

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

Задание 1. Выполнено. Добавил поле `updated_at` в таблицу `users`. Заполнил поля `created_at` и `updated_at` в таблице `users` текущей датой и временем.

```
ALTER TABLE users ADD COLUMN updated_at DATETIME NULL AFTER created_at;

UPDATE users
SET
    created_at=NOW(),
    updated_at=NOW();

SELECT * FROM users;
```

users 1

SELECT * FROM users

	birthday	photo_id	hometown	password_hash	created_at	updated_at
1	1986-05-20	[NULL]	Dickensfurt	cac98dc926f4964a11af5fc9c230f29c83c3a77	2021-06-12 03:06:28	2021-06-12 03:06:28
2	2011-08-14	[NULL]	Daronstad	d3e7c4903876eef15fcd43a2430976b969631	2021-06-12 03:06:28	2021-06-12 03:06:28
3	2019-07-18	[NULL]	Port Kaitlin	c6a05a438ed93439010945e1a74d98375b28e	2021-06-12 03:06:28	2021-06-12 03:06:28
4	2007-12-30	[NULL]	Danykastad	1e36fc055794258a677bad389b2874828cd3e	2021-06-12 03:06:28	2021-06-12 03:06:28
5	2014-11-13	[NULL]	Zettaville	7fa67e066fa36b5f553170dc76f67a1343bc73	2021-06-12 03:06:28	2021-06-12 03:06:28
6	1972-12-10	[NULL]	South Forestview	a500ed6650216b7c23c5c2eb25fecff54dd6a4	2021-06-12 03:06:28	2021-06-12 03:06:28
7	1997-04-30	[NULL]	D'Amorefort	13c57cd1f944b234ed654779a10f97138c620f	2021-06-12 03:06:28	2021-06-12 03:06:28
8	2012-01-14	[NULL]	Alfonzobury	eace9b757daa893627ac4bf6bdea9170f9f5d	2021-06-12 03:06:28	2021-06-12 03:06:28
9	1988-03-02	[NULL]	Kuhicfurt	9f3ce296eef78c25327801251edc8a5b03b05	2021-06-12 03:06:28	2021-06-12 03:06:28
10	1975-06-26	[NULL]	Lilianemouth	9aac6e82a36c54768a814f1ed26330b4b73ee	2021-06-12 03:06:28	2021-06-12 03:06:28

Задание 2. Выполнено. Создал дополнительную таблицу `users2` для выполнения задания. Поля `created_at` и `updated_at` задал типом `VARCHAR` и помещал значения в формате `20.10.2017 8:10`. Затем преобразовал поля к типу `DATETIME`, сохранив введённые ранее значения.

```
-- Create initial table with data

CREATE TABLE IF NOT EXISTS users2 AS SELECT id, firstname, lastname, mail, phone, gender, birthday FROM users;

ALTER TABLE users2
ADD COLUMN created_at VARCHAR(255) AFTER birthday;

ALTER TABLE users2
ADD COLUMN updated_at VARCHAR(255) AFTER created_at;

UPDATE users2
SET
    created_at = '20.10.2017 8:10',
    updated_at = '01.11.2017 12:05';

-- Change data in the table

UPDATE users2
SET created_at = STR_TO_DATE(created_at, '%d.%m.%Y %k:%i');

ALTER TABLE users2
MODIFY COLUMN created_at DATETIME;

UPDATE users2
SET updated_at = STR_TO_DATE(updated_at, '%d.%m.%Y %k:%i');

ALTER TABLE users2
MODIFY COLUMN updated_at DATETIME;

SELECT * FROM users2;
```

users2 1

SELECT * FROM users2

	firstname	lastname	mail	phone	gender	birthday	created_at	updated_at
1	da	Jenkins	ljerde@example.org	374	f	1986-05-20	2017-10-20 08:10:00	2017-11-01 12:05:00
2	rda	Blick	xabbott@example.or	833	m	2011-08-14	2017-10-20 08:10:00	2017-11-01 12:05:00
3	a	Prosacco	raynor.matteo@exa	1	m	2019-07-18	2017-10-20 08:10:00	2017-11-01 12:05:00
4	rbin	Hyatt	braxton.maggio@exa	929,338	f	2007-12-30	2017-10-20 08:10:00	2017-11-01 12:05:00
5	adison	Welch	floy.cruickshank@exa	9,366,410,464	f	2014-11-13	2017-10-20 08:10:00	2017-11-01 12:05:00

Задание 3. Выполнено. Создал отдельную таблицу `storehouses_products`, заполнил её. Отсортировал записи таким образом, чтобы они выводились в порядке увеличения значения `value`, а нулевые запасы в конце после всех записей.

SQL Editor:

```
CREATE TABLE IF NOT EXISTS storehouses_products (
  id SERIAL PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  value BIGINT
);

INSERT INTO storehouses_products
VALUES (1, 'product_1', 100), (2, 'product_2', 50), (3, 'product_3', 0), (4, 'product_4', 0), (5, 'product_5', 200);

SELECT * FROM storehouses_products
ORDER BY IF(`value` > 0, 0, 1), `value`;
```

storehouses_products 1

SQL Editor:

```
SELECT * FROM storehouses_products ORDER BY IF(`value` > 0, 0, 1), `value`;
```

	id	name	value
1	2	product_2	50
2	1	product_1	100
3	5	product_5	200
4	3	product_3	0
5	4	product_4	0

Задание 4. Выполнено. Создал запрос на извлечение пользователей, родившихся в августе и мае. Месяцы задавал в виде списка английских названий (may, august).

SQL Editor:

```
SELECT * FROM users
WHERE MONTHNAME(birthday) IN('may', 'august');
```

users 1

SQL Editor:

```
SELECT * FROM users WHERE MONTHNAME(birthday) IN('may', 'august');
```

	id	firstname	lastname	mail	phone	gender	birthday
1	1,001	Lyda	Jenkins	ljerde@example.org	374	f	1986-05-20
2	1,002	Verda	Blick	xabbott@example.org	833	m	2011-08-14
3	1,024	Wallace	Kirlin	adolph40@example.org	342	f	2002-08-05
4	1,034	Alfredo	Nienow	wyman.lily@example.org	39	f	1984-05-04
5	1,036	Korey	Hirthe	xmetz@example.org	111	m	1979-05-20
6	1,041	Nathaniel	Swift	haleigh.stark@example.org	511	f	2002-05-22
7	1,051	Verdie	Prosacco	hoyt.hauck@example.org	61	m	2011-05-16
8	1,061	Davonte	Wolff	layla.hessel@example.org	4,513,381,642	m	1983-05-12
9	1,062	Roy	Bechtelar	tommie49@example.org	73	m	2012-05-23
10	1,064	Ezequiel	Effertz	modesta50@example.org	8,532,246,972	f	2019-08-14
11	1,067	Sammie	Hudson	mberge@example.org	5,214,309,869	m	1996-08-31
12	1,079	Kelton	Mills	noelia05@example.org	31	f	2001-05-08
13	1,104	Nils	Farrell	eudora78@example.org	10	m	2013-08-30
14	1,113	Linda	Bradtke	gmaggio@example.org	876,288	m	1998-08-13
15	1,125	Velda	Wilderman	claudia64@example.org	497,855	f	1973-05-18
16	1,131	Charlotte	Will	alejandra27@example.org	105,737	m	1986-05-11
17	1,134	Abby	Rohan	ethyl76@example.org	50	f	1991-05-03
18	1,137	Arden	Olson	nlowe@example.org	227	m	1974-08-10
19	1,138	Alvena	Krajcik	wgleason@example.org	938,276	f	1976-05-10
20	1,140	Cayla	Fisher	ibeier@example.org	58	f	2020-05-08
21	1,143	Keven	Erdman	runte.burnice@example.org	517	m	2001-08-16
22	1,147	Trisha	Steuber	rosemarie.koss@example.org	35	m	2003-05-13
23	1,150	Gisselle	Bartell	edwin.jacobs@example.org	220	m	1987-05-29

Задание 5. Выполнено. Создал временную таблицу catalogs, заполнил её. Отсортировал записи в порядке, заданном в списке IN при выполнении запроса SELECT * FROM catalogs WHERE id IN (5, 1, 2).

```
CREATE TABLE IF NOT EXISTS catalogs (
  id SERIAL PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  value BIGINT
);

INSERT INTO catalogs
VALUES (1, 'name_1', 100), (2, 'name_2', 50), (3, 'name_3', 0), (4, 'name_4', 0), (5, 'name_5', 200);

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

catalogs 1

SELECT * FROM catalogs WHERE id IN (5, 1, 2)

Grid	id	name	value
1	5	name_5	200
2	1	name_1	100
3	2	name_2	50

Практическое задание теме «Агрегация данных»

Задание 1. Выполнено. Подсчитал средний возраст пользователей в таблице users.

```
SELECT AVG(TIMESTAMPDIFF(YEAR, birthday, NOW())) FROM users;
```

Results 1

SELECT AVG(TIMESTAMPDIFF(YEAR, birthday, NOW()))

Grid	AVG(TIMESTAMPDIFF(YEAR, birthday, NOW()))
1	24.7898

Задание 2. Выполнено. Подсчитал количество дней рождения, которые приходятся на каждый из дней недели. Учел, что необходимы дни недели текущего года, а не года рождения.

```
SELECT
  DAYOFWEEK(DATE(CONCAT_WS('-', YEAR(NOW()), MONTH(birthday), DAY(birthday)))) AS day_week,
  COUNT(*)
FROM users
GROUP BY day_week
ORDER BY day_week;
```

Results 1

SELECT DAYOFWEEK(DATE(CONCAT_WS('-', YEAR(NOW()), MONTH(birthday), DAY(birthday)))) AS day_week, COUNT(*)

Grid	day_week	COUNT(*)
1	1	76
2	2	73
3	3	70
4	4	78
5	5	59
6	6	71
7	7	82

Задание 3. Выполнено. Создал тестовую таблицу и заполнил данными. Подсчитал произведение чисел в соответствующем столбце таблицы. Реализовал два варианта расчета. Учитывается знак чисел и возможность присутствия нулевых значений.

```
CREATE TABLE table_test (  
    id SERIAL PRIMARY KEY,  
    value BIGINT  
);  
  
INSERT INTO table_test  
VALUES (1,2), (2,3), (3,5), (4,5), (5,6);  
  
UPDATE table_test  
SET value = -2  
WHERE id = 1;  
  
-- Variant 1  
  
/*SET @PRODUCT = 1;  
SELECT @PRODUCT := @PRODUCT * value  
FROM table_test;*/  
  
-- Variant 2  
  
SELECT  
    IF(SUM(IF(value = 0, 0, NULL)) = 0,0,1) *  
    IF(SUM(IF(value < 0, 1, 0))%2 = 0,1,-1) *  
    EXP(SUM(LN(IF(value > 0, value, -value)))) AS PRODUCT  
FROM table_test;
```

Results 1

SELECT IF(SUM(IF(value = 0, 0, NULL)) = 0,0,1) Enter a SQL expression to filter results (use Ctrl+Space)

Grid	123 PRODUCT
1	-900