

# Sprawozdanie BD1

## Laboratorium nr 1 – SQL1

Mateusz Dylong  
Informatyka, gr. 3

1. Podać nazwę tematu o numerze 5

```
SELECT temat FROM tematy WHERE nr_tem = 5
```

2. Podać nazwiska pracowników zaczynające się od litery K

```
SELECT nazwisko FROM pracownicy WHERE nazwisko LIKE 'K%'
```

3. Podać nazwy tematów, których kierownikiem jest pracownik MISIURA

```
SELECT temat FROM tematy JOIN pracownicy ON tematy.nr_prac_kt = pracownicy.nr_prac AND  
pracownicy.nazwisko LIKE 'Misiura'
```

4. Podać daty wypłat, które otrzymały kobiety urodzone przed 1970 rokiem według daty wypłaty.

```
SELECT data_wypł FROM wypłaty JOIN pracownicy ON wypłaty.nr_prac = pracownicy.nr_prac  
AND pracownicy.plec LIKE 'K' AND pracownicy.data_ur < '1970-01-01' ORDER BY data_wypł  
ASC
```

5. Podać wszystkie funkcje, które pełni pracownik MISIURA.

```
SELECT DISTINCT nazwa_funkcji FROM funkcje JOIN przydzialy ON funkcje.kod_funkcji =  
przydzialy.kod_funkcji JOIN pracownicy ON przydzialy.nr_prac = pracownicy.nr_prac AND  
pracownicy.nazwisko LIKE 'Misiura'
```

6. Podać w odwrotnej kolejności alfabetycznej nazwiska pracowników mężczyzn, którzy przydzieleni są do tematu DYSK lub tematu GUPTA.

```
SELECT DISTINCT nazwisko FROM pracownicy JOIN przydzialy ON pracownicy.nr_prac =  
przydzialy.nr_prac JOIN tematy ON przydzialy.nr_tem = tematy.nr_tem WHERE tematy.temat  
LIKE 'Dysk' OR tematy.temat LIKE 'Gupta'
```

7. Podać wszystkie funkcje pełnione przez pracowników w tematach, które rozpoczęły się i zostały odebrane przed rokiem 1990. Każda funkcja powinna pojawić się tylko raz

```
SELECT DISTINCT nazwa_funkcji FROM funkcje JOIN przydzialy ON funkcje.kod_funkcji =  
przydzialy.kod_funkcji JOIN tematy ON przydzialy.nr_tem = tematy.nr_tem WHERE  
tematy.data_rozp < '1990-01-01' AND tematy.data_odb < '1990-01-01'
```

8. Podać nazwy tematów i nazwy ich tematów nadrzędnych.

```
SELECT A.temat, B.temat FROM tematy A LEFT JOIN tematy B ON A.nr_tem = B.nr_tem_nadrz
```

9. Podać ile przydziałów ma pracownik MISIURA.

```
SELECT COUNT(przydzialy.nr_prac) FROM przydzialy JOIN pracownicy ON przydzialy.nr_prac =  
pracownicy.nr_prac WHERE pracownicy.nazwisko LIKE 'Misiura'
```

10. Podać sumę wszystkich wypłat w temacie DYSK.

```
SELECT SUM(kwota) FROM wypłaty JOIN tematy ON wypłaty.nr_tem = tematy.nr_tem WHERE  
tematy.temat LIKE 'Dysk'
```

11. Podać średnią wypłatę wszystkich pracowników będących kierownikami tematów.

```
SELECT AVG(kwota) FROM wypłaty JOIN pracownicy ON wypłaty.nr_prac = pracownicy.nr_prac  
JOIN tematy ON tematy.nr_prac_kt = pracownicy.nr_prac
```

12. Podać ilość i sumę wypłat w poszczególnych tematach.

```
SELECT temat, COUNT(kwota), SUM(kwota) FROM wypłaty JOIN tematy ON wypłaty.nr_tem =  
tematy.nr_tem GROUP BY temat
```

13. Podać średnie wypłaty dla każdego pracownika.

```
SELECT nazwisko, AVG(kwota) FROM wypłaty JOIN pracownicy ON wypłaty.nr_prac =  
pracownicy.nr_prac GROUP BY nazwisko
```

14. Podać daty urodzenia najstarszych pracowników przydzielonych do każdej z funkcji.

```
SELECT nazwa_funkcji, MIN(data_ur) FROM pracownicy JOIN przydziały ON pracownicy.nr_prac  
= przydziały.nr_prac JOIN funkcje ON funkcje.kod_funkcji = przydziały.kod_funkcji GROUP  
BY nazwa_funkcji
```

15. Podać wszystkie tematy w których są przydziały do funkcji Tester i ilość tych przydziałów.

```
SELECT temat, COUNT(temat) FROM tematy JOIN przydziały ON tematy.nr_tem =  
przydziały.nr_tem JOIN funkcje ON przydziały.kod_funkcji = funkcje.kod_funkcji WHERE  
nazwa_funkcji LIKE 'Tester' GROUP BY temat
```

16. Podać wszystkie tematy dla których suma wypłat jest mniejsza od 400.

```
SELECT temat, SUM(kwota) FROM tematy JOIN wypłaty ON tematy.nr_tem = wypłaty.nr_tem  
GROUP BY temat HAVING SUM(kwota) < 400
```

17. Podać nazwiska wszystkich pracowników, którzy są przydzieleni do więcej niż 5 różnych tematów.

```
SELECT nazwisko, COUNT(DISTINCT temat) FROM pracownicy JOIN przydziały ON  
pracownicy.nr_prac = przydziały.nr_prac JOIN tematy ON przydziały.nr_tem = tematy.nr_tem  
GROUP BY nazwisko HAVING COUNT(DISTINCT temat) > 5
```

18. Podać tematy, których data rozpoczęcia jest wcześniejsza niż data rozpoczęcia tematu DYSK.

```
SELECT A.temat, A.data_rozp FROM tematy A, tematy B WHERE A.data_rozp < B.data_rozp AND  
B.temat LIKE 'Dysk'
```

19. Podać nazwiska pracowników, których suma wypłat jest większa niż suma wypłat pracownika MISIURY.

```
SELECT nazwisko FROM pracownicy JOIN wypłaty ON pracownicy.nr_prac = wypłaty.nr_prac  
GROUP BY nazwisko HAVING SUM(kwota) > (SELECT SUM(kwota) FROM pracownicy JOIN wypłaty ON  
pracownicy.nr_prac = wypłaty.nr_prac WHERE nazwisko LIKE 'Misiura')
```

20. Podać tematy do których przydzielono więcej pracowników niż do tematu DYSK.

```
SELECT temat, COUNT(DISTINCT nr_prac) FROM tematy JOIN przydziały ON tematy.nr_tem =  
przydziały.nr_tem GROUP BY temat HAVING COUNT(DISTINCT nr_prac) > (SELECT COUNT(DISTINCT  
nr_prac) FROM przydziały JOIN tematy ON przydziały.nr_tem = tematy.nr_tem WHERE temat  
LIKE 'Dysk')
```

# Sprawozdanie BD1

## Laboratorium nr 2 – SQL2

Mateusz Dylong  
Informatyka, gr. 3

1. Dla każdego pracownika odpowiedzialnego za jakiś przedmiot podaj liczbę przedmiotów z których prowadzi zajęcia (tabela ROZKLADY).

```
SELECT nazwisko, COUNT(DISTINCT nr_przedm)
FROM pracownicy p, rozklady r
WHERE p.nr_prac=r.nr_prac AND r.nr_prac IN
(SELECT DISTINCT nr_odp_prac
FROM przedmioty)
GROUP BY nazwisko;
```

2. Podaj nazwiska i średnią ocen z przedmiotu ALGEBRA tych studentów, którzy z tego przedmiotu mają więcej niż jedną ocenę.

```
SELECT nazwisko, AVG(ocena)
FROM studenci s, oceny o, przedmioty p
WHERE s.nr_stud=o.nr_stud AND nazwa_przedm='ALGEBRA' AND o.nr_przedm=p.nr_przedm
GROUP BY nazwisko
HAVING COUNT(ocena)>1;
```

3. Podaj przedmioty realizowane przez największą liczbę pracowników.

```
SELECT nazwa_przedm, COUNT(DISTINCT nr_prac)
FROM przedmioty p, rozklady r
WHERE p.nr_przedm=r.nr_przedm
GROUP BY nazwa_przedm
ORDER BY COUNT(DISTINCT nr_prac) DESC
LIMIT 1;
```

4. Podaj numery i nazwiska pracowników, którzy w temacie DYSK otrzymali wypłatę mniejszą od co najmniej jednej z wypłat pracownika o nazwisku GRZYBEK w dowolnym temacie.

```
SELECT p1.nr_prac, nazwisko
FROM pracownicy p1, wypłaty w1, tematy t
WHERE p1.nr_prac=w1.nr_prac AND w1.nr_tem=t.nr_tem AND t.temat='DYSK'
AND kwota < ANY
(SELECT kwota
FROM wypłaty w2, pracownicy p2
WHERE w2.nr_prac=p2.nr_prac AND nazwisko='GRZYBEK')
GROUP BY nr_prac, nazwisko;
```

5. Podaj numery, nazwiska i daty urodzenia mężczyzn zatrudnionych w zespole OPROGRAMOWANIE, którzy są młodszy od każdego pracownika zespołu BUDOWA.

```
SELECT p1.nr_prac, p1.nazwisko, p1.data_ur
FROM pracownicy p1 JOIN zespoly z1 ON p1.nr_zesp=z1.nr_zesp
WHERE plec='M' AND nazwa_zesp='OPROGRAMOWANIE' AND data_ur > ALL
(SELECT data_ur
FROM pracownicy p2 JOIN zespoly z2 ON p2.nr_zesp=z2.nr_zesp
WHERE nazwa_zesp='BUDOWA');
```

6. Podaj nazwiska najstarszych studentów na poszczególnych kierunkach.

```
SELECT DISTINCT nazwa_kier, nazwisko
FROM studenci s, kierunki k
WHERE s.nr_kier=k.nr_kier AND
data_ur=(SELECT MIN(data_ur)
FROM studenci s2
WHERE s2.nr_kier=s.nr_kier);
```

7. Podaj nazwy zespołów, w których nie pracuje żadna kobieta.

```
SELECT nazwa_zesp
FROM zespoly z
WHERE nr_zesp NOT IN
(SELECT DISTINCT nr_zesp
FROM pracownicy
WHERE plec='K');
```

8. Zestawienie zespołów z liczbą pracowników o postaci: Nrz nazwa\_zespołu liczebność

W zestawieniu powinny znaleźć się również zespoły z liczebnością zerową.

```
SELECT z.nr_zesp AS 'Nrz', z.nazwa_zesp AS 'nazwa_zespołu',
COUNT(p.nr_prac) AS 'liczebność'
FROM zespoly z LEFT OUTER JOIN pracownicy p ON z.nr_zesp=p.nr_zesp
GROUP BY z.nr_zesp, z.nazwa_zesp;
```

9. Utworzyć perspektywę zawierającą wyszczególnione informacje, oraz określić, czy można aktualizować dane w tej perspektywie. Jeżeli nie można - podać wszystkie tego przyczyny. Po pomyślnym wykonaniu zadania należy usunąć utworzoną przez siebie perspektywę.

Zestawienie pracowników mających wypłaty w poszczególnych tematach w postaci:

nr\_tem nazwa\_tematu nr\_prac nazwisko

Rekordy nie powinny powtarzać się nawet w sytuacji, gdy pracownik uzyskał wielokrotnie dochód w tym samym temacie.

```
CREATE VIEW zestawienie_prac
AS (SELECT DISTINCT w.nr_tem AS 'nr_tem', temat AS 'nazwa_tematu',
p.nr_prac AS 'nr_prac', nazwisko AS 'nazwisko'
FROM pracownicy p, wypłaty w, tematy t
WHERE p.nr_prac=w.nr_prac AND w.nr_tem=t.nr_tem);
```

```
SELECT * FROM zestawienie_prac;  
DROP VIEW zestawienie_prac;
```

Dane można aktualizować. Użytkownik administracyjny posiada wszelkie prawa.

10. Zrealizować zestawienie średnich, minimalnych i maksymalnych zarobków pracowników w poszczególnych zespołach.

**Uwaga!** Zestawienie zrealizować w dwóch krokach:

w pierwszym utworzyć perspektywę (widok) o postaci:

nrz nazwa\_zesp nr\_prac nazwisko suma\_zarobków\_prac

uwzględnić również fakt, że pracownik mógł nie dostać żadnej wypłaty,  
utworzoną perspektywę wykorzystać w zadaniu.

```
CREATE VIEW zarobkip  
AS (SELECT z.nr_zesp AS 'nrz', z.nazwa_zesp AS 'nazwa_zesp', p.nr_prac  
AS 'nr_prac', p.nazwisko AS 'nazwisko',  
SUM(kwota) AS 'suma_zarobków_prac'  
FROM (zespoly z JOIN pracownicy p ON z.nr_zesp=p.nr_zesp) LEFT  
OUTER JOIN wypłaty w ON p.nr_prac=w.nr_prac  
GROUP BY z.nr_zesp, z.nazwa_zesp, p.nr_prac, p.nazwisko);
```

```
SELECT nrz, nazwa_zesp, AVG(suma_zarobków_prac),  
MIN(suma_zarobków_prac), MAX(suma_zarobków_prac)  
FROM zarobkip  
GROUP BY nrz, nazwa_zesp;
```

```
DROP VIEW zarobkip;
```

11. Podaj nazwiska pracowników, którzy zostali przydzieleni do tematu, lecz nie otrzymali w tym temacie ani jednej wypłaty.

```
SELECT DISTINCT nazwisko  
FROM (pracownicy p JOIN przydziały przy ON p.nr_prac=przy.nr_prac)  
LEFT OUTER JOIN wypłaty w ON  
przy.nr_prac=w.nr_prac AND przy.nr_tem=w.nr_tem  
WHERE kwota IS NULL;
```

12. Sprawdzić poprawność przydziału kierowników do tematu (jest to niezależnie określone za pomocą klucza obcego nr\_prac\_kt w tabeli tematY oraz przez zapis w tabeli PRZYDZIAŁY – należy sprawdzić czy oba te zapisy sobie odpowiadają).

```
SELECT *  
FROM tematy t LEFT OUTER JOIN przydziały p ON t.nr_tem=p.nr_tem AND  
t.nr_prac_kt=p.nr_prac  
WHERE kod_funkcji IS NULL;
```

13. Podaj numery i nazwiska pracowników nie biorących udziału w żadnym z tematów, w którym pracuje pracownik NIEZALEZNY.

```
SELECT DISTINCT p1.nr_prac, p1.nazwisko
FROM pracownicy p1
WHERE NOT EXISTS
(SELECT *
FROM przydzialy przy1
WHERE p1.nr_prac=przy1.nr_prac AND EXISTS
(SELECT *
FROM pracownicy p2, przydzialy przy2
WHERE p2.nr_prac=przy2.nr_prac AND p2.nazwisko='NIEZALEZNY' AND
przy1.nr_tem=przy2.nr_tem));
```

14. Podaj nazwiska pracowników którzy realizowali wszystkie tematy, których kierownikiem jest JASKOLA.

```
SELECT nazwisko
FROM pracownicy p, przydzialy przy
WHERE p.nr_prac=przy.nr_prac AND przy.nr_tem IN
(SELECT t2.nr_tem
FROM pracownicy p2 JOIN tematy t2 ON p2.nr_prac=t2.nr_prac_kt
WHERE p2.nazwisko='JASKOLA')
GROUP BY nazwisko
HAVING COUNT(DISTINCT przy.nr_tem)=(SELECT COUNT(DISTINCT t3.nr_tem)
FROM pracownicy p3 JOIN tematy t3 ON p3.nr_prac=t3.nr_prac_kt
WHERE p3.nazwisko='JASKOLA');
```