1. Задание. Пусть в таблице users поля created\_at и updated\_at оказались незаполненными. Заполните их текущими датой и временем.

CREATE DATABASE users;

USE users;

CREATE TABLE users (

id INT UNSIGNED NOT NULL PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255),

created\_at DATETIME,

updated\_at DATETIME

) ;

SHOW databases;

SHOW TABLES;

SELECT \* FROM users;

INSERT INTO users (name) VALUES ('Геннадий'), ('Наталья'), ('Александр'), ('Сергей'), ('Иван'), ('Мария');

SELECT \* FROM users;

UPDATE users SET created\_at = NOW(), updated\_at = NOW();

SELECT \* FROM users;

1. Задание. 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 введеные ранее значения.

CREATE DATABASE users;

USE users;

CREATE TABLE users (

id INT UNSIGNED NOT NULL PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255),

created\_at VARCHAR(255),

updated\_at VARCHAR(255)

) ;

SHOW databases;

SHOW TABLES;

SELECT \* FROM users;

INSERT INTO users (name, created\_at, updated\_at) VALUES ('Геннадий', '20.10.2017 8:10', '21.11.2017 8:10'), ('Наталья', '22.12.2017 8:10', '23.01.2018 8:10'), ('Александр', '24.02.2017 8:10', '25.03.2017 8:10'), ('Сергей', '26.04.2018 8:10', '27.05.2018 8:10'), ('Иван', '28.06.2017 8:10', '29.07.2017 8:10'), ('Мария', '30.08.2019 8:10', '30.09.2019 8:10');

SELECT \* FROM users;

\*\*\* Мой вариант решения

CREATE TABLE users\_temp (

id INT UNSIGNED NOT NULL PRIMARY KEY AUTO\_INCREMENT,

created\_at VARCHAR(255),

updated\_at VARCHAR(255)

) ;

INSERT INTO users\_temp (id, created\_at, updated\_at) SELECT id, created\_at, updated\_at FROM users;

UPDATE users SET created\_at = NULL;

UPDATE users SET updated\_at = NULL;

ALTER TABLE users CHANGE created\_at created\_at DATETIME;

ALTER TABLE users CHANGE updated\_at updated\_at DATETIME;

DESCRIBE users;

UPDATE users SET created\_at = (SELECT STR\_TO\_DATE(created\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=1) WHERE id=1;

UPDATE users SET created\_at = (SELECT STR\_TO\_DATE(created\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=2) WHERE id=2;

UPDATE users SET created\_at = (SELECT STR\_TO\_DATE(created\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=3) WHERE id=3;

UPDATE users SET created\_at = (SELECT STR\_TO\_DATE(created\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=4) WHERE id=4;

UPDATE users SET created\_at = (SELECT STR\_TO\_DATE(created\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=5) WHERE id=5;

UPDATE users SET created\_at = (SELECT STR\_TO\_DATE(created\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=6) WHERE id=6;

UPDATE users SET created\_at = (SELECT STR\_TO\_DATE(created\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=7) WHERE id=7;

UPDATE users SET updated\_at = (SELECT STR\_TO\_DATE(updated\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=1) WHERE id=1;

UPDATE users SET updated\_at = (SELECT STR\_TO\_DATE(updated\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=2) WHERE id=2;

UPDATE users SET updated\_at = (SELECT STR\_TO\_DATE(updated\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=3) WHERE id=3;

UPDATE users SET updated\_at = (SELECT STR\_TO\_DATE(updated\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=4) WHERE id=4;

UPDATE users SET updated\_at = (SELECT STR\_TO\_DATE(updated\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=5) WHERE id=5;

UPDATE users SET updated\_at = (SELECT STR\_TO\_DATE(updated\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=6) WHERE id=6;

UPDATE users SET updated\_at = (SELECT STR\_TO\_DATE(updated\_at, '%d.%m.%Y %H:%i') AS created\_at FROM users\_temp WHERE id=7) WHERE id=7;

понимаю что как-то длинно и много повторов, но работает.

улучшение пока не придумала.

\*\*\*Правильный вариант решения

UPDATE

users

SET

created\_at = STR\_TO\_DATE(created\_at, '%d.%m.%Y %k:%i'),

updated\_at = STR\_TO\_DATE(updated\_at, '%d.%m.%Y %k:%i');

ALTER TABLE

users

CHANGE

created\_at created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP;

ALTER TABLE

users

CHANGE

updated\_at updated\_at DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP;

DESCRIBE users;

1. Задание.

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 запасы должны выводиться в конце, после всех записей.

CREATE DATABASE shop;

USE shop;

CREATE TABLE storehouses\_products (

id INT UNSIGNED NOT NULL PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255),

value INT

) ;

SHOW databases;

SHOW TABLES;

SELECT \* FROM storehouses\_products;

INSERT INTO storehouses\_products (name, value) VALUES ('potato', 30), ('tomato', 2500), ('cucumber', 0), ('apple', 500), ('cherry', 0), ('orange', 1);

SELECT \* FROM storehouses\_products;

\*\*\* 1 вариант

SELECT \* FROM storehouses\_products ORDER BY IF(value > 0, 0, 1), value;

\*\*\* 2 вариант

SELECT \* FROM storehouses\_products ORDER BY value = 0, value;

1. Задание.

Из таблицы users необходимо извлечь пользователей, родившихся в

-- августе и мае. Месяцы заданы в виде списка английских названий ('may', 'august')

SELECT name FROM users WHERE DATE\_FORMAT(birthday\_at, '%M') IN ('may', 'august');

1. Задание.

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.

INSERT INTO catalogs VALUES

(NULL, 'Процессоры'),

(NULL, 'Материнские платы'),

(NULL, 'Видеокарты'),

(NULL, 'Жесткие диски'),

(NULL, 'Оперативная память');

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

1. Задание.

Подсчитайте средний возраст пользователей в таблице users

SELECT AVG(TIMESTAMPDIFF(YEAR, birthday\_at, NOW())) AS age FROM users;

1. Задание.

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

SELECT DATE\_FORMAT(DATE(CONCAT\_WS('-', YEAR(NOW()), MONTH(birthday\_at), DAY(birthday\_at))), '%W') AS day, COUNT(\*) AS total FROM users GROUP BY day ORDER BY total DESC;

1. Задание.

Подсчитайте произведение чисел в столбце таблицы

CREATE DATABASE math;

USE math;

CREATE TABLE value (id INT UNSIGNED NOT NULL PRIMARY KEY AUTO\_INCREMENT, value INT) ;

SHOW databases;

SHOW TABLES;

SELECT \* FROM value;

INSERT INTO value (value) VALUES (1), (2), (3), (4), (5);

SELECT \* FROM value;

SELECT ROUND(EXP(SUM(LN(value)))) FROM value;