**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 | | | | |
|  | 3 |  |
|  | номер задания (при наличии) |  |
| НАЧАЛЬНОЕ НАПОЛНЕНИЕ БД И СОЗДАНИЕ ЗАПРОСОВ | | | |
| 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**

На базе инструментов PostgreSQL и среды pgAdmin была разработана и реализована база данных, моделирующая ключевые процессы криптовалютной биржи. Структура базы охватывает основные сущности, включая пользователей, спотовые кошельки, криптовалюты, ордера, транзакции, операции по счёту и стейкинг.

С помощью SQL-скриптов были созданы необходимые таблицы, настроены внешние ключи с различными режимами каскадных операций, а также произведено начальное наполнение базы данных тестовыми данными. Это позволило воссоздать реалистичную модель поведения пользователей на платформе.

В рамках работы были реализованы три аналитических SQL-запроса, отражающих реальные кейсы использования базы данных:

1. Подсчёт количества исполненных ордеров по каждой криптовалюте для каждого пользователя в рамках текущего месяца.

2. Анализ состояния кошельков: количество активных стейкингов, ожидаемая и уже полученная доходность.

3. Получение топ-5 криптовалют по объёму транзакций за месяц, с детализацией по пользователям и группировкой результатов.

**Код базы данных**

Приложение 1.

**Заполнение данными**

Приложение 2 и приложение 3.

**Результаты работы**

Запрос 1 выводит все оредера по пользователям в рамках текущего месяца, из 30 транзакций, которые были выполнены на n период, вывелись 20 тех которые были выполнены за апрель 2025 года. Вывод происходит по каждой криптовалюте и типу, если пользователь совершил несколько транзакций по одной валюте, то они суммируются, как например qgrimes, который за месяц сделал несколько транзакций с валютой BTC. Результат представлен в таблице 1 и на рисунке 1.

Запрос 1

SELECT

u.id AS user\_id,

u.name,

c.currency\_name,

CASE

WHEN o.order\_type = TRUE THEN 'BUY'

ELSE 'SELL'

END AS order\_type,

COUNT(\*) AS orders\_count,

SUM(o.order\_count\_currency) AS total\_amount

FROM

orders o

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

JOIN users u ON o.id\_users = u.id

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

WHERE

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

AND o.order\_status = TRUE

GROUP BY

u.id, c.currency\_name, o.order\_type

ORDER BY

u.id, c.currency\_name;

Таблица 1 - Результат выполнения запроса 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **user\_id** | **name** | **currency\_name** | **order\_type** | **first\_order\_date** | **orders\_count** | **total\_amount** |
| **1** | achang | BTC | SELL | 2025-04-16 | 1 | 1.0366 |
| **1** | achang | TON | SELL | 2025-04-16 | 1 | 3.4096 |
| **2** | juancampos | AVAX | SELL | 2025-04-02 | 1 | 0.6758 |
| **2** | juancampos | DOGE | BUY | 2025-04-08 | 1 | 17661.32 |
| **3** | thomas15 | BTC | SELL | 2025-04-03 | 1 | 17103.19 |
| **3** | thomas15 | DOGE | BUY | 2025-04-07 | 1 | 10101.88 |
| **4** | qgrimes | BTC | BUY | 2025-04-05 | 2 | 6874.9854 |
| **4** | qgrimes | ETH | SELL | 2025-04-12 | 1 | 11336.55 |
| **4** | qgrimes | TON | SELL | 2025-04-07 | 1 | 1000.7676 |
| **5** | john51 | SOL | BUY | 2025-04-01 | 1 | 14239.51 |
| **6** | jrodriguez | SOL | BUY | 2025-04-06 | 1 | 2146.21 |
| **6** | jrodriguez | TRX | SELL | 2025-04-04 | 1 | 19234.66 |
| **6** | jrodriguez | XRP | SELL | 2025-04-15 | 1 | 10324 |
| **7** | tamaramorrison | DOGE | SELL | 2025-04-13 | 1 | 8474.73 |
| **8** | barbara42 | DOGE | BUY | 2025-04-10 | 1 | 5333.58 |
| **9** | sandrakirby | APT | SELL | 2025-04-11 | 1 | 2044.92 |
| **9** | sandrakirby | ARB | BUY | 2025-04-16 | 1 | 400.733 |
| **10** | criley | AVAX | BUY | 2025-04-02 | 1 | 2089.38 |
| **10** | criley | BTC | BUY | 2025-04-09 | 1 | 13783.93 |
| **10** | criley | TON | SELL | 2025-04-02 | 1 | 3000.5647 |

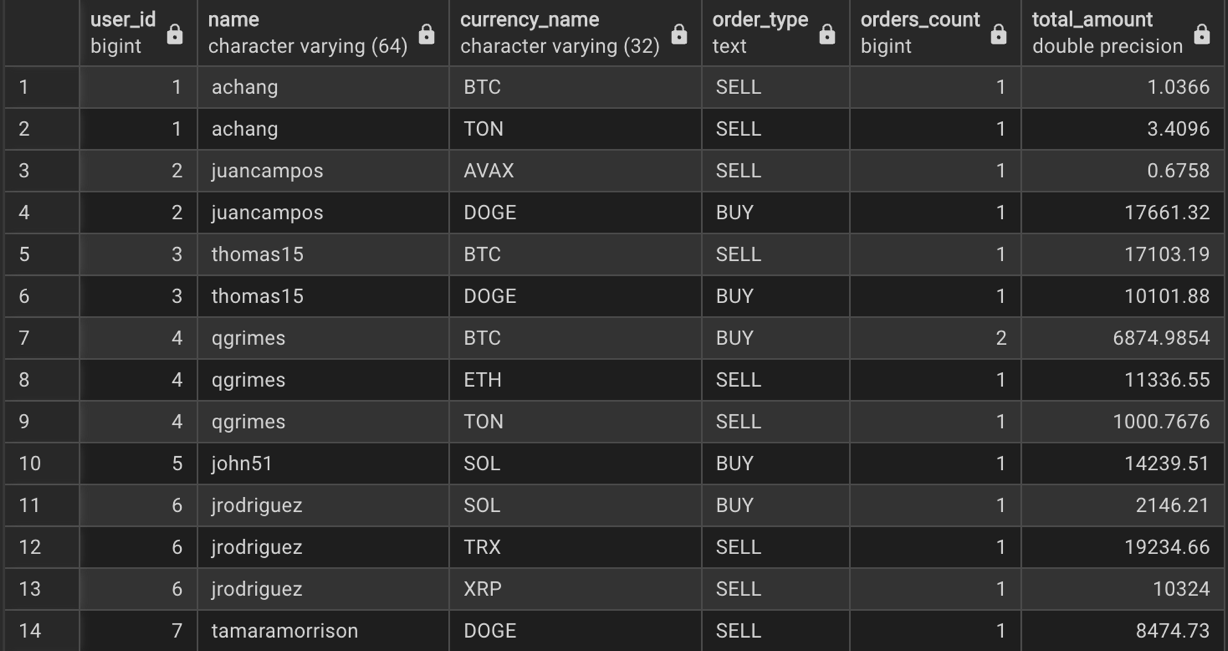


Рисунок 1 - Результат выполнения 1 запроса, часть таблицы

Запрос 2 выводит количество активных стейкингов на кошельках пользователя, а также ожидаемую и уже полученную доходность. Если у пользователя нет и не было стейкингов то итоговый доход NULL, если у пользователя ранее был стейкинг и он уже закрыт то его доход в earned\_from\_staking и total\_expected\_profit одинаковый, если стейкинг еще открыт значение помещается и total\_expected\_profit и будет больше чем earned\_from\_staking.

Запрос 2

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

ORDER BY

sw.id\_wallet;

Таблица 2 - Результат выполнения запроса 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id\_wallet** | **id\_user** | **total\_stakings** | **active\_stakings** | **earned\_from\_staking** | **total\_expected\_profit** |
| **1** | 9 | 3 | 1 | 431.35 | 821.82 |
| **2** | 3 | 2 | 2 | 0.00 | 377.72 |
| **3** | 10 | 2 | 0 | 648.62 | 648.62 |
| **4** | 9 | 3 | 2 | 80.61 | 1082.53 |
| **5** | 10 | 1 | 1 | 0.00 | 143.00 |
| **6** | 5 | 0 | 0 | 0.00 | NULL |
| **7** | 2 | 2 | 0 | 602.98 | 602.98 |
| **8** | 6 | 1 | 0 | 329.23 | 329.23 |
| **9** | 2 | 1 | 0 | 394.97 | 394.97 |
| **10** | 6 | 0 | 0 | 0.00 | NULL |
| **11** | 4 | 2 | 2 | 0.00 | 886.24 |
| **12** | 8 | 1 | 1 | 0.00 | 22.62 |
| **13** | 9 | 2 | 1 | 194.01 | 926.00 |
| **14** | 9 | 0 | 0 | 0.00 | NULL |
| **15** | 2 | 3 | 0 | 1351.63 | 1351.63 |
| **16** | 7 | 1 | 1 | 0.00 | 201.02 |
| **17** | 1 | 2 | 1 | 849.88 | 855.53 |
| **18** | 6 | 0 | 0 | 0.00 | NULL |
| **19** | 6 | 1 | 0 | 214.44 | 214.44 |
| **20** | 2 | 1 | 0 | 156.08 | 156.08 |
| **21** | 10 | 1 | 1 | 0.00 | 635.44 |
| **22** | 3 | 1 | 0 | 565.81 | 565.81 |
| **23** | 8 | 0 | 0 | 0.00 | NULL |
| **24** | 6 | 0 | 0 | 0.00 | NULL |

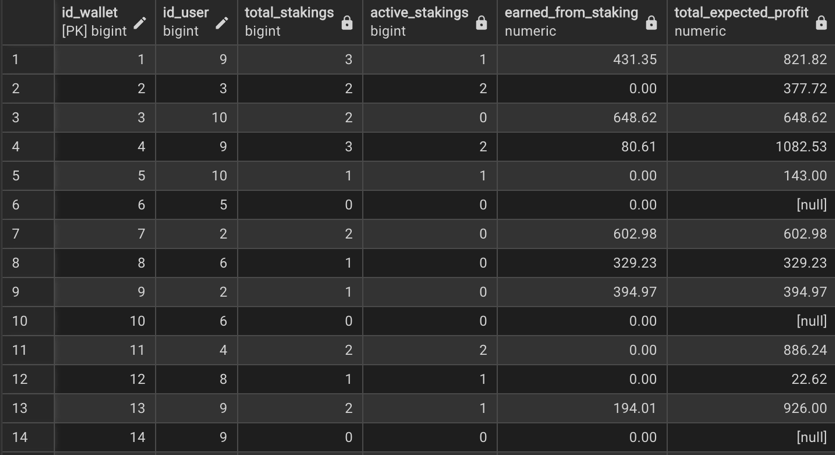


Рисунок 2 - Результат выполнения второго запроса

Чтобы убедиться в корректности, выведем все необходимые данные по пользователю WHERE id = 1.

Вспомогательный вывод:

SELECT

s.id\_staking,

s.id\_wallet,

s.staking\_start\_date,

s.staking\_end\_date,

s.staking\_sum,

s.staking\_percentage,

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

CASE

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

ELSE 'Active'

END AS staking\_status

FROM

staking s

WHERE

s.id\_wallet = 1

ORDER BY

s.staking\_start\_date;

Таблица 3 - Вывод стейкингов для случайного пользователя

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **id\_staking** | **id\_wallet** | **staking\_start\_date** | **staking\_end\_date** | **staking\_sum** | **staking\_percentage** | **expected\_profit** | **staking\_status** |
| **21** | 1 | 2024-04-22 | 2024-04-27 | 3706.32 | 8.26 | 306.14 | Closed |
| **30** | 1 | 2024-05-07 | 2024-05-28 | 2995.52 | 4.18 | 125.21 | Closed |
| **2** | 1 | 2024-11-07 | 2025-08-08 | 2263.54 | 17.25 | 390.46 | Active |

Запрос 3 Получение топ-5 криптовалют по объёму транзакций за месяц, с детализацией по пользователям и группировкой результатов.

Cначала группируются все транзакции одного пользователя, вторым этапом группируются все суммы сум транзакций по пользователям по заданной криптовалюте, формируется одно единое число по обороту криптовалют, и оно характеризует оборот криптовалют за текущий месяц.

Пример:

Пользователь 1:

покупка 1: Ton 20, дата: 10.04.2025 по цене 2.97

покупка 2: Ton 10, дата: 12.04.2025 по цене 2.97

продажа 1: Ton 10, дата: 13.04.2025 по цене 2.97

Общий оборот криптовалюты 40 Ton по пользователю за месяц

Пользователь 2:

покупка 1: Ton 30, дата: 01.04.2025 по цене 2.97

покупка 2: Ton 20, дата: 11.04.2025 по цене 2.97

продажа 1: Ton 50, дата: 10.04.2025 по цене 2.97

Общий оборот криптовалюты 100 Ton по пользователю за месяц

Итого оборото то за месяц сумма по всем пользователям: 140

Ton | 140

Запрос 3

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;

Таблица 4 - Результат выполнения запроса 3

|  |  |  |
| --- | --- | --- |
| **currency\_name** | **total\_turnover** | **turnover\_usd** |
| **BTC** | 37763.14 | 3197228124.00 |
| **ETH** | 11336.55 | 18141540.87 |
| **SOL** | 16385.72 | 2085902.16 |
| **AVAX** | 2090.06 | 40735.19 |
| **XRP** | 10324.00 | 21886.88 |

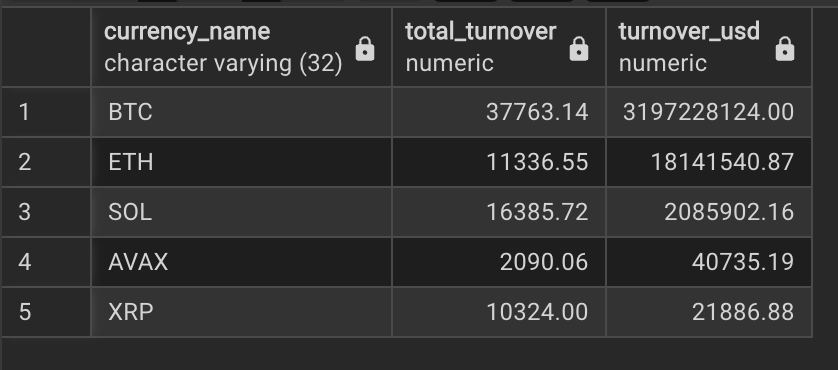


Рисунок 3 - Результат выполнения запроса 3

Заключение

SQL запросы позволяет очень удобно группировать информацию и обрабатывать её с помощью агрегатных функций и запросов. Это даёт возможность быстро получать обобщённые данные — например, подсчитать суммы, средние значения, количество записей и многое другое — всего за пару строк кода.

Приложение 1

CREATE TABLE users (

id BIGINT NOT NULL PRIMARY KEY,

name VARCHAR(64) NOT NULL,

email VARCHAR(64) NOT NULL

);

CREATE TABLE spot\_wallet (

id\_wallet BIGINT NOT NULL PRIMARY KEY,

currency VARCHAR(32) NOT NULL,

balance FLOAT NOT NULL,

id\_user BIGINT NOT NULL,

CONSTRAINT id\_user\_fk FOREIGN KEY (id\_user) REFERENCES users(id)

);

CREATE TABLE cryptocurrency (

id\_currency BIGINT NOT NULL PRIMARY KEY,

currency\_name VARCHAR(32) NOT NULL,

currency\_price\_usd FLOAT NOT NULL

);

CREATE TABLE orders (

id\_order BIGINT NOT NULL PRIMARY KEY,

id\_users BIGINT NOT NULL,

id\_currency BIGINT NOT NULL,

order\_count\_currency FLOAT NOT NULL,

order\_type BOOLEAN NOT NULL,

order\_status BOOLEAN NOT NULL,

CONSTRAINT id\_order\_user\_fk FOREIGN KEY (id\_users) REFERENCES users(id),

CONSTRAINT id\_order\_currency\_fk FOREIGN KEY (id\_currency) REFERENCES cryptocurrency(id\_currency)

);

CREATE TABLE transactions (

id\_trans BIGINT NOT NULL PRIMARY KEY,

id\_wallet BIGINT NOT NULL,

id\_order BIGINT NOT NULL,

id\_currency BIGINT NOT NULL,

trans\_data DATE NOT NULL,

trans\_status BOOLEAN NOT NULL,

CONSTRAINT id\_transactions\_wallet\_fk FOREIGN KEY (id\_wallet) REFERENCES spot\_wallet(id\_wallet),

CONSTRAINT id\_transactions\_order\_fk FOREIGN KEY (id\_order) REFERENCES orders(id\_order),

CONSTRAINT id\_transactions\_currency\_fk FOREIGN KEY (id\_currency) REFERENCES cryptocurrency(id\_currency)

);

CREATE TABLE account\_oper (

id\_account\_oper BIGINT NOT NULL PRIMARY KEY,

id\_wallet BIGINT NOT NULL,

account\_oper\_sum FLOAT NOT NULL,

account\_oper\_currency VARCHAR(32) NOT NULL,

account\_oper\_status BOOLEAN NOT NULL,

account\_oper\_type BOOLEAN NOT NULL,

CONSTRAINT id\_account\_oper\_wallet\_fk FOREIGN KEY (id\_wallet) REFERENCES spot\_wallet(id\_wallet)

);

CREATE TABLE staking (

id\_staking BIGINT NOT NULL PRIMARY KEY,

id\_wallet BIGINT NOT NULL,

staking\_start\_date DATE NOT NULL,

staking\_end\_date DATE NOT NULL,

staking\_sum FLOAT NOT NULL,

staking\_percentage FLOAT NOT NULL,

CONSTRAINT id\_staking\_wallet\_fk FOREIGN KEY (id\_wallet) REFERENCES spot\_wallet(id\_wallet)

);

ALTER TABLE spot\_wallet DROP CONSTRAINT id\_user\_fk;

ALTER TABLE orders DROP CONSTRAINT id\_order\_user\_fk;

ALTER TABLE orders DROP CONSTRAINT id\_order\_currency\_fk;

ALTER TABLE transactions DROP CONSTRAINT id\_transactions\_wallet\_fk;

ALTER TABLE transactions DROP CONSTRAINT id\_transactions\_order\_fk;

ALTER TABLE transactions DROP CONSTRAINT id\_transactions\_currency\_fk;

ALTER TABLE account\_oper DROP CONSTRAINT id\_account\_oper\_wallet\_fk;

ALTER TABLE staking DROP CONSTRAINT id\_staking\_wallet\_fk;

ALTER TABLE spot\_wallet ADD CONSTRAINT id\_user\_fk FOREIGN KEY (id\_user) REFERENCES users(id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE orders ADD CONSTRAINT id\_order\_user\_fk FOREIGN KEY (id\_users) REFERENCES users(id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE orders ADD CONSTRAINT id\_order\_currency\_fk FOREIGN KEY (id\_currency) REFERENCES cryptocurrency(id\_currency) ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE transactions ADD CONSTRAINT id\_transactions\_wallet\_fk FOREIGN KEY (id\_wallet) REFERENCES spot\_wallet(id\_wallet) ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE transactions ADD CONSTRAINT id\_transactions\_order\_fk FOREIGN KEY (id\_order) REFERENCES orders(id\_order) ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE transactions ADD CONSTRAINT id\_transactions\_currency\_fk FOREIGN KEY (id\_currency) REFERENCES cryptocurrency(id\_currency) ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE account\_oper ADD CONSTRAINT id\_account\_oper\_wallet\_fk FOREIGN KEY (id\_wallet) REFERENCES spot\_wallet(id\_wallet) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE staking ADD CONSTRAINT id\_staking\_wallet\_fk FOREIGN KEY (id\_wallet) REFERENCES spot\_wallet(id\_wallet) ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE transactions ADD CONSTRAINT unique\_transaction\_order UNIQUE (id\_order);

Приложение 2

INSERT INTO users (id, name, email) VALUES

(1, 'achang', 'tammy76@faulkner-howard.com'),

(2, 'juancampos', 'kyleblair@hotmail.com'),

(3, 'thomas15', 'cheryl38@yahoo.com'),

(4, 'qgrimes', 'davismary@summers.info'),

(5, 'john51', 'johnponce@west.net'),

(6, 'jrodriguez', 'kelleylisa@hotmail.com'),

(7, 'tamaramorrison', 'zdavis@yahoo.com'),

(8, 'barbara42', 'antoniozavala@alvarez-davis.com'),

(9, 'sandrakirby', 'aimee33@hotmail.com'),

(10, 'criley', 'christopher91@yahoo.com');

INSERT INTO cryptocurrency (id\_currency, currency\_name, currency\_price\_usd) VALUES

(1, 'BTC', 84665.31),

(2, 'ETH', 1600.27),

(3, 'SOL', 127.3),

(4, 'DOGE', 0.15),

(5, 'AVAX', 19.49),

(6, 'ARB', 0.28),

(7, 'APT', 4.66),

(8, 'TON', 2.97),

(9, 'XRP', 2.12),

(10, 'TRX', 0.25);

INSERT INTO spot\_wallet (id\_wallet, currency, balance, id\_user) VALUES

(1, 'USD', 90081.13, 9),

(2, 'USD', 1392.74, 3),

(3, 'USD', 7558.04, 10),

(4, 'USD', 194.03, 9),

(5, 'USD', 982.85, 10),

(6, 'USD', 9021.66, 5),

(7, 'USD', 987.63, 2),

(8, 'USD', 8988.38, 6),

(9, 'USD', 4721.43, 2),

(10, 'USD', 3537.91, 6),

(11, 'USD', 6108.87, 4),

(12, 'USD', 9666.06, 8),

(13, 'USD', 4426.93, 9),

(14, 'USD', 2604.92, 9),

(15, 'USD', 9159.94, 2),

(16, 'USD', 7197.05, 7),

(17, 'USD', 702.53, 1),

(18, 'USD', 6118.97, 6),

(19, 'USD', 20439.11, 6),

(20, 'USD', 7036.43, 2),

(21, 'USD', 10.67, 10),

(22, 'USD', 2017.04, 3),

(23, 'USD', 8031.79, 8),

(24, 'USD', 912.16, 6);

INSERT INTO orders (id\_order, id\_users, id\_currency, order\_count\_currency, order\_type, order\_status) VALUES

(1, 2, 5, 15876.23, TRUE, FALSE),

(2, 9, 4, 2033.76, FALSE, FALSE),

(3, 2, 10, 19124.44, FALSE, FALSE),

(4, 5, 3, 14239.51, TRUE, TRUE),

(5, 10, 5, 2089.38, TRUE, TRUE),

(6, 3, 1, 17103.19, FALSE, TRUE),

(7, 9, 4, 12115.99, FALSE, FALSE),

(8, 8, 8, 8223.47, FALSE, FALSE),

(9, 6, 10, 19234.66, FALSE, TRUE),

(10, 4, 1, 6874.04, TRUE, TRUE),

(11, 6, 3, 2146.21, TRUE, TRUE),

(12, 3, 4, 10101.88, TRUE, TRUE),

(13, 2, 4, 17661.32, TRUE, TRUE),

(14, 2, 6, 12678.14, TRUE, FALSE),

(15, 10, 1, 13783.93, TRUE, TRUE),

(16, 8, 4, 5333.58, TRUE, TRUE),

(17, 9, 7, 2044.92, FALSE, TRUE),

(18, 4, 2, 11336.55, FALSE, TRUE),

(19, 3, 1, 19028.22, TRUE, FALSE),

(20, 7, 4, 8474.73, FALSE, TRUE);

(21, 4, 1, 0.9454, TRUE, TRUE),

(22, 6, 9, 10324, FALSE, TRUE),

(23, 1, 8, 3.4096, FALSE, TRUE),

(24, 7, 5, 0.7755, TRUE, FALSE),

(25, 2, 6, 3.6977, FALSE, FALSE),

(26, 4, 8, 1000.7676, FALSE, TRUE),

(27, 10, 10, 1000.6704, FALSE, FALSE),

(28, 7, 2, 2.0176, TRUE, FALSE),

(29, 3, 4, 1.1233, FALSE, FALSE),

(30, 10, 7, 0.1674, FALSE, FALSE),

(31, 3, 8, 0.3288, TRUE, TRUE),

(32, 9, 8, 4.5411, TRUE, FALSE),

(33, 8, 6, 1.5672, FALSE, TRUE),

(34, 7, 4, 2.7473, TRUE, TRUE),

(35, 1, 7, 4.7482, FALSE, TRUE),

(36, 1, 4, 2000.0813, TRUE, FALSE),

(37, 9, 10, 10000, TRUE, TRUE),

(38, 10, 4, 4.3695, FALSE, TRUE),

(39, 3, 2, 2.3833, FALSE, TRUE),

(40, 2, 1, 1.3808, FALSE, TRUE),

(41, 2, 5, 0.6758, FALSE, TRUE),

(42, 2, 3, 1.3993, TRUE, TRUE),

(43, 1, 1, 1.0366, FALSE, TRUE),

(44, 9, 6, 400.733, TRUE, TRUE),

(45, 10, 8, 3000.5647, FALSE, TRUE);

INSERT INTO transactions (id\_trans, id\_wallet, id\_order, id\_currency, trans\_data, trans\_status) VALUES

(1, 21, 4, 3, '2025-04-01', FALSE),

(2, 18, 5, 5, '2025-04-02', TRUE),

(3, 13, 6, 1, '2025-04-03', FALSE),

(4, 1, 9, 10, '2025-04-04', TRUE),

(5, 9, 10, 1, '2025-04-05', FALSE),

(6, 12, 11, 3, '2025-04-06', FALSE),

(7, 20, 12, 4, '2025-04-07', TRUE),

(8, 9, 13, 4, '2025-04-08', TRUE),

(9, 19, 15, 1, '2025-04-09', FALSE),

(10, 13, 16, 4, '2025-04-10', TRUE),

(11, 10, 17, 7, '2025-04-11', FALSE),

(12, 24, 18, 2, '2025-04-12', TRUE),

(13, 10, 20, 4, '2025-04-13', TRUE),

(14, 10, 21, 1, '2025-04-14', FALSE),

(15, 10, 22, 9, '2025-04-15', TRUE),

(16, 4, 23, 8, '2025-04-16', FALSE),

(17, 16, 26, 8, '2025-04-07', FALSE),

(18, 4, 31, 8, '2025-02-24', TRUE),

(19, 23, 33, 6, '2024-05-08', FALSE),

(20, 2, 34, 4, '2024-05-23', FALSE),

(21, 24, 35, 7, '2025-02-12', TRUE),

(22, 21, 37, 10, '2024-06-09', FALSE),

(23, 21, 38, 4, '2025-03-12', TRUE),

(24, 3, 39, 2, '2025-01-09', TRUE),

(25, 2, 40, 1, '2024-07-07', TRUE),

(26, 4, 41, 5, '2025-04-02', FALSE),

(27, 15, 42, 3, '2025-02-25', TRUE),

(28, 14, 43, 1, '2025-04-16', FALSE),

(29, 8, 44, 6, '2025-04-16', TRUE),

(30, 14, 45, 8, '2025-04-02', FALSE);

INSERT INTO staking (id\_staking, id\_wallet, staking\_start\_date, staking\_end\_date, staking\_sum, staking\_percentage) VALUES

(1, 4, '2025-04-12', '2025-05-19', 4111.68, 21.47),

(2, 1, '2024-11-07', '2025-08-08', 2263.54, 17.25),

(3, 4, '2024-12-03', '2025-08-19', 1995.79, 5.97),

(4, 2, '2025-01-03', '2025-04-28', 3120.84, 3.51),

(5, 15, '2024-05-12', '2024-11-26', 1814.59, 20.85),

(6, 4, '2024-04-24', '2025-01-03', 750.53, 10.74),

(7, 22, '2024-06-01', '2025-01-08', 4389.51, 12.89),

(8, 11, '2024-12-16', '2025-09-02', 4990.93, 16.24),

(9, 7, '2024-06-26', '2025-04-04', 3052.76, 6.25),

(10, 11, '2024-12-05', '2025-05-21', 4092.41, 1.85),

(11, 5, '2025-03-02', '2025-07-16', 3016.91, 4.74),

(12, 13, '2024-11-23', '2025-04-27', 3593.48, 20.37),

(13, 3, '2024-05-30', '2024-07-17', 2587.47, 22.97),

(14, 3, '2024-09-12', '2025-01-10', 661.17, 8.21),

(15, 15, '2024-05-19', '2024-11-09', 4310.9, 20.22),

(16, 21, '2024-09-04', '2025-05-02', 4843.31, 13.12),

(17, 17, '2025-03-03', '2025-11-20', 177.75, 3.18),

(18, 17, '2024-05-29', '2024-07-12', 3737.36, 22.74),

(19, 7, '2024-04-28', '2024-09-28', 2681.73, 15.37),

(20, 16, '2024-12-05', '2025-06-26', 3041.08, 6.61),

(21, 1, '2024-04-22', '2024-04-27', 3706.32, 8.26),

(22, 19, '2024-06-28', '2024-11-10', 1670.1, 12.84),

(23, 9, '2024-06-24', '2024-12-01', 3864.68, 10.22),

(24, 13, '2025-01-17', '2025-02-22', 827.34, 23.45),

(25, 8, '2024-06-01', '2024-10-30', 3649.96, 9.02),

(26, 2, '2024-10-17', '2025-06-25', 2095.15, 12.8),

(27, 20, '2024-07-04', '2024-08-19', 3370.98, 4.63),

(28, 12, '2025-01-21', '2025-08-24', 185.55, 12.19),

(29, 15, '2025-02-18', '2025-03-19', 516.39, 19.68),

(30, 1, '2024-05-07', '2024-05-28', 2995.52, 4.18);

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

(1, 11, 114.24, 'USD', FALSE, TRUE),

(2, 13, 785.58, 'USD', FALSE, TRUE),

(3, 2, 614.02, 'USD', FALSE, FALSE),

(4, 21, 133.08, 'USD', FALSE, TRUE),

(5, 13, 801.35, 'USD', TRUE, TRUE),

(6, 2, 784.29, 'USD', FALSE, FALSE),

(7, 7, 460.92, 'USD', TRUE, TRUE),

(8, 2, 491.3, 'USD', TRUE, TRUE),

(9, 8, 102.54, 'USD', FALSE, FALSE),

(10, 4, 154.23, 'USD', FALSE, TRUE),

(11, 4, 421.48, 'USD', TRUE, FALSE),

(12, 5, 736.94, 'USD', TRUE, FALSE),

(13, 14, 689.55, 'USD', TRUE, FALSE),

(14, 11, 725.32, 'USD', TRUE, FALSE),

(15, 3, 130.16, 'USD', TRUE, FALSE),

(16, 12, 186.13, 'USD', TRUE, FALSE),

(17, 3, 600.62, 'USD', TRUE, TRUE),

(18, 1, 212.74, 'USD', TRUE, TRUE),

(19, 10, 375.44, 'USD', TRUE, TRUE),

(20, 5, 195.12, 'USD', TRUE, FALSE);

Приложение 3

Во время работы над практической работой 5, для реализации авторизации и регистрации в сущность users был добавлен атрибут password

ALTER TABLE users ADD COLUMN password VARCHAR(128);

UPDATE users SET password = 'abc12345' WHERE id = 1;

UPDATE users SET password = 'pass4321' WHERE id = 2;

UPDATE users SET password = 'hello123' WHERE id = 3;

UPDATE users SET password = 'qwerty89' WHERE id = 4;

UPDATE users SET password = 'testpass' WHERE id = 5;

UPDATE users SET password = 'letmein12' WHERE id = 6;

UPDATE users SET password = 'userpass' WHERE id = 7;

UPDATE users SET password = '1234abcd' WHERE id = 8;

UPDATE users SET password = 'easy9999' WHERE id = 9;

UPDATE users SET password = 'mypwd321' WHERE id = 10;

Приложение 4

Сделать автоматическую автоинкрементацию для spot\_wallet

DROP TABLE IF EXISTS spot\_wallet CASCADE;

CREATE TABLE spot\_wallet (

id\_wallet BIGSERIAL PRIMARY KEY, -- автоинкремент, без всякого setval

currency VARCHAR(32) NOT NULL,

balance FLOAT NOT NULL,

id\_user BIGINT NOT NULL,

CONSTRAINT id\_user\_fk FOREIGN KEY (id\_user) REFERENCES users(id) ON DELETE CASCADE ON UPDATE CASCADE

);

Заполнение

INSERT INTO spot\_wallet (currency, balance, id\_user) VALUES

('USD', 90081.13, 9),

('USD', 1392.74, 3),

('USD', 7558.04, 10),

('USD', 194.03, 9),

('USD', 982.85, 10),

('USD', 9021.66, 5),

('USD', 987.63, 2),

('USD', 8988.38, 6),

('USD', 4721.43, 2),

('USD', 3537.91, 6),

('USD', 6108.87, 4),

('USD', 9666.06, 8),

('USD', 4426.93, 9),

('USD', 2604.92, 9),

('USD', 9159.94, 2),

('USD', 7197.05, 7),

('USD', 702.53, 1),

('USD', 6118.97, 6),

('USD', 20439.11, 6),

('USD', 7036.43, 2),

('USD', 10.67, 10),

('USD', 2017.04, 3),

('USD', 8031.79, 8),

('USD', 912.16, 6);

Важно в ходе написания конечного проекта, авто присвоение id было также сделано, users, orders, transactions, staking, account\_oper, у wallets был вообще изменен приоритет.

Для изменения правил формирования id.

ALTER TABLE staking

ALTER COLUMN id\_staking ADD GENERATED BY DEFAULT AS IDENTITY;

Устанавливает начальное значение.

ALTER TABLE staking

ALTER COLUMN id\_staking RESTART WITH 6;