

Запрос баз данных

Цели урока

В этом уроке вы узнаете о том, как:

- Определять свойства архитектуры клиент-сервер
- Определять требования ко входу запроса
- Планировать и создавать формат выхода запроса
- Определять и применять источники данных для запроса
- Определять и применять различные типы запросов
- Применять команду SELECT вместе с логическими операторами

Введение в SQL Server 2010

- Архитектура клиент-сервер
 - В модели клиент-сервер работа приложения разделяется на два процесса: процесс клиента и процесс сервера
 - Процесс клиента обрабатывает входные данные и вопросы интерфейса пользователя
 - Процесс сервера выполняет все проверки данных

Введение в SQL Server 2010 (продолжение)

- Преимущества 2-уровневой (клиент/сервер) модели:
 - совместное использование данных: данные разделяются со стороны клиента и хранятся в центральном месте, куда имеют доступ все пользователи
 - снижение дублирования и обслуживания: т.к. данные хранятся централизованно, упрощается обслуживание данных. нет дублирования данных, поэтому нет противоречивости хранимых данных

Запрос баз данных

Введение в SQL Server 2010 (продолжение)

- Свойства SQL Server от Microsoft
 - SQL Server от Microsoft предоставляет различные способы управления данными. Некоторыми свойствами являются:
 - полная совместимость с сетью WWW
 - значительная масштабируемость и надежность
 - кратчайший срок от начала разработки приложения до его выхода на рынок

Введение в SQL Server 2010 (продолжение)

- Язык структурированных запросов
 - для доступа к объектам данных от сервера, SQL Server представляет язык структурированных запросов (SQL)
 - В SQL Server от Microsoft добавлены функции синтаксиса базового уровня SQL и имеется свой SQL (Transact-SQL или T-SQL)
- Типы данных в SQL Server
 - SQL Server может хранить различные типы данных: char (символьный), money (денежный), int (целочисленный) или datetime (временной)

Запрос баз данных

Отображение особых атрибутов из таблицы

Формулировка задачи

Подробности приема на работу хранятся в базе RECRUITMENT на SQL сервере. Эти таблицы содержат данные о кандидатах, их квалификации и контактные данные, имеющиеся вакансии и др. информацию, требуемую кадровым агентством. Требуется отчет с именами, городами и телефонами всех внешних кандидатов.

Запрос баз данных

Решение

1. *Создать формат выхода запроса*
2. *Составить запрос*
3. *Выполнить запрос*
4. *Проверить, что выход запроса соответствует требуемым результатам*

Запрос баз данных

Создать формат выхода запроса

- *Результат:*
 - *Требуемый выход запроса, - это особые подробные данные обо всех внешних кандидатах*
 - *Заголовки столбцов, требуемые в отчете, - это имена, города и телефоны внешних кандидатов*
 - *Формат отчета см. ниже:*

<i>vFirstName</i>	<i>vLastName</i>	<i>cCity</i>	<i>cPhone</i>

Запрос баз данных

Составить запрос

- Команда *SELECT*
- Дает доступ и извлечение данных из базы

- Синтаксис :

```
SELECT [ALL | DISTINCT] список_выбираемых_столбцов  
[INTO [имя_новой_таблицы]]  
[FROM {имя_таблицы| имя_просмотра}{(подсказки_оптимизатора)}],  
{имя_таблицы2 | имя_просмотра2}{(подсказки_оптимизатора)}  
[WHERE выражение]  
[GROUP BY выражение][HAVING выражение]  
[ORDER BY выражение][COMPUTE выражение]  
[FOR BROWSE]
```

Составить запрос (продолжение)

- *Применение букв*
 - *Результирующее множество команды запроса данных можно сделать более удобочитаемым, вставив строку `literal` в список `SELECT`*
 - *Буквы заключаются в одинарные кавычки и печатаются точно так, как они написаны в списке `SELECT`*

Составить запрос (продолжение)

- *Результат:*
 - *Информация доступна в базе Recruitment (прием на работу)*
 - *Требуется информация, доступная в таблице ExternalCandidate (ВнешнийКандидат)*
 - *Особые атрибуты, требуемые из таблицы ВнешнийКандидат, - vFirstName (имя), vLastName (фамилия), cCity (город) и cPhone (телефон)*

Запрос баз данных

Выполнить запрос

- *Действие*
 - *Результирующее множество команды запроса данных можно сделать более удобочитаемым, вставив строку `literal` в список `SELECT`*
 - *Соединиться с сервером*
 - *Соединиться с базой*
 - *Метод 1: Выбрать базу (`RECRUITMENT`) из выпадающего списка баз `SQL Query Analyzer`*
 - *Метод 2: В окне `Query Analyzer` набрать:
`USE RECRUITMENT`*

Запрос баз данных

Выполнить запрос

- *В окне Query Analyzer набрать:*
SELECT vFirstName, vLastName, cCity, cPhone
FROM ExternalCandidate
- *Выполнить запрос:*
 - *нажать Execute в меню Query Menu на панели меню*
- или
 - *нажать Execute на панели инструментов*

Запрос баз данных

Проверить, что выход запроса соответствует требуемым результатам

- *Проверить, что:*
 - *Отображены требуемые столбцы*
 - *Отображены требуемые заголовки столбцов*

Запрос баз данных

Отображение особых столбцов с заголовками, определяемыми пользователем

Формулировка задачи

Анализируются расходы в процентах на контрактных кадровых агентов для приема кандидатов. Требуются имена контрактных кадровых агентов и расходы на них. Желательно, чтобы отчет содержал пользовательские заголовки в следующем формате:

<i>Recruiter Name</i>	<i>Hire Charge</i>

Запрос баз данных

Решение

- *Составить запрос*
- *Выполнить запрос*
- *Проверить, что выход запроса соответствует требуемым результатам*

Запрос баз данных

Составить запрос

- *Отображение столбцов с заголовками, определяемыми пользователем*
- *Синтаксис*
 - *Метод 1:*
SELECT заголовок_столбца=имя_столбца[,имя_столбца...]
FROM имя_таблицы
 - *Метод 2:*
SELECT имя_столбца заголовок_столбца [,имя_столбца...]
FROM имя_таблицы

Запрос баз данных

Составить запрос (продолжение)

- *Результат:*
 - *Информация доступна в таблице ContractRecruiter (контрактный агент)*
 - *Требуемые столбцы – cName (имя) и siPercentageCharge (расходы в процентах) с заголовками 'Recruiter Name' (имя агента) и 'Hire Charge' (расходы) из таблицы.*

Запрос баз данных

Выполнить запрос

- *Действие :*
 - *В окне Query Analyzer набрать:*
SELECT 'имя агента' = cName, 'расходы' =
siPercentageCharge
FROM ContractRecruiter
 - *Выполнить запрос*

Запрос баз данных

Проверить, что выход запроса соответствует требуемым результатам

- *Проверить, что:*
 - *Отображены требуемые столбцы*
 - *Отображены требуемые заголовки столбцов*

Запрос баз данных

Отображение выбранных Строк из таблицы

Формулировка задачи

Прием из студенческого городка на работу в компанию Tebisco должен проводиться в штате Калифорния. Т.е. требуются подробности обо всех университетах штата Калифорния

Запрос баз данных

Решение

- *Создать формат выхода запроса*
- *Составить запрос*
- *Выполнить запрос*
- *Проверить, что выход запроса соответствует требуемым результатам.*

Запрос баз данных

Создать формат выхода запроса

- *Результат:*
 - *Требуемый выход – подробности обо всех университетах штата Калифорния*
 - *Формат отчета:*

<i>cCollegeCode</i>	<i>cCollegeName</i>	<i>vCollegeAddress</i>	<i>cCity</i>	<i>cState</i>	<i>cZip</i>	<i>cPhone</i>

Запрос баз данных

Составить запрос

- *Команда SELECT ... WHERE*
 - *Извлекает и отображает данные при особом условии*

Синтаксис

SELECT список_столбцов

FROM имя_таблицы

WHERE условие_поиска

Запрос баз данных

Составить запрос (продолжение)

- *Результат:*
 - *Информация доступна в таблице College (университет)*
 - *Заголовки столбцов – имена атрибутов таблицы College*
 - *Условие таково, что штатом является 'California', т.е. cState='California'*
 - *Т.е. запрос, использующий команду SELECT, должен быть следующим:*
*SELECT * FROM College*
WHERE cState = 'California'

Запрос баз данных

Выполнить запрос

- *Действие :*
 - *В окне Query Analyzer набрать запрос.*
 - *Выполнить запрос*

Запрос баз данных

Проверить, что выход запроса соответствует требуемым результатам

- *Проверить, что:*
 - *Отображены все столбцы*
 - *Отображены только строки таблицы, в которых значением атрибута cState является California*

Арифметические операторы

- *SQL Server поддерживает арифметические операции сложения, вычитания, деления и умножения числовых столбцов.*
- *Арифметические операторы, поддерживаемые SQL Server:*
 - *+* (сложение)
 - *-* (вычитание)
 - */* (деление)
 - *** (умножение)
 - *%* (по модулю)

Арифметические операторы (продолжение)

- *Некоторыми правилами применения арифметических операторов являются:*
 - *арифметические операции могут выполняться над числовыми столбцами или числовыми константами*
 - *оператор "по модулю" (%) не может выполняться над столбцами следующих типов данных: money (денежный), smallmoney (мелкие деньги), float (плавающая точка) или real (действительное число)*

Арифметические операторы (продолжение)

- Приоритет оператора
 - Если в одном запросе применяются несколько арифметических операторов, они выполняются согласно их приоритету
 - В выражении приоритет умножения (*), деления (/), модуля (%) выше, чем вычитания (-) и сложения (+)
 - Приоритет изменяется объектом первичной группировки - круглыми скобками (())

Запрос баз данных

Поиск по условиям

- SQL Server дает несколько методов поиска строк таблицы. В общих чертах эти методы можно разделить на следующие:
 - Логические операторы
 - Операторы сравнения
 - Операторы диапазона
 - Операторы списка

Отображение Строк, удовлетворяющих нескольким условиям

Формулировка задачи

Кандидаты с оценками между 80 и 100 по испытанию, должны быть вызваны на собеседование. Т.е. требуется отчет, отображающий имена кандидатов и их очки.

Запрос баз данных

Решение

1. *Создать формат выхода запроса*
2. *Составить запрос*
3. *Выполнить запрос*
4. *Проверить, что выход запроса соответствует требуемым результатам*

Запрос баз данных

Создать формат выхода запроса

- *Результат:*
 - *Выходные требования к отчету – имена и оценки по испытанию только тех кандидатов, оценки по испытанию которых лежат между 80 и 100*
 - *Формат отчета:*

<i>vFirstName</i>	<i>vLastName</i>	<i>siTestScore</i>

Запрос баз данных

Составить запрос

- *Логический оператор - несколько условий поиска могут сочетаться следующими логическими операторами:*
 - *OR - дает результат, если выполняются любое из данных условий*
 - *AND - дает результат, если выполняются все данные условия*
 - *NOT - нейтрализует выражение перед собой*
- *Если более одного логического оператора сочетаются с предложении WHERE, порядок выполнения следующий: NOT, AND и OR*
- *Круглые скобки можно применить для изменения логического порядка выполнения*

Запрос баз данных

Составить запрос (продолжение)

- *Синтаксис*

SELECT список_столбцов

FROM имя_таблицы

WHERE условное_выражение{AND/OR} [NOT]
условное_выражение

Запрос баз данных

Составить запрос (продолжение)

- *Результат:*
 - *Требуемая информация доступна в таблице ExternalCandidate*
 - *Заголовки столбцов – имена атрибутов соответствующих столбцов таблицы ExternalCandidate*
 - *По условию, которое должно выполняться, оценка теста должна быть выше или равна 80 ($siTestScore \geq 80$) и ниже или равна 100 ($siTestScore \leq 100$)*
 - *Т.е. запрос, использующий команду SELECT, должен быть следующим:*
SELECT vFirstName, vLastName, siTestScore
FROM ExternalCandidate
WHERE siTestScore >=80 AND siTestScore <=100

Запрос баз данных

Выполнить запрос

Действие :

- *В окне Query Analyzer набрать запрос.*
- *Выполнить запрос*

Запрос баз данных

Проверить, что выход запроса соответствует требуемым результатам

- *Проверить, что:*
 - *Отображены требуемые столбцы*
 - *Все отображаемые строки имеют значения siTestScore от 80 до 100*

Дополнительные сведения об операторах

- Кроме логических операторов, SQL Server поддерживает следующие операторы:
 - Операторы сравнения
 - Операторы диапазона
 - Операторы списка

Арифметические операторы (продолжение)

- Операторы сравнения
 - Операторы сравнения – позволяют извлекать строку из таблицы по условию, заданному в выражении WHERE
 - Синтаксис

SELECT список_столбцов FROM имя_таблицы

WHERE выражение1 оператор_сравнения выражение2

Арифметические операторы (продолжение)

- Операторы диапазона
 - Оператор диапазона применяется для извлечения данных, которые могут выделяться в диапазоны. Операторы диапазона следующие:
 - BETWEEN
 - NOT BETWEEN
 - Синтаксис

SELECT список_столбцов FROM имя_таблицы

WHERE выражение1 оператор_диапазона выражение2 AND выражение3

Арифметические операторы (продолжение)

- Операторы списка
 - Оператор IN позволяет выбрать значения, соответствующие любому из значений в списке
 - Оператор NOT IN ограничивает выбор значений, соответствующих любому из значений в списке
 - Синтаксис

SELECT список_столбцов FROM имя_таблицы

WHERE выражение оператор_списка ('список_значений')

Запрос баз данных

Выводы

Из этого урока Вы узнали, что:

- Система управления базами данных состоит из Сервера, базы данных (или нескольких баз) с таблицами, содержащими данные, и клиента (интерфейсная часть), помогающего пользователю взаимодействовать с сервером для извлечения данных.
- Язык, предусмотренный SQL сервером для доступа к данным из базы, называется языком структурированных запросов (SQL).
- Microsoft SQL Server обеспечивает гибкую реализацию языка SQL, называемого T-SQL.

Выводы (продолжение)

- SQL Server предусматривает команду SELECT для доступа и извлечения данных из базы. Команда SELECT запрашивает сервер подготовить результат и вернуть его в приложение-клиент.
- Команду SELECT можно применять для извлечения особых столбцов из таблицы, задавая имена столбцов из таблицы.
- Команда SELECT вместе со звездочкой (*) дает результат, содержащий подробную информацию о всех столбцах таблицы.

Выводы (продолжение)

- Порядок столбцов можно изменить в результирующем множестве оператора `SELECT`, задав через запятую имена отдельных столбцов.
- SQL Server дает 2 метода задания заголовка столбца. В 1-м методе заголовок столбца задается до имени столбца, а во 2-м – наоборот.
- SQL сервером предусмотрено предложение `WHERE` для задания условия извлечения особых данных.
- Результирующее множество оператора запроса данных можно сделать более удобочитаемым, вставив строку в список `SELECT`.

Выводы (продолжение)

- SQL Server поддерживает операторы, выполняющие арифметические операции сложения, вычитания, деления и умножения числовых столбцов.
- В случае сочетания арифметических операций, значение данных нижнего типа преобразуется в значение высокого типа согласно приоритету типа данных.
- SQL Server дает следующий набор операторов:
 - логические операторы AND, OR и NOT
 - Операторы сравнения =, >, <, >=, <=, !=, !< и !>
 - Операторы диапазона BETWEEN и NOT BETWEEN
 - Операторы списка IN и NOT IN