jako 4.0 i na to program jest odporny i poprawnie rzutuje. Poniższe zdjęcie przedstawia logi z błędów podczas sprawdzania pliku tmp\_oceny

	id_elog...	opis_n	dt	u_na...	h_name
1	40	Błąd w procedurze: tmp_ocena_check / Plik z listą ocen PUSTY!	2024-05-11 21:01:53.493	dbo	LAPTOP-CG651VMM
2	41	Błąd w procedurze: tmp_ocena_check / Ocena z poza przedziału	2024-05-11 21:07:50.470	dbo	LAPTOP-CG651VMM
3	42	Błąd w procedurze: tmp_ocena_check / Więcej jak jeden rok szkolny w pliku	2024-05-11 21:07:50.477	dbo	LAPTOP-CG651VMM
4	43	Błąd w procedurze: tmp_ocena_check / Błędny rok szkolny	2024-05-11 21:07:50.477	dbo	LAPTOP-CG651VMM
5	44	Błąd w procedurze: tmp_ocena_check / Nie ma ucznia do którego należy ocena	2024-05-11 21:07:50.480	dbo	LAPTOP-CG651VMM
6	45	Błąd w procedurze: tmp_ocena_check / Uczeń ma więcej niż jedną ocenę z tego samego przedmiotu	2024-05-11 21:07:50.480	dbo	LAPTOP-CG651VMM

	id_elog...	opis_d
1	40	0 wierszy w tmp_oceny
2	41	Nr legitymacji z oceną z poza przedziału: 313565, przedmiot: WF, ocena: 7
3	42	W pliku występuje rok szkolny: 22/23
4	42	W pliku występuje rok szkolny: 22/24
5	43	Niepoprawny rok szkolny: 22/24
6	44	Nr legitymacji który nie należy do żadnego ucznia: 310000
7	45	Nr legitymacji Ucznia: 313565, powtarzający się przedmiot: WF

Rysunek 4: Logi błędów dla sprawdzenia tabeli tmp\_oceny

## 8 Wnioski i rozwój

Aplikacja spełnia wszystkie pierwotne założenia które miały być spełnione. Dane są wczytywane, walidowane, przesyłane do odpowiednich tabel i można je pobrać w formacie JSON. Takie narzędzie może bardzo ułatwić pracę nad aplikacją do automatycznego generowania świadectw albo czymś podobnym. Praca i język SQL nie jest czasami tak intuicyjny jak inne języki programowania Java/Python i tworzenie JSON przy pomocy tylko i wyłącznie SQL nie jest proste. Ale raczej jest wydajniejsze niż w języku programowania bo te JSON są specyficzne i działają z języka programowania poprzez ORM (Object-Relational Mapping) może dawać JSON ale bez relacji a tu jest dość customowy i przygotowany pod świadectwa które łatwo byłoby je generować.

Walidacja danych mogła być być nieco łatwiejsza w języku programowania ale logowanie w bazie danych znowu byłoby utrudnione. Ciężko stwierdzić w jaki sposób jest lepiej łatwiej bo zależy to od wiedzy i umiejętności programisty. Gdybym dostał takie zadanie bez narzuconej technologii Na pewno w większym stopniu opierałbym się na Python, jednak po wykonaniu tego projektu widzę że dużo rzeczy można zrobić łatwiej i szybciej w SQL i zamierzam wykorzystać zdobyte umiejętności w przyszłości.

Skrypt SQL dostępny jest w repozytorium na [GitHub](#).