

КАФЕДРА №

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

---

должность, уч. степень, звание

---

подпись, дата

---

инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

Триггеры. Обеспечение активной целостности данных базы данных  
по дисциплине: Проектирование баз данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

---

подпись, дата

---

инициалы, фамилия

Санкт-Петербург  
2023

**Цель работы** - По аналогии с примерами, приведенными в п. 1 реализовать запросы г) .. ж), указанные в варианте задания.

Один из запросов на максимум/минимум реализовать с помощью директивы all

Запрос на «все» (реляционное деление) реализовать с помощью 2 not exists

Запросы на разность реализовать в 3 вариантах: Not in, except (MySQL не поддерживает, поэтому только синтаксис), с использованием левого/правого соединения

**Задание (13 вариант):**

калькулятор бюджета семьи: категория дохода (продажа, зарплата), категория расхода (еда, счета за КУ, здоровье ...), статьи дохода и расхода, дата расхода/дохода. Категория - более общее понятие чем статья. Например категория - еда, а статьи в ней мясо, рыба, вкусное к чаю, а конкретный расход «печенье «курабье» к чаю 11.09»

а. расходы всех категорий, которые содержат часть слова «транспорт», но не заканчиваются на него

б. месяц, в котором были статьи дохода от работы и от дарения

в. пользователь без доходов

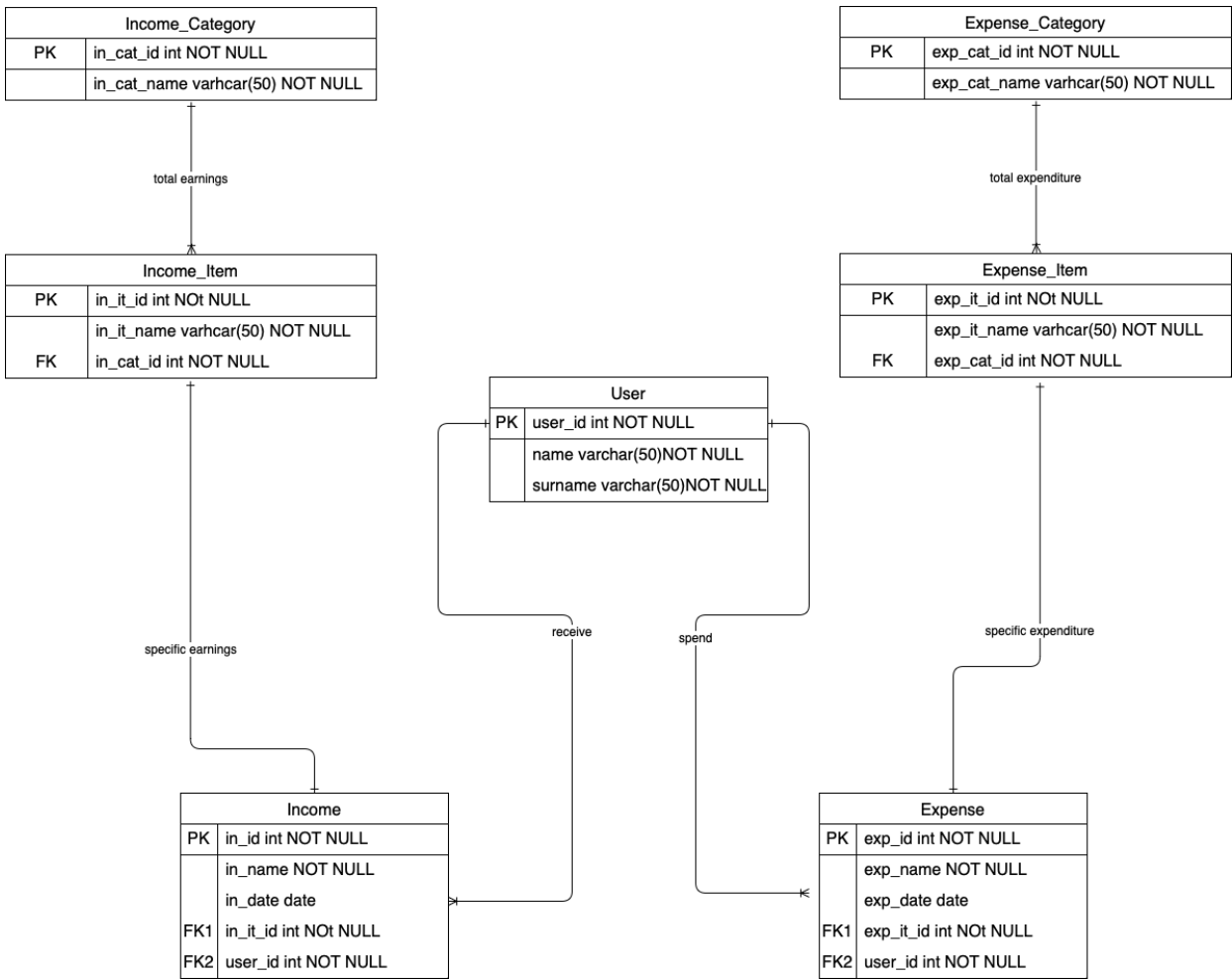
г. категория, по которой были расходы в этом году максимальные по сумме

д. категория, по которой не было расходов в январе, но были в мае

е. категория расхода, по которой траты были у всех членов семьи

ж. месяц, в котором были расходы максимального количества статей

Физическая модель предметной области:



# 1. BEFORE INSERT триггер для проверки ссылочной целостности на таблице Income

Этот триггер будет проверять перед вставкой записи в таблицу Income, что user\_id и in\_it\_id ссылаются на существующие записи в таблицах User и Income\_Item соответственно. Если ссылки на записи не существуют, триггер прервет вставку и выведет сообщение об ошибке.

```
DELIMITER //
```

```
CREATE TRIGGER before_insert_income
```

```
BEFORE INSERT ON Income
```

```
FOR EACH ROW
```

```
BEGIN
```

```
-- Объявляем переменные для проверки существования пользователя и категории
```

```
DECLARE user_exists INT DEFAULT 0;
```

```
DECLARE item_exists INT DEFAULT 0;
```

```
-- Проверяем, существует ли пользователь с данным user_id
```

```
SELECT COUNT(*) INTO user_exists FROM User WHERE user_id = NEW.user_id;
```

```
-- Проверяем, существует ли пункт дохода с данным in_it_id
```

```
SELECT COUNT(*) INTO item_exists FROM Income_Item WHERE in_it_id =  
NEW.in_it_id;
```

```
-- Если пользователь или категория не найдены, выдаем ошибку
```

```
IF user_exists = 0 OR item_exists = 0 THEN
```

```
    SIGNAL SQLSTATE '45000'
```

```
    SET MESSAGE_TEXT = 'Ошибка целостности ссылок: пользователь или категория  
не существует.';
```

```
END IF;
```

```
END;
```

```
//
```

DELIMITER ;

DROP TRIGGER before\_insert\_income;

INSERT INTO Income (in\_id, in\_name, in\_date, in\_money, in\_it\_id, user\_id)

VALUES (6, 'Доход Один', '2023-04-01', 45000.00, 1, 1);

DELETE FROM Income WHERE in\_id = 6;

INSERT INTO Income (in\_name, in\_date, in\_money, in\_it\_id, user\_id)

VALUES ('Неверный доход', '2023-04-01', 50000.00, 999, 999);

|  | in_id | in_name                | in_date    | in_money | in_it_id | user_id |  |
|--|-------|------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра    | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу        | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены          | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра     | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | NULL  | NULL                   | NULL       | NULL     | NULL     | NULL    |  |

|  | in_id | in_name                | in_date    | in_money | in_it_id | user_id |  |
|--|-------|------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра    | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу        | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены          | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра     | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | 6     | Доход Один             | 2023-04-01 | 45000.00 | 1        | 1       |  |
|  | NULL  | NULL                   | NULL       | NULL     | NULL     | NULL    |  |

|     | Time     | Action  | Response   |
|-----|----------|---|--|
| 574 | 02:28:29 | INSERT INTO Income (in_id, in_name, in_date, in_money, in_it_id, user_id) VALUES (6, 'До... | 1 row(s) affected  |
| 575 | 02:28:33 | SELECT * FROM Income LIMIT 0, 1000  | 6 row(s) returned  |
| 576 | 02:28:55 | DELETE FROM Income WHERE in_id = 6  | 1 row(s) affected  |
| 577 | 02:29:00 | SELECT * FROM Income LIMIT 0, 1000  | 5 row(s) returned  |
| 578 | 02:29:10 | INSERT INTO Income (in_name, in_date, in_money, in_it_id, user_id) VALUES ('Неверный...     | Error Code: 1644. Ошибка целостности ссылок: пользователь или категория не существует. |

## 2. AFTER INSERT триггер для логирования добавления записей в таблицу Income

Этот триггер будет срабатывать после каждой успешной вставки записи в таблицу Income. Он будет записывать информацию об этом событии в отдельную таблицу логов, которую назовем Income\_Log.

```
DELIMITER //
```

```
CREATE TRIGGER after_insert_income
```

```
AFTER INSERT ON Income
```

```
FOR EACH ROW
```

```
BEGIN
```

```
-- Добавляем запись в таблицу логов с полной информацией о вставленной записи
```

```
INSERT INTO Income_Log (in_id, in_name, in_date, in_money, in_it_id, user_id, action,  
action_time)
```

```
VALUES (NEW.in_id, NEW.in_name, NEW.in_date, NEW.in_money, NEW.in_it_id,  
NEW.user_id, 'INSERT', NOW());
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
DELIMITER ;
```

```
DROP TRIGGER after_insert_income;
```

```
INSERT INTO Income (in_name, in_date, in_money, in_it_id, user_id)
```

```
VALUES ('Тестовый доход', '2023-04-01', 30000.00, 1, 1);
```

```
SELECT * FROM Income_Log;
```

```
DELETE FROM Income WHERE in_name = 'Тестовый доход' AND in_date = '2023-04-01';
```

```
DELETE FROM Income_Log;
```

[illegible][illegible]

### 3. BEFORE DELETE триггер для обеспечения ссылочной целостности в таблице Income\_Item

Этот триггер будет срабатывать перед каждым удалением записи из таблицы Income\_Item. Его цель — удалять все связанные записи из таблицы Income, которые относятся к удаляемой статье дохода. Это обеспечивает целостность данных, так как удаляет все элементы, связанные с удаляемой категорией.

```
DELIMITER //
```

```
CREATE TRIGGER before_delete_income_item
```

```
BEFORE DELETE ON Income_Item
```

```
FOR EACH ROW
```

```
BEGIN
```

```
-- Удаляем все связанные записи из таблицы Income
```

```
DELETE FROM Income WHERE in_it_id = OLD.in_it_id;
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
DROP TRIGGER before_delete_income_item;
```

```
INSERT INTO Income_Category (in_cat_id, in_cat_name) VALUES (999, 'Тестовая категория');
```

```
INSERT INTO Income_Item (in_it_id, in_it_name, in_cat_id) VALUES (999, 'Тестовый элемент дохода', 999);
```

```
INSERT INTO Income (in_name, in_date, in_money, in_it_id, user_id) VALUES ('Тестовый доход', '2023-04-01', 10000.00, 999, 1);
```

```
DELETE FROM Income_Item WHERE in_it_id = 999;
```

```
DELETE FROM Income_Category WHERE in_cat_id = 999;
```



Result Grid Filter Rows: Search

|  | in_cat_id | in_cat_name        |  |
|--|-----------|--------------------|--|
|  | 1         | Зарплата           |  |
|  | 2         | Подарок            |  |
|  | 3         | Фриланс            |  |
|  | 999       | Тестовая категория |  |
|  | NULL      | NULL               |  |

Result Grid Filter Rows: Search

|  | in_it_id | in_it_name              | in_cat_id |  |
|--|----------|-------------------------|-----------|--|
|  | 1        | Работа Александра       | 1         |  |
|  | 2        | День рождения Михаила   | 2         |  |
|  | 3        | Фриланс Елены           | 3         |  |
|  | 4        | Фриланс Александра      | 3         |  |
|  | 999      | Тестовый элемент дохода | 999       |  |
|  | NULL     | NULL                    | NULL      |  |

|  | in_id | in_name                | in_date    | in_money | in_it_id | user_id |  |
|--|-------|------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра    | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу        | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены          | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра     | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | 16    | Тестовый доход         | 2023-04-01 | 10000.00 | 999      | 1       |  |
|  | NULL  | NULL                   | NULL       | NULL     | NULL     | NULL    |  |

|  | in_cat_id | in_cat_name        |  |
|--|-----------|--------------------|--|
|  | 1         | Зарплата           |  |
|  | 2         | Подарок            |  |
|  | 3         | Фриланс            |  |
|  | 999       | Тестовая категория |  |
|  | NULL      | NULL               |  |

|  | in_it_id | in_it_name            | in_cat_id |  |
|--|----------|-----------------------|-----------|--|
|  | 1        | Работа Александра     | 1         |  |
|  | 2        | День рождения Михаила | 2         |  |
|  | 3        | Фриланс Елены         | 3         |  |
|  | 4        | Фриланс Александра    | 3         |  |
|  | NULL     | NULL                  | NULL      |  |

Result Grid Filter Rows: Search Edit: [Icons]

|  | in_id | in_name                | in_date    | in_money | in_it_id | user_id |  |
|--|-------|------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра    | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу        | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены          | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра     | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | NULL  | NULL                   | NULL       | NULL     | NULL     | NULL    |  |

#### 4. AFTER DELETE триггер для логирования удаления записей из таблицы Income

Этот триггер будет срабатывать после каждого успешного удаления записи из таблицы Income. Он будет записывать информацию об удаленной записи в таблицу логов, которую мы можем назвать Income\_Delete\_Log

```
DELIMITER //
```

```
CREATE TRIGGER after_delete_income
```

```
AFTER DELETE ON Income
```

```
FOR EACH ROW
```

```
BEGIN
```

```
-- Добавляем запись в таблицу логов с информацией об удаленной записи
```

```
INSERT INTO Income_Delete_Log (in_id, in_name, in_date, in_money, in_it_id, user_id,  
action, action_time)
```

```
VALUES (OLD.in_id, OLD.in_name, OLD.in_date, OLD.in_money, OLD.in_it_id,  
OLD.user_id, 'DELETE', NOW());
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
INSERT INTO Income (in_name, in_date, in_money, in_it_id, user_id)
```

```
VALUES ('Тестовый доход', '2023-04-01', 20000.00, 1, 1);
```

```
DELETE FROM Income WHERE in_name = 'Тестовый доход' AND in_date = '2023-04-01';
```

```
SELECT * FROM Income_Delete_Log;
```

```
DELETE FROM Income_Delete_Log;
```



5. BEFORE UPDATE триггер для проверки корректности обновляемых данных в таблице Income

Этот триггер предназначен для проверки данных перед их обновлением в таблице Income. Он гарантирует, что определенные критерии соблюдаются при обновлении записей, например, что сумма дохода не отрицательная.

Предотвращение обновления записей в таблице Income с некорректными данными, такими как отрицательные суммы дохода.

```
DELIMITER //
```

```
CREATE TRIGGER before_update_income
```

```
BEFORE UPDATE ON Income
```

```
FOR EACH ROW
```

```
BEGIN
```

```
-- Проверяем, что сумма дохода не отрицательная
```

```
IF NEW.in_money < 0 THEN
```

```
    SIGNAL SQLSTATE '45000'
```

```
    SET MESSAGE_TEXT = 'Ошибка: сумма дохода не может быть отрицательной.';
```

```
END IF;
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
SELECT * FROM Income;
```

```
INSERT INTO Income (in_name, in_date, in_money, in_it_id, user_id)
```

```
VALUES ('Тестовый доход', '2023-04-01', 10000.00, 1, 1);
```

```
UPDATE Income SET in_money = 15000.00 WHERE in_name = 'Тестовый доход' AND  
in_date = '2023-04-01';
```

UPDATE Income SET in\_money = -15000.00 WHERE in\_name = 'Тестовый доход' AND in\_date = '2023-04-01';

DELETE FROM Income WHERE in\_name = 'Тестовый доход' AND in\_date = '2023-04-01';

|  | in_id | in_name                | in_date    | in_money | in_it_id | user_id |  |
|--|-------|------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра    | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу        | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены          | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра     | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | 18    | Тестовый доход         | 2023-04-01 | 10000.00 | 1        | 1       |  |
|  | NULL  | NULL                   | NULL       | NULL     | NULL     | NULL    |  |

|  | in_id | in_name                | in_date    | in_money | in_it_id | user_id |  |
|--|-------|------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра    | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу        | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены          | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра     | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | 18    | Тестовый доход         | 2023-04-01 | 15000.00 | 1        | 1       |  |
|  | NULL  | NULL                   | NULL       | NULL     | NULL     | NULL    |  |

|     | Time     | Action  | Response  | Duration / Fetch Time   |
|-----|----------|---|---|-------------------------|
| 611 | 02:40:10 | INSERT INTO Income (in_name, in_date, in_money, in_it_id, user_id) VALUES ('Тестовый доход', '2023-04-01', 100... | 1 row(s) affected   | 0.0036 sec              |
| 612 | 02:40:16 | SELECT * FROM Income LIMIT 0, 1000  | 6 row(s) returned   | 0.00065 sec / 0.000...  |
| 613 | 02:40:35 | UPDATE Income SET in_money = 15000.00 WHERE in_name = 'Тестовый доход' AND in_date = '2023-04-01'                 | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0            | 0.0045 sec              |
| 614 | 02:40:39 | SELECT * FROM Income LIMIT 0, 1000  | 6 row(s) returned   | 0.00076 sec / 0.0000... |
| 615 | 02:40:53 | UPDATE Income SET in_money = -15000.00 WHERE in_name = 'Тестовый доход' AND in_date = '2023-04-01'                | Error Code: 1644. Ошибка: сумма дохода не может быть отрицательной. | 0.0012 sec              |

## 6. AFTER UPDATE триггер для логирования обновлений записей в таблице Income

Этот триггер предназначен для логирования всех обновлений, производимых в записях таблицы Income. При каждом обновлении записи информация об этом обновлении будет записываться в таблицу логов Income\_Update\_Log.

Записывать детали каждого обновления, произведенного в таблице Income, включая старые и новые значения ключевых полей, а также время обновления.

```
CREATE TABLE Income_Update_Log (
```

```
    log_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    in_id INT,
```

```
    old_in_money DECIMAL(10, 2),
```

```
    new_in_money DECIMAL(10, 2),
```

```
    update_time DATETIME
```

```
);
```

```
DELIMITER //
```

```
CREATE TRIGGER after_update_income
```

```
AFTER UPDATE ON Income
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    -- Добавляем запись в таблицу логов с информацией об обновлении
```

```
    INSERT INTO Income_Update_Log (in_id, old_in_money, new_in_money, update_time)
```

```
    VALUES (NEW.in_id, OLD.in_money, NEW.in_money, NOW());
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
INSERT INTO Income (in_name, in_date, in_money, in_it_id, user_id)
```

```
VALUES ('Тестовый доход для обновления', '2023-04-01', 10000.00, 1, 1);
```

UPDATE Income SET in\_money = 20000.00 WHERE in\_name = 'Тестовый доход для обновления' AND in\_date = '2023-04-01';

SELECT \* FROM Income\_Update\_Log;

DELETE FROM Income WHERE in\_name = 'Тестовый доход для обновления' AND in\_date = '2023-04-01';

DELETE FROM Income\_Update\_Log;

|  | log_id | in_id | old_in_money | new_in_money | update_time |  |
|--|--------|-------|--------------|--------------|-------------|--|
|  | NULL   | NULL  | NULL         | NULL         | NULL        |  |

|  | in_id | in_name                       | in_date    | in_money | in_it_id | user_id |  |
|--|-------|-------------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра           | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу               | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены                 | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра            | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу        | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | 19    | Тестовый доход для обновления | 2023-04-01 | 10000.00 | 1        | 1       |  |
|  | NULL  | NULL                          | NULL       | NULL     | NULL     | NULL    |  |

|  | in_id | in_name                       | in_date    | in_money | in_it_id | user_id |  |
|--|-------|-------------------------------|------------|----------|----------|---------|--|
|  | 1     | Зарплата Александра           | 2023-01-15 | 60000.00 | 1        | 1       |  |
|  | 2     | Подарок Михаилу               | 2023-02-20 | 5000.00  | 2        | 3       |  |
|  | 3     | Фриланс Елены                 | 2023-03-05 | 15000.00 | 3        | 2       |  |
|  | 4     | Фриланс Александра            | 2023-02-20 | 10000.00 | 4        | 1       |  |
|  | 5     | Другой подарок Михаилу        | 2023-01-20 | 3000.00  | 2        | 3       |  |
|  | 19    | Тестовый доход для обновления | 2023-04-01 | 20000.00 | 1        | 1       |  |
|  | NULL  | NULL                          | NULL       | NULL     | NULL     | NULL    |  |

|  | log_id | in_id | old_in_money | new_in_money | update_time         |  |
|--|--------|-------|--------------|--------------|---------------------|--|
|  | 3      | 19    | 10000.00     | 20000.00     | 2023-12-21 02:42:34 |  |
|  | NULL   | NULL  | NULL         | NULL         | NULL                |  |