**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**

**(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кафедра |  | О7 |  | Программная инженерия |
|  |  | шифр |  | наименование кафедры, по которой выполняется работа |
| Дисциплина |  | Базы данных | | |
|  |  | наименование дисциплины | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ПРАКТИЧЕСКАЯ РАБОТА | | | | |
|  | 4 |  |
|  | номер задания (при наличии) |  |
| ПОЛНОЦЕННОЕ НАПОЛНЕНИЕ БД И СОЗДАНИЕ ВСЕХ | | | |
| ЗАПРОСОВ И ПРОЦЕДУР ПРЕДМЕТНОЙ ОБЛАСТИ КРИПТОБИРЖИ | | | |

при наличии указать тему учебно-практической работы и (или) номер варианта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ОБУЧАЮЩИЙСЯ** | | | | | |
| группы | | | |  | О726Б |
|  |  | Махов Н.М. | | | |
| подпись |  | фамилия и инициалы | | | |
|  | | | | |
| дата сдачи | | | | |
| **ПРОВЕРИЛ** | | | | | |
| Преподаватель | | | | | | |
| ученая степень, ученое звание, должность | | | | | | |
|  |  | Шибаев А.А. | | | | |
| подпись |  | фамилия и инициалы | | | | |
| Оценка / балльная оценка | | |  | | |
|  | | | | |
| дата проверки | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Санкт-Петербург | | | | | | |
|  |  |  | 20 | 25 | г. |  |

**Введение**

Настоящая работа направлена на разработку прикладной логики к уже сформированной структуре базы данных криптовалютной биржи, реализованной средствами PostgreSQL. Исходная модель включает основные сущности платформы — пользователей, криптовалюты, кошельки, ордера, транзакции и стейкинг — и была предварительно заполнена тестовыми данными.

Целью работы является создание пользовательских функций, хранимых процедур и представлений. Разработка проводилась в среде pgAdmin с использованием языка SQL.

**Результаты работы**

1. Хранимые функции

Функция 1: Получение общего оборота по заданной валюте за указанный месяц

CREATE OR REPLACE FUNCTION get\_currency\_turnover(currency\_code VARCHAR, year INT, month INT)

RETURNS NUMERIC AS $$

DECLARE

result NUMERIC := 0;

BEGIN

SELECT COALESCE(SUM(o.order\_count\_currency), 0)

INTO result

FROM transactions t

JOIN orders o ON t.id\_order = o.id\_order

JOIN cryptocurrency c ON t.id\_currency = c.id\_currency

WHERE c.currency\_name = currency\_code

AND EXTRACT(YEAR FROM t.trans\_data) = year

AND EXTRACT(MONTH FROM t.trans\_data) = month;

RETURN result;

END;

$$ LANGUAGE plpgsql;

Вызов: SELECT get\_currency\_turnover('BTC', 2025, 4); -- валюта, год, месяц

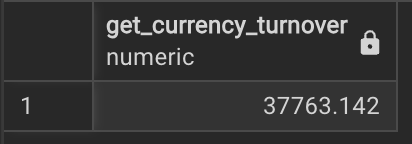


Рисунок 1 - Результат выполнения функции 1

Функция 2: Проверка, участвует ли пользователь в стейкинге

CREATE OR REPLACE FUNCTION user\_has\_active\_staking(user\_id BIGINT)

RETURNS BOOLEAN AS $$

DECLARE

res BOOLEAN := FALSE;

BEGIN

SELECT TRUE

INTO res

FROM staking s

JOIN spot\_wallet sw ON s.id\_wallet = sw.id\_wallet

WHERE sw.id\_user = user\_id AND s.staking\_end\_date > CURRENT\_DATE

LIMIT 1;

RETURN COALESCE(res, FALSE);

END;

$$ LANGUAGE plpgsql;

Вызов: SELECT user\_has\_active\_staking(2); -- user\_id

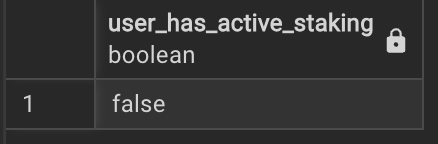


Рисунок 2 - Результат выполнения функции 1

2. Процедура: Пополнение баланса кошелька по id\_wallet

CREATE OR REPLACE PROCEDURE refill\_wallet(p\_wallet\_id BIGINT, p\_amount DECIMAL)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE spot\_wallet

SET balance = ROUND((balance + p\_amount)::NUMERIC, 2)

WHERE id\_wallet = p\_wallet\_id;

END;

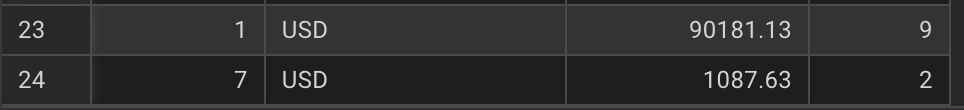
$$;

Рисунок 3 - Вывод до процедур

Применим CALL refill\_wallet(7, -100.00); баланс на кошельке должен уменьшиться.

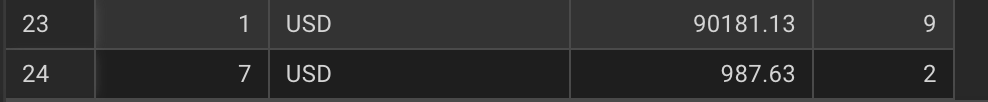


Рисунок 4 - Результат после выполнения процедуры после

3. Представление 1: 5 лидеров по обороту валют за текущий месяц

CREATE OR REPLACE VIEW top5\_currency\_turnover\_month AS

SELECT

c.currency\_name,

ROUND(SUM(o.order\_count\_currency)::NUMERIC, 2) AS total\_turnover,

ROUND(SUM(o.order\_count\_currency \* c.currency\_price\_usd)::NUMERIC, 2) AS turnover\_usd

FROM

transactions t

JOIN orders o ON t.id\_order = o.id\_order

JOIN cryptocurrency c ON t.id\_currency = c.id\_currency

WHERE

DATE\_TRUNC('month', t.trans\_data) = DATE\_TRUNC('month', CURRENT\_DATE)

GROUP BY

c.currency\_name

ORDER BY

turnover\_usd DESC

LIMIT 5;

Вызов: SELECT \* FROM top5\_currency\_turnover\_month;

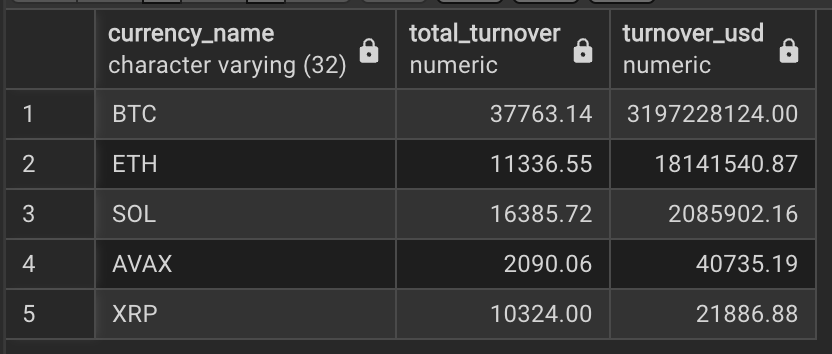


Рисунок 5 - Результат выполнения представления 1

Представление 2: Сводка по стейкингу каждого кошелька

CREATE OR REPLACE VIEW wallet\_staking\_summary AS

SELECT

sw.id\_wallet,

sw.id\_user,

COUNT(s.id\_staking) AS total\_stakings,

COUNT(CASE WHEN s.staking\_end\_date > CURRENT\_DATE THEN 1 END) AS active\_stakings,

ROUND(SUM(CASE

WHEN CURRENT\_DATE > s.staking\_end\_date THEN

(s.staking\_sum::numeric \* s.staking\_percentage::numeric / 100)

ELSE 0

END), 2) AS earned\_from\_staking,

ROUND(SUM(s.staking\_sum::numeric \* s.staking\_percentage::numeric / 100), 2) AS total\_expected\_profit

FROM

spot\_wallet sw

LEFT JOIN staking s ON sw.id\_wallet = s.id\_wallet

GROUP BY

sw.id\_wallet;

Вызов: SELECT \* FROM wallet\_staking\_summary;

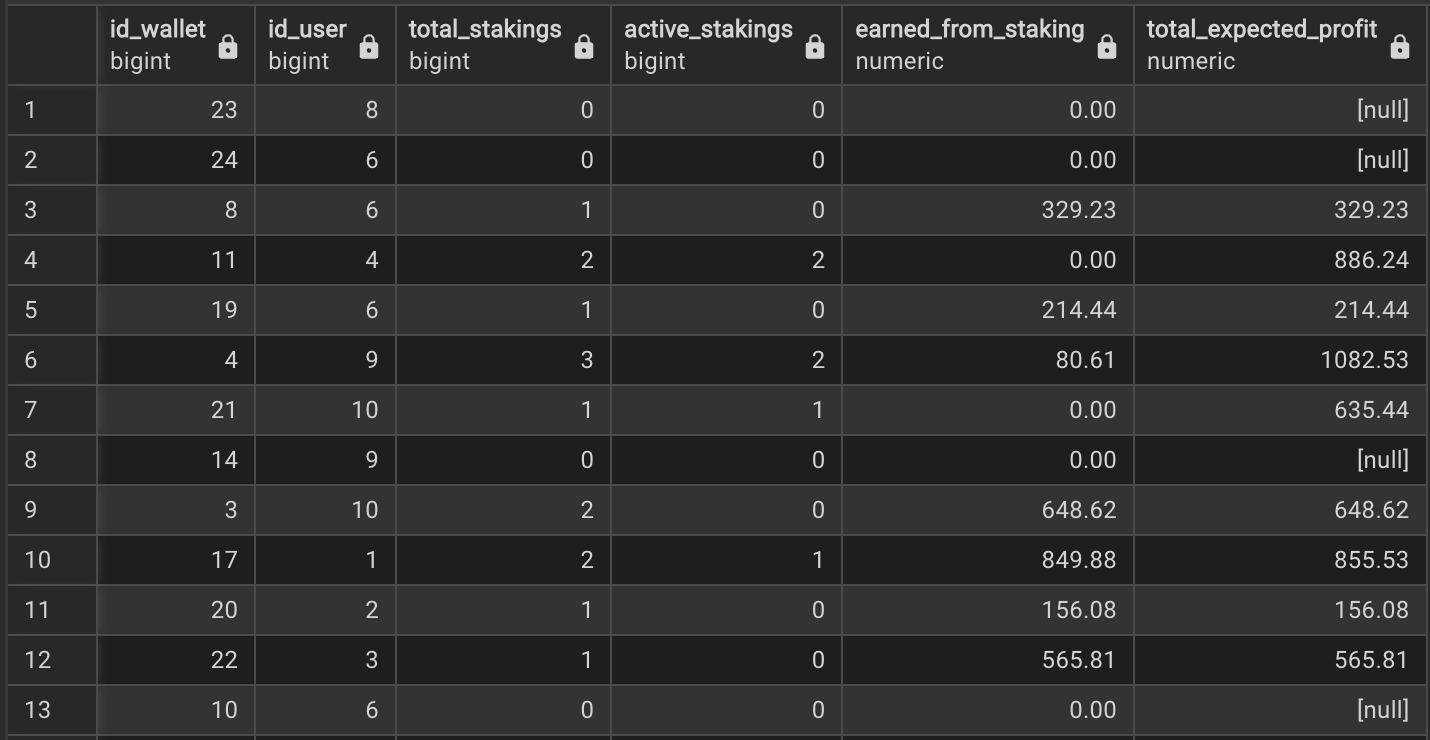


Рисунок 6 - Результат выполнения представления 2

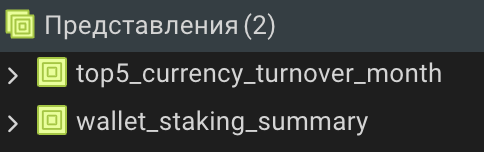


Рисунок 7 - Процедуры в среде pgAdmin

4. Автоматизация

Автоматически фиксирует изменение баланса кошелька и добавляет запись в account\_oper, в основе триггеры.

CREATE OR REPLACE FUNCTION log\_wallet\_refill()

RETURNS TRIGGER AS $$

BEGIN

INSERT INTO account\_oper (id\_wallet, account\_oper\_sum, account\_oper\_currency, account\_oper\_status, account\_oper\_type)

VALUES (NEW.id\_wallet, NEW.balance - OLD.balance, NEW.currency, TRUE, TRUE);

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER after\_wallet\_update

AFTER UPDATE OF balance ON spot\_wallet

FOR EACH ROW

WHEN (NEW.balance > OLD.balance)

EXECUTE FUNCTION log\_wallet\_refill();

**Заключение**

Функции, представления и процедуры упрощают работу с базами данных и расширяют их границы использования.