Projekt zaliczeniowy

temat: Baza danych szkoły

Twórcy:

- Michał Cyran
- Vladislav Kozulin
- Marcin Starzak

Podstawowe założenia projektu

Przedstawiona poniżej baza danych służy do obsługi pewnego liceum w Polsce. Jej główne cele to:

- przechowywanie danych o nauczycielach, pracownikach, uczniach i ich rodzicach;
- przechowywanie danych o klasach, zajęciach, salach lekcyjnych, planach lekcji;
- przechowywanie ocen zdobywanych przez uczniów na zajęciach oraz ich obecności na tychże;
- podstawowe funkcje elektronicznego dziennika dla nauczycieli i uczniów – nauczyciele mogą wpisywać oceny, sprawdzać obecność, natomiast uczniowie mogą przeglądać swoje oceny, sprawdzać swój plan lekcji itp.
- przechowywanie danych o nowych kandydatach do liceum oraz automatyczna obsługa rekrutacji – obliczanie punktów do rekrutacji, akceptacja kandydatów, przypisywanie zaakceptowanych kandydatów do klas;
- przechowywanie informacji o stypendiach i stypendystach
- podstawowa obsługa sklepiku szkolnego

Podczas projektowania bazy danych zostały założone następujące fakty na temat zasad pracy obsługiwanego liceum:

- Nie każdy pracownik/uczeń/rodzic jest Polakiem, a to oznacza że nie każdy z nich ma numer PESEL. Tym samym nie możemy identyfikować osób po tymże numerze. Zamiast tego, każda osoba posiada swoje ID unikalne w obrębie bazy danych, które zostaje jej nadane po wpisaniu do bazy (konkretnie do tabeli *people*, o czym później).
- Dla każdego rocznika szkoła przewiduje maksymalnie trzy 2-osobowe klasy. Limit 2 osób w klasie został wprowadzony przez twórców projektu, aby duża liczba rekordów nie zaciemniała wyników kwerend podczas prezentacji projektu.
- Każda z trzech klas ma przypisaną jedną z trzech specjalizacji oznaczonych symbolami:
  - m – klasa matematyczna
  - p – klasa polonistyczna
  - s – klasa biologiczno-chemiczna (s od science).
- Każda ocena wpisana przez nauczyciela ma do siebie przypisaną odpowiednią wagę, która obrazuje, jak znacząca jest dana ocena w danym przedmiocie.

Dodatkowo, poniżej znajduje się opis procesu rekrutacji w naszej szkole.

<u>Każdy kandydat ma wpisane następujące rzeczy w tabeli *candidates*:</u>

- *pl_exam_result* - wynik egzaminu (po zakończeniu szkoły podstawowej) z języka polskiego
- *math_exam_result* - wynik egzaminu z matematyki
- *science_exam_result* - wynik egzaminu z przedmiotów ścisłych (chemia, biologia itp.)
- *extracurlicural_act* - wyrażona przez true lub false informacja o tym, czy kandydat bierze udział w wolontariatach, udziela się społecznie itp.
- *chosen_class_symbol* - preferowana przez kandydata specjalizacja klasy licealnej
- *filling_date* - data i godzina zgłoszenia się do rekrutacji

Każdy kandydat ma na podstawie powyższych danych wyliczaną liczbę punktów wzorem:
*pts = pl_exam_result * 30% + math_exam_result * 30% + science_exam_result * 30% + extracurlicural_act * 10%*

Do szkoły przyjmowani są wyłącznie uczniowie, których liczba punktów jest >= 50. Dodatkowo, szkoła przyjmuje wyłącznie 6 najlepszych uczniów (po 2 na klasę). W przypadku gdy dwoje uczniów uzyskało taką samą liczbę punktów, wyższa pozycja w rankingu jest wyznaczana przez wcześniejszą datę zgłoszenia się do rekrutacji (*filling_date*).

Jak uczniowie są przydzielani do klas?

Uczniowie przyjęci do szkoły są ustawiani w kolejności względem ilości punktów uzyskanych przez nich na rekrutacji. Następnie każdy z nich po kolei jest przypisywany do klasy o takiej specjalizacji, jaka jest określona w *chosen_class_symbol*. Jeśli klasa z wybraną przez ucznia specjalizacją została już zapełniona (ma 2 osoby), to uczeń zostaje przypisany do kolejnej, najbliższej mu specjalizacji, według schematu:

Jeśli kandydat wybrał klasę polonistyczną (p):

p -> s -> m

Jeśli kandydat wybrał klasę matematyczną (m):

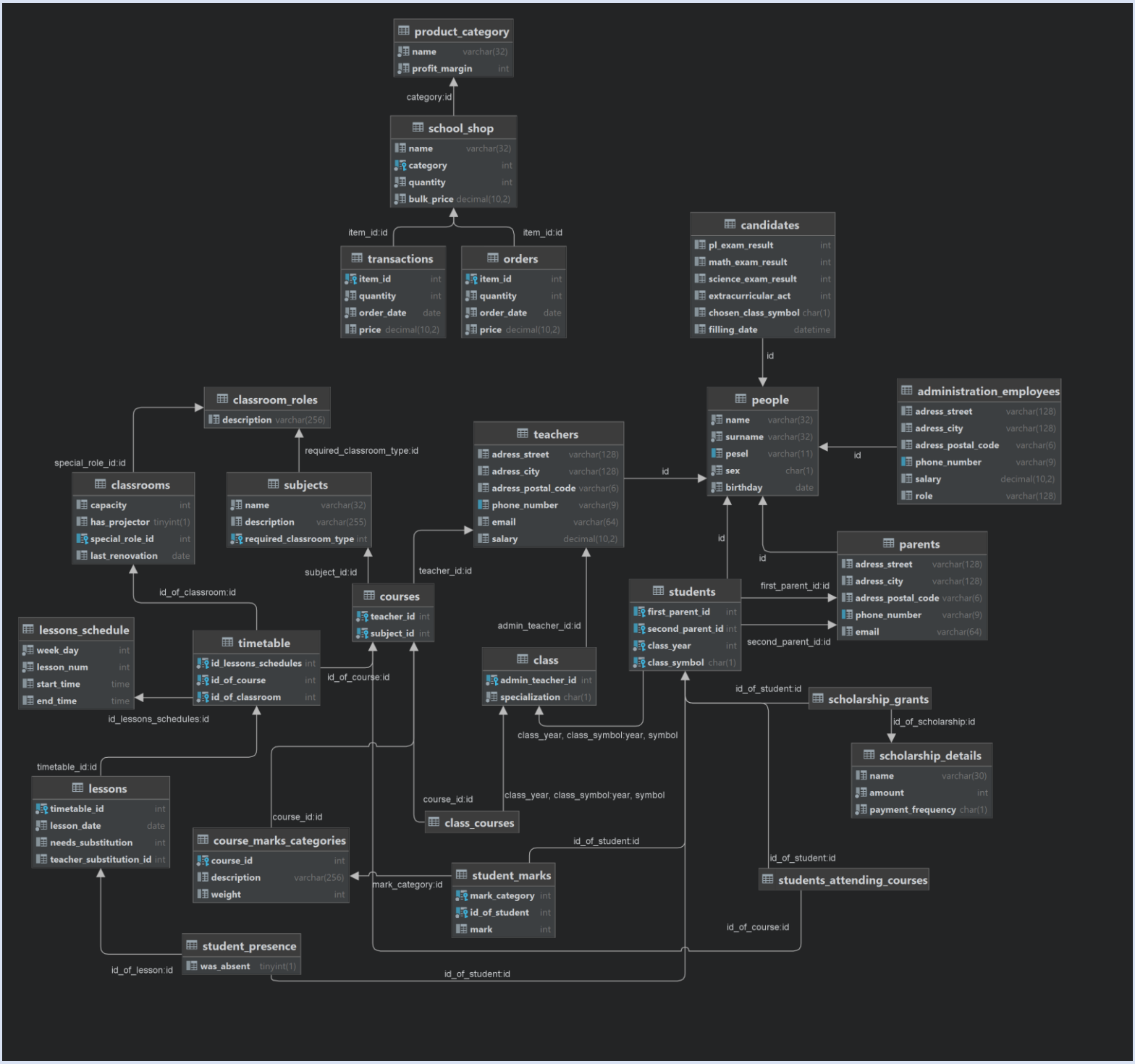
m -> s -> p

Jeśli kandydat wybrał klasę biol-chem (s):

s -> m -> p

W przypadku gdy w klasie o specjalizacji x zostało jedno miejsce, dwójka następnych w kolejności kandydatów ma tę samą liczbę punktów oraz obaj chcą trafić do klasy o specjalizacji x, miejsce w klasie dostanie ten kandydat, który miał lepszy wynik z egzaminu który odpowiada danej specjalizacji. Jeśli nadal nie pozwala to wyłonić który kandydat powinien dostać to miejsce (obaj mają taki sam wynik z egzaminu x), o przyznaniu miejsca decyduje kolejność zgłoszenia zapisana w kolumnie *filling_date*.

Diagram relacyjny



Opis realizacji dziedziczenia w bazie danych

W opisywanej bazie danych zastosowano metodę dziedziczenia o nazwie *Table Per Type (TPT).* Podstawowe dane wszystkich osób w bazie danych (nauczyciele, uczniowie, rodzice itd.) są przechowywane w ogólnej tabeli *people.* Następnie każda kategoria osób ma swoją podtabelę, połączoną kluczem obcym z tabelą *people.* Każda z tych podtabel zawiera już wyłącznie informacje istotne dla danej kategorii – np. *students* przechowuje w której klasie jest dany student, ale nie ma kolumny zawierającej pensję, bo tej uczniowie nie pobierają. Powyżej opisane rozwiązanie jest widoczne w prawym górnym rogu diagramu relacyjnego.

Opis stworzonych widoków

- **students_timetable** – wypisuje wszystkie zajęcia, na które uczęszcza każdy z uczniów liceum

	id	name	surname	subject_name	id_of_classroom	`week_day(ls.week_day)`	start_time	end_time
1	6	Pola	Kowalska	matematyka	108	poniedzialek	08:00:00	08:45:00
2	6	Pola	Kowalska	fizyka	50	poniedzialek	08:55:00	09:40:00
3	7	Miroslaw	Ptak	fizyka	50	poniedzialek	08:00:00	08:45:00
4	13	Marek	Ryba	matematyka	108	poniedzialek	08:00:00	08:45:00
5	13	Marek	Ryba	fizyka	50	poniedzialek	08:55:00	09:40:00

Z powyższego, przykładowego wyniku kwerendy *SELECT * FROM students_timetable* można oczytać na przykład, że uczeń o ID = 13, Marek Ryba, w poniedziałki od godziny 8:55 do 9:40 ma lekcje fizyki w sali nr 50.

- **journal_marks** – wypisuje wszystkie oceny wystawione w szkole

	id_of_student	name	surname	subject	description	mark	weight
1	6	Pola	Kowalska	fizyka	kartkowka - hydrostatyka	6	2
2	6	Pola	Kowalska	matematyka	sprawdzian - trygonometria	4	3
3	6	Pola	Kowalska	matematyka	aktywnosc	5	1
4	13	Marek	Ryba	fizyka	kartkowka - hydrostatyka	4	2
5	13	Marek	Ryba	matematyka	sprawdzian - trygonometria	4	3

Z powyższego, przykładowego wyniku kwerendy *SELECT * FROM journal_marks* można oczytać na przykład, że uczennica o ID = 6, Pola Kowalska, otrzymała ze sprawdzianu z trygonometrii ocenę 4 z wagą 3.

- **journal_presence** – wypisuje wszystkie lekcje na których powinien być każdy uczeń liceum oraz informację o tym, czy był na nich obecny

	id_of_student	name	surname	subject	lesson_date	status
1	6	Pola	Kowalska	fizyka	2012-12-04	present
2	6	Pola	Kowalska	matematyka	2012-12-05	present
3	6	Pola	Kowalska	matematyka	2012-12-07	absent
4	7	Miroslaw	Ptak	fizyka	2012-12-02	present

Z powyższego, przykładowego wyniku kwerendy *SELECT * FROM journal_presence* można oczytać na przykład, że uczennica o ID = 6, Pola Kowalska, była obecna na lekcji matematyki w dniu 5 grudnia, ale już 7 grudnia była nieobecna.

- **lessons_needing_substitution** – wypisuje wszystkie lekcje, które nie mogą się odbyć, ponieważ prowadzący je nauczyciel danego dnia jest na urlopie/zwolnieniu lekarskim.

	id	lesson_date	week_day	start_time	end_time	name
1	5	2023-04-01	poniedzialek	08:00:00	08:45:00	matematyka

Z powyższego, przykładowego wyniku kwerendy *SELECT * FROM lessons_needing_substitution* można oczytać, że 1 kwietnia 2023r o godzinie 8:00 miała się odbyć matematyka, ale nie będzie wtedy nauczyciela prowadzącego. Sekretariat powinien znaleźć innego matematyka na zastępstwo i wpisać go w odpowiednie miejsce w bazie danych.

- **candidates_stats** – wypisuje wszystkich kandydatów na nowych uczniów liceum wraz z liczbą punktów uzyskanych przez nich w rekrutacji oraz informacją, czy się dostaną.

	id	name	surname	points	Decision
1	19	Lukasz	Mickiewicz	92.20	Approved
2	18	Joanna	Paczesniak	81.00	Approved
3	14	Anna	Kandydacka	69.60	Approved
4	17	Mikolaj	Matejko	67.00	Approved
5	20	Marzena	Nowicka	61.00	Approved
6	21	Jakub	Iwanecki	60.00	Approved
7	16	Maciej	Geniusz	54.00	Not approved (reserve)
8	15	Marcin	Sredni-Kandydat	45.00	Not approved

Z powyższego, przykładowego wyniku kwerendy *SELECT * FROM candidates_stats* można oczytać na przykład, że Mikołaj Matejko uzyskał w rekrutacji 67 punków i dostał się do szkoły, ale Marcin Średni-Kandydat dostał tylko 45 punków, co sprawiło, że nie został przyjęty.

Opis procedur

- **`get_average_mark (student_id int, course_id int)`** – wylicza i wypisuje średnią ocen ucznia o ID zadanym pierwszym argumentem z przedmiotu (kursu) zadanego drugim argumentem. Przykładowo wynikiem `call get_average_mark(6, 1)` jest:

	srednia
1	4.25

- **`get_students_of_teacher (teacher_id int)`** – wypisuje listę wszystkich uczniów nauczanych przez nauczyciela o zadanym ID. Przykładowo wynikiem `call get_students_of_teacher(8)` jest:

	id	name	surname	class
1	6	Pola	Kowalska	1a
2	13	Marek	Ryba	1a

- **`get_parents_contact_info (student_id int)`** – wypisuje informacje kontaktowe do rodziców ucznia o zadanym ID. Przykładowo wynikiem `call get_parents_contact_info(6)` jest:

	Name	Surname	Adress	Phone
1	Jerzy	Kowalski	ul. Polska 3, 12-345 Krakow	123456789
2	Anna	Kowalska	ul. Norymberska 21, 33-333 Warszawa	987333321

- **`class_timetable (year int, symbol char)`** – wypisuje plan lekcji dla klasy zadanej argumentem. Przykładowo `call class_timetable(1, 'a')` wypisuje plan lekcji klasy 1a:

	week_d...	lesson_num	start...	end...	name	id_of_classroom	name	surname
1	poniedzialek	1	08:00:00	08:45:00	matematyka	108	Bob	Borsuk
2	poniedzialek	2	08:55:00	09:40:00	fizyka	50	Bob	Borsuk

- **`show_class (year int, symbol char)`** – wypisuje uczniów z klasy zadanej argumentem. Przykładowo `call show_class(1, 'a')` wypisuje uczniów klasy 1a:

	id	name	surname
1	6	Pola	Kowalska
2	13	Marek	Ryba

- **`propose_new_classes()`** – bezargumentowa procedura, która wybiera najlepszych 6 kandydatów do liceum oraz proponuje nowe klasy zgodnie z opisanymi wcześniej zasadami rekrutacji do liceum. Po zakończeniu działania wypisuje wynik – nowe klasy:

	ID	Name	Surname
1	Klasa A	Specializacja: m	Wychowawaca: Bob Borsuk
2	18	Joanna	Paczesniak
3	19	Lukasz	Mickiewicz
4	Klasa B	Specializacja: s	Wychowawaca: Aneta Zalewska
5	14	Anna	Kandydacka
6	17	Mikolaj	Matejko
7	Klasa C	Specializacja: p	Wychowawaca: Aneta Zalewska
8	20	Marzena	Nowicka
9	21	Jakub	Iwanecki

Opis wyzwalaczy

- **`insert_transaction`** – wyzwalacz działający na tabeli *transactions,* wykonujący się przed operacją wstawienia nowej transakcji w sklepiku szkolnym do tabeli. Wyzwalacz automatycznie ustawia kwotę transakcji z uwzględnieniem ilości zakupionych produktów oraz marży przypisanej do danej kategorii produktów. Dodatkowo, wyzwalacz sprawdza stan magazynowy sklepiku szkolnego. Jeżeli transakcja ma przypisaną do siebie większą liczbę produktów niż na stanie, to jest odrzucana. W przeciwnym przypadku sprawdza, czy po dokonaniu transakcji liczba sztuk danego produktu jest mniejsza niż 5 - jeśli tak, to dodaje do tabeli *orders* zamówienie na pewną liczbę sztuk danego produktu, które trzeba zamówić na kolejny dzień.
- **`insert_class_courses`** – wyzwalacz działający na tabeli *class_courses*, wykonujący się po operacji wstawiania nowego rekordu. Po przypisaniu kursu do nowej klasy, wyzwalacz automatycznie zapisuje wszystkich członków tej klasy na ten kurs.
- **`insert_students`** – wyzwalacz działający na tabeli *students*, wykonujący się po operacji wstawienia nowego rekordu. Po przypisaniu nowego studenta do klasy, jest on automatycznie zapisywany na wszystkie zajęcia przewidziane dla tej kasy.
- **`insert_vacations`** – wyzwalacz działający na tabeli vacations, wykonujący się po operacji wstawienia nowego rekordu. Po wpisaniu nowego urlopu dla danego nauczyciela, wyzwalacz sprawdza czy podczas jego urlopu nie są wpisane zajęcia które ma on prowadzić. Jeśli są, to do każdych z tych zajęć dodawana jest flaga, że potrzebują one zastępstwa.
- **`delete_vacations`** – wyzwalacz działający na tabeli *vacations*, wykonujący się po operacji usuwania rekordu z tabeli. Działa analogicznie do *insert_vacations*, z tą różnicą, że sprawdza on, czy usunięcie urlopu sprawia, że zajęcia, dla których dodano flagę informującą o potrzebie zastępstwa, mogą być jednak poprowadzone przez nauczyciela, któremu usunięto urlop. W takich sytuacjach usuwa on tę flagę.
- **`update_vacations`** – wyzwalacz działający na tabeli *vacations*, wykonujący się po operacji aktualizacji rekordu z tabeli. Działa jak połączenie *insert_vacations* i *delete_vacations*.
- **`insert_lesson`** – wyzwalacz działający na tabeli *lessons*, wykonujący się przed operacją wstawiania nowego rekordu do tabeli. Sprawdza on listę wpisanych urlopów i na jej podstawie decyduje, czy dana lekcja wymaga dodania flagi informującej o potrzebie znalezienia nauczyciela na zastępstwo.
- **`update_lesson`** - wyzwalacz działający na tabeli *lessons*, wykonujący się przed operacją aktualizacji rekordu z tabeli. Działa analogicznie do *insert_lesson.*

Wartości zmieniające się w czasie oraz kopie zapasowe

Tabela *lessons* posiada kolumnę *is_in_progress*, która przechowuje wartość true/false informującą o tym, czy dana lekcja jest w trakcie. Flaga ta jest aktualizowana na bieżąco, przy użyciu skryptu napisanego w języku Bash. Ten sam skrypt zajmuje się także obsługą kopii zapasowej bazy danych. Kopia ta jest zapisywana co godzinę, a każda z nich jest przechowywana na serwerze przez kolejne 10 godzin.

Typowe zapytania

Przykładowym zapytaniem do naszej bazy danych jest wyświetlenie planu lekcji danej osoby:

```
SELECT * FROM students_timetable
WHERE id = (
    SELECT id FROM people
    WHERE name = "Pola" AND surname = "Kowalska")
```

W wyniku tej kwerendy otrzymamy plan lekcji uczennicy Poli Kowalskiej:

	id	name	surname	subject_name	id_of_classroom	week_day(ls.week_day)	start_time	end_time
1	6	Pola	Kowalska	matematyka	108	poniedzialek	08:00:00	08:45:00
2	6	Pola	Kowalska	fizyka	50	poniedzialek	08:55:00	09:40:00

Innym, dla ucznia szkoły nawet ważniejszym zapytaniem, będzie zapytanie dotyczące wyświetlenia ocen z danego przedmiotu. Stosownym zapytaniem będzie na przykład:

```
SELECT * FROM journal_marks
WHERE id_of_student = (
    SELECT id FROM people
    WHERE name = ‘Pola’
    AND surname = ‘Kowalska’)
AND subject = 'matematyka'
```

W wyniku tego zapytania otrzymamy oceny Poli Kowalskiej z matematyki:

	id_of_student	name	surname	subject	description	mark	weight
1	6	Pola	Kowalska	matematyka	sprawdzian - trygonometria	4	3
2	6	Pola	Kowalska	matematyka	aktywnosc	5	1

Możemy także pokusić się o bardziej skomplikowane zapytania do bazy danych. Poniższa kwerenda pozwala uzyskać procent frekwencji danego ucznia na danym przedmiocie.

```
SELECT TRUNCATE(
        100 * count(*) / (
        SELECT count(*) FROM journal_presence
        WHERE id_of_student = (
            SELECT id FROM people
            WHERE name = "Pola"
            AND surname = "Kowalska"
            )
        AND subject = 'matematyka'
    ), 2) AS '% frekwencji' FROM journal_presence
    WHERE id_of_student = (
        SELECT id FROM people
        WHERE name = "Pola"
        AND surname = "Kowalska"
    )
    AND status = 'present'
    AND subject = 'matematyka'
```

W wyniku czego uzyskamy dość niezadowalającą informację o frekwencji Poli Kowalskiej na matematyce.

	`% frekwencji`
1	0.00