

**Politechnika Łódzka**

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki

# **Szkoła Muzyczna**

Obiektowa Baza Danych Oracle

Wersja 3.0 – Uproszczona

Rozproszone i Obiektowe Bazy Danych

**Autorzy:** Igor Typiński (251237)  
Mateusz Mróz (251190)

**Grupa:** 5

**Temat:** Szkoła muzyczna (z naciskiem na rozwój ucznia)

Łódź, styczeń 2026

# Spis treści

# 1 Opis projektu

## 1.1 Cel i zakres

Projekt przedstawia obiektową bazę danych dla szkoły muzycznej, ze szczególnym uwzględnieniem śledzenia rozwoju uczniów. System umożliwia:

- Zarządzanie danymi uczniów i nauczycieli
- Planowanie i prowadzenie lekcji indywidualnych
- Ocenianie postępów uczniów w różnych obszarach muzycznych
- Przypisywanie nauczycieli do instrumentów (VARRAY)
- Automatyczną walidację reguł biznesowych (triggery)
- Generowanie raportów i statystyk

## 1.2 Technologia

- Oracle Database 19c+
- Podejście obiektowo-relacyjne
- Język PL/SQL
- Typy obiektowe z metodami
- Referencje REF/DEREF

## 1.3 Założenia uproszczone (v3)

W wersji 3.0 przyjęto następujące uproszczenia względem wcześniejszych wersji:

- **Jeden semestr** – brak tabeli t\_semestr (eliminacja problemu mutating table)
- **Szkoła czynna Pn–Pt** – uproszczenie logiki godzin dla dzieci
- **Brak audytu** – zredukowano logowanie do minimum
- **Brak porównywania uczniów** – funkcja porownaj\_uczniow usunięta
- **Założenie kompetencji** – dyrektor weryfikuje instrumenty nauczyciela przy zatrudnieniu

# 2 Realizacja założeń projektowych

## 2.1 Typy obiektowe

W projekcie zdefiniowano 8 typów obiektowych z łącznie 13 metodami:

## 2.2 Tabele obiektowe

Utworzono 7 tabel obiektowych przechowujących dane:

## 2.3 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

Przykład użycia Deref w triggerze:

Typ	Metody	Opis
t_instrument_obj	1	Instrument muzyczny (info)
t_lista_instrumentow	–	VARARRAY(5) nazw instrumentów
t_sala_obj	1	Sala lekcyjna (opis)
t_nauczyciel_obj	3	Nauczyciel (staz, liczba_instr, info)
t_uczen_obj	3	Uczeń (wiek, czy_dziecko, info)
t_kurs_obj	2	Kurs nauki (cena_brutto, info)
t_lekcja_obj	2	Pojedyncza lekcja (godzina_konca, info)
t_ocena_obj	1	Ocena postępu (info)
<b>Razem:</b>		<b>8 typów, 13 metod</b>

Tabela 1: Typy obiektowe w projekcie

Tabela	Typ bazowy	Referencje (REF)
t_instrument	t_instrument_obj	–
t_sala	t_sala_obj	–
t_nauczyciel	t_nauczyciel_obj	– (zawiera VARARRAY)
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_postepu	t_ocena_obj	ref_uczen, ref_nauczyciel

Tabela 2: Tabele obiektowe i ich referencje

```

SELECT Deref(:NEW.ref_uczen) INTO v_uczen FROM DUAL;
IF v_uczen.czy_dziecko() = 'T' THEN
    -- walidacja godzin dla dziecka
END IF;

```

## 2.4 Kolekcja VARARRAY

Typ t\_lista\_instrumentow jako VARARRAY(5) przechowuje listę instrumentów, których może uczyć nauczyciel:

```

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

-- Przykład użycia:
INSERT INTO t_nauczyciel VALUES (
    t_nauczyciel_obj(1, 'Jan', 'Kowalski', 'j.kowalski@szkola.pl',
        DATE '2020-01-01',
        t_lista_instrumentow('Fortepian', 'Skrzypce'))
);

```

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

## 2.5 Pakiety PL/SQL

Logika biznesowa zaimplementowana w 3 pakietach z łącznie 15 podprogramami:

Pakiet	Podpr.	Główne funkcjonalności
pkg_uczen	6	dodaj, lista, lista_dzieci, info, srednia_ocen, liczba_lekcji
pkg_lekcja	6	zaplanuj, oznacz_odbyta, odwołaj, plan_dnia, plan_nauczyciela, raport_obciazenia
pkg_ocena	3	dodaj, historia_ucznia, raport_postepu
<b>Razem:</b>		<b>15 podprogramów</b>

Tabela 3: Pakiety PL/SQL

## 2.6 Kursory

W projekcie wykorzystano trzy rodzaje kursorów:

- **Niejawne (FOR)** – uproszczona składnia FOR rec IN (SELECT...)
- **Parametryzowane** – kursory z parametrami wejściowymi
- **VALUE/TREAT** – pobranie obiektu z tabeli obiektowej

Przykład:

```
FOR r IN (
    SELECT u.id_ucznia, u.imie, u.nazwisko,
           TREAT(VALUE(u) AS t_uczen_obj).wiek() AS wiek
    FROM t_uczen u
    WHERE TREAT(VALUE(u) AS t_uczen_obj).czy_dziecko() = 'T'
) LOOP
    DBMS_OUTPUT.PUT_LINE(r.imie || ' - lat ' || r.wiek);
END LOOP;
```

## 2.7 Wyzwalacze (Triggery)

Zdefiniowano 10 wyzwalaczy realizujących reguły biznesowe:

Trigger	Typ	Działanie
trg_uczen_wiek	BEFORE I/U	Walidacja min. 5 lat
trg_lekcja_dni_robocze	BEFORE I/U	Tylko Pn-Pt
trg_lekcja_godziny_dziecka	BEFORE I/U	Dzieci: 14:00–19:00
trg_lekcja_limit_nauczyciela	BEFORE I/U	Max 6h/dzień
trg_lekcja_limit_ucznia	BEFORE I/U	Max 2 lekcje/dzień
trg_lekcja_konflikt_sali	BEFORE I/U	Brak nakładania sal
trg_lekcja_konflikt_nauczyciela	BEFORE I/U	Nauczyciel w 1 miejscu
trg_lekcja_konflikt_ucznia	BEFORE I/U	Uczeń w 1 miejscu
trg_blokada_usun_nauczyciela	BEFORE D	Ochrona z zaplanowanymi lekcjami
trg_blokada_usun_ucznia	BEFORE D	Ochrona z zaplanowanymi lekcjami

Tabela 4: Wyzwalacze w projekcie

## 2.8 Obsługa błędów

Zastosowano mechanizmy obsługi wyjątków z własnymi kodami błędów:

Kod	Znaczenie
-20101	Wiek ucznia poniżej 5 lat
-20102	Lekcja w weekend
-20103	Dziecko poza godzinami 14:00–19:00
-20104	Nauczyciel przekracza 6h/dzień
-20105	Uczeń ma już 2 lekcje w danym dniu
-20106	Konflikt sali
-20107	Konflikt nauczyciela
-20108	Konflikt ucznia
-20109	Próba usunięcia nauczyciela z lekcjami
-20110	Próba usunięcia ucznia z lekcjami

Tabela 5: Kody błędów aplikacji

## 3 Przyjęte ograniczenia biznesowe

1. **Wiek ucznia:** minimum 5 lat
2. **Dzieci** (poniżej 15 lat): lekcje tylko 14:00–19:00
3. **Dni pracy szkoły:** poniedziałek–piątek
4. **Godziny pracy:** 08:00–20:00
5. **Nauczyciel:** max 5 instrumentów (VARRAY), max 6h lekcji/dzień
6. **Uczeń:** max 2 lekcje dziennie
7. **Czas lekcji:** 30, 45, 60 lub 90 minut
8. **Oceny:** skala 1–6 (polska)
9. **Obszary oceny:** technika, teoria, słuch, rytm, interpretacja, ogólna
10. **Poziomy kursów:** początkujący, średni, zaawansowany
11. **Statusy lekcji:** zaplanowana, odbyta, odwołana
12. **Kategorie instrumentów:** dęte, strunowe, perkusyjne, klawiszowe

## 4 Role użytkowników

### 4.1 Administrator (rola\_admin)

Pełny dostęp do systemu:

- Zarządzanie wszystkimi danymi (CRUD na wszystkich tabelach)
- Dostęp do wszystkich sekwencji
- Wykonywanie wszystkich pakietów

### 4.2 Nauczyciel (rola\_nauczyciel)

Prowadzenie lekcji i ocenianie:

- Odczyt wszystkich tabel
- Aktualizacja statusu lekcji

- Dodawanie ocen postępu
- Wykonywanie wszystkich pakietów

### 4.3 Sekretariat (rola\_sekretariat)

Zarządzanie harmonogramem:

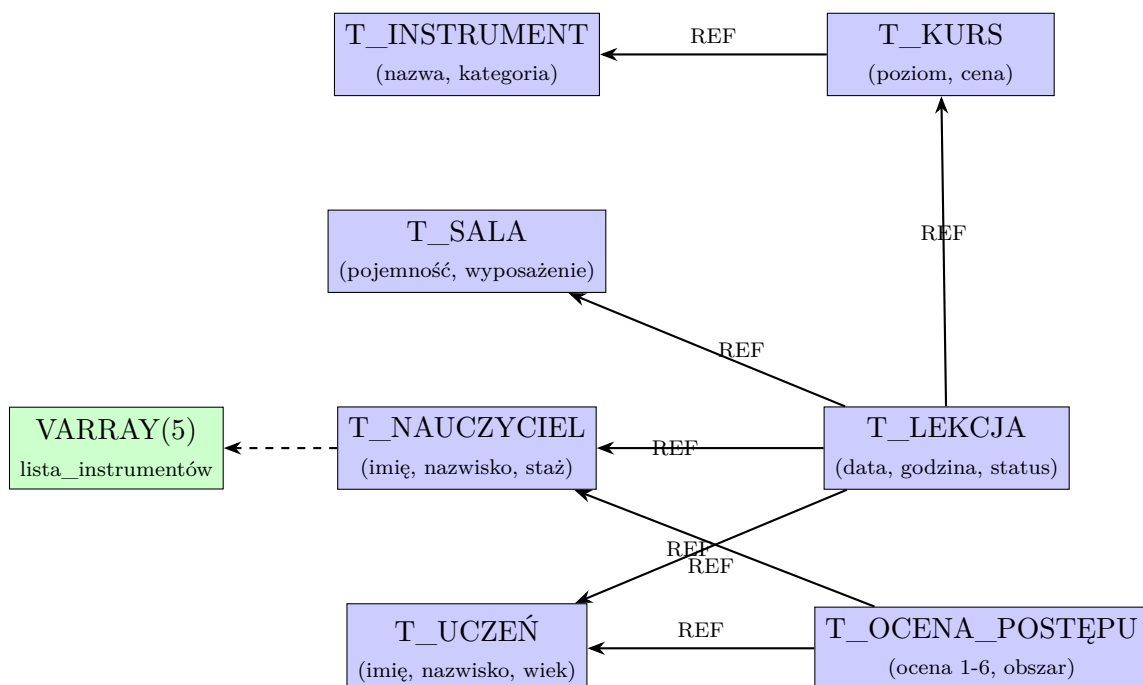
- Pełne zarządzanie danymi uczniów (INSERT, UPDATE)
- Planowanie i modyfikowanie lekcji
- Odczyt wszystkich tabel
- Pakiety: pkg\_uczen, pkg\_lekcja (bez pkg\_ocena)

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_postepu	SIUD	SI	S

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

(\*) Nauczyciel może aktualizować tylko kolumnę status w tabeli t\_lekcja.

## 5 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

## 6 Scenariusze testowe

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

1. **Dane podstawowe** – weryfikacja poprawnego załadowania instrumentów, sal, nauczycieli, uczniów, kursów
2. **Walidacja wieku** – test triggera `trg_uczen_wiek`:
  - 3-latek: odrzucony (-20101)
  - 4-latek: odrzucony
  - 5-latek: akceptowany
3. **Dni robocze** – test triggera `trg_lekcja_dni_robocze`:
  - Sobota: odrzucona (-20102)
  - Niedziela: odrzucona
  - Poniedziałek: akceptowany
4. **Godziny dla dzieci** – test triggera `trg_lekcja_godziny_dziecka`:
  - Dziecko o 08:00: odrzucone (-20103)
  - Dziecko o 13:00: odrzucone
  - Dziecko o 14:00: akceptowane
  - Dziecko 18:15–19:00: akceptowane
  - Dziecko 18:30–19:15: odrzucone (przekracza 19:00)
  - Dorosły o 08:00: akceptowany
5. **Limit nauczyciela** – test triggera `trg_lekcja_limit_nauczyciela`:
  - 6 lekcji po 60 min (360 min): akceptowane
  - 7. lekcja: odrzucona (-20104)
6. **Limit ucznia** – test triggera `trg_lekcja_limit_ucznia`:
  - 2 lekcje: akceptowane
  - 3. lekcja: odrzucona (-20105)
7. **Konflikty** – testy triggerów konfliktów:
  - Ta sama sala: odrzucone (-20106)
  - Ten sam nauczyciel: odrzucone (-20107)
  - Ten sam uczeń: odrzucone (-20108)
8. **Blokada usuwania** – ochrona danych z zaplanowanymi lekcjami
9. **Pakiety** – test procedur CRUD
10. **Metody obiektów** – test funkcji: `wiek()`, `czy_dziecko()`, `staz()`, `opis()`



Plik	Zawartość
01_typy.sql	Definicje 8 typów obiektowych z metodami
02_tabele.sql	7 tabel, 7 sekwencji, 5 indeksów
03_pakiety.sql	3 pakiety PL/SQL (15 podprogramów)
04_triggery.sql	10 wyzwalaczy walidacyjnych
05_dane.sql	Dane testowe (10 instrum., 5 sal, 5 naucz., 10 uczn., 10 kursów)
06_testy.sql	10 scenariuszy testowych
07_uzytkownicy.sql	3 role, 3 użytkownicy, synonimy

Tabela 7: Pliki projektu

## 7 Struktura plików projektu

Kolejność uruchamiania:

1. 01\_typy.sql
2. 02\_tabele.sql
3. 03\_pakiety.sql
4. 04\_triggery.sql
5. 05\_dane.sql
6. 06\_testy.sql (opcjonalne)
7. 07\_uzytkownicy.sql (wymaga uprawnień DBA)

## 8 Podsumowanie

Projekt obiektowej bazy danych dla szkoły muzycznej w wersji 3.0 spełnia wszystkie wymagania:

Wymaganie	Realizacja
Definicje typów obiektowych z metodami	8 typów, 13 metod
Tabele obiektowe (wierszowe)	7 tabel
Referencja i dereferencja	7 REF, DEREf
Wstawianie danych z referencją	Tak
VARRAY do relacji 1:N	t_lista_instrumentow(5)
Pakiety PL/SQL	3 pakiety, 15 podprogramów
Kursory i pętle FOR	Tak
Obsługa błędów	EXCEPTION, RAISE_APPLICATION_ERROR
Wyzwalacze	10 triggerów
Role użytkowników	3 role, 3 użytkownicy
Testy	10 scenariuszy

Tabela 8: Realizacja wymagań projektowych

### 8.1 Zmiany względem wersji 2.0

- Usunięto t\_semestr – eliminacja problemu mutating table
- Usunięto t\_zapis – redundancja z t\_lekcja
- Dodano t\_sala – lepsza walidacja konfliktów

- Zredukowano audyt – prostota
- Usunięto porównaj\_uczniow – zbędna komplikacja
- Dodano scenariusze testowe – kompletna weryfikacja
- Szkoła tylko Pn–Pt – uproszczenie logiki dzieci