

Baza danych szkoły

Pola Walczyńska

Podstawowe założenia projektu

Przedstawiona poniżej baza danych służy do obsługi szkoły średniej. 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
- 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 sklepu 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 tym 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).
- Dla każdego rocznika szkoła przewiduje maksymalnie trzy 2-osobowe klasy. Limit 2 osób w klasie został wprowadzony, 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.

Każdy kandydat ma wpisane następujące informacje w tabeli candidates:

- 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.)
- extracurricular_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
- `filling_date` - data i godzina zgłoszenia się do rekrutacji

Każdy kandydat ma na podstawie powyższych danych wyliczaną liczbę punktów wzorem:

$$pts = pl_exam_result * 30\% + math_exam_result * 30\% + science_exam_result * 30\% + extracurricular_act * 10\%$$

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 na podstawie ilości punktów uzyskanych podczas 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 specjalizacją została już zapełniona, to uczeń zostaje przypisany do kolejnej, najbliższej mu specjalizacji, według schematu:

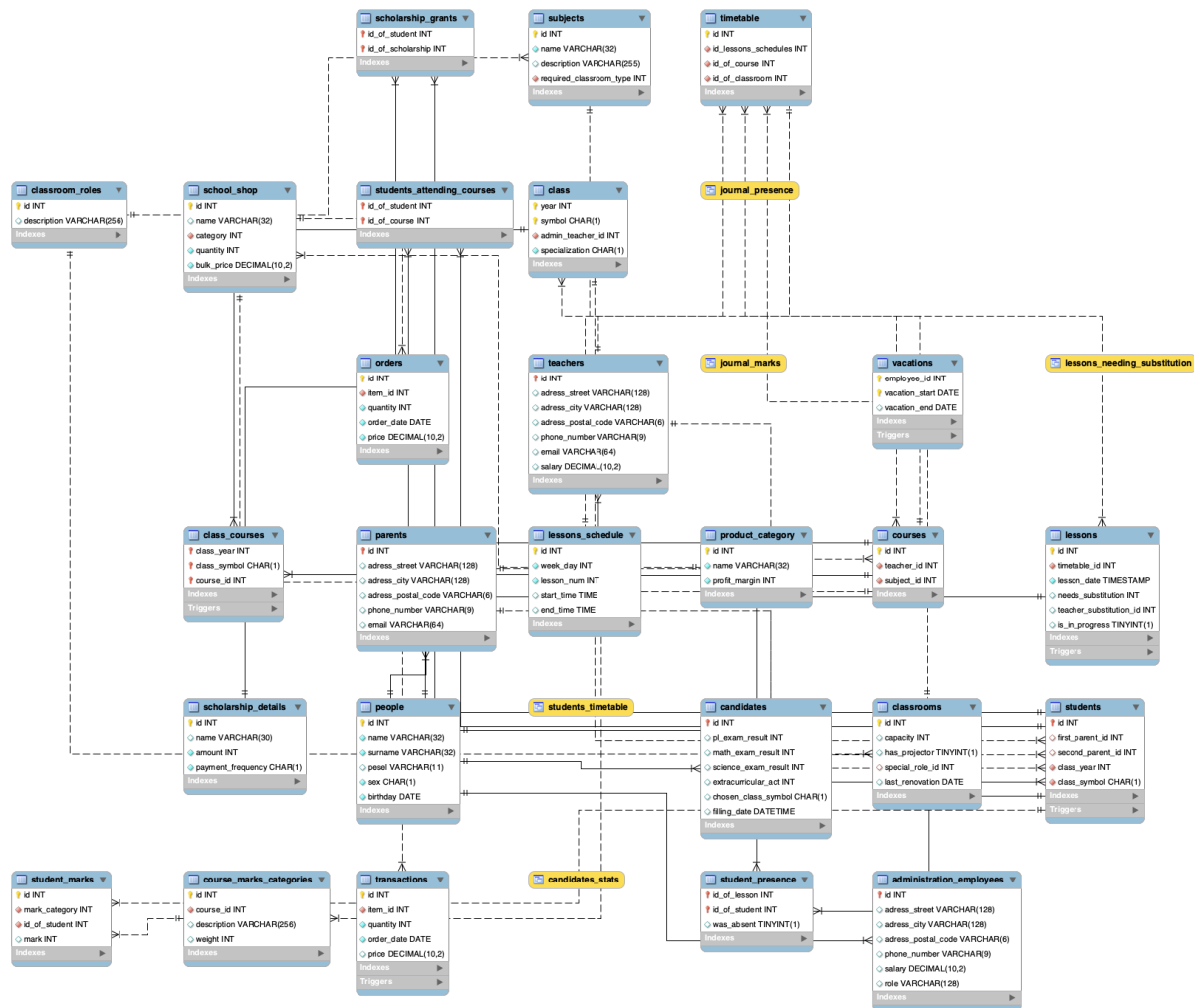
*** Jeśli kandydat wybrał klasę polonistyczną (p): p -> s -> m

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



Realizacja dziedziczenia w bazie danych

W bazie danych została zastosowana metoda dziedziczenia 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 uczeń, ale nie ma kolumny zawierającej pensję, bo uczniowie jej nie pobierają.

Widoki

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

id	name	surname	subject_name	id_of_classroom	week_day	start_time	end_time
6	Vlad	Kowalski	matematyka	108	1	08:00:00	08:45:00
6	Vlad	Kowalski	fizyka	50	1	08:55:00	09:40:00
7	Mirosław	Ptak	fizyka	50	1	08:00:00	08:45:00
13	Marek	Ryba	matematyka	108	1	08:00:00	08:45:00
13	Marek	Ryba	fizyka	50	1	08:55:00	09:40:00

- journal_marks – wypisuje wszystkie oceny wystawione w szkole

id_of_stude...	name	surname	subject	description	mark	weight
6	Vlad	Kowalski	fizyka	kartkowka - hydrostatyka	6	2
6	Vlad	Kowalski	matematyka	sprawdzian - trygonometria	4	3
6	Vlad	Kowalski	matematyka	aktywnosc	5	1
13	Marek	Ryba	fizyka	kartkowka - hydrostatyka	4	2
13	Marek	Ryba	matematyka	sprawdzian - trygonometria	4	3

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

id_of_stude...	name	surname	subject	lesson_date	status
6	Vlad	Kowalski	matematyka	2012-12-05 09:40:00	present
6	Vlad	Kowalski	matematyka	2012-12-07 08:00:00	absent
6	Vlad	Kowalski	fizyka	2023-02-01 13:30:00	present
7	Mirosław	Ptak	fizyka	2012-12-02 08:00:00	present

- 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
5	2023-04-01 00:00:00	poniedziałek	08:00:00	08:45:00	matematyka

- `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
19	Lukasz	Mickiewicz	92.20	Approved
18	Joanna	Paczesniak	81.00	Approved
14	Anna	Kandydacka	69.60	Approved
17	Mikolaj	Matejko	67.00	Approved
20	Marzena	Nowicka	61.00	Approved
21	Jakub	Iwanecki	60.00	Approved
16	Maciej	Geniusz	54.00	Not approved (reserve)
15	Marcin	Sredni-Kandydat	45.00	Not approved

Procedury

- `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.

srednia
4.25

- `get_students_of_teacher` (`teacher_id` int) – wypisuje listę wszystkich uczniów nauczanych przez nauczyciela o zadanym ID

id	name	surname	class
6	Vlad	Kowalski	1a
13	Marek	Ryba	1a

- `get_parents_contact_info` (`student_id` int) – wypisuje informacje kontaktowe rodziców ucznia o zadanym ID

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

- `class_timetable` (year int, symbol char) – wypisuje plan lekcji dla klasy zadanej argumentem

week_day...	lesson_num	start_time	end_time	name	id_of_classroom	name	surname
poniedziałek	1	08:00:00	08:45:00	matematyka	108	Bob	Borsuk
poniedziałek	2	08:55:00	09:40:00	fizyka	50	Bob	Borsuk

- `show_class` (year int, symbol char) – wypisuje uczniów z klasy zadanej argumentem

id	name	surname
6	Vlad	Kowalski
13	Marek	Ryba

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

ID	Name	Surname
Klasa A	Specializacja: m	Wychowawca: Bob Borsuk
18	Joanna	Paczesniak
19	Lukasz	Mickiewicz
Klasa B	Specializacja: s	Wychowawca: Aneta Zalewska
14	Anna	Kandydacka
17	Mikolaj	Matejko
Klasa C	Specializacja: p	Wychowawca: Aneta Zalewska
20	Marzena	Nowicka
21	Jakub	Iwanecki

Wyzwalacze

- `insert_transaction` – wyzwalacz działający na tabeli `transactions`, wykonujący się przed operacją wstawienia nowej transakcji w sklepiu 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 sklepu 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 klasy.
- `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 każdego 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 i 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.