

Politechnika Łódzka

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki

Szkoła Muzyczna

Obiektowa Baza Danych Oracle

Rozproszone i Obiektowe Bazy Danych

Autorzy: Igor Typinski (251237)
Mateusz Mroz (251190)

Grupa: 5

Temat: Szkoła muzyczna (z naciskiem na rozwój ucznia)

Łódź, styczeń 2026

Spis treści

1 Opis projektu	3
1.1 Cel i zakres	3
1.2 Przyjęte ograniczenia	3
1.3 Technologia	4
2 Typy obiektowe	5
2.1 t_instrument_obj	5
2.2 t_lista_instrumentow	5
2.3 t_sala_obj	5
2.4 t_nauczyciel_obj	6
2.5 t_uczen_obj	6
2.6 t_kurs_obj	6
2.7 t_lekcja_obj	6
2.8 t_ocena_obj	7
3 Tabele obiektowe	8
3.1 t_instrument	8
3.2 t_sala	8
3.3 t_nauczyciel	8
3.4 t_uczen	8
3.5 t_kurs	9
3.6 t_lekcja	9
3.7 t_ocena	9
3.8 Referencje (REF/DEREF)	9
3.9 Sekwencje	9
4 Pakiety PL/SQL	11
4.1 pkg_uczen	11
4.2 pkg_lekcja	11
4.3 pkg_ocena	11
5 Wyzwalacze (Triggery)	12
6 Obsługa błędów	13
6.1 Kody błędów aplikacji	13
6.2 Scenariusze testowe	13
6.2.1 Scenariusz 1: Dane podstawowe	13
6.2.2 Scenariusz 2: Walidacja wieku ucznia	13
6.2.3 Scenariusz 3: Dni robocze	13
6.2.4 Scenariusz 4: Godziny dla dzieci	14
6.2.5 Scenariusz 5: Limit nauczyciela	14
6.2.6 Scenariusz 6: Limit ucznia	14
6.2.7 Scenariusz 7: Konflikty czasowe	14
6.2.8 Scenariusz 8: Kompetencje nauczyciela	14
6.2.9 Scenariusz 9: Blokada usuwania	14
6.2.10 Scenariusz 10: Pakiety CRUD	14
6.2.11 Scenariusz 11: Metody obiektów	15

7	Role użytkowników	16
7.1	Administrator (rola_admin)	16
7.2	Nauczyciel (rola_nauczyciel)	16
7.3	Sekretariat (rola_sekretariat)	16
7.4	Macierz uprawnień	16
7.5	Użytkownicy testowi	17
8	Diagram relacji obiektów	18
9	Struktura plików projektu	19

1 Opis projektu

1.1 Cel i zakres

Projekt przedstawia obiektową bazę danych dla szkoły muzycznej, ze szczególnym uwzględnieniem śledzenia rozwoju uczniów. Szkoła zajmuje się wyłącznie nauką muzyki w trybie indywidualnym. System został zaprojektowany z myślą o codziennej pracy sekretariatu, nauczycieli oraz dyrekcji szkoły.

Głównym celem projektu jest stworzenie kompleksowego systemu informatycznego umożliwiającego:

- **Zarządzanie danymi uczniów** – rejestracja nowych uczniów, przechowywanie danych osobowych, automatyczne obliczanie wieku i statusu (dziecko/dorosły)
- **Zarządzanie danymi nauczycieli** – ewidencja kadry pedagogicznej wraz z listą instrumentów, których mogą uczyć (przechowywana jako kolekcja VARRAY)
- **Planowanie lekcji** – tworzenie harmonogramu zajęć z uwzględnieniem dostępności sal, nauczycieli oraz ograniczeń czasowych dla dzieci
- **Rezerwacja sal lekcyjnych** – przydzielanie sal z odpowiednim wyposażeniem (fortepian, perkusja) do poszczególnych lekcji
- **Ocenianie postępu uczniów** – wystawianie ocen w różnych obszarach muzycznych (technika, teoria, słuch, rytm, interpretacja)
- **Kontrola reguł biznesowych** – automatyczna walidacja limitów obciążenia nauczycieli, liczby lekcji uczniów, konfliktów czasowych
- **Generowanie raportów** – plan dnia, obciążenie nauczycieli, historia ocen, raporty postępu uczniów

System uwzględnia specyfikę szkoły muzycznej, w której większość uczniów to dzieci uczęszczające równolegle do szkoły ogólnokształcącej. Dlatego lekcje dla dzieci (poniżej 15 lat) mogą odbywać się wyłącznie w godzinach popołudniowych (14:00–19:00), po zakończeniu zajęć szkolnych.

1.2 Przyjęte ograniczenia

W projekcie przyjęto następujące ograniczenia biznesowe:

1. **Minimalny wiek ucznia:** 5 lat – młodsze dzieci nie są przyjmowane do szkoły
2. **Definicja dziecka:** uczeń poniżej 15 lat – podlega dodatkowym ograniczeniom godzinowym
3. **Godziny lekcji dla dzieci:** tylko 14:00–19:00 – po zakończeniu zajęć w szkole ogólnokształcącej
4. **Dni pracy szkoły:** poniedziałek–piątek – szkoła nie prowadzi zajęć w weekendy
5. **Godziny pracy szkoły:** 08:00–20:00 – lekcje mogą być planowane tylko w tych godzinach

6. **Maksymalna liczba instrumentów nauczyciela:** 5 – ograniczenie kolekcji VAR-RAY
7. **Maksymalne obciążenie nauczyciela:** 6 godzin (360 minut) dziennie
8. **Maksymalna liczba lekcji ucznia:** 2 lekcje dziennie
9. **Czas trwania lekcji:** 30, 45, 60 lub 90 minut
10. **Skala ocen:** 1–6 (polska skala szkolna)
11. **Obszary oceny:** technika, teoria, słuch, interpretacja, improwizacja
12. **Poziomy kursów:** początkujący, średniozaawansowany, zaawansowany
13. **Statusy lekcji:** zaplanowana, odbyta, odwołana
14. **Kategorie instrumentów:** dęte, strunowe, perkusyjne, klawiszowe
15. **Brak konfliktów czasowych:** ta sama sala, nauczyciel lub uczeń nie mogą mieć dwóch lekcji w tym samym czasie
16. **Ochrona danych:** nie można usunąć nauczyciela lub ucznia z zaplanowanymi lekcjami

1.3 Technologia

- Oracle Database
- Podejście obiektowo-relacyjne
- Język PL/SQL

2 Typy obiektowe

W projekcie zdefiniowano 8 typów obiektowych z łącznie 14 metodami.

Typ	Metody	Opis
t_instrument_obj	1	Instrument muzyczny
t_lista_instrumentow	–	VARRAY(5) nazw instrumentow
t_sala_obj	1	Sala lekcyjna z wyposażeniem
t_nauczyciel_obj	3	Nauczyciel z lista instrumentow
t_uczen_obj	4	Uczeń szkoły muzycznej
t_kurs_obj	1	Kurs nauki gry na instrumencie
t_lekcja_obj	2	Pojedyncza lekcja muzyki
t_ocena_obj	2	Ocena postępu ucznia
Razem:		8 typów, 14 metod

Tabela 1: Typy obiektowe w projekcie

2.1 t_instrument_obj

Reprezentuje instrument muzyczny w słowniku instrumentów.

Atrybuty:

- id_instrumentu (NUMBER) – unikalny identyfikator
- nazwa (VARCHAR2(50)) – nazwa instrumentu (np. Fortepian, Gitara)
- kategoria (VARCHAR2(20)) – kategoria: dęte, strunowe, perkusyjne, klawiszowe

Metody:

- info() RETURN VARCHAR2 – zwraca nazwę z kategorią w nawiasie

2.2 t_lista_instrumentow

Kolekcja VARRAY przechowująca nazwy instrumentów, których może uczyć nauczyciel.

```
CREATE OR REPLACE TYPE t_lista_instrumentow AS VARRAY(5) OF VARCHAR2(50);
```

Ograniczenie do 5 elementów wynika z założenia, że nauczyciel specjalizuje się w kilku pokrewnych instrumentach. Weryfikacja kompetencji odbywa się przy zatrudnieniu przez dyrektora szkoły.

2.3 t_sala_obj

Reprezentuje salę lekcyjną z informacją o wyposażeniu.

Atrybuty:

- id_sali (NUMBER) – unikalny identyfikator
- nazwa (VARCHAR2(50)) – nazwa sali (unikalna, np. Sala A1)
- pojemnosc (NUMBER) – maksymalna liczba osób (1–20)
- ma_fortepian (CHAR(1)) – 'T' lub 'N'
- ma_perkusje (CHAR(1)) – 'T' lub 'N'

Metody:

- opis_pelny() RETURN VARCHAR2 – zwraca nazwę z informacją o wyposażeniu

2.4 t_nauczyciel_obj

Reprezentuje nauczyciela szkoły muzycznej wraz z listą instrumentów.

Atrybuty:

- id_nauczyciela (NUMBER) – unikalny identyfikator
- imie (VARCHAR2(50)) – imię nauczyciela
- nazwisko (VARCHAR2(50)) – nazwisko nauczyciela
- email (VARCHAR2(100)) – adres email (unikalny)
- telefon (VARCHAR2(20)) – numer telefonu kontaktowego
- data_zatrudnienia (DATE) – data rozpoczęcia pracy
- instrumenty (t_lista_instrumentow) – VARRAY instrumentów

Metody:

- pelne_dane() RETURN VARCHAR2 – zwraca imię i nazwisko
- lata_stazu() RETURN NUMBER – oblicza liczbę lat pracy w szkole
- liczba_instrumentow() RETURN NUMBER – zwraca liczbę elementów w VARRAY

2.5 t_uczen_obj

Reprezentuje ucznia szkoły muzycznej.

Atrybuty:

- id_ucznia (NUMBER) – unikalny identyfikator
- imie (VARCHAR2(50)) – imię ucznia
- nazwisko (VARCHAR2(50)) – nazwisko ucznia
- data_urodzenia (DATE) – data urodzenia
- email (VARCHAR2(100)) – adres email (opcjonalny)
- data_zapisu (DATE) – data rejestracji w szkole

Metody:

- wiek() RETURN NUMBER – oblicza aktualny wiek w latach
- pelne_dane() RETURN VARCHAR2 – zwraca imię, nazwisko i wiek
- czy_pelnoletni() RETURN VARCHAR2 – zwraca 'TAK' jeśli wiek ≥ 18 , inaczej 'NIE'
- czy_dziecko() RETURN CHAR – zwraca 'T' jeśli wiek < 15 lat, inaczej 'N'

2.6 t_kurs_obj

Reprezentuje kurs nauki gry na instrumencie.

Atrybuty:

- id_kursu (NUMBER) – unikalny identyfikator
- nazwa (VARCHAR2(100)) – nazwa kursu
- poziom (VARCHAR2(20)) – poczatkujacy, sredniozaawansowany, zaawansowany
- cena_za_lekcje (NUMBER) – cena jednej lekcji w PLN
- ref_instrument (REF t_instrument_obj) – referencja do instrumentu

Metody:

- info() RETURN VARCHAR2 – zwraca nazwę z poziomem i ceną

2.7 t_lekcja_obj

Reprezentuje pojedynczą lekcję muzyki.

Atrybuty:

- id_lekcji (NUMBER) – unikalny identyfikator
- data_lekcji (DATE) – data lekcji
- godzina_start (VARCHAR2(5)) – godzina rozpoczęcia (HH:MM)
- czas_trwania (NUMBER) – czas w minutach (30, 45, 60, 90)
- status (VARCHAR2(20)) – zaplanowana, odbyta, odwołana
- ref_uczen (REF t_uczen_obj) – referencja do ucznia
- ref_nauczyciel (REF t_nauczyciel_obj) – referencja do nauczyciela
- ref_kurs (REF t_kurs_obj) – referencja do kursu
- ref_sala (REF t_sala_obj) – referencja do sali

Metody:

- czas_txt() RETURN VARCHAR2 – zwraca czas trwania jako tekst (np. '45 min')
- czy_odbыта() RETURN VARCHAR2 – zwraca 'TAK' jeśli status = 'odbыта', inaczej 'NIE'

2.8 t_ocena_obj

Reprezentuje ocenę postępu ucznia.

Atrybuty:

- id_oceny (NUMBER) – unikalny identyfikator
- data_oceny (DATE) – data wystawienia oceny
- ocena (NUMBER) – wartość 1–6
- obszar (VARCHAR2(50)) – obszar oceny
- komentarz (VARCHAR2(500)) – komentarz nauczyciela (opcjonalny)
- ref_uczen (REF t_uczen_obj) – referencja do ucznia
- ref_nauczyciel (REF t_nauczyciel_obj) – referencja do nauczyciela

Metody:

- ocena_słownie() RETURN VARCHAR2 – zwraca ocenę słownie (celujący, bardzo dobry, itd.)
- czy_pozitwna() RETURN VARCHAR2 – zwraca 'TAK' jeśli ocena ≥ 2 , inaczej 'NIE'

3 Tabele obiektowe

Utworzono 7 tabel obiektowych przechowujących dane.

Tabela	Typ bazowy	Referencje (REF)
t_instrument	t_instrument_obj	–
t_sala	t_sala_obj	–
t_nauczyciel	t_nauczyciel_obj	– (zawiera VARRAY)
t_uczen	t_uczen_obj	–
t_kurs	t_kurs_obj	ref_instrument
t_lekcja	t_lekcja_obj	ref_uczen, ref_nauczyciel, ref_kurs, ref_sala
t_ocena	t_ocena_obj	ref_uczen, ref_nauczyciel

Tabela 2: Tabele obiektowe i ich referencje

3.1 t_instrument

Słownik instrumentów muzycznych.

- Klucz główny: id_instrumentu
- Ograniczenia NOT NULL: nazwa, kategoria
- CHECK: kategoria IN ('dete', 'strunowe', 'perkusyjne', 'klawiszowe')

3.2 t_sala

Informacje o salach lekcyjnych z wyposażeniem.

- Klucz główny: id_sali
- Ograniczenie UNIQUE: nazwa
- CHECK: pojemsosc BETWEEN 1 AND 20
- CHECK: ma_fortepian IN ('T', 'N')
- CHECK: ma_perkusje IN ('T', 'N')

3.3 t_nauczyciel

Dane nauczycieli wraz z kolekcją instrumentów (VARRAY).

- Klucz główny: id_nauczyciela
- Ograniczenie UNIQUE: email
- Ograniczenia NOT NULL: imie, nazwisko, email, data_zatrudnienia
- CHECK: email LIKE '%@%'
- Zawiera kolekcję t_lista_instrumentow

3.4 t_uczen

Dane uczniów szkoły muzycznej.

- Klucz główny: id_ucznia
- Ograniczenie UNIQUE: email (jeśli podany)
- Ograniczenia NOT NULL: imie, nazwisko, data_urodzenia, data_zapisu
- CHECK: email IS NULL OR email LIKE '%@%'
- Indeks: idx_uczen_nazwisko (nazwisko)

3.5 t_kurs

Oferta kursów nauki gry na instrumentach.

- Klucz główny: id_kursu
- Ograniczenia NOT NULL: nazwa, poziom, cena_za_lekcje
- CHECK: poziom IN ('poczatkujacy', 'sredniozaawansowany', 'zaawansowany')
- CHECK: cena_za_lekcje > 0
- Referencja: ref_instrument SCOPE IS t_instrument

3.6 t_lekcja

Zaplanowane i odbyte lekcje muzyki.

- Klucz główny: id_lekcji
- Ograniczenia NOT NULL: data_lekcji, godzina_start, czas_trwania, status
- CHECK: czas_trwania IN (30, 45, 60, 90)
- CHECK: status IN ('zaplanowana', 'odbyta', 'odwołana')
- CHECK: godzina_start >= '08:00' AND godzina_start <= '20:00'
- Referencje: ref_uczen, ref_nauczyciel, ref_kurs, ref_sala (wszystkie ze SCOPE IS)
- Indeksy: idx_lekcja_data, idx_lekcja_status

3.7 t_ocena

Oceny postępu uczniów w różnych obszarach.

- Klucz główny: id_oceny
- Ograniczenia NOT NULL: data_oceny, ocena, obszar
- CHECK: ocena BETWEEN 1 AND 6
- CHECK: obszar IN ('technika', 'teoria', 'słuch', 'interpretacja', 'improwizacja')
- Referencje: ref_uczen, ref_nauczyciel (ze SCOPE IS)
- Indeks: idx_ocena_data

3.8 Referencje (REF/DEREF)

W projekcie zastosowano 7 referencji do modelowania relacji między obiektami:

- Kurs wskazuje na instrument, którego dotyczy
- Lekcja wskazuje na ucznia, nauczyciela, kurs i salę
- Ocena wskazuje na ucznia i nauczyciela wystawiającego

Dzięki DEREF możliwe jest odwołanie się do atrybutów i metod obiektu wskazywanego:

```
SELECT DEREF(l.ref_uczen).imie || ' ' || DEREF(l.ref_uczen).nazwisko AS uczen,
       DEREF(l.ref_sala).nazwa AS sala,
       DEREF(l.ref_kurs).nazwa AS kurs
  FROM t_lekcja l WHERE l.status = 'zaplanowana';
```

3.9 Sekwencje

Utworzono 7 sekwencji do generowania identyfikatorów:

- seq_instrument – dla t_instrument
- seq_sala – dla t_sala

- seq_nauczyciel – dla t_nauczyciel
- seq_uczen – dla t_uczen
- seq_kurs – dla t_kurs
- seq_lekcja – dla t_lekcja
- seq_ocena – dla t_ocena

4 Pakiety PL/SQL

Logika biznesowa zaimplementowana w 3 pakietach z łącznie 10 podprogramami.

Pakiet	Podprogramy	Funkcjonalności
pkg_uczen	3	Zarządzanie uczniami
pkg_lekcja	4	Zarządzanie lekcjami
pkg_ocena	3	Zarządzanie ocenami
Razem:		10 podprogramów

Tabela 3: Pakiety PL/SQL

4.1 pkg_uczen

Pakiet do zarządzania uczniami szkoły muzycznej.

Procedury:

- dodaj(imie, nazwisko, data_urodzenia, email) – dodaje nowego ucznia z walidacją wieku (min. 5 lat)
- lista() – wyświetla listę wszystkich uczniów z wiekiem i statusem
- info(id_ucznia) – wyświetla szczegółowe informacje o uczniu

4.2 pkg_lekcja

Pakiet do zarządzania lekcjami i harmonogramem. Kluczowa procedura `zaplanuj` zawiera pełną walidację:

- Sprawdzenie kompetencji nauczyciela (czy zna instrument kursu)
- Limit nauczyciela: max 6 godzin lekcji dziennie
- Limit ucznia: max 2 lekcje dziennie
- Konflikt sali: brak nakładających się terminów
- Konflikt nauczyciela: brak nakładających się terminów
- Konflikt ucznia: brak nakładających się terminów

Procedury:

- `zaplanuj(id_ucznia, id_nauczyciela, id_kursu, id_sali, data, godzina, czas)` – planuje nową lekcję z pełną walidacją
- `oznacz_odbyta(id_lekcji)` – zmienia status lekcji na 'odbyta'
- `odwołaj(id_lekcji)` – zmienia status lekcji na 'odwołana'
- `plan_dnia(data)` – wyświetla wszystkie lekcje danego dnia

4.3 pkg_ocena

Pakiet do zarządzania ocenami postępu uczniów.

Procedury:

- `dodaj(id_ucznia, id_nauczyciela, ocena, obszar, komentarz)` – dodaje nową ocenę
- `historia_ucznia(id_ucznia)` – wyświetla wszystkie oceny ucznia
- `statystyki(id_ucznia)` – wyświetla średnie oceny wg obszarów

5 Wyzwalacze (Triggers)

Zdefiniowano 5 wyzwalaczy realizujących reguły biznesowe. Walidacja limitów i konfliktów jest zaimplementowana w pakiecie `pkg_lekcja.zaplanuj`, aby uniknąć błędu ORA-04091 (Mutating Table).

Trigger	Typ	Działanie
trg_uczen_wiek	BEFORE INSERT	Walidacja minimalnego wieku (5 lat)
trg_lekcja_dni_robocze	BEFORE I/U	Lekcje tylko w dni robocze (Pn–Pt)
trg_lekcja_godziny_dziecka	BEFORE I/U	Dzieci: lekcje tylko 14:00–19:00
trg_blokada_usun_nauczyciela	BEFORE DELETE	Ochrona nauczyciela z lekcjami
trg_blokada_usun_ucznia	BEFORE DELETE	Ochrona ucznia z lekcjami

Tabela 4: Wyzwalacze w projekcie

6 Obsługa błędów

W projekcie zastosowano mechanizmy obsługi wyjątków z własnymi kodami błędów.

6.1 Kody błędów aplikacji

Kod	Trigger/Procedura	Znaczenie
-20010	pkg_uczen.info	Nie znaleziono ucznia
-20020	pkg_lekcja	Nie znaleziono lekcji
-20030	pkg_lekcja.zaplanuj	Nauczyciel nie ma kompetencji do kursu
-20101	trg_uczen_wiek	Wiek ucznia poniżej 5 lat
-20102	trg_lekcja_dni_robocze	Lekcja zaplanowana w weekend
-20103	trg_lekcja_godziny_dziecka	Dziecko poza godzinami 14:00–19:00
-20104	pkg_lekcja.zaplanuj	Nauczyciel przekracza 6h dziennie
-20105	pkg_lekcja.zaplanuj	Uczeń ma już 2 lekcje w danym dniu
-20106	pkg_lekcja.zaplanuj	Konflikt rezerwacji sali
-20107	pkg_lekcja.zaplanuj	Nauczyciel ma inną lekcję w tym czasie
-20108	pkg_lekcja.zaplanuj	Uczeń ma inną lekcję w tym czasie
-20201	trg_blokada_usun_nauczyciela	Próba usunięcia nauczyciela z lekcjami
-20202	trg_blokada_usun_ucznia	Próba usunięcia ucznia z lekcjami

Tabela 5: Kody błędów aplikacji

6.2 Scenariusze testowe

Projekt zawiera kompleksowe testy w pliku 06_testy.sql obejmujące 11 scenariuszy:

6.2.1 Scenariusz 1: Dane podstawowe

Weryfikacja poprawnego załadowania danych (instrumenty, sale, nauczyciele, uczniowie, kursy).

6.2.2 Scenariusz 2: Walidacja wieku ucznia

Test triggera trg_uczen_wiek:

- 3-latek: odrzucony (błąd -20101)
- 5-latek: akceptowany

6.2.3 Scenariusz 3: Dni robocze

Test triggera trg_lekcja_dni_robocze:

- Sobota: odrzucona (błąd -20102)
- Poniedziałek: akceptowany

6.2.4 Scenariusz 4: Godziny dla dzieci

Test triggera trg_lekcja_godziny_dziecka:

- Dziecko o 08:00: odrzucone (błąd -20103)
- Dziecko o 15:00: akceptowane
- Dorosły o 08:00: akceptowany

6.2.5 Scenariusz 5: Limit nauczyciela

Test procedury pkg_lekcja.zaplanuj:

- 6 lekcji po 60 minut (360 min): akceptowane
- 7. lekcja (30 min): odrzucona (błąd -20104)

6.2.6 Scenariusz 6: Limit ucznia

Test procedury pkg_lekcja.zaplanuj:

- 1. lekcja: akceptowana
- 2. lekcja: akceptowana
- 3. lekcja: odrzucona (błąd -20105)

6.2.7 Scenariusz 7: Konflikty czasowe

Test procedury pkg_lekcja.zaplanuj:

- Ta sama sala, nakładający się czas: odrzucone (błąd -20106)
- Ten sam nauczyciel, nakładający się czas: odrzucone (błąd -20107)
- Ten sam uczeń, nakładający się czas: odrzucone (błąd -20108)

6.2.8 Scenariusz 8: Kompetencje nauczyciela

Test procedury pkg_lekcja.zaplanuj:

- Nauczyciel z kompetencjami: akceptowany
- Nauczyciel bez kompetencji: odrzucony (błąd -20030)

6.2.9 Scenariusz 9: Blokada usuwania

Test triggerów ochrony danych:

- Usunięcie nauczyciela z lekcjami: odrzucone (błąd -20201)
- Usunięcie ucznia z lekcjami: odrzucone (błąd -20202)

6.2.10 Scenariusz 10: Pakiety CRUD

Test procedur pakietów:

- pkg_uczen.lista() – wyświetla listę uczniów
- pkg_uczen.info(1) – szczegóły ucznia
- pkg_lekcja.plan_dnia(data) – plan dnia
- pkg_ocena.historia_ucznia(1) – historia ocen

6.2.11 Scenariusz 11: Metody obiektów

Test metod typów obiektowych:

- t_uczen_obj.wiek() – oblicza wiek
- t_uczen_obj.czy_dziecko() – zwraca 'T' lub 'N'
- t_nauczyciel_obj.lata_stazu() – oblicza staż
- t_nauczyciel_obj.liczba_instrumentow() – liczba instrumentów
- t_sala_obj.opis_pelny() – opis sali z wyposażeniem

7 Role użytkowników

W projekcie zdefiniowano 3 role z różnymi poziomami uprawnień.

7.1 Administrator (rola_admin)

Pełny dostęp do systemu:

- SELECT, INSERT, UPDATE, DELETE na wszystkich tabelach
- Dostęp do wszystkich sekwencji
- Wykonywanie wszystkich pakietów
- Zarządzanie użytkownikami

7.2 Nauczyciel (rola_nauczyciel)

Prowadzenie lekcji i ocenianie:

- SELECT na wszystkich tabelach
- UPDATE na tabeli t_lekcja (zmiana statusu)
- INSERT do tabeli t_ocena
- Wykonywanie pakietów: pkg_uczen, pkg_lekcja, pkg_ocena

7.3 Sekretariat (rola_sekretariat)

Zarządzanie harmonogramem i uczniami:

- SELECT na wszystkich tabelach
- INSERT, UPDATE na tabeli t_uczen
- INSERT, UPDATE na tabeli t_lekcja
- Wykonywanie pakietów: pkg_uczen, pkg_lekcja (bez pkg_ocena)

7.4 Macierz uprawnień

Tabela	Admin	Nauczyciel	Sekretariat
t_instrument	SIUD	S	S
t_sala	SIUD	S	S
t_nauczyciel	SIUD	S	S
t_uczen	SIUD	S	SIU
t_kurs	SIUD	S	S
t_lekcja	SIUD	SU	SIU
t_ocena	SIUD	SI	S

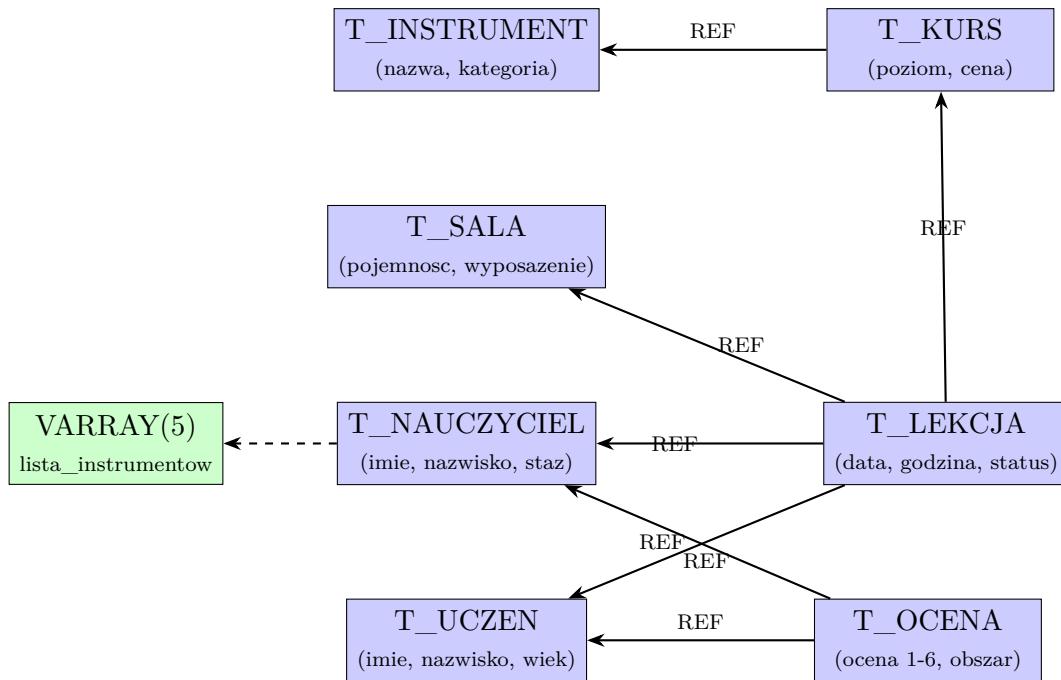
Tabela 6: Macierz uprawnień (S=SELECT, I=INSERT, U=UPDATE, D=DELETE)

7.5 Użytkownicy testowi

Użytkownik	Rola	Hasło
usr_admin	rola_admin	Admin123!
usr_nauczyciel	rola_nauczyciel	Naucz123!
usr_sekretariat	rola_sekretariat	Sekr123!

Tabela 7: Użytkownicy testowi

8 Diagram relacji obiektów



Legenda:

- Strzałka ciągła – referencja REF (wskaźnik do obiektu)
- Strzałka przerywana – zawiera VARRAY (kolekcja)
- Niebieskie prostokąty – typy obiektowe / tabele
- Zielony prostokąt – typ kolekcji VARRAY

9 Struktura plików projektu

Plik	Zawartość
01_typy.sql	Definicje 8 typów obiektowych z 14 metodami
02_tabele.sql	7 tabel obiektowych, 7 sekwencji, 5 indeksów
03_pakiety.sql	3 pakiety PL/SQL z 10 podprogramami
04_triggery.sql	5 wyzwalaczy walidacyjnych
05_dane.sql	Dane testowe (10 instr., 5 sal, 5 naucz., 10 uczn., 10 kursów)
06_testy.sql	11 scenariuszy testowych
07_uzytkownicy.sql	3 role, 3 użytkowników

Tabela 8: Pliki projektu

Kolejność uruchamiania:

1. 01_typy.sql – typy obiektowe
2. 02_tabele.sql – tabele i sekwencje
3. 03_pakiety.sql – pakiety PL/SQL
4. 04_triggery.sql – wyzwalacze
5. 05_dane.sql – dane testowe
6. 06_testy.sql – testy (opcjonalne)
7. 07_uzytkownicy.sql – role i użytkownicy (wymaga uprawnień DBA)