

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

Разработка SQL запросов: запросы с подзапросами
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:

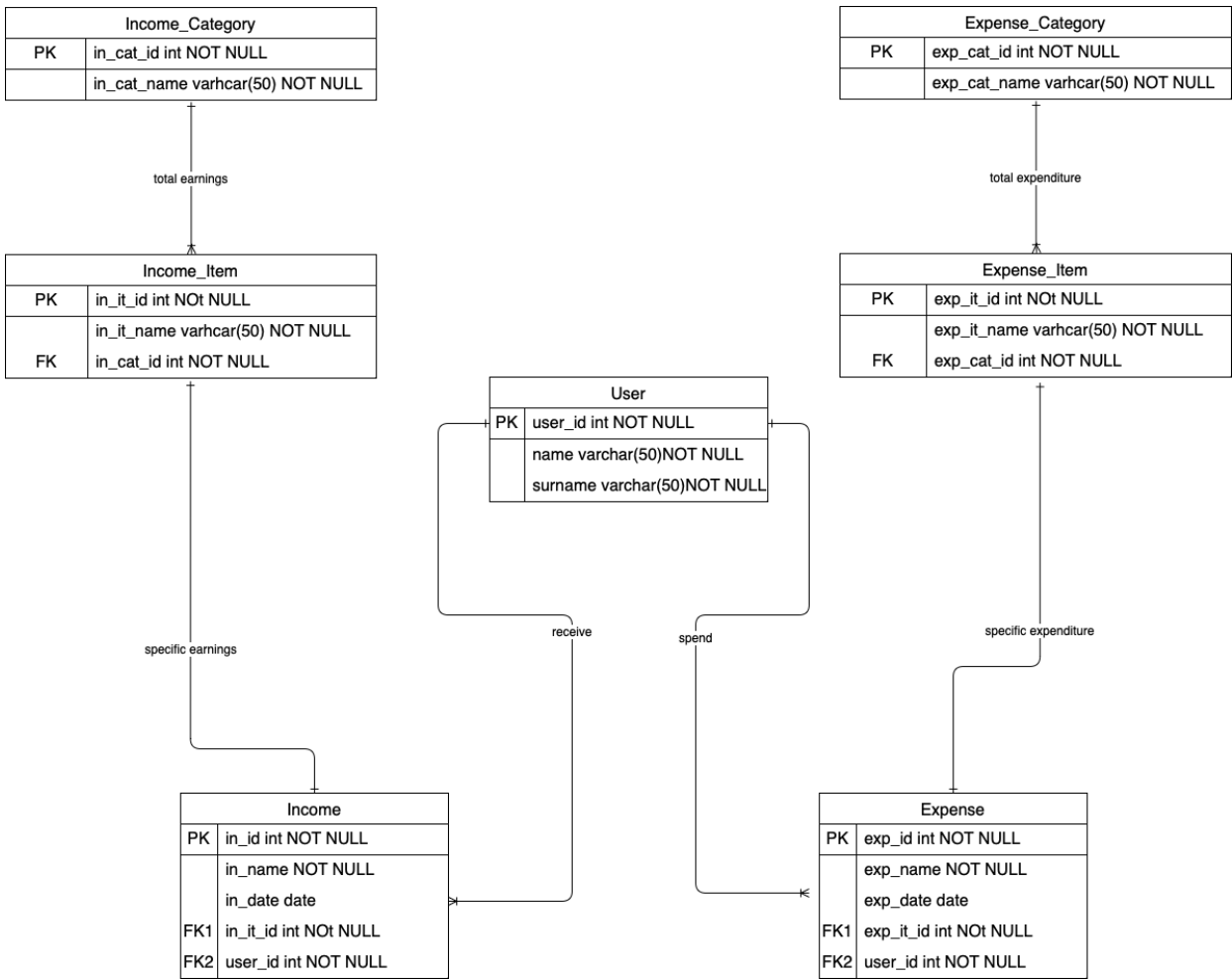


Таблица тестовых данных

Запрос	Данные, Удовлетворяющие Условиям	Данные, Не Удовлетворяющие Условиям
а. Расходы всех категорий, которые содержат часть слова «транспорт», но не заканчиваются на него	(5 таблица) Расходы на транспорт служебный	(5 таблица) Городской транспорт, транспорт
б. Месяц, в котором были статьи дохода от работы и от дарения	(6 таблица) 1	(6 таблица) 2, 3, 4
в. Пользователь без доходов	(4 таблица) 4 - Анастасия Иванова	(4 таблица) 1 - Александр Иванов, 2 - Елена Иванова, 3 - Михаил Иванов,
г. Категория, по которой были расходы в этом году максимальные по сумме	(5 таблица) Продукты	(5 таблица) Транспорт, Образование, Городской транспорт, Транспорт служебный
д. Категория, по которой не было расходов в январе, но были в мае	(5 таблица) Образование	(5 таблица) Продукты, Транспорт, Городской транспорт, Транспорт служебный

Запрос	Данные, Удовлетворяющие Условиям	Данные, Не Удовлетворяющие Условиям
е. Категория расхода, по которой траты были у всех членов семьи	(5 таблица) Продукты	(5 таблица) Транспорт, Образование, Городской транспорт, Транспорт служебный
ж. Месяц, в котором были расходы максимального количества статей	(6 таблица) 1	(6 таблица) 2, 3, 4, 5

Запросы:

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

запросы на максимум реализовать и с помощью директивы all(где возможно) и с помощью агрегатных функций

вывести должно: Продукты

агрегатная функция:

SELECT EC.exp_cat_name -- Выбираем название категории расходов

FROM Expense_Category EC -- Из таблицы категорий расходов

JOIN Expense_Item EI ON EC.exp_cat_id = EI.exp_cat_id -- Соединяем с таблицей элементов расходов по ID категории

JOIN Expense E ON EI.exp_it_id = E.exp_it_id -- Соединяем с таблицей расходов по ID элемента расхода

WHERE YEAR(E.exp_date) = YEAR(CURRENT_DATE) -- Фильтруем расходы текущего года

GROUP BY EC.exp_cat_name -- Группируем результаты по названию категории расходов

HAVING SUM(E.exp_money) = (-- Используем HAVING для фильтрации категории с максимальной суммой расходов

SELECT MAX(total_expense) -- Выбираем максимальную сумму расходов

FROM (-- Начинаем подзапрос для определения максимальной суммы расходов по категориям

```

SELECT EC2.exp_cat_id, SUM(E2.exp_money) AS total_expense -- Считаем сумму расходов для
каждой категории

FROM Expense_Category EC2 -- Из таблицы категорий расходов

JOIN Expense_Item EI2 ON EC2.exp_cat_id = EI2.exp_cat_id -- Соединяем с таблицей элементов
расходов по ID категории

JOIN Expense E2 ON EI2.exp_it_id = E2.exp_it_id -- Соединяем с таблицей расходов по ID
элемента расхода

WHERE YEAR(E2.exp_date) = YEAR(CURRENT_DATE) -- Фильтруем расходы текущего года

GROUP BY EC2.exp_cat_id -- Группируем результаты по ID категории расходов

) AS SubQuery -- Завершаем подзапрос и называем его SubQuery

);

```

All:

```

SELECT EC.exp_cat_name -- Выбираем название категории расходов

FROM Expense_Category EC -- Из таблицы категорий расходов

JOIN Expense_Item EI ON EC.exp_cat_id = EI.exp_cat_id -- Соединяем с таблицей элементов расходов
по ID категории

JOIN Expense E ON EI.exp_it_id = E.exp_it_id -- Соединяем с таблицей расходов по ID элемента
расхода

WHERE YEAR(E.exp_date) = YEAR(CURRENT_DATE) -- Фильтруем расходы текущего года

GROUP BY EC.exp_cat_id, EC.exp_cat_name -- Группируем результаты по ID и названию категории

HAVING SUM(E.exp_money) >= ALL ( -- Сравниваем сумму расходов каждой категории

    SELECT SUM(E2.exp_money) -- С суммой расходов по другим категориям

    FROM Expense E2

    JOIN Expense_Item EI2 ON E2.exp_it_id = EI2.exp_it_id

    WHERE YEAR(E2.exp_date) = YEAR(CURRENT_DATE) -- Фильтруем расходы текущего года

    GROUP BY EI2.exp_cat_id -- Группируем по ID категории расходов

);

```

Result Grid		Filter Rows:
exp_cat_name		
Продукты		

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

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

вывести должно: Образование

Использование NOT IN:

```
SELECT exp_cat_name
FROM Expense_Category
WHERE exp_cat_id NOT IN (
    -- Выбираем категории, которые были в январе
    SELECT exp_cat_id
    FROM Expense
    WHERE MONTH(exp_date) = 1
) AND exp_cat_id IN (
    -- Выбираем категории, которые были в мае
    SELECT exp_cat_id
    FROM Expense
    WHERE MONTH(exp_date) = 5
);
```

Синтаксис EXCEPT (для наглядности, так как MySQL не поддерживает):

```
SELECT EC.exp_cat_id, EC.exp_cat_name -- Выбираем ID и название категории расходов
FROM Expense_Category EC
WHERE EC.exp_cat_id IN (
    SELECT EI.exp_cat_id -- Выбираем ID категорий расходов за май
    FROM Expense E
    JOIN Expense_Item EI ON E.exp_it_id = EI.exp_it_id
    WHERE MONTH(E.exp_date) = 5

    EXCEPT -- Исключаем категории, по которым были расходы в январе

    SELECT EI.exp_cat_id -- Выбираем ID категорий расходов за январь
    FROM Expense E
    JOIN Expense_Item EI ON E.exp_it_id = EI.exp_it_id
```

```

WHERE MONTH(E.exp_date) = 1
);

SELECT EC.exp_cat_id , EC.exp_cat_name
FROM Expense E
JOIN Expense_Item EI ON E.exp_it_id = EI.exp_it_id
JOIN Expense_Category EC on EC.exp_cat_id=EI.exp_cat_id
WHERE MONTH(E.exp_date) = 5

```

EXCEPT

```

SELECT EC.exp_cat_id , EC.exp_cat_name
FROM Expense E
JOIN Expense_Item EI ON E.exp_it_id = EI.exp_it_id
JOIN Expense_Category EC on EC.exp_cat_id=EI.exp_cat_id
WHERE MONTH(E.exp_date) = 1



```

Использование левого соединения:

```

SELECT EC.exp_cat_id , EC.exp_cat_name
FROM Expense E
JOIN Expense_Item EI ON E.exp_it_id = EI.exp_it_id
JOIN Expense_Category EC on EC.exp_cat_id=EI.exp_cat_id
LEFT JOIN (SELECT EC.exp_cat_id , EC.exp_cat_name
FROM Expense E
JOIN Expense_Item EI ON E.exp_it_id = EI.exp_it_id
JOIN Expense_Category EC on EC.exp_cat_id=EI.exp_cat_id
WHERE MONTH(E.exp_date) = 1) AS nnn on EC.exp_cat_id = nnn.exp_cat_id
WHERE MONTH(E.exp_date) = 5
and nnn.exp_cat_id is null;

```


Result Grid   Filter Rows:	
exp_cat_name	
Образование	

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

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

вывести должно: Продукты

Использование двух NOT EXISTS

-- Expense_Category / User

SELECT EC.exp_cat_name -- Выбираются (exp_cat_name) из Expense_Category

FROM Expense_Category EC

WHERE NOT EXISTS (-- фильтрует категории расходов

SELECT * -- проверяет каждого пользователя U в таблице User

FROM User U

WHERE NOT EXISTS (-- Для каждого пользователя проверяется условие:

SELECT * -- проверяет, совершал ли данный пользователь (U.user_id = E.user_id)

FROM Expense E -- расходы в определенной категории (EI.exp_cat_id = EC.exp_cat_id)

JOIN Expense_Item EI ON E.exp_it_id = EI.exp_it_id

WHERE U.user_id = E.user_id AND EI.exp_cat_id = EC.exp_cat_id

) -- не существует записей о расходах в этой категории, то NOT EXISTS возвращает TRUE

);

Использование агрегатной функции

SELECT EC.exp_cat_name

FROM Expense_Category EC -- соединяются три таблицы

JOIN Expense_Item EI ON EC.exp_cat_id = EI.exp_cat_id

JOIN Expense E ON EI.exp_it_id = E.exp_it_id



GROUP BY EC.exp_cat_name -- Результаты группируются по названию категории

HAVING COUNT(DISTINCT E.user_id) = (SELECT COUNT(*) FROM User);

-- COUNT(DISTINCT E.user_id) подсчитывает количество уникальных пользователей,

-- совершивших расходы в каждой категории. Это число сравнивается с общим количеством

-- пользователей в системе (SELECT COUNT(*) FROM User).

Result Grid   Filter Rows	
exp_cat_name	
Продукты	

ж. Месяц, в котором были расходы максимального количества статей

запросы на максимум реализовать и с помощью директивы all(где возможно) и с помощью агрегатных функций

вывести должно: 1

Агрегатная функция:

```
SELECT MONTH(exp_date) AS month -- выбирается месяц из даты каждого расхода
FROM Expense
GROUP BY MONTH(exp_date) -- группируются по месяцу
HAVING COUNT(DISTINCT exp_it_id) = ( -- для фильтрации месяцев
    SELECT MAX(article_count) -- определяет количество статей расходов в каждом месяце,
    FROM ( -- а затем выбирает максимальное значение из этих чисел
        SELECT COUNT(DISTINCT exp_it_id) AS article_count
        FROM Expense -- подсчитывает количество уникальных статей расходов в каждом месяце
        GROUP BY MONTH(exp_date) -- Результаты группируются по месяцам
    ) AS SubQuery
);
```

All:

```
SELECT MONTH(exp_date) AS month -- выбирается месяц из даты каждого расхода
```

```
FROM Expense
```

```
GROUP BY MONTH(exp_date) -- группируются по месяцу
```


```
HAVING COUNT(DISTINCT exp_it_id) >= ALL ( -- условие фильтрует месяцы, оставл >= AL
```

```
    SELECT COUNT(DISTINCT exp_it_id) -- колво уникальных статей для каждого месяца.
```

```
FROM Expense
```

```
GROUP BY MONTH(exp_date)
```

```
);
```

Result Grid  Filter F	
Month	
1	