

1.1. Stworzenia bazy danych:

```
01 | CREATE DATABASE IF NOT EXISTS spoleczenstwo;  
02 | USE spoleczenstwo;
```

1.2. Stworzenia tabeli Ludzie:

```
01 | CREATE TABLE IF NOT EXISTS Ludzie(  
02 |     id int AUTO_INCREMENT PRIMARY KEY,  
03 |     PESEL char(11),  
04 |     imie varchar(30) NOT NULL,  
05 |     nazwisko varchar(30) NOT NULL,  
06 |     data_urodzenia date,  
07 |     plec ENUM('K', 'M')  
08 | );
```

1.3. Stworzenia tabeli Zawody:

```
01 | CREATE TABLE IF NOT EXISTS Zawody(  
02 |     zawod_id int AUTO_INCREMENT PRIMARY KEY,  
03 |     nazwa varchar(50),  
04 |     pensja_min float,  
05 |     pensja_max float,  
06 |     CHECK(pensja_min > 0),  
07 |     CHECK(pensja_max > 0),  
08 |     CHECK(pensja_min < pensja_max)  
09 | );
```

1.4. Stworzenia tabeli Pracownicy:

```
01 | CREATE TABLE IF NOT EXISTS Pracownicy(  
02 |     id int NOT NULL,  
03 |     zawod_id int NOT NULL,  
04 |     pensja float NOT NULL,  
05 |     FOREIGN KEY(id) REFERENCES Ludzie(id),  
06 |     FOREIGN KEY(zawod_id) REFERENCES Zawody(zawod_id),  
07 |     CHECK(pensja > 0)  
08 | );
```

1.5. PESEL nie jest dobrym kluczem glownym, poniewaz:

- PESEL to dana wrażliwa, która powinno się 'ukrywać' a nie jest to proste jeśli występuje ona w wielu tabelach
- PESEL może się zmienić, co powoduje zmiany w wielu tabelach
- możemy chcieć wpisać do bazy danych osobę, która nie posiada PESELu (np. obcokrajowca)

1.6. Dodanie triggera walidującego PESEL przy dodaniu nowej osoby:

```
01 | DELIMITER $$
02 | CREATE TRIGGER PESELisValid
03 | BEFORE INSERT ON Ludzie
04 | FOR EACH ROW
05 | BEGIN
06 |     DECLARE tmp varchar(1);
07 |     DECLARE digit1, digit2, digit3, digit4, digit5, digit6, digit7, digit8,
          digit9, digit10, digit11, controlSum int;
08 |
09 |     IF NEW.PESEL NOT REGEXP
10 |         "[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]" THEN
11 |         signal sqlstate '45000'
12 |         set message_text = 'Invalid PESEL';
13 |     END IF;
14 |
15 |     SELECT SUBSTRING(NEW.PESEL, 1, 1) INTO tmp;
16 |     SELECT CAST(tmp AS int) * 1 INTO digit1;
17 |     SELECT SUBSTRING(NEW.PESEL, 2, 1) INTO tmp;
18 |     SELECT CAST(tmp AS int) * 3 INTO digit2;
19 |     SELECT SUBSTRING(NEW.PESEL, 3, 1) INTO tmp;
20 |     SELECT CAST(tmp AS int) * 7 INTO digit3;
21 |     SELECT SUBSTRING(NEW.PESEL, 4, 1) INTO tmp;
22 |     SELECT CAST(tmp AS int) * 9 INTO digit4;
23 |     SELECT SUBSTRING(NEW.PESEL, 5, 1) INTO tmp;
24 |     SELECT CAST(tmp AS int) * 1 INTO digit5;
25 |     SELECT SUBSTRING(NEW.PESEL, 6, 1) INTO tmp;
26 |     SELECT CAST(tmp AS int) * 3 INTO digit6;
27 |     SELECT SUBSTRING(NEW.PESEL, 7, 1) INTO tmp;
28 |     SELECT CAST(tmp AS int) * 7 INTO digit7;
29 |     SELECT SUBSTRING(NEW.PESEL, 8, 1) INTO tmp;
30 |     SELECT CAST(tmp AS int) * 9 INTO digit8;
31 |     SELECT SUBSTRING(NEW.PESEL, 9, 1) INTO tmp;
32 |     SELECT CAST(tmp AS int) * 1 INTO digit9;
33 |     SELECT SUBSTRING(NEW.PESEL, 10, 1) INTO tmp;
34 |     SELECT CAST(tmp AS int) * 3 INTO digit10;
35 |     SELECT SUBSTRING(NEW.PESEL, 11, 1) INTO tmp;
36 |     SELECT CAST(tmp AS int) INTO digit11;
37 |
38 |     SELECT (digit1 + digit2 + digit3 + digit4 + digit5 + digit6 + digit7 +
          digit8 + digit9 + digit10) INTO controlSum;
39 |     SELECT MOD(controlSum, 10) INTO controlSum;
40 |
41 |     SELECT IF(controlSum=0, 0, 10-controlSum) INTO controlSum;
42 |
43 |     IF controlSum <> digit11 THEN
44 |         signal sqlstate '45000'
45 |         set message_text = 'Invalid PESEL';
46 |     END IF;
47 | END$$
48 | DELIMITER ;
```

1.7. Dodanie danych do tabeli Ludzie:

```
01 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
02 | VALUE ('18262925559', 'Albin', 'Kaczmarczyk', '2018-06-29', 'M');
03 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
04 | VALUE ('11231603016', 'Gerard', 'Majchrzak', '2011-03-16', 'M');
05 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
06 | VALUE ('17282201227', 'Hanna', 'Matuszewska', '2017-08-22', 'K');
07 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
08 | VALUE ('05302156826', 'Bronislawa', 'Kania', '2005-10-21', 'K');
09 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec) \
10 | VALUE ('07221812878', 'Kornel', 'Kowalik', '2007-02-18', 'M');
11 |
12 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
13 | VALUE ('84013146089', 'Magdalena', 'Baran', '1984-01-31', 'K');
14 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
15 | VALUE ('93010315910', 'Zenon', 'Domanski', '1993-01-03', 'M');
16 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
17 | VALUE ('74122545204', 'Henryka', 'Laskowska', '1974-12-25', 'K');
18 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
19 | VALUE ('80120903317', 'Sylwester', 'Nowacki', '1980-12-09', 'M');
20 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
21 | VALUE ('80050393589', 'Sandra', 'Szczepaniak', '1980-05-03', 'K');
22 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
23 | VALUE ('80122902567', 'Cecylia', 'Maj', '1980-12-29', 'K');
24 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
25 | VALUE ('69051984538', 'Romuald', 'Owczarek', '1969-05-19', 'M');
26 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
27 | VALUE ('81103028003', 'Urszula', 'Pawlowska', '1981-10-30', 'K');
28 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
29 | VALUE ('80091053831', 'Alfred', 'Czajka', '1980-09-10', 'M');
30 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
31 | VALUE ('90091939000', 'Natasza', 'Blaszczyk', '1990-09-19', 'K');
32 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
33 | VALUE ('79012318731', 'Edward', 'Kurek', '1979-01-23', 'M');
34 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
35 | VALUE ('91071083636', 'Ihor', 'Gurecki', '1991-07-10', 'M');
36 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
37 | VALUE ('90101643154', 'Denis', 'Kasprzak', '1990-10-16', 'M');
38 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
39 | VALUE ('64111714582', 'Zaneta', 'Karpinska', '1964-11-17', 'K');
40 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
41 | VALUE ('86082423908', 'Longina', 'Gajda', '1986-08-24', 'K');
42 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
43 | VALUE ('76102127761', 'Alfreda', 'Zak', '1976-10-21', 'K');
44 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
45 | VALUE ('80090681626', 'Zuzanna', 'Musial', '1980-09-06', 'K');
46 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
47 | VALUE ('78080795909', 'Brygida', 'Krupa', '1978-08-07', 'K');
48 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
49 | VALUE ('78051992357', 'Oliwier', 'Czajka', '1978-05-19', 'M');
50 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
51 | VALUE ('77100195196', 'Marcin', 'Tomczyk', '1977-10-01', 'M');
52 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
53 | VALUE ('77031448026', 'Teofila', 'Sliwinska', '1977-03-14', 'K');
54 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
55 | VALUE ('79110334547', 'Edyta', 'Wolska', '1979-11-03', 'K');
```

```
56 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
57 |     VALUE ('85042799172', 'Kacper', 'Szymczak', '1985-04-27', 'M');
58 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
59 |     VALUE ('74101010518', 'Kajetan', 'Walczak', '1974-10-10', 'M');
60 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
61 |     VALUE ('62021341218', 'Denis', 'Lipinski', '1962-02-13', 'M');
62 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
63 |     VALUE ('95022302030', 'Teodor', 'Marciniak', '1995-02-23', 'M');
64 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
65 |     VALUE ('83021482763', 'Olivia', 'Brzozowska', '1983-02-14', 'K');
66 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
67 |     VALUE ('91080819585', 'Krystyna', 'Blaszczyk', '1991-08-08', 'K');
68 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
69 |     VALUE ('82021093052', 'Piotr', 'Karpinski', '1982-02-10', 'M');
70 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
71 |     VALUE ('72011510319', 'Ryszard', 'Pawlik', '1972-01-15', 'M');
72 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
73 |     VALUE ('75101063124', 'Boguslawa', 'Szewczyk', '1975-10-10', 'K');
74 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
75 |     VALUE ('91081472327', 'Adrianna', 'Jakubowska', '1991-08-14', 'K');
76 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
77 |     VALUE ('95022848699', 'Andrii', 'Sowa', '1995-02-28', 'M');
78 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
79 |     VALUE ('63052287492', 'Czeslaw', 'Bednarek', '1963-05-22', 'M');
80 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
81 |     VALUE ('93122451528', 'Jagoda', 'Nowacka', '1993-12-24', 'K');
82 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
83 |     VALUE ('81091034505', 'Dagmara', 'Lewandowska', '1981-09-10', 'K');
84 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
85 |     VALUE ('86042047913', 'Oleh', 'Polak', '1986-04-20', 'M');
86 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
87 |     VALUE ('86042687337', 'Teofil', 'Szczepaniak', '1986-04-26', 'M');
88 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
89 |     VALUE ('97061502170', 'Wojciech', 'Orzechowski', '1997-06-15', 'M');
90 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
91 |     VALUE ('94060193439', 'Marceli', 'Czajkowski', '1994-06-01', 'M');
92 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
93 |     VALUE ('64092130944', 'Adela', 'Markowska', '1964-09-21', 'K');
94 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
95 |     VALUE ('71052347645', 'Natasza', 'Marciniak', '1971-05-23', 'K');
96 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
97 |     VALUE ('66042722128', 'Milena', 'Krajewska', '1966-04-27', 'K');
98 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
99 |     VALUE ('76121330085', 'Maja', 'Marek', '1976-12-13', 'K');
100 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
101 |     VALUE ('95091564032', 'Slawomir', 'Sawicki', '1995-09-15', 'M');
102 |
103 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
104 |     VALUE ('47042977367', 'Teodozja', 'Laskowska', '1947-04-29', 'K');
105 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
106 |     VALUE ('26030617111', 'Natan', 'Michalski', '1926-03-06', 'M');
107 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
108 |     VALUE ('37040563051', 'Juliusz', 'Karpinski', '1937-04-05', 'M');
109 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
110 |     VALUE ('51022260967', 'Svitlana', 'Dobrowolska', '1951-02-22', 'K');
111 | INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec)
112 |     VALUE ('38062126488', 'Barbara', 'Kwiatkowska', '1938-06-21', 'K');
```

1.7. Dodanie danych do tabeli Zawody:

```
01 | INSERT INTO Zawody (nazwa, pensja_min, pensja_max)
02 |     VALUE ('polityk', 6000.0, 60000.0);
03 | INSERT INTO Zawody (nazwa, pensja_min, pensja_max)
04 |     VALUE ('nauczyciel', 3000.0, 5000.0);
05 | INSERT INTO Zawody (nazwa, pensja_min, pensja_max)
06 |     VALUE ('lekarz', 10000.0, 40000.0);
07 | INSERT INTO Zawody (nazwa, pensja_min, pensja_max)
08 |     VALUE ('informatyk', 5000.0, 50000.0);
```

1.8 Dodanie danych do tabeli Pracownicy procedura korzystająca z kursora:

```
01 | DELIMITER $$
02 | CREATE PROCEDURE IF NOT EXISTS nadajZawodyLudziom()
03 | BEGIN
04 |     DECLARE done int DEFAULT FALSE;
05 |     DECLARE czlowiekId, zawodId, lekarzId int;
06 |     DECLARE minPlaca, maxPlaca, placa float;
07 |     DECLARE plec enum('K', 'M');
08 |     DECLARE wiek int;
09 |     DECLARE ludzieCursor CURSOR FOR SELECT id, plec,
10 |         TIMESTAMPDIFF(YEAR, data_urodzenia, NOW()) AS wiek FROM Ludzie;
11 |     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
12 |
13 |     OPEN ludzieCursor;
14 |
15 |     read_loop: LOOP
16 |         FETCH ludzieCursor INTO czlowiekId, plec, wiek;
17 |
18 |         IF done THEN
19 |             LEAVE read_loop;
20 |         END IF;
21 |
22 |         IF (plec='K' AND wiek>60) OR (plec='M' AND wiek>65) THEN
23 |             SELECT zawod_id INTO zawodId FROM Zawody
24 |             WHERE zawod_id NOT IN (SELECT zawod_id FROM Zawody
25 |                                     WHERE nazwa = 'lekarz')
26 |             ORDER BY RAND() LIMIT 1;
27 |         ELSEIF wiek > 17 THEN
28 |             SELECT zawod_id INTO zawodId FROM Zawody
29 |             ORDER BY RAND() LIMIT 1;
30 |         ELSE
31 |             ITERATE read_loop;
32 |         END IF;
33 |
34 |         SELECT pensja_min INTO minPlaca FROM Zawody WHERE zawod_id=zawodId;
35 |         SELECT pensja_max INTO maxPlaca FROM Zawody WHERE zawod_id=zawodId;
36 |         SET placa = FLOOR(minPlaca + RAND()*(maxPlaca-minPlaca));
37 |
38 |         INSERT INTO Pracownicy (id, zawod_id, pensja) VALUE (czlowiekId,
39 |             zawodId, placa);
40 |     END LOOP;
41 |
42 |     CLOSE ludzieCursor;
43 | END$$
44 | DELIMITER ;
```

2.1. Dodanie indeksow:

```
01 | CREATE INDEX custom_index_1 ON Ludzie(plec, imie);
02 | CREATE INDEX custom_index_2 ON Pracownicy(pensja);
```

2.2. Indeksy stworzone na tabeli

```
01 | SHOW INDEX FROM Ludzie;
02 | SHOW INDEX FROM Zawody;
03 | SHOW INDEX FROM Pracownicy;
```

- Ludzie: Indeksy zalozone na: id, plec, imie;
- Zawody: Indeks zalozony na zawod_id;
- Pracownicy: Indeksy zalozone na: id, zawod_id, pensja;

2.3. Wypisanie kobiet, ktorzych imie zaczyna sie na 'A'

(Uzywa indeks custom_index_1):

```
01 | EXPLAIN SELECT id, imie, nazwisko FROM Ludzie
02 | WHERE plec='K' AND imie LIKE "A%";
```

2.4. Wypisanie wszystkich kobiet

(Nie uzywa indeksow):

```
01 | EXPLAIN SELECT id, imie, nazwisko FROM Ludzie
02 | WHERE plec='K';
```

2.5. Wypisanie osob, ktorzych imie zaczyna sie na 'K'

(Nie uzywa indeksow):

```
01 | EXPLAIN SELECT id, imie, nazwisko FROM Ludzie
02 | WHERE imie LIKE "K%";
```

2.6. Wypisanie osob, ktore zarabiaja ponizej 4000 PLN

(Uzywa indeksy: custom_index_2, PRIMARY):

```
01 | EXPLAIN SELECT Pracownicy.id, imie, nazwisko, pensja FROM Pracownicy
02 | LEFT JOIN Ludzie ON Pracownicy.id = Ludzie.id
03 | WHERE pensja < 4000;
```

2.7. Wypisanie informatykow plci meskiej, ktorzy zarabiaja powyzej 10000 PLN

(Uzywa indeksy: zawod_id, PRIMARY):

```
01 | EXPLAIN SELECT Pracownicy.id, imie, nazwisko, nazwa, pensja FROM Pracownicy
02 | LEFT JOIN Ludzie ON Pracownicy.id = Ludzie.id
03 | LEFT JOIN Zawody ON Zawody.zawod_id=Pracownicy.zawod_id
04 | WHERE pensja > 10000 AND plec='M' AND nazwa='informatyk';
```

3.1. Stworzenie procedury dającej podwyżkę konkretnej grupie zawodowej:

```
01 | DELIMITER $$
02 | CREATE PROCEDURE IF NOT EXISTS dajPodwyzke(nazwaZawodu varchar(50))
03 | exit_label:BEGIN
04 |     DECLARE n, i, zawodId, currZawodId, pracownikId int;
05 |     DECLARE placa, maxPlaca, nowaPlaca float;
06 |
07 |     SELECT zawod_id INTO zawodId FROM Zawody WHERE nazwa=nazwaZawodu;
08 |     SELECT pensja_max INTO maxPlaca FROM Zawody WHERE zawod_id=zawodId;
09 |     SELECT COUNT(*) INTO n FROM Pracownicy;
10 |
11 |     START TRANSACTION;
12 |
13 |     SET i=0;
14 |     WHILE i<n DO
15 |         SELECT id INTO pracownikId FROM Pracownicy LIMIT i,1;
16 |         SELECT zawod_id INTO currZawodId FROM Pracownicy LIMIT i,1;
17 |         SELECT pensja INTO placa FROM Pracownicy LIMIT i,1;
18 |
19 |         IF currZawodId = zawodId THEN
20 |             SET nowaPlaca = 1.05 * placa;
21 |             IF nowaPlaca > maxPlaca THEN
22 |                 ROLLBACK;
23 |                 LEAVE exit_label;
24 |             END IF;
25 |             UPDATE Pracownicy SET pensja=nowaPlaca WHERE id=pracownikId;
26 |         END IF;
27 |
28 |         SET i = i + 1;
29 |     END WHILE;
30 |     COMMIT;
31 | END$$
32 | DELIMITER ;
```


6.1. SQL Injection(introduction):

```
01 | SELECT department FROM employees WHERE first_name='Bob';
```

```
01 | UPDATE employees SET department='Sales' WHERE first_name='Tobi';
```

```
01 | ALTER TABLE employees ADD phone varchar(20);
```

```
01 | GRANT ALL ON grant_rights TO unauthorized_user;
```

```
01 | SELECT * FROM user_data
```

```
02 | WHERE first_name = 'John' AND last_name = '' OR '1' = '1'
```

```
01 | SELECT * From user_data WHERE Login_Count = 1 and userid= 1 OR TRUE
```

```
01 | Employee Name: 1
```

```
02 | Auth TAN: ' OR TRUE -- -
```

```
01 | Employee Name: '; UPDATE employees SET salary=1000000
```

```
02 | WHERE last_name='Smith';-- -
```

```
01 | Action contains: '; DROP TABLE access_log;-- -
```

6.2. SQL Injection(advanced):

```
01 | Name: ' OR TRUE UNION SELECT 1,'2','3','4','5',password, 7 from
```

```
02 | user_system_data WHERE user_name='dave'-- -
```

```
03 | Pass: passW0rD
```

```
01 | 1.D
```

```
02 | 2.C
```

```
03 | 3.B
```

```
04 | 4.C
```

```
05 | 5.D
```

6.3. SQL Injection(mitigation):

```
01 | getConnection, PreparedStatement, preparedStatement, ?, ?, getString, getString
```

```
01 | try {
02 |     Connection connection = null;
03 |     connection = DriverManager.getConnection(DBURL, DBUSER, DBPW);
04 |     PreparedStatement preparedStatement =
05 |         connection.prepareStatement("SELECT * FROM USERS WHERE NAME=?");
06 |     preparedStatement.setString(1, "3");
07 |     ResultSet resultState = preparedStatement.executeQuery();
08 | }
09 | catch(Exception expception) {
10 |     System.out.println("Error!");
11 | }
```


01 | `a';/**/select/**/*/**/from/**/user_system_data;--`

6.4: Sposoby zapobiegania przed atakami:

- Statyczne zapytania
- Parametryzowane zapytania
- Przechowywanie procedur bez dynamicznego ich budowania