

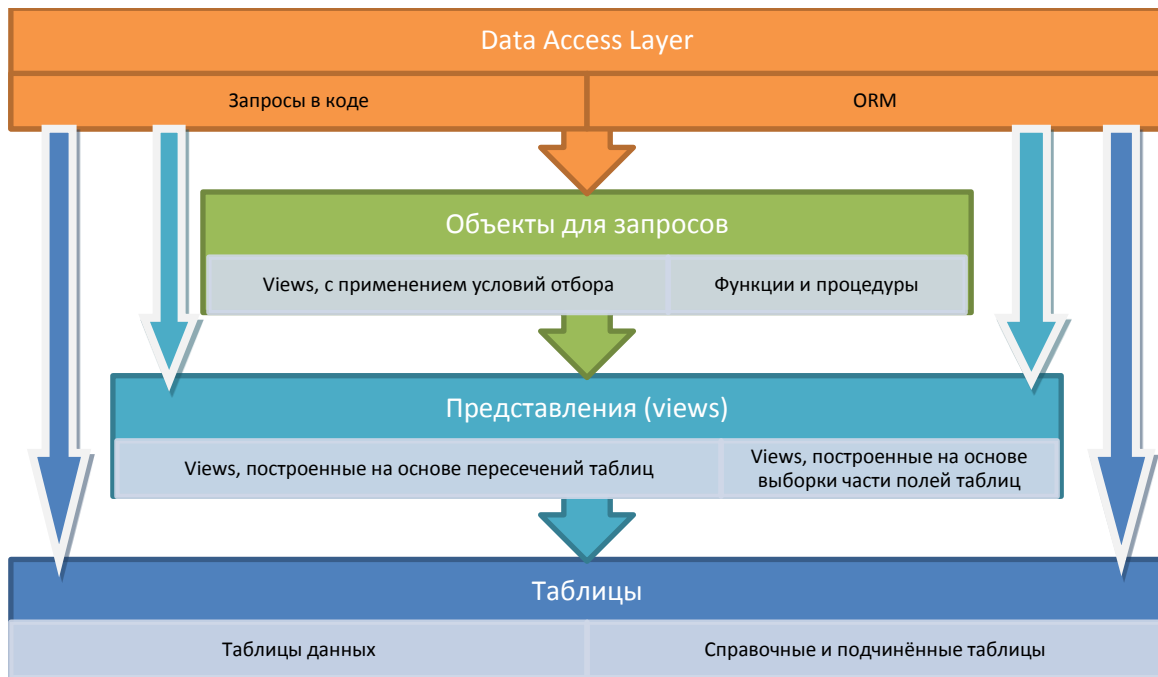
Технология работы с БД

Правила наименования объектов

Тип объекта	Правило наименования	Пример
Таблица	Имя сущности в единственном числе	Subject
Таблица справочника с возможностью редактирования со стороны пользователя	D_ИмяСправочника	D_SubjectType
Таблица справочник для системных нужд	DS_ИмяСправочник	DS_OrderStatus
Имя поля первичного ключа	ИмяТаблицыId	SubjectId
Вью	V_ИмяСущностиList	V_OrganizationList
Хранимая процедура, выполняющая фундаментальное действие (INSERT, UPDATE, DELETE)	P_ИмяСущности_INS P_ИмяСущности_UPD P_ИмяСущности_DEL	P_Order_INS P_Service_UPD P_Subject_DEL
Триггер последствия (after)	TA_ИмяТаблицы_INS TA_ИмяТаблицы_UPD TA_ИмяТаблицы_DEL	TA_Subject_INS TA_AddressTree_UPD TA_Order_DEL
Триггер преддействия (before), замещающий триггер	TB_ИмяТаблицы_INS TB_ИмяТаблицы_UPD TB_ИмяТаблицы_DEL	TB_Subject_INS TB_AddressTree_UPD TB_Order_DEL
Функция, возвращающая значение	F_ИмяЗначения	F_OrderPrice
Функция, принимающая на вход значение идентификатора	F_ИмяФункцииВуИмяСущности	F_ActiveOrdersCountBySubject

Технология запросов данных БД

В большинстве случаев, данные, хранящиеся в таблицах БД, не являются теми данными, которые запрашиваются клиентом (приложением). Зачастую, требуется некое логическое объединение данных нескольких таблиц, возможно, с применением каких-то условий на выборку. Обобщённую схему организации объектов БД и запросов к ним со стороны клиента можно изобразить следующим образом:



Обобщённая схема организации объектов БД и доступа к ним

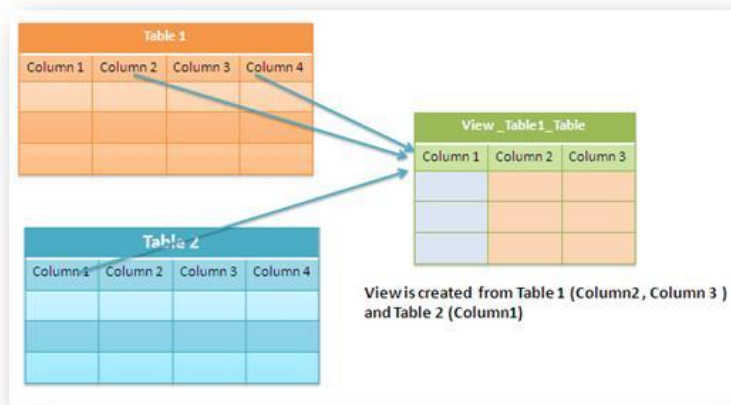
Прямой доступ к таблицам

Схематично, прямой доступ к таблицам обозначен **стрелками тёмно-синего цвета**. Прямой доступ обеспечивается в следующих случаях:

- Запрос данных таблицы только на чтение
- Запрос данных из справочных таблиц
- Запрос данных из таблиц связи многие ко многим

Преимуществом данного подхода является гибкость и управляемость с точки зрения разработчика клиента – им может быть реализован любой запрос, при этом на стороне БД ничего менять не надо. Платой за это (недостатком) является снижение безопасности (например, риск sql инъекций) и неконтролируемая несогласованность (при изменении схемы БД, код клиента становится несогласованным, но компилируемым). Для повышения управляемости, снижения рисков нарушения безопасности и несогласованности могут применяться средства ORM.

Доступ к представлениям (views)



Представления, построенные на основе пересечения таблиц или выборки части полей, предназначены для создания полноценных (с точки зрения характеристик объекта) списков (коллекций) объектов. В тех случаях, когда поле главной таблицы связано с полем подчиненной таблицы, для более удобного и быстрого просмотра данных об объектах одного типа (коллекции объекта), требуется получить строковое (в большинстве случаев) значение подчиненной таблицы. Классический пример – список сотрудников с указанием наименования должности и отдела, в котором работает сотрудник – наименование должности и отдела хранятся в справочных таблицах и связаны с таблицей первичных данных о сотрудниках по ключу. Значение ключа не информативно, поэтому уместно использовать выборку самих строковых значений наряду, например, с ФИО сотрудника. Представления, созданные в рамках приведенной методики могут использоваться в объектах более высокого уровня: представления, с применением условий отбора, в процедурах и функциях.

Объекты для запросов

Объекты для запросов – это объекты самого высокого уровня, которые могут строиться на основе объектов более низких уровней, и предназначены для точного выбора необходимых данных. Основные примеры использования таких объектов:

- Выбор данных с применением условий отбора
- Выбор данных с вычислением значений некоторых столбцов
- Горизонтальное объединение однородных списков
- Реализация не реляционных методов работы с данными.

Сценарии разработки

В данном разделе описываются основные сценарии разработки в области баз данных в рамках типовых задач. Эти сценарии носят рекомендательный характер.

Групповое изменение данных

Источником возникновения задачи группового изменения данных может быть бизнес-процесс, требование пользователя, обнаруженная ошибка в структуре данных или в самих данных.

Сценарий группового изменения данных состоит из следующих этапов:

1. Анализ текущей структуры данных
2. Создание тестового набора данных
3. Построение плана изменения данных
4. Построение плана обработки ограничений
5. Написание скрипта изменения
6. Написание теста

7. Выполнение скрипта изменения и теста в рамках одной транзакции на тестовых данных с откатом транзакции
8. Создание резервной копии рабочей базы данных
9. Выполнение скрипта изменения и теста в рамках одной транзакции на рабочих данных с откатом транзакции
10. Выполнение скрипта изменения и теста в рамках одной транзакции на рабочих данных с подтверждением транзакции

Анализ текущей структуры данных

Групповое изменение данных может быть нацелено на изменение данных одной или нескольких таблиц. В ходе анализа необходимо выявить набор таблиц, подлежащих изменению, зависимых таблиц и природу (тип) зависимости, ограничения, налагаемые на таблицы (ключи и др.), а также наличие связанных объектов, такие как триггеры, хранимые процедуры и функции. Желательно построить схему той области базы данных, которая содержит все выявленные объекты.

Создание тестового набора данных

Для создания скрипта группового изменения данных необходимо использовать тестовый набор данных, максимально приближенный к рабочему набору в области изменения и в области зависимых объектов, полученных в ходе [анализа](#).

Тестовый набор данных может быть создан при помощи генератора данных, создание и использование которого описано в разделе [«Создание и запуск генераторов данных»](#).

Построение плана изменения данных

План изменения данных – это свободное текстовое описание действий, которые должны быть предприняты для группового изменения данных без учета ограничений. В нём описываются условия отбора данных, подлежащих изменению, сами изменения и ожидаемый результат. В дальнейшем, план сохраняется как комментарий к скрипту, производящего изменения.

Построение плана обработки ограничений

План обработки ограничений – это свободное текстовое описание действий, которые должны быть предприняты для отключения ограничений на момент изменений и их включения после изменений. Следует отметить, что результат изменений не должен противоречить существующим ограничениям. Отключение и включение ограничений не обязательные шаги, и необходимы для того, чтобы ускорить или упростить операцию изменения. К ограничениям относятся не только ключи и другие стандартные изменения, но и некоторые триггеры, в которых реализованы механизмы поддержания ссылочной целостности. Такие триггеры так же стоит рассматривать как объекты ограничения и отключать на момент изменений, если это необходимо. Аналогично плану изменения данных, план обработки сохраняется как комментарий к скрипту, производящего изменения. Возможно комбинирование планов для установления единой последовательности действий.

Написание скрипта изменения

Скрипт изменения оформляется как обычный T-SQL скрипт. В некоторых случаях, для выполнения требуемого действия проще использовать сочетание хранимых процедур и функций в составе сложной операции группового изменения. В скрипте могут быть использованы DDL скрипты создания всех необходимых процедур и функций, но в целях безопасности следует также описать их удаление после использования. Если используемые процедуры и функции не противоречат требованиям безопасности, то их стоит оформлять как отдельные объекты БД и закреплять в проекте БД.

Скрипт содержит все операции отключения ограничений, выполнение изменений и включение ограничений, а так же планы в виде комментариев.

Написание теста

Тест предназначен для фиксирования результата изменений и сопоставляет их с ожиданием. Кроме того, могут быть протестированы зависимые таблицы, на предмет ожидаемого изменения или ожидаемого отсутствия изменения.

Выполнение скрипта изменения и теста в рамках одной транзакции на тестовых данных с откатом транзакции

Написанные скрипт изменения и тест выполняются в рамках единой транзакции с откатом. Единственный результат, который фиксируется в рамках такого выполнения – это положительное исполнение теста.

Создание резервной копии рабочей базы данных

Перед выполнением группового изменения создается резервная копия рабочей базы данных. В случае обнаружения серьезных нарушений в данных, база данных восстанавливается из созданной резервной копии.

Выполнение скрипта изменения и теста в рамках одной транзакции на рабочих данных с откатом транзакции

Аналогично пункту [«Выполнение скрипта изменения и теста в рамках одной транзакции на тестовых данных с откатом транзакции»](#) с тем лишь изменением, что операции выполняются на рабочих данных.

Выполнение скрипта изменения и теста в рамках одной транзакции на рабочих данных с подтверждением транзакции

В случае положительного результата выполнения теста в рамках пункта [«Выполнение скрипта изменения и теста в рамках одной транзакции на рабочих данных с откатом транзакции»](#), эти же действия выполняются с подтверждением транзакции.