Определения

Важные пунктики

1 лекция:

**База данных** — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

Модель данных – абстракция, описывающая структуру (организацию) данных и методы их обработки.

Реляционная БД – набор плоских взаимосвязанных таблиц.

Основные понятия реляционной базы данных:

* Тип данных - характеристика объекта в языке программирования.
* Домен - множество отдельных допустимых значений
* Атрибут - столбцы таблицы
* Кортеж - строка таблицы
* Отношение - список имен атрибутов отношения с указанием имен типов.
* Первичный ключ - специальное дополнительное поле (атрибут) таблицы, которое создается для обеспечения уникальности идентификации записей таблицы

**Система управления базами данных(**СУБД) - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных

Операторы определения доступа к данным:

***-GRANT*** – предоставление пользователю (группе пользователей) разрешения на определенные действия в БД

***- REVOKE –*** отзыв ранее выданных разрешений

***- DENY –*** запрет определенных действий, имеющий приоритет над разрешением

Операторы управления транзакциями:

***-COMMIT*** – фиксация транзакции

***- ROLLBACK* –** откат изменений, сделанных в транзакции

***- SAVEPOINT –*** создание точки сохранения в транзакции

###### 2 лекция

Структура БД в SSMS:

* Diagrams
* Tables
* Views
* Synonyms
* Programmability
* Service Broker
* Storage
* Security

Структура таблицы в SSMS:

* Columns
* Primary key
* Foreign key
* Constraint
* Trigger
* Index
* Statistic

###### 3 лекция

Целостность БД – соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам.

Безопасность БД – защита БД от несанкционированного разрушения, изменения и получения данных.

Достоверность БД – соответствие фактов, хранящихся в базе данных, реальному миру.

Ограничение целостности:

* NULL / NOT NULL
* DEFAULT
* CHECK
* PRIMARY KEY
* FOREIGN KEY
* UNIQUE

###### 4 лекция

Транзакция – набор операций (изменений), который должен быть выполнен полностью или не выполнен совсем (единый логический блок).

Требования к системе транзакций:

* Атомарность: Никакая транзакция не может быть зафиксирована частично.
* Cогласованность: Транзакция переводит БД из одного согласованного (целостного) состояния в другое.
* Изолированность: Параллельные транзакции не должны влиять друг на друга.
* Устойчивость (долговечность):Изменения, сделанные завершенной транзакцией, должны остаться в БД независимо от сбоев в работе.

BEGIN TRANSACTION <t1>

…

SAVE TRANSACTION <t1>

…

COMMIT TRANSACTION <t1> / ROLLBACK TRANSACTION <t1>

Блокировка – отметка о захвате объекта транзакцией в ограниченный или исключительный доступ с целью предотвращения конфликтов и поддержания целостности данных.

Эскалация блокировок – замена множества низкоуровневых блокировок одной или несколькими блокировками более высокого уровня.

Блокирование (blocking) – процесс не может получить доступ к ресурсу (наложить блокировку), так как он (ресурс) заблокирован другим процессом.

Взаимоблокировка (deadlock) – процессы заблокированы друг другом (каждый ждет освобождения ресурса, занятого другим).

###### 6 лекция

Страница - основная (минимальная) единица хранения, которой манипулирует сервер. Размер – 8 килобайт.

Экстент – непрерывная область дисковой памяти, состоит из заданного числа страниц (в MS SQL Server – 8 страниц). Используется для эффективного управления страницами.

Секция – пользовательская единица организации данных (по умолчанию секция единственная).

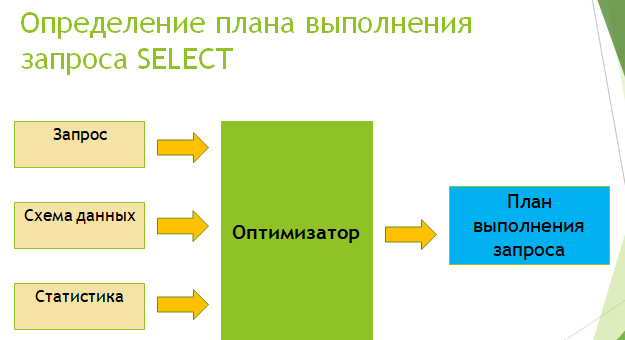
Индекс (INDEX) – вспомогательная структура данных, используемая для доступа к данным. Хранит указатели на места хранения записей. Создается для столбцов таблицы.

CREATE INDEX Index\_1 ON Table\_1 (Column\_1, Column\_2);

Модель востановления БД:

* **Модель восстановления (recovery model)** — это свойство базы данных, которое управляет процессом регистрации транзакций, определяет, требуется ли для журнала транзакций резервное копирование, а также определяет, какие типы операций восстановления доступны.
* **Простая модель восстановления (simple)**. Автоматически освобождает место на диске, занятое журналами, устраняя таким образом необходимость в управлении размером журналов транзакций.
* **Модель полного восстановления (full)**. Содержимое журнала хранится до выполнения его резервного копирования. Потеря результатов работы из-за повреждения файлов данных исключена.
* **Модель восстановления с неполным протоколированием (bulk-logged)**. Дополнение к полной модели полного восстановления, позволяющее выполнять высокопроизводительные операции массового копирования.  
  Уменьшает место, занимаемое журналами, за счет неполного протоколирования большинства массовых операций.

###### 7 лекция



Логический оператор - концептуальное выполнение операции реляционной алгебры

Физический оператор - практическая реализация действий, описанных логическим оператором.



###### 8 лекция

Резервное копирование:

1.Полное (архивная копия БД на момент сохранения)

BACKUP DATABASE db1 TO DISK=‘…’

2. Дифференциальное (архив данных, измененившихся после последнего полного копирования)

BACKUP DATABASE db1 TO DISK=‘…’ WITH DIFFERENTIAL

3. Журналов (архив списка транзакций с момента последнего резервного копирования (полного, дифференциального, журнала транзакций))

BACKUP LOG db1 TO DISK=‘…’

4. Файловое

BACKUP DATABASE db1 FILEGROUP=fg1 TO DISK=‘…’

Восстановление БД:

1.Полная копия:

RESTORE DATABASE db1 FROM DISK=‘…’ WITH NORECOVERY;

2. Разностная копия:

RESTORE DATABASE db1 FROM DISK=‘…’ WITH NORECOVERY;

3. Копии журналов:

RESTORE LOG db1 FROM DISK=‘…’ WITH NORECOVERY;

4…

RESTORE LOG db1 FROM DISK=‘…’ WITH RECOVERY;

###### 9 лекция

Объявление переменных

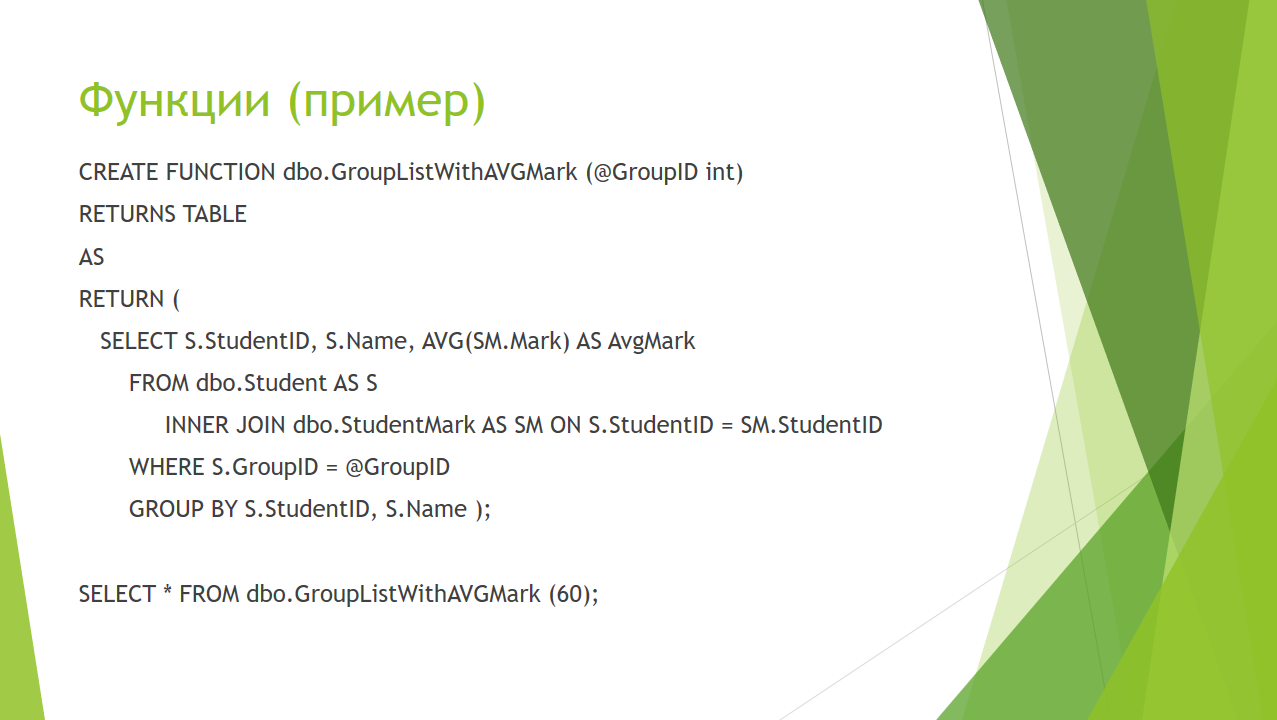
DECLARE @MyCounter int;

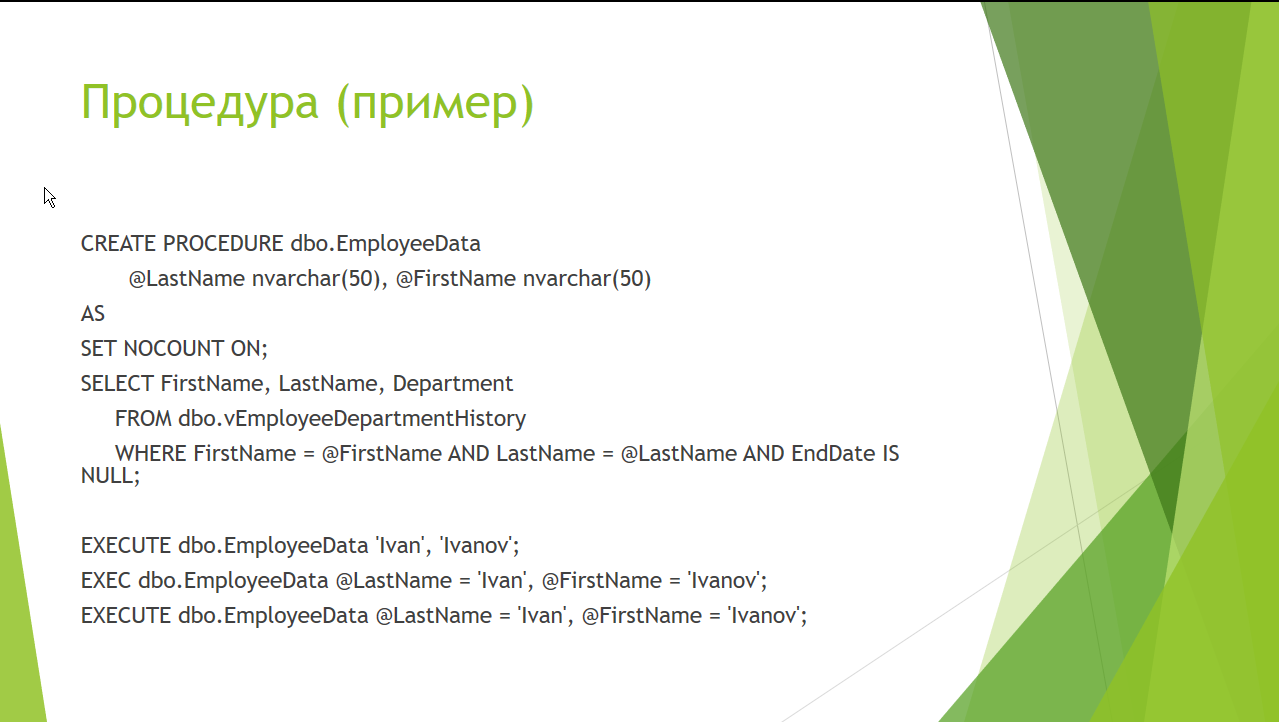
SET @MyCounter = 1;

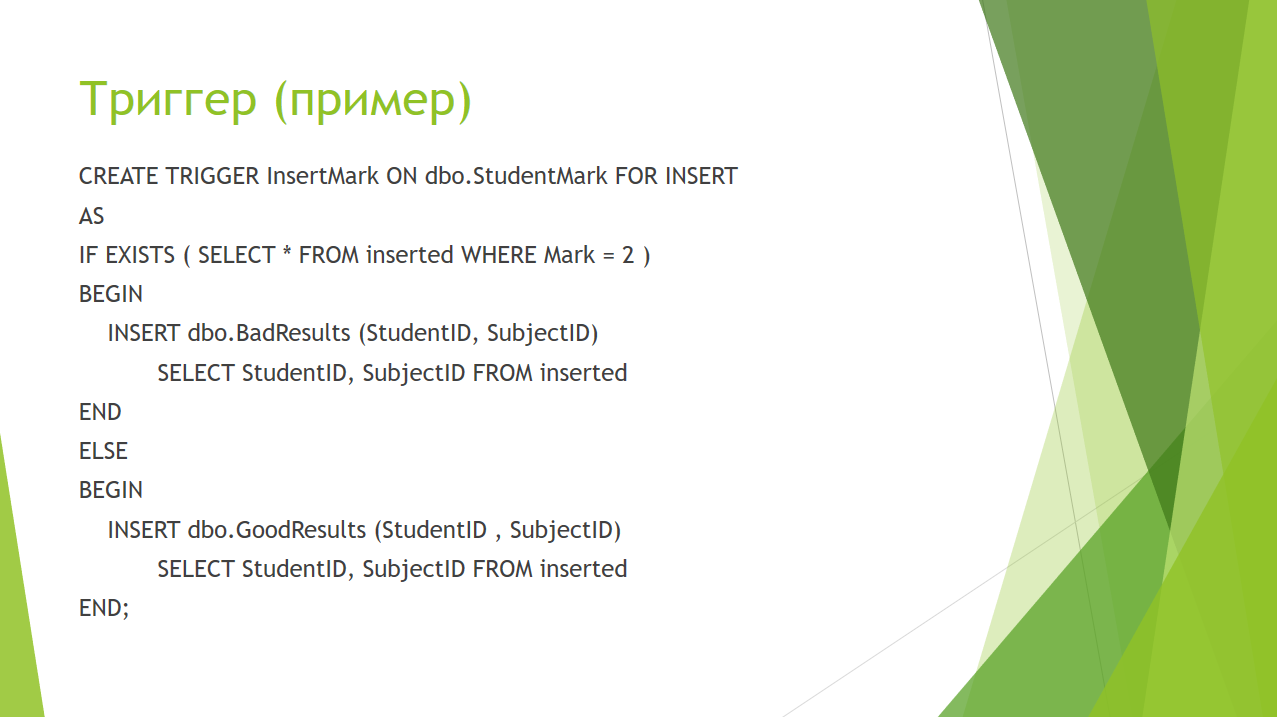
BEGIN { sql\_statement | statement\_block } END

IF Boolean\_expression { sql\_statement | statement\_block } [ ELSE { sql\_statement | statement\_block } ]

WHILE Boolean\_expression { sql\_statement | statement\_block | BREAK | CONTINUE }

****





Функции – подпрограммы, которые принимают параметры, выполняют действие и возвращают результат этого действия в виде значения.

Процедура - это группа инструкций Transact-SQL, которые обрабатывают входные параметры и возвращают вызывающей программе значения в виде выходных параметров; и содержат программные инструкции, которые выполняют операции в базе данных, включая вызов других процедур.

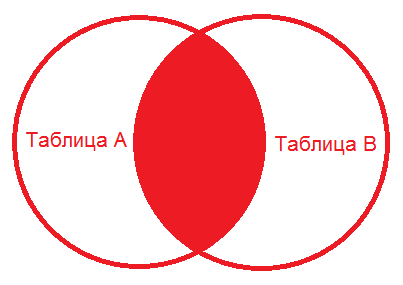
Триггеры DML — это хранимые процедуры особого типа, автоматически выполняемые, если происходит событие языка обработки данных DML, которое затрагивает таблицу или представление, определенное в триггере.

###### 10 лекция

****

### JOIN

**Внутреннее соединение** (**INNER JOIN**)



SELECT \* FROM 'Auto'

INNER JOIN 'Selling' ON 'Auto'.id = 'Selling'.id

И результат:

id name id sum

-- ---- -- ----

1 bmw 1 250

3 kia 3 300

**Внешнее соединение** (**OUTER JOIN**)



SELECT \* FROM 'Auto'

FULL OUTER JOIN 'Selling' ON 'Auto'.id = 'Selling'.id

Результат:

id name id sum

-- ---- -- ----

1 bmw 1 250

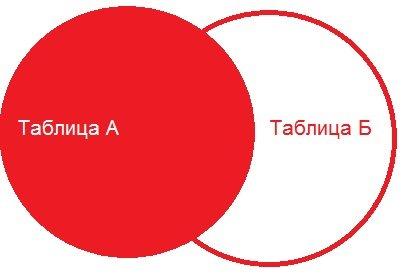
2 opel NULL NULL

3 kia 3 300

4 audi NULL NULL

NULL NULL 5 450

NULL NULL 6 400

**Левое внешнее объединение** (**LEFT OUTER JOIN**)

SELECT \* FROM 'Auto'

LEFT OUTER JOIN 'Selling' ON 'Auto'.id = 'Selling'.id

Результат:

id name id sum

-- ---- -- ----

1 bmw 1 250

2 opel NULL NULL

3 kia 3 300

4 audi NULL NULL

**CROSS JOIN** – это объединение SQL по которым каждая строка одной таблицы объединяется с каждой строкой другой таблицы.

***Умножение***

