



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI**
POLITECHNIKI RZESZOWSKIEJ

**Bazy danych
Laboratorium**

Zapytania DDL SQL Oracle 1

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

11 maja 2024

Spis treści

1	Wprowadzenie	2
1.1	Cel ćwiczenia	2
1.2	Przygotowanie	2
2	Realizacja	2
3	Wnioski	5

Spis rysunków

1	<i>Tworzenie tabeli “TEST” zawierającej 4 przykładowe pola przechowujące informacje z Identyfikatorem, tekstem, czasem i wartością zmiennoprzecinkową</i>	2
2	<i>Usunięcie stworzonej tabeli odpowiednią komendą SQL</i>	2
3	<i>Tworzenie tabeli “ZADANIA”</i>	3
4	<i>Dodanie nowej kolumny “KOMENTARZ”</i>	3
5	<i>Realizacja powiązania tabeli “ZADANIA” z tabelą “PRACOWNICY”</i>	3
6	<i>Modyfikacja tabeli “ZADANIA”</i>	4
7	<i>Dodanie do tabeli “ZADANIA”</i>	4
8	<i>Wyświetlenie następującego wyniku z użyciem zapytania SQL</i>	4
9	<i>Wyświetlenie następującego wyniku z użyciem zapytania SQL</i>	4
10	<i>Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania</i>	5

Spis poleceń

1	<i>Tworzenie tabeli “TEST” zawierającej 4 przykładowe pola przechowujące informacje z Identyfikatorem, tekstem, czasem i wartością zmiennoprzecinkową</i>	2
2	<i>Usunięcie stworzonej tabeli odpowiednią komendą SQL</i>	2
3	<i>Tworzenie tabeli “ZADANIA”</i>	3
4	<i>Dodanie nowej kolumny “KOMENTARZ”</i>	3
5	<i>Realizacja powiązania tabeli “ZADANIA” z tabelą “PRACOWNICY”</i>	3
6	<i>Modyfikacja tabeli “ZADANIA”</i>	4
7	<i>Dodanie do tabeli “ZADANIA”</i>	4
8	<i>Wyświetlenie następującego wyniku z użyciem zapytania SQL</i>	4
9	<i>Wyświetlenie następującego wyniku z użyciem zapytania SQL</i>	4
10	<i>Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania</i>	5

1 Wprowadzenie

1.1 Cel ćwiczenia

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1.2 Przygotowanie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2 Realizacja

1. Tworzenie tabeli *TEST* zawierającej 4 przykładowe pola przechowujące informacje z Identyfikatorem, tekstem, czasem i wartością zmiennoprzecinkową

```
CREATE TABLE TEST (ID INT PRIMARY KEY, TEKST VARCHAR(255), CZAS TIMESTAMP, WARTOSC  
FLOAT);
```

Polecenie 1. *Tworzenie tabeli "TEST" zawierającej 4 przykładowe pola przechowujące informacje z Identyfikatorem, tekstem, czasem i wartością zmiennoprzecinkową*

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID	NUMBER(38,0)	No	(null)	1 (null)	
2	TEKST	VARCHAR2(255 BYTE)	Yes	(null)	2 (null)	
3	CZAS	TIMESTAMP(6)	Yes	(null)	3 (null)	
4	WARTOSC	FLOAT	Yes	(null)	4 (null)	

Rysunek 1: *Tworzenie tabeli "TEST" zawierającej 4 przykładowe pola przechowujące informacje z Identyfikatorem, tekstem, czasem i wartością zmiennoprzecinkową*

2. Usunięcie stworzonej tabeli odpowiednią komendą SQL.

```
DROP TABLE TEST;
```

Polecenie 2. *Usunięcie stworzonej tabeli odpowiednią komendą SQL*

```
table TEST created.  
table TEST dropped.
```

Rysunek 2: *Usunięcie stworzonej tabeli odpowiednią komendą SQL*

3. Tworzenie tabeli *ZADANIA*(*TASKS*) zawierającą następujące kolumny:

- ID_ZADANIA
- NAZWA
- DATA_ROZPOCZECIA
- DATA_ZAKONCZENIA

```
CREATE TABLE ZADANIA (
  ID_ZADANIA INT PRIMARY KEY,
  NAZWA VARCHAR(255),
  DATA_ROZPOCZECIA DATE,
  DATA_ZAKONCZENIA DATE
);
```

Polecenie 3. Tworzenie tabeli “ZADANIA”

	2	COLUMN_NAME	2	DATA_TYPE	2	NULLABLE	DATA_DEFAULT	2	COLUMN_ID	2	COMMENTS
1	ID_ZADANIA	NUMBER(38,0)	No	(null)		1	(null)				
2	NAZWA	VARCHAR2(255 BYTE)	Yes	(null)		2	(null)				
3	DATA_ROZPOCZECIA	DATE	Yes	(null)		3	(null)				
4	DATA_ZAKONCZENIA	DATE	Yes	(null)		4	(null)				

Rysunek 3: Tworzenie tabeli “ZADANIA”

4. Dodanie nowej kolumny *KOMENTARZ*(*ALTER*)

```
ALTER TABLE ZADANIA ADD KOMENTARZ VARCHAR(255);
```

Polecenie 4. Dodanie nowej kolumny “KOMENTARZ”

	2	COLUMN_NAME	2	DATA_TYPE	2	NULLABLE	DATA_DEFAULT	2	COLUMN_ID	2	COMMENTS
1	ID_ZADANIA	NUMBER(38,0)	No	(null)		1	(null)				
2	NAZWA	VARCHAR2(255 BYTE)	Yes	(null)		2	(null)				
3	DATA_ROZPOCZECIA	DATE	Yes	(null)		3	(null)				
4	DATA_ZAKONCZENIA	DATE	Yes	(null)		4	(null)				
5	KOMENTARZ	VARCHAR2(255 BYTE)	Yes	(null)		5	(null)				

Rysunek 4: Dodanie nowej kolumny “KOMENTARZ”

5. Realizacja powiązania tabeli *ZADANIA* z tabelą *PRACOWNICY*

```
ALTER TABLE ZADANIA ADD ID_PRACOWNIKA NUMBER(4);
ALTER TABLE ZADANIA ADD CONSTRAINT FK_PRACOWNIK FOREIGN KEY (ID_PRACOWNIKA) REFERENCES
PRACOWNICY(ID_PRACOWNIKA);
```

Polecenie 5. Realizacja powiązania tabeli “ZADANIA” z tabelą “PRACOWNICY”

	2	COLUMN_NAME	2	DATA_TYPE	2	NULLABLE	DATA_DEFAULT	2	COLUMN_ID	2	COMMENTS
1	ID_ZADANIA	NUMBER(38,0)	No	(null)		1	(null)				
2	NAZWA	VARCHAR2(255 BYTE)	Yes	(null)		2	(null)				
3	DATA_ROZPOCZECIA	DATE	Yes	(null)		3	(null)				
4	DATA_ZAKONCZENIA	DATE	Yes	(null)		4	(null)				
5	KOMENTARZ	VARCHAR2(255 BYTE)	Yes	(null)		5	(null)				
6	ID_PRACOWNIKA	NUMBER(4,0)	Yes	(null)		6	(null)				

Rysunek 5: Realizacja powiązania tabeli “ZADANIA” z tabelą “PRACOWNICY”

6. Modyfikacja tabeli *ZADANIA* w taki sposób, aby rekordy miały określone ograniczenia:

- ID_ZADANIE – PRIMARY KEY
- NAZWA – UNIQUE
- DATA_ROZPOCZECIA – NOT NULL
- KOMENTARZ – NOT NULL, DEFAULT
- ID_PRACOWNIKA – FOREIGN KEY

```
CREATE TABLE ZADANIA (  
  ID_ZADANIA INT PRIMARY KEY,  
  NAZWA VARCHAR(100) UNIQUE,  
  DATA_ROZPOCZECIA DATE NOT NULL,  
  DATA_ZAKONCZENIA DATE,  
  KOMENTARZ VARCHAR(255) NOT NULL DEFAULT 'komentarz',  
  ID_PRACOWNIKA INT,  
  FOREIGN KEY (ID_PRACOWNIKA) REFERENCES PRACOWNICY(ID_PRACOWNIKA)  
);
```

Polecenie 6. *Modyfikacja tabeli “ZADANIA”*

Rysunek 6: *Modyfikacja tabeli “ZADANIA”*

7. Dodanie do tabeli *ZADANIA* siedem przykładowych zadań powiązanych odpowiednią relacją z tabelą *PRACOWNICY*. Jeden pracownik musi posiadać przynajmniej trzy zadania.

Polecenie 7. *Dodanie do tabeli “ZADANIA”*

Rysunek 7: *Dodanie do tabeli “ZADANIA”*

8. Wyświetlenie następującego wyniku z użyciem zapytania SQL (nagłówek tabeli wynikowej): ID_PRACOWNIKA|NAZW
ZADAN

Polecenie 8. *Wyświetlenie następującego wyniku z użyciem zapytania SQL*

Rysunek 8: *Wyświetlenie następującego wyniku z użyciem zapytania SQL*

9. Wyświetlenie następującego wyniku z użyciem zapytania SQL (nagłówek tabeli wynikowej): ID_PRACOWNIKA|NAZW
TRWAJACYCH ZADAN

Polecenie 9. *Wyświetlenie następującego wyniku z użyciem zapytania SQL*

Rysunek 9: *Wyświetlenie następującego wyniku z użyciem zapytania SQL*

10. Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania

Polecenie 10. *Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania*

Rysunek 10: *Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania*

3 Wnioski