

Студ. Т.Л. Нгуен

Науч. рук.: доц. Н.Н. Пустовалова

(Кафедра информационных систем и технологий, БГТУ);

доц. Н.П. Коровкина

(Кафедра автоматизации производственных процессов и электротехники, БГТУ)

ИСПОЛЬЗОВАНИЕ СУБД MATERIALIZE В ФИНАНСОВОЙ СФЕРЕ

Современный мир насыщен различными данными, которые нуждаются в хранении, обработке и анализе. Финансовые учреждения, такие как банки, являются одними из крупнейших потребителей такой информации. Для обработки больших объемов данных и принятия быстрых решений финансовым учреждениям нужны высокопроизводительные системы управления базами данных (СУБД).

Одним из новых инструментов в этой области является Materialize. Он относится к семейству СУБД с потоковой обработкой данных (stream processing), которые получили распространение в последние годы. Materialize позволяет обрабатывать большие объемы данных в режиме реального времени и быстро извлекать информацию для анализа, обладает множеством преимуществ по сравнению с традиционными СУБД и может использоваться в различных отраслях, включая финансовые операции.

Данная СУБД предоставляет простой и интуитивно понятный SQL-интерфейс, который позволяет пользователям легко извлекать информацию из базы данных. Он также обладает мощной системой оптимизации запросов, которая обеспечивает быстрое выполнение запросов на больших объемах данных.

Рассмотрим примеры использования СУБД Materialize для решения задач в финансовой сфере.

Мониторинг активности клиентов в финансовой организации. Пусть имеется таблица "transactions" с информацией о финансовых транзакциях, которая включает следующие столбцы: *transaction_id* – уникальный идентификатор транзакции; *account_id* – идентификатор счета клиента; *transaction_date* – дата транзакции; *transaction_type* – тип транзакции (например, "покупка", "перевод"); *transaction_amount* – сумма транзакции.

Нужно выявить наиболее активных клиентов.

Для этого можно использовать материализованное представление Materialized View, которое будет отображать все транзакции, совершенные каждым клиентом за месяц, и сравнивать эту информацию с общим количеством транзакций клиента за все время:

```

CREATE MATERIALIZED VIEW monthly_transactions
AS SELECT account_id, COUNT(*) as transactions_count
FROM transactions
WHERE transaction_date >= DATE_TRUNC ('month',
CURRENT_DATE) GROUP BY account_id;

```

Этот запрос создает представление с названием "monthly_transactions", которое вычисляет количество транзакций, совершенных каждым клиентом за текущий месяц. Представление можно использовать для поиска наиболее активных клиентов с помощью запроса, определяющего все аккаунты, количество транзакций которых за текущий месяц превышает среднее количество транзакций:

```

SELECT * SELECT account_id, transactions_count
FROM monthly_transactions
WHERE transactions_count >
(SELECT AVG(transactions_count) FROM monthly_transactions);

```

Таким образом, Materialized View позволяет быстро обрабатывать большие объемы данных и выявлять наиболее активных клиентов, что делает его полезным инструментом в мониторинге активности клиентов в финансовой организации.

Аналитика клиентской базы. Предположим, имеется база данных клиентов с таблицей "customers", содержащей следующие столбцы: *customer_id* – уникальный идентификатор клиента; *first_name* – имя клиента; *last_name* – фамилия клиента; *email* – адрес электронной почты клиента; *phone* – номер телефона клиента; *registration_date* – дата регистрации клиента в системе; *account_balance* – баланс на счете клиента.

Следует создать материализованное представление Materialized View, которое будет отображать информацию о клиентах с отрицательным балансом на счете:

```

CREATE MATERIALIZED VIEW negative_balance_customers
AS SELECT * FROM customers WHERE account_balance < 0;

```

Данное представление можно использовать в запросе, определяющем клиентов с отрицательным балансом и датой последней транзакции:

```

SELECT c.first_name, c.last_name, c.email, c.phone,
c.registration_date, c.account_balance, MAX(t.transaction_date) as
last_transaction_date
FROM negative_balance_customers c JOIN transactions t
ON c.customer_id = t.customer_id
GROUP BY c.customer_id ORDER BY last_transaction_date DESC;

```

Полученную в результате выполнения запроса информацию можно использовать для дополнительного анализа и принятия соответствующих мер, например, связанных с блокировкой счетов или уведомлением клиентов о необходимости пополнения счета.

Риск-менеджмент. Используется та же таблица, что была приведена при мониторинге активности клиентов.

Задача – мониторить транзакции и быстро выявлять любые подозрительные операции, которые могут указывать на мошенническую деятельность. Для этого надо создать материализованное представление, которое будет отображать все транзакции, где сумма операции превышает заданный порог:

```
CREATE MATERIALIZED VIEW high_value_transactions
AS SELECT * FROM transactions
WHERE transaction_amount > 1000;
```

Этот запрос создает представление с названием "high_value_transactions", которое определяет те транзакции из таблицы "transactions", где сумма операции превышает 1000.

Теперь можно использовать это представление для мониторинга денежных операций с большими суммами и быстрого обнаружения потенциальных проблем:

```
SELECT * FROM high_value_transactions
WHERE account_id IN (SELECT account_id FROM suspicious_accounts);
```

Этот запрос показывает все денежные операции, произведенные на счетах клиентов, которые были отмечены как подозрительные. Можно использовать эту информацию для дополнительного анализа и принятия соответствующих мер.

Materialized View позволяет быстро обрабатывать большие объемы данных и быстро реагировать на потенциальные проблемы, что делает его полезным инструментом в риск-менеджменте.

Таким образом, использование СУБД Materialize в финансовых операциях имеет множество преимуществ для финансовых учреждений. Materialize позволяет обрабатывать большие объемы данных в режиме реального времени, улучшать процесс принятия решений и снизить затраты на обработку данных.

Система управления базами данных Materialize полностью совместима со стандартом SQL, что делает ее легко интегрируемой с другими приложениями.

Этот инструмент может быть особенно полезен в анализе данных о финансовых рынках и прогнозировании их изменений.