

# Wyższa Szkoła Informatyki i Zarządzania w Rzeszowie

Temat: Finalna dokumentacja projektu systemu bazodanowego  
do zarządzania szkołą

Jakub Gut w65539  
Grupa SPL02  
Kolegium Informatyki Stosowanej - Informatyka



# 1 Dostęp do bazy danych

- Adres IP serwera: 10.200.2.28
- Autentykacja: Windows
- Nazwa Bazy: w65539Project

## 2 Założenia projektu

Tematem projektu jest system bazodanowy do obsługi uczelni. Baza danych zawierać będzie dane osobowe studentów oraz wykładowców. Umożliwi sprawdzenie aktywności studentów w tym oceny, frekwencje oraz ostatni czas logowania do strony uczelnianej. Pozwoli na sprawdzenie opłat oraz podań złożonych przez studentów, studenci będą mieli dostęp do tych informacji za pośrednictwem ich szkolnego konta. W systemie tym zawarta będzie całkowita frekwencja jak i oceny studenta na poszczególnych zajęciach. Umożliwi dostęp do danych odnośnie, zajęć prowadzonych przez wykładowców takich jak numer pokoju oraz nazwa przedmiotu. Pozwala na wgląd do poszczególnych wydziałów i sprawdzenie przynależności do wydziału każdego studenta jak i pracownika. Umożliwia sprawdzenie, do którego kierunku należy student oraz jego oceny w aktualnym semestrze, a także oceny końcowe z poprzednich semestrów.

## 3 Użytkownicy systemu bazodanowego

Na podstawie założeń tego systemu 3 grupy uzyskają dostęp do bazy danych w określonym zakresie.

- Zarząd uczelni otrzyma wgląd do danych osobowych uczęszczających studentów oraz pracowników. Posiadać będzie możliwość dodawania nowych studentów, usuwania ich z listy uczęszczających, oraz sprawdzania danych odnośnie ich aktywności i kierunku, na który uczęszczają. Będzie miał wgląd do kont uczelnianych studentów co pozwala na sprawdzenie ich podań, opłat oraz czas ostatniego logowania.
- Wykładowcy z danego przedmiotu będą mogli dodawać oraz modyfikować oceny studentów, zarządzać aktywnością oraz obecnością studenta. Zarówno wykładowcy jak i studenci otrzymają wgląd do przedmiotów, przypisanych im sal oraz numer budynku, na którym znajduje się ich wydział. Uzyskają również możliwość sprawdzania przedmiotów i do jakiego typu należą spośród trzech: wykład, laboratorium, projekt.
- Studenci dostaną możliwość wglądu i tworzenia części danych znajdujących się w bazie poprzez konto uczelniane. Po zalogowaniu się na stronę otrzymają wgląd do swoich ocen końcowych, do tworzenia skarg oraz podań, a także sprawdzania informacji dotyczących opłat na wydatki uczelniane typu czesne lub opłata za legitymację. Otrzymają dostęp do informacji odnośnie ich aktualnego kierunku, przedmiotów, specjalizacji oraz przypisane im zajęcia.

## 4 Dane

Dane znajdujące się w bazie:

- Informacje o danych osobowych, ocenach, aktywności, obecności, przedmiotach, kierunkach, semestrze na którym obecnie się znajdują, a także zgłoszeniach i opłatach studentów
- Informacje o pracownikach uczelni, informacja o posadzie, tytuł oraz przypisane im przedmioty do nauczania
- Informacje o kierunkach, specjalnościach, przedmiotach znajdujących się na kierunku, pokoje w których są prowadzone oraz typ zajęć
- Informacje o wydziałach i katedrach w tym: budynki które przypisane są danemu wydziałowi, studenci uczęszczający na dany wydział, kierunki należące do danych wydziałów, pracownicy należący do danej katedry

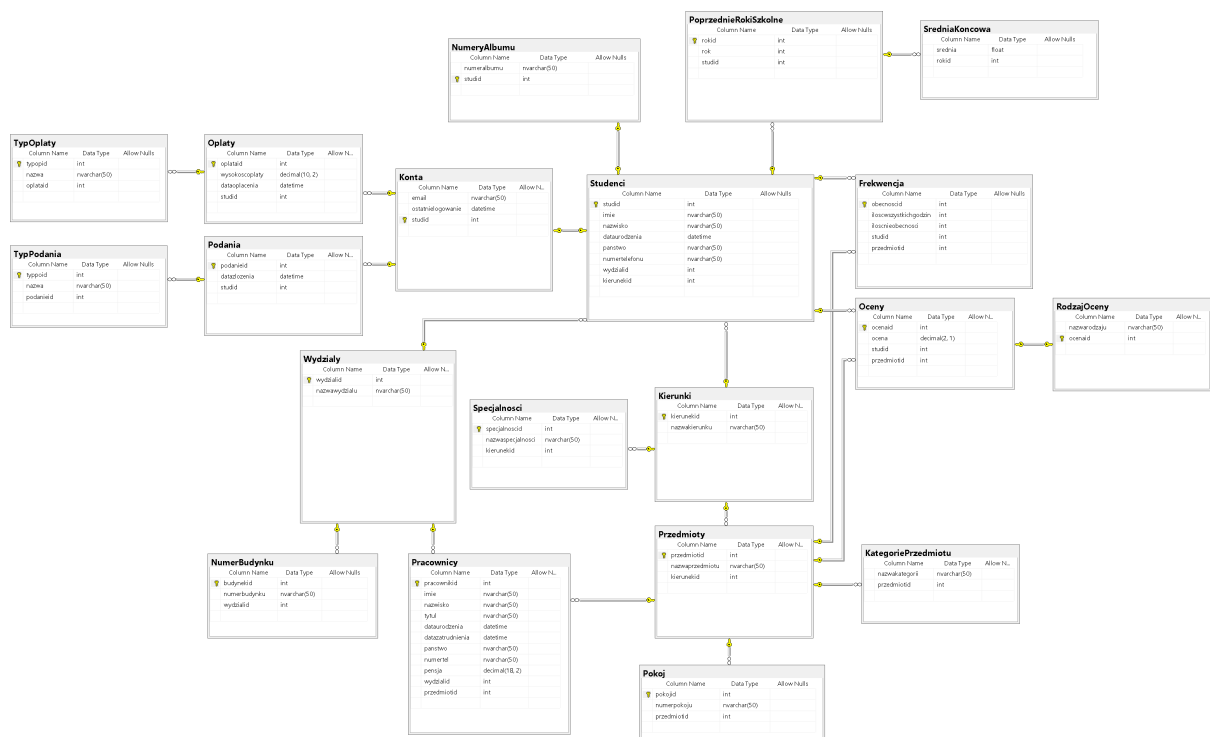
## 5 Raporty i zestawienia

Raporty i zestawienia sporządzone na podstawie utworzonej bazy:

- Lista pracowników i ich pensji
- Lista studentów z największą ilością ocen dostatecznych w obecnym roku szkolnym
- Wyświetlanie najwyższych nie powtarzających się średnich końcowych ze wszystkich lat
- Wyświetlanie numerów albumu z konkretnego przedziału
- Tworzenie ciągu reprezentującego nazwę przedmiotu i przypisanego pokoju w którym jest wykładany
- Lista studentów z nieuregulowanymi opłatami
- Wyświetlanie procentowej frekwencji każdego studenta na wybranym przedmiocie
- Wyświetlanie wszystkich ocen wszystkich uczniów oraz ich średnią końcową
- Lista statusu zaliczeń studentów z każdego przedmiotu na ich kierunku
- Lista wszystkich kierunków i specjalności jakie oferują
- Wyświetlanie wszystkich studentów i wytypowanie tych znajdujących się na wybranym kierunku
- Lista uczniów którzy ostatni raz zalogowali się do swojego konta przed wybraną godziną
- Wyświetlanie listy studentów i przypisanie każdemu ich numeru albumu

- Wyświetlanie ilości podań złożonych przez każdego ucznia
- Wskazanie ucznia z największą ilością oraz sumą nieuregulowanych opłat
- Lista studentów których suma ocen jest większa od wskazanej liczby
- Lista studentów których suma opłat przekracza wybraną liczbę
- Wyświetlanie średnich ze wszystkich średnich uczniów w wskazanych latach
- Lista państw, z których pochodzą zarówno studenci jak i pracownicy
- Wyświetlanie średniej z wszystkich średnich uzyskanych przez uczniów w wybranym roku
- Lista wydziałów, w których ilość uczęszczających studentów jest większa od wskazanej liczby

## 6 Diagram



## 7 Tabele, kolumny oraz liczba rekordów

W bazie znajduje się 20 połączonych relacyjnie tabel. Poniżej zaprezentowana jest lista wszystkich tabel wraz z kolumnami oraz ilością rekordów:

- dbo.Studenci [studid, imie, nazwisko, dataurodzenia, panstwo, numertelefonu, wydzialid, kierunekid] - 500 rekordów
- dbo.Pracownicy [pracownikid, imie, nazwisko, tytul, dataurodzenia, datazatrudnienia, panstwo, numertel, pensja, wydzialid, przedmiotid] - 150 rekordów
- dbo.Wydzialy [wydzialid, nazwawydzialu] - 6 rekordów
- dbo.NumerBudynku [budynekid, numerbudynku, wydzialid] - 6 rekordów
- dbo.Kierunki [kierunekid, nazwakierunku] - 7 rekordów
- dbo.Specjalnosci [specjalnoscid, nazwaspecjalnosci, kierunekid] - 10 rekordów
- dbo.Przedmioty [przedmiotid, nazwaprzedmiotu, kierunekid] - 28 rekordów
- dbo.KategoriePrzedmiotu [nazwakategorii, przedmiotid] - 112 rekordów
- dbo.Pokoj [pokojid, numerpokoju, przedmiotid] - 28 rekordów
- dbo.Oceny [ocenaid, ocena, studid, przedmiotid] - 2000 rekordów
- dbo.RodzajOceny [nazwarodzaju, ocenaid] - 2000 rekordów
- dbo.SredniaKoncowa [srednia, rokid] - 1500 rekordów
- dbo.PoprzednieRokiSzkolne [rokid, rok, studid] - 1500 rekordów
- dbo.NumeryAlbumu [numeralbumu, studid] - 500 rekordów
- dbo.Frekwencja [obecnoscid, iloscwszystkichgodzin, iloscnieobecnosci, studid, przedmiotid] - 2000 rekordów
- dbo.Konta [email, ostatnieologowanie, studid] - 500 rekordów
- dbo.Oplaty [oplataid, wysokoscoplaty, dataopłacenia, studid] - 700 rekordów
- dbo.Podania [podanieid, datazlozenia, studid] - 300 rekordów
- dbo.TypOplaty [typopid, nazwa, oplata] - 700 rekordów
- dbo.TypPodania [typoid, nazwa, podanieid] - 300 rekordów

## 8 Generowanie danych

Dane wykorzystane w bazie zostały wygenerowane przy wykorzystaniu:

- strony internetowej [www.mockaroo.com](http://www.mockaroo.com) - do generowania danych losowych
- języka programowania python - do generowania danych spójnych między poszczególnymi tabelami

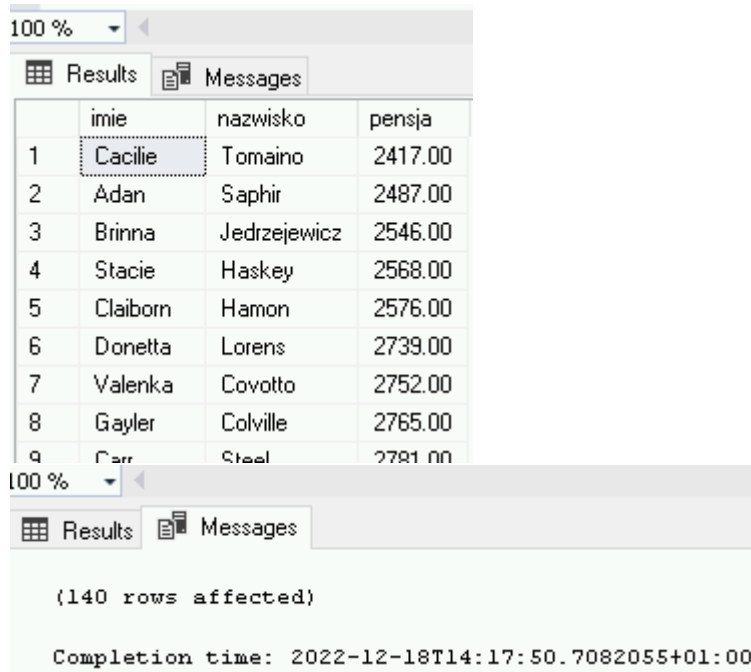
Podczas generowania danych do systemu zarządzania uczelnią kierowałem się podanymi założeniami:

- Aktualny rok szkolny rozpoczął się 1 września 2018 roku.
- Wszyscy studenci zaczęli studiować od tego samego roku, a czas ukończenia studiów trwa 4 lata, obecnie znajdują się na ostatnim roku studiów.
- Dane z poprzednich років szkolnych w połączeniu z danymi średnich końcowych umożliwiają jedynie na wgląd do ocen końcowych ucznia z trzech poprzednich lat.
- Każdy uczeń ma przypisane jedno konto uczelniane, jeden numer albumu, jeden kierunek oraz 4 przedmioty.
- W bazie znajduje się 10 przedmiotów.
- Kierunek może nie mieć żadnej specjalności oraz mieć więcej niż jedną.
- Każdy pokój ma przypisany 1 przedmiot, który jest w nim prowadzony.
- Każdy przedmiot posiada po cztery kategorie: wykład, ćwiczenia, laboratorium, projekt.
- Każdy student jak i pracownik przydzielony jest do jednego wydziału.

## 9 Proste zapytania do jednej tablicy

- Utwórz listę pracowników i ich pensji z pominięciem 10 najniższych

```
SELECT imie, nazwisko, pensja
FROM Pracownicy
ORDER BY pensja ASC
OFFSET 10 ROWS
```



The screenshot shows a SQL Server Enterprise Manager window with a query results grid. The grid has four columns: 'imie', 'nazwisko', and 'pensja'. The first row is highlighted. Below the grid, the message '(140 rows affected)' is displayed, followed by the completion time: '2022-12-18T14:17:50.7082055+01:00'.

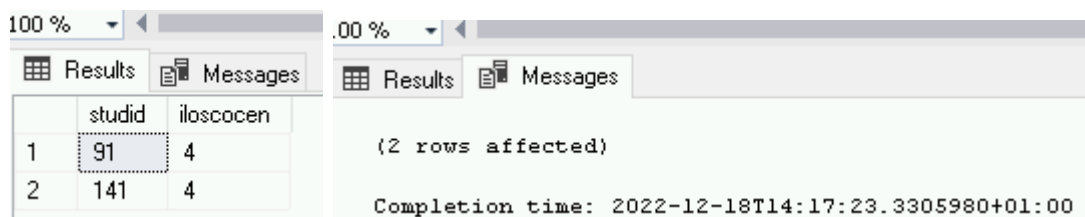
	imie	nazwisko	pensja
1	Cacilie	Tomaino	2417.00
2	Adan	Saphir	2487.00
3	Brinna	Jedrzejewicz	2546.00
4	Stacie	Haskey	2568.00
5	Claiborn	Hamon	2576.00
6	Donetta	Lorens	2739.00
7	Valenka	Covotto	2752.00
8	Gayler	Colville	2765.00
9	Carr	Steel	2781.00

(140 rows affected)

Completion time: 2022-12-18T14:17:50.7082055+01:00

- Wyświetl id studentów z największą ilością ocen dostatecznych w obecnym roku

```
SELECT DISTINCT TOP 1 WITH TIES studid, COUNT(ocena) AS iloscocen
FROM Oceny
WHERE ocena = 3
GROUP BY studid
ORDER BY iloscocen DESC
```



The screenshot shows a SQL Server Enterprise Manager window with a query results grid. The grid has two columns: 'studid' and 'iloscocen'. The first two rows are visible. Below the grid, the message '(2 rows affected)' is displayed, followed by the completion time: '2022-12-18T14:17:23.3305980+01:00'.

	studid	iloscocen
1	91	4
2	141	4

(2 rows affected)

Completion time: 2022-12-18T14:17:23.3305980+01:00

- Wybierz 3 najwyższe, nie powtarzające się średnie końcowe ze wszystkich lat

```
SELECT DISTINCT TOP 3 srednia
FROM SredniaKoncowa
ORDER BY srednia DESC
```

100 %	Results	Messages
1	srednia	
2	5	
3	4,99	
4	4,98	

100 %	Results	Messages
{3 rows affected}		
Completion time: 2022-12-18T14:16:57.5615269+01:00		

- Wyświetl numery albumu od 6500 do 6675 i dodaj każdemu literkę 'w' na początek

```
SELECT CONCAT('w', numeralbumu) AS NumeryAlbumu
FROM NumeryAlbumu
WHERE numeralbumu BETWEEN 6500 AND 6675
```

100 %	Results	Messages
1	NumeryAlbumu	
2	w6500	
3	w6501	
4	w6502	
5	w6503	
6	w6504	
7	w6505	

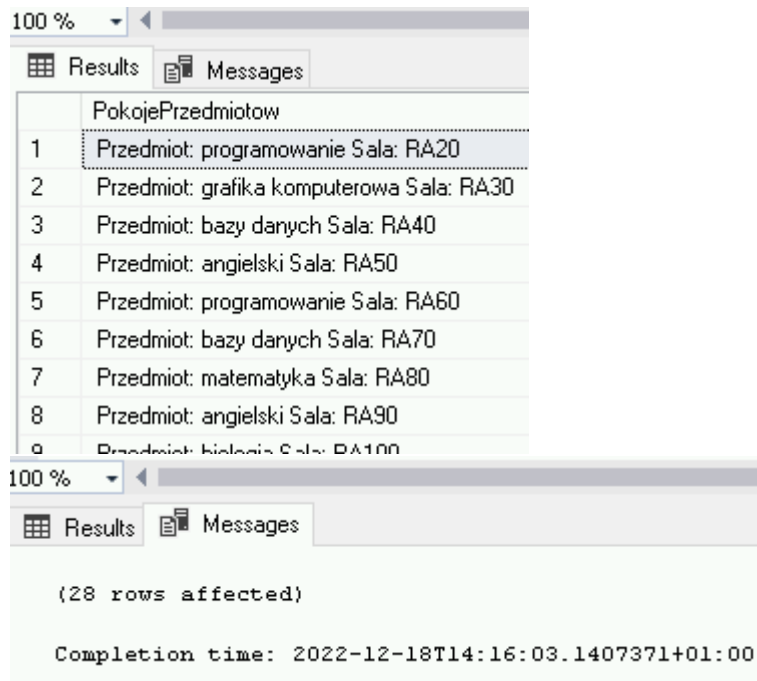
100 %	Results	Messages
{176 rows affected}		
Completion time: 2022-12-18T14:16:23.6173309+01:00		



## 10 Zapytania do kilku tablic z użyciem JOIN

- Utwórz ciąg w którym dodasz nazwę przedmiotu do każdej nazwy pokoju

```
SELECT  
CONCAT('Przedmiot: ', nazwaprzedmiotu, ' Sala: ', numerpokoju)  
AS PokojePrzedmiotow  
FROM Pokoj AS PO INNER JOIN Przedmioty AS PR  
ON PO.przedmiotid = PR.przedmiotid
```



100 %

Results Messages

	PokojePrzedmiotow
1	Przedmiot: programowanie Sala: RA20
2	Przedmiot: grafika komputerowa Sala: RA30
3	Przedmiot: bazy danych Sala: RA40
4	Przedmiot: angielski Sala: RA50
5	Przedmiot: programowanie Sala: RA60
6	Przedmiot: bazy danych Sala: RA70
7	Przedmiot: matematyka Sala: RA80
8	Przedmiot: angielski Sala: RA90
9	Przedmiot: biologia Sala: RA100

(28 rows affected)

Completion time: 2022-12-18T14:16:03.1407371+01:00

- Imiona i nazwiska studentów, którzy posiadają nieuregulowane opłaty

```
SELECT DISTINCT imie, nazwisko  
FROM Studenci AS S  
INNER JOIN Oplaty AS O ON S.studid = O.studid  
WHERE dataoplacenia IS NULL
```



100 %

Results Messages

	imie	nazwisko
1	Abdel	Featherstonhaugh
2	Abelard	Longea
3	Abram	Leale
4	Addie	Gritskov
5	Adler	O'Kielt
6	Adria	Thinn
7	Albertina	Tithacott
8	Aldis	Studd
9	Aleina	Germeon

100 %

Results Messages

(236 rows affected)

Completion time: 2022-12-18T14:15:40.7338350+01:00

- Wypisz procentową frekwencję każdego studenta na angielskim

```
SELECT imie, nazwisko, iloscnieobecnosci,
       iloscwszystkichgodzin, nazwaprzedmiotu,
       CONCAT(IIF(iloscnieobecnosci IS NULL, 100,
ROUND(((CAST(iloscwszystkichgodzin AS FLOAT)-CAST(iloscnieobecnosci
AS FLOAT))/iloscwszystkichgodzin*100),0)), '%')
       AS ProcentFrekwencji
FROM Studenci AS S
INNER JOIN Frekwencja AS F ON S.studid = F.studid
INNER JOIN Przedmioty AS P ON F.przedmiotid = P.przedmiotid
WHERE nazwaprzedmiotu = 'angielski'
```

100 %

Results Messages

	imie	nazwisko	iloscnieobecnosci	iloscwszystkichgodzin	nazwaprzedmiotu	ProcentFrekwencji
1	Ali	Reina	11	160	angielski	93%
2	Neal	Moss	2	210	angielski	99%
3	Stanton	Matfin	2	160	angielski	99%
4	Jone	Klesel	NULL	210	angielski	100%
5	Laurens	Cruess	NULL	200	angielski	100%
6	Any	Tallyn	21	200	angielski	90%
7	Coletta	Jamnes	12	220	angielski	95%

00 %

Results Messages

(500 rows affected)

Completion time: 2022-12-18T14:15:13.8018259+01:00

- Wyświetl wszystkie oceny uzyskane przez każdego ucznia ze wszystkich przedmiotów oraz wyświetl osobno dla każdego ucznia średnią z tych ocen. NULL w kolumnie 'ocena', oznacza że student nie uzyskał jeszcze oceny za dany przedmiot

```
SELECT S.studid, S.imie, S.nazwisko, O.ocena,
       ROUND(CAST(AVG(O.ocena)
OVER(PARTITION BY S.studid ORDER BY S.studid) AS FLOAT), 2) AS Srednia
FROM Oceny AS O
INNER JOIN Studenci AS S
ON O.studid = S.studid
```

100 %

Results Messages					
	studid	imie	nazwisko	ocena	Srednia
1	1	Ali	Reina	5.0	3,67
2	1	Ali	Reina	3.0	3,67
3	1	Ali	Reina	NULL	3,67
4	1	Ali	Reina	3.0	3,67
5	2	Neal	Moss	5.0	3,67
6	2	Neal	Moss	3.0	3,67
7	2	Neal	Moss	3.0	3,67
8	2	Neal	Moss	NULL	3,67
9	3	Stephen	Maffin	2.0	2

100 %

Results Messages

Warning: Null value is eliminated by an aggregate or other SET operation.

(2000 rows affected)

Completion time: 2022-12-18T14:14:53.1836103+01:00

## 11 Zapytania do kilku tablic z użyciem (LEFT JOIN, ...)

- Wypisz na podstawie oceny jaką uzyskał lub nie uzyskał uczeń jaki jest status zaliczenia przedmiotu. Jeżeli student uzyskał ocenę równą lub powyżej 3 to przedmiot jest zaliczony, jeżeli uzyskał niższą to jest niezaliczony, natomiast jeżeli nie uzyskał oceny wcale to przedmiot jest do zaliczenia

```

SELECT S.imie, S.nazwisko,
CASE WHEN O.ocena >= 3 THEN 'Zaliczone'
WHEN O.ocena < 3 THEN 'Niezaliczone'
ELSE 'Do zaliczenia' END AS StatusPrzedmiotu
FROM Oceny AS O
LEFT JOIN Studenci AS S
ON O.studid = S.studid

```

100 %

	imie	nazwisko	StatusPrzedmiotu
1	Ali	Reina	Zaliczone
2	Ali	Reina	Zaliczone
3	Ali	Reina	Do zaliczenia
4	Ali	Reina	Zaliczone
5	Neal	Moss	Zaliczone
6	Neal	Moss	Zaliczone
7	Neal	Moss	Zaliczone
8	Neal	Moss	Do zaliczenia
9	Stanton	Matfin	Zaliczone
10	Stanton	Matfin	Do zaliczenia
11	Stanton	Matfin	Zaliczone

100 %

(2000 rows affected)

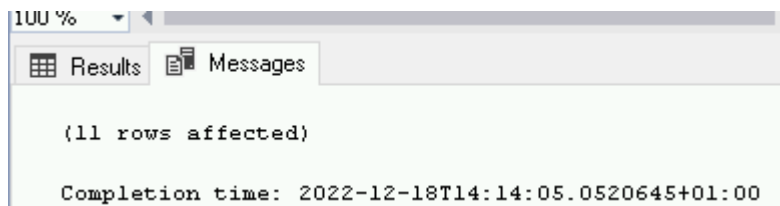
Completion time: 2022-12-18T14:14:27.6002277+01:00

- Wyświetl wszystkie kierunki i ich specjalności. Jeżeli dany kierunek nie posiada specjalności wyświetl w miejscu nazwy tekst: 'nie posiada specjalności'

```
SELECT K.kierunekid, K.nazwakierunku,
IIF(S.nazwaspecjalnosci IS NULL,
'nie posiada specjalnosci', S.nazwaspecjalnosci) AS NazwaSpecjalnosci
FROM Kierunki AS K
LEFT JOIN Specjalnosci AS S
ON K.kierunekid = S.kierunekid
```

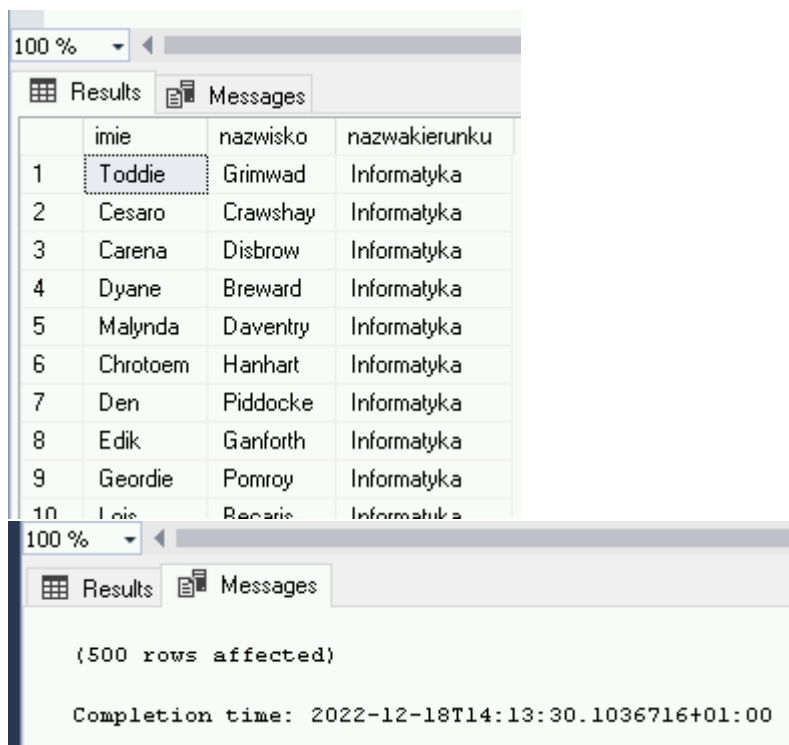
100 %

	kierunekid	nazwakierunku	NazwaSpecjalnosci
1	1	Informatyka	programowanie
2	1	Informatyka	technologie internetowe
3	2	Informatyka stosowana	inżynieria danych
4	3	Kosmetologia	kosmetologia dermatologiczna
5	4	Pielęgniarstwo	pielęgniarstwo chirurgiczne
6	5	Finanse i rachunkowość	nie posiada specjalnosci
7	6	Zarządzanie	zarządzanie w biznesie
8	6	Zarządzanie	marketing
9	6	Zarządzanie	bankowość
10	7	Grafika komputerowa	grafik 2D
11	7	Grafika komputerowa	grafik 3D



- Wyświetl wszystkich studentów, ci którzy są z kierunku Informatyka powinni mieć wypisaną nazwę ich kierunku. Studenci którzy są na kierunku Informatyka powinni zostać wyświetleni w pierwszej kolejności

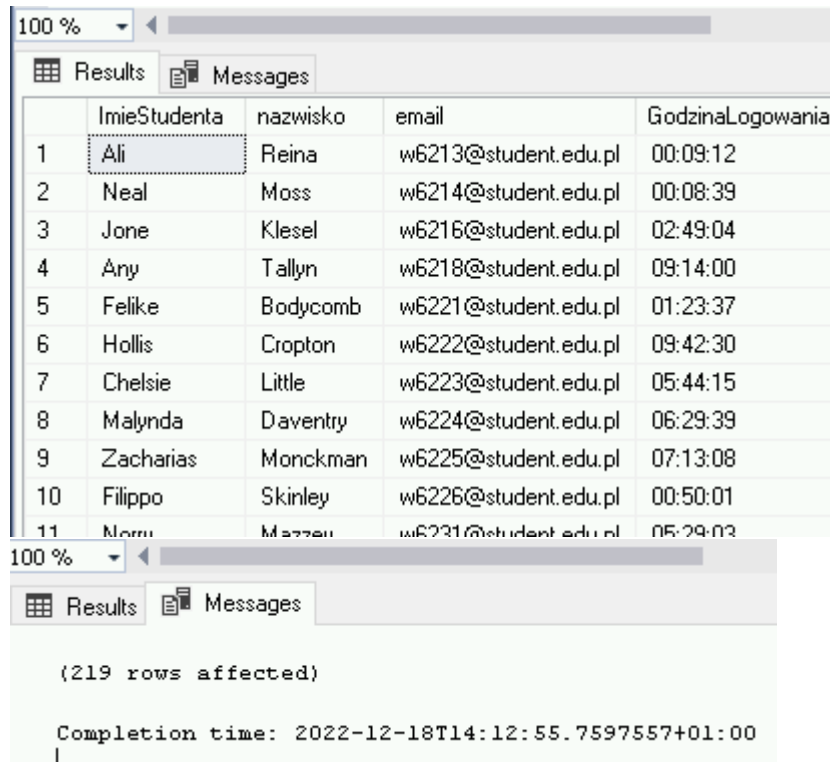
```
SELECT DISTINCT S.imie, S.nazwisko, INF.nazwakierunku
FROM Studenci AS S
LEFT JOIN
(
SELECT S.studid, S.imie, S.nazwisko, K.kierunekid, K.nazwakierunku
FROM Studenci AS S
INNER JOIN Kierunki AS K
ON S.kierunekid = K.kierunekid
WHERE nazwakierunku = 'Informatyka'
) AS INF
ON S.kierunekid = INF.kierunekid
ORDER BY INF.nazwakierunku DESC
```



## 12 Zapytania z podzapytaniem w części SELECT

- Wyświetl listę uczniów (imiona, nazwiska) wraz z mailem, którzy ostatni raz zalogowali się do swojego konta przed godziną 10 i wyświetl godzinę o jakiej się zalogowali

```
SELECT
(
SELECT imie
FROM Studenci AS S
WHERE S.studid = K.studid) AS ImieStudenta,
nazwisko, email,
CONVERT(VARCHAR, ostatnielogowanie, 108) AS GodzinaLogowania
FROM Konta AS K
INNER JOIN Studenci AS S
ON K.studid = S.studid
WHERE DATEPART(HOUR, ostatnielogowanie) < 10
```



The screenshot shows a SQL Server Enterprise Manager window with a query results grid. The grid has 5 columns: 'ImieStudenta', 'nazwisko', 'email', and 'GodzinaLogowania'. It displays 11 rows of student data. Below the grid, the 'Results' tab is active, showing '(219 rows affected)' and the 'Completion time: 2022-12-18T14:12:55.7597557+01:00'.

	ImieStudenta	nazwisko	email	GodzinaLogowania
1	Ali	Reina	w6213@student.edu.pl	00:09:12
2	Neal	Moss	w6214@student.edu.pl	00:08:39
3	Jone	Klesel	w6216@student.edu.pl	02:49:04
4	Any	Tallyn	w6218@student.edu.pl	09:14:00
5	Felike	Bodycomb	w6221@student.edu.pl	01:23:37
6	Hollis	Cropton	w6222@student.edu.pl	09:42:30
7	Chelsie	Little	w6223@student.edu.pl	05:44:15
8	Malynda	Daventry	w6224@student.edu.pl	06:29:39
9	Zacharias	Monckman	w6225@student.edu.pl	07:13:08
10	Filippo	Skinley	w6226@student.edu.pl	00:50:01
11	Morru	Mazzari	w6231@student.edu.pl	05:29:03

(219 rows affected)

Completion time: 2022-12-18T14:12:55.7597557+01:00

- Wyświetl id studenta, imie oraz nazwisko. Następnie przypisz każdemu studentowi jego numer albumu z wykorzystaniem podzapytania w części SELECT

```
SELECT DISTINCT S.studid, S.imie, S.nazwisko,
(
SELECT numeralbumu
FROM NumeryAlbumu AS NA
WHERE S.studid = NA.studid
```

```
) AS NumerAlbumu
FROM Studenci AS S
```

100 %

	studid	imie	nazwisko	NumerAlbumu
1	1	Ali	Reina	6213
2	2	Neal	Moss	6214
3	3	Stanton	Matfin	6215
4	4	Jone	Klesel	6216
5	5	Laurens	Cruess	6217
6	6	Any	Tallyn	6218
7	7	Coletta	Jammes	6219
8	8	Bradly	Downs	6220
9	9	Felike	Bodycomb	6221
10	10	Hollis	Cropton	6222
11	11	Chelsie	Little	6223
12	12	Malinda	Davenport	6224

100 %

Results Messages

{500 rows affected}

Completion time: 2022-12-18T14:12:20.2224489+01:00

- Wyświetl ilość podań złożonych przez każdego ucznia. Posortuj wyniki malejąco zależnie od ilości podań

```
SELECT Studenci.imie, Studenci.nazwisko,
(
SELECT COUNT(*)
FROM Podania
WHERE Podania.studid = Konta.studid) AS LiczbaPodan
FROM Konta
INNER JOIN Studenci
ON Konta.studid = Studenci.studid
ORDER BY LiczbaPodan DESC
```

100 %

	imie	nazwisko	LiczbaPodan
1	Doreen	Ferrillo	4
2	Nigel	Brolly	4
3	Arne	Redholls	4
4	Quinton	Willmore	4
5	Cecilio	Entwhistle	3
6	Toddie	Grimwad	3
7	Sophia	Adderley	3
8	Dulcia	Streight	3
9	Sid	Ansteys	3
10	TAMARA	OWIER	3

100 %

(500 rows affected)

Completion time: 2022-12-18T14:11:25.2933363+01:00

## 13 Zapytania z podzapytaniem w części FROM

- Wypisz id studenta wraz z największą ilością oraz sumą nieuregulowanych opłat

```
SELECT TOP 1 O.studid, NI.ilosc, SUM(O.wysokoscoplaty) AS suma
FROM Oplaty AS O
INNER JOIN
(
SELECT TOP 1 WITH TIES studid, COUNT(wysokoscoplaty) AS ilosc
FROM Oplaty
WHERE dataoplatenia IS NULL
GROUP BY studid
ORDER BY ilosc DESC
) AS NI
ON O.studid = NI.studid
GROUP BY O.studid, NI.ilosc
ORDER BY suma DESC
```

100 %

	studid	ilosc	suma
1	375	3	1130.00



100 %	Results	Messages
(1 row affected)		
Completion time: 2022-12-18T14:10:42.7615763+01:00		

- Wypisz imiona i nazwiska studentów których suma ocen jest większa od 11.  
Podaj sumy ich ocen

```

SELECT S.imie, S.nazwisko, SO.SumaOcen
FROM Studenci AS S
INNER JOIN
(
SELECT studid, SUM(ocena) AS SumaOcen
FROM Oceny
GROUP BY studid
HAVING SUM(ocena) > 11
) AS SO
ON S.studid = SO.studid
ORDER BY S.studid ASC

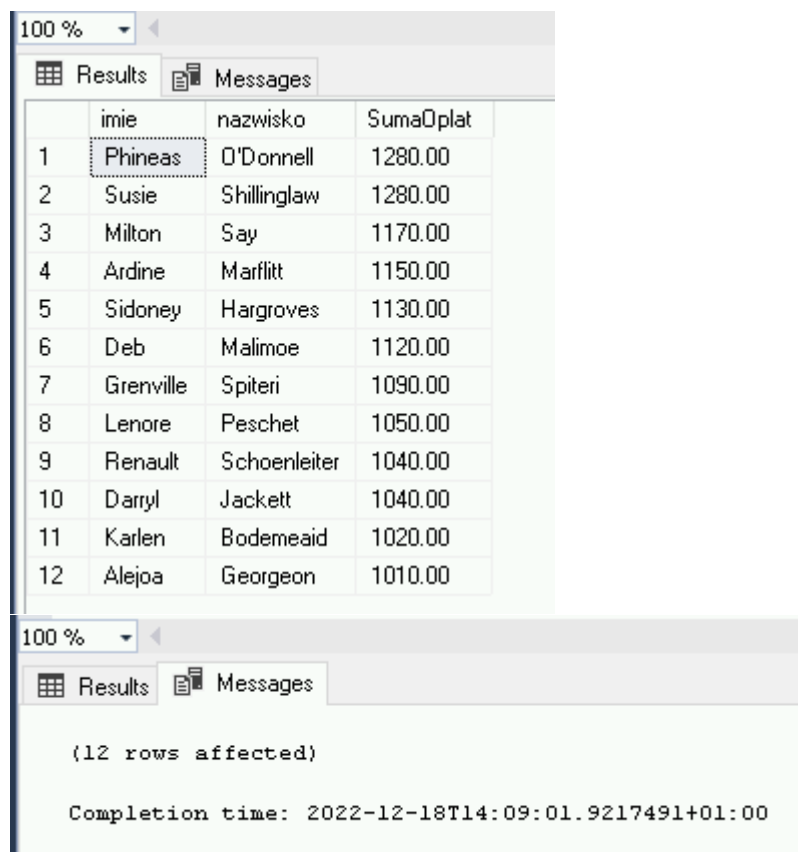
```

100 %			
Results		Messages	
	imie	nazwisko	SumaOcen
1	Laurens	Cruess	18.0
2	Bradly	Downs	14.0
3	Hollis	Cropton	14.0
4	Chelsie	Little	14.0
5	Zacharias	Monckman	13.0
6	June	Whoston	19.0
7	Norry	Mazzey	12.0
8	Emmet	Winkworth	16.0
9	Kristine	Izakov	13.0
10	Heddi	Domerq	13.0
11	Howard	Loosmore	12.0
12	Lockwood	Dulson	12.0
13	Carmela	Weber	17.0
14	Bail	Lackey	14.0
15	Tamara	Quar	14.0

100 %	Results	Messages
Warning: Null value is eliminated by an aggregate or other SET operation.		
(271 rows affected)		
Completion time: 2022-12-18T14:09:29.4161221+01:00		

- Wyświetl studentów, których suma opłat przekracza 1000 złotych

```
SELECT S.imie, S.nazwisko, SO.SumaOplat
FROM Studenci AS S
INNER JOIN
(
SELECT studid, SUM(wysokoscoplaty) AS SumaOplat
FROM Oplaty
GROUP BY studid
HAVING SUM(wysokoscoplaty) > 1000
) AS SO
ON S.studid = SO.studid
ORDER BY SO.SumaOplat DESC
```



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays the results of a query in a table with 4 columns: an index, 'imie' (first name), 'nazwisko' (last name), and 'SumaOplat' (sum of fees). The bottom pane shows the execution status, indicating that 12 rows were affected and providing the completion time.

	imie	nazwisko	SumaOplat
1	Phineas	O'Donnell	1280.00
2	Susie	Shillinglaw	1280.00
3	Milton	Say	1170.00
4	Ardine	Marflitt	1150.00
5	Sidoney	Hargroves	1130.00
6	Deb	Malimoe	1120.00
7	Grenville	Spiteri	1090.00
8	Lenore	Peschet	1050.00
9	Renault	Schoenleiter	1040.00
10	Darryl	Jackett	1040.00
11	Karlen	Bodemeaid	1020.00
12	Alejoa	Georgeon	1010.00

(12 rows affected)

Completion time: 2022-12-18T14:09:01.9217491+01:00

- Z wykorzystaniem Operatora PIVOT wyświetl średnie ze wszystkich średnich uczniów w latach 2015, 2016 oraz 2017

```
SELECT ROUND([2015],2) AS SredniaWszystkich2015,
ROUND([2016],2) AS SredniaWszystkich2016,
ROUND([2017],2) AS SredniaWszystkich2017
FROM (
SELECT P.rok, AVG(S.srednia) AS SredniaSrednich
FROM PoprzednieRokiSzkolne AS P
INNER JOIN SredniaKoncowa AS S
```

```

ON P.rokid = S.rokid
GROUP BY P.rok
) AS Tymczas
PIVOT(
AVG(SredniaSrednich)
FOR rok IN ([2015], [2016], [2017])) AS PV

```

The screenshot shows two windows from SQL Server Enterprise Manager. The top window displays the results of a PIVOT query. The bottom window shows the execution messages.

	SredniaWszystkich2015	SredniaWszystkich2016	SredniaWszystkich2017
1	4	3,99	4

Messages window:

```

(1 row affected)

Completion time: 2022-12-18T14:08:30.5324354+01:00

```

## 14 Zapytania z podzapytaniem w części WHERE

- Wypisz państwa, z których pochodzą zarówno studenci jak i pracownicy

```

SELECT DISTINCT S.panstwo
FROM Studenci AS S
WHERE EXISTS(
SELECT P.panstwo
FROM Pracownicy AS P
WHERE S.panstwo=P.panstwo)

```

The screenshot shows two windows from SQL Server Enterprise Manager. The top window displays the results of a query using EXISTS. The bottom window shows the execution messages.

	panstwo
1	Ukraine
2	United States

Messages window:

```

(2 rows affected)

Completion time: 2022-12-18T14:08:06.5216754+01:00

```

- Utwórz średnią z wszystkich średnich uczniów uzyskanych w 2016 roku

```

SELECT ROUND(AVG(srednia),2) AS SredniaSrednichZ2016Roku
FROM SredniaKoncowa
WHERE EXISTS(
SELECT rokid
FROM PoprzednieRokiSzkolne
WHERE SredniaKoncowa.roid = PoprzednieRokiSzkolne.roid
AND rok = 2016
)

```

Results Messages	
	SredniaSrednichZ2016Roku
1	3,99

Results Messages

(1 row affected)

Completion time: 2022-12-18T14:07:12.7527360+01:00

- Wypisz wydziały, w których ilość uczęszczających studentów jest większa niż 90

```

SELECT nazwawydzialu
FROM Wydzialy
WHERE EXISTS (
SELECT S.wydzialid, COUNT(S.studid) AS IloscStudentowNaWydziale
FROM Studenci AS S
WHERE S.wydzialid = Wydzialy.wydzialid
GROUP BY S.wydzialid
HAVING COUNT(S.studid) > 90
)

```

Results Messages	
	nazwawydzialu
1	Wydział Zarządzania
2	Wydział Sztuki

Results Messages

(2 rows affected)

Completion time: 2022-12-18T14:03:58.9087792+01:00