

- **Практическое задание по теме «Операторы, фильтрация, сортировка и ограничение»**
- **Практическое задание по теме «Агрегация данных»**

Оглавление

1	Практическое задание по теме «Операторы, фильтрация, сортировка и ограничение»	3
1.1	Заполните в таблице users поля created_at и updated_at текущими датой и временем.	3
1.1.1	Подготовка таблицы	3
1.1.2	Решение	4
1.2	Преобразовать поля к типу DATETIME, сохранив введенные ранее значения.	5
1.2.1	Подготовка таблицы	5
1.2.2	Решение	5
1.3	Отсортировать записи, чтобы они выводились в порядке увеличения значения value. Нулевые записи должны выводиться в конце.	7
1.3.1	Подготовка таблицы	7
1.3.2	Решение	8
1.4	Из таблицы users показать пользователей, родившихся в августе и мае.	9
1.4.1	Подготовка таблицы	9
1.4.2	Решение	9
1.5	Отсортируйте записи в порядке, заданном в списке IN.	10
1.5.1	Подготовка таблицы	10
1.5.2	Решение	10
2	Практическое задание по теме «Агрегация данных»	11
2.1	Подсчитайте средний возраст пользователей в таблице users.	11
2.1.1	Подготовка таблицы	11
2.1.2	Решение Вариант 1	11
2.1.3	Решение Вариант 2	11
2.2	Подсчитайте количество дней рождения, которые приходятся на каждый из дней недели.	12
2.2.1	Подготовка таблицы	12
2.2.2	Решение	12
2.2.3	Проверка:	13

2.3	(по желанию) Подсчитайте произведение чисел в столбце таблицы.	14
2.3.1	Подготовка таблицы	14
2.3.2	Решение	14

1 Практическое задание по теме «Операторы, фильтрация, сортировка и ограничение»

1.1 Заполните в таблице users поля created_at и updated_at текущими датой и временем.

Пусть в таблице users поля created_at и updated_at оказались незаполненными. Заполните их текущими датой и временем.

1.1.1 Подготовка таблицы

```
USE homework_5;
```

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(255) COMMENT 'Имя покупателя',  
  birthday_at DATE COMMENT 'Дата рождения',  
  created_at DATETIME DEFAULT NULL,  
  updated_at DATETIME DEFAULT NULL  
) COMMENT = 'Покупатели';
```

```
INSERT INTO users (name, birthday_at) VALUES  
( 'Геннадий', '1990-10-05'),  
( 'Наталья', '1984-11-12'),  
( 'Александр', '1985-05-20'),  
( 'Сергей', '1988-02-14'),  
( 'Иван', '1998-01-12'),  
( 'Мария', '1992-08-29');
```

```
SELECT * FROM users;
```

	id	name	birthday_at	created_at	updated_at
▶	1	Геннадий	1990-10-05	NULL	NULL
	2	Наталья	1984-11-12	NULL	NULL
	3	Александр	1985-05-20	NULL	NULL
	4	Сергей	1988-02-14	NULL	NULL
	5	Иван	1998-01-12	NULL	NULL
	6	Мария	1992-08-29	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL

1.1.2 Решение

UPDATE users SET

created_at = NOW(),

updated_at = NOW();

SELECT * FROM users;

	id	name	birthday_at	created_at	updated_at
▶	1	Геннадий	1990-10-05	2022-09-01 11:33:02	2022-09-01 11:33:02
	2	Наталья	1984-11-12	2022-09-01 11:33:02	2022-09-01 11:33:02
	3	Александр	1985-05-20	2022-09-01 11:33:02	2022-09-01 11:33:02
	4	Сергей	1988-02-14	2022-09-01 11:33:02	2022-09-01 11:33:02
	5	Иван	1998-01-12	2022-09-01 11:33:02	2022-09-01 11:33:02
	6	Мария	1992-08-29	2022-09-01 11:33:02	2022-09-01 11:33:02
•	NULL	NULL	NULL	NULL	NULL

1.2 Преобразовать поля к типу DATETIME, сохранив введённые ранее значения.

Таблица users была неудачно спроектирована. Записи created_at и updated_at были заданы типом VARCHAR и в них долгое время помещались значения в формате 20.10.2017 8:10. Необходимо преобразовать поля к типу DATETIME, сохранив введённые ранее значения.

1.2.1 Подготовка таблицы

```
USE homework_5;
```

```
DROP TABLE IF EXISTS users;  
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(255) COMMENT 'Имя покупателя',  
  birthday_at DATE COMMENT 'Дата рождения',  
  created_at VARCHAR(255) DEFAULT NULL,  
  updated_at VARCHAR(255) DEFAULT NULL  
) COMMENT = 'Покупатели';
```

```
INSERT INTO users (name, birthday_at, created_at, updated_at ) VALUES  
( 'Геннадий', '1990-10-05', '20.10.2017 8:10', '21.10.2017 8:10'),  
( 'Наталья', '1984-11-12', '20.10.2017 8:10', '22.10.2017 8:10'),  
( 'Александр', '1985-05-20', '20.10.2017 8:10', '23.10.2017 8:10'),  
( 'Сергей', '1988-02-14', '20.10.2017 8:10', '24.10.2017 8:10'),  
( 'Иван', '1998-01-12', '20.10.2017 8:10', '25.10.2017 8:10'),  
( 'Мария', '1992-08-29', '20.10.2017 8:10', '26.10.2017 8:10');
```

```
SELECT * FROM users;
```

	id	name	birthday_at	created_at	updated_at
▶	1	Геннадий	1990-10-05	20.10.2017 8:10	21.10.2017 8:10
	2	Наталья	1984-11-12	20.10.2017 8:10	22.10.2017 8:10
	3	Александр	1985-05-20	20.10.2017 8:10	23.10.2017 8:10
	4	Сергей	1988-02-14	20.10.2017 8:10	24.10.2017 8:10
	5	Иван	1998-01-12	20.10.2017 8:10	25.10.2017 8:10
	6	Мария	1992-08-29	20.10.2017 8:10	26.10.2017 8:10
●	NULL	NULL	NULL	NULL	NULL

1.2.2 Решение

```
ALTER TABLE users ADD created_at_dt DATETIME, updated_at_dt DATETIME;
```

	id	name	birthday_at	created_at	updated_at	created_at_new	updated_at_new
▶	1	Геннадий	1990-10-05	20.10.2017 8:10	21.10.2017 8:10	NULL	NULL
	2	Наталья	1984-11-12	20.10.2017 8:10	22.10.2017 8:10	NULL	NULL
	3	Александр	1985-05-20	20.10.2017 8:10	23.10.2017 8:10	NULL	NULL
	4	С Александр	1988-02-14	20.10.2017 8:10	24.10.2017 8:10	NULL	NULL
	5	Иван	1998-01-12	20.10.2017 8:10	25.10.2017 8:10	NULL	NULL
	6	Мария	1992-08-29	20.10.2017 8:10	26.10.2017 8:10	NULL	NULL
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

UPDATE users

SET created_at_new = STR_TO_DATE(created_at, '%d.%m.%Y %h:%i');

UPDATE users

SET updated_at_new = STR_TO_DATE(updated_at, '%d.%m.%Y %h:%i');

	id	name	birthday_at	created_at	updated_at	created_at_new	updated_at_new
▶	1	Геннадий	1990-10-05	20.10.2017 8:10	21.10.2017 8:10	2017-10-20 08:10:00	2017-10-21 08:10:00
	2	Наталья	1984-11-12	20.10.2017 8:10	22.10.2017 8:10	2017-10-20 08:10:00	2017-10-22 08:10:00
	3	Александр	1985-05-20	20.10.2017 8:10	23.10.2017 8:10	2017-10-20 08:10:00	2017-10-23 08:10:00
	4	Сергей	1988-02-14	20.10.2017 8:10	24.10.2017 8:10	2017-10-20 08:10:00	2017-10-24 08:10:00
	5	Иван	1998-01-12	20.10.2017 8:10	25.10.2017 8:10	2017-10-20 08:10:00	2017-10-25 08:10:00
	6	Мария	1992-08-29	20.10.2017 8:10	26.10.2017 8:10	2017-10-20 08:10:00	2017-10-26 08:10:00
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM users;

ALTER TABLE users DROP created_at;

ALTER TABLE users DROP updated_at;

ALTER TABLE users RENAME COLUMN created_at_new TO created_at;

ALTER TABLE users RENAME COLUMN updated_at_new TO updated_at;

SELECT * FROM users;

	id	name	birthday_at	created_at	updated_at
▶	1	Геннадий	1990-10-05	2017-10-20 08:10:00	2017-10-21 08:10:00
	2	Наталья	1984-11-12	2017-10-20 08:10:00	2017-10-22 08:10:00
	3	Александр	1985-05-20	2017-10-20 08:10:00	2017-10-23 08:10:00
	4	Сергей	1988-02-14	2017-10-20 08:10:00	2017-10-24 08:10:00
	5	Иван	1998-01-12	2017-10-20 08:10:00	2017-10-25 08:10:00
	6	Мария	1992-08-29	2017-10-20 08:10:00	2017-10-26 08:10:00
•	NULL	NULL	NULL	NULL	NULL

1.3 Отсортировать записи, чтобы они выводились в порядке увеличения значения value. Нулевые записи должны выводиться в конце.

В таблице складских запасов storehouses_products в поле value могут встречаться самые разные цифры: 0, если товар закончился и выше нуля, если на складе имеются запасы. Необходимо отсортировать записи таким образом, чтобы они выводились в порядке увеличения значения value. Однако нулевые записи должны выводиться в конце, после всех записей.

value		value
0		1
2500		30
0	→	500
30		2500
500		0
1		0

1.3.1 Подготовка таблицы

```
USE homework_5;
```

```
DROP TABLE IF EXISTS products;  
CREATE TABLE products (  
  id SERIAL PRIMARY KEY,  
  prod_name CHAR(100),  
  value INT UNSIGNED  
) comment 'table of products';
```

```
INSERT INTO products (prod_name, value ) VALUES  
('item1', 0),  
('item2', 2500),  
('item3', 0),  
('item4', 30),  
('item5', 500),  
('item6', 1)  
;
```

```
SELECT * FROM products;
```

	id	prod_name	value
▶	1	item1	0
	2	item2	2500
	3	item3	0
	4	item4	30
	5	item5	500
	6	item6	1

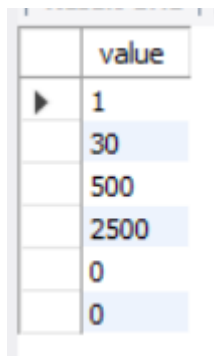
1.3.2 Решение

-- Variant 1

```
SELECT value FROM products ORDER BY IF( value = 0, 1, 0 ) , value;
```

-- Variant 2

```
SELECT value FROM products ORDER BY value = 0, value;
```



	value
▶	1
	30
	500
	2500
	0
	0

1.4 Из таблицы users показать пользователей, родившихся в августе и мае.

(по желанию) Из таблицы users необходимо извлечь пользователей, родившихся в августе и мае. Месяцы заданы в виде списка английских названий (may, august)

1.4.1 Подготовка таблицы

```
USE homework_5;
```

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(255) COMMENT 'Имя покупателя',  
  birthday_at char(20) COMMENT 'Месяц рождения'  
) COMMENT = 'Люди';
```

```
INSERT INTO users (name, birthday_at ) VALUES
```

```
('Gena', 'January'),  
( 'Nataly', 'February'),  
( 'Alex', 'March'),  
( 'Serge', 'April'),  
( 'Ivan', 'May'),  
( 'Santa-Maria', 'June'),  
( 'Vladimir', 'July'),  
( 'Mikhail', 'August'),  
( 'Dmitriy', 'September'),  
( 'Andrey', 'October');
```

```
SELECT * FROM users;
```

	id	name	birthday_at
▶	1	Gena	January
	2	Nataly	February
	3	Alex	March
	4	Serge	April
	5	Ivan	May
	6	Santa-Maria	June
	7	Vladimir	July
	8	Mikhail	August
	9	Dmitriy	September
	10	Andrey	October

1.4.2 Решение

```
SELECT * FROM users WHERE birthday_at IN ('May', 'August');
```

	id	name	birthday_at
▶	5	Ivan	May
	8	Mikhail	August
•	NULL	NULL	NULL

1.5 Отсортируйте записи в порядке, заданном в списке IN.

(по желанию) Из таблицы catalogs извлекаются записи при помощи запроса. SELECT * FROM catalogs WHERE id IN (5, 1, 2); Отсортируйте записи в порядке, заданном в списке IN.

1.5.1 Подготовка таблицы

```
USE homework_5;
```

```
DROP TABLE IF EXISTS catalogs;  
CREATE TABLE catalogs (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(255) COMMENT 'Название раздела',  
  UNIQUE unique_name(name(10))  
) COMMENT = 'Разделы интернет-магазина';
```

```
INSERT INTO catalogs VALUES  
(NULL, 'Процессоры'),  
(NULL, 'Материнские платы'),  
(NULL, 'Видеокарты'),  
(NULL, 'Жесткие диски'),  
(NULL, 'Оперативная память');
```

```
SELECT * FROM catalogs;
```

	id	name
▶	1	Процессоры
	2	Материнские платы
	3	Видеокарты
	4	Жесткие диски
	5	Оперативная память
•	NULL	NULL

1.5.2 Решение

```
SELECT  
  id, name  
FROM  
  catalogs  
WHERE  
  id IN (5, 1, 2)  
ORDER BY  
  FIELD(id, 5, 1, 2);
```

	id	name
▶	5	Оперативная память
	1	Процессоры
	2	Материнские платы
•	NULL	NULL

2 Практическое задание теме «Агрегация данных»

2.1 Подсчитайте средний возраст пользователей в таблице users.

2.1.1 Подготовка таблицы

```
USE homework_5;
```

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(255) COMMENT 'Имя покупателя',  
  birthday_at DATE COMMENT 'Дата рождения',  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,  
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
) COMMENT = 'Покупатели';
```

```
INSERT INTO users (name, birthday_at) VALUES
```

```
('Геннадий', '1990-10-05'),  
( 'Наталья', '1984-11-12'),  
( 'Александр', '1985-05-20'),  
( 'Сергей', '1988-02-14'),  
( 'Иван', '1998-01-12'),  
( 'Мария', '1992-08-29');
```

```
SELECT * FROM users;
```

	id	name	birthday_at	created_at	updated_at
▶	1	Геннадий	1990-10-05	2022-09-01 15:37:04	2022-09-01 15:37:04
	2	Наталья	1984-11-12	2022-09-01 15:37:04	2022-09-01 15:37:04
	3	Александр	1985-05-20	2022-09-01 15:37:04	2022-09-01 15:37:04
	4	Сергей	1988-02-14	2022-09-01 15:37:04	2022-09-01 15:37:04
	5	Иван	1998-01-12	2022-09-01 15:37:04	2022-09-01 15:37:04
	6	Мария	1992-08-29	2022-09-01 15:37:04	2022-09-01 15:37:04
*	NULL	NULL	NULL	NULL	NULL

2.1.2 Решение Вариант 1

```
SELECT ROUND(AVG((TO_DAYS(NOW()) - TO_DAYS(birthday_at)) / 365.25), 0) AS AVG_Age  
FROM users;
```

	AVG_Age
▶	33

2.1.3 Решение Вариант 2

```
SELECT ROUND(AVG(TIMESTAMPDIFF(YEAR, birthday_at, NOW())) , 0) AS AVG_Age FROM users;
```

	AVG_Age
▶	32

2.2 Подсчитайте количество дней рождения, которые приходятся на каждый из дней недели

Подсчитайте количество дней рождения, которые приходятся на каждый из дней недели. Следует учесть, что необходимы дни недели текущего года, а не года рождения.

2.2.1 Подготовка таблицы

```
USE homework_5;
```

```
DROP TABLE IF EXISTS users;
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  name VARCHAR(255) COMMENT 'Имя покупателя',
  birthday_at DATE COMMENT 'Дата рождения',
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
) COMMENT = 'Покупатели';
```

```
INSERT INTO users (name, birthday_at) VALUES
('Геннадий', '1990-10-05'),
('Наталья', '1984-11-12'),
('Александр', '1985-05-20'),
('Сергей', '1988-02-14'),
('Иван', '1998-01-12'),
('Мария', '1992-08-29');
```

```
SELECT * FROM users;
```

	id	name	birthday_at	created_at	updated_at
▶	1	Геннадий	1990-10-05	2022-09-01 15:54:16	2022-09-01 15:54:16
	2	Наталья	1984-11-12	2022-09-01 15:54:16	2022-09-01 15:54:16
	3	Александр	1985-05-20	2022-09-01 15:54:16	2022-09-01 15:54:16
	4	Сергей	1988-02-14	2022-09-01 15:54:16	2022-09-01 15:54:16
	5	Иван	1998-01-12	2022-09-01 15:54:16	2022-09-01 15:54:16
	6	Мария	1992-08-29	2022-09-01 15:54:16	2022-09-01 15:54:16
*	NULL	NULL	NULL	NULL	NULL

2.2.2 Решение

```
SELECT
  DAYNAME(CONCAT(YEAR(NOW()), '-', SUBSTRING(birthday_at, 6, 10))) AS week_day,
  COUNT(*) AS number_of_birthdays
FROM
  users
GROUP BY
  week_day
ORDER BY
  number_of_birthdays DESC;
```

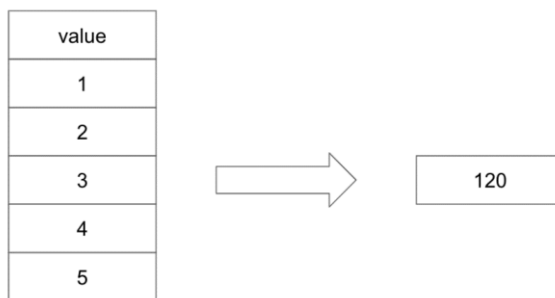
	week_day	number_of_birthdays
▶	Wednesday	2
	Monday	2
	Saturday	1
	Friday	1

2.2.3 Проверка:

```
-- Script for checking
SELECT
    name, birthday_at,
    CONCAT(YEAR(NOW()), '-', SUBSTRING(birthday_at, 6, 10)) AS birthday_in_this_Year,
    DAYNAME(CONCAT(YEAR(NOW()), '-', SUBSTRING(birthday_at, 6, 10))) AS week_day
FROM
    users
ORDER BY
    week_day;
```

	name	birthday_at	birthday_in_this_Year	week_day
▶	Александр	1985-05-20	2022-05-20	Friday
	Сергей	1988-02-14	2022-02-14	Monday
	Мария	1992-08-29	2022-08-29	Monday
	Наталья	1984-11-12	2022-11-12	Saturday
	Геннадий	1990-10-05	2022-10-05	Wednesday
	Иван	1998-01-12	2022-01-12	Wednesday

2.3 (по желанию) Подсчитайте произведение чисел в столбце таблицы.



2.3.1 Подготовка таблицы

```
CREATE TABLE integers (  
  value SERIAL PRIMARY KEY  
);
```

```
INSERT INTO integers VALUES  
(NULL),  
(NULL),  
(NULL),  
(NULL),  
(NULL);
```

```
SELECT * FROM digits;
```

	value
▶	1
	2
	3
	4
	5
⚙	NULL

2.3.2 Решение

Логарифм произведения равен сумме логарифмов

```
SELECT ROUND(exp(SUM(ln(value))), 0) AS result FROM digits;
```

	result
▶	120