

# ACCESS- zadania z wykorzystaniem poleceń SQL

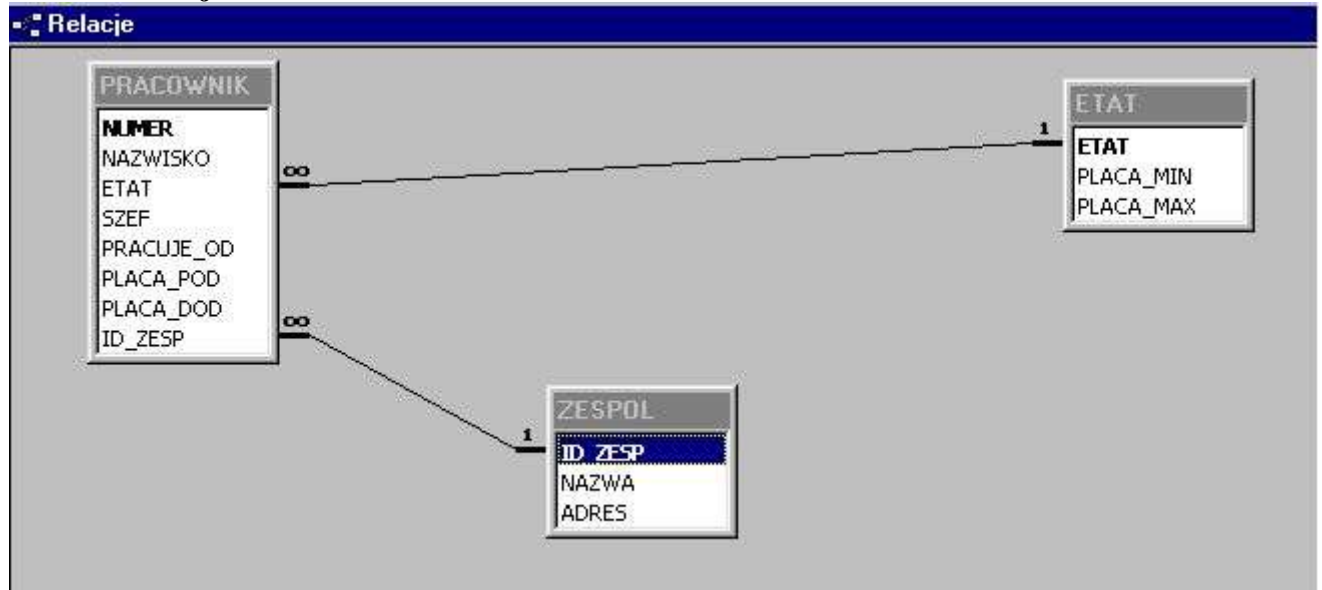
Dane są relacje o schematach:

**Pracownik** ( (nr integer, nazwisko text(12) , etat text(10), szef integer, pracuje\_od date, placa\_pod Currency, placa\_dod Currency, id\_zesp integer, constraint klucz primary key (nr) );

**Zespól** (id\_zesp integer, nazwa text(20), adres text(13))

**Etat** (etat text(10, placa\_min single, placa\_max currency)

Schemat relacji



**Zadania:**

Skopiować i rozpakować plik [Prac\\_dane.zip](#) ,  
skopiować plik pod nazwą Prac\_Nazwisko.mdb.

Zmodyfikować szerokości pól tekstowych: nazwisko na 20, etat na 15.

Sprawdzić i ewentualnie zmodyfikować powiazania między tabelami

Otworzyć tabele, dopisać kilka pozycji

Założyć formularze różnego typu (kreator, autoformularz).

Utworzyć proste kwerendy z wykorzystaniem kreatora a następnie dodać tabele do tych kweend

**Wykonać zadania zgodnie z poniższym opisem, wykorzystując kwerendy i SQL**

## I. Zapytania tworzące i modyfikujące dane

### 1. Tworzenie tabeli - komenda CREATE i SELECT FROM

Utworzenie tabeli **Pracownik1**(nr integer, nazwisko text(12),  
etat text(10), szef integer, pracuje\_od date, placa\_pod Currency, placa\_dod Currency,  
id\_zesp integer,

z kluczem głównym nr

```
create table Pracownik1  
(nr integer,  
nazwisko text(12),  
etat text(10),  
szef integer,  
pracuje_od date,  
placa_pod Currency,  
placa_dod Currency,  
id_zesp integer,  
constraint klucz primary key (nr)  
);
```

Utworzenie tabeli **etat2**

```
CREATE TABLE etat2  
( etat text(10) ,  
placa_min single,  
placa_max double  
);
```

Utworzenie tabeli **Pracownicy**(nr integer, nazwisko text(30), stanowisko text(20), data\_zatr date)  
z kluczem głównym nr

```
CREATE TABLE Pracownicy  
(nr integer ,  
nazwisko text(30) ,  
stanowisko text(20),  
data_zatr date,  
constraint klucz primary key (nr)  
);
```

Utworzenie tabeli **Worker**

```
create table worker  
(nr integer,  
nazwisko text,  
constraint indeks  
foreign key (nr) references Pracownicy(Nr));
```

Utworzenie nowej tabeli **Etat40** z wprowadzeniem danych z tabeli źródłowej **Etat**

```
SELECT * INTO ETAT40  
FROM ETAT;
```

```
SELECT * INTO ETAT10  
FROM ETAT  
where etat like "a*";
```

```
SELECT ETAT, PLACA_MIN INTO ETAT66  
FROM ETAT
```

**WHERE** placa\_max >=1600;

## 2. Dołączenie danych: INSERT, SELECT INTO

Wprowadzenie do wyżej utworzonej tabeli Pracownicy danych z tabeli Pracownik dla pól

o etatach: adiunkt, asystent, profesor

```
INSERT INTO Pracownicy ( nr, nazwisko, stanowisko, data_zatr )  
SELECT numer, nazwisko, etat, pracuje_od  
FROM pracownik  
WHERE etat IN ("ADIUNKT", "ASYSTENT", "Profesor");
```

Wprowadzenie do tabeli Etat1 wszystkich etatów z tabeli Etat na literę s

```
INSERT INTO ETAT1  
SELECT *  
FROM ETAT  
WHERE etat LIKE "s*";
```

Wprowadzenie 2 pól z tabeli Pracownik do tabeli Dodatki

```
SELECT Numer, Placa_dod INTO Dodatki  
FROM Pracownik;
```

Wprowadzenie do tabeli Pracownicy jednego pracownika (Mleczo) o podanych wartościach

```
INSERT INTO Pracownicy ( nr, nazwisko, stanowisko, data_zatr )  
VALUES ("3001", "Mleczo", "portier", "1997-01-01");  
lub  
INSERT INTO Pracownicy ( nr, nazwisko, stanowisko, data_zatr )  
SELECT "3001", "Mleczo", "portier", "1997-01-01";
```

## 3. Modyfikacja struktury tabeli istniejącej - ALTER TABLE

Dodanie kolumny (pola) PlacaPod typu liczba podwójnej precyzji

```
Alter Table Pracownicy  
ADD COLUMN PlacaPod float;
```

## 4. Kasowanie danych - Delete

Skasować z tabeli Pracownicy pracowników o nazwisku zaczynającym się na Ml (np. Mleczo, Mlaska)

```
DELETE *  
FROM Pracownicy
```

**WHERE** nazwisko **like** "MI\*";

Usunięcie wszystkich danych z tabeli Personel

```
DELETE *  
FROM Personel;
```

## 5. Aktualizacja danych - UPDATE

Wstawić wszystkim pracownikom płacę 700.25

```
UPDATE Pracownicy SET PlacaPod = 700.25;
```

Zmodyfikować płacę adiunktom i profesorom

```
UPDATE Pracownicy SET Pracownicy.PlacaPod = PlacaPod*1.25  
WHERE (stanowisko="Profesor" Or stanowisko="Adiunkt") AND Year(Data_zatr)<1980;
```

Ustawienie płacy dod. na 0 dla pola Numer równego 1050

```
UPDATE Pracownik SET PLACA_DOD = 0  
WHERE Numer=1050;
```

## 6. DROP

Usunięcie całej tabeli

```
DROP TABLE [Wykładowcy1];
```

Usunięcie pola

```
ALTER TABLE Etat4  
  DROP COLUMN Salary;
```

## II. ZAPYTANIA WYBIERAJĄCE - przykłady

```
SELECT nazwisko, imię, dział  
FROM personel  
WHERE Stanowisko = 'Kierownik';
```

```
SELECT *  
FROM pracownik  
WHERE placa_pod > 1500 AND (etat='ADIUNKT' OR etat = 'ASYSTENT');
```

```
SELECT nazwisko, etat, placa_pod  
FROM pracownik  
WHERE etat='ADIUNKT' or placa_pod between 1000 and 2000;
```

```
SELECT [Imię] AS [First Name (Imię)], [Nazwisko] AS [Last Name - Nazwisko],  
  Year(Date())- Year([urodzony]) AS Wiek  
FROM Personel AS [Staff - Personel];
```

```
SELECT [Imię], [Nazwisko], Year(Date()-Year([Urodzony]) AS Wiek
FROM Personel
WHERE Right([Imię],1)="a";
```

### Porządkowanie - ORDER BY

```
SELECT *
FROM pracownik
ORDER BY id_zesp, placa_pod DESC;
```

```
SELECT personel.imię, nazwisko, year(date()-year([urodzony]) AS Wiek
FROM Personel
ORDER BY year(date()-year([urodzony]) DESC , [Nazwisko];
```

```
SELECT Nazwisko, Imię
FROM Personel
WHERE [Dane żony/męża] is not null;
```

```
SELECT TOP 3 *
FROM Personel
ORDER BY [Zatrudniony od];
```

```
SELECT nazwisko, imię
FROM personel
WHERE nazwisko not like "D*";
```

```
SELECT nazwisko, imię
FROM personel
WHERE nazwisko <> "Dąbek";
```

```
SELECT Nazwisko, Imię
FROM Personel
WHERE nazwisko >= 'K%';
```

```
SELECT nazwisko, imię
FROM Personel
WHERE nazwisko not in ('Barska','[Śliwa]);
```

### Parametry

```
PARAMETERS [Podaj nazwisko:] Text;
SELECT *
FROM Personel
WHERE [Nazwisko]=[Podaj nazwisko:];
```

```
ALL, DISTINCT
SELECT [Zamieszkały]
FROM Personel
WHERE [Zamieszkały] <> "Wrocław";
```

```
SELECT DISTINCT [Zamieszkały]
FROM Personel
WHERE [Zamieszkały] <> "Wrocław";
```

## **GROUP BY...HAVING**

```
SELECT etat, avg(placa_pod)
FROM Pracownik
WHERE etat <> 'DYREKTOR'
GROUP BY etat;
```

```
SELECT id_zesp, etat, ccur(avg(placa_pod+placa_dod)) AS [Płaca średnia]
FROM Pracownik
GROUP BY id_zesp, etat;
```

```
PARAMETERS [(K)obieta czy (M)ężczyzna?] Text;
SELECT [Dział], [Płeć], Count([Nazwisko]) AS [Ilość zatrudnionych]
FROM Personel
GROUP BY [Dział], [Płeć]
HAVING [Płeć]=[K]obieta czy (M)ężczyzna?
ORDER BY [Dział];
```

```
SELECT id_zesp, avg(placa_pod)
FROM pracownik
GROUP BY id_zesp
HAVING count(*) >3;
```

## **Połączenia w SQL (INNER, LEFT, RIGHT)**

```
SELECT *
FROM pracownik, zespól;
```

```
SELECT [ID Pracownika], Personel.Nazwisko, Personel.Imię, [Zasadnicza]
FROM Płace, Personel
WHERE Płace.[ID Pracownika]=Personel.[ID]
and Płace.Zasadnicza in (700, 1200, 1400, 1500);
```

```
SELECT [ID Pracownika], Personel.Nazwisko, Personel.Imię, [Zasadnicza]
FROM Płace, Personel
WHERE Płace.[ID Pracownika]=Personel.[ID]
and Zasadnicza between 800 and 1500;
```

```
SELECT p.[ID Pracownika], o.Nazwisko, o.Imię, p.[Zasadnicza]
FROM Płace AS p, Personel AS o
WHERE p.[ID Pracownika]=o.[ID]
and Zasadnicza between 800 and 1500;
```

```
SELECT [ID Pracownika], Personel.Nazwisko, Personel.Imię, [Zasadnicza]
FROM Płace, Personel
WHERE Płace.[ID Pracownika]=Personel.[ID]
```

**and** Personel.Nazwisko like 'B\*';

```
SELECT pracownik.nazwisko, zespol.nazwa
FROM Pracownik, Zespol
WHERE Pracownik.id_zesp=zespol.id_zesp;
```

```
SELECT pracownik.nazwisko, zespol.nazwa
FROM Pracownik INNER JOIN zespol ON pracownik.id_zesp=zespol.id_zesp;
```

```
SELECT pracownik.nazwisko, zespol.nazwa
FROM Pracownik RIGHT JOIN zespol ON pracownik.id_zesp=zespol.id_zesp;
```

```
SELECT [ID Pracownika], Personel.Nazwisko, Personel.Imię, [Zasadnicza]
FROM Płace, Personel
WHERE Płace.[ID Pracownika]=Personel.[ID]
and Zasadnicza not BETWEEN 800 and 1500;
```

```
SELECT Personel.Nazwisko, Personel.Imię, Płace.Zasadnicza, Personel.ID
FROM Personel INNER JOIN Płace ON Personel.ID = Płace.[ID Pracownika]
WHERE (((Personel.Nazwisko) Like 'B*') AND ((Personel.ID)=[ID Pracownika]));
```

## Aliasy

```
SELECT nazwisko, nazwa
FROM Pracownik AS p, Zespol AS z
WHERE p.id_zesp=z.id_zesp;
```

```
SELECT nazwisko, nazwa
FROM Pracownik AS p INNER JOIN zespol AS z ON p.id_zesp = z.id_zesp
ORDER BY nazwisko;
```

```
SELECT nazwisko, nazwa
FROM Pracownik AS p LEFT JOIN zespol AS z ON p.id_zesp = z.id_zesp
ORDER BY nazwisko;
```

```
SELECT nazwisko, nazwa, z.id_zesp
FROM Pracownik AS p RIGHT JOIN zespol AS z ON p.id_zesp = z.id_zesp;
```

```
SELECT Personel.Nazwisko, Oferta.[Nazwa kursu]
FROM Personel
INNER JOIN (Wykładowcy
INNER JOIN Oferta
ON Wykładowcy.[ID Kursu] = Oferta.[ID kursu])
ON Personel.ID = Wykładowcy.[ID Pracownika]
ORDER BY Personel.Nazwisko;
```

## Zapytania zagnieżdżone

```
SELECT NAZWISKO, PLACA_POD, ETAT, ID_ZESP
FROM PRACOWNIK
WHERE PLACA_POD > ANY
```

```
(select distinct placa_pod from Pracownik where id_zesp=30);
```

```
SELECT Personel.Dział, Format(Count([Nazwisko])/  
(SELECT Count([Nazwisko])  
FROM [Personel]),  
"0%") AS [Odsetek zatrudnionych]  
FROM Personel  
GROUP BY Personel.Dział;
```

```
SELECT NAZWISKO, PLACA_POD, ETAT, ID_ZESP  
FROM PRACOWNIK  
WHERE PLACA_POD >= ANY  
(select distinct placa_pod as placa_min from Pracownik m where id_zesp=[Podaj dział]);
```

```
SELECT Nazwisko, numer, szef, etat, id_zesp  
FROM Pracownik AS p  
WHERE exists  
( select numer from Pracownik  
  where Pracownik.szef=p.numer);
```

## Zapytania SQL-właściwe

### UNION

```
SELECT etat from Pracownik  
  where id_zesp=10  
UNION select etat from pracownik  
  where id_zesp=30;
```