

Bazy Danych 1
Projekt, implementacja bazy danych
oraz aplikacji Java
Temat projektu:
“Uniwersytecki System Studentów”

Wykonali:
Marcin Kotulski
Marcin Wojtkowski

1.Zakres i cel projektu	1
1.1. Opis założeń funkcjonalnych projektowanej bazy danych	1
2.Model konceptualny	1
2.1. Definicje zbiorów encji określonych w projekcie	1
2.2. Ustalenie związków między encjami i ich typów	2
2.3. Schemat ER na poziomie konceptualnym	2
3. Model logiczny	2
3.1. Charakterystyka modelu relacyjnego	2
3.2. Model ER na poziomie modelu logicznego	3
4. Faza fizyczna	3
4.1. Opis procedur, funkcji i wyzwalaczy.	3
5. Aplikacja	4

1.Zakres i cel projektu

Celem projektu jest stworzenie modelu bazy danych na poziomie konceptualnym, logicznym oraz fizycznym. A także napisanie aplikacji połączonej z bazą danych w języku Java. Baza danych zawiera informacje na temat studentów, wykładowców, przedmiotów oraz ocen. Wykorzystaliśmy do tego silnik bazy danych firmy Oracle.

Oprogramowanie z którego korzystaliśmy to:

- SQL Developer
- Toad Data Modeler
- Oracle Database 12
- Vistal Studio Code

1.1. Opis założeń funkcjonalnych projektowanej bazy danych

System ma za zadanie zarządzać ocenami studenta oraz przypisywać go do odpowiednich przedmiotów. Każdy przedmiot ma swojego wykładowcę jak i przypisany uniwersytet.

2.Model konceptualny

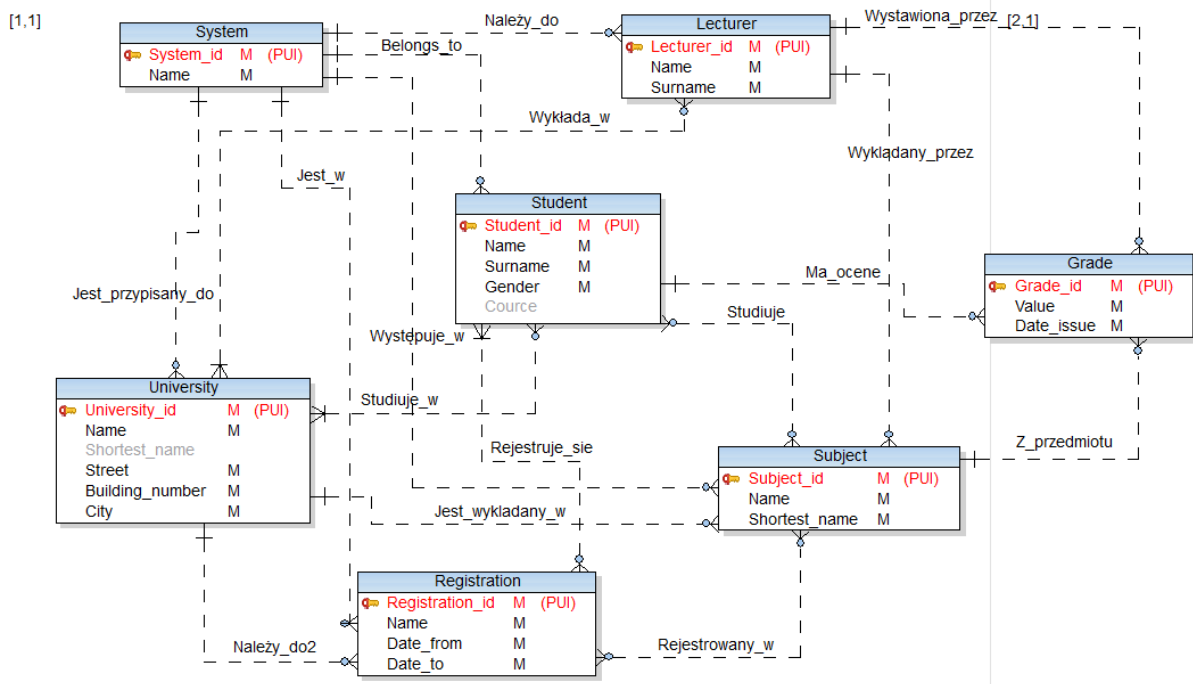
2.1. Definicje zbiorów encji określonych w projekcie

- System- Encja główna projektu
- Lecturer- Zawiera informacje o wykładowcy,
- University - Encja przechowująca dane uniwersytetów,
- Student - Podstawowe dane na temat studenta,
- Subject - informacje na temat przedmiotu wykładowanego na uniwersytecie,
- Registration- Zawiera Informacje o rejestracji na przedmioty,
- Grade - Informacje o ocenach studentów

2.2. Ustalenie związków między encjami i ich typów

Relationship Name	Relationship Type	Relationship Between	Cardinality
Belongs_to	Non-Identifying	System - Student	1..1 - 0..m
Jest przypisany do	Non-Identifying	System - University	1..1 - 0..m
Jest w	Non-Identifying	System - Registration	1..1 - 0..m
Jest wykładany w	Non-Identifying	University - Subject	1..1 - 0..m
Ma ocene	Non-Identifying	Student - Grade	1..1 - 0..m
Należy do	Non-Identifying	System - Lecturer	1..1 - 0..m
Należy do2	Non-Identifying	University - Registration	1..1 - 0..m
Rejestrowany w	Non-Identifying	Registration - Subject	0..n - 0..m
Rejestruje sie	Non-Identifying	Registration - Student	0..n - 1..m
Studiuje	Non-Identifying	Student - Subject	0..n - 0..m
Studiuje w	Non-Identifying	University - Student	1..n - 0..m
Wykładany przez	Non-Identifying	Lecturer - Subject	1..1 - 0..m
Wykłada w	Non-Identifying	University - Lecturer	1..n - 0..m
Wystawiona przez	Non-Identifying	Lecturer - Grade	1..1 - 0..m
Występuje w	Non-Identifying	System - Subject	1..1 - 0..m
Z przedmiotu	Non-Identifying	Subject - Grade	1..1 - 0..m

2.3. Schemat ER na poziomie konceptualnym



3. Model logiczny

3.1. Charakterystyka modelu relacyjnego

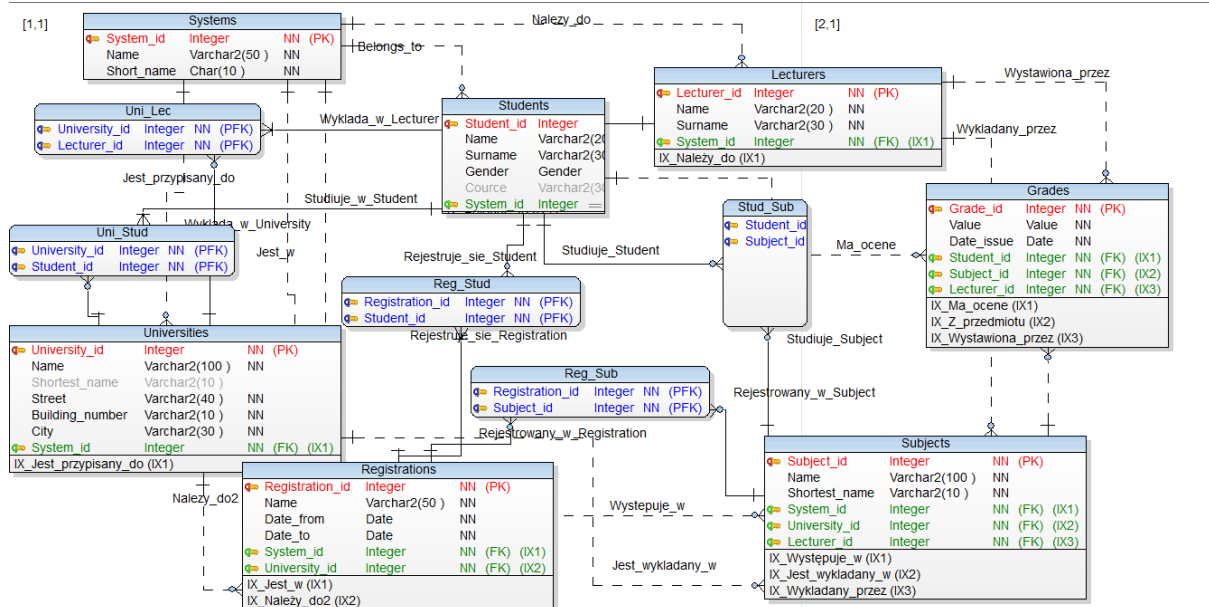
Po przekształceniu naszego modelu konceptualnego na model logiczny za pomocą Toad Data Modeler nastąpiły poniższe zmiany:

- Usunięcie relacji M:N na rzecz tabel łączących
- Dodanie do tabel kluczy obcych

Model relacyjny wymaga też następujących zmian:

- Zmiana atrybutów segmentowych na atomowe
- Usunięcie atrybutów wielowartościowych i zamiana ich na osobne encje
- Ustalenie właściwych typów relacji

3.2. Model ER na poziomie modelu logicznego



4. Faza fizyczna

W programie Toad Data Modeler został wygenerowany skrypt DDL do założenia bazy danych. Został on załadowany w programie SQL Developer.

Zostały też napisane odpowiednie skrypty ładujące dane do bazy oraz testujące procedury, funkcje i wyzwalacze.

4.1. Opis procedur, funkcji i wyzwalaczy.

Funkcje:

- **calculate_avg** - funkcja obliczająca średnią wszystkich ocen danego studenta,
- **add_grade** - funkcja dodająca ocenę studentowi. Przyjmuje parametry: grade_val, lecturer_id, subject_id i student_id. Funkcja wykorzystuje również sekwencje grade_seq.

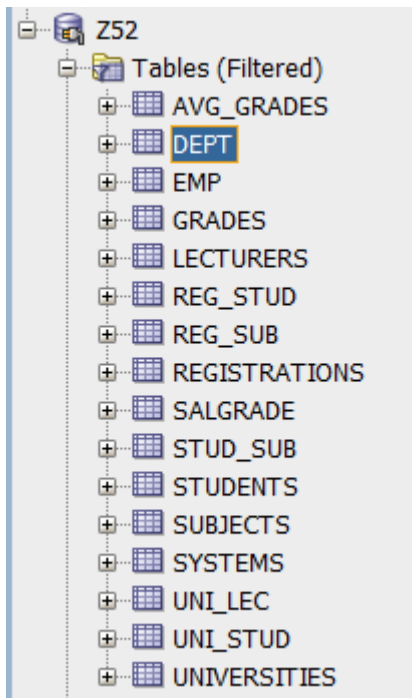
Procedury:

- **add_bonus** - procedura wystawiająca ocenę 5 z przedmiotu, z którego student ma średnią powyżej 4,5. Wykorzystuje ona kursory niejawne,
- **avg_grade** - Funkcja wyliczająca średnie studentów z przedmiotów i dodająca je do tabeli avg_grades. Jeżeli średnia jest już zapisana, procedura avg_grade jedynie aktualizuje jej wartość. Wykorzystuje ona kursory niejawne,

Wyzwalacze:

- **add_stud_sub** - trigger sprawdzający przez dodaniem wiersza do tabeli stud_sub czy student i przedmiot o danym ID istnieją w tabelach. W przypadku kiedy nie znajdzie studenta lub przedmiotu wyświetla błąd.

- **check_viability** - wyzwalacz sprawdzający przed dodaniem oceny czy dany prowadzący prowadzi przedmiot, z którego jest wystawiana ocena. Jeżeli nie, wyświetli się błąd a ocena nie zostanie wstawiona.



5. Aplikacja

Elementem projektu było również napisanie prostej aplikacji w języku Java z użyciem technologii JDBC. W naszym przypadku funkcjonalność aplikacji polega na łączeniu się z bazą danych, możliwości wyświetlenia listy studentów na konkretnym kierunku studiów oraz wprowadzenia nowej oceny studentowi.