

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 10

Объектно-реляционные базы данных. Манипуляция данными и
пользовательские операторы

по курсу: Проектирование баз данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2024

Текст задания:

- 1) Выполнить вставку тестовых данных в таблицы, созданные в ходе выполнения лабораторной работы 9.
- 2) Придумать (задание) и реализовать запрос выборки с условием к таблицам и предку и потомку. Придумать и реализовать запросы выборки с условием к таблице только к предку и только к потомку.
- 3) Придумать и создать пользовательский оператор для своей предметной области
- 4) Придумать и создать пользовательскую агрегатную функцию для своей предметной области

Предметная область:

13. калькулятор бюджета физического лица: пользователь калькулятора (разные уровни доступа), категория дохода (продажа, зарплата), категория расхода (еда, счета за КУ, здоровье...), статьи дохода и расхода, дата расхода/дохода
- а. статьи категорий, которые относятся к спорту (содержат слово спорт)
 - б. месяц, в котором были разные статьи дохода
 - в. категория, по которой наибольшие расходы в текущем году
 - г. категория, по которой не было расходов в феврале
 - д. пользователь, добавивший наименьшее количество статей

Физическая модель БД:

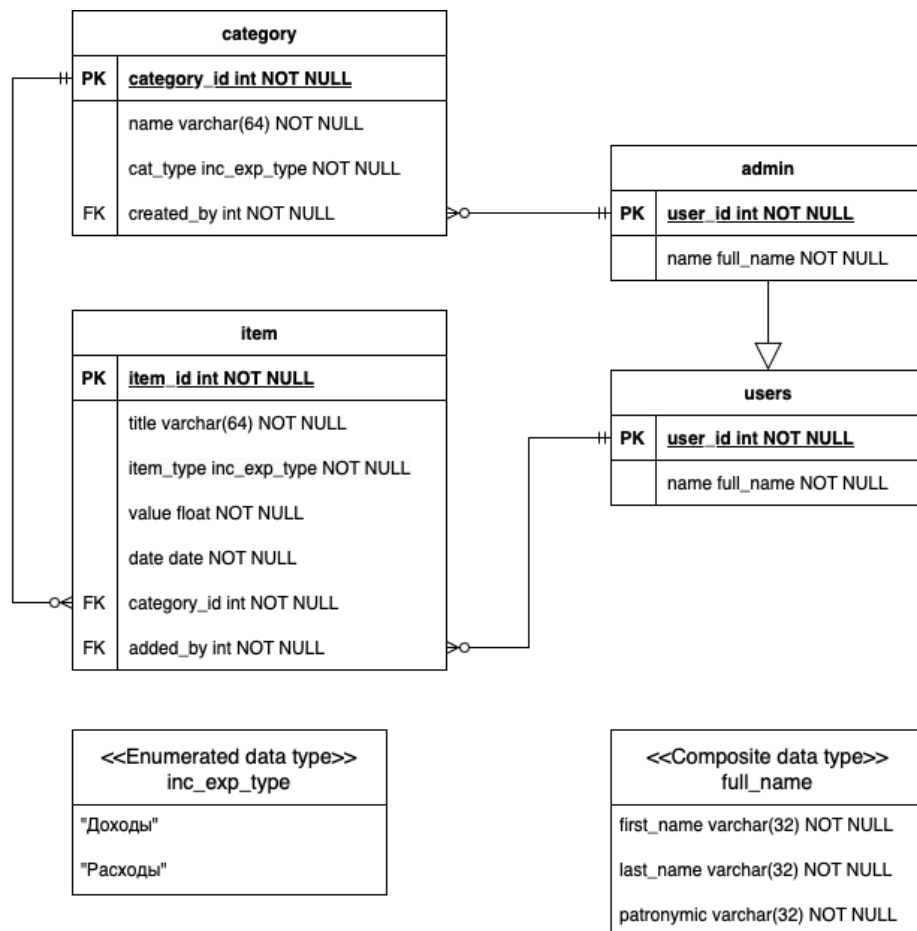


Таблица тестовых данных:

Текст запроса	Данные, удовлетворяющие условиям	Данные, не удовлетворяющие условиям
а. Статьи категорий, которые относятся к спорту (содержат слово "спорт")	- Абонемент в спортзал (категория: Спорт и фитнес) - Спортивная одежда (категория: Спорт и фитнес) - Покупка велосипеда (категория: Спорт и фитнес)	- Продажа автомобиля - Зарплата за январь - Покупка продуктов - Оплата коммунальных услуг - Медицинская страховка - Зарплата за февраль
б. Месяц, в котором были разные статьи дохода	- Январь 2024 (Продажа автомобиля, Зарплата за январь) - Февраль 2024 (Зарплата за февраль)	- Март 2024 (нет статей дохода) - Апрель 2024 (нет статей дохода) - Май 2024 (нет статей дохода)
в. Категория, по которой наибольшие расходы в текущем году	- Спорт и фитнес (3 статьи на общую сумму 25000)	- Еда (1 статья на сумму 5000) - Счета за КУ (1 статья на сумму 8000) - Здоровье (1 статья на сумму 12000)
г. Категория, по которой не было расходов в феврале	- Здоровье - Спорт и фитнес	- Еда (расход в феврале) - Счета за КУ (расход в феврале)
д. Пользователь, добавивший наименьшее количество статей	- Ян Кузнецов (user_id = 13) (0 статей) - Алексей Иванов (user_id = 11) (0 статей) - Мария Смирнова (user_id = 12) (0 статей)	- Иван Сидоров (user_id = 14) (4 статьи) - Елена Козлова (user_id = 15) (5 статей)

Вставка данных:

-- Вставка данных в таблицу users

-- Пользователи калькулятора бюджета

INSERT INTO users (name) VALUES

((('Алексей', 'Иванов', 'Петрович'))), -- user_id = 1

((('Мария', 'Смирнова', 'Игоревна'))), -- user_id = 2

((('Ян', 'Кузнецов', 'Сергеевич'))); -- user_id = 3

-- Вставка данных в таблицу admin (администраторы)

-- Администраторы калькулятора бюджета

INSERT INTO admin (name, description) VALUES

((('Иван', 'Сидоров', 'Алексеевич'), 'Главный администратор')), -- user_id = 4

((('Елена', 'Козлова', 'Викторовна'), 'Администратор категории доходов')); -- user_id = 5

-- Вставка данных в таблицу category (категории доходов и расходов)

-- Категории доходов и расходов, добавленные администраторами

INSERT INTO category (name, cat_type, created_by) VALUES

('Продажа', 'Доходы', 4), -- Иван Сидоров добавил категорию Продажа
('Зарплата', 'Доходы', 5), -- Елена Козлова добавила категорию Зарплата
('Еда', 'Расходы', 4), -- Иван Сидоров добавил категорию Еда
('Счета за КУ', 'Расходы', 5), -- Елена Козлова добавила категорию Счета за КУ
('Здоровье', 'Расходы', 4), -- Иван Сидоров добавил категорию Здоровье
('Спорт и фитнес', 'Расходы', 4); -- Иван Сидоров добавил категорию Спорт и фитнес

-- Вставка данных в таблицу item (статьи доходов и расходов)

-- Статьи доходов и расходов, добавленные пользователями





INSERT INTO item (title, item_type, value, date, category_id, added_by) VALUES

('Продажа автомобиля', 'Доходы', 500000, '2024-01-15', 1, 1),
('Зарплата за январь', 'Доходы', 60000, '2024-01-31', 2, 2),
('Покупка продуктов', 'Расходы', 5000, '2024-02-10', 3, 3),
('Оплата коммунальных услуг', 'Расходы', 8000, '2024-02-20', 4, 1),
('Медицинская страховка', 'Расходы', 12000, '2024-03-01', 5, 2),
('Абонемент в спортзал', 'Расходы', 3000, '2024-04-05', 6, 3),
('Спортивная одежда', 'Расходы', 7000, '2024-04-15', 6, 1),
('Зарплата за февраль', 'Доходы', 60000, '2024-02-28', 2, 2),
('Покупка велосипеда', 'Расходы', 15000, '2024-05-20', 6, 1);

Придумать (задание) и реализовать запрос выборки с условием к таблицам и предку и потомку. Придумать и реализовать запросы выборки с условием к таблице только к предку и только к потомку.:




Запрос к предку и потомку (д)

```
WITH user_article_counts AS (  
    SELECT u.user_id, (u.name).first_name, (u.name).last_name, COUNT(i.item_id) AS  
    item_count  
    FROM users u  
    LEFT JOIN item i ON u.user_id = i.added_by  
    GROUP BY u.user_id  
)  
SELECT user_id, first_name, last_name, item_count  
FROM user_article_counts  
WHERE item_count = (SELECT MIN(item_count) FROM user_article_counts);
```

	user_id [PK] integer 	first_name character varying (32) 	last_name character varying (32) 	item_count bigint 
1	5	Елена	Козлова	0
2	4	Иван	Сидоров	0





Запрос к предку (Пользователи, чьи фамилии начинаются с буквы 'И')

```
SELECT user_id, (name).first_name, (name).last_name  
FROM ONLY users  
WHERE (name).last_name LIKE 'И%';
```

	user_id [PK] integer 	first_name character varying (32) 	last_name character varying (32) 
1	1	Алексей	Иванов

Запрос к потомку (Администраторы, добавившие категории доходов)

```
SELECT a.user_id, (u.name).first_name, (u.name).last_name, a.description  
FROM admin a  
JOIN users u ON a.user_id = u.user_id  
JOIN category c ON a.user_id = c.created_by  
WHERE c.cat_type = 'Доходы';
```

	user_id [PK] integer 	first_name character varying (32) 	last_name character varying (32) 	description text 
1	4	Иван	Сидоров	Главный администратор
2	5	Елена	Козлова	Администратор категории доходов

Придумать и создать пользовательский оператор для своей предметной области.

Оператор будет проверять, делится ли сумма value на 7, либо введенное значение без остатка

```
CREATE OR REPLACE FUNCTION is_divisible_by(value FLOAT, multiple FLOAT)
```

```
RETURNS BOOLEAN AS $$
```

```
BEGIN
```

```
    RETURN MOD(value::NUMERIC, multiple::NUMERIC) = 0;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OPERATOR %% (
```

```
    LEFTARG = FLOAT,
```

```
    RIGHTARG = FLOAT,
```

```
    PROCEDURE = is_divisible_by
```



```
);
```

```
-- Пример использования
```

```
SELECT title, value
```

```
FROM item
```

```
WHERE value %% 7;
```

	title character varying (64) 	value double precision 
1	Спортивная одежда	7000

```
SELECT title, value, value % 7
FROM item
```

	title character varying (64) 	value double precision 	?column? boolean 
1	Продажа автомобиля	500000	false
2	Зарплата за январь	60000	false
3	Покупка продуктов	5000	false
4	Оплата коммунальных услуг	8000	false
5	Медицинская страховка	12000	false
6	Абонемент в спортзал	3000	false
7	Спортивная одежда	7000	true
8	Зарплата за февраль	60000	false
9	Покупка велосипеда	15000	false

Придумать и создать пользовательскую агрегатную функцию для своей предметной области.

Поиск самого короткого имени

```
CREATE OR REPLACE FUNCTION find_shortest_full_name(a full_name, b full_name)
RETURNS full_name AS
$$
DECLARE
    a_length INT;
    b_length INT;
BEGIN
    -- проверки на полное NULL значение
    IF a IS NULL THEN
        RETURN b;
    END IF;
    IF b IS NULL THEN
```

```

        RETURN a;
    END IF;

    -- Вычисление длины полного имени для каждого аргумента с учётом NULL значений
    a_length := LENGTH(COALESCE(a.first_name, '')) + LENGTH(COALESCE(a.last_name, '')) + LENGTH(COALESCE(a.patronymic, ''));
    b_length := LENGTH(COALESCE(b.first_name, '')) + LENGTH(COALESCE(b.last_name, '')) + LENGTH(COALESCE(b.patronymic, ''));


    -- Сравнение длин полных имен
    IF a_length < b_length THEN
        RETURN a;
    ELSE
        RETURN b;
    END IF;
END;
$$ LANGUAGE plpgsql;

-- Пересоздаем агрегатную функцию
DROP AGGREGATE IF EXISTS shortest_full_name(full_name);

CREATE AGGREGATE shortest_full_name(full_name) (
    SFUNC = find_shortest_full_name,
    STYPE = full_name
);

-- Тестовый запрос для проверки функции:
SELECT shortest_full_name(name) FROM users;

```

	shortest_full_name full_name 
1	(Ян,Кузнецов,Сергеевич)

Проверка:

1. Алексей Иванов Петрович

- Имя: Алексей (7 символов)
- Фамилия: Иванов (6 символов)
- Отчество: Петрович (8 символов)
- Общая длина: $7+6+8=21$ 7+6+8=21 символ

2. Мария Смирнова Игоревна

- Имя: Мария (5 символов)
- Фамилия: Смирнова (8 символов)
- Отчество: Игоревна (7 символов)
- Общая длина: $5+8+7=20$ 5+8+7=20 символов

3. Ян Кузнецов Сергеевич

- Имя: Ян (2 символа)
- Фамилия: Кузнецов (8 символов)
- Отчество: Сергеевич (9 символов)
- Общая длина: $2+8+9=19$ 2+8+9=19 символов

4. Иван Сидоров Алексеевич

- Имя: Иван (4 символа)
- Фамилия: Сидоров (7 символов)
- Отчество: Алексеевич (10 символов)
- Общая длина: $4+7+10=21$ 4+7+10=21 символ

5. Елена Козлова Викторовна

- Имя: Елена (5 символов)
- Фамилия: Козлова (7 символов)
- Отчество: Викторовна (10 символов)
- Общая длина: $5+7+10=22$ 5+7+10=22 символа

Исходя из расчета, самое короткое полное имя среди этих — это "Ян Кузнецов Сергеевич" с 19 символами.