

# Zaposleni – employees

Podatkovne baze

- 1. Napišite SQL skripto za ustvarjanje tabel dept\_manager in salaries. Ustvarjeni tabeli naj vsebujeta primarne in tuje ključe ter omejitve, da datum v polju to\_date ne sme biti manjši od datuma v polju from\_date. Polje to\_date je lahko tudi prazno (vrednost NULL).**

```
CREATE TABLE dept_manager (  
    dept_no CHAR(4) NOT NULL,  
    emp_no INT(11) NOT NULL,  
    from_date DATE NOT NULL,  
    to_date DATE CHECK (to_date>=from_date),  
    FOREIGN KEY (dept_no) REFERENCES departments(dept_no),  
    FOREIGN KEY (emp_no) REFERENCES employees(emp_no),  
    PRIMARY KEY (dept_no, emp_no)  
);
```

```
CREATE TABLE salaries (  
    emp_no INT(11) NOT NULL,  
    salary INT(11) NOT NULL,  
    from_date DATE NOT NULL,  
    to_date DATE CHECK (to_date>=from_date),  
    PRIMARY KEY (emp_no, from_date),  
    FOREIGN KEY (emp_no) REFERENCES employees(emp_no)  
);
```

- 1.1. Obstoječa zbirka podatkov obravnava nezaključene časovne intervale na poseben način. Opišite kateri? Opišite tudi kako bi lahko alternativno zaključili te intervale, ter kakšne so prednosti/slabosti teh dveh pristopov.**

V obstoječi zbirki je nezaključen časovni interval realiziran z uporabo največjega datuma možno, torej datum z letom 9999. V primerjavi z drugim načinom tukaj delamo izključno z datumi in lahko omoči lažje delo v poizvedbah, kjer želiš delati z časovnimi obdobji. Težave se lahko pojavijo pri "končnih" uporabnikih, ki bi jih leto 9999 zmedlo, zato se to vrednost lahko na uporabniškem vmesniku pretvori v vrednost npr. "brez izteka", "do nadaljnjega".

Alternativno to lahko realiziramo, da največji možni datum nadomestimo z vrednostjo NULL. S tem bi skrbeli za čisti jezik, bili bi "puristi". Podatki pa bi tudi bili lahko bolj berljivi, razumljivi, a bi se delo s časovnimi poizvedbami toliko bolj zakompliciralo.

## 2.

- a) Ustvarite pogled (CREATE VIEW) place\_in\_nazivi, ki bo prikazal združene podatke iz tabel employees, salaries in titles, tako da pri normalizaciji tega pogleda ne bo izgubljenih ali podvojenih podatkov. Nato ustvarite še pogled pogled\_nazivi, ki bo črpal podatke iz pogleda place\_in\_nazivi in prikaže enake podatke, kot jih vsebuje tabela titles. Namig: Pazite na imena stolpcev. (5%)

```
CREATE VIEW place_in_nazivi AS
```

```
SELECT e.emp_no, e.birth_date, e.first_name, e.last_name, e.gender,
       e.hire_date,
       s.salary, s.from_date AS "placa_od", s.to_date AS "placa_do",
       t.title, t.from_date AS "naziv_od", t.to_date AS "naziv_do"
FROM employees e JOIN salaries s USING(emp_no)
JOIN titles t USING(emp_no);
```

```
CREATE VIEW pogled_nazivi AS
```

```
SELECT DISTINCT emp_no, title, naziv_od, naziv_do
FROM place_in_nazivi;
```

```
SELECT *
```

```
FROM pogled_nazivi;
```

emp_no	title	naziv_od	naziv_do
10001	Senior Engineer	1986-06-26	9999-01-01
10002	Staff	1996-08-03	9999-01-01
10003	Senior Engineer	1995-12-03	9999-01-01
10004	Engineer	1986-12-01	1995-12-01
10004	Senior Engineer	1995-12-01	9999-01-01
10005	Senior Staff	1996-09-12	9999-01-01
10005	Staff	1989-09-12	1996-09-12
10006	Senior Engineer	1990-08-05	9999-01-01
10007	Senior Staff	1996-02-11	9999-01-01
10007	Staff	1989-02-10	1996-02-11
10008	Assistant Engi...	1998-03-11	2000-07-31
10009	Assistant Engi...	1985-02-18	1990-02-18
10009	Engineer	1990-02-18	1995-02-18
10009	Senior Engineer	1995-02-18	9999-01-01
10010	Engineer	1996-11-24	9999-01-01
10011	Staff	1990-01-22	1996-11-09
10012	Engineer	1992-12-18	2000-12-18

- b) Z uporabo CTE preverite, če pogled pogled\_nazivi in tabela titles vsebujeta enake vrstice. CTE naj prikaže vrstice, ki se ne ujemajo. (10%)

```
WITH pogled AS (  
    SELECT *  
    FROM pogled_nazivi  
)  
, nazivi AS (  
    SELECT *  
    FROM titles  
)  
SELECT *  
FROM nazivi  
where NOT EXISTS  
    (SELECT *  
     FROM pogled);
```

	emp_no	title	from_date	to_date

- c) \*S pomočjo ustreznih DDL ukazov ustvarite tabelo top3(dept\_no, SteviloZaposlenih), ki hrani le 3 oddelke z največ zaposlenimi. Dodajte bazne prožilce tako, da se bo lista posodobila vsakič, ko se spremeni število zaposlenih. (10%)

```
CREATE TABLE top3 (  
    dept_no CHAR(4) PRIMARY KEY,  
    SteviloZaposlenih INTEGER  
) AS  
  
SELECT dept_no, COUNT(emp_no) AS SteviloZaposlenih  
FROM dept_emp  
WHERE YEAR(to_date) = '9999'  
GROUP BY dept_no  
ORDER BY COUNT(emp_no) DESC  
LIMIT 3;
```

```

DELIMITER $
CREATE TRIGGER posodobi_UPTop3
AFTER UPDATE ON dept_emp
FOR EACH ROW
BEGIN
    DECLARE st_zapNEW INT;
    DECLARE st_zapT3 INT;
    DECLARE depT3 CHAR(4);

    SET st_zapNEW = ( SELECT COUNT(emp_no) FROM dept_emp WHERE dept_no =
NEW.dept_no AND YEAR(to_date) = '9999' );
    SET st_zapT3 = ( SELECT SteviloZaposlenih FROM top3 ORDER BY
SteviloZaposlenih ASC LIMIT 1 );
    SET depT3 = ( SELECT dept_no FROM top3 ORDER BY SteviloZaposlenih
ASC LIMIT 1 );

    IF st_zapNEW > st_zapT3
        THEN
            DELETE FROM top3 WHERE dept_no = depT3;
            INSERT INTO top3 VALUES(NEW.dept_no, st_zapNEW);
            ALTER TABLE top3 ORDER BY SteviloZaposlenih DESC;
        END IF;
END $
DELIMITER;

DELIMITER $
CREATE TRIGGER posodobi_INStop3
AFTER INSERT ON dept_emp
FOR EACH ROW
BEGIN
    DECLARE st_zapNEW INT;
    DECLARE st_zapT3 INT;
    DECLARE depT3 CHAR(4);

    SET st_zapNEW = ( SELECT COUNT(emp_no) FROM dept_emp WHERE dept_no =
NEW.dept_no AND YEAR(to_date) = '9999' );
    SET st_zapT3 = ( SELECT SteviloZaposlenih FROM top3 ORDER BY
SteviloZaposlenih ASC LIMIT 1 );
    SET depT3 = ( SELECT dept_no FROM top3 ORDER BY SteviloZaposlenih
ASC LIMIT 1 );

    IF st_zapNEW > st_zapT3
        THEN
            DELETE FROM top3 WHERE dept_no = depT3;
            INSERT INTO top3 VALUES(NEW.dept_no, st_zapNEW);
            ALTER TABLE top3 ORDER BY SteviloZaposlenih DESC;
        END IF;
END $
DELIMITER;

```

**3. V jeziku SQL napišite poizvedbe, s pomočjo katerih boste odgovorili na naslednja vprašanja. Upoštevajte, da se je baza nazadnje posodobila 1.8.2002.**

- a) Med katerimi leti obstajajo podatki za plače v bazi? Napišite eno poizvedbo, ki vam poda odgovor.

```
SELECT MIN(YEAR(from_date)) AS "OD", MAX(YEAR(to_date)) AS "DO"
FROM salaries
WHERE YEAR(to_date) != '9999';
```

\*Vzel sem najmanjše leto plač in največje, ki ni leto 9999.

OD	DO
1985	2002

- b) Katerih 10 zaposlenih je dobilo največje izplačilo. Izpišite priimke, imena in največje izplačilo, ter izpis uredite po priimku in imenu.

```
SELECT *
FROM (SELECT e.first_name, e.last_name, SUM(s.salary) AS izplacilo
      FROM employees e join salaries s USING(emp_no)
      GROUP BY e.emp_no
      ORDER BY izplacilo desc
      LIMIT 10
     ) AS naj
ORDER BY first_name, last_name
```

first_name	last_name	izplacilo
Aral	Hoppenstand	2305351
Arnd	Junot	2271104
Chirstian	Kobara	2279289
Lansing	Kambil	2383923
Odoardo	Lindenbaum	2270899
Tokuyasu	Pesch	2492873
Tsutomu	Alameldin	2553036
Weicheng	Hatcliff	2381119
Willard	Baca	2368170
Xiahua	Whitcomb	2374024

\*Izplačilo sem vzel kot celoten zaslužek zaposlenega pri tem podjetju, uporabim SUM, saj so plače napisane za vsako leto.

- c) Koliko je bila povprečna starost vseh zaposlenih na dan 1.1.2000.

```
SELECT AVG(DATEDIFF('2000-01-01', birth_date) / 365.2425) as povp_starost
FROM employees
WHERE hire_date <= '2000-01-01';
```

povp_starost
41.41991787

\*Povprečna starost zaposlenih na dan 1.1.2000, torej so bili zaposleni do tega datuma oz. na ta datum.

- d) Koliko časa v letih (lahko uporabite decimalna števila) je bila zaposlena vsaka oseba v tem podjetju. Poleg časa izpišite tudi zaposlenčevo šifro, ime in priimek.

```
SELECT e.emp_no, e.first_name, e.last_name,
       IF(YEAR(MAX(t.to_date)) = '9999', DATEDIFF('2002-8-1', hire_date) /
       365.2425, DATEDIFF(MAX(t.to_date), hire_date) / 365.2425) AS
       cas_zaposlenosti
FROM employees e JOIN titles t USING(emp_no)
GROUP BY e.emp_no, e.first_name, e.last_name
ORDER BY e.emp_no;
```

emp_no	first_name	last_name	cas_zaposlenosti
10001	Georgi	Facello	16.0989
10002	Bezael	Simmel	16.6930
10003	Parto	Bamford	15.9264
10004	Chirstian	Koblick	15.6663
10005	Kyoichi	Maliniak	12.8846
10006	Anneke	Preusig	13.1639
10007	Tzvetan	Zielinski	13.4705
10008	Saniya	Kalloufi	5.8755
10009	Sumant	Peac	17.4487
10010	Duangkaew	Piveteau	12.9366
10011	Mary	Sluis	6.7982
10012	Patricio	Bridgland	9.6183
10013	Eberhardt	Terkki	16.7806
10014	Berni	Genin	15.3925

- e) Izpišite imena in priimke zaposlenih, katerih imena se začnejo na dva zaporedna samoglasnika, priimki pa kjerkoli v primku vsebujejo dva zaporedna samoglasnika in se ne končajo na samoglasnik.

```
SELECT first_name, last_name
FROM employees
WHERE first_name RLIKE '^(a|e|i|o|u)(a|e|i|o|u). '
AND last_name RLIKE '.(a|e|i|o|u)(a|e|i|o|u).[^aeiou]$';
```

first_name	last_name
Aamod	Rijckaert
Aamer	Schwartzbauer
Aiman	Georg
Augustine	Menhoudj
Iara	Schwartzbauer
Aamer	Vieth
Eishiro	DuBourdieux
Uinam	Fairtlough
IEEE	Gniady

- f) Za dan 1.1.1990 želimo izpisati najmlajše zaposlene. Izpišite imena, priimke in njihovo starost v letih. Urejeno padajoče po starosti. Pozor: Urejeni naj ne bodo le na letnico starosti.

```
SELECT first_name, last_name, DATEDIFF('1990-1-1', birth_date) / 365.2425
AS leta
FROM employees
WHERE hire_date <= '1990-1-1'
ORDER BY birth_date DESC;
```

first_name	last_name	leta
Patricio	Itzfeldt	24.9150
Shay	Servieres	24.9150
Surveyors	Bade	24.9150
Hirochika	Goldhammer	24.9150
Malu	Ossenbruggen	24.9150
Adamantios	Vanwelkenhuysen	24.9150
Fei	Erez	24.9150
Kazunori	Perz	24.9150
Hinrich	Perin	24.9150
Zsolt	Anily	24.9150
Jaewon	Thummel	24.9150
Zhiguo	Staudhammer	24.9150

- g) Za vsakega zaposlenega ugotovite njegovo tedensko plačo in število dni zaposlitve. Izpišite zaposlenčevo šifro, tedensko plačo in število dni zaposlitve padajoče po tedenski plači.

```
SELECT emp_no, AVG(salary) / 52 AS teden_placa,
IF(YEAR(MAX(to_date)) = '9999', DATEDIFF('2002-8-1', MIN(from_date)),
DATEDIFF(MAX(to_date), MIN(from_date))) AS dni
FROM salaries
GROUP BY emp_no
ORDER BY teden_placa DESC;
```

emp_no	teden_placa	dni
109334	2727.60256346	6376
205000	2712.78146923	3946
43624	2663.32585385	6337
493158	2659.86298077	5563
37558	2657.99725192	4907
276633	2629.07179423	5411
238117	2615.88846154	5324
46439	2610.53333269	5188
254466	2606.55889423	5837

\*Tedensko plačo sem dobil da sem povprečno letno plačo delil z 52, saj je toliko tednov v letu.



**h) Kateri oddelki imajo zaposlenih nadpovprečno število žensk?\***

```
DELIMITER //
```

```
CREATE FUNCTION st_zaposlenih () RETURNS INTEGER
```

```
BEGIN
```

```
    DECLARE X INTEGER;  
    SELECT DISTINCT COUNT(d.emp_no) INTO X  
    FROM employees e JOIN dept_emp d USING(emp_no);  
    RETURN X;
```

```
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE FUNCTION povp_zensk () RETURNS REAL
```

```
BEGIN
```

```
    DECLARE X REAL;  
    SELECT COUNT(d.emp_no)/st_zaposlenih() AS povprecno_z INTO X  
    FROM employees e JOIN dept_emp d USING(emp_no)  
    WHERE e.gender = 'F';  
    RETURN X;
```

```
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE FUNCTION za_skupino (skupina VARCHAR(4)) RETURNS INTEGER
```

```
BEGIN
```

```
    DECLARE X INTEGER;  
    SELECT DISTINCT COUNT(d.emp_no) AS st_v_skupini INTO X  
    FROM dept_emp d  
    WHERE d.dept_no = skupina;  
    RETURN X;
```

```
END //
```

```
DELIMITER ;
```

```

DELIMITER //

CREATE FUNCTION za_skupino_zensk (st_skupina VARCHAR(4)) RETURNS INTEGER
BEGIN
    DECLARE X INTEGER;
    SELECT DISTINCT COUNT(d.emp_no) AS st_z INTO X
    FROM employees e JOIN dept_emp d USING(emp_no)
    WHERE d.dept_no = st_skupina AND e.gender = 'F';
    RETURN X;
END //

DELIMITER ;

SELECT DISTINCT dept_no, IF(za_skupino_zensk(dept_no)/za_skupino(dept_no)
> povp_zensk(),"YES","NO") as "je vec kot povprecno"
FROM dept_emp
group by dept_no;

```

dept_no	je vec kot povprecno
d001	NO
d002	YES
d003	NO
d004	YES
d005	NO
d006	YES
d007	NO
d008	NO
d009	YES

\*Naredil sem funkcijo ki vrne število vseh zaposlenih, funkcijo, ki vrne delež vseh žensk. Funkcijo, ki prejme ime skupino in vrne število vseh zaposlenih te skupine in podobno funkcijo le za ženske.

- i) Izpišite imena zaposlenih, ki so delali natanko na enem oddelku in niso bili vodje (managers). Izpis uredite po naraščajoči starosti zaposlenega in imenu.

```
SELECT e.first_name, e.last_name
FROM employees e JOIN dept_emp de USING(emp_no)
WHERE e.emp_no NOT IN
      (SELECT emp_no
       FROM dept_manager)
GROUP BY de.emp_no
HAVING COUNT(de.dept_no) = 1
ORDER BY e.birth_date DESC, e.first_name;
```

first_name	last_name
Adamantios	Vanwelkenhuysen
Anneli	Pappas
Armond	Perly
Badri	Schapiro
Berni	Stranks
Chiranjit	Dredge
Dannz	Zhang
Deniz	Thibadeau
Divier	Ishibashi
Fun	Seiwald
Gopalakris...	Angel
Henk	Anger

- j) Imate naslednjo poizvedbo:

```
SELECT *
FROM salaries
WHERE from_date >= '1990-01-01'
ORDER BY from_date ASC;
```

Pospešite izvajanje te poizvedbe. Ustvarite potrebne strukture za pohitritev poizvedbe in primerjajte čase izvajanja. Ugotovite faktor pohitritve in v poročilo zapišite čas izvajanja pred in po pohitritvi ter faktor pohitritve. (za čas lahko uporabite ukaz `set profiling=1`; vaš `sql`; `show profiles`);).

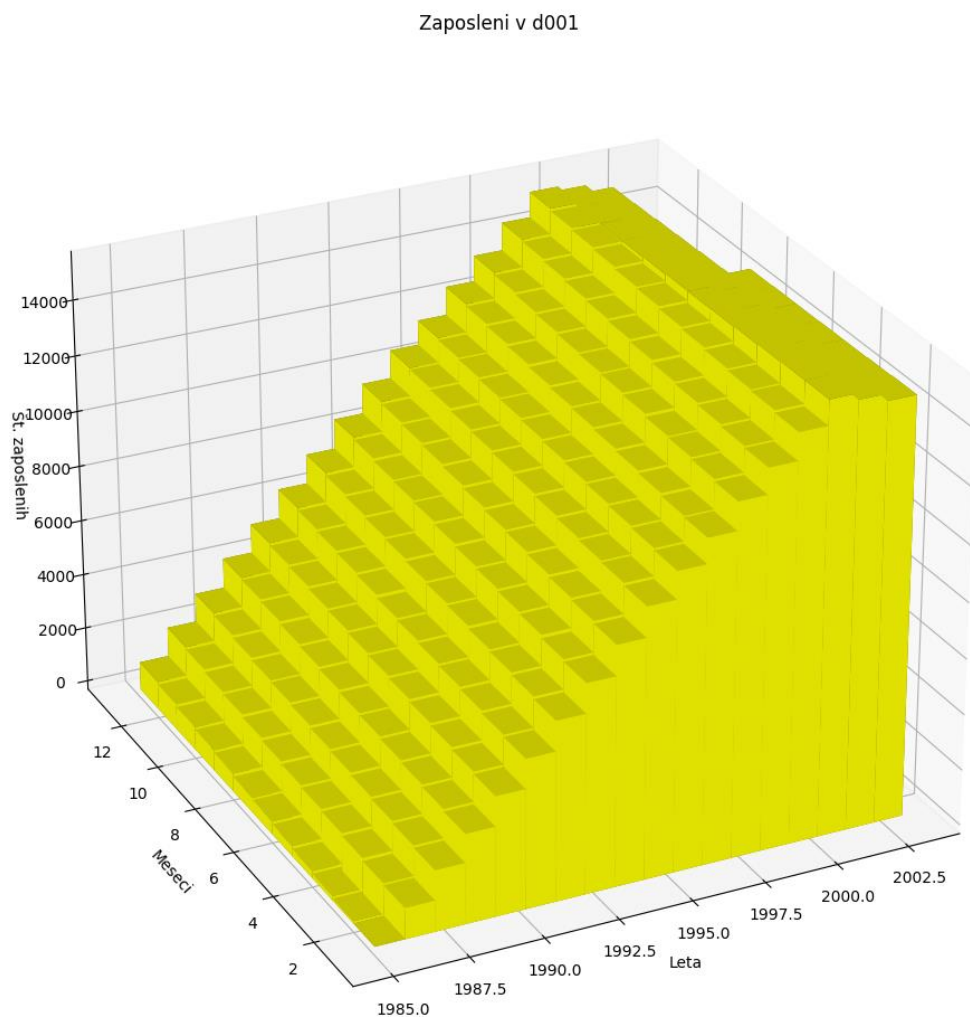
```
CREATE INDEX po_datumu
ON salaries (from_date);
```

0.00014080	set profiling=1
1.06422550	SELECT * FROM salaries WHERE from_date >= '1990-01-01' ORDER BY from_date ASC LIMIT 0, 1000
4.06630210	CREATE INDEX po_datumu ON salaries (from_date)
0.00915890	SELECT * FROM salaries WHERE from_date >= '1990-01-01' ORDER BY from_date ASC LIMIT 0, 1000

#### 4. + 5. (ODBC)

V python zbrani podatki iz baze Employees in preoblikovani v grafe, ki prikazujejo št. zaposlenih skozi leta podjetja. Za uporabniški vmesnik sem uporabil Tkinter, za prikaz grafa pa matplotlib.

Podatke iz baze shranim v slovar, iz katerega te podatke z numpy pretvorim v seznam, ki je ustrezne oblike za prikaz na grafu. Uporabniški vmesnik je narejen s Tkinterjem na katerem so za gumbi za vsakega od oddelkov in za celotno podjetje skupaj. S pritiskom na gumb se sproži funkcija za prikaz grafa v novem oknu.



*Primer grafa za oddelek d001*