

## Метод реализации статистических отчетов с поддержкой детализации

С. А. Петров

Национальный исследовательский университет "МЭИ"  
111250, Россия, Москва, Красноказарменная ул., 14.

e-mail: PetrovSA@mpei.ru

*Аннотация.* Представлен один из вариантов реализации типовой для информационных систем задачи — построение сводного статистического отчета. Рассматриваются аспекты применения технологии OLAP. Предлагается альтернативный подход реализации сводных отчетов, не требующий внедрения OLAP-системы. На практическом примере разработки отчета ВПО проводится сравнение стандартного варианта решения для реляционных баз данных — написание команды на структурированном языке запросов (T-SQL) и предлагаемого метода — реализации процедуры, позволяющей получить как сводные данные, так и детализацию каждой отдельной ячейки сводки. Представлена блок-схема процедуры и выдержки программного кода. На основании апробации в информационных системах НИУ МЭИ и представленных результатах тестирования на выборках различного размера выделены плюсы и минусы метода, даны рекомендации по использованию и перспективные направления развития.

*Ключевые слова:* информационная система, статистика, отчет, сводка, детализация, структурированный запрос.

### 1. Введение

Современное общество не представляет своего существования без информационных технологий. В повседневной жизни любого человека окружает множество информационных систем (ИС), обрабатывающих данные о различных аспектах его существования. Не исключением является и сфера образования [1]. Одна из актуальных задач для современных ИС — это предоставление различных аналитических сводок. Высшие учебные заведения обрабатывают сведения о контингенте сотрудников, студентов, аспирантов и ежегодно перед ними встает задача по заполнению формы федерального статистического наблюдения ВПО-1, состоящей из множества сводных отчетов [2]. Такие отчеты позволяют увидеть состояние дел в вузе на 1 октября рассматриваемого года в различных разрезах.

При решении этой задачи для сверки данных часто требуется увидеть реальных людей, скрывающихся за сводными числами. Статья рассматривает одно из возможных решений этой проблемы — процедуру на языке SQL, позволяющую

определить как совокупное значение для каждой позиции отчета, так и их отдельные составляющие (например, конкретных сотрудников). Предлагаемый метод имеет свои плюсы и минусы, о которых будет рассказано ниже.

Подход был апробирован при подготовке отчетов ВПО по контингенту сотрудников и студентов федерального государственного бюджетного образовательного учреждения высшего образования Национального исследовательского университета МЭИ (ФГБОУ ВО НИУ МЭИ).

В НИУ МЭИ функционирует ИС Кадры и ИС Студент, в которых ведется учет сотрудников и студентов, имеющих отношение к деятельности вуза [3, 4]. Системы хранят информацию об активном контингенте (порядка 3500 работающих сотрудников и 13000 обучающихся студентов) и неактивном контингенте (порядка 12000 уволенных сотрудников и 50000 студентов, завершивших обучение) НИУ МЭИ. В качестве системы управления базами данных (СУБД) выступает Microsoft SQL Server 2012 R2, хорошо зарекомендовавшая себя в решении задач масштаба предприятий. Данный программный продукт поддерживает реализацию языка структурированных запросов под названием Transact-SQL (T-SQL). Все примеры программного кода будут представлены на данном диалекте.

## 2. Технология OLAP

Типовым вариантом решения задачи анализа данных с помощью сводных таблиц является применение технологии OLAP (online analytical processing). Подход позволяет сократить время на получение сводных значений и, при необходимости, произвести анализ с помощью детализации ячеек (операции drill through и drill down). Достигается это за счет вычисления и сохранения агрегированных значений. Возможны различные способы хранения исходных данных и агрегаций со своими плюсами и недостатками, среди которых обычно выделяют следующие [5]:

- MOLAP (multidimensional OLAP, многомерный OLAP) — исходные данные и агрегаты хранятся в отдельной многомерной базе. Скорость обработки запросов выше за счет работы с многомерными структурами, но построение и обновление многомерной базы данных (БД) требует соответствующих навыков и инструментов. Пример успешного внедрения можно найти в работе [6].
- ROLAP (relational OLAP, реляционный OLAP) — исходные данные и агрегаты хранятся в реляционных таблицах. Для агрегатов создаются отдельные таблицы, увеличивая размер исходной БД. Запросы на агрегированные значения адресуются к этим дополнительным таблицам. Данный подход был успешно применен при разработке системы, описанной в работе [7].

- HОLAP (hybrid OLAP, гибридный OLAP) — исходные данные хранятся в реляционной БД, агрегированные — в многомерной. Разделение позволяет работать с актуальными данными на нижнем уровне иерархии, но проигрывает по скорости MOLAP и требует поддержки многомерного и реляционного вариантов хранения.

В рассмотренных типах OLAP предполагается периодическое вычисление агрегатных значений и выделение дополнительной памяти для их хранения [8]. В связи с этим с многомерными БД тесно связаны проблема оценки объемов БД и риск «взрыва» данных, при котором размер базы агрегатов увеличивается экспоненциально и затрудняет использование OLAP [9]. Исключить данный риск позволяет технология R-ROLAP (real-time ROLAP, ROLAP реального времени), при использовании которой агрегаты вычисляются динамически. Этот вариант наиболее близок к предлагаемому решению, но требует навыков построения многомерных БД и написания к ним запросов.

В контексте OLAP нельзя не упомянуть популярную и быстроразвивающуюся технологию in-memory computing, которая обеспечивает хранение и обработку данных в оперативной памяти, и ее частный случай — in-memory OLAP. Использование оперативной памяти заметно увеличивает обработку запросов, но размер базы ограничен объемом доступной оперативной памяти.

Несомненно, технология OLAP является мощным инструментом для получения сводных таблиц и анализа данных с их помощью. В работах [10–12] описано применение технологии OLAP в сфере образования. В ходе начальных этапов разработки ИС Студент для НИУ МЭИ также имел место подход с использованием технологии OLAP, но в силу ряда причин было принято решение от него отказаться.

Как правило, выделяют следующие объективные сложности, с которыми сталкиваются предприятия при внедрении OLAP-систем [5, 13–15]:

- Высокие трудозатраты на разработку схемы хранилища данных и интеграцию с имеющимися системами. Для работы с многомерными БД необходимо знание технологий, отличных от используемых в реляционных СУБД, например языка запросов MDX. Как правило, необходимы и дополнительные аппаратные ресурсы.
- «Отставание» данных в многомерной базе OLAP от значений в оперативной БД, так как значения агрегатов рассчитываются в момент построения многомерной БД.
- Настроенная база OLAP не позволяет полностью исключить работу программистов при разработке регламентированных отчетных форм, включающих различные группировки по разным разделам, в том числе по иерар-

хиям, не предусмотренным в многомерной БД, и дополнительные вычисления. Самостоятельная работа с многомерной БД, например через Excel, также требует высокой квалификации пользователей.

Технология OLAP не является заменой стандартных средств реляционных БД для построения сводных отчетов, а представляет собой дополнение. Хотя это и типовое решение, но для внедрения необходимо рассматривать отдельно каждый конкретный случай. Предлагаемое решение является менее затратной альтернативой, которая может быть использована без внедрения OLAP-технологии.

### 3. Пример из практики

Рассмотрим задачу построения формы ВПО 3.1 «Распределение численности основного персонала по уровню образования» из раздела, посвященного сведениям о персонале. Необходимо заполнить таблицу, в строках которой имеется разбивка по должностям профессорско-преподавательского состава (ППС): ассистент, старший преподаватель, доцент, профессор, заведующий кафедрой, а в столбцах указывается:

- общее количество сотрудников указанной должности;
- количество сотрудников (указанной должности), имеющих ученую степень кандидата наук;
- количество сотрудников, имеющих ученую степень доктора наук;
- количество сотрудников, имеющих ученое звание доцента;
- количество сотрудников, имеющих ученое звание профессора;
- количество сотрудников, работающих на ставку в диапазоне от 0,1 до 1,0 с шагом 0,1.

Описание формы упрощено, чтобы сделать пример более наглядным. Полное описание ВПО 3.1 можно найти в [2]. Для построения такого сводного отчета в ИС должны содержаться фамилия, имя и отчество сотрудника; ученая степень; ученое звание; должность; размер занимаемой ставки. Соответствующая таблица в БД создается следующей командой:

```
CREATE TABLE [Test].[dbo].[Person] (  
  [ID] int IDENTITY(1,1) NOT NULL  
  -- фамилия, имя, отчество сотрудника  
  , [FIO] nvarchar(50) NOT NULL  
  -- Ученая степень (0 – нет, 1 – к. н., 2 – д. н.)  
  , [AcademicDegree] int NOT NULL  
  -- Ученая степень (0 – нет, 1 – доц., 2 – проф.)  
  , [AcademicRank] int NOT NULL  
  -- Должность (1 – ассистент, 2 – старший преподаватель, 3 – доцент, 4 –  
  профессор, 5 – зав. кафедрой)  
  , [Position] int NOT NULL
```

```
-- Ставка
, [Rate] decimal(2,1) NOT NULL)
```

Для генерации тестовых данных использовалось процентное соотношение реальных значений по НИУ МЭИ, представленное в табл. 1, за исключением размера занимаемой ставки. Эта величина определялась с помощью дискретного равномерного распределения. Реализация алгоритма генерации исходных данных находится в открытом доступе [16]. Она применялась для получения исходных данных различного размера.

Таблица 1. Процентное соотношение сотрудников ППС

Должность	Процент сотрудников ППС	Процент сотрудников, имеющих ученую степень		Процент сотрудников, имеющих ученое звание	
		доктор наук	кандидат наук	профессор	доцент
Зав. каф.	3	50	43	41	37
Профессор	21	78	21	61	24
Доцент	44	1	91	0	49
Ст. преп.	20	0	22	0	1
Ассистент	12	0	10	0	0

#### 4. «Стандартный» запрос SQL

Стандартный для реляционных БД и наименее трудоемкий для программиста способ решения указанной задачи — это выполнение следующего запроса с использованием оператора group by:

```
select case p.Position
  when '5' then 'Зав. каф.'
  when '4' then 'Профессор'
  when '3' then 'Доцент'
  when '2' then 'Стар. преп.'
  when '1' then 'Ассистент'
end as 'Должность'
, count(*) as 'Всего'
, sum (case when p.AcademicDegree = 1 then 1 else 0 end) as 'к.н.'
, sum (case when p.AcademicDegree = 2 then 1 else 0 end) as 'д.н.'
, sum (case when p.AcademicRank = 1 then 1 else 0 end) as 'доц.'
, sum (case when p.AcademicRank = 2 then 1 else 0 end) as 'проф.'
, sum (case when p.Rate >= 0.1 and p.Rate < 0.2 then 1 else 0 end) as
'0.1'
, sum (case when p.Rate >= 0.2 and p.Rate < 0.3 then 1 else 0 end) as
'0.2'
-- ...
, sum (case when p.Rate >= 0.9 and p.Rate < 1 then 1 else 0 end) as
'0.9'
```

```
, sum (case when p.Rate = 1 then 1 else 0 end) as '1.0'
from [Test].[dbo].[Person] p
group by p.Position
order by p.Position desc
```

Результат на тестовой выборке в 1500 записей (примерное количество сотрудников ППС в НИУ МЭИ) выглядит следующим образом.

Должность	Всего	к.н.	д.н.	доц.	проф.	0.1	0.2	0.9	1.0
Зав. каф.	47	19	26	12	20	6	3	6	3
Профессор	336	87	245	71	209	37	33	29	35
Доцент	635	581	9	301	0	65	55	59	60
Стар. преп.	290	56	0	3	0	34	38	29	23
Ассистент	193	17	0	0	0	17	22	23	13

Рисунок 1. Результат решения на тестовой выборке в 1500 записей

Поставленная задача решена запросом на языке T-SQL. Минус решения заключается в отсутствии возможности оперативно узнать, кто «скрывается» за конкретными числами. Чтобы получить такие сведения, необходимо уйти от группировки (оператор group by) и добавить блок с условием, соответствующим конкретной ячейке (оператор where). Если результирующая таблица содержит множество строк и столбцов, то детализация каждой ячейки не менее трудоемкая задача, чем первоначальная (построение сводного отчета).

## 5. Запрос с детализацией

Для решения обозначенной проблемы предлагается использовать следующий метод, разделяющий задачу на три этапа:

1. Получение выборки со всеми необходимыми характеристиками для классификации записи. Под классификацией будем понимать определение строк и столбцов, показывающих ячейки, к которым относится запись. Например, заведующий кафедрой, д. т. н., профессор, работающий на 1,0 ставку, относится сразу к 4 ячейкам (классам) — на пересечении строки «зав. каф.» и столбцов «всего», «д. н.», «проф.» и «1,0».
2. Перебор всех записей из пункта 1 и определение ячеек, к которым относится запись.
3. Инкрементирование значения в ячейках, определенных в пункте 2.

Для реализации механизма детализации в указанный метод добавляются входные параметры — строка и столбец, определяющие ячейку, которую необходимо детализировать. Действие на этапе 3 заменяется выводом сведений о текущей записи в таблицу детализации. Блок-схема метода представлена на рисунке.

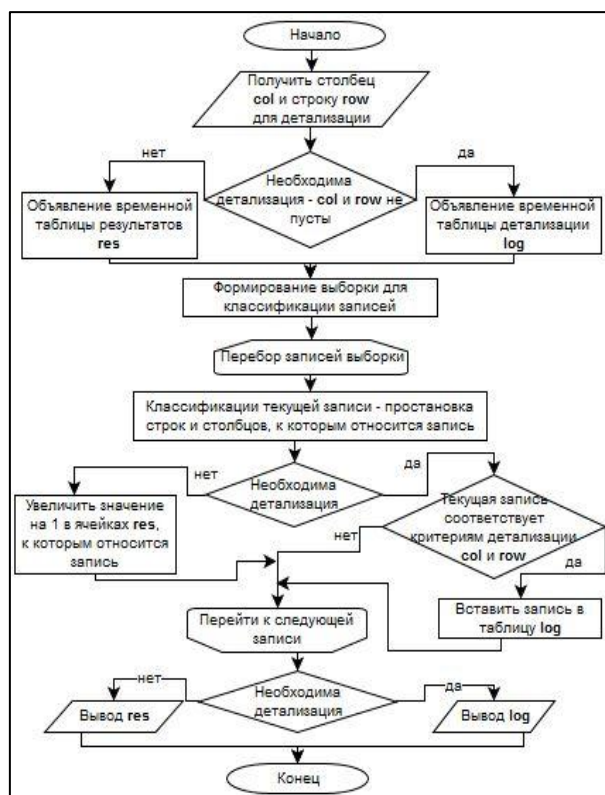


Рисунок 2. Блок-схема метода

Практическая реализация может быть выполнена на различных языках программирования, в том числе и на языке структурированных запросов T-SQL. Программный код для рассматриваемого примера находится в открытом доступе [16]. Рассмотрим основные моменты.

Столбцы таблицы результатов именуются однотипно и содержат номер. Например, c1, c2, c3 и т. д. После объявления в эту таблицу вставляются строки, содержащие номер и отображаемое имя критерия (в данном случае должность). В остальных ячейках должны оказаться нули. Программный код выглядит следующим образом:

```
declare @res TABLE
(c1 nvarchar(5), c2 nvarchar(50), c3 int default 0, ...)
insert into @res (c1, c2)
values ('1', 'Ассистент'), ('2', 'Стар. преп.') ...
```

Таблица детализации включает столбцы, интересные пользователю с точки зрения анализа результатов, например все характеристики, необходимые для классификации. В нашем примере это столбец FIO.

Объявление курсора и перебор данных с его помощью выполняется типовыми командами: `declare cursorName cursor for, fetch next cursorName into, while @@FETCH_STATUS = 0` и др.

Сведения о принадлежности к строкам текущей записи сохраняются в переменной `@targetRowCodes`, к столбцам — в переменной `@targetColCodes`. Каждый раз при переходе к новой записи значения этих переменных очищаются. В представленном листинге происходит классификация записи по должности и наличию ученой степени:

```
-- Добавление строки «должность»
set @targetRowCodes = @targetRowCodes
+ cast(@position as nvarchar(10))
-- Добавление столбца «всего»
set @targetColCodes = @targetColCodes + ';3;'
-- Добавление столбца с соответствующей ученой степенью
if (@academicDegree = 1)
    set @targetColCodes = @targetColCodes + ';4;'
else if (@academicDegree = 2)
    set @targetColCodes = @targetColCodes + ';5;'
```

Классификация по остальным критериям выполняется аналогичным образом. При указании номера столбца используется разделитель «;». Если в именовании строк есть пересечения и/или запись может относиться к нескольким строкам, необходимо также использовать разделитель. После определения целевых ячеек в зависимости от необходимости детализации происходит либо обновление таблицы результатов `@res`, либо вставка текущей записи в таблицу детализации `@log`:

```
if @logMode = 0
begin
    -- Инкрементирование значения необходимых ячеек
    update @res
    set c3 = CASE WHEN @targetColCodes like '%;3;%'
    THEN c3 + 1 ELSE c3 END
    , c4 = CASE WHEN @targetColCodes like '%;4;%'
    THEN c4 + 1 ELSE c4 END
    ...
    , c17 = CASE WHEN @targetColCodes like '%;17;%'
    THEN c17 + 1 ELSE c17 END
    where @targetRowCodes like '%' + c1 + '%'
end
else if (@targetColCodes like '%' + @logColName + ';%' )
    and (@targetRowCodes like '%' + @logRowName + '%')
begin
    -- Вставка текущей записи в таблицу детализации
    insert into @log(FIO) values (@fio)
end
```



В завершение программного кода находятся команды, отвечающие за вывод данных из нужной таблицы — @res или @log.

## 6. Сравнительный анализ

Очевидно, что запрос, реализованный с помощью описанного подхода, выполняется дольше, чем запрос с оператором группировки group by. Основными причинами являются использование операций с временными таблицами, применение оператора like и обход выборки с помощью курсора. Данный факт стоит отнести к недостаткам решения.

Для определения целесообразности применения предлагаемого метода выделим следующие его преимущества.

1. Некоторые запросы с категоризацией по строкам достаточно сложно реализуются с помощью операторов group by и при этом усложняются блоки подсчета итоговой суммы. Предлагаемый метод легко адаптируется до случая, когда одна запись может попадать в несколько строк. Этого требует, например, базовый вариант рассмотренного отчета ВПО 3.1.1, в котором считаются промежуточные суммы по отдельным группам сотрудников (руководящий состав, ППС, научные работники, работники-иностранцы и др.) с дальнейшим разбиением. Кроме этого, менее опытным программистам легче реализовать сводку при наличии «каркаса», декомпозирующего исходную задачу.

2. Единую программную реализацию сводного отчета и детализации отдельных ячеек легче поддерживать разработчикам. Полученный программный код оформляется в хранимую процедуру MS SQL, для которой реализуется пользовательский интерфейс посредством технологии Microsoft Reporting Services. При этом, как сводный отчет, так и отчет-детализация используют единую хранимую процедуру. Трудоемкость внедрения указанного метода значительно ниже по сравнению с внедрением OLAP-технологии.

3. Наконец, сама возможность детализации позволяет быстрее найти и исправить возможные ошибки как при разработке отчета, так и позднее, при построении отчета пользователями.

Все эти плюсы позволяют принять в рассмотрение предлагаемый подход для решения задачи получения сводных данных. Отметим, что построение сводного отчета не является задачей, требуемой моментального решения. Она может выполняться на отдельном сервере и занимать определенное время. Конечно, не стоит забывать о требуемом пользователю быстродействии. Рассмотрим необходимое время для решения описанной выше задачи на различных размерах исходных данных (табл. 2). Запуск производился на машине со следующими характеристиками:

Intel Xeon CPU X5675 3.07GHz, 8GB RAM, Windows Server 2012 R2 x64, SQL Server 2012 SP 4.

Таблица 2. Время выполнения запросов в зависимости от размера исходных данных

Размер выборки	Выполнение запроса с оператором <i>group by</i> , миллисекунд	Выполнение процедуры с возможностью детализации, миллисекунд	
		сводный отчет	детализация ячейки
1500	40	213	76
3000	43	326	123
6000	46	633	233
10 000	56	1110	376
20 000	76	2210	853
50 000	170	5213	1850
100 000	243	10730	3680
200 000	366	21273	8183
500 000	433	54766	19066
1 000 000	826	105533	39956

Из табл. 2 хорошо заметно «отставание» процедуры от запроса с оператором *group by*, но также видно, что время выполнения линейно зависит от размера входных данных. Поэтому в качестве рекомендации при выборе метода решения предлагается оценить размер реальных входных данных (после фильтрации в блоке *where*), скорость увеличения этого размера и готовность пользователя «пождать». Согласно тесту FASMI (Fast Analysis of Shared Multidimensional Information), являющемуся универсальным критерием определения OLAP решения, время выполнения запроса должно измеряться секундами и обычно не занимать более 20 секунд [17]. В рамках задач НИУ МЭИ по учету контингента размер входных данных редко бывает выше 30 тысяч записей, что позволяет успешно использовать описанный метод для большого количества задач. Отметим, что детализация отдельной ячейки выполняется быстрее, чем построение всей сводки.

## 7. Заключение

Предлагаемый метод и его модификации использованы в более 30 отчетных формах, реализованных в информационных системах НИУ МЭИ различного направления, и хорошо зарекомендовал себя в решении отдельного класса задач. Приведенный опыт может быть тиражирован для других программных систем, в которых необходима аналитика и нет возможности внедрения OLAP-технологий. Решение может быть актуальным для других образовательных учреждений, основная сфера деятельности которых не связана с информационными технологиями и ресурсы на их освоение зачастую ограничены [18, 19].

В качестве перспективных направлений развития можно выделить следующие: во-первых, динамическая генерация программного кода по критериям, описанным пользователю на понятном ему языке; во-вторых, исследование механизмов выполнения агрегирующих запросов средствами СУБД и адаптация их для получения возможности оперативно детализировать нужные позиции.

Программный код, включающий создание тестовой таблицы, наполнение ее данными заданного объема, запрос с оператором group by и процедуру для построения сводки с детализацией, доступен для загрузки [16].

## Литература

- [1] Шиняева Т. С. Опыт разработки информационной системы «Электронное студенческое портфолио» // *Cloud of Science*. 2017. Т. 4. № 3. С. 492–511.
- [2] Приказ Росстата от 15.08.2017 № 535. [http://www.gks.ru/bgd/free/b16\\_27/Main.htm](http://www.gks.ru/bgd/free/b16_27/Main.htm)
- [3] Крепков И. М., Овсянникова М. Р., Петров С. А., Федоров А. Б. Реализация кадрового комплекса — компоненты единой информационной среды НИУ МЭИ // *Открытое образование*. 2016. № 3. С. 55–60. DOI: 10.21686/1818-4243-2016-3-55-60
- [4] Петров С. А., Овсянникова М. Р., Федоров А. Б. Кадровый комплекс корпоративной информационной среды НИУ «МЭИ» // Новые информационные технологии в образовании (НИТО). Материалы 9-й МНПК. — Екатеринбург : ФГАОУ ВО «РГППУ», 2016. С. 177–181.
- [5] Миронов А. А., Мордвинов В. А., Скуратов А. К. Семантико-энтропийное управление OLAP и модели интеграции xOLAP в SemanticNET (ONTONET) // *Информатизация образования и науки*. 2009. № 2. С. 21–30.
- [6] Фаминская М. В., Балтер Б. М., Балтер Д. Б., Стальная М. В. OLAP-система для моделирования риска здоровью населения от загрязнения воздуха // *Программные продукты и системы*. 2014. № 4 (108). С. 234–241. DOI: 10.15827/0236-235X.108.234-241.
- [7] Мкртычев С. В. Автоматизированная система управления эффективностью работы страховых агентов // *Современные проблемы науки и образования*. 2014. № 6. <http://www.science-education.ru/ru/article/view?id=15365>
- [8] Григорьев Ю. А., Зеленков С. Ю. Сравнение технологий хранения данных в OLAP-системах // *Информатика и системы управления*. 2001. № 1. С. 3–12.
- [9] Дорожкин А. К. Оценка объемов многомерного куба в OLAP-системах // *Научно-технический вестник информационных технологий, механики и оптики*. 2005. № 19. С. 94–104.
- [10] Овсянникова Л. Ю. Применение технологии OLAP-кубов для анализа данных педагогического мониторинга // *Известия Российского государственного педагогического университета им. А. И. Герцена*. 2014. № 171. С. 271–279.

- [11] *Тимова Г. С.* Создание информационно-аналитической системы в ВУЗе на основе OLAP-технологии // *Современные проблемы науки и образования*. 2012. № 1. С. 148–152.
- [12] *Горнева Е. А., Копелиович Д. И.* Применение технологии OLAP в системе менеджмента качества образования // *Фундаментальные и прикладные исследования: проблемы и результаты*. 2014. № 12. С. 25–34.
- [13] *Ляхов В.* Мифы и реальность OLAP-систем // *Директор информационной службы*. 2008. № 5. <https://www.osp.ru/cio/2008/05/4948702/>
- [14] *Ожегов А. Н., Петропавловский К. М., Головкин А. А.* Применение технологии OLAP при анализе показателей надежности кабельных линий электропередачи // *Фундаментальные исследования*. 2015. № 12 (Ч. 1). С. 74–78.
- [15] Недостатки OLAP-системы оперативной аналитической обработки данных [Электронный ресурс]. URL: <http://www.itstan.ru/it-i-is/nedostatki-olap.html>
- [16] Программный код рассмотренного примера на GitHub [Электронный ресурс] URL: <https://github.com/PetrovSerega/SQLSample/blob/master/Main.sql>
- [17] *Pendse, Nigel*, What is OLAP? [Электронный ресурс] URL: <http://dssresources.com/subscriber/password/papers/features/pendse04072002.htm>
- [18] *Миронов В. В., Латыпов С. Н.* Всеобщая информатизация образования: внутри и вне процесса // *Cloud of Science*. 2017. Т. 4. № 2. С. 282–300.
- [19] *Петрова А. С., Афанасьева Ю. В., Левкина Н. Н.* Информатизация образования: проблемы и перспективы // *Интерактивная наука*. 2017. № 11(21). С. 39–41.

**Автор:**

*Сергей Андреевич Петров* — кандидат технических наук, ведущий программист Отдела разработки и внедрения информационных систем Информационно-вычислительного центра; доцент кафедры безопасности и информационных технологий, Национальный исследовательский университет «МЭИ»

---

## The Method of Implementation of Statistical Reports with Support for Detail

*S. A. Petrov*

*National Research University Moscow Power Engineering Institute  
14, Krasnokazarmennaya st., Moscow, 111250  
e-mail: PetrovSA@mpei.ru*

*Abstract.* The article presents one of the options for implementing a typical task for information systems – the construction of a composite statistical report. The aspects of application of OLAP technology are considered. An alternative approach to the implementation of summary reports is proposed that does not require the introduction of an OLAP system. A practical example of the development compares command in the structured query language

(T-SQL) and the proposed procedure that allows obtaining both summary data and detailing of each individual report cell. The block diagram of the procedure and the code snippet are presented. Based on approbation in the information systems of MPEI and presented test results on samples of various sizes, the pros and cons of the approach are highlighted, recommendations for use and perspective directions of development are given..

*Key words:* information system, statistics, report, summary, detail, structured query.

## References

- [1] *Shinyayeva T. S.* (2017) *Cloud of Science*. 4(3):492–511.
- [2] [http://www.gks.ru/bgd/free/b16\\_27/Main.htm](http://www.gks.ru/bgd/free/b16_27/Main.htm)
- [3] *Krepkov I. M., Ovsyannikova M. R., Petrov S. A., Fedorov A. B.* (2016) *Otkrytoye obrazovaniye*. 3:55–60. DOI: 10.21686/1818-4243-2016-3-55-60.
- [4] *Petrov S. A., Ovsyannikova M. R., Fedorov A. B.* (2016) Kadrovyy kompleks korporativnoy informatsionnoy sredey NIU «MEI». Novyye informatsionnyye tekhnologii v obrazovanii (NITO). Materialy 9-y MNPk, 2016. Yekaterinburg, FGAOU VO «RGPPU». P. 177–181.
- [5] *Mironov A. A., Mordvinov V. A., Skuratov A. K.* (2009) *Informatizatsiya obrazovaniya i nauki*. 2:21–30.
- [6] *Faminskaya M. V., Balter B. M., Balter D. B., Stal'naya M. V.* (2014) *Programmnyye produkty i sistemy*. 4(108):234–241. DOI: 10.15827/0236-235X.108.234-241.
- [7] *Mkrtychev S. V.* (2014) *Sovremennyye problemy nauki i obrazovaniya*. 6.
- [8] *Grigor'yev Yu. A., Zelenkov S. Yu.* (2001) *Informatika i sistemy upravleniya*. 1:3–12.
- [9] *Dorozhkin A. K.* (2005) *Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki*. 19:94–104.
- [10] *Ovsyanitskaya L. Yu.* (2014) *Izvestiya Rossiyskogo gosudarstvennogo pedagogicheskogo universiteta im. A. I. Gertsena*. 171:271–279.
- [11] *Titova G. S.* (2012) *Sovremennyye problemy nauki i obrazovaniya*. 1:148–152.
- [12] *Gorneva Ye. A., Kopeliovich D. I.* (2014) *Fundamental'nyye i prikladnyye issledovaniya: problemy i rezul'taty*. 12:25–34.
- [13] <https://www.osp.ru/cio/2008/05/4948702/>
- [14] *Ozhegov A. N., Petropavlovskiy K. M., Golovko A. A.* (2015) *Fundamental'nyye issledovaniya*. 12-1:74–78.
- [15] <http://www.itstan.ru/it-i-is/nedostatki-olap.html>
- [16] <https://github.com/PetrovSerega/SQLSample/blob/master/Main.sql>
- [17] <http://dssresources.com/subscriber/password/papers/features/pendse04072002.htm>
- [18] *Mironov V. V., Latypov S. N.* (2017) *Cloud of Science*. 4(2):282–300.
- [19] *Petrova A. S., Afanas'yeva Yu. V., Levkina N. N.* (2017) *Interaktivnaya nauka*. 11(21):39–41. DOI 10.21661/r-465136