SQL. Старт. Зима-весна 2024

Для решения сгенерированы данные при помощи Python (Приложение 1). Проверка решения проводилась <https://sqliteonline.com/> в СУБД PostgreSQL (Приложение 2).

Внимание проверяющим! В решение используются названия таблиц "Sfaff"и "Jobtitles" вместо Sfaff и Jobtitles.

1. Напишите запрос, с помощью которого можно найти дубли в поле email из таблицы “Sfaff”.

**SELECT email AS double\_email FROM "Staff"**

**GROUP BY email**

**HAVING COUNT(email) > 1;**

1. Напишите запрос, с помощью которого можно определить возраст каждого сотрудника из таблицы “Staff” на момент запроса.

**SELECT name,**

**EXTRACT(YEAR FROM AGE(CURRENT\_DATE, TO\_DATE(bithday, 'DD.MM.YYYY'))) AS years\_old**

**FROM "Staff"**

**WHERE bithday IS NOT NULL**

1. Напишите запрос, с помощью которого можно определить должность (Jobtitles.name) со вторым по величине уровнем зарплаты.

Решение 1.

**SELECT name FROM "Jobtitles"**

**WHERE jobtitle\_id =**

**(SELECT jobtitle\_id FROM "Staff"**

**WHERE salary is NOT NULL**

**ORDER BY salary DESC**

**OFFSET 1**

**LIMIT 1)**

Решение. Учитывает одинаковый уровень зарплаты у нескольких сотрудников.

**WITH cte AS (**

**SELECT salary, jobtitle\_id, dense\_rank() OVER(ORDER BY salary DESC) AS salary\_level**

**FROM "Staff"**

**WHERE salary IS NOT NULL)**

**SELECT name FROM "Jobtitles"**

**WHERE jobtitle\_id in (SELECT jobtitle\_id FROM cte WHERE salary\_level=2)**

Приложение 1.

Python get data script

import pandas as pd

from random import choice, randint

from string import ascii\_lowercase

from datetime import datetime as dt

data = pd.DataFrame()

n = 100

data["staff\_id"] = list(range(1, n+1))

name = ["James", "John", "Robert", "Michae",

"William", "Charles", "Richard", "Paul",

"Christopher", "David", "Donald", ]

surname = ["Ross", "Ramirez", "Sanders", "Bailey",

"Morris", "Gonzalez", "Hill", "Hernandez", "Walker" ]

data["name"] = [f"{choice(name)} {choice(surname)}" for \_ in range(n)]

data["salary"] = [randint(27\_894, 1\_047\_587) for \_ in range(n)]

def get\_email():

l = randint(7, 13)

txt = "".join(choice(ascii\_lowercase) for \_ in range(l))

return f"{txt}@{choice(domain)}"

domain = ["mail.ru", "ya.ru", "yandex.ru", "gmail.com", "rambler.ru", "bk.ru", "list.ru"]

data["email"] = [get\_email() for \_ in range(n)]

def get\_bithday():

start\_date = dt(randint(1985, 2005), randint(1,12), randint(1, 28))

return start\_date.strftime("%d.%m.%Y")

data["bithday"] = [get\_bithday() for \_ in range(n)]

data["jobtitle\_id"] = [randint(1, 7) for \_ in range(n)]

data\_2 = pd.DataFrame({"jobtitle\_id": list(range(1,8)),

"name": ["Developer", "System Analyst", "Project Manager",

"System Administrator", "Group Leader",

"Testing Engineer", "Support", ]

})

data\_2

data.to\_csv("Staff.csv", index=False)

data\_2.to\_csv("Jobtitles.csv", index=False)

Приложение 2.

Подготовка таблиц для запросов SQL.

CREATE TABLE "Jobtitles" ("jobtitle\_id" SMALLINT,"name" VARCHAR(20));

INSERT INTO "Jobtitles" ("jobtitle\_id","name") VALUES

('1','Developer'),

('2','System Analyst'),

('3','Project Manager'),

('4','System Administrator'),

('5','Group Leader'),

('6','Testing Engineer'),

('7','Support');

CREATE TABLE "Staff" ("staff\_id" SMALLINT,"name" VARCHAR(30),"salary" INTEGER,"email" VARCHAR(30),"bithday" VARCHAR(10),"jobtitle\_id" SMALLINT);

INSERT INTO "Staff" ("staff\_id","name","salary","email","bithday","jobtitle\_id") VALUES

('1','William Hernandez','616342','abfnnta@mail.ru','06.01.1986','6'),

('2','David Morris','747609','fwfswem@rambler.ru','08.03.1986','2'),

('3','Richard Ramirez','656008','wemxyzljgjhsv@bk.ru','20.03.2002','4'),

……………………………………………………………………………………………………………………………………………………….

('98','Paul Ross','977718','pqrmwyurjaap@list.ru','25.04.2003','4'),

('99','Christopher Hill','268727','wnuxswmldwbf@bk.ru','27.03.1992','7'),

('100','Michae Morris','245944','kseewamtqtm@list.ru','14.06.1997','2');

В генерации отсутствуют NULL, поэтому:

UPDATE "Staff"

SET email = NULL

WHERE staff\_id % 13 = 0;

SELECT \* FROM "Staff";

UPDATE "Staff"

SET bithday = NULL

WHERE (staff\_id-5) % 29 = 0;

UPDATE "Staff"

SET salary = NULL

WHERE (staff\_id+13) % 29 = 0;

В генерации отсутствуют дубли, поэтому:

UPDATE "Staff"

SET email = (SELECT email FROM "Staff" WHERE staff\_id=15 )

WHERE staff\_id IN (20,40);

UPDATE "Staff"

SET email = (SELECT email FROM "Staff" WHERE staff\_id=3 )

WHERE staff\_id = 50;