1.1 CREATE DATABASE census;

1.2 CREATE TABLE Ludzie( PESEL char(11) NOT NULL, imie varchar(30) NOT NULL, nazwisko varchar(30) NOT NULL, data_urodzenia date NOT NULL,plec enum('F', 'M') NOT NULL, PRIMARY KEY (PESEL));

1.3 CREATE TABLE Zawody( zawod_id int NOT NULL AUTO_INCREMENT, nazwa varchar(50) NOT NULL, pensja_min float NOT NULL, pensja_max float NOT NULL, PRIMARY KEY(zawod_id), CONSTRAINT minmax CHECK(pensja_min >= 0 and pensja_max >=0 and pensja_min < pensja_max));

1.4 CREATE TABLE Pracownicy (PESEL char(11) NOT NULL, zawod_id int NOT NULL, pensja float NOT NULL);

1.5 INSERT INTO ludzie VALUES  ... ze srciptu w pythonie

1.6 INSERT INTO zawody VALUES (1, "polityk", 6000.0, 25000.0), (2, "nauczyciel", 3200.51, 6210.2), (3, "lekarz", 4200.0, 20000.0), (4, "informatyk", 5000.0, 50000.0);

1.7

```
DELIMITER //


CREATE PROCEDURE people_to_profession()
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE psl char(11);
  DECLARE birth_date date;
  DECLARE gender enum('F', 'M');
  DECLARE proff_id int;
  DECLARE earn float;
  DECLARE min_ern float;
  DECLARE max_ern float;
  DECLARE cur_ludzie CURSOR FOR SELECT PESEL, data_urodzenia, plec FROM ludzie;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;


  OPEN cur_ludzie;


  gen_loop: LOOP
    FETCH cur_ludzie INTO psl, birth_date, gender;
  IF TIMESTAMPDIFF(YEAR, birth_date, CURDATE()) >= 18 THEN
   IF done THEN
    LEAVE gen_loop;
   END IF;

   fetch_proffesion: LOOP
    SET proff_id = FLOOR(1 + RAND()*4);
    IF proff_id <> 3 THEN
     LEAVE fetch_proffesion;
    ELSEIF gender = "M" THEN
     IF TIMESTAMPDIFF(YEAR, birth_date, CURDATE()) <= 65 THEN
      LEAVE fetch_proffesion;
     END IF;
    ELSE
     IF TIMESTAMPDIFF(YEAR, birth_date, CURDATE()) <= 06 THEN
      LEAVE fetch_proffesion;
     END IF;
    END IF;
   END LOOP;

   SET min_ern = (SELECT pensja_min FROM zawody WHERE zawod_id = proff_id);
   SET max_ern = (SELECT pensja_max FROM zawody WHERE zawod_id = proff_id);

   SET earn = ROUND((min_ern + (RAND() * (max_ern - min_ern + 1))), 2);

   INSERT INTO pracownicy VALUES (psl, proff_id, earn);
  END IF;
 END LOOP;

 CLOSE cur_ludzie;
END; //
```

```
DELIMITER ;
```

2.1 ALTER TABLE ludzie ADD INDEX compound(plec, imie);

2.2 CREATE INDEX indx ON pracownicy (pensja);

2.3 SELECT * FROM ludzie AS l JOIN pracownicy AS p ON l.PESEL = p.PESEL WHERE imie LIKE 'A%' AND plec = 'F';

2.4 SELECT * FROM ludzie AS l JOIN pracownicy AS p ON l.PESEL = p.PESEL WHERE plec = 'F';

2.5 SELECT * FROM ludzie AS l JOIN pracownicy AS p ON l.PESEL = p.PESEL WHERE imie LIKE 'K%';

2.6 SELECT * FROM ludzie AS l JOIN pracownicy AS p ON l.PESEL = p.PESEL WHERE pensja > 10000 AND plec = 'M' AND  zawod_id = 4;

3.

```
DELIMITER //


CREATE PROCEDURE podwyzka(IN praca varchar(50))
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE psl char(11);
  DECLARE pen float;
  DECLARE wanted_id int DEFAULT (SELECT zawod_id FROM zawody WHERE nazwa = praca);
  DECLARE id int;
  DECLARE cur CURSOR FOR SELECT PESEL, zawod_id, pensja FROM pracownicy;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
  START TRANSACTION;
    raise_loop: LOOP
      FETCH cur INTO psl, id, pen;
      IF id = wanted_id THEN
        IF done THEN
          LEAVE raise_loop;
        END IF;
    IF (pen * 1.05) > (SELECT pensja_max FROM zawody WHERE zawod_id = id) THEN
     ROLLBACK;
     LEAVE raise_loop;
    ELSE
     UPDATE pracownicy SET pensja = (pen * 1.05) WHERE PESEL = psl;
    END IF;
   END IF;
  END LOOP;
 COMMIT;
END; //

DELIMITER ;
```

4.1 PREPARE num_of_F_in_proff FROM 'SELECT COUNT(*) FROM pracownicy AS p JOIN ludzie AS l ON p.PESEL = l.PESEL JOIN zawody AS z ON z.zawod_id = p.zawod_id WHERE plec = "F" AND nazwa = ?';

4.2 EXECUTE num_of_F_in_proff USING 'informatyk';

5.1 mysqldump -u root -p -x -A census > census.sql

5.2 DROP DATABASE census;

5.3 CREATE DATABASE census;

5.4 mysql -u root -p census < data-dump.sql

6.1.2 SELECT department FROM Employees WHERE first_name = 'Bob' AND last_name = 'Franco' ;

6.1.3 UPDATE employees SET department = 'Sales' WHERE first_name = 'Tobi' AND last_name = 'Barnett' ;

6.1.4 ALTER TABLE employees ADD phone varchar(20) ;

6.1.5 GRANT ALL PRIVILEGES ON grant_rights TO unauthorized_user;

6.1.9 Smith' OR '1' = '1

6.1.10 1 ,  '2137' OR 1 = 1

6.1.11 Smith' OR '1' = '1, 3SL99A' OR '1' = '1

6.1.12 Smith' OR '1' = '1, 3SL99A' OR '1' = '1' ; UPDATE employees SET SALARY = 99999 WHERE USERID = 37648; --

6.1.13 %' ;DROP TABLE  access_log ; --

6.2.3 SELECT * FROM user_data WHERE last_name = 'Dave' OR '1' = '1' ; SELECT * FROM user_data JOIN user_system_data ON user_data.cookie = user_system_data.cookie; --'

6.2.5

6.2.6 4, 3, 2, 3, 4

6.3.5 getConnection, PreparedStatement statement, prepareStatement, ?, ?, statement.setString(1, name), statement.setString(2, mail)

6.3.6

6.3.9 a';/**/select/**/*/**/from/**/user_system_data;--

6.3.10 a';/**/seselectlect/**/*/**/frfromom/**/user_system_data;--

6.3.12

6. wnioski: trzeba kożystać z Prepared Statements i Parametrized Statements i filtracji inputu