

Politechnika Łódzka

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki

Szkoła Muzyczna

Obiektowa Baza Danych Oracle

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	3
1.1	Cel i zakres	3
1.2	Technologia	3
2	Typy obiektowe	3
2.1	Szczegółowy opis typów	3
2.1.1	t_uczen_obj	3
2.1.2	t_nauczyciel_obj	4
2.1.3	t_kurs_obj	4
2.1.4	t_zapis_obj	4
2.1.5	t_lekcja_obj	5
2.1.6	t_ocena_obj	5
2.1.7	t_sala_obj	5
2.1.8	t_semestr_obj	6
2.1.9	t_lista_instrumentow	6
3	Tabele obiektowe	6
3.1	Szczegółowy opis tabel	6
3.1.1	t_uczen	6
3.1.2	t_nauczyciel	7
3.1.3	t_kurs	7
3.1.4	t_zapis	7
3.1.5	t_lekcja	7
3.1.6	t_ocena_postepu	7
3.1.7	t_sala	7
3.1.8	t_semestr	8
3.2	Referencje (REF/DEREF)	8
3.3	Sekwencje	8
4	Pakiety PL/SQL	8
4.1	pkg_semestr	8
4.2	pkg_sala	9
4.3	pkg_uczen	9
4.4	pkg_lekcja	9
4.5	pkg_ocena	10
4.6	Kursory	10
5	Wyzwalacze (Triggery)	10
5.1	Triggery walidacyjne	10
5.2	Triggery kontroli semestrów	10
5.3	Triggery blokujące usuwanie	10
5.4	Triggery audytowe	10
5.5	Szczegółowy opis wybranych triggerów	10
5.5.1	trg_lekcja_godziny_dziecka	10
5.5.2	trg_lekcja_limit_nauczyciela	11
5.5.3	trg_semestr_tylko_jeden_aktywny	11
5.6	Kody błędów	11

6	Przyjęte ograniczenia	11
6.1	Ograniczenia dotyczące uczniów	11
6.2	Ograniczenia dotyczące nauczycieli	12
6.3	Ograniczenia dotyczące lekcji	12
6.4	Ograniczenia dotyczące ocen	12
6.5	Ograniczenia dotyczące kursów	12
6.6	Ograniczenia dotyczące sal	13
6.7	Ograniczenia dotyczące semestrów	13
6.8	Ograniczenia dotyczące zapisów	13
7	Role użytkowników	13
7.1	Administrator (ROLA_ADMIN_SZKOLY)	13
7.2	Nauczyciel (ROLA_NAUCZYCIEL)	14
7.3	Sekretariat (ROLA_SEKRETARIAT)	14
7.4	Użytkownicy testowi	14
8	Testy	14
8.1	Przykładowe testy	14
9	Diagram relacji obiektów	15
10	Podsumowanie	15
10.1	Pliki projektu	16

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. System umożliwia:

- Zarządzanie danymi uczniów i nauczycieli
- Planowanie i prowadzenie lekcji w ramach semestrów
- Rezerwację sal lekcyjnych z odpowiednim wyposażeniem
- Ocenianie postępów uczniów w różnych obszarach muzycznych
- Przypisywanie nauczycieli do instrumentów
- Kontrolę limitów obciążenia nauczycieli i uczniów
- Uwzględnienie grafiku dzieci uczęszczających do zwykłej szkoły
- Generowanie raportów i statystyk

1.2 Technologia

- Oracle Database
- Podejście obiektowo-relacyjne
- Język PL/SQL
- Typy obiektowe z metodami
- Referencje REF/DEREF
- Kolekcje VARRAY

2 Typy obiektowe

W projekcie zdefiniowano 9 typów obiektowych z łącznie 15 metodami.

Typ	Metody	Opis
t_uczen_obj	3	Uczeń szkoły muzycznej
t_nauczyciel_obj	3	Nauczyciel z listą instrumentów
t_kurs_obj	2	Kurs nauki gry na instrumencie
t_zapis_obj	2	Zapis ucznia na kurs
t_lekcja_obj	2	Pojedyncza lekcja muzyki
t_ocena_obj	1	Ocena postępu ucznia
t_sala_obj	1	Sala lekcyjna z wyposażeniem
t_semestr_obj	2	Semestr akademicki
t_lista_instrumentow	–	VARRAY(5) nazw instrumentów

Tabela 1: Typy obiektowe w projekcie

2.1 Szczegółowy opis typów

2.1.1 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 (unikalny)
- telefon (VARCHAR2(15)) – numer telefonu
- data_rejestracji (DATE) – data zapisu do szkoły

Metody:

- wiek() RETURN NUMBER – oblicza aktualny wiek ucznia
- czy_pelnoletni() RETURN VARCHAR2 – zwraca 'TAK' lub 'NIE'
- pelne_dane() RETURN VARCHAR2 – zwraca imię i nazwisko

2.1.2 t_nauczyciel_obj

Reprezentuje nauczyciela szkoły muzycznej.

Atrybuty:

- id_nauczyciela (NUMBER) – unikalny identyfikator
- imie (VARCHAR2(50)) – imię nauczyciela
- nazwisko (VARCHAR2(50)) – nazwisko nauczyciela
- data_urodzenia (DATE) – data urodzenia
- email (VARCHAR2(100)) – adres email (unikalny)
- telefon (VARCHAR2(15)) – numer telefonu
- instrumenty (t_lista_instrumentow) – VARRAY instrumentów
- wykształcenie (VARCHAR2(50)) – poziom wykształcenia
- lata_doswiadczenia (NUMBER) – staż pracy

Metody:

- pelne_dane() RETURN VARCHAR2 – zwraca imię i nazwisko
- liczba_instrumentow() RETURN NUMBER – liczba instrumentów w VARRAY
- czy_senior() RETURN VARCHAR2 – 'TAK' jeśli staż ≥ 10 lat

2.1.3 t_kurs_obj

Reprezentuje kurs nauki gry na instrumencie.

Atrybuty:

- id_kursu (NUMBER) – unikalny identyfikator
- nazwa (VARCHAR2(100)) – nazwa kursu
- typ (VARCHAR2(20)) – 'indywidualny' lub 'grupowy'
- opis (VARCHAR2(500)) – opis kursu
- cena_za_lekcje (NUMBER(10,2)) – cena jednej lekcji
- domyslne_czas (NUMBER) – domyślny czas trwania w minutach

Metody:

- czy_indywidualny() RETURN VARCHAR2 – 'TAK' lub 'NIE'
- info() RETURN VARCHAR2 – zwraca nazwę z typem i ceną

2.1.4 t_zapis_obj

Reprezentuje zapis ucznia na kurs.

Atrybuty:

- id_zapisu (NUMBER) – unikalny identyfikator

- data_zapisu (DATE) – data dokonania zapisu
- status (VARCHAR2(20)) – 'aktywny', 'zawieszony', 'zakończony'
- ref_uczen (REF t_uczen_obj) – referencja do ucznia
- ref_kurs (REF t_kurs_obj) – referencja do kursu
- ref_nauczyciel (REF t_nauczyciel_obj) – referencja do nauczyciela

Metody:

- czy_aktywny() RETURN VARCHAR2 – 'TAK' lub 'NIE'
- dni_od_zapisu() RETURN NUMBER – liczba dni od zapisu

2.1.5 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)
- temat (VARCHAR2(200)) – temat lekcji
- uwagi (VARCHAR2(500)) – uwagi nauczyciela
- 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:

- czy_odbyta() RETURN VARCHAR2 – 'TAK' lub 'NIE'
- krotki_opis() RETURN VARCHAR2 – data, godzina, status

2.1.6 t_ocena_obj

Reprezentuje ocenę postępu ucznia.

Atrybuty:

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

Metody:

- poziom_slowny() RETURN VARCHAR2 – 'niedostateczny' do 'celujący'

2.1.7 t_sala_obj

Reprezentuje salę lekcyjną z wyposażeniem.

Atrybuty:

- id_sali (NUMBER) – unikalny identyfikator
- nazwa (VARCHAR2(50)) – nazwa sali (unikalna)
- pojemnosc (NUMBER) – maksymalna liczba osób

- ma_fortepian (CHAR(1)) – 'T' lub 'N'
- ma_perkusje (CHAR(1)) – 'T' lub 'N'
- opis (VARCHAR2(200)) – dodatkowy opis wyposażenia

Metody:

- opis_pelny() RETURN VARCHAR2 – nazwa z informacją o wyposażeniu

2.1.8 t_semestr_obj

Reprezentuje semestr akademicki.

Atrybuty:

- id_semestru (NUMBER) – unikalny identyfikator
- nazwa (VARCHAR2(50)) – nazwa semestru (unikalna)
- data_od (DATE) – data rozpoczęcia
- data_do (DATE) – data zakończenia
- czy_aktywny (CHAR(1)) – 'T' lub 'N'

Metody:

- czy_w_trakcie() RETURN VARCHAR2 – 'TAK' jeśli SYSDATE w zakresie
- dni_do_konca() RETURN NUMBER – liczba dni do końca semestru

2.1.9 t_lista_instrumentow

Kolekcja VARRAY przechowująca nazwy instrumentów.

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

Ograniczenie do 5 elementów wynika z założenia, że nauczyciel specjalizuje się w kilku instrumentach.

3 Tabele obiektowe

Utworzono 8 tabel obiektowych przechowujących dane.

Tabela	Typ bazowy	Referencje (REF)
t_uczen	t_uczen_obj	–
t_nauczyciel	t_nauczyciel_obj	– (zawiera VARRAY)
t_kurs	t_kurs_obj	–
t_zapis	t_zapis_obj	ref_uczen, ref_kurs, ref_nauczyciel
t_lekcja	t_lekcja_obj	ref_uczen, ref_nauczyciel, ref_kurs, ref_sala
t_ocena_postepu	t_ocena_obj	ref_uczen, ref_nauczyciel
t_sala	t_sala_obj	–
t_semestr	t_semestr_obj	–

Tabela 2: Tabele obiektowe i ich referencje

3.1 Szczegółowy opis tabel

3.1.1 t_uczen

Przechowuje dane uczniów szkoły muzycznej.

- Klucz główny: id_ucznia
- Ograniczenie UNIQUE: email
- Indeks: idx_uczen_nazwisko (nazwisko)

3.1.2 t_nauczyciel

Przechowuje dane nauczycieli wraz z listą instrumentów (VARRAY).

- Klucz główny: id_nauczyciela
- Ograniczenie UNIQUE: email
- Zawiera kolekcję t_lista_instrumentow

3.1.3 t_kurs

Przechowuje ofertę kursów szkoły.

- Klucz główny: id_kursu
- CHECK: typ IN ('indywidualny', 'grupowy')
- CHECK: cena_za_lekcje > 0
- CHECK: domyslne_czas IN (30, 45, 60, 90)

3.1.4 t_zapis

Przechowuje zapisy uczniów na kursy.

- Klucz główny: id_zapisu
- CHECK: status IN ('aktywny', 'zawieszony', 'zakończony')
- Referencje do: t_uczen, t_kurs, t_nauczyciel

3.1.5 t_lekcja

Przechowuje zaplanowane i odbyte lekcje.

- Klucz główny: id_lekcji
- CHECK: status IN ('zaplanowana', 'odbyta', 'odwołana')
- CHECK: czas_trwania IN (30, 45, 60, 90)
- Referencje do: t_uczen, t_nauczyciel, t_kurs, t_sala
- Indeks: idx_lekcja_data (data_lekcji)

3.1.6 t_ocena_postepu

Przechowuje oceny postępów uczniów.

- Klucz główny: id_oceny
- CHECK: ocena BETWEEN 1 AND 6
- CHECK: obszar IN ('technika', 'teoria', 'sluch', 'rytm', 'interpretacja', 'ogolna')
- Referencje do: t_uczen, t_nauczyciel

3.1.7 t_sala

Przechowuje informacje o salach lekcyjnych.

- Klucz główny: id_sali
- Ograniczenie UNIQUE: nazwa
- CHECK: pojemnosc > 0
- CHECK: ma_fortepian IN ('T', 'N')

- CHECK: ma_perkusje IN ('T', 'N')

3.1.8 t__semestr

Przechowuje informacje o semestrach.

- Klucz główny: id_semestru
- Ograniczenie UNIQUE: nazwa
- CHECK: czy_aktywny IN ('T', 'N')
- CHECK: data_do > data_od

3.2 Referencje (REF/DEREF)

W projekcie zastosowano 8 referencji do modelowania relacji między obiektami:

- Zapis wskazuje na ucznia, kurs i nauczyciela prowadzącego
- 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(1.ref_uczen).pelne_dane() AS uczen,
       Deref(1.ref_sala).nazwa AS sala
FROM t_lekcja 1;
```

3.3 Sekwencje

Utworzono 8 sekwencji do generowania identyfikatorów:

- seq_uczen, seq_nauczyciel, seq_kurs, seq_zapis
- seq_lekcja, seq_ocena, seq_sala, seq_semestr

4 Pakiety PL/SQL

Logika biznesowa zaimplementowana w 5 pakietach z łącznie 26 procedurami i funkcjami.

Pakiet	Procedur	Główne funkcjonalności
pkg_semestr	4	Zarządzanie semestrami
pkg_sala	4	Zarządzanie salami lekcyjnymi
pkg_uczen	5	Zarządzanie uczniami
pkg_lekcja	8	Planowanie i zarządzanie lek- cjami
pkg_ocena	5	Ocenianie postępów uczniów

Tabela 3: Pakiety PL/SQL

4.1 pkg_semestr

Pakiet do zarządzania semestrami akademickimi.

Procedury i funkcje:

- `utworz_semestr(p_nazwa, p_data_od, p_data_do, p_aktywny)` – tworzy nowy semestr
- `aktywuj_semestr(p_id_semestru)` – aktywuje wybrany semestr (dezaktywuje pozostałe)
- `pobierz_aktywny_semestr()` RETURN `t_semestr_obj` – zwraca aktywny semestr
- `info_semestr()` – wyświetla informacje o aktywnym semestrze

4.2 pkg_sala

Pakiet do zarządzania salami lekcyjnymi.

Procedury i funkcje:

- `dodaj_sale(p_nazwa, p_pojemnosc, p_ma_fortepian, p_ma_perkusje, p_opis)` – dodaje salę
- `sprawdz_dostepnosc(p_id_sali, p_data, p_godzina)` RETURN VARCHAR2 – 'WOLNA' lub 'ZAJETA'
- `lista_sal()` – wyświetla listę wszystkich sal
- `sale_wolne(p_data, p_godzina)` RETURN SYS_REFCURSOR – zwraca wolne sale

4.3 pkg_uczen

Pakiet do zarządzania uczniami.

Stałe:

- `c_min_wiek` CONSTANT NUMBER := 5 – minimalny wiek ucznia

Procedury i funkcje:

- `dodaj_ucznia(p_imie, p_nazwisko, p_data_urodzenia, p_email, p_telefon)` – dodaje ucznia
- `liczba_uczniow()` RETURN NUMBER – zwraca liczbę uczniów
- `uczniowie_wiek(p_wiek_min, p_wiek_max)` RETURN SYS_REFCURSOR – filtruje wg wieku
- `lista_uczniow()` – wyświetla listę uczniów (kursor jawny)
- `srednia_ocen(p_id_ucznia)` RETURN NUMBER – oblicza średnią ocen

4.4 pkg_lekcja

Pakiet do planowania i zarządzania lekcjami.

Procedury i funkcje:

- `zaplanuj_lekcje(p_id_ucznia, p_id_nauczyciela, p_id_kursu, p_id_sali, p_data, p_godzina, p_czas_trwania)` – planuje lekcję
- `oznacz_odbyta(p_id_lekcji, p_temat, p_uwagi)` – oznacza lekcję jako odbytą
- `odwolaj_lekcje(p_id_lekcji)` – odwołuje lekcję
- `lekcje_ucznia(p_id_ucznia, p_miesiac)` RETURN SYS_REFCURSOR – lekcje ucznia w miesiącu
- `lekcje_tygodniowo(p_id_nauczyciela)` RETURN NUMBER – liczba lekcji w tygodniu
- `raport_dzienny(p_data)` – wyświetla lekcje na dany dzień
- `sprawdz_dostepnosc_kompleksowa(p_id_ucznia, p_id_nauczyciela, p_id_sali, p_data, p_godzina)` RETURN VARCHAR2 – sprawdza konflikty

- `statystyki_nauczyciela(p_id_nauczyciela, p_data)` – wyświetla obciążenie nauczyciela

4.5 pkg_ocena

Pakiet do oceniania postępów uczniów.

Procedury i funkcje:

- `dodaj_ocene(p_id_ucznia, p_id_nauczyciela, p_ocena, p_obszar, p_komentarz)` – dodaje ocenę
- `ostatnie_oceny(p_id_ucznia, p_limit) RETURN SYS_REFCURSOR` – ostatnie N ocen
- `raport_postepu(p_id_ucznia)` – wyświetla raport postępu ucznia
- `srednia_obszar(p_id_ucznia, p_obszar) RETURN NUMBER` – średnia w danym obszarze
- `porownaj_uczniow(p_id_ucznia_1, p_id_ucznia_2)` – porównuje dwóch uczniów

4.6 Kursory

W projekcie wykorzystano trzy rodzaje kursorów:

- **Jawne** – deklarowane z `CURSOR`, obsługiwane przez `OPEN/FETCH/CLOSE` (`pkg_uczen.lista_uczniow`)
- **Niejawne (FOR)** – uproszczona składnia `FOR rec IN (SELECT...)` (`pkg_lekcja.raport_dzienny`)
- **REF CURSOR** – dynamiczne kursory zwracane przez funkcje (`SYS_REFCURSOR`)

5 Wyzwalacze (Triggery)

Zdefiniowano 16 wyzwalaczy podzielonych na kategorie.

5.1 Triggery walidacyjne

5.2 Triggery kontroli semestrów

5.3 Triggery blokujące usuwanie

5.4 Triggery audytowe

5.5 Szczegółowy opis wybranych triggerów

5.5.1 trg_lekcja_godziny_dziecka

Uczniowie poniżej 15 lat uczęszczają do zwykłej szkoły (podstawowej lub średniej). Ich zajęcia kończą się około 13:00–14:00, dlatego lekcje muzyki mogą odbywać się tylko w godzinach 14:00–19:00.

```
IF v_wiek < 15 THEN
    v_godzina := TO_NUMBER(SUBSTR(:NEW.godzina_start, 1, 2));
    IF v_godzina < 14 OR v_godzina >= 19 THEN
        RAISE_APPLICATION_ERROR(-20050,
            'Dziecko moze miec lekcje tylko 14:00-19:00');
    END IF;
END IF;
```

Trigger	Typ	Działanie
trg_uczen_wiek	BEFORE INSERT	Sprawdza minimalny wiek ucznia (5 lat)
trg_lekcja_konflikt_ucznia	BEFORE INSERT	Sprawdza czy uczeń nie ma innej lekcji o tej godzinie
trg_lekcja_konflikt_nauczyciela	BEFORE INSERT	Sprawdza czy nauczyciel nie ma innej lekcji o tej godzinie
trg_lekcja_godziny_dziecka	BEFORE INSERT/UPDATE	Dzieci (<15 lat) mogą mieć lekcje tylko 14:00–19:00
trg_lekcja_limit_nauczyciela	BEFORE INSERT	Sprawdza limit 6h dziennie dla nauczyciela
trg_lekcja_limit_ucznia	BEFORE INSERT	Sprawdza limit 2 lekcji dziennie dla ucznia
trg_lekcja_konflikt_sali	BEFORE INSERT	Sprawdza czy sala nie jest zajęta
trg_lekcja_w_semestrze	BEFORE INSERT	Sprawdza czy data lekcji mieści się w aktywnym semestrze

Tabela 4: Triggery walidacyjne

Trigger	Typ	Działanie
trg_semestr_tylko_jeden_aktywny	AFTER UPDATE	Dezaktywuje inne semestry przy aktywacji jednego

Tabela 5: Triggery kontroli semestrów

5.5.2 trg_lekcja_limit_nauczyciela

Nauczyciel może prowadzić maksymalnie 6 godzin (360 minut) lekcji dziennie. Trigger sumuje czas wszystkich zaplanowanych lekcji nauczyciela w danym dniu i blokuje dodanie kolejnej, jeśli limit zostałby przekroczony.

5.5.3 trg_semestr_tylko_jeden_aktywny

W systemie może być aktywny tylko jeden semestr. Po ustawieniu flagi czy_aktywny='T' dla jednego semestru, trigger automatycznie dezaktywuje wszystkie pozostałe.

5.6 Kody błędów

6 Przyjęte ograniczenia

6.1 Ograniczenia dotyczące uczniów

1. Minimalny wiek ucznia: 5 lat
2. Adres email musi być unikalny
3. Uczeń może mieć maksymalnie 2 lekcje dziennie
4. Dzieci poniżej 15 lat mogą mieć lekcje tylko w godzinach 14:00–19:00 (uczęszczają do zwykłej szkoły)
5. Nie można usunąć ucznia, który ma aktywne zapisy na kursy

Trigger	Typ	Działanie
trg_uczen_przed_usunieciami	BEFORE DELETE	Blokuje usunięcie ucznia z zapisami
trg_nauczyciel_przed_usunieciami	BEFORE DELETE	Blokuje usunięcie nauczyciela z lekcjami
trg_kurs_przed_usunieciami	BEFORE DELETE	Blokuje usunięcie kursu z zapisami
trg_sala_przed_usunieciami	BEFORE DELETE	Blokuje usunięcie sali z lekcjami
trg_semestr_przed_usunieciami	BEFORE DELETE	Blokuje usunięcie aktywnego semestru

Tabela 6: Triggery blokujące usuwanie

Trigger	Typ	Działanie
trg_ocena_audit	AFTER INSERT/UPDATE/DELETE	Logowanie operacji na ocenach
trg_lekcja_status_audit	AFTER UPDATE	Logowanie zmian statusu lekcji

Tabela 7: Triggery audytowe

6.2 Ograniczenia dotyczące nauczycieli

1. Nauczyciel może uczyć maksymalnie 5 instrumentów (VARRAY)
2. Adres email musi być unikalny
3. Nauczyciel może prowadzić maksymalnie 6 godzin (360 minut) lekcji dziennie
4. Nauczyciel nie może mieć dwóch lekcji o tej samej godzinie
5. Nie można usunąć nauczyciela, który ma przypisane lekcje

6.3 Ograniczenia dotyczące lekcji

1. Lekcje tylko w godzinach 08:00–20:00
2. Czas trwania lekcji: 30, 45, 60 lub 90 minut
3. Statusy lekcji: zaplanowana, odbyta, odwołana
4. Uczeń nie może mieć dwóch lekcji o tej samej godzinie
5. Nauczyciel nie może mieć dwóch lekcji o tej samej godzinie
6. Sala nie może być zajęta przez dwie lekcje o tej samej godzinie
7. Data lekcji musi mieścić się w zakresie dat aktywnego semestru

6.4 Ograniczenia dotyczące ocen

1. Oceny w skali 1–6 (polska skala szkolna)
2. Obszary oceny: technika, teoria, słuch, rytm, interpretacja, ogólna

6.5 Ograniczenia dotyczące kursów

1. Typy kursów: indywidualny, grupowy
2. Cena za lekcję musi być większa od 0
3. Domyślny czas lekcji: 30, 45, 60 lub 90 minut

Kod	Opis
-20001	Uczeń musi mieć minimum 5 lat
-20020	Konflikt nauczyciela (ma już lekcję o tej godzinie)
-20021	Konflikt ucznia (ma już lekcję o tej godzinie)
-20030	Nie można usunąć ucznia z zapisami
-20031	Nie można usunąć nauczyciela z lekcjami
-20032	Nie można usunąć kursu z zapisami
-20033	Nie można usunąć sali z lekcjami
-20050	Dziecko może mieć lekcje tylko 14:00–19:00
-20051	Nauczyciel przekroczył limit 6h dziennie
-20052	Uczeń przekroczył limit 2 lekcji dziennie
-20054	Sala jest zajęta o tej godzinie
-20055	Lekcja poza zakresem dat aktywnego semestru
-20060	Nie można usunąć aktywnego semestru

Tabela 8: Kody błędów aplikacji

4. Nie można usunąć kursu, który ma aktywne zapisy

6.6 Ograniczenia dotyczące sal

1. Nazwa sali musi być unikalna
2. Pojemność sali musi być większa od 0
3. Nie można usunąć sali, która ma przypisane lekcje

6.7 Ograniczenia dotyczące semestrów

1. Nazwa semestru musi być unikalna
2. Data zakończenia musi być późniejsza niż data rozpoczęcia
3. W systemie może być aktywny tylko jeden semestr
4. Nie można usunąć aktywnego semestru

6.8 Ograniczenia dotyczące zapisów

1. Statusy zapisu: aktywny, zawieszony, zakończony

7 Role użytkowników

7.1 Administrator (ROLA_ADMIN_SZKOLY)

Pełny dostęp do systemu:

- Zarządzanie wszystkimi danymi (CRUD na wszystkich tabelach)
- Zarządzanie semestrami i salami
- Wykonywanie wszystkich pakietów
- Zarządzanie użytkownikami i uprawnieniami

7.2 Nauczyciel (ROLA_NAUCZYCIEL)

Prowadzenie lekcji i ocenianie:

- Odczyt danych uczniów, kursów, sal, semestrów
- Tworzenie i aktualizacja lekcji
- Dodawanie i modyfikacja ocen postępu
- Generowanie raportów dziennych
- Sprawdzanie dostępności sal

7.3 Sekretariat (ROLA_SEKRETARIAT)

Zarządzanie uczniami i zapisami:

- Pełne zarządzanie danymi uczniów
- Pełne zarządzanie zapisami na kursy
- Odczyt danych nauczycieli, kursów, lekcji, ocen
- Sprawdzanie dostępności sal
- Generowanie list i raportów

Tabela	Admin	Nauczyciel	Sekretariat
t_uczen	CRUD	R	CRUD
t_nauczyciel	CRUD	R	R
t_kurs	CRUD	R	R
t_zapis	CRUD	R	CRUD
t_lekcja	CRUD	CRU	R
t_ocena_postepu	CRUD	CRU	R
t_sala	CRUD	R	R
t_semestr	CRUD	R	R

Tabela 9: Macierz uprawnień (C=Create, R=Read, U=Update, D=Delete)

7.4 Użytkownicy testowi

- admin_szkoly (hasło: Admin123!) – rola ROLA_ADMIN_SZKOLY
- nauczyciel_jan (hasło: Nauczyciel123!) – rola ROLA_NAUCZYCIEL
- sekretariat_anna (hasło: Sekretariat123!) – rola ROLA_SEKRETARIAT

8 Testy

Plik 06_testy.sql zawiera 46 kompleksowych testów w 7 kategoriach.

8.1 Przykładowe testy

- Test 1.1: Metody t_uczen_obj (wiek, czy_pelnoletni, pelne_dane)
- Test 2.1: CHECK – nieprawidłowy status lekcji
- Test 3.5: pkg_uczen.lista_uczniow
- Test 4.2: Trigger – dziecko nie może mieć lekcji o 10:00
- Test 4.4: Trigger – uczeń max 2 lekcje dziennie
- Test 5.2: Scenariusz – sprawdzenie wszystkich godzin dla dziecka

Kategoria	Testów	Zakres
1. Typy (metody)	8	Wszystkie 9 typów obiektowych
2. Tabele (CHECK)	6	Ograniczenia CHECK na kolumnach
3. Pakiety	13	Wszystkie 26 procedur/funkcji
4. Triggery	8	Walidacja i blokady
5. Scenariusze	4	Cykle życia, planowanie, limity
6. Usuwanie	5	Blokowanie usuwania powiązanych danych
7. Użytkownicy	2	Role i uprawnienia
RAZEM	46	

Wymaganie	Realizacja
Definicje typów obiektowych z metodami	9 typów, 15 metod
Tabele obiektowe (wierszowe)	8 tabel
Referencja i dereferencja	8 REF, Deref
Wstawianie danych z referencją	Tak
VARRAY do relacji 1:N	t_lista_instrumentow
Pakiety PL/SQL	5 pakietów, 26 procedur
Kursory i REF kursory	3 typy kursorów
Obsługa błędów	EXCEPTION, RAISE
Wyzwalacze	16 triggerów
Role użytkowników	3 role
Testy	46 testów

Tabela 11: Realizacja wymagań projektowych

10.1 Pliki projektu

1. 01_typy.sql – definicje 9 typów obiektowych z 15 metodami
2. 02_tabele.sql – 8 tabel obiektowych, 8 sekwencji, indeksy, CHECK
3. 03_pakiety.sql – 5 pakietów z 26 procedurami i funkcjami
4. 04_triggery.sql – 16 wyzwalaczy
5. 05_dane.sql – dane testowe
6. 06_testy.sql – 46 testów w 7 kategoriach
7. 07_uzytkownicy.sql – 3 role, 3 użytkownicy