



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI**
POLITECHNIKI RZESZOWSKIEJ

**Bazy danych
Laboratorium**

Zapytania DDL SQL Oracle 1

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

12 maja 2024

Spis treści

1	Wprowadzenie	2
1.1	Cel ćwiczenia	2
1.2	Przygotowanie	2
2	Realizacja	2
3	Wnioski	6

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>	5
8	<i>Wyświetlenie następującego wyniku z użyciem zapytania SQL</i>	5
9	<i>Wyświetlenie następującego wyniku z użyciem zapytania SQL</i>	6
10	<i>Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania</i>	6

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>	5
9	<i>Wyświetlenie następującego wyniku z użyciem zapytania SQL</i>	5
10	<i>Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania</i>	6

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_ZADANIE 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”

	⑧ COLUMN_NAME	⑧ DATA_TYPE	⑧ NULLABLE	DATA_DEFAULT	⑧ COLUMN_ID	⑧ COMMENTS
1	ID_ZADANIE	NUMBER(38,0)	No	{null}	1 {null}	
2	NAZWA	VARCHAR2(255 BYTE)	Yes	{null}	2 {null}	
3	DATA_ROZPOCZECIA	DATE	No	{null}	3 {null}	
4	DATA_ZAKONCZENIA	DATE	Yes	{null}	4 {null}	
5	KOMENTARZ	VARCHAR2(255 BYTE)	No	'KOMENTARZ'	5 {null}	
6	ID_PRACOWNIKA	NUMBER(38,0)	Yes	{null}	6 {null}	

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.

```
INSERT INTO ZADANIA (ID_ZADANIE, NAZWA, DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, KOMENTARZ,
    ID_PRACOWNIKA) VALUES (1, 'ZADANIE 1', '2024-05-12', '2024-05-13', 'ZADANIE 1',
    7369);

INSERT INTO ZADANIA (ID_ZADANIE, NAZWA, DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, KOMENTARZ,
    ID_PRACOWNIKA) VALUES (2, 'ZADANIE 2', '2024-05-13', '2024-05-14', 'ZADANIE 2',
    7369);

INSERT INTO ZADANIA (ID_ZADANIE, NAZWA, DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, KOMENTARZ,
    ID_PRACOWNIKA) VALUES (3, 'ZADANIE 3', '2024-05-14', '2024-05-15', 'ZADANIE 3',
    7369);

INSERT INTO ZADANIA (ID_ZADANIE, NAZWA, DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, KOMENTARZ,
    ID_PRACOWNIKA) VALUES (4, 'ZADANIE 4', '2024-05-15', '2024-05-16', 'ZADANIE 4',
    7499);

INSERT INTO ZADANIA (ID_ZADANIE, NAZWA, DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, KOMENTARZ,
    ID_PRACOWNIKA) VALUES (5, 'ZADANIE 5', '2024-05-16', '2024-05-17', 'ZADANIE 5',
    7499);

INSERT INTO ZADANIA (ID_ZADANIE, NAZWA, DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, KOMENTARZ,
    ID_PRACOWNIKA) VALUES (6, 'ZADANIE 6', '2024-05-17', '2024-05-18', 'ZADANIE 6',
    7505);

INSERT INTO ZADANIA (ID_ZADANIE, NAZWA, DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, KOMENTARZ,
    ID_PRACOWNIKA) VALUES (7, 'ZADANIE 7', '2024-05-18', '2024-05-19', 'ZADANIE 7',
    7505);
```

Polecenie 7. Dodanie do tabeli “ZADANIA”

	ID_ZADANIE	NAZWA	DATA_ROZPOCZECIA	DATA_ZAKONCZENIA	KOMENTARZ	ID_PRACOWNIKA
1	1	ZADANIE 1	24/05/12	24/05/13	ZADANIE 1	7369
2	2	ZADANIE 2	24/05/13	24/05/14	ZADANIE 2	7369
3	3	ZADANIE 3	24/05/14	24/05/15	ZADANIE 3	7369
4	4	ZADANIE 4	24/05/15	24/05/16	ZADANIE 4	7499
5	5	ZADANIE 5	24/05/16	24/05/17	ZADANIE 5	7499
6	6	ZADANIE 6	24/05/17	24/05/18	ZADANIE 6	7505
7	7	ZADANIE 7	24/05/18	24/05/19	ZADANIE 7	7505

Rysunek 7: Dodanie do tabeli "ZADANIA"

8. Wyświetlenie następującego wyniku z użyciem zapytania SQL (nagłówki tabeli wynikowej): ID_PRACOWNIKA|NAZWISKO|IMIE|ILOSC ZADAN

```
SELECT P.ID_PRACOWNIKA, P.NAZWISKO, P.IMIE, COUNT(Z.ID_ZADANIE) AS "ILOSC ZADAN" FROM
PRACOWNICY P LEFT JOIN ZADANIA Z ON P.ID_PRACOWNIKA = Z.ID_PRACOWNIKA GROUP BY P.
ID_PRACOWNIKA, P.NAZWISKO, P.IMIE;
```

Polecenie 8. Wyświetlenie następującego wyniku z użyciem zapytania SQL

ID_PRACOWNIKA	NAZWISKO	IMIE	ILOSC ZADAN
7505	DOYLE	JEAN	2
7950	JENSEN	ALICE	0
7521	WARD	CYNTHIA	0
7954	MURRAY	JAMES	0
7839	KING	FRANCIS	0
7919	DOUGLAS	MICHAEL	0
7600	PORTER	RAYMOND	0
7499	ALLEN	KEVIN	2
7676	SOMMERS	DENISE	0
7566	JONES	TERRY	0
7782	CLARK	CAROL	0
7564	LANGE	GREGORY	0
7507	BAKER	LESLIE	0
7506	DENNIS	LYNN	0
7902	FORD	JENNIFER	0
7788	SCOTT	DONALD	0
7698	BLAKE	MARION	0
7900	JAMES	FRED	0
7654	MARTIN	KENNETH	0
7876	ADAMS	DIANE	0
7569	ALBERTS	CHRIS	0
7916	ROBERTS	GRACE	0
7844	TURNER	MARY	0
7789	WEST	LIVIA	0
7555	PETERS	DANIEL	0
7934	MILLER	BARBARA	0
7560	DUNCAN	SARAH	0
7369	SMITH	JOHN	3
7557	SHAW	KAREN	0
7799	FISHER	MATTHEW	0
7820	ROSS	PAUL	0
7609	LEWIS	RICHARD	0

Rysunek 8: Wyświetlenie następującego wyniku z użyciem zapytania SQL

9. Wyświetlenie następującego wyniku z użyciem zapytania SQL (nagłówki tabeli wynikowej): ID_PRACOWNIKA|NAZWISKO|IMIE|ILOSC TRWAJACYCH ZADAN

```
SELECT P.ID_PRACOWNIKA, P.NAZWISKO, P.IMIE, COUNT(Z.ID_ZADANIE) AS "ILOSC TRWAJACYCH
ZADAN" FROM PRACOWNICY P INNER JOIN ZADANIA Z ON P.ID_PRACOWNIKA = Z.ID_PRACOWNIKA
WHERE Z.DATA_ZAKONCZENIA >= SYSDATE GROUP BY P.ID_PRACOWNIKA, P.NAZWISKO, P.IMIE;
```

Polecenie 9. Wyświetlenie następującego wyniku z użyciem zapytania SQL

ID_PRACOWNIKA	NAZWISKO	IMIE	ILOSC TRWAJACYCH ZADAN
7505	DOYLE	JEAN	2
7499	ALLEN	KEVIN	2
7369	SMITH	JOHN	3

Rysunek 9: Wyświetlenie następującego wyniku z użyciem zapytania SQL

10. Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania

```
SELECT P.ID_PRACOWNIKA, P.NAZWISKO, P.IMIE FROM PRACOWNICY P INNER JOIN ZADANIA Z ON P.ID_PRACOWNIKA = Z.ID_PRACOWNIKA GROUP BY P.ID_PRACOWNIKA, P.NAZWISKO, P.IMIE HAVING COUNT(Z.ID_ZADANIE) >= 2;
```

Polecenie 10. Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania

ID_PRACOWNIKA	NAZWISKO	IMIE
7505	DOYLE	JEAN
7499	ALLEN	KEVIN
7369	SMITH	JOHN

Rysunek 10: Wyświetlenie wszystkich pracowników posiadających przynajmniej dwa zadania

3 Wnioski