

# ОБ ОДНОМ СПОСОБЕ РЕАЛИЗАЦИИ ПРОЦЕДУР ВЫБОРКИ ДАННЫХ

Олейник П.П.

E-mail: [xsl@list.ru](mailto:xsl@list.ru)

ООО "Волшебный край", Ростов-на-Дону

**Аннотация.** Сформулированы критерии оптимальности архитектуры процедур выборки данных. Рассмотрена типовая реализация, соответствующая данным критериям.

## Implementing stored procedures for data view

Oleynik P.P.

**Abstract.** Criteria of optimality for the stored procedure architecture are listed. Present typical implementation for data view procedure.

В роли основного элемента современных корпоративных информационных систем выступает сервер СУБД[1]. В настоящее время доминирующей является реляционная модель данных [2]. Использование реляционной СУБД в качестве хранилища информации подразумевает создание хранимых процедур (ХП) на сервере, реализующих бизнес-логику приложения. Унификация структуры и способа передачи параметров значительно облегчает процесс написания, отладки и модификации программного кода.

При разработке архитектуры были сформулированы следующие критерии оптимальности (КО):

1. Унификация представления переданных параметров.
2. Единая структура процедур одного класса.
3. Унификация управления ошибками.
4. Возможность внесения изменений и дополнений в структуру ХП.

На рис. 1 представлена типовая реализация, соответствующая выдвинутым критериям.

Рисунок требует пояснений. Для его построения использовалась UML нотация. Несмотря на то, что UML диаграммы рассчитаны на проектирование объектно-ориентированных приложений, существуют способы применять этот язык для представления зависимостей между объектами (ХП, таблицы, представления) БД [3].

Для спецификации объектов были введены следующие стереотипы (stereotype):

- <<table>> - Таблица БД. Применяется для классов, представляющих собой таблицы БД. В роли атрибутов выступают поля таблицы.
- <<PK>> - Первичный ключ. Применяется для атрибутов класса, представляющего собой таблицу БД (стереотип <<table>>) и помечает поля, входящие в первичный ключ таблицы.
- <<FK>> - Внешний ключ. Применяется для атрибутов класса, представляющего собой таблицу БД (стереотип <<table>>) и помечает поля, которые являются внешним ключом и ссылаются на первичные ключи таблиц.
- <<view>> - Представление (виртуальная таблица) БД. Применяется для классов, являющихся представлениями, сохранёнными в БД. В роли атрибутов выступают поля результирующего набора данных. Типы полей соответствуют типам столбцов исходных таблиц, на основе которых построено представление.
- <<table function>> - Табличная функция (ТФ). Применяется для классов, представляющих собой функции, хранящиеся в БД и возвращающие набор данных в качестве результата.
- <<stored procedure>> - Хранимая процедура (ХП). Применяется для классов, представляющих процедуры, хранящиеся в БД. Результатом выполнения может быть как набор данных, так и значения выходных параметров (output parameters). При этом ведётся логирование ошибок, которые по окончании работы ХП возвращаются пользователю.

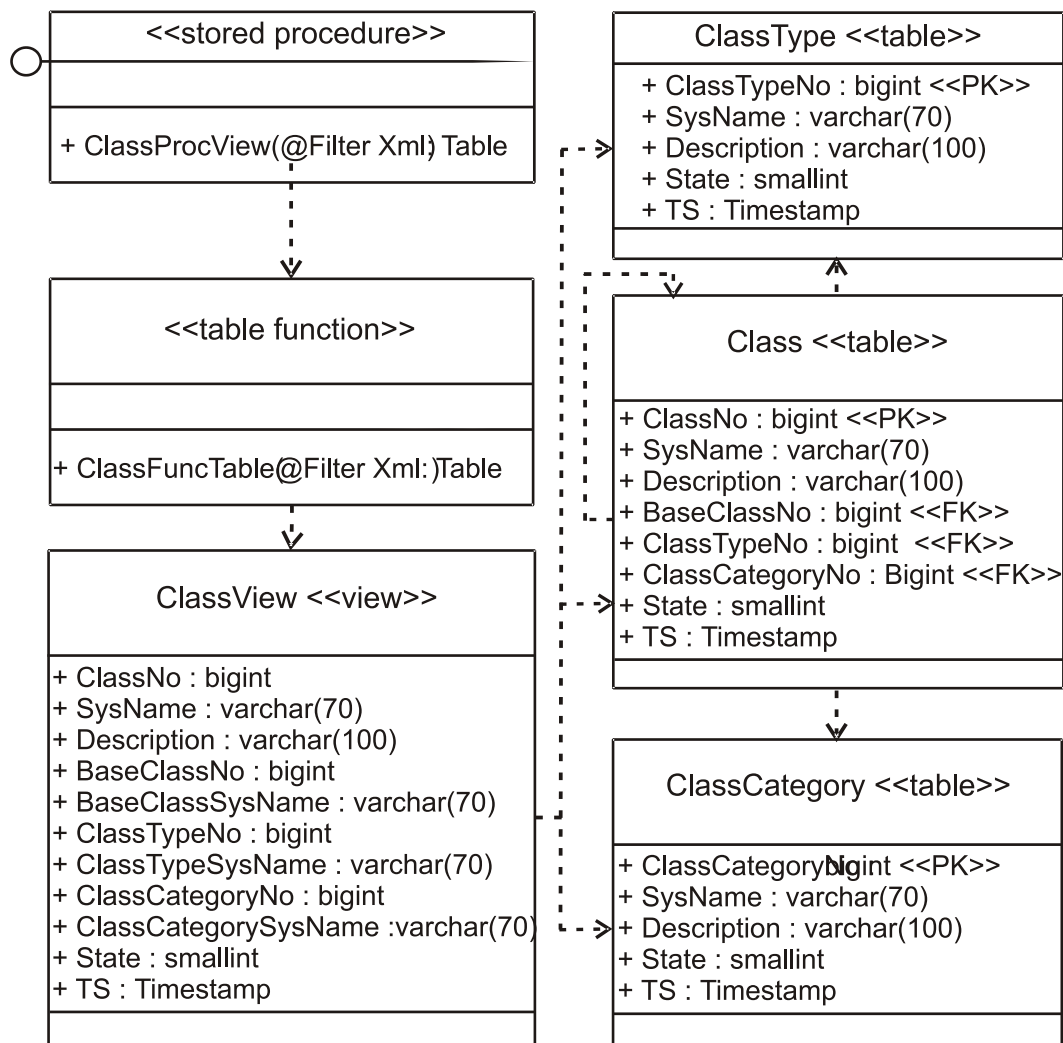


Рис.1. Типовая реализация процедуры выборки данных.

На рис. 1 видно, что обрабатываемые данные физически хранятся в таблицах Class, ClassType, ClassCategory. На основе этих таблиц строится представление ClassView, данные из которого передаются в табличную функцию ClassTableFunc. В функции имеется параметр (@Filter), в котором хранятся условия фильтрации данных. Клиентское приложение непосредственно вызывает ХП ClassProcView, единственным параметром которой является фильтр, сформированный на клиенте.

Отметим некоторые особенности представленной реализации:

1. Программная реализация предложенной архитектуры может быть сгенерирована автоматически. Если при разработке структуры БД была внесена метамодель системы, то генерация программного кода может быть автоматизирована [4-5]. Таким способом достигается унифицированность структуры (КО 2).
2. В представлении, в табличной функции и хранимой процедуре присутствуют текстовые метки, позволяющие прикладному программисту добавить программный код, реализующий бизнес-логику приложения. Таким образом реализован КО 4.
3. Для передачи параметров в ХП и ТФ используется XML документ. Если учесть, что во многих современных СУБД введён тип данных xml, то такая реализация является для них оптимальной (КО 1).
4. Все ошибки, возникающие в ходе выполнения, а также сообщения о нарушении бизнес-правил приложения записываются в выходной параметр (@Filter). В результате значение этого параметра передается на клиентское приложение (в случае реализации архитектуры клиент-сервер) или на сервер приложений (при реализации трехзвенной архитектуры). Таким способом реализован КО 3.

В статье предложена оптимальная архитектура процедур выборки данных. Рассмотрены основные структурные элементы, зависимости объектов (Рис. 1), а также особенности программной реализации.

## Литература

1. Флауер М., Архитектура корпоративных программных приложений, Пер. с англ. – М.: Издательский дом “Вильямс”, 2004. – 544 с.: ил. – Парал. тит. англ.
2. Коннолли Т., Бегг К., Страчан А., Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд. – М: Издательский дом “Вильямс”, 2000. – 1120 с.: ил. – Парал. тит. англ.
3. Ambler S., Persistence Modeling in the UML, <http://www.ddj.com/dept/architect/184415725?cid=Ambysoft>
4. Олейник П.П., Представление метамодели объектной системы в реляционной базе данных // Известия высших учебных заведений. Северо-Кавказский регион. Спецвыпуск “Математическое моделирование и компьютерные технологии”, 2005.
5. Олейник П.П., Организация метамодели объектной системы на основе реляционной СУБД // Научное творчество молодёжи: Материалы X Всероссийской научно-практической конференции (21-22 апреля 2006 г.) Ч.1. – Томск: Изд-во Том. ун-та, 2006.