

SPRAWOZDANIE Z PROJEKTU

TWORZENIE DIAGRAMU ENCJI I RELACJI

SYSTEMY BAZ DANYCH

KIERUNEK: Informatyka i ekonometria

SEMESTR: IV

GRUPA ZAJĘCIOWA: PS 2

SKŁAD GRUPY PROJEKTOWEJ: Pragacz Michał, Rutkowska Maja, Szewczyk Izabela, Wasiluk Michalina, Wójcik Bartłomiej

PROWADZĄCY: dr inż. Eugenia Busłowska

DATA ODDANIA PROJEKTU: 20.06.2023

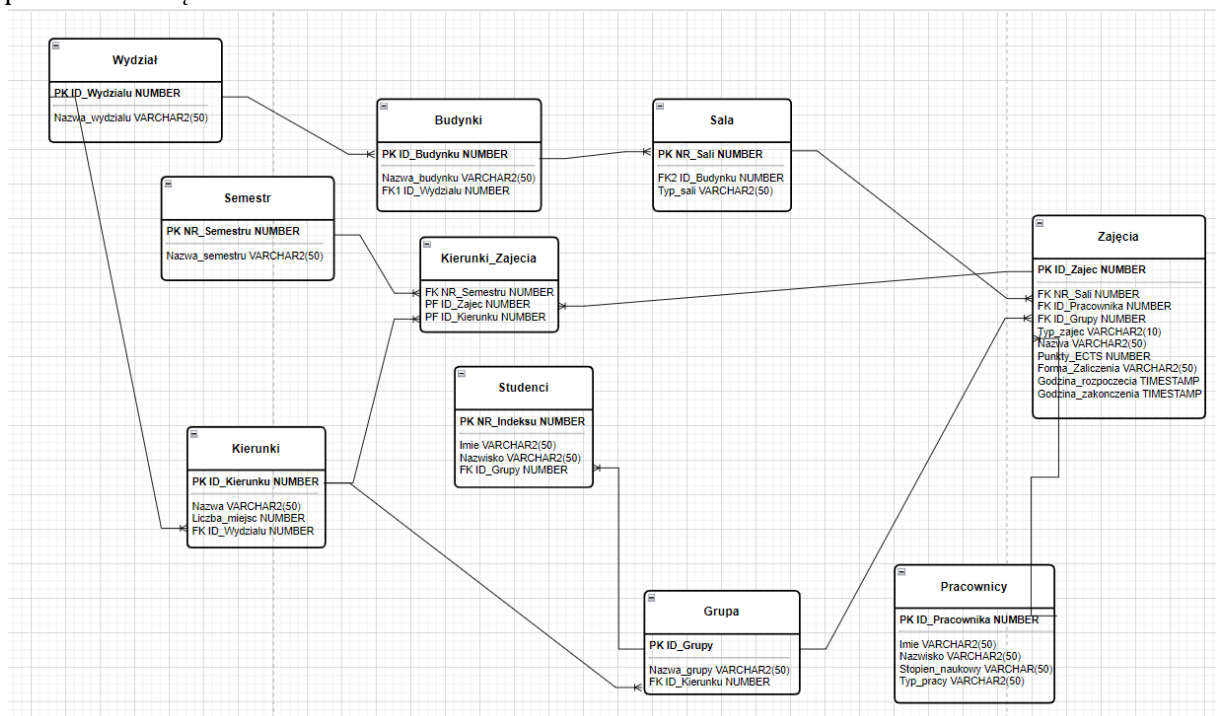
TREŚĆ ZADANIA PROJEKTOWEGO:

1. Stworzyć model logiczny w Oracle SQL Developer Data Modeler.
2. Dokonać transformacji diagramu ERD na model relacyjny.
3. Utworzyć własne tabele w języku SQL.
4. Dla utworzonych tabel stworzyć diagram relacji w Oracle SQL Developer Data Modeler.
5. Wprowadzić dane (1000 rekordów)
6. Wykonać zapytania do danych, sprawdzić wydajność wykonywanych poleceń.
7. Zaproponować podwyższenie wydajności wprowadzając indeksy.
8. Wprowadzić ewentualne zmiany schematów tabel lub połączeń tabel.
9. Przygotować dokumentację projektu.
10. Zaprezentować wnioski.

Utworzyliśmy model danych w postaci diagramów związków encji (ERD) spełniający poniższe wymagania funkcjonalne:

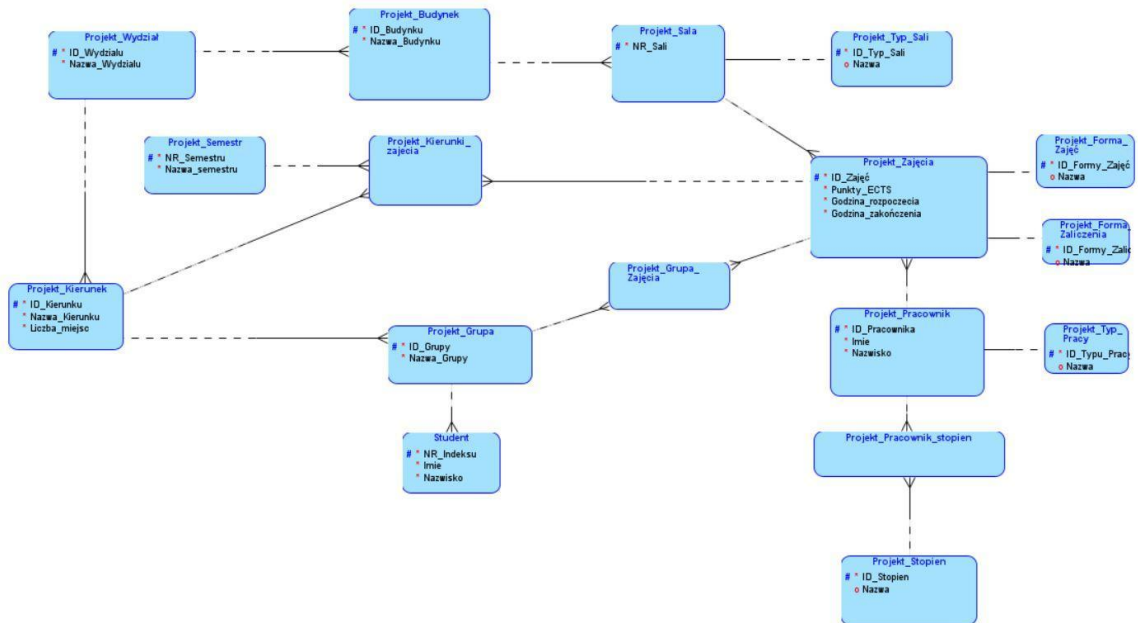
- Wydział dysponuje z góry znaną liczbą sal w dwóch budynkach,
- Sale są wykładowe, komputerowe, ćwiczeniowe i laboratoryjne,
- Sale mają swoje numery,
- W salach odbywają się zajęcia od godz. 8.00 do 21.00,
- Wszystkie zajęcia trwają 90 min,
- Jest znana nazwa zajęć na poszczególnych semestrach,
- Zajęcia mogą mieć formę: W, PS, ĆW, L, LEK, SEM,
- Zajęcia są prowadzone przez pracowników etatowych i nieetatowych,
- Pracownicy posiadają stopnie naukowe,
- Jeden pracownik nie może mieć jednocześnie zajęć w dwóch salach,
- Studenci mają zajęcia wg podziału na grupy.

Przeprowadziliśmy proces normalizacji, określiliśmy klucze główne i klucze obce, dokonaliśmy zmian atrybutów oraz określiliśmy typy wszystkich atrybutów, uwzględniając ich ograniczenia poza dziedziną.

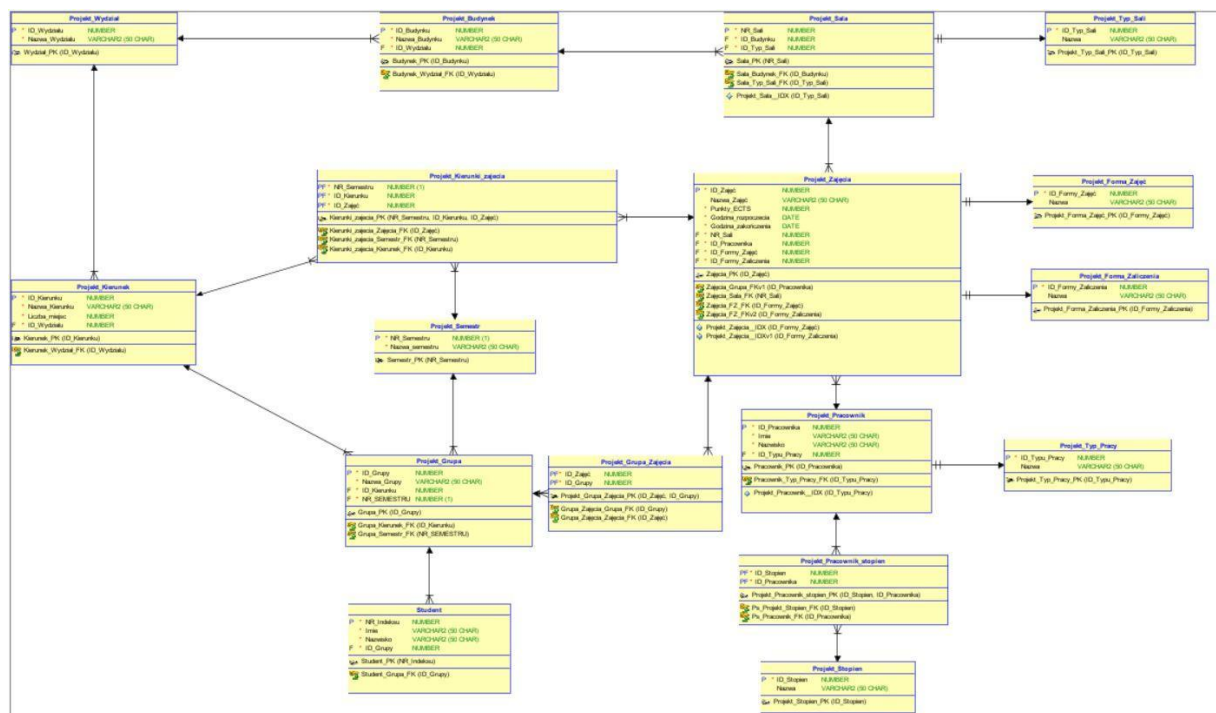


NASTĘPNIE STWORZYLIŚMY MODEL LOGICZNY W ORACLE SQL DEVELOPER DATA MODELER ORAZ DOKONALIŚMY JEGO TRANSFORMACJI NA MODEL RELACYJNY, WPROWADZILIŚMY RÓWNIEŻ KILKA POPRAWEK, KTÓRE ZWIĘKSZYŁY SPÓJNOŚĆ BAZY DANYCH.

MODEL LOGICZNY:



MODEL RELACYJNY:



OKREŚLILIŚMY NASTĘPUJĄCE DZIEDZINY:

Typ_Sali: wykładowa, komputerowa, ćwiczeniowa, laboratoryjna

Typ_Pracy: etatowy, nieetatowy

Forma_Zaliczenia: zaliczenie pisemne, egzamin, odpowiedź ustna, sprawozdania

Forma_Zajęc: W, PS, CW, L, LEK, SEM

Stopien: dr hab, prof, dr, mgr, inż

UTWORZYLIŚMY WŁASNE TABELY, ORAZ DODALIŚMY DO NICH OGRANICZENIA:

1		
2	CREATE TABLE projekt_budynek (
3	id_budynku NUMBER NOT NULL,	
4	nazwa_budynku VARCHAR2(50 CHAR) NOT NULL,	
5	id_wydzialu NUMBER NOT NULL	
6);	
7		
8	ALTER TABLE projekt_budynek ADD CONSTRAINT budynek_pk PRIMARY KEY (id_budynku);	
9		
10	CREATE TABLE projekt_forma_zajęc (
11	id_formy_zajęc NUMBER NOT NULL,	
12	nazwa VARCHAR2(50 CHAR)	
13);	
14		
15	ALTER TABLE projekt_forma_zajęc ADD CONSTRAINT projekt_forma_zajęc_pk PRIMARY KEY (id_formy_zajęc);	
16		
17	CREATE TABLE projekt_forma_zaliczenia (
18	id_formy_zaliczenia NUMBER NOT NULL,	
19	nazwa VARCHAR2(50 CHAR)	
20);	
21		
22	ALTER TABLE projekt_forma_zaliczenia ADD CONSTRAINT projekt_forma_zaliczenia_pk PRIMARY KEY (id_formy_zaliczenia);	
23		
24	CREATE TABLE projekt_grupa (
25	id_grupy NUMBER NOT NULL,	
26	nazwa_grupy VARCHAR2(50 CHAR) NOT NULL,	
27	id_kierunku NUMBER NOT NULL	
28);	
29		
30	ALTER TABLE projekt_grupa ADD CONSTRAINT grupa_pk PRIMARY KEY (id_grupy);	
31		
32	CREATE TABLE projekt_kierunek (
33	id_kierunku NUMBER NOT NULL,	
34	nazwa_kierunku VARCHAR2(50 CHAR) NOT NULL,	
35	liczba_miejsc NUMBER NOT NULL,	
36	id_wydzialu NUMBER NOT NULL	
37);	
38		
39	ALTER TABLE projekt_kierunek ADD CONSTRAINT kierunek_pk PRIMARY KEY (id_kierunku);	
40		
41	CREATE TABLE projekt_kierunki_zajecia (
42	nr_semestru NUMBER(1) NOT NULL,	
43	id_kierunku NUMBER NOT NULL,	
44	id_zajęc NUMBER NOT NULL	
45);	

```

46
47 ALTER TABLE projekt_kierunki_zajecia
48     ADD CONSTRAINT kierunki_zajecia_pk PRIMARY KEY ( nr_semestru,
49                                                         id_kierunku,
50                                                         id_zajęc );
51
52 CREATE TABLE projekt_pracownik (
53     id_pracownika    NUMBER NOT NULL,
54     imie              VARCHAR2(50 CHAR) NOT NULL,
55     nazwisko          VARCHAR2(50 CHAR) NOT NULL,
56     stopien_naukowy   VARCHAR2(50 CHAR) NOT NULL,
57     typ_pracy         VARCHAR2(50 CHAR) NOT NULL,
58     id_typu_pracy     NUMBER NOT NULL
59 );
60

```

```

61
62 ALTER TABLE projekt_pracownik ADD CONSTRAINT pracownik_pk PRIMARY KEY ( id_pracownika );
63
64 CREATE TABLE projekt_pracownik_stopien (
65     id_stopien        NUMBER NOT NULL,
66     id_pracownika     NUMBER NOT NULL
67 );
68
69 ALTER TABLE projekt_pracownik_stopien ADD CONSTRAINT projekt_pracownik_stopien_pk PRIMARY KEY ( id_stopien,
70                                                                                                    id_pracownika );
71
72 CREATE TABLE projekt_sala (
73     nr_sali           NUMBER NOT NULL,
74     typ_sali          VARCHAR2(50 CHAR) NOT NULL,
75     id_budynku        NUMBER NOT NULL,

```

```

76     id_typ_sali       NUMBER NOT NULL
77 );
78
79
80 ALTER TABLE projekt_sala ADD CONSTRAINT sala_pk PRIMARY KEY ( nr_sali );
81
82 CREATE TABLE projekt_semestr (
83     nr_semestru       NUMBER(1) NOT NULL,
84     nazwa_semestru    VARCHAR2(50 CHAR) NOT NULL
85 );
86
87 ALTER TABLE projekt_semestr ADD CONSTRAINT semestr_pk PRIMARY KEY ( nr_semestru );
88
89 CREATE TABLE projekt_stopien (
90     id_stopien        NUMBER NOT NULL,

```

```

91     nazwa      VARCHAR2(50 CHAR)
92 );
93
94 ALTER TABLE projekt_stopien ADD CONSTRAINT projekt_stopien_pk PRIMARY KEY ( id_stopien );
95
96 CREATE TABLE projekt_typ_pracy (
97     id_typu_pracy NUMBER NOT NULL,
98     nazwa          VARCHAR2(50 CHAR)
99 );
100
101 ALTER TABLE projekt_typ_pracy ADD CONSTRAINT projekt_typ_pracy_pk PRIMARY KEY ( id_typu_pracy );
102
103 CREATE TABLE projekt_typ_sali (
104     id_typ_sali NUMBER NOT NULL,
105     nazwa       VARCHAR2(50 CHAR)
106 );
107
108 ALTER TABLE projekt_typ_sali ADD CONSTRAINT projekt_typ_sali_pk PRIMARY KEY ( id_typ_sali );
109
110 CREATE TABLE projekt_wydział (
111     id_wydziału   NUMBER NOT NULL,
112     nazwa_wydziału VARCHAR2(50 CHAR) NOT NULL
113 );
114
115 ALTER TABLE projekt_wydział ADD CONSTRAINT wydział_pk PRIMARY KEY ( id_wydziału );
116
117 CREATE TABLE projekt_zajęcia (
118     id_zajęcie    NUMBER NOT NULL,
119     nazwa         VARCHAR2(50) NOT NULL,
120     punkty_ects   NUMBER NOT NULL,
121     godzina_rozpoczenia TIMESTAMP NOT NULL,
122     godzina_zakończenia TIMESTAMP NOT NULL,
123     nr_sali       NUMBER NOT NULL,
124     id_pracownika NUMBER NOT NULL,
125     id_grupy      NUMBER NOT NULL,
126     id_formy_zajęcie NUMBER NOT NULL,
127     id_formy_zaliczenia NUMBER NOT NULL
128 );
129
130
131
132 ALTER TABLE projekt_zajęcia ADD CONSTRAINT zajęcia_pk PRIMARY KEY ( id_zajęcie );
133 ALTER TABLE projekt_zajęcia ADD CONSTRAINT godziny CHECK((EXTRACT(HOUR FROM godzina_rozpoczenia)=8 AND EXTRACT(MINUTE FROM godzina_rozpoczenia)=0
134 AND EXTRACT(HOUR FROM godzina_zakończenia)=9 AND EXTRACT(MINUTE FROM godzina_zakończenia)=30)
135 OR (EXTRACT(HOUR FROM godzina_rozpoczenia)=9 AND EXTRACT(MINUTE FROM godzina_rozpoczenia)=45 AND EXTRACT(HOUR FROM godzina_zakończenia)=11
136 AND EXTRACT(MINUTE FROM godzina_zakończenia)=15)
137 OR (EXTRACT(HOUR FROM godzina_rozpoczenia)=11 AND EXTRACT(MINUTE FROM godzina_rozpoczenia)=45 AND EXTRACT(HOUR FROM godzina_zakończenia)=13
138 AND EXTRACT(MINUTE FROM godzina_zakończenia)=15)
139 OR (EXTRACT(HOUR FROM godzina_rozpoczenia)=13 AND EXTRACT(MINUTE FROM godzina_rozpoczenia)=30 AND EXTRACT(HOUR FROM godzina_zakończenia)=15
140 AND EXTRACT(MINUTE FROM godzina_zakończenia)=0)
141 OR (EXTRACT(HOUR FROM godzina_rozpoczenia)=15 AND EXTRACT(MINUTE FROM godzina_rozpoczenia)=30 AND EXTRACT(HOUR FROM godzina_zakończenia)=17
142 AND EXTRACT(MINUTE FROM godzina_zakończenia)=0)
143 OR (EXTRACT(HOUR FROM godzina_rozpoczenia)=17 AND EXTRACT(MINUTE FROM godzina_rozpoczenia)=15 AND EXTRACT(HOUR FROM godzina_zakończenia)=18
144 AND EXTRACT(MINUTE FROM godzina_zakończenia)=45)
145 OR (EXTRACT(HOUR FROM godzina_rozpoczenia)=19 AND EXTRACT(MINUTE FROM godzina_rozpoczenia)=30 AND EXTRACT(HOUR FROM godzina_zakończenia)=21
146 AND EXTRACT(MINUTE FROM godzina_zakończenia)=0));
147 ALTER TABLE projekt_zajęcia ADD CONSTRAINT pracownik_sala UNIQUE(godzina_rozpoczenia,godzina_zakończenia,NR_Sali,ID_Pracownika);
148
149 CREATE TABLE student (
150     nr_indeksu NUMBER NOT NULL,

```

```

151     imie      VARCHAR2(50 CHAR) NOT NULL,
152     nazwisko  VARCHAR2(50 CHAR) NOT NULL,
153     id_grupy  NUMBER NOT NULL
154 );
155
156 ALTER TABLE student ADD CONSTRAINT student_pk PRIMARY KEY ( nr_indeksu );
157 CREATE TABLE projekt_grupa_zajęcia(
158     id_zajęc NUMBER NOT NULL,
159     id_grupy NUMBER NOT NULL
160 );
161 ALTER TABLE projekt_grupa_zajęcia ADD CONSTRAINT projekt_grupa_zajęcia_pk PRIMARY KEY ( id_zajęc,
162                                                                                               id_grupy );
163 ALTER TABLE projekt_grupa_zajęcia
164     ADD CONSTRAINT grupa_zajęcia_zajęcia_fk FOREIGN KEY ( id_grupy )
165     REFERENCES projekt_grupa ( id_grupy );

```

```

166
167 ALTER TABLE projekt_grupa_zajęcia
168     ADD CONSTRAINT grupa_zajęcia_grupa_fk FOREIGN KEY ( id_zajęc )
169     REFERENCES projekt_zajęcia ( id_zajęc );
170
171
172 ALTER TABLE projekt_budynek
173     ADD CONSTRAINT budynek_wydział_fk FOREIGN KEY ( id_wydziału )
174     REFERENCES projekt_wydział ( id_wydziału );
175
176 ALTER TABLE projekt_grupa
177     ADD CONSTRAINT grupa_kierunek_fk FOREIGN KEY ( id_kierunku )
178     REFERENCES projekt_kierunek ( id_kierunku );
179
180 ALTER TABLE projekt_kierunek

```

```

166
167 ALTER TABLE projekt_grupa_zajęcia
168     ADD CONSTRAINT grupa_zajęcia_grupa_fk FOREIGN KEY ( id_zajęc )
169     REFERENCES projekt_zajęcia ( id_zajęc );
170
171
172 ALTER TABLE projekt_budynek
173     ADD CONSTRAINT budynek_wydział_fk FOREIGN KEY ( id_wydziału )
174     REFERENCES projekt_wydział ( id_wydziału );
175
176 ALTER TABLE projekt_grupa
177     ADD CONSTRAINT grupa_kierunek_fk FOREIGN KEY ( id_kierunku )
178     REFERENCES projekt_kierunek ( id_kierunku );
179
180 ALTER TABLE projekt_kierunek

```

```

181     ADD CONSTRAINT kierunek_wydział_fk FOREIGN KEY ( id_wydziału )
182     REFERENCES projekt_wydział ( id_wydziału );
183
184 ALTER TABLE projekt_kierunki_zajęcia
185     ADD CONSTRAINT kierunki_zajęcia_kierunek_fk FOREIGN KEY ( id_kierunku )
186     REFERENCES projekt_kierunek ( id_kierunku );
187
188 ALTER TABLE projekt_kierunki_zajęcia
189     ADD CONSTRAINT kierunki_zajęcia_semestr_fk FOREIGN KEY ( nr_semestru )
190     REFERENCES projekt_semestr ( nr_semestru );
191
192 ALTER TABLE projekt_kierunki_zajęcia
193     ADD CONSTRAINT kierunki_zajęcia_zajęcia_fk FOREIGN KEY ( id_zajęc )
194     REFERENCES projekt_zajęcia ( id_zajęc );
195

```



```

196 ALTER TABLE projekt_pracownik
197     ADD CONSTRAINT pracownik_typ_pracy_fk FOREIGN KEY ( id_typu_pracy )
198     REFERENCES projekt_typ_pracy ( id_typu_pracy );
199
200 ALTER TABLE projekt_pracownik_stopien
201     ADD CONSTRAINT ps_pracownik_fk FOREIGN KEY ( id_pracownika )
202     REFERENCES projekt_pracownik ( id_pracownika );
203
204 ALTER TABLE projekt_pracownik_stopien
205     ADD CONSTRAINT ps_projekt_stopien_fk FOREIGN KEY ( id_stopien )
206     REFERENCES projekt_stopien ( id_stopien );
207
208 ALTER TABLE projekt_sala
209     ADD CONSTRAINT sala_budynek_fk FOREIGN KEY ( id_budynku )
210     REFERENCES projekt_budynek ( id_budynku );

```

```

211
212 ALTER TABLE projekt_sala
213     ADD CONSTRAINT sala_typ_sali_fk FOREIGN KEY ( id_typ_sali )
214     REFERENCES projekt_typ_sali ( id_typ_sali );
215
216 ALTER TABLE student
217     ADD CONSTRAINT student_grupa_fk FOREIGN KEY ( id_grupy )
218     REFERENCES projekt_grupa ( id_grupy );
219
220 ALTER TABLE projekt_zajęcia
221     ADD CONSTRAINT zajęcia_fz_fk FOREIGN KEY ( id_formy_zajęć )
222     REFERENCES projekt_forma_zajęć ( id_formy_zajęć );
223
224 ALTER TABLE projekt_zajęcia
225     ADD CONSTRAINT zajęcia_fz_fkv2 FOREIGN KEY ( id_formy_zaliczenia )

```

```

226     REFERENCES projekt_forma_zaliczenia ( id_formy_zaliczenia );
227
228 ALTER TABLE projekt_zajęcia
229     ADD CONSTRAINT zajęcia_grupa_fk FOREIGN KEY ( id_grupy )
230     REFERENCES projekt_grupa ( id_grupy );
231
232 ALTER TABLE projekt_zajęcia
233     ADD CONSTRAINT zajęcia_pracownik_fk FOREIGN KEY ( id_pracownika )
234     REFERENCES projekt_pracownik ( id_pracownika );
235
236 ALTER TABLE projekt_zajęcia
237     ADD CONSTRAINT zajęcia_sala_fk FOREIGN KEY ( nr_sali )
238     REFERENCES projekt_sala ( nr_sali );
239
240 ALTER TABLE PROJEKT_GRUPA ADD NR_SEMESTRU NUMBER NOT NULL;

```

```

241 ALTER TABLE PROJEKT_GRUPA
242     ADD CONSTRAINT grupa_semestr_fk
243     FOREIGN KEY ( NR_SEMESTRU )
244     REFERENCES PROJEKT_SEMESTR ( NR_SEMESTRU );
245

```

WPROWADZILIŚMY DANE DO TABEL, A W KOLEJNYM KROKU WYKONALIŚMY ZAPYTANIA. Napisaliśmy sekwencję generującą kolejne numery indeksów dla studentów oraz wyzwalacz który przydzielał je studentom przy dodawaniu ich do tabeli Student:


```

246 CREATE SEQUENCE indeksy
247 MINVALUE 110902
248 MAXVALUE 999999
249 START WITH 110902
250 INCREMENT BY 1
251 CACHE 20;
252
253 CREATE OR REPLACE TRIGGER wyzwalacz_studenci
254 BEFORE INSERT ON STUDENT
255 FOR EACH ROW
256 BEGIN
257     SELECT indeksy.NEXTVAL INTO :NEW.NR_INDEKSU FROM dual;
258 END;

```

W związku z poprawkami wprowadzanymi w trakcie pracy nad bazą danych, indeks pierwszego studenta został ustawiony na 111022.

Przykładowe dane wprowadzone przez nas do bazy:

- Projekt_Wydział:

	ID_WYDZIALU	NAZWA_WYDZIALU
1	1	WYDZIAŁ INFORMATYKI

- Projekt_Budynek:

	ID_BUDYNKU	NAZWA_BUDYNKU	ID_WYDZIALU
1	1	BUDYNEK A	1
2	2	BUDYNEK B	1

- Projekt_Typ_Sali:

	ID_TYP_SALI	NAZWA
1	1	WYKŁADOWA
2	2	KOMPUTEROWA
3	3	ĆWICZENIOWA
4	4	LABORATORYJNA

- Projekt_Kierunek:

	ID_KIERUNKU	NAZWA_KIERUNKU	LICZBA_MIEJSC	ID_WYDZIALU
1	1	INFORMATYKA	200	1
2	2	INFORMATYKA I EKONOMETRIA	60	1
3	3	MATEMATYKA STOSOWANA	100	1

- Projekt_Kierunki_Zajęcia:

	NR_SEMESTRU	ID_KIERUNKU	ID_ZAJĘĆ
1	1	1	1
2	1	1	2
3	1	1	3
4	1	1	4
5	1	1	5
6	1	1	6
7	1	1	7
8	1	1	16
9	1	1	17
10	1	1	18
11	1	1	19
12	1	1	20
13	1	1	21

- Projekt_Semestr:

	NR_SEMESTRU	NAZWA_SEMESTRU
1	1	ZIMOWY
2	2	LETNI
3	3	ZIMOWY
4	4	LETNI
5	5	ZIMOWY
6	6	LETNI
7	7	ZIMOWY

- Projekt_Grupa:

	ID_GRUPY	NAZWA_GRUPY	ID_KIERUNKU	NR_SEMESTRU
1	1	INF PS1/1	1	1
2	2	INF PS2/1	1	1
3	3	INF PS3/1	1	1
4	4	INF PS4/1	1	1
5	5	INF PS5/1	1	1
6	6	INF PS1/2	1	2
7	7	INF PS2/2	1	2
8	8	INF PS3/2	1	2
9	9	INF PS4/2	1	2
10	10	INF PS5/2	1	2
11	11	INF PS1/3	1	3
12	12	INF PS2/3	1	3
13	13	INF PS3/3	1	3

- Student:

	NR_INDEKSU	IMIE	NAZWISKO	ID_GRUPY
1	111022	TYTUS	BOMBA	108
2	111023	KAMIL	ZDUN	1
3	111024	ANTONI	BUCZEK	2
4	111025	GREG	STIKIHENDS	201
5	111026	JAKUB	KOWALSKI	27
6	111027	ZUZANNA	NOWAK	14
7	111028	MICHAŁ	WIŚNIEWSKI	3
8	111029	JULIA	WÓJCIK	9
9	111030	MATEUSZ	KOWALCZYK	21
10	111031	LENA	KAMIŃSKA	32
11	111032	JAN	LEWANDOWSKI	6
12	111033	OLIWIA	DĄBROWSKA	17
13	111034	SZYMON	ZAJĄC	23

- Projekt_Grupa_Zajęcia:

	ID_ZAJĘĆ	ID_GRUPY
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	1
7	2	2
8	2	3
9	2	4
10	2	5
11	3	1
12	3	2
13	3	3

- Projekt_Zajęcia:

ID_ZAJĘĆ	NAZWA	PUNKTY_ECTS	GODZINA_ROZPOCZECIA	GODZINA_ZAKOŃCZENIA	NR_SALI	ID_PRACOWNIKA	ID_FORMY_ZAJĘĆ	ID_FORMY_ZALICZENIA
1	8 BAZY DANYCH	6	23/06/01 15:30:00,000000000	23/06/01 17:00:00,000000000	30	49	1	3
2	9 INŻYNIERIA OPROGRAMOWANIA	4	23/06/01 13:30:00,000000000	23/06/01 15:00:00,000000000	30	40	1	3
3	10 SZTUCZNA INTELIGENCJA	2	23/06/01 09:45:00,000000000	23/06/01 11:15:00,000000000	30	23	1	1
4	11 RACHUNKOWOŚĆ KOMPUTEROWA	5	23/06/01 17:15:00,000000000	23/06/01 18:45:00,000000000	30	15	1	3
5	12 STATYSTYKA I METODY PROBABILISTYCZNE	5	23/06/01 17:15:00,000000000	23/06/01 18:45:00,000000000	23	38	1	2
6	13 ALGORYTMY I STRUKTURY DANYCH	7	23/06/01 08:00:00,000000000	23/06/01 09:30:00,000000000	23	19	1	3
7	14 WSTĘP DO SŁUCHANIA MUZYKI POWAŻNEJ	3	23/06/01 08:00:00,000000000	23/06/01 09:30:00,000000000	24	30	1	1
8	1 PODSTAWY PROGRAMOWANIA	6	23/06/01 08:00:00,000000000	23/06/01 09:30:00,000000000	102	10	1	3
9	2 ALGEBRA LINIOWA	5	23/06/01 09:45:00,000000000	23/06/01 11:15:00,000000000	101	1	1	3
10	3 ANALIZA MATEMATYCZNA	5	23/06/01 11:45:00,000000000	23/06/01 13:15:00,000000000	102	7	1	3
11	4 FIZYKA	4	23/06/01 08:00:00,000000000	23/06/01 09:30:00,000000000	101	4	1	1
12	5 WPROWADZENIE DO LINUXA	4	23/06/01 08:00:00,000000000	23/06/01 09:30:00,000000000	1	3	1	1
13	6 WDI	3	23/06/01 09:45:00,000000000	23/06/01 11:15:00,000000000	102	2	1	1
14	7 LOGIKA	2	23/06/01 08:00:00,000000000	23/06/01 09:30:00,000000000	2	9	1	1
15	15 BEZPIECZEŃSTWO SIECI KOMPUTEROWYCH	2	23/06/01 19:30:00,000000000	23/06/01 21:00:00,000000000	24	12	1	1
16	16 PODSTAWY PROGRAMOWANIA	6	23/06/01 09:45:00,000000000	23/06/01 11:15:00,000000000	13	17	2	1
17	17 ALGEBRA LINIOWA	5	23/06/01 08:00:00,000000000	23/06/01 09:30:00,000000000	21	20	3	1
18	18 ANALIZA MATEMATYCZNA	5	23/06/01 13:30:00,000000000	23/06/01 15:00:00,000000000	21	45	3	1
19	19 FIZYKA	5	23/06/01 17:15:00,000000000	23/06/01 18:45:00,000000000	21	33	3	1

- Projekt_Forma_Zajęć:

	ID_FORMY_ZAJĘĆ	NAZWA
1	1	W
2	2	PS
3	3	ĆW
4	4	L
5	5	LEK
6	6	SEM

- Projekt_Forma_Zaliczenia:

	ID_FORMY_ZALICZENIA	NAZWA
1	1	ZALICZENIE PISEMNE
2	2	ODPOWIEDŹ USTNA
3	3	EGZAMIN
4	4	SPRAWOZDANIA

- Projekt_Pracownik:

	ID_PRACOWNIKA	IMIE	NAZWISKO	ID_TYPU_PRACY
1	1	ANDRZEJ	KMICIC	1
2	2	JAN	SKRZETUSKI	1
3	3	JURKO	BOHUN	2
4	4	MICHAŁ	WOŁODYJOWSKI	1
5	5	ONUFRY	ZAGŁOBA	2
6	6	LONGINUS	PODBIPIĘTA	1
7	7	IGA	ŚWIĄTEK	1
8	8	ANNA	SZIRLEJ	2
9	9	JOANNA	DARK	1
10	10	ELŻBIETA	KORONA	1
11	11	JAN	NOWAK	1
12	12	JAKUB	KOWALEWSKI	1

- Projekt_Typ_Pracy:

	ID_TYPU_PRACY	NAZWA	
1	1	ETATOWY	
2	2	NIEETATOWY	

- Projekt_Pracownik_Stopień:

	ID_STOPIEN	ID_PRACOWNIKA
1	1	1
2	1	2
3	1	5
4	1	6
5	1	8
6	1	9
7	1	11
8	1	14
9	1	16
10	1	24
11	1	25
12	1	29

- Projekt_Stopien:

	ID_STOPIEN	NAZWA
1	5	dr hab
2	1	inż
3	2	mgr
4	3	dr
5	4	prof

ABY SPRAWDZIĆ WYDAJNOŚĆ NASZEJ BAZY DANYCH NAPISALIŚMY KILKA PROCEDUR:

PROCEDURA WYPISUJĄCA WSZYSTKIE TYTUŁY NAUKOWE DANEGO PRACOWNIKA:

```
CREATE OR REPLACE PROCEDURE WypiszTytulyNaukowe(Identyfikator IN PROJEKT_PRACOWNIK.ID_PRACOWNIKA%TYPE) AS
CURSOR c_tytuly IS
    SELECT s.nazwa FROM PROJEKT_STOPIEN s,PROJEKT_PRACOWNIK_STOPIEN pps
WHERE s.ID_STOPIEN=pps.ID_STOPIEN AND pps.ID_PRACOWNIKA=Identyfikator;
    v_tytul PROJEKT_STOPIEN.nazwa%TYPE;
BEGIN
    OPEN c_tytuly;
    DBMS_OUTPUT.PUT_LINE('Tytuly:');
    LOOP
        FETCH c_tytuly INTO v_tytul;
        EXIT WHEN c_tytuly%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE( v_tytul);
    END LOOP;

    CLOSE c_tytuly;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nie znaleziono tytułów naukowych dla pracownika o ID: ' || Identyfikator);
END;
```

Procedura wypisująca informacje na temat danego pracownika, wykorzystująca poprzednią procedurę:

```
746 CREATE OR REPLACE PROCEDURE wyswietl_pracownik (Identyfikator IN PROJEKT_PRACOWNIK.ID_PRACOWNIKA%TYPE)
747 IS
748 imiep PROJEKT_PRACOWNIK.IMIE%TYPE;
749 surname PROJEKT_PRACOWNIK.NAZWISKO%TYPE;
750 zatrudnienie PROJEKT_TYP_PRACY.nazwa%TYPE;
751 BEGIN
752 SELECT imie,nazwisko,nazwa INTO imiep,surname,zatrudnienie
753 FROM PROJEKT_PRACOWNIK pp,PROJEKT_TYP_PRACY ptp
754 WHERE pp.ID_TYPU_PRACY=ptp.ID_TYPU_PRACY AND pp.ID_PRACOWNIKA=Identyfikator;
755 DBMS_OUTPUT.PUT_LINE('ID: ' || Identyfikator);
756 DBMS_OUTPUT.PUT_LINE('Imię: ' || imiep);
757 DBMS_OUTPUT.PUT_LINE('Nazwisko: ' || surname);
758 DBMS_OUTPUT.PUT_LINE('Typ pracy: ' || zatrudnienie);
759 WypiszTytulyNaukowe(Identyfikator);
760 EXCEPTION
761 WHEN NO_DATA_FOUND THEN
762 DBMS_OUTPUT.PUT_LINE('Nie znaleziono pracownika o ID: ' || Identyfikator);
763 END;
```


Procedura wypisująca informacje na temat danego studenta:

```
765 CREATE OR REPLACE PROCEDURE wyswietl_student (Indeks IN STUDENT.NR_INDEKSU%TYPE)
766 IS
767   imiep STUDENT.IMIE%TYPE;
768   surname STUDENT.NAZWISKO%TYPE;
769   grupa PROJEKT_GRUPA.NAZWA_GRUPY%TYPE;
770 BEGIN
771   SELECT imie,nazwisko,nazwa_grupy INTO imiep,surname,grupa
772   FROM STUDENT s,PROJEKT_GRUPA pg
773   WHERE s.ID_GRUPY=pg.ID_GRUPY AND s.NR_INDEKSU=Indeks;
774   DBMS_OUTPUT.PUT_LINE('Indeks: ' || Indeks);
775   DBMS_OUTPUT.PUT_LINE('Imię: ' || imiep);
776   DBMS_OUTPUT.PUT_LINE('Nazwisko: ' || surname);
777   DBMS_OUTPUT.PUT_LINE('Grupa: ' || grupa);
778
779 EXCEPTION
780   WHEN NO_DATA_FOUND THEN
781     DBMS_OUTPUT.PUT_LINE('Nie znaleziono studenta o Indeksie: ' || Indeks);
782 END;
```

```
dr hab
inż
prof

PL/SQL procedure successfully completed.

Indeks: 111076
Imię: ROBERT
Nazwisko: DULSKI
Grupa: IiE PS1/7

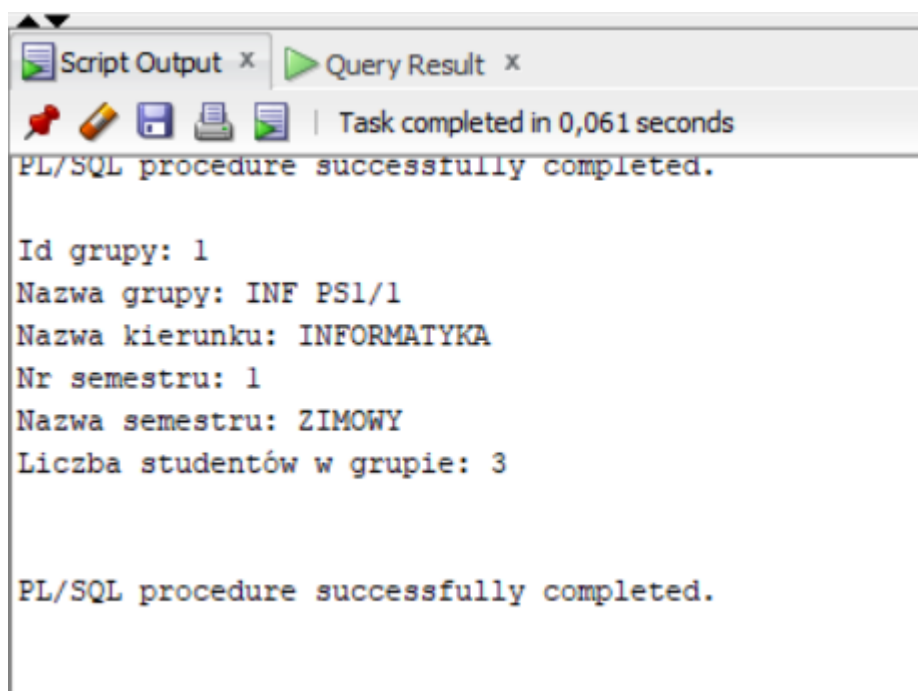
PL/SQL procedure successfully completed.
```

Procedura zliczająca studentów w danej grupie:

```
784 CREATE OR REPLACE PROCEDURE LiczStudentowWGrupie (Identyfikator PROJEKT_GRUPA.ID_GRUPY%TYPE) IS
785   v_liczba_studentow NUMBER;
786 BEGIN
787
788   SELECT COUNT(*) INTO v_liczba_studentow
789   FROM STUDENT
790   WHERE ID_GRUPY = Identyfikator;
791
792   DBMS_OUTPUT.PUT_LINE('Liczba studentów w grupie: ' || v_liczba_studentow);
793 EXCEPTION
794   WHEN NO_DATA_FOUND THEN
795     DBMS_OUTPUT.PUT_LINE('Nie znaleziono grupy o ID: ' || Identyfikator);
796 END;
```

Procedura wyświetlająca informacje o danej grupie:

```
799 CREATE OR REPLACE PROCEDURE wyswietl_grupe (Identyfikator IN PROJEKT_GRUPA.ID_GRUPY%TYPE)
800 IS
801   nazwag PROJEKT_GRUPA.NAZWA_GRUPY%TYPE;
802   nazwak PROJEKT_KIERUNEK.NAZWA_KIERUNKU%TYPE;
803   nr_sem PROJEKT_GRUPA.NR_SEMESTRU%TYPE;
804   nazwa_sem PROJEKT_SEMESTR.NAZWA_SEMESTRU%TYPE;
805   liczba NUMBER;
806 BEGIN
807   SELECT nazwa_grupy,nazwa_kierunku,pg.nr_semestru,nazwa_semestru INTO nazwag,nazwak,nr_sem,nazwa_sem
808   FROM PROJEKT_GRUPA pg,PROJEKT_KIERUNEK pk,PROJEKT_SEMESTR ps
809   WHERE pg.ID_KIERUNKU=pk.ID_KIERUNKU AND pg.ID_GRUPY=Identyfikator AND ps.NR_SEMESTRU=pg.NR_SEMESTRU;
810   DBMS_OUTPUT.PUT_LINE('Id grupy: ' || Identyfikator);
811   DBMS_OUTPUT.PUT_LINE('Nazwa grupy: ' || nazwag);
812   DBMS_OUTPUT.PUT_LINE('Nazwa kierunku: ' || nazwak);
813   DBMS_OUTPUT.PUT_LINE('Nr semestru: ' || nr_sem);
814   DBMS_OUTPUT.PUT_LINE('Nazwa semestru: ' || nazwa_sem);
815   liczstudentowwgrupie(Identyfikator);
816
817 EXCEPTION
818   WHEN NO_DATA_FOUND THEN
819     DBMS_OUTPUT.PUT_LINE('Nie znaleziono grupy o Id: ' || Identyfikator);
820 END;
```



```
Script Output x Query Result x
Task completed in 0,061 seconds
PL/SQL procedure successfully completed.

Id grupy: 1
Nazwa grupy: INF PS1/1
Nazwa kierunku: INFORMATYKA
Nr semestru: 1
Nazwa semestru: ZIMOWY
Liczba studentów w grupie: 3

PL/SQL procedure successfully completed.
```

Procedura wyświetlająca plan studiów dla danego semestru i kierunku:

```

824 CREATE OR REPLACE PROCEDURE plan_studiov (Nr_semestr IN PROJEKT_SEMESTR.NR_SEMESTRUWYTYPE, Kierunek IN PROJEKT_KIERUNEK.NAZWA_KIERUNKUWYTYPE)
825 IS
826 CURSOR c_przedmioty IS
827 SELECT z.nazwa as a, z.PUNKTY_ECTS, f.fz.NAZWA as c, TO_CHAR(z.GODZINA_ROZPOCZECIA, 'HH24:MI:SS') AS GODZINA_ROZ, TO_CHAR(z.GODZINA_ZAKOŃCZENIA, 'HH24:MI:SS') AS GODZINA_ZAK, z.NR_SALI, fz.NAZWA as b
828 FROM PROJEKT_ZAJĘCIA z, PROJEKT_KIERUNKI_ZAJĘCIA pki, PROJEKT_SEMESTR ps, PROJEKT_KIERUNEK pk, projekt_forma_zaliczenia fz, projekt_forma_zajęc f;
829 WHERE z.ID_ZAJĘĆ=pki.ID_ZAJĘĆ AND pki.ID_KIERUNKU=pk.ID_KIERUNKU AND pki.NR_SEMESTRU=ps.NR_SEMESTRU AND ps.NR_SEMESTRU=Nr_semestr AND pk.nazwa_kierunku=Kierunek
830 AND fz.ID_FORMY_ZALICZENIA=z.ID_FORMY_ZALICZENIA AND f.fz.ID_FORMY_ZAJĘĆ=z.ID_FORMY_ZAJĘĆ;
831
832 BEGIN
833 DBMS_OUTPUT.PUT_LINE('Kierunek: ' || Kierunek);
834 DBMS_OUTPUT.PUT_LINE('Semestr: ' || Nr_semestr);
835 FOR x IN c_przedmioty
836 LOOP
837 DBMS_OUTPUT.PUT_LINE('Przedmiot: ' || x.a || ' Forma zajęć: ' || x.c || ' ECTS: ' || x.PUNKTY_ECTS || ' START: ' || x.GODZINA_ROZ || ' KONIEC: ' || x.GODZINA_ZAK || ' SALA: ' || x.NR_SALI || ' FORMA ZALICZENIA: ' || x.b);
838 END LOOP;
839
840 EXCEPTION
841 WHEN NO_DATA_FOUND THEN
842 DBMS_OUTPUT.PUT_LINE('Nie znaleziono takiego semestru lub kierunku');
843 END;
844

```

Script Output x Query Result x	
Task completed in 0,088 seconds	
<p>Kierunek: INFORMATYKA</p> <p>Semestr: 1</p> <p>Przedmiot: LOGIKA Forma zajęć: W ECTS: 2 START: 08:00:00 KONIEC: 09:30:00 SALA: 2 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: WDI Forma zajęć: W ECTS: 3 START: 09:45:00 KONIEC: 11:15:00 SALA: 102 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: WPROWADZENIE DO LINUXA Forma zajęć: W ECTS: 4 START: 08:00:00 KONIEC: 09:30:00 SALA: 1 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: FIZYKA Forma zajęć: W ECTS: 4 START: 08:00:00 KONIEC: 09:30:00 SALA: 101 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: ANALIZA MATEMATYCZNA Forma zajęć: W ECTS: 5 START: 11:45:00 KONIEC: 13:15:00 SALA: 102 FORMA ZALICZENIA: EGZAMIN</p> <p>Przedmiot: ALGEBRA LINIOWA Forma zajęć: W ECTS: 5 START: 09:45:00 KONIEC: 11:15:00 SALA: 101 FORMA ZALICZENIA: EGZAMIN</p> <p>Przedmiot: PODSTAWY PROGRAMOWANIA Forma zajęć: W ECTS: 6 START: 08:00:00 KONIEC: 09:30:00 SALA: 102 FORMA ZALICZENIA: EGZAMIN</p> <p>Przedmiot: PODSTAWY PROGRAMOWANIA Forma zajęć: PS ECTS: 6 START: 09:45:00 KONIEC: 11:15:00 SALA: 13 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: LOGIKA Forma zajęć: ĆW ECTS: 2 START: 13:30:00 KONIEC: 15:00:00 SALA: 125 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: WDI Forma zajęć: ĆW ECTS: 3 START: 11:45:00 KONIEC: 13:15:00 SALA: 125 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: FIZYKA Forma zajęć: ĆW ECTS: 5 START: 17:15:00 KONIEC: 18:45:00 SALA: 21 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: ANALIZA MATEMATYCZNA Forma zajęć: ĆW ECTS: 5 START: 13:30:00 KONIEC: 15:00:00 SALA: 21 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: ALGEBRA LINIOWA Forma zajęć: ĆW ECTS: 5 START: 08:00:00 KONIEC: 09:30:00 SALA: 21 FORMA ZALICZENIA: ZALICZENIE PISEMNE</p> <p>Przedmiot: WPROWADZENIE DO LINUXA Forma zajęć: L ECTS: 4 START: 09:45:00 KONIEC: 11:15:00 SALA: 29 FORMA ZALICZENIA: SPRAWOZDANIA</p> <p>PL/SQL procedure successfully completed.</p>	

Wszystkie procedury wykonywały się dosyć szybko, postanowiliśmy dokładniej zbadać wydajność naszej bazy używając zapytania zwracającego wszystkie grupy, które mają przedmiot o nazwie “Podstawy Programowania”:

```

719 | SELECT DISTINCT g.nazwa_grupy, z.nazwa FROM PROJEKT_ZAJĘCIA z, PROJEKT_GRUPA_ZAJĘCIA gz, PROJEKT_GRUPA g
720 | WHERE g.ID_GRUPY=gz.ID_GRUPY AND gz.ID_ZAJĘĆ=z.ID_ZAJĘĆ AND z.nazwa='PODSTAWY PROGRAMOWANIA';

```

Script Output x Query Result x Explain Plan x		
All Rows Fetched: 9 in 0,048 seconds		
	NAZWA_GRUPY	NAZWA
1	INF PS4/1	PODSTAWY PROGRAMOWANIA
2	IiE PS1/1	PODSTAWY PROGRAMOWANIA
3	MS PS1/1	PODSTAWY PROGRAMOWANIA
4	INF PS1/1	PODSTAWY PROGRAMOWANIA
5	IiE PS2/1	PODSTAWY PROGRAMOWANIA
6	INF PS2/1	PODSTAWY PROGRAMOWANIA
7	INF PS5/1	PODSTAWY PROGRAMOWANIA
8	MS PS2/1	PODSTAWY PROGRAMOWANIA
9	INF PS3/1	PODSTAWY PROGRAMOWANIA

OBJECT_NAME	OPTIONS	CARDINALITY	COST
		8	7
	UNIQUE	8	7
		8	6
	CARTESIAN	63	6
PROJEKT_ZAJĘCIA	FULL	1	3
PROGRAMOWANIA			
	SORT	63	3
PROJEKT_GRUPA	FULL	63	3
PROJEKT_GRUPA_ZAJĘCIA_PK	UNIQUE SCAN	1	0

Dodaliśmy następujące indeksy żeby zwiększyć wydajność tego zapytania:

```

721 CREATE INDEX idx_projekt_grupa_id_grupy ON PROJEKT_GRUPA (ID_GRUPY);
722 CREATE INDEX idx_grupa_zajecia_id_grupy ON PROJEKT_GRUPA_ZAJĘCIA (ID_GRUPY);
723 CREATE INDEX idx_zajecia_id_zajec ON PROJEKT_ZAJĘCIA (ID_ZAJĘĆ);
724 CREATE INDEX idx_zajecia_nazwa ON PROJEKT_ZAJĘCIA (nazwa);

```

Script Output	Explain Plan	Query Result
All Rows Fetched: 9 in 0,022 seconds		
NAZWA_GRUPY	NAZWA	
1 INF PS4/1	PODSTAWY PROGRAMOWANIA	
2 IIE PS1/1	PODSTAWY PROGRAMOWANIA	
3 MS PS1/1	PODSTAWY PROGRAMOWANIA	
4 INF PS1/1	PODSTAWY PROGRAMOWANIA	
5 IIE PS2/1	PODSTAWY PROGRAMOWANIA	
6 INF PS2/1	PODSTAWY PROGRAMOWANIA	
7 INF PS5/1	PODSTAWY PROGRAMOWANIA	
8 MS PS2/1	PODSTAWY PROGRAMOWANIA	
9 INF PS3/1	PODSTAWY PROGRAMOWANIA	

Czas wykonania zapytania się zmniejszył.

OBJECT_NAME	OPTIONS	CARDINALITY	COST
		8	6
	UNIQUE	8	6
		8	5
	CARTESIAN	63	5
PROJEKT_ZAJĘCIA	BY INDEX ROWID	1	2
IDZ_ZAJĘCIA_NAZWA	RANGE SCAN	1	1
PROGRAMOWANIA			
	SORT	63	3
PROJEKT_GRUPA	FULL	63	3
PROJEKT_GRUPA_ZAJĘCIA_PK	UNIQUE SCAN	1	0

Zmniejszył się również koszt wykonania zapytania. Jednak były to zmiany nieznaczne. Możemy więc wywnioskować, że w naszej bazie danych wprowadzanie indeksów zwiększyłoby wydajność wykonywania zapytań, ale nie były by to znaczące zmiany, zatem ze względu na rozmiar naszej bazy, oraz jej strukturę nie jest to opłacalne w ostatecznym rozrachunku.