simbioz

BI SQL

Решение проблем растущего бизнеса с помощью Simbioz ERPM

Оглавление

1. Определение типа счете
2. Общие настройки
3. Создание SQL запроса
4. Определение измерений и значений куба
5. Задание отображаемого имени поля
6. Задание типа поля
7. Прочие настройки
8. Настройка прав доступа
9. Создание отчета
10. Настройка автоматического обновления
11. Оплата счетов (Settlements with partners)
12. Настройки

1. Определение типа отчёта

Система позволяет анализировать данные в 2-х режимах (поле **“Is Materialized View”):**

- Режим “**Table”** (по умолчанию) позволяет кешировать большие объёмы данных во вспомогательную таблицу, но не даёт возможности видеть текущую ситуацию. Обновление данных может производиться вручную или по расписанию (см. ниже).
- Режим **“View”** позволяет видеть актуальные данные, но при обработке больших массивов или сложных запросов может приводить к большой нагрузке на сервер и большим задержкам.

2. Общие настройки

Техническое имя отчёта (поле “**Technical Name”)** ограниченно по длине. Системный префикс достаточно длинен, но 3-хзначные комбинации пропускает. При превышении длины имени во время выполнения различных действий над настройкой будет получено сообщение об ошибке.

3. Создание SQL запроса (Требование к запросу)

Имена всех полей, отображение которых необходимо, должны начинаться с префикса “x_”. При разделении частей имени поля символом “_” система автоматически разобьёт отображаемое имя. (См. ниже)

Имена полей и их список должны быть продуманы изначально. Нежелательно переименовывать или удалять поля, если настройка прошла этап “**Validate SQL Expression”.** (Подробнее см. ниже).

Обращение к таблицам происходит по именам моделей. При этом необходимо указать префикс имени “**public.”,** а символы “.” в имени заменить на символы “_”.

В результате использования в моделях Odoo механизма наследования некоторые поля модели могут храниться не в ее, а в родительской таблице. В описании моделей этот факт отдельно не раскрывается, из-за чего

необходимо исследовать не только структуру моделей, но и с помощью стороннего ПО структуру самой базы данных.

Порядок объявления полей в запросе влияет на порядок структуры измерений, действующих по умолчанию при открытии отчёта.

При создании запроса необходимо учитывать специфику работы **OLAP-куба.** При отображении данных он суммирует значения разных измерений в соответствии с настройкой. Т.о. вычисление в запросе процентов или отношений (весов) в общем случае бессмысленно.

Нет необходимости в запросе ссылочные коды на другие таблицы в измерениях преобразовывать в текстовое представление. Система может автоматически преобразовывать коды в запросе в поля “name” соответствующих таблиц. (См. ниже).

Если запрос состоит из нескольких подзапросов, объединённых SQL командой “UNION”[1], то все подзапросы должны содержать все поля как измерений, так и значений. Если это идеологически невозможно, то:

- Для таких измерений проводить преобразование кода в текст, а для тех подзапросов, где эти измерения отсутствуют, задавать некое предопределённое текстовое значение вручную.
- Для таких значений использовать SQL конструкцию “CAST(0.0 AS float)”.

Необходимо учитывать, что в некоторых строках запроса могут быть получены пустые (“NULL”) значения. Для избежания этого рекомендуется использование конструкции “CAST(COALESCE(<имя поля>, 0.0) AS float)”.

[1]См. документацию: https://postgrespro.ru/docs/postgrespro/9.6/

Для уменьшения нагрузки желательно отфильтровывать строки запроса, у которых все значения равны 0, с помощью конструкции “WHEN”. Также результирующие строки желательно группировать с помощью конструкции “GROUP BY”.

Переход к следующему этапу и проверка синтаксиса запроса осуществляется по нажатию кнопки “Validate SQL Expression”. Возврат к режиму редактирования запроса на любом этапе возможен по нажатию кнопки “Set Draft”.

4. Определение измерений и значений куба

При проведении синтаксической проверки текста запроса модуль распознает список заданных полей, которые отображаются во вкладке “**SQL Fields”.**

При этом следует учесть, что если ранее был возврат к коррекции текста запроса с изменением имён полей или их удалением, то их настройки не отображаются, но хранятся. Это препятствует переходу к следующим этапам настройки отчёта. Лёгким вариантом выхода из такой ситуации может быть переименование отчёта. Удаление и повторное создание под тем же именем результата может не дать.

5. Задание отображаемого имени поля

Не рекомендуется использование в отображаемых именах полей (“**Field Description”)** кириллических символов. Это может привести к ошибке при открытии отчёта. Причём такая ошибка может возникать в режиме **“Table”,** но не возникать в режиме “**View”,** при одних и тех же настройках.

6. Задание типа поля

В общем случае переопределение типа поля (“Field Type”) применяется для:

- Переопределения типа “datetime” (использование может приводить к ошибкам) в тип “date”.
- Преобразования кодов измерений в имена, используя тип “many2one” и определение соответствующей модели (“Model”).

7. Прочие настройки

Для запроса с типом “Table” имеется возможность оптимизации размера/скорости путём создания индексов по измерениям (“Is Index”).

В интерфейсе отчёта могут быть созданы кнопки группировок (“Is Group by”).

Преднастройка вида отчёта по умолчанию может быть выполнена (“Graph Type”):

- Для измерений типами “Row” и “Column”.
- Для значений типом “Measure”.

8. Настройка прав доступа

Настройка прав доступа осуществляется на вкладке “Secutity” согласно общим правилам.

9. Создание отчёта

По нажатию кнопки “Create SQL View, Indexes and Models” производится генерация внутренних структур модуля.

Кнопкой “Create UI” производится генерация пункта меню в пути “Reporting — SQL Reports”.

В режиме “Table” кнопка “Refresh Materialized View” позволяет вручную заполнить/обновить содержимое отображаемой таблицы.

Кнопка “Open View” позволяет из формы модуля проверить функционирование отчёта.

10. Настройка автоматического обновления

В режиме “Table” имеется возможность настройки автоматического обновления по расписанию “Odoo Cron”.

Форма имеет следующие настройки:

- “Active” включает/выключает обновление по расписанию
- “User” задаёт пользователя, от имени которого будет выполняться обновление

- “Priority” задаёт приоритет выполнения обновления
- “Interval Number” указывает через сколько интервалов будет повторяться запуск обновления
- “Interval Unit” определяет тип интервала
- “Number of Calls” устанавливает количество повторений (-1 — постоянно, 0 — отключено)
- “Repeat Missed” включает/выключает принудительное выполнение пропущенных заданий.

11. Оплаты счетов (Settlements with partners)

Ограничения реализации

Анализ оплат ведется по элементам модели account.voucher („Sales Receipts“, „Customer Payments“, „Purchase Receipts“, „Supplier Payments“).

12. Настройки

Название: Settlements with partners
Техническое название: swp
Таблица: -
Запрос:

```
SELECT
    q.company AS x_company,
    q.journal AS x_journal,
    q.partner AS x_partner,
    q.type AS x_type,
    q.voucher AS x_voucher,
    q.invoice AS x_invoice,
    q.date AS x_date,
    SUM(q.original) AS x_original,
    SUM(q.payment) AS x_payment,
    SUM(q.unreconciled - q.payment) AS x_unreconciled
FROM
```

```
(SELECT
    av.company_id AS company,
    av.journal_id AS journal,
    av.partner_id AS partner,
    av.type AS type,
    av.id AS voucher,
    (COALESCE(am.name, '<PREPAID>') || CASE WHEN (am.ref IS NOT NULL AND am.ref != '') THEN ' (' || am.ref || ')' ELSE '' END) AS invoice,
    av.date AS date,
    SUM((CASE WHEN (av.type = 'payment' OR av.type = 'sale') THEN (0.0 - COALESCE(avl.amount_original, 0.0)) ELSE COALESCE(avl.amount_original, 0.0) END)) AS original,
    SUM((CASE WHEN (av.type = 'payment' OR av.type = 'sale') THEN (0.0 - avl.amount) ELSE avl.amount END)) AS payment,
    SUM((CASE WHEN (av.type = 'payment' OR av.type = 'sale') THEN (0.0 - COALESCE(avl.amount_unreconciled, 0.0)) ELSE COALESCE(avl.amount_unreconciled, 0.0) END)) AS
    unreconciled
FROM
    public.account_voucher AS av
    LEFT JOIN public.account_voucher_line AS avl ON
        avl.voucher_id = av.id
    LEFT JOIN public.account_move_line AS aml ON
        aml.id = avl.move_line_id
    LEFT JOIN public.account_move AS am ON
        am.id = aml.move_id
WHERE
    (av.state = 'posted') AND
    (avl.amount IS NOT NULL) AND
    (avl.amount != 0.0)
GROUP BY
    av.company_id,
    av.journal_id,
    av.partner_id,
    av.type,
    av.id,
    invoice,
    av.date
UNION ALL
SELECT
    id AS company,
    av.journal_id AS journal,
```

```
        av.partner_id AS partner,
        av.type AS type,
        av.id AS voucher,
        '<PREPAID>' AS invoice,
        av.date AS date,
        SUM(CAST(0.0 AS float)) AS original,
        SUM((CASE WHEN (av.type = 'payment' OR av.type = 'sale') THEN (0.0 - (av.amount - avl.payment)) ELSE (av.amount - avl.payment) END)) AS payment,
        SUM(CAST(0.0 AS float)) AS unreconciled
    FROM
        public.account_voucher AS av
        LEFT JOIN
            (SELECT
                voucher_id,
                SUM(COALESCE(amount, 0.0)) AS payment
            FROM
                public.account_voucher_line
            GROUP BY
                voucher_id) AS avl ON
                avl.voucher_id = av.id
    WHERE
        (av.state = 'posted')
    GROUP BY
        av.company_id,
        av.journal_id,
        av.partner_id,
        av.type,
        av.id,
        invoice,
        av.date) AS q
WHERE
    ((q.original != 0.0) OR
        (q.payment != 0.0) OR
        (q.unreconciled != 0.0))
GROUP BY
    x_company,
    x_journal,
```

x_partner,
x_type,
x_voucher,
x_invoice,
x_dat

Поля SQL:

Имя	SQL тип	Описание	Тип поля	Модель	Индекс	Группа	Тип графика
x_company	integer	Company	many2one	Companies		√	Row
x_journal	integer	Journal[2]	many2one	Journal		√	Row
x_partner	integer	Partner	many2one	Partner		√	Row
x_type	character varying	Type[3]	char			√	Row
x_voucher	integer	Voucher	many2one	Accounting Voucher		√	Row
x_invoice	text	Invoice	char			√	Row
x_date	date	Date	date			√	Column
x_original	double precision	Original[4]	float				Measure
x_payment	numeric	Payment[5]	float				Measure
x_unreconciled	double precision	Unreconciled[6]	float				Measure

[2]Предполагается к использованию в понятии расчётного счета.
[3]Тип оплаты.
[4]Сумма по счету.
[5]Сумма оплаты.
[6]Остаток к оплате.

Будем рады ответить на все Ваши вопросы!

simbioz

E-mail: lead@simbioz.ua
www.simbioz.ua