**ЛЕКЦИЯ 1. ПРОЕКТИРОВАНИЕ**

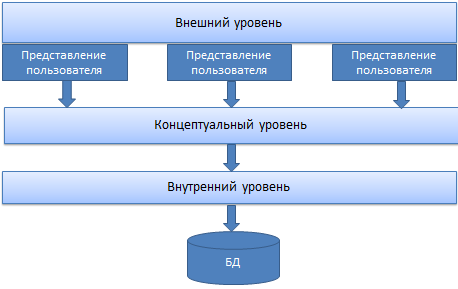
**Требования к данным:**  
\* многократное использование  
\* простота и легкость исп-я  
\* гибкость исп-я  
\* быстрая обработка запросов на д-е

**Проектирование БД:**

1. **концептуальное** : цель – созд. концепт. модель д-х исходя из представлений юзеров о предметной области  
 1) опред. сущностей + их документирование  
 2) опред. связей м/сущностями и их документир  
 3) создание ER-модели предметной области  
 4) опр. атрибутов и их документирование  
 - имя атриб, его описание  
 - тип и размерность значений  
 - значение, приним. для атрибута по умолч  
 - может ли атр иметь Null-знач  
 - явл ли атрибут составным  
 5) опр. значений атрибутов и их документир  
 6) опр. PK для сущностей и их докум.  
 7) обсуждение концепт. модели д-х с конеч. юзерами

2. **логическое** : цель – преобр. концепт. модели на основе выбранной модели д-х в лог. модель, не зависимую от особенностей исп-мой в дальнейшем СУБД для физ. реализации БД  
 1) выбор модели д-х  
 2) опр. набора таблиц исходя из ER-модели и их докум.  
 3) нормализация таблиц  
 4) проверка лог. модели д-х на предмет возможности вып-я всех транзакций юзеров  
 5) опр. требований поддержки целостности д-х и их докум.  
 - обяз. д-е  
 - ограничений для значений атрибутов  
 - целостность сущностей  
 - ссылочная целостность  
 - ограничения, наклад. бизнес-правилами  
 6) созд. оконч. варианта лог. модели д-х и обсуждение его с юзерами

3. **физическое** : цель – описание конкр. реализации БД  
 1) проектирование таблиц БД ср-вами выбранной СУБД  
 2) реализация бизнес-правил в среде выбранной СУБД  
 3) проектирование физ. организации БД  
 4) разработка стратегии защиты БД  
 5) организация мониторинга функц-я БД и ее настройка

**Модель организации баз данных:**  


**Модели данных:**\* сетевая  
\* иерархическая  
\* реляционная  
\* NoSQL-модели

**Реляционная модель:**

***Рел. мод.***– представление БД в виде сов-сти упорядоченных нормализованных отношений

**12 правил Кодда** – с-ма правил, к-й д. удовл каждая СУ(реляц.)БД

**Декартово произведение:** для заданных конечных множеств D1…Dn декарт. произв-ем D1\*…\*Dn наз. мн-во произведений вида d1\*…\*dn, где d1 э D1

**Отношение:** отнош. R, опр-ным на мн-вах D1…Dn наз. подмн-во декартова произв-я D1\*…\*Dn:  
- мн-ва D1… наз. **атрибутами** отн-я  
- эл-ты мн-в D1… наз. **доменами** отн-я  
- число n наз. **степенью** отн-я  
- эл-ты декартова произв-я d1\*…\*dn наз. **кортежами**  
- кол-во кортежей наз **мощностью** отн-я

***Схема отношения*** – именованное мн-во пар имя атр – имя домена  
***Схема БД*** – набор именованных схем отношений  
***Ключ*** – атрибут, значение к-го однозначно идентифиц. кортежи (1 из них ***первичный***)  
***Возм. ключи*** – все остальные ключи отношения

**Теория нормальных форм:**

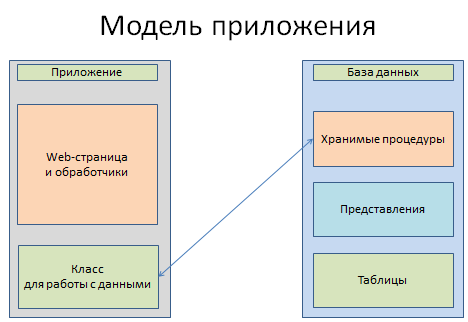
если даны два атриб X, Y нек. отн-я, то говорят, что Y **функционально зависит** от Х, если в любой момент t каждому значению Х соотв. ровно одно значение Y

**избыточная функц. зависимость** – завис., заключ. в себе такую инфу, кот. мб получена на основе др. завис-стей, имеющихся в БД

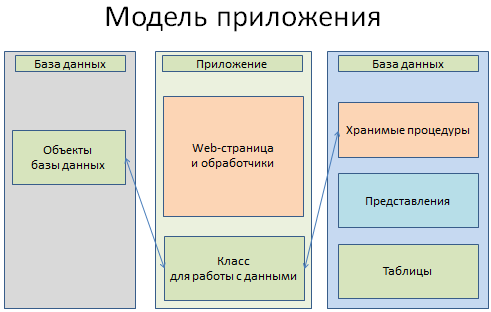
**нормализация** – (6+2) обратимый пошаговый процесс замены данной сов-сти отн-й другой схемой с устранением избыт. функц. завис-стей, причем:  
 \* не д. появляться ранее отсутс. кортежи  
 \* на отн-ях новой схемы дб вып исх. мн-во функц. завис-стей

**простой атрибут** – атр, значения к-го неделимы  
**сложный атр** – получен соед-ем неск. атомар. атр-тов, кот. мб определены на 1/разных доменах

**денормализация** – намеренное приведение структ. БД в сост, не соотв. критериям норм-ции, проводимое с целью ускорения оп-ций чтения из бд за счет добавл избыт. д-х

**ДОСТУП К ДАННЫМ ИЗ ПРИЛОЖЕНИЯ**

**Модели приложений:** \* прил – бд  
\* прил – неск. бд  
\* SOA

**SOA – Service-Oriented Architecture**  
это модульный подход к разработке ПО, осн. на исп-и распредеел, слабо связанных заменяемых компонентов, оснащенных стандартизированными интерфейсами для взаимод-я по стандартизир. протоколам

**Особенности бд web-приложений:**  
\* пользователи  
\* передача д-х  
\* клиентские у-ва  
\* безопасность

**API для доступа к данным:**  
\* Доступ к д-м – прикладной программный интерфейс для СУБД  
\* Набор ф-й:  
 - установления и закр соед-я  
 - обновление д-х  
 - передача запросов серверу  
 - получ. ре-тов вып-я запросов  
 - получение кодов ошибок  
 - хар-ки структуры набора рез-та

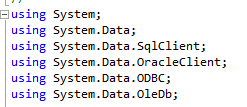
**Универсальный механизм доступа к д-м:**  
\* Универсальных мех. доступа к д-м обычно реализован в виде библиотек и доп. модулей – драйверов или провайдеров  
\* Библиотеки сод. стандартный набор ф-й или классов, подчин. спецификации  
\* Доп. модули реализуют непосред. обращение к ф-ям конкр. СУБД  
\* Легко модиф., если нужна смена СУБД  
\* Измен. только настройки доступа к д-м  
\* Невозм. доступа к уник. ф-ности, специфичной для конкр. СУБД  
\* Снижение произв-сти прил  
\* Усложнение процедуры поставки прил

**Microsoft Data Access Components:**  
\* ODBC – Open Database Connectivity  
\* OLE DB – Object Linking Embedding Database  
\* ADO.NET

**ADO.NET:**  
интерфейс прикладного уровня MS  
- простой доступ к рел. д-м  
- унификация доступа к рел. д-м  
- расширяемость  
- поддержка многоуровневых прил и XML (EF, LINQ to SQL, обработка наборов д-х)

**ADO.NET Режимы работы с данными:**  
**- постоянное подключ**  
 \* устан соедин  
 \* подготовка и вып. к-ды  
 \* работа с д-ми (чт, зап, фильтр, сортир, тоже в пакет. режиме, блокировки)  
 \* закр. соединение и обр ошибок  
 раб. в режиме удерж подкл к бд, макс гибкость и эффек, мин расход ОП

**- отсоединенные д-е**  
 \* загрузка д-х с сервера  
 \* измен д-х в наборе на лок. машине  
 \* обновл д-х на сервере на основе лок. копии  
 раб с д-ми в отсут подключ к БД, удобна для переноса д-х по сети, расход много ОП

**Провайдеры д-х:  
Провайдеры (поставщики) д-х** – извлечение д-х из источника д-х:  
- SQL Server .NET Data Provider (SQL Server)  
- Oracle Data Provider (Oracle Server)  
- ODBC.NET Data Provider (ODBC source – ODBC driver)  
- OleDB.NET Data Provider (OLE DB Source – Data Provider for OLE DB)

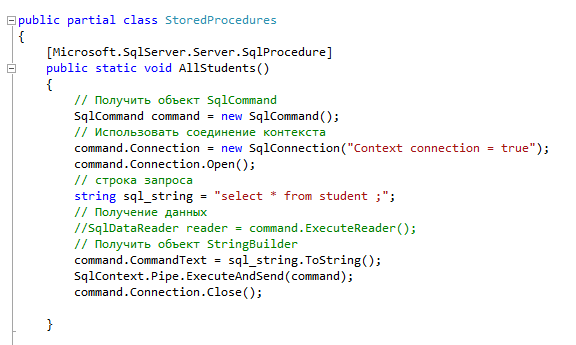
\* обесп большую произв-сть  
\* позв. раб со специф. ТД для данной субд  
\* лучше вып. специф. для данной субд ф-и  
\* м. использ неспециализ. провайдера д-х  
\* м. определить набор провайдеров д-х (модель ProviderBase)

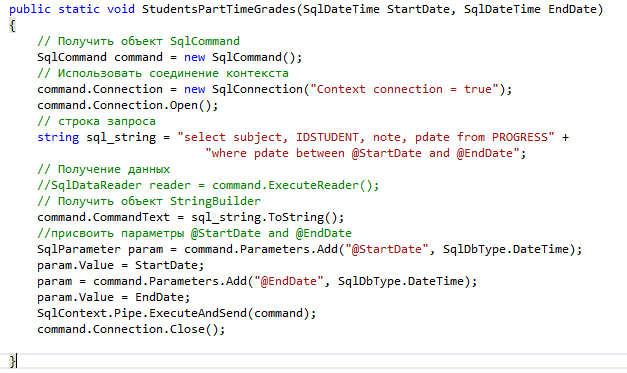
**СБОРКИ CLR**

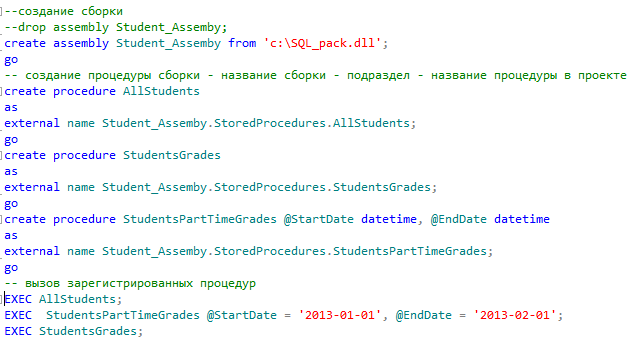
**CLR – Common Language Runtime** – общеязык исполн среда  
\* поддерж языки – C#, VB  
управляемая среда, домены приложений, CLR развернута в SQL Server  
потоковая обраб рез. наборов – по одной строке

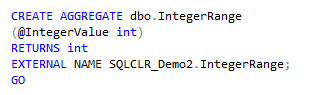
**Интеграция .NET и SQL Server:**  
\* сложная логика  
\* решение матем. задач  
\* подкл к удал. службам  
\* работа с удал. хранилищами д-х  
\* управление внеш файлами

**SQL CLR:**  
процедуры, ф-и (скалярные и табличные), типы, триггеры  
включение: use master; exec sp\_configure ‘clr enabled’, 1; reconfigure

**Создание процедуры:**  


**Добавление параметров:**

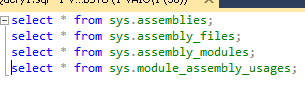
**Регистрация сборки в SQL Server:**

**Создание агрегатной ф-и и типа:**

**SQL User Defined Aggregate:**\* Init – иниц. корректными значениями при отсут строк  
\* Accumulate – вычисление агрегации  
\* Merge – примен. агрегата над агрегатом  
\* Terminate – возвр. конечный рез-т

**SQL User Defined Type:**  
\* INullable : isNull (проверка на null), null (вновь созданный экземпляр тип)  
\* ToString – строк. предст экземпляра типа  
\* Parse – получение экземпляра из строки

**CAS:**  
**Code Access Security** – механизм упр-я безопасности доступа к коду  
\* SAFE – не нужны внешние ресурсы, управляет SQL Server  
\* EXTERNAL\_ACCESS – доступны внеш. ресурсы (файлы, сет. службы, реестр)  
\* UNSAFE – разрешены любые д-вия

**Просмотр сведений о сборках:**  


**ПРАВИЛА КОДДА**

|  |  |
| --- | --- |
| 0) основное | с-ма, кот. позиционируется как реляц. СУБД, дб способна управлять бд, используя только свои рел. возм-сти |
| 1) информационное | вся инфа в рел. БД на лог. уровне дб явно представлена един. способом: значениями в таблицах |
| 2) гарант-й доступ к д-м | в рел. бд каждое отд. (атомарное) значение д-х дб логически доступно с пом. комбинации имени таблицы, значения PK и имени столбца |
| 3) систем. поддержка отсут. значений | неизвест. (или null) знач, отличные от любого извест. значения, д. поддерж для всех ТД при вып. любых операций для числ.д-х неизвестные знач не дб рассм как 0, для симв – как пустые строки |
| 4) доступ к словарю д-х в терминах рел. модели | словарь д-х дб сохр в форме рел.таблиц, и субд д. поддерж доступ к нему при пом. стандартных язык.ср-в, тех же, кот исполз. для работы с рел. таблицами сод. польз. д-е |
| 5) полнота подмн-ва языка | СУрБД д. поддерж хоть 1 рел. язык, кот: а) имеет линейный синтаксис б) мб исп как интерактивно, так и в прикл. прогах в) поддерж операции опр-я д-х…. |
| 6) возм-сть измен предстй | к. предст д. поддерж все оп-ции манипул д-ми, кот. поддерж. рел. таблицы: операции выборки, вставки, измен и удал д-х |
| 7) наличие выс-ур опер. упр. д-ми | оп-ции вставки, изм, удал дб поддерж не только по отн к 1 строке рел. таблицы, но и по отн к любому мн-ву строк |
| 8) физ. независ. д-х | прил не д. зависеть от способов хран д-х на носителях, от аппар.обесп пк, на кот. нах. рел.бд |
| 9) лог. независ д-х | предст д-х в прил не д. зависеть от структ рел.таблиц. Если в процессе нормализации 1 рел.таблица раздел на 2, предст д. обесп объединение этих д-х, чтобы измен структ рел.таблиц не сказывалось на работе приложений |
| 10) независимость контроля цел-сти | вся инфа, необх для поддерж цел-сти, д нах в словаре д-х. Язык для работы с д-ми д. вып проверку вх. д-х и авто- поддерж цел-сть д-х |
| 11) независ от распол | бд мб распределеной, м наз на неск пк, это не должно влиять на приложения |
| 12) согласование язык. уровней | если исп низкоур. зык доступа к д-м, он не д. игнор правила безоп и цел-сти, кот. поддерж языком более выс. уровня |

**НОРМАЛЬНЫЕ ФОРМЫ:**

\* **1NF** – все атрибуты отношения атомарны (ФИО, Отдел, Дети)  
\* **2NF** – 1+к. неключ атрибут функц. полно завис от ключа  
неключ. атрибут **функц. полно зависит** от составн.ключа, если он завис от всего ключа, но не завис. от входящих в него атрибутов (Поставщик, Товар, Цена)  
\* **3NF** – 2+к. неключ. атрибут нетранзитивно завис от PK (Ханение-Склад-Объем)  
X Y Z – атрибуты нек. отношения  
X-->Y, Y-->Z, но обратное соответствие отсут Z-/->Y, Y-/->X  
тогда Z **транзитивно зависит** от X

\* **BCNF (Нормальная форма Бойса-Кодда)** – 3+нет зависимостей атрибутов PK от неключ.атрибутов  
отличие от 3НФ: в 3НФ отношение имеет 2+ возм. ключа, кот. явл составными и имеют общий атрибут (Аренда помещений – Тариф)

\* **4NF** – BCNF+в нем нет многозначных завис-стей, кот. не явл. функц. завис-стями  
многознач.зав : Препод – (Курс) – (Учебник)

\* **5NF** – любая зависимость по соед-ю в нем опр-ся только его возм. ключами (Оператор связи – Производитель – Оборудование)

**\* Доменно-ключевая нормальная форма:** каждое налож. на пер-ную ограничение явл. лог. следствием ограничений доменов и ограничений ключей, налож. на д-ю пер-ную отн-я

огран. домена – огр, предпис. использовать для опр. атрибута значения только из нек. заданного домена  
огран. ключа – огр, утвержд., что нек. атрибут (их комбинация) явл. потенциальным ключом

\* **6NF** – пер-ная удовл всем нетривиальным завис-стям соединения  
неприводима : не мб подвергнута дальнейшей декомпозиции без потерь

**ЛЕКЦИЯ 2. ПРОСТР. Д-Е**

**Протранственные д-е:**  
\* земельный кадастр, агро-мониторинг, задачи землеу-ва, геодезия и недропользование, мониторинг техники, учет объектов недвижимости, построение с-м инженерных коммуникаций

**ГИС (Геоинформационные системы)** – с-мы, предн. для сбора, хран, анализа и граф. визуализации простр. д-х и связанной с ними инфы о представленных в ГИС объектах

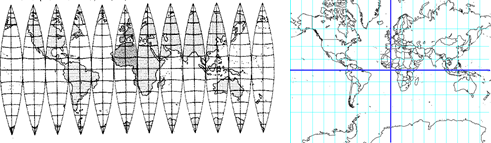
**Источники д-х для ГИС:**   
карты, геодез. д-е точного измерения коорд, пов-сти: воздушные, наземные подземные, водные, космич; аэрокосм. фотосъемка и сканирование, стереосъемка; д-е из арх-строит и инженерно-коммун САПР

**Хранение д-х в ГИС:**

\* **Растровое:**  
 - разби всю карту на эл-ты регулярной сетки  
 - к. ячейка сод. только 1 значение  
 \* *площадный контур* (*зона*) – набор смежных местоположений с один. св-вами  
 \* *класс* (*район*) – все самост. зоны с один. св-вами  
 \* осн. *компонентами* зоны явл. ее значение и местоположение

\* **Векторное:**  
 - базовый примитив – точка  
 - объекты созд. путем соед точек прямыми линиями или дугами

**Атрибутивная инфа** – характ. простр. объекты  
**Послойная организация д-х:** тематические слои, временные срезы, уровни по вертикали

**Проекции (Гаусса-Крюгера и Меркатора (UTM)):**  


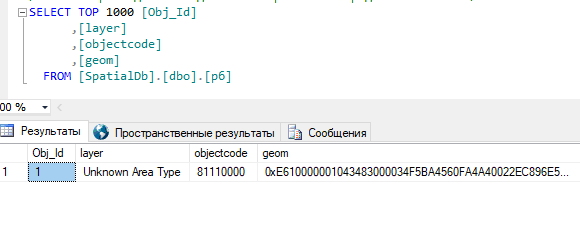
**Хранение д-х в ГИС:**\* Номенклатура – способ идент. листов геогр. карты  
\* Топология – набор правил целостности, опр. повдение простр-связанных геогр. объектов

**Возможности ГИС:**  
\* ввод д-х (2-3-мерные коорд в опр. СК, указание атрибут.столбцов, формирование слоев)  
\* преобр СК и порекций  
\* растрово-векторные операции  
\* цифр. обраб изобр (фильтрация, сводка листов, привязка к геогр основе, темат. классификация изобр)  
\* вывод д-х и документир рез-тов  
\* визуализация карт (палитры, штриховки, легенда карты)  
\* измерительные и полигональные операции  
\* аналит. операции (поиск ближ.соседа, выбор оптим. маршрута, анализ сетей…)  
\* анализ пов-стей (вычисл углов наклона и экспозиций склонов, опр. зон видимости)

**Популярные ГИС:**  
ESRI – ArcGIS, MapInfo – MapInfo Pro, Autodesk – AutoCAD Map 3D, LizardTech – GeoExpress, Кредо-диалог – CREDO, QGIS

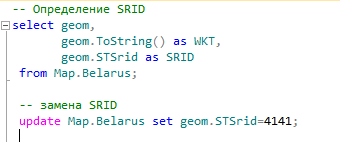
**Shape-файлы:** dbf, prj, shp, shx

**СУБД простр.д-х**: Open Geospatial Consortium (OGC) – разработка простр. д-х и сервисов  
***MS SQL Server – импорт:*** SqlSpatial, Shape2SQL, Catfood.Shapefile

**Пространственные данные:**  
\* **GEOMETRY** – плоские модели (предст. д-е в 2мер Эвклидовой ск)  
 - point (10 15)  
 - multipoint (10 10, 50 50)  
 - linestring (10 10, 20 20, 31 35) – прост. линия если не пересек себя  
 - multiline string  
 - polygon ((10 10, 10 20, 20 20, 20 15, 10 10))  
 - multipoligon  
 - geometry collection

\* **GEOGRAPHY** – геодезические модели (коорд широты и долготы)  
 - долгота – гориз. угол в диап -180\* +180\*  
 - широта – верт. угол в диап -90\* +90\*

**Внешние форматы д-х:**  
\* **WKT** (Well-Known Text) – предст с-м простр. коорд простр. структур и преобразоваий между с-ми простр. координат  
\* **WKB** (Well-Known-Binary)   
\* **GML** (Geography Markup Language) – грамматика языка разметки XML

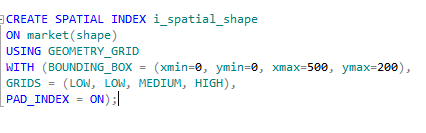
**SRID (Spatial Reference ID):**  
– простр. с-ма привязки координат  
могут сравниваться только объекты в одной SRID  
  
каталог – sys.spatial\_reference\_systems

**Методы:**  
\* методы по стандарту OGC и доп  
\* статические GEOMETRY::STGeomFromText()  
\* экземпляров класса @g.STContains()

**Действия с объектами:**\* Создание нового из *текст, двоич, GML* описания: STGeomFromText, …FromWKB, GeomFromGml  
\* Получение описания: STAsText, STAsBinary, ToString, AsTextZM, AsGml  
\* Создание объекта из сущ-щего: STBuffer, BufferWithTolerance, Reduce, STItersection, STUnion, STDifference, STPointOnSurface, STBoundary

**Сведения об объектах:**  
\* тип, действительность  
\* состав объекта  
\* размер, длина, ограничения  
\* SRID  
STGeometryType / STSrid, STIsValid / MakeValid , STNumGeometries / STGeometryN, STNumPoints / STPointN, STStartPoint / STEndpoint, STX / STY, STDimension / STLength / STArea , STIsEmpty

**Взаимоположение объектов:**  
пересечение, объед, вложенность, расстояние  
STEquals, STDisjoint, STIntersects, STTouch / STOverlaps / STCrosses, STWithin / STContains, STOverlaps, STRelate, STDistance

**Пространственные индексы:**  
\* В-деревья \* create spatial index \* geometry\_grid / geography\_grid \* bounding\_box / grids  


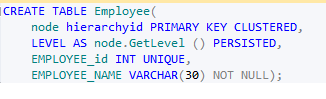
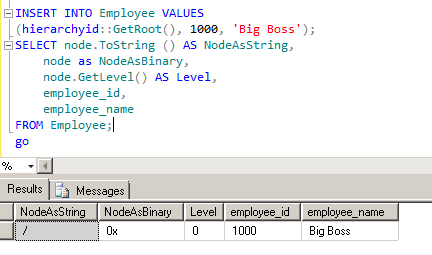
**Представления каталога:**   
sys.spatial\_indexes, sys.spatial\_index\_tessellations, sys.spatial\_reference\_systems

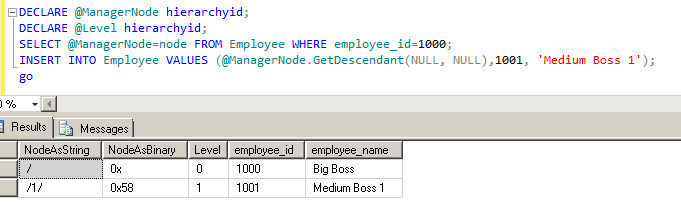
**ЛЕКЦИЯ 3. ТИПЫ ДАННЫХ XML, ИЕРАРХИИ**

**Иерархии:** организ. графики, структуры преприятий, списки файлов и папок, каталоги продуктов, ветки форумов  
\* *Adjency* *List* – список смежных вершин (parent\_id)  
\* *Nested Set* – указание влож-сти (left/right lower/upper)  
\* *Materialized Path* – полный путь (1/2/1)  
\* Добавление уровня (*level*)

**Иерархический ТД:**  
**hiararchyid** – системный тд пер-ной длины  
исп. для предст. положения в иерархии  
путь логически предст в виде посл-сти меток всех посещенных дочер. узлов, нач. с корня

биг босс /  
ср.босс /1 ср.босс/2  
мал.босс /1/1 мал.босс/1/2….

**Корневой эл-т:**  
  


**Добавление дочернего узла:**  


**GetDescendant (child1, child2):**Если родитель – null, возвр null  
Если одитель – не null, а потомки child1 и child2 – null, возвр одного потомка  
Если родитель и потомок child1 – не null, а child2 – null, возвр. потомка, кот. больше чем child1  
Если родитель, child1, child2 не null, возвр дочерний узел больше child1 и менше child2  
Если child1 (child2) не null и не явл дочерним узлом, то исключение  
Если child1 >= child2, то исключение

**Методы:**  
GetRoot, GetLevel, GetDescendant (child1, child2), GetAncestor(n), IsDescendantOf, GetReparentedValue(oldRoot, newRoot), Parse, ToString

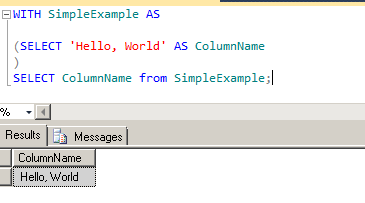
**Индекс по иерарх. типу:**  
depth-first, breadth-first  


**Иерархические запросы:**

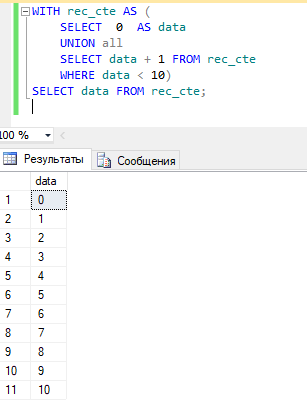
select **level**, empno, ename, mgr, *sys\_connect\_by\_path(ename, ‘/’),* *prior ename parent\_bode, connect\_by\_root ename root\_node*  
 from EMP  
 start with ename=’KING’  
 connect by prior empno=mgr  
 order siblings by ename

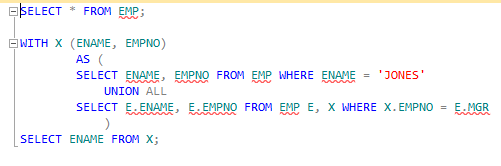
\* Start with – нач. узел иерархии   
\* Connect by – связь тек. с родительским  
\* Порядок прохода записей – от корня обход в глубину  
\* Level – уровень иерархии  
\* order siblings – упорядочние в рамках уровня  
\* sys\_connect\_by\_path – материализованный путь в иерархии  
\* connect\_by\_isleaf – явл ли узел листовым  
\* prior – родительский узел  
\* connect\_by\_root – корневой узел в иерархии  
\* connect\_by\_loop – зацикливание  
\* connect\_by\_nocycle – соединение до зацикливания  
\* connect\_by\_iscycle – 1 если вызыв. зацикл, иначе – 0

**CTE**

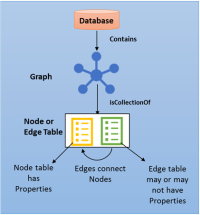
**Common Table Expressions** – общие табличные выражения   


\* обощ. таблич.выраж – временные рез. наборы, опр. в области вып-я единич. инструкций select  
\* не сохр в БД в виде объектов  
\* время жизни ограничено продолж-стью запроса  
\* могут ссылаться сами на себя  
\* 1 и тот же запрос м. ссылаться на СТЕ неск раз  
+ повыш. читамесоть и упрощ работу со слож запросами  
+ м. составлять промеж. СТЕ, не повыш произв-сть запроса

**Рекурсивность** – обращение через СТЕ к самому себе  


**Рекурсивные запросы:**  
исп. для: \* создание реку. запросов  
\* многократ ссылок на рез. таблицу из одной и той же инструкции  
\* замены предст-й  
\* группирования по столбцу, производного от скаляр. подзапроса выборки или ф-и, кот. недетерминирована

**ГРАФЫ**

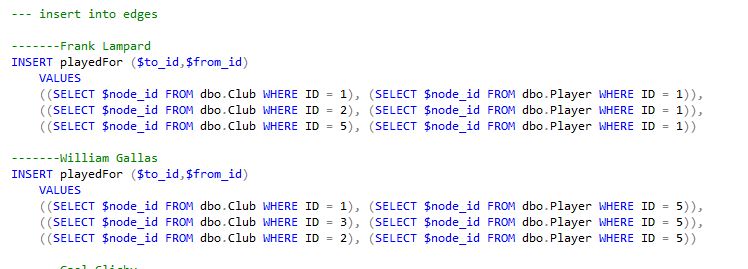
**Графовые БД:**  
м. создать один граф на БД, он сост из таблиц узлов и ребер  
 (edge – ребро)

**База данных графа** - коллекцию узлов (или вершин) и ребер (или связей). Реляционная база данных может обеспечить все, что может сделать база данных Graph. Однако база данных Graph упрощает выражать определенные типы запросов.

**Создание таблиц в граф. БД:**

create table (…) as NODE (узел)   
или просто “create table playFor as EDGE” (ребро)

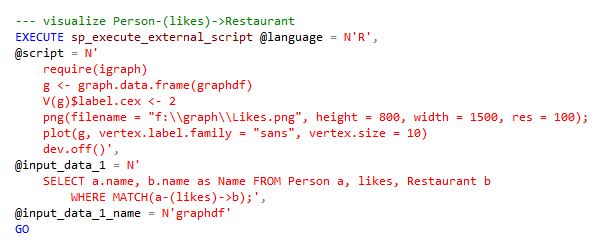
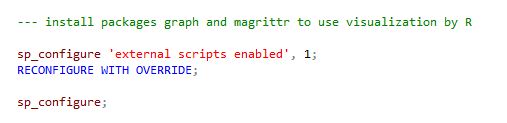
или с парам: table likes (rating INTEGER)

**Добавление ребер и узлов:**  


insert у NODE как обычно, у EDGE -   
 insert playFor**($to\_id**, **$from\_id**)   
 values ((select **$node\_id** from club where id=1), (select **$node\_id** from player where id=1));

**Выборка:**

//за какие клубы играл игрок?  
select Club.name from Club, layedFor, Player  
where MATCH ( Player –(playedFor)-> Club)  
and Player.name = ‘Julia’;

**Визуализация графа:**  


**XML**

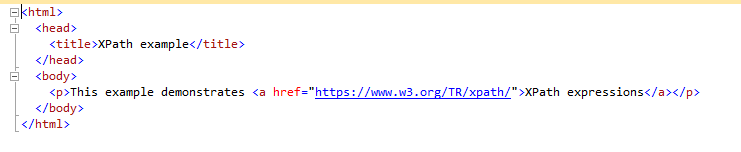
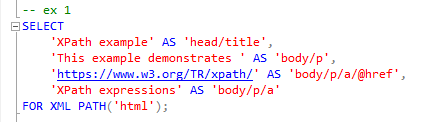
\* Самодостаточен – сод. данные и метад-е  
\* м. содерж д-е сложной структуры (деревья, рекрсия, графы, структ. д-е: повтор однрод структ, полуструкт д-е: разнородные структ, редкие д-е; д-е с разметкой тэгами)  
\* гарант порядок д-х  
\* отнош м/д-ми выражаются как «включено в», или атрибутами-ссылками на эл-ты (при наличии схемы)

**Сценарии использования ХМL:**  
\* транспорт д-х/обмен д-ми: business\_to\_bussiness (B2B), bus\_to\_consumer(B2C), appl\_to\_appl(A2A)  
\* управление хран-ем доков  
\* раздел хран-я и презентации д-х  
\* обмен сообщ-ми – Simple Object Access Protocol (SOAP)

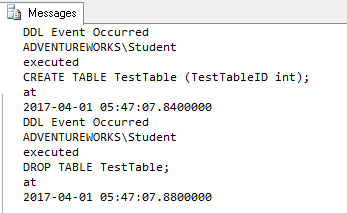
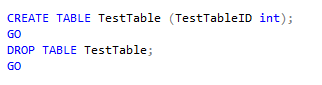
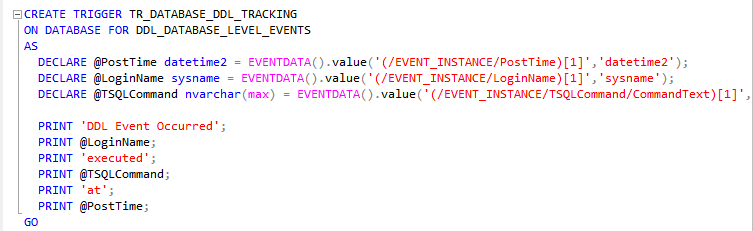
**Действия над ХМL:**\* вывод ХМL – мб сгенерирован хм-док вместо рез-тов запроса  
\* ввод ХМL – м. добавить д-е из хм-дока в таблицу  
\* обмен д-ми м/СУБД  
\* хранение и интеграция ХМL: демаршалинг – декомпозиция хм-дока на эл-ты и атрибуты, маршалинг – обр. процесс сборки хм-дока

**Тип д-х ХМL SQL Server:**  
\* м. содерж доки, фрагменты хм, м. связ с хм-схема-коллекшн  
\* запросы через XQuery  
\* модиф черех хмl-dml  
\* возможно индексирование  
\* проверка на корректность и соотв хм-схема

**Запрос д-х из ХМL:**\* язык запросов XPath – к эл-там хм-дока  
\* язык запросов XQuery – разраб для обраб д-х в формате хм

**Создание XML – FOR ХМL PATH:**

**Запросы к ХМL:**\*query() – вып запроса  
\* exit() – проверка сущ  
\* value() – возвр значение атрибута  
\* nodes() – возвр набор узлов  
\* modify() – измен дока: insert, delete, replace value of

**Применение XML в триггерах:**  


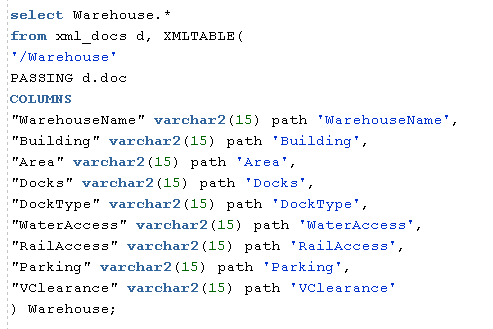
**XQuery:**  
\* FLWOR: FOR/LET/WHERE/ORDER BY/RETURN  
\* конструкторы хм-элементов  
\* включает в себя xpath 2.0 (/dox[@id=123])  
\* операторы, сохр порядок: порядок xquery посл-сти – flwr, хм-дока – выражения xpath  
\* статическая типизация: строгая с хм-схема, нестрогая – без схемы

**FLWOR:**

|  |  |
| --- | --- |
| for | итерация через ветки siblings |
| let | обознач значение как переменную |
| where | фильтр для итерации |
| order by | сортирует значения в рез наборе |
| return | указывается что будет возвращен хмl |

**Полнотекст. индекс:**\* хмl-разметка восприним. как граница слова  
\* содержание тэгов и атрибутов индексируется  
\* синтаксис – как и для всех ост. типов  
 create fulltext index on t (xDoc)  
\* м. комбинировать с XQuery:  
 - сначала full-text index для предвар отбора  
 - потом хм-индекс

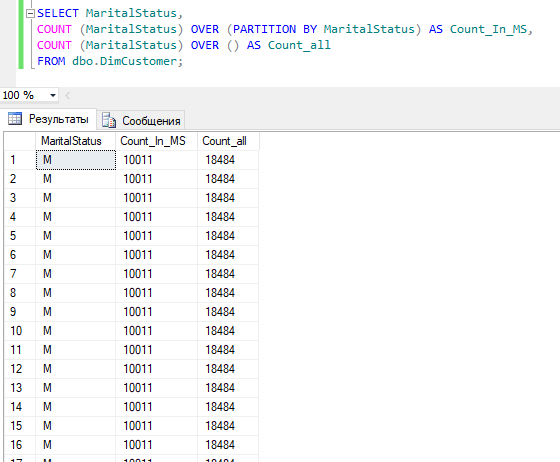
**Oracle XMLTYPE:**  
\* тип д-х, облегч обработку д-х ХМL  
\* мб исп в хранимых процедурах  
\* м представлять ХМ в SQL  
\* имеет встроен. ф-и, кот. работают с содержимым ХМ  


**XMLTABLE:**  


**ЛЕКЦИЯ 4. АНАЛИТИКА TSQL**

**ОКОННЫЕ ФУНКЦИИ**

**Обработка наборов строк, надо:**\* соотносить строки с пред. или послед. строками  
\* выделять группы строк, обраб независимо от других

**Окно** – набор строк, в рамках к-го происходит вычисление  
\* ОФ позвол. разбивать весь набор данных на окна  
+ ОФ не приводят к группированию строк  
\* строки сохраняют id-ы, а агрегированное значение доб. к каждой строке  


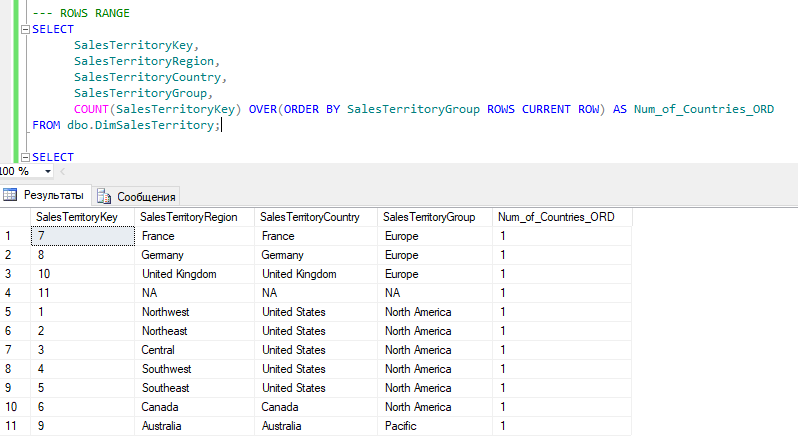
**ОФ** – ф., позв. осущ вычисления в зад. диапазоне строк внутри одного предложения select

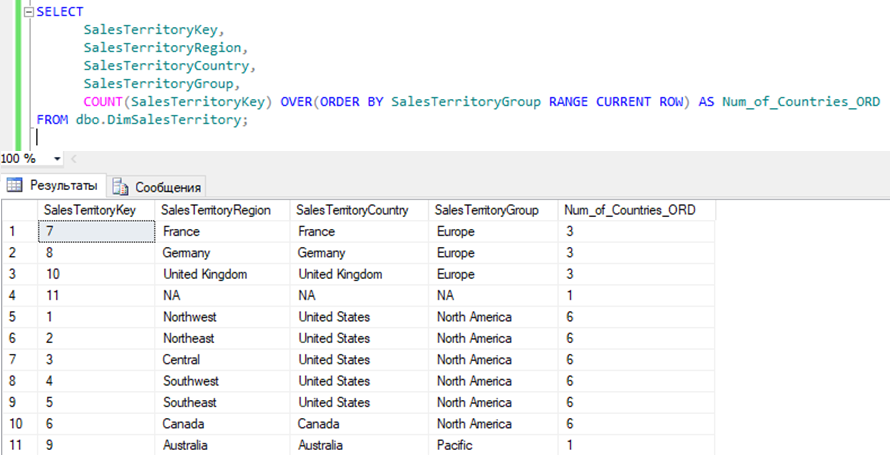
**OVER():**\* function(столбец) OVER  
\* PARTITION BY столбец для вычисл групп. значения (деление на окна)  
\* ORDER BY столбец для сортировки в окне  
\* ROWS или RANGE выражение для ограничения строк в пределах группы

**Виды ОФ:**  
\* *агрегатные* – возвр. значение, получ путем арифм. вычислений: sum, max, min, avg, count  
\* *ранжирования* – позв полу порядковые № записей в окне: ranl, dense\_rank, row\_number, ntile  
\* *сдвига* – возвр значение из др. строки окна: lag, lead, first\_value, last\_value  
\* *аналитические* – предост инфу о распределении д-х: percent\_rank, cume\_dist, percentile\_cont, percentile\_disc

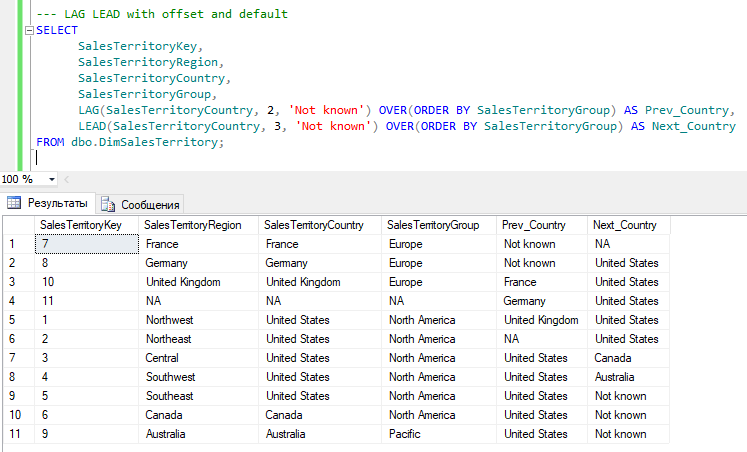
**Строки и диапазоны:**  
\* ROWS – строки рез. набора – ограничивает строки в окне фикс. кол-вом строк  
\* RANGE – диапазоны рез. набора  
они исп-ся вместе с ORDER BY

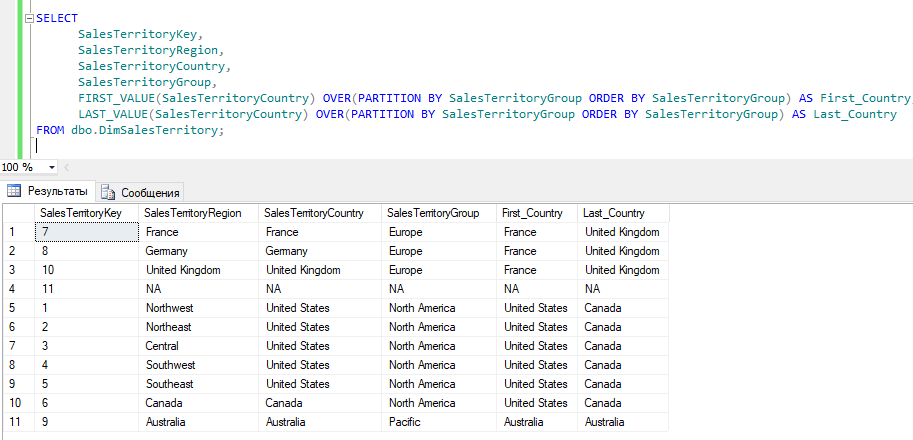
**Методы:**  
\* current row – тек. строка  
\* unbounded following – все записи после тек  
\* unbounded preceding – все пред. записи  
\* <N> preceding – зад. число пред. строк  
\* <N> following – зад. число послед. строк

**OVER (ORDER BY ROWS):**  


**OVER (ORDER BY RANGE):**  


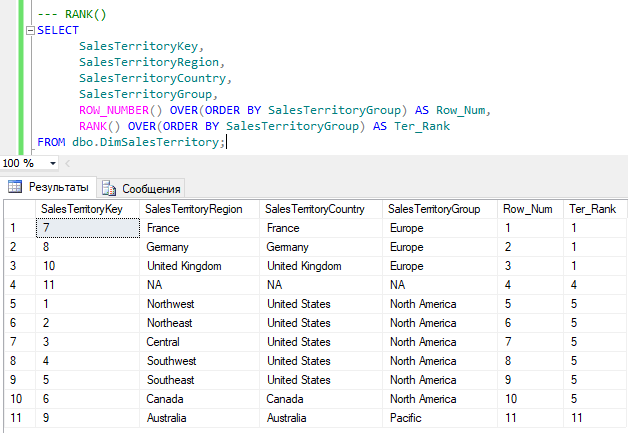
**ФУНКЦИИ СДВИГА ОФ:**

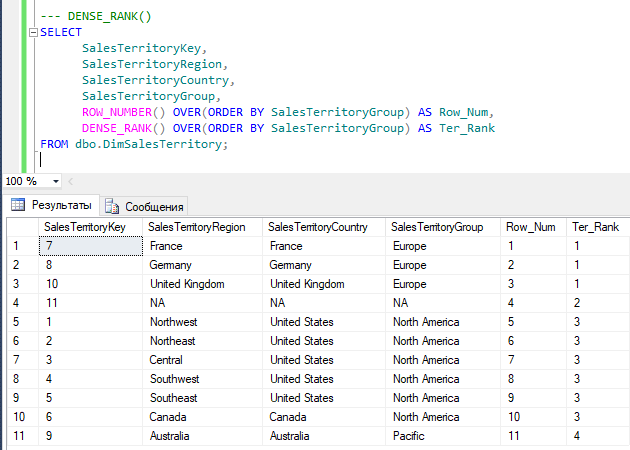
**LAG() и LEAD():**LAG (столбец, предыдущее значение к-го выводится) [ , число шагов назад, ‘not known’ вместо null  
LEAD (столбец, след. значение к-го выводится)  


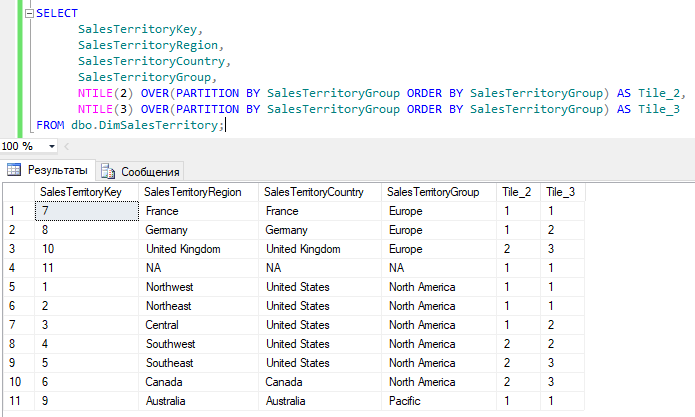
**FIRST\_VALUE() и LAST\_VALUE():**  
first(столбец, первое значение его указ. (мб в группе))  


**ФУНКЦИИ РАНЖИРОВАНИЯ**

**возвр. значение для к. строки группы в рез. наборе (для нумерации списка, составления рейтинга, постраничной выборки…)**

\* **ROW\_NUMBER()** – порядковый № строки в рез.наборе (order by обяз, по нему д-е отсорт, затем пронумерованы); если over (partition by… ) то № в мини-группе  
\* **RANK()** – в отличии от row\_number идет анализ значений, если нах. одинаковые, ф. возвр одинаковый ранг с пропуском след-щего  
(order by обяз, partition by мб)   
  


\* **DENSE\_RANK()** – как rank() но в случае нахожд. повтор. значений, возвр. ранг без пропуска следующего  


\* **NTILE()** – делит рез. набор на группы по столбцу [(кол-во групп)]  
если в группах не равное кол-во строк, в первой будет больше всего (10 : 4 3 3 )  


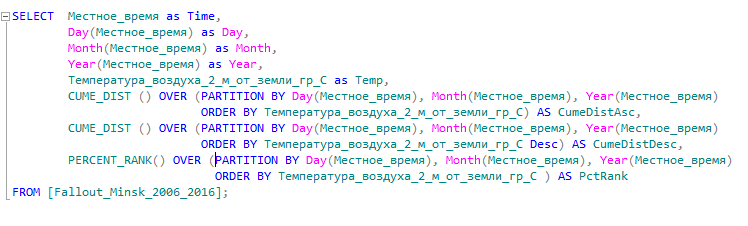
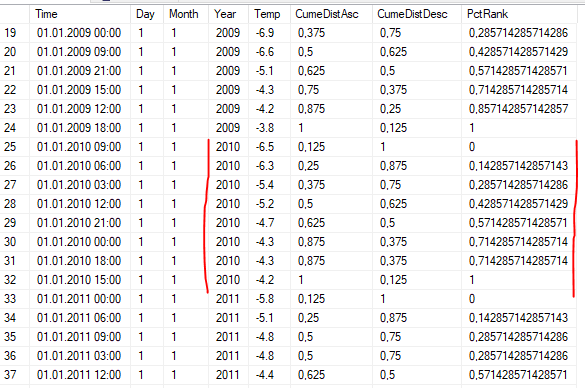
**АНАЛИТИЧЕСКИЕ ФУНКЦИИ**

**предост. инфу о распределении д-х и исп. в осн. для статистического анализа**

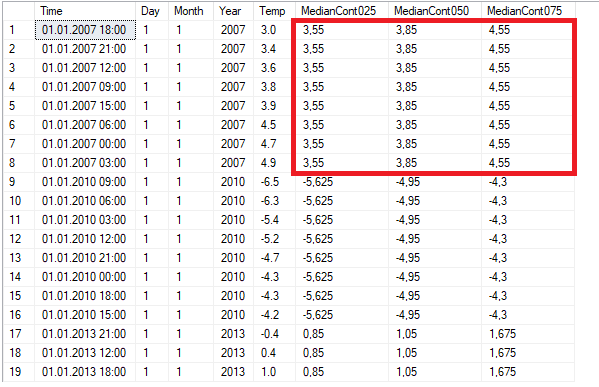
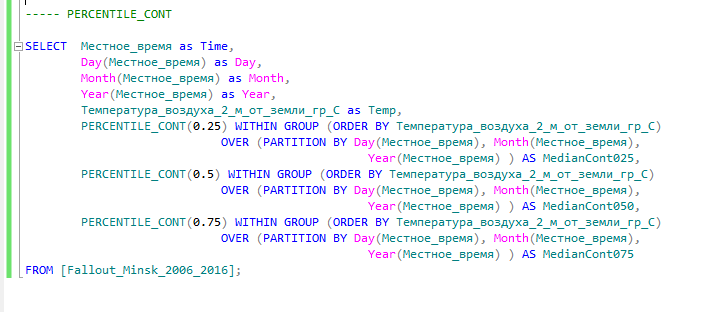
\* percent\_rank //распред-я рангов (обяз order by, мб partition by)  
\* cume\_dist  
\* percentile\_cont //обратного распред. (мб partition by)  
\* percentile\_disc

**Процентиль** – мера, в к-й % значение равно этой мере или меньше ее  
90% значений д-х находятся ниже 90-го процентиля  
10% значений д-х находятся ниже 10-го процентиля

\* **PERCENT\_RANK** – для к. записи возвр. % других записей в той же группе, которые имеют более низкие значений

\* **CUME\_DIST** – аналогично, но возвр. % значений, меньших или равных текущему  
  


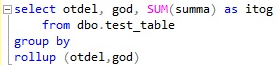
\* **PERCENTILE\_CONT** – вовзр. среднее значение суммы первой соотв. записи и след. записи  
\* **PERCENTILE\_DISC** – отобр значение первой записи, где *кумулят. распр* (% знач <= тек.) выше чем процентиль

значения сгруппир по набору строк как указ. в **within group**  
  


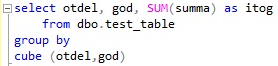
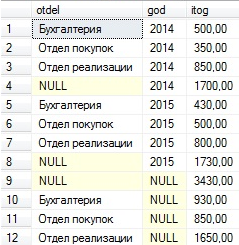
**РАСШИРЕННЫЕ ГРУППИРОВКИ**

\*rollup // расширения group by  
\* cube //  
\* grouping sets //  
\* grouping явл. ли агрегированным?   
\* grouping\_id уровень группировки

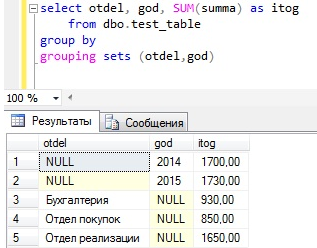
- **rollup** – формир промеж. итоги для к. указ. эл-та и общий итог  
строки со знач. null и есть промеж. итоги, а самая посл. строка – общий итог

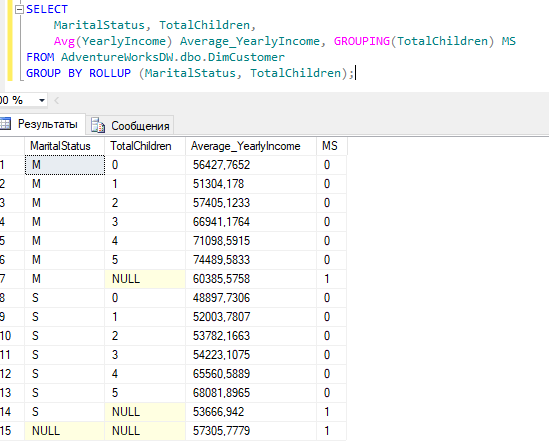
  

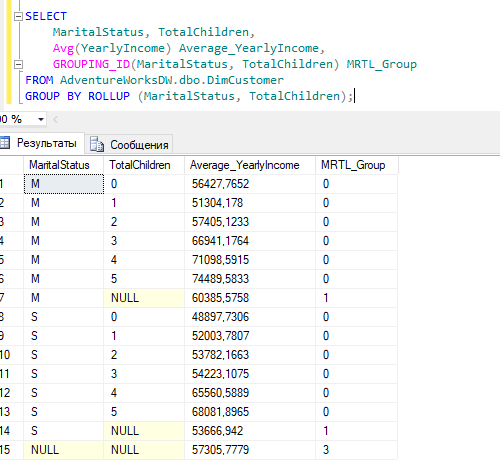

- **cube** – формир. рез-ты для всех возм. перекрестных вычислений  
как rollup, но группировка и промеж. итоги выполнены для указ. столбцов (,) отдельно

- **grouping sets** – формир. рез-ты неск. группировок в один набор д-х

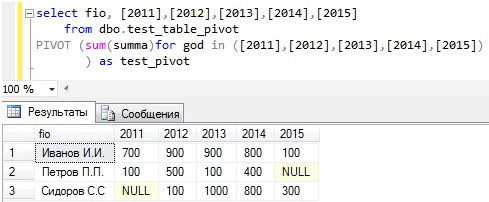


\* **GROUPING** – явл. ли значение агрегированным? (различ обыч. null от того кот. в кач-ве рез. *rollup, cube, grouping sets* = значит «всё»)  


\* **GROUPING\_ID** – уровень группирования  


**АНАЛИТИЧЕСКИЕ ЗАПРОСЫ**

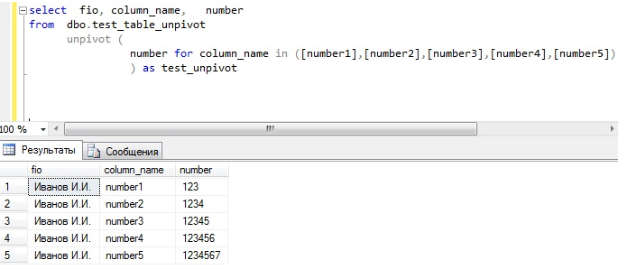
\* **OFFSET-FETCH** – часть order by ! – фильтр к уже отсорт. набору д-х  
- для разбиения рез. набор на части (страницы) и огранич. кол-ва строк  
 **offset** 5 **rows** //кол-во строк, кот надо пропустить  
 **fetch next** 3 **rows only** //кол-во строк, кот надо вернуть

\* **PIVOT** – поворачивает рез. набор, т.е. транспонирует таблицу  
- при этом исп. агрег.ф. и д-е группируются  
 **select** столбец для группировки, [знач по гор],…  
 **from** …  
 **pivot** (агр.ф. **for** столбец, сод. значения, кот. станут именами столбцов  
 **in** ([знач по гор],…))  


**обращение данных pivot:**  
\* сущности – атрибуты  
\* итоги и пром. итог  
\* соединение 2 таблиц с один. столбцом

\* **UNPIVOT** – обратные д-вия, но ничего не (раз)агрегирует  
 **select** столбцы **from** …  
 **unpivot** (псевдоним для значений из столбцов num1,num2…  
 **for** псевдоним столбца, кот будет сод. значений наших колонок **in** …)





**ЛЕКЦИЯ 5. АНАЛИТИКА CUBE**

**Бизнес-аналитика** – область технологии БД  
\* процесс интегрирования всех необх. д-х в 1 источник д-х, к к-му конечные юзеры могут осущ. нерегламентированные запросы для анализа этих д-х  
\* цель: принятие лучших биз.решений

**Работа с д-ми:**  
\* **OLTP** – online transaction processing – оперативная обработка транзакций  
 *\** *доступна непрерывно  
 \* хранит тек. данные*  
 - реляционность – обычно  
 - много юзеров  
 - постоянное вып. операций чт/зап над небольшим V строк

\* **OLAP** – с-мы биз.аналитики  
 *\* доступна т. после очистки д-х и их загрузки в БД  
 \* сохр. историю д-х*  
 - д-е собир из различ. источников  
 - д-е приводятся в соглас. состояние  
 - д-е регулярно загруж в БД (хранилище д – data warehouse, витрина д – data mart)  
 - осн. режим – только чтение  
 - кол-во одоврем. юзеров небольшое  
 - создание отчетов по различ. аспектам  
 - вып. сложных запросов для сравн д-х

**Хранилище д-х** – бд всех данных организации, к к-ым юзеры могут иметь единообразный доступ  
\* *проблемы*: большой V д-х, получены в разное время, хран. в разных БД  
\* *решение*: консолидация д-х

**Консолидация д-х** – все один. запросы, вып-мые к хранилищу д-х в разное время, д. возвр. один. рез-ты. Процесс консолидации – ETL (Extract, Transform, Load)  
\* сбор д-х из различ. источников  
\* очистка д-х  
\* обесп. кач-ва д-х

**Организация д-х:**  
\* **Витрина** (киоск) – бд всех д-х на уровне подразделения организации, к к-м юзеры м. иметь единообразный доступ  
\* **Хранилище** – бд всех д-х на уровне организации, к к-ым юзеры могут иметь единообразный доступ

**Проектирование:**  
\* для проектирования Х и В примен размерная и простр. модель  
\* центр. таблица – *fact table*  
\* таблицы измерений – *dimension table*  
\* числ. столбцы т-цы фактов, кроме PK, наз. мерами – *measure*  
\* имена столбцов т-ц измерений исп. в кач-ве заголовков в отчетах

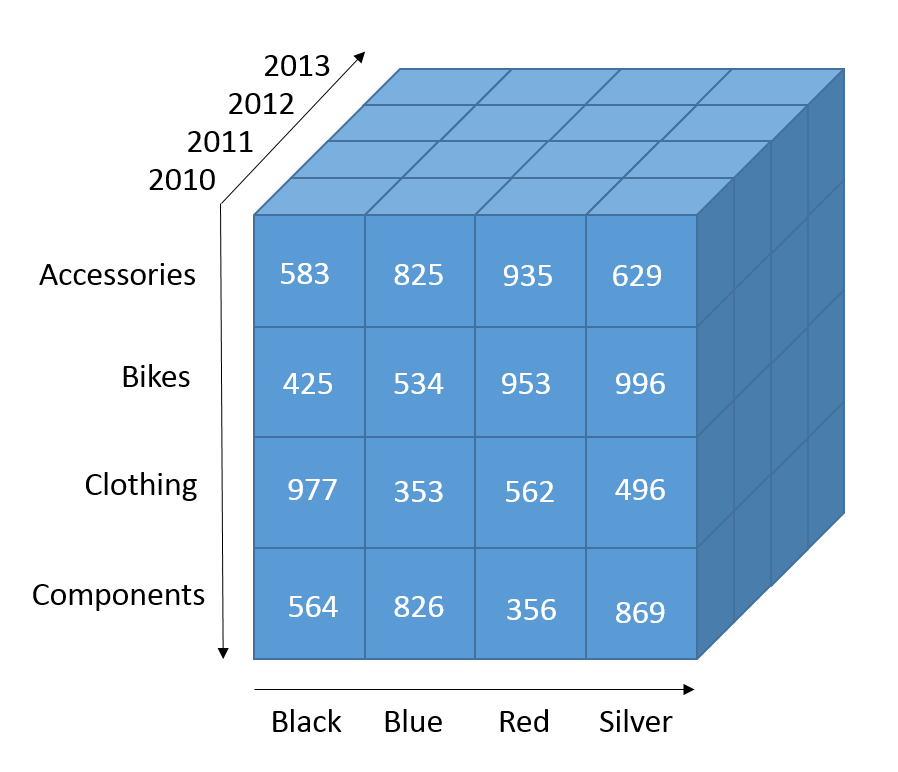
**Проектирование хранилищ д-х:**звезда .--.<: снежинка .-.<:\_.

|  |  |
| --- | --- |
| **Таблица фактов:** \* обычно одна таблица \* сод. больш-во строк хр-ща \* имеет составой PK \* неключ. столбцы им. числовой ТД \* осн. задача – дать сумму, макс, мин, ср… знач | **Таблица измерений:** \* много т-ц \* сод. сравнительно небольшой набор строк \* PK сост. из 1 столбца \* столбц сод. описания измерений \* разреженность измерений |

**Агрегирование:**  
\* агрег. т-цы – промеж. т-цы для хран. заранее подсчитанных низкоур. д-х  
\* процесс создания – агрегирование д-х

**Уровень агрегирования:**  
\* не выполнять агрегирование  
\* вып. исчерпыв. агрегирование для к. возм. комбинации запросов  
\* вып. необх. агрегирование

*куб (cube), измер (dimension), эл-т (member), кортеж (tuple), срез (slice), иерарх (hierarchy, ячейка – cell, мера – meause, группа мер – measure group, подкуб – subcube)*

\* **Куб** – подмн-во д-х из хр-ща, м. иметь любое кол-во измерений, кот. можно организовать в многомерную структуру (это основа многомерной БД)   
 

\* **Измерение** – описание для эл-та д-х, по к-му провод. анализ; его **размер** - кол-во эл-тов измер  
\* **Элемент** – 1 значение изм-я; его **значение** – знач ассоциируемое с эл.; его **атрибут** – полная коллеция эл-тов измер  
\* **Кортеж** – одно значение д-х с координатами по всем измерениям  
\* **Срез** – 1+ координат произвольная, остальные фиксир  
\* **Иерарх** – день,нед,мес… Иерархия атрибутов: день-нед, год-квар-мес,день  
\* **Ячейка** – набор точек простр-ва многомер. куба  
\* **Мера** – значения в ячейке; мб неск (сумма,кол); все вместе – **измерение мер**; частич. объединение – **группа мер**  
\* **Подкуб** – часть пр-ва куба  
\* **Набор** – сов. кортежей, опр-ых с пом. одних и тех же измерений

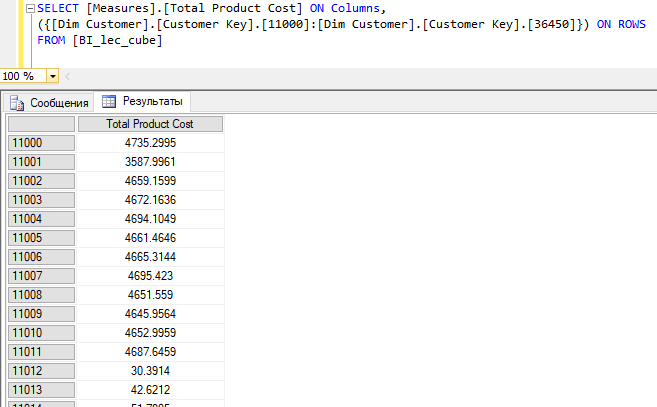
**Виды архитектур:**  
\* *реляц*. OLAP-ROLAP – relation online analytical processing – предвар. вычисленные д-е не сохр  
\* *многомер* OLAP-MOLAP – multidimensional… - д-е на уровне листьев и их агрегир. д-е сохр в кубе  
\* *гибридная* OLAP-HOLAP – hybrid… - агрегир. д-е сохр, данные на ур. листьев не сохр

**Доступ к д-м:**  
\* составление отчетов  
\* оперативная аналитика  
\* извлеч инфы из д-х – data mining

**Службы SQL Server Analysis Services** – группа сервисов кот. исп-ся для созд многомерных бд и управления этими д-ми  
(1994-5 – Панорама, 1996 – MS, 1998 – MS OLAP Services и MDX)

**Создание и обработка многомерного куба:**  
\* создание проекта биз-аналитики  
\* опред источников д-х  
\* созд представлений источников д-х  
\* создание куба  
\* агрегирование; \* обработка ку-ба; \* его просмотр

**Извлечение д-х из многомерных БД:**  
1) Power Pivot for Excel  
2) язык **MDX** **– multi-dimensional eXpressions** (1998, часть спецификации OLEDB для OLAP)  
 select задает рез. набор (подкуб), сод. подмн-во многомер.д-х, выбр из куба  
 select … from … where …  
 \* необх. указ. **оси** (подмн-ва измерений):  
 ON [ columns | rows | pages ]   
 \* вычисляемые знач: with [ set | member ] … as  
 \* коммент -- // /\*..\*/  
 \* арифм + - \* /  
 \* опер для наборов + - \*  
 \* лог опер ><=.. and or xor not  
 \* спец. операторы {} , :  
 \* **функции**: числовые, обработки строк. значений и наборов, для эл-та измер, для иерархий, прочие



**ЛЕКЦИЯ 6. БЕЗОПАСНОСТЬ**

**Задачи:**  
\* компромисс между потребностями юзеров и необх-стью защиты д-х  
\* ограничение прав доступа к БД до оптим. уровня  
\* закрытие уязвимых мест в с-ме безопасности

**Аспекты безопасности:**  
\* физ. защита  
\* привилегии бд  
\* безоп. прил-я, передачи, клиента  
\* шифрование  
\* аудит

**Аутентификация** – опред, кто такой конкр. юзер  
**виды**: \* однофакторная: - известное юзеру, - принадлежащее юзеру, - исп-щее физ. хар-ки юзера  
 \* многофакторная  
**режимы**: \* аут. Windows (для доменных БД – внутри организации)  
 \* смешанная (не исп. домены, доступ сторонних юзеров)

**Авторизация** – опр какие д-вия м. производить аутентиф. юзеру

**Спец. учетные записи:**   
SQL Server - sa; доменные – Administrator, Administrators

**Политики безопасности:**  
\* сложность паролей  
\* длина паролей  
\* история паролей  
\* порог блокировки уч.записи  
\* длительность блокировки уч. записи  
\* возраст пароля

**Модель безоп-сти:**  
\* **Участники** – security principals – объекты, к-ым мб выдано право доступа к опр. объектам БД, имеют уник. SID  
\* **Объекты** – securables – это объекты доступ к к-ым контролируется  
\* **Разрешения** – владельцы объектов имеют неотзываемые разреш. на их управление  
\* **Схема** – именованный контейнер объектов БД, позв. сгруппировать объекты в отд. пр-вах имен

**Участники:**  
\* ОС : уч. запись домена Windows; лок. уч.запись Windows; группа Windows  
\* SQL Server : уч. запись SQL Server; уч. запись SQL Server, сопост. с асимм.ключом или сертиф-ом  
\* БД : юзер БД, сопост. с ключом/сертиф; юзер БД, сопост. с уч.записью Windows; роль прил-я; роль БД

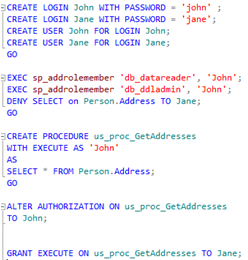
**Защищаемые объекты:**  
\* Сервер : тек. экземпляр Са, БД, уч.запись (login), роль Са  
\* БД : роли БД и прил-я, юзеры  
\* Схема : процедуры и ф-и, т-цы и предст коллекция схем ХМL

**Встроенные роли Сервера:**  
\* bulkadmin – копирование  
\* dbcreator – создание бд  
\* diskadmin – упр-е дисковыми файлами  
\* processadmin – упр. процессами  
\* securityadmin – безоп  
\* serveradmin – конфигурация Са  
\* setupadmin – процедуры запуска  
\* sysadmin – любые д-вия, безотзывная

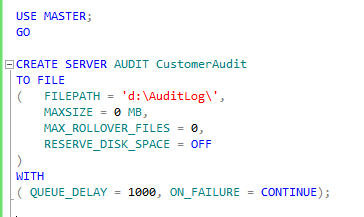
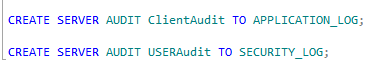
**Роли БД:** встроенные, пользовательские, приложений  
**Встроенные роли БД:** public, db\_accessadmin, db\_backupoperator, db\_datareader(writer), db\_denydatareader(writer), db\_dlladmin, db\_owner

**Роль public:**  
сод. в каждой БД, нельзя удалить, можно изменять разрешения  
нельзя доб/удал юзеров, разрешения наслед. всеми ост. юзерами и ролями,  
только разрешения, необх. для всех юзеров

**Привилегии:** alter, control, create, delete, execute, impersonate, insert, references, select, take ownership, update, view definition  
**просмотр привил:**   
sys.fn\_builtin\_permissions() – опис иер-хии встроен. разрешений Са  
has\_perms\_by\_name() – д-вующие разрешения на объект:  
 \* предост. непосред юзеру  
 \* вытек. из ращрешений более выс. ур  
 \* предост. роли, членом к-ой явл. юзер  
 \* принадл. роли, членом к-ой явл. юзер

**Заимствование прав:**\* **Имперсонификация** – разрешение заимствовать права у др. юзера для вып-я опр. блока кода  
 

**Цепочка владения** (ownership chain) – sql server пропускает проверку привилегий для улучшения произв-сти для 2 объектов с один. владельцем  
 sp\_configure ‘cross db ownership chaining’ , 1;  
 reconfigure;

**Аудит** – отслеживание и протоколирование событий, происх. на Се  
в SQL Server: - аудит Са для событий на уровне Са  
 - аудит БД для событий на уровне БД  
**Журнал аудита** – файл вс, жур. Windows Application Log, Windows Securty Log  
  


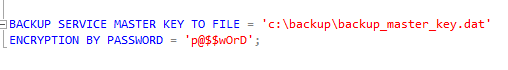
\* **queue\_delay** – задерэка очереди сброса на диск (0 – синхронно, 1000 – 1 сек)  
\* **on\_failure** – д-вия в случае невозм-сти продолжения аудита (continue, fail operation, shutdown)  
\* **reserve\_disk\_space** – заранее размещ. на диске файл в соотв с maxsize  
\* **max\_rollover\_files** – макс кол-во файлов кроме тек  
\* **max\_files** – макс число файлов аудита, кот. мб созданы. При достиж предела переключение на 1й файл не происх. При достиж предела max\_files любое д-вие, кот. вызывает создание доп. событий аудита, заверш ошибкой

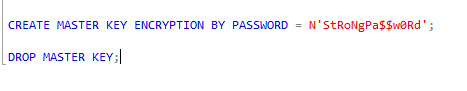
|  |  |
| --- | --- |
| database\_change\_group | изменение бд |
| database\_object\_change\_group | измен на объектами бд |
| login\_change\_password\_group | измен паролей |
| server\_state\_change\_group | модификация служб |
| successful\_login\_group | успешный вход |
| failed\_login\_group | неудачная попытка входа |
| logout\_group | выход юзеров из sql server |
| database\_object\_access\_group | обращ к сертификатам |
| database\_ownership\_change\_group | alter authorization |
| select, insert, delete, update, execute | выполн операторов |

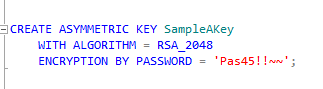
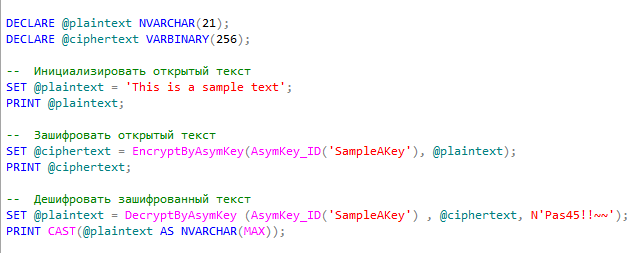
**Каталоги аудита:**  
fn\_get\_audit\_file, sys.server\_audits, sys.database\_audit\_specifications, sys.database\_audit\_details, sys.dm\_server\_audit\_status

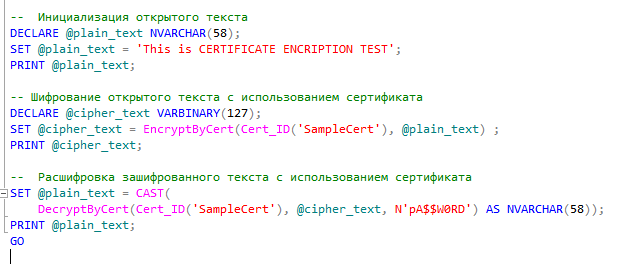
**Шифрование:**

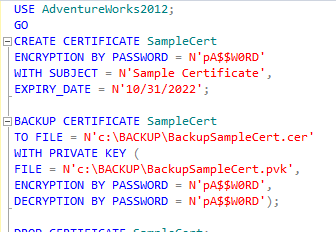
\* ServiceMaster Key – шифрует мастер-ключи бд  
\* Database Master Key – шифрует симм. ключи  
\* Симм. ключи (произв-сть) – шифруют д-е – главный ключ службы и бд (3des, 2017-aes)

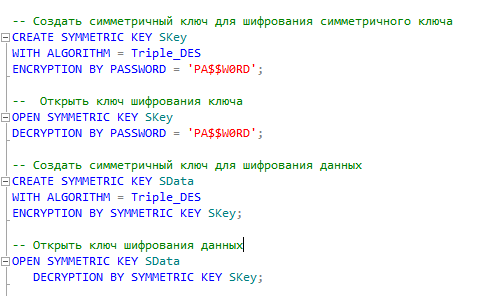
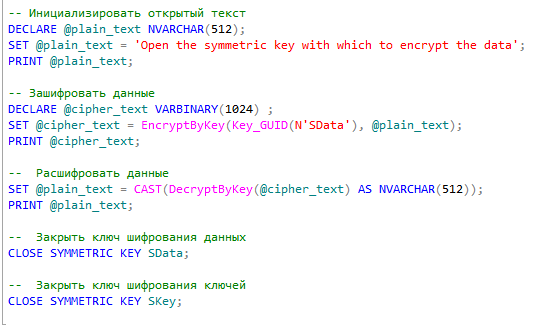
**Service Master Key:**  
мастер-ключ службы, создается при созд. экземпляра, исп-ся для шифр. пароля связанного Са, уч. данных или главного ключа бд  
\* alter, restore, backup  
\* обяз. создавать резерв. копию!  


**Database Master Key:**  
мастер-ключ бд, примен. для защиты закр. ключей сертификатов и асимм. ключей  
\* create, alter, drop; restore, backup; open, close  
\* обяз. создавать резерв. копию!  


**Ассим. ключи:**  
пара ключей – public и private  
алгоритм – rsa (512,1024,3072,2048,4096)  
\* create, alter, drop  
  


**Сертификат** – подписанная цифр. подписью инструкция, кот. связывает значение откр. ключа с id-ом юзера, у-ва или службы, имеющей соотв. закрытый ключ  
\* сод: откр. ключ субъекта  
\* идент. д-е субъекта: имя, емэйл  
\* срок д-вия (Valif From, Valid To)  
\* идент. данные поставщика сертификата  
\* цифр. подпись поставщика  


SQL Server м. генер автоподписанные серт-ты:   


**Симметричные ключи** – des, 3des, rc2, rc4, aes  
\* create, alter, drop; open, close; close all symmetric keys  
  


**Прозрачное шифрование:**  
**TDE – transparent data encryption**  
шифрует каждую страницу (перед записью на диск), расшифровывает при к. запросе; обнуление и шифрование журнала транзакций; шифрование tempid

**Представления каталога:**  
\* sys.asymmetric\_keys  
\* sys.certificates  
\* sys.crypt\_properties  
\* sys.dm\_database\_encryption\_keys  
\* sys.key\_encryptions  
\* sys.master\_key\_passwords  
\* sys.openkeys  
\* sys.symmetric\_keys

**SQL Инъекция** – атака путем внедрение кода sql – это атака, при к-й вредоносный код вставляется в строки, передающиеся затем в экземпляр sql server для синтаксического анализа и выполнения  
 ! sql server вып. все получаемые синтаксически правильные запросы  
\* вставка кода в польз. входные пер-ные, кот. объединяются с к-дами  
\* внедрение небезопас. кода в строки т-ц или метаданные – для послед. вып-я динамического sql  
\* завершение текст. строки и добавл к ней новой к-ды  
\* внедряемая строка заканч меткой комментария «--»

var sql\_string = “select \* from Orders where UserName = ‘” + UserName + “”;  
Blinova’; drop table Orders –

**Проверка достоверности вводимых д-х:**\* размер и тип ввод.д-х – устн ограничения  
\* содержимое строк. пер-ных – только ожидаемые значения  
\* все вводимые д-е – проверить на соотв. схеме при работе с ХМL-доками  
\* только хранимые процедуры  
\* никогда не созд инструкции tsql непоср. из данных, вводимых юзером  
\* многоэтапная проверка достоверности  
\* объединение введенных д-х – после проверки  
\* проверять и отклон исп-е строк, из к-х м. создать имена файлов, и д-е, содержащие символы ; -- ‘ /\\* \*/ xp\_

**ЛЕКЦИЯ 7. SQLLITE**

**SQLLITE:**  
\* встроенная, однопользовательская  
\* фс: БД – обычно 1 файл (+ файл журнала)  
\* ограниченный sql92  
\* не строго типизируемая  
\* есть нек. объекты (индексы, view)  
\* встроенной с-мы программирования нет  
\* прогр. объекты : триггеры  
\* прогр. интерфейс : библиотека ф-й  
\* *ПО*: DbBrowser for SOLite, SOLite Expert Professional, SOLite Studio  
\* *подключ*: скачать с оф. сайта, подкл к проекту ч/NuGet

**Классы хранения:**  
\* не типы данных, а класс хранения  
\* в любом классе м. хранить д-е любого типа  
\* есть: integer (ц.ч.), text (utf-8-16), blob, real (IEE-752), null (пустое знач)

**Аффинированные типы:**  
\* рекомендуемый тип для сохраняемых в столбце значений  
\* не явл. обязательным  
\* аф-сть столбца опр-ся неяво во t создание т-цы при объявлении ТД, кот будет хранится в столбце

**TEXT:**  
\* предназначены для хран значений с классом null, text, blob  
\* числ. значение будет преобразовано в строку

**NUMERIC, INTEGER, REAL:**\* мб исп для хран значений всех классов хран  
\* есть помест текст.знач, то SOLite постарается преобр его в integer / real  
\* если невозм, то сохр значение с классом TEXT  
\* для real – ц.ч.-знач будут преобр в числа с плав.т

**BLOB:**  
\* не предпоч. какой-либо класс хр  
\* не предприним. попыток преобр д-х из одного класса в другой

**Порядок аффинированности:**  
1. Если объявление типа сод. строку “INT”, то столбец ассоц. с аффин. INTEGER  
2. “CHAR”, “CLOB” или “TEXT” – как TEXT  
3. “BLOB” и тип не указан – с NONE  
4. “REAL”, “FLOA” или “DOUB” – REAL  
5. иначе NUMERIC

**Операторы сравнения:**  
\* допустимо сравн значения с разными классами хр  
\* =, ==, <,.. !=,<>, BETWEEN, IN, IS, NOT IN, IS NOT  
\* знач integer или real считается меньше чем знач text или blob  
\* при сравн integer или real исп числовое сравн  
\* знач с классом null счит меньше любого другого, даже другого знач null  
\* знач text меньше blob  
\* сравнение знач blob опр-ся с исп-ем ф-и memcmp()  
\* нет булевого типа д-х – ц.ч. 0 (false) и 1 (true)  
\* выраж, кот. явл. ссылкой на знач столбца, им. такую е афф-сть как и столбец  
\* выраж “CAST (выраж AS тип)” имеет ту же афф-сть  
\* все матем. оп-ры преобр операнды в NUMERIC до вып-я оп-ций  
\* преобр происх даже когда оно необратимое и с потерями  
\* операнд null приводит к пустому рез-ту любую матем. оп-цию  
\* операнд в матем. оп-ции, кот не привод-ся к числовому и не null, преобр в 0 или 0.0  
\* a BETWEEN b AND c – в каждом из сравнений примен различ афф-сти

**SQLITE DDL**

|  |  |
| --- | --- |
| CREATE TABLE | \* обяз имя таблицы \* имена с solite\_ зарезервированы \* temp(orary) созд временный объект \* if not exist созд, если не было \* нельзя т-цу, предст и индекс с один. именами, а триггер можно \* удал : drop table |

**ROWID** – уник. № записи в каждой таблице  
\* в запросе select \* from table не показ.  
\* записи в т-це – И-дерево по rowid ==> есть индекс

**PRIMARY KEY:**  
\* unique + not null  
\* по умолч автоинкремент на 1  
\* значения ключа совп с rowid  
\* мб составным  
\* при insert м. не указ. столбец PK  
\* если PK задан, то ROWID авто- продвигается

**WITHOUT ROWID:**  
\* м. создать т-цу с опцией такой – к. таблица д. иметь PK  
\* нет механизма авто-инкремента  
\* исп-ся для оптимизации хранения и обраб

**ON CONFLICT:**  
\* метод разреш конфликтов при вставке  
\* rollback (генер ошибку, вып rollback для всей транз)  
\* abort (генер ошибку, вып rollback т. для тек. оператора)  
\* fail (генер ошибку, вып rollback т. д/последней оп-ции в рамках оператора)  
\* ignore (сбрас тек. операция, оператор продолж. работу)  
\* replace (для огран. PK и UNIQUE: при insert, update удал конфликтную строку и доб. новую)  
 (для огран. NOT NULL: замен на знач по умолч, если такого нет, то abort)  
 (для огран. CHECK: abort)

**PRAGMA foreign\_keys:**  
\* опред включены ли в данный момент внеш. ключи,  
\* по умолч они отключены для обрат. совместимости

**PRAGMA:**  
\* расширение sql, специфичное для sql-lite  
\* исп. для изменения работы библиотеки solite  
\* нет гарантии обр. совместимости для утверждений prag  
\* сообщения об ошибках не генер если выдается неизвестная prag

**FOREIGN KEY ON DELETE / UPDATE:**  
\* restrict – явное указ на запрет измен/удал PK значение, если на него ссылаются FK – действ по умолч  
\* no action – явное указание не предприним д-вий при измен или удал PK значений (фактически отсут ограничение)  
\* set null \* set default \* cascade

**FOREIGN KEY DEFERRABLE:**  
\* отлож проверка ограничения  
\* deferred – проверка осущ при commit  
\* immediate – проверка осущ сразу

**SQLLITE DML**

**Последовательность секций:**  
\* select distinct, from, where, group by, having, order by, limit offset

**LIMIT:**  
\* limit 10 (число возвр строк)  
\* limit 10 (число возвр строк) offset 5 (число строк сдвига)  
\* limit 5 (число строк сдвига), 10 (число возвр строк)

**COLLATE:**  
\* позв. задать способ сравнение д-х:  
- *binary* – побайтовое сопост. строк независимо от кодировки текста  
- *nocase* – как binary кроме 26 пропис. букв ASCII  
- *rtrim* – как binary, но игнорир пробелы в конце строки

**Операторы сравнения:**  
\* **LIKE** – срав текста с образцом : like ‘A%’ (% ноль+ символов, \_точно один символ)  
\* **GLOB** – побайтовое сравн текста с образцом (\* ноль+ символов, ? точно один символ)

**Обработка дат SQLLITE:**  
\* текст в виде строки формата “yyyy-mm-dd hh:mm:ss.sss”  
\* real как числа юлианского календаря – число дней с полудня 24 ноя 4714 г. до н.э.  
\* integer как время unix – кол-во сек с 1970-01-01 00:00  
\* врем. зоны не поддерж  
\* смещ на лет/зим время не поддерж  
\* поддерж формат UTC-Universal coordinated time – почти идентичен t по Гринвичу  
\* поддерж литералы времени: current\_time, current\_date (ymd), current\_timestamp (yy-m-d h:m:s)

**Функции работы с датами:**  
\* date | time | datetime (*timestring, modifier, modifier* …)  
\* strftime (format, time, modifier…)  
 %d – день месяца (1-31)  
 %f – секунды с .sss (00-59)  
 %H – час (00-23)  
 %j – день года (001-366)  
 %J – юлианский № дня  
 %m – месяц (1-12)  
 %M – минут (0-59)  
 %s – секунд с 01.01.1970  
 %S – секунд (0-59)  
 %w – день недели (0-6, 0 – ВС)  
 %W – неделя года (00-53)  
 %Y – год

**SQLITE JOIN**

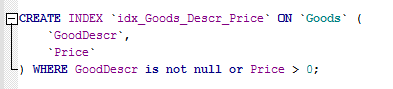
*поддерживаются:* подзапросы, конструкция WITH, соединения, в т.ч. внешние, объединения мн-в UNION ALL, UNION, INTERSECT, EXCEPT

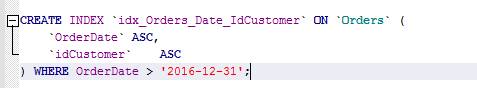
**СОЕДИНЕНИЯ JOINS:**  
inner join, cross join, join … using (id), natural join, self join, left outer join  
*запись*: select .. from t1 **join** t2 **on** (t1.id = t2.id) – ansi  
 select .. from t1, t2 **where** t1.id = t2.id

**SQLITE VIEW, INDEX, TRIGGER**

\* **представление** – view – поименов. select запрос  
\* **индекс** – стандартный префикс idx\_, строится на столбцах одной таблицы, на предст-нии индекс построить нельзя, на вирт. таблице индекс построить нельзя, нет ограничений на кол-во индексов для 1 таблицы

**автоиндексы** – при указании unique, pk; не мб удалены drop index; показаны в slite\_master; индекс по rowid не показ в slite\_master

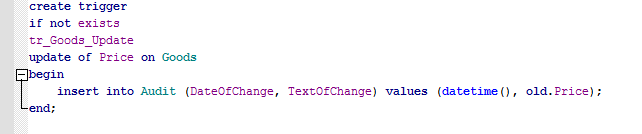
**индексы where** – если в инд исп OR, то в запросе, исп-щем инд-с мб указано 1 из условий:  
 - если в инд-се исп точное рав-во (=а), то запрос будет исп ин-с при таком же условии  
 (between a and a – не подойдет)  
 - если в инд-се указано is not null, то в запросе, исп-щем <> инд-с не применяется  


**составные индексы** – исп. для ускор вып-я запросов по неск. полям;   
 порядок расп столбцов важен!  


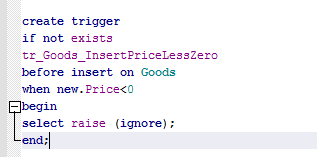
**индексы с порядком сортировки** – указ. порядка хранения знач ключей оправдано для запроса с предложением order by  


**индексы в выражениях** – в инд м. использ выражение от столбца/набора столбцов

**ТРИГГЕР**

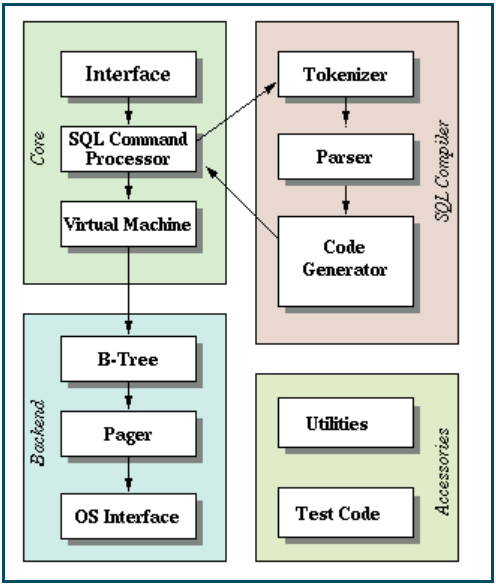
для к-ды update м. указ. столбец, кот. будет отслеживать триггер  


\* м. исп фразу **when** для уточнения условия, при кот. тр вып  
\* усл when интерпр как логическое  
\* при when оп-ры вып. только для строк, для кот. это предл истинно  
\* могут примен old и new префиксы

\* **RAISE()** – спец. ф-я обработки ошибок в тр  
\*2 аргум – действие и комментарий  
\* ignore – игнор строку, породившую конфликт и прод. вып послед. оп-ций  
\* rollback – откатить все оп-ции к исх. сост  
\* abort – отмен не все вып-ные ранее оп-ции, а т. ту, при к-й возник конфликт  
\* fail – прервать вып тек. оп-ции, сохр рез-ты успешных оп-ций, остальные не вып  


*! При удалении таблицы все связ. тр-ры тоже удал !  
Если удалена табл, на к-ю есть ссылка в теле тр-ра, то возвр ошибка, тр. надо переопр*

**SQLITE СТРУКТУРА**

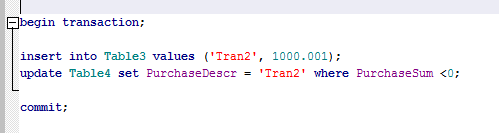
**внутр. таблицы SLITE:**  
\* SLITE\_MASTER – табл. с метаданными, индекс PK не показ. через нее; индекс по ROWID не виден в ней  
\* SLITE\_SEQUENCE – необх. для реализации AUTOINCREMENT; м. доб, изм, удал строки из таблицы, но нельзя удалять ее саму  
\* SLITE\_STATN

**Оптимизация запросов:**  
Исп. только для опер. анализа:  
\* **ANALYZE** – сбор статистики о т-цах и инд-сах, хран. в slite\_stat1; оптимизатор запросов м. получить доступ к инфе и исп. ее, чтобы помочь улучшить выбор планирования запросов

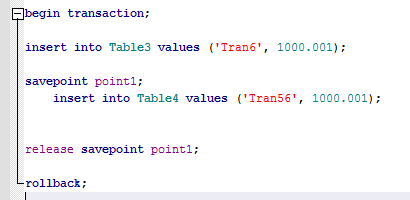
\* **EXPLAIN** – пошаговое исполн. оператора; оператор при этом не вып-ся  
\* **EXPLAIN QUERY PLAN** – показ, каким образом будет проводится поиск в таблице

**PRAGMA**  
\* index\_info (ind\_name) – возвр одну строку для к. столбца ключа в названном инд-се  
\* index\_list (tablename) – возвр одну строку для к. инд-са, связ. с данной таблицей  
\* index\_xinfo (ind\_name) – возвр инфу о к. столбце в инд-се

**SQLLITE ТРАНЗАКЦИИ**

\* к. оператор идет в неявной тр-ции с автокоммитом  
\* м. тр-ю задать явно  
  
\* разница между 127 вставками д-х в таблицк в неявных тр. и в одной явной тр:  
 *запрос успешно вып: commit; (заняло 106мс)  
 запрос успешно вып: insert into T2 values (‘a’,2): (заняло 155 мс, 1 строк изменено)*

\* rollback если бд закрылась (откат тр-и к началу или to *savepoint*)  
\* rollback если возник ошибка и алг. разрешения конфликтов rollback  
\* end transaction – commit

\* **SAVEPOINT** – имеет имя, необяз. влож. тр, необяз. в рамках begin tran / commit  


\* **Вложенные тр:** внешняя – begin tr / commit  
 внутр – savepoint / release savepoint