



**WYDZIAŁ  
ELEKTROTECHNIKI  
I INFORMATYKI**  
POLITECHNIKI RZESZOWSKIEJ

**Bazy danych  
Laboratorium**

*Zarządzanie bazą danych Oracle*

**Stanislau Antanovich**  
nr. indeksu: 173590  
gr. lab: L04

## Spis treści

<b>1</b>	<b>Realizacja</b>	<b>2</b>
<b>2</b>	<b>Wnioski</b>	<b>4</b>

## Spis rysunków

1	<i>Tworzenie przestrzeni TABLESPACE o nazwie “project_tablespace”</i>	2
2	<i>Wdrożenie bazy danych</i>	3
3	<i>Tworzenie sekwencji “student_seq”</i>	3
4	<i>Funkcja, która dodaje nowego studenta</i>	3
5	<i>Wprowadzenie danych do tablicy</i>	4
6	<i>Tworzenie widoków</i>	4

## Spis poleceń

1	<i>Tworzenie przestrzeni TABLESPACE o nazwie “project_tablespace”</i>	2
2	<i>Tworzenie użytkownika</i>	2
3	<i>Wdrożenie bazy danych</i>	2
4	<i>Tworzenie sekwencji “student_seq”</i>	3
5	<i>Funkcja, która dodaje nowego studenta</i>	3
6	<i>Tworzenie widoków</i>	4
7	<i>Zwracanie raportu do pliku</i>	4

# 1 Realizacja

1. Tworzenie przestrzeni TABLESPACE o nazwie "project\_tablespace".

```
CREATE TABLE SPACE project_tablespace  
DATAFILE 'project_tablespace.dbf' SIZE 100M  
AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
```

Polecenie 1. *Tworzenie przestrzeni TABLESPACE o nazwie "project\_tablespace"*

```
tablespace PROJECT_TABLESPACE created.
```

Rysunek 1: *Tworzenie przestrzeni TABLESPACE o nazwie "project\_tablespace"*

2. Tworzenie użytkownika dla nowej bazy danych oraz przestrzeni o danych:

- nazwa: "student1"
- hasło: zgodnym z dniem tworzenia np. "07052023"

Przypisanie do konta użytkownika przestrzeń "project\_tablespace" oraz odpowiednie dostępy oraz role.

```
CREATE USER student1 IDENTIFIED BY '07052023';  
ALTER USER student1 DEFAULT TABLESPACE project_tablespace;  
GRANT ALL PRIVILEGES TO student1;
```

Polecenie 2. *Tworzenie użytkownika*

3. Wdrożenie bazy danych.

```
CREATE TABLE student (  
    id INTEGER PRIMARY KEY,  
    name VARCHAR2(255),  
    surname VARCHAR2(255),  
    index_num INTEGER  
);  
  
CREATE TABLE topic(  
    id INTEGER PRIMARY KEY,  
    name VARCHAR2(255)  
);  
  
CREATE TABLE assignment(  
    student_id INTEGER,  
    topic_id INTEGER,  
    start_date DATE,  
    rating FLOAT,  
    FOREIGN KEY (student_id) REFERENCES student(id),  
    FOREIGN KEY (topic_id) REFERENCES topic(id)  
);  
  
CREATE TABLE topic_assignment(  
    topic_id INTEGER,  
    assignment_id INTEGER  
);  
  
CREATE TABLE issue(  
    id INTEGER PRIMARY KEY,  
    name VARCHAR2(255),  
    topic_id INTEGER,  
    FOREIGN KEY (topic_id) REFERENCES topic(id)  
);
```

Polecenie 3. *Wdrożenie bazy danych*

```

table STUDENT created.
table TOPIC created.
table ASSIGNMENT created.
table TOPIC_ASSIGNMENT created.
table ISSUE created.

```

Rysunek 2: *Wdrożenie bazy danych*

4. Tworzenie sekwencji “*student\_seq*” dedykowaną dla studenta w przedziale od 0 do 10000.

```

CREATE SEQUENCE student_seq
START WITH 1
INCREMENT BY 1
MAXVALUE 10000
NOCYCLE
NOCACHE;

```

Polecenie 4. *Tworzenie sekwencji “student\_seq”*

```

sequence STUDENT_SEQ created.

```

Rysunek 3: *Tworzenie sekwencji “student\_seq”*

5. Funkcja, która pozwala dodać nowego studenta (addStudent). Wykorzystanie utworzonej sekwencji “*student\_seq*” dla ustalenia kolejnego ID. Zwracanie z funkcji ID dodanego studenta.

```

CREATE OR REPLACE FUNCTION addStudent(
    p_name IN VARCHAR2,
    p_surname IN VARCHAR2,
    p_index_num IN INTEGER
) RETURN INTEGER IS
    v_id INTEGER;
BEGIN
    SELECT student_seq.NEXTVAL INTO v_id FROM DUAL;
    INSERT INTO student (id, name, surname, index_num)
    VALUES (v_id, p_name, p_surname, p_index_name);
    RETURN v_id;
END;

```

Polecenie 5. *Funkcja, która dodaje nowego studenta*

	ID	NAME	SURNAME	INDEX_NUM
1	1	Jan	Kowalski	123456
2	3	Jan	Nowak	123456
3	4	Adam	Kowalski	123457
4	5	Justyna	Duda	123459
5	6	Artur	Waclaw	123459

Rysunek 4: *Funkcja, która dodaje nowego studenta*

6. Wprowadzenie danych do tablicy. Wywołanie funkcji addStudent.

	STUDENT_ID	TOPIC_ID	START_DATE	RATING
1	1	5	24/05/01	(null)
2	3	6	24/05/02	4
3	4	7	24/05/03	4,5
4	5	8	24/05/04	3

  

ID	NAME
1	5 Projekt dla klasyfikacji Irysów
2	6 Aplikacja bazodanowa do obsługi wypożyczalni samochodów
3	7 Gra platformowa w środowisku UNITY
4	8 Kalkulator naukowy w środowisku Android

  

	TOPIC_ID	ASSIGNMENT_ID
1	5	1
2	6	2
3	6	3
4	7	4
5	8	3

  

ID	NAME
1	5 Analiza danych
2	6 Bezpieczeństwo systemów
3	7 Programowanie w Pythonie
4	8 Sieci komputerowe

Rysunek 5: Wprowadzenie danych do tablicy

7. Tworzenie widoków odpowiedzialnych za:

- wyświetlenie wszystkich tematów projektów
- wyświetlenie wszystkich studentów, którzy nie mają przypisanego tematu

```
CREATE OR REPLACE VIEW topcis AS
SELECT name
FROM topic;
CREATE OR REPLACE VIEW students_lacking_topic AS
SELECT *
FROM student WHERE NOT EXISTS (
SELECT 1
FROM assignment
WHERE assignment.student_id=student.id
);
```

Polecenie 6. Tworzenie widoków

```
view TOPCIS created.
view STUDENTS_LACKING_TOPIC created.
```

Rysunek 6: Tworzenie widoków

8. Zwracanie raportu do pliku o rozszerzeniu .csv z informacją o nagłówku(kolumny):

- dane studenta(indeks, imię oraz nazwisko), nazwa projektu, ocena
- zwracanie danych osób, które uzyskały pozytywną ocenę(większą lub równą niż 3.0)

```
SELECT student.index_num , student.name , student.surname , topic.name
AS assignment.rating
FROM student
JOIN assignment ON student.id = assignment.student_id
JOIN topic ON topic.id = assignment.topic_id
WHERE assignment.rating >= 3;
```

Polecenie 7. Zwracanie raportu do pliku

## 2 Wnioski

Dzięki działaniom wykonanym podczas laboratorium udało się poszerzyć wiedzę na temat baz danych oraz zrozumieć, jakie możliwości oferują. Pozwoliło to nie tylko na zdobycie teoretycznej wiedzy, ale również na praktyczne zastosowanie różnych funkcji SQL w rzeczywistych scenariuszach.