

ПРАКТИЧЕСКАЯ РАБОТА по теме:

Создание запросов на выборку данных средствами SQL

Построение простых запросов баз данных

Простые, или так называемы прямые запросы, строить проще всего. Именно они максимально понятны, точки зрения логики. Как правило, такого рода запросы делаются с одной или нескольких, объединённых таблиц и имеют очень простую структуру. Для примера можно рассмотреть три таблицы, а именно: Counterfoil – условия оплаты, Person – сотрудники и таблица Depart – отделы. Из этих таблиц, скажем в первом случае необходимо просто выбрать всю имеющуюся в них информацию. Для этого нужно выполнить скрипт такого плана:

- `Select * from Counterfoil`
- `Inner join Person on PE_ID = Cou_Person`
- `Order by PE_ID`

Данный скрипт вернёт пользователю все данные из таблиц упорядоченный по сотрудникам. Это самая простая конструкция, которую, при необходимости можно усложнить условиями. Использование условий обозначается служебным словом `Where`, после которого можно указывать, те поля таблицы, к которым необходимо применить фильтры.

Структура сложных (вложенных) запросов баз данных

Простые запросы не всегда эффективны. Если в таблицах много полей и записей, то выборка может продолжаться от нескольких секунд, до нескольких минут, что не совсем целесообразно. Задача любого программиста баз данных – это максимальная оптимизация работы запросов, не только для получения максимально корректной информации, но и ускорение времени их выполнения. Как раз для этих целей и существуют так называемые вложенные запросы.

Как правило, в таких запросах идёт выборка не по всем полям, а только, по тем, которые необходимы. Можно рассмотреть такой пример. Необходимо вывести номер условия оплаты, сотрудника для которого производится оплата и отдел, где сотрудник работает. Для этого можно воспользоваться вложенным скриптом, который вернёт не все, а только нужные поля таблицы. Такой скрипт будет иметь следующий вид:

- Номер условия
- Наименование
- Код сотрудника

- Сотрудник
- Код отдела

В конце скрипта поставлен фильтр, который выведет информацию только о работающих, на данный момент, сотрудниках компании.

Скрипты на выборку данных с базы могут быть ещё сложнее, но общая структура запросов будет полностью похожа на выше описанные.

Ход работы:

1) База данных "Company 1"

На панели инструментов SQL Server Management Studio нажмите "Создать запрос". Откроется окно запросов. Скопируйте в окно запросов следующий код для создания базы данных, её таблицы и заполнения таблицы данными:

```
CREATE DATABASE company1; GO USE company1; CREATE TABLE Org ( ID INT PRIMARY KEY
IDENTITY, Deptnumb INT, Deptname VARCHAR(30), Manager INT, Division VARCHAR(30),
Location VARCHAR(30) UNIQUE (Deptnumb) ); INSERT INTO Org (Deptnumb, Deptname,
Manager, Division, Location) VALUES (10, 'Head Office', 160, 'Corporate', 'New York'), (15, 'New
England', 50, 'Eastern', 'Boston'), (20, 'Mid Atlantic', 10, 'Eastern', 'Washington'), (38, 'South
Atlantic', 30, 'Eastern', 'Atlanta'), (42, 'Great Lakes', 100, 'Midwest', 'Chicago'), (51, 'Plains', 140,
'Midwest', 'Dallas'), (66, 'Pacific', 270, 'Western', 'San Francisco'), (84, 'Mountain', 290,
'Western', 'Denver'); CREATE TABLE Staff ( ID INT PRIMARY KEY IDENTITY (10, 10), Name
VARCHAR(30) NOT NULL, Dept INT, Job VARCHAR(30), Years INT, Salary FLOAT, Comm FLOAT
FOREIGN KEY (Dept) REFERENCES Org (Deptnumb) ON DELETE CASCADE ); INSERT INTO Staff
(Name, Dept, Job, Years, Salary, Comm) VALUES ('Sanders', 20, 'Mgr', 7, 18357.50, NULL),
('Pernal', 20, 'Sales', 8, 18171.25, 612.45), ('Marengi', 38, 'Mgr', 5, 17506.75, NULL), ('O'Brien',
38, 'Sales', 6, 18006.00, 846.55), ('Hanes', 15, 'Mgr', 10, 20659.80, NULL), ('Quigley', 38, 'Sales',
NULL, 16808.30, 650.25), ('Rothman', 15, 'Sales', 7, 16502.83, 1152.00), ('James', 20, 'Clerc',
NULL, 13504.60, 128.20), ('Koonitz', 42, 'Sales', 6, 18001.75, 1386.70), ('Plotz', 42, 'Mgr', 7,
18352.80, NULL), ('Ngan', 15, 'Clerc', 5, 12508.20, 206.60), ('Naughton', 38, 'Clerc', NULL,
12954.75, 180.00), ('Yamaguchi', 42, 'Clerc', 6, 10505.90, 75.60), ('Fraye', 51, 'Mgr', 6, 21150.00,
NULL), ('Williams', 51, 'Sales', 6, 19456.50, 637.65), ('Molinare', 10, 'Mgr', 7, 22959.20, NULL),
('Kermisch', 15, 'Clerc', 4, 12258.50, 110.10);*
```

* Код читается, просто не структуризирован *

Оператор языка SQL SELECT - основная строительная конструкция для создания любого, простого или сложного запроса к базе данных. Без него, как и без фундамента для постройки, невозможно получить ни одну выборку данных из базы. На этом уроке мы узнаем, как построить запрос для получения

- простой выборки данных без условий (выбор всех строк во всех столбцах или всех строк в определённых столбцах);
- выборки данных с одним или несколькими условиями (выбор определённых строк), которые заданы в секции WHERE с помощью предикатов и дополнительных операторов;
- как использовать оператор SELECT в подзапросах

2) Итак, есть база данных фирмы - Company1. В ней есть таблица Org (Структура фирмы) и Staff (Сотрудники). Требуется выбрать из таблиц все столбцы. Соответствующий запрос для выбора всех столбцов из таблицы Org выглядит следующим образом (на MS SQL Server - с предваряющей конструкцией USE company1;):

```
SELECT * FROM Org
```

3) Запрос для выбора всех столбцов из таблицы Staff выглядит следующим образом (на MS SQL Server - с предваряющей конструкцией USE company1;):

```
SELECT * FROM Staff
```

*Для выбора определённых столбцов таблицы нам потребуется вместо звёздочки перечислить через запятую названия всех столбцов, которые требуется выбрать:

```
SELECT ВЫБИРАЕМЫЕ_СТОЛБЦЫ FROM ИМЯ_ТАБЛИЦЫ*
```

4) Пусть требуется из таблицы Org выбрать столбцы Deptnumb и Deptname, в которых содержатся данные соответственно о номерах отделов фирмы и об их названиях. Запрос для получения такой выборки будет следующим (на MS SQL Server - с предваряющей конструкцией USE company1;):

```
SELECT Deptnumb, Deptname, FROM Org
```

А из таблицы Staff нужно выбрать столбцы Dept, Name, Job, в которых содержатся соответственно данные о номере отдела, в котором трудится сотрудник, его имени и должности (на MS SQL Server - с предваряющей конструкцией USE company1;):

```
SELECT Dept, Name, Job FROM Staff
```

* Для выбора определённых строк таблицы вместе с оператором SELECT уже потребуется ключевое слово WHERE, указывающее на некоторое

значение или несколько значений, содержащиеся в интересующих нас строках. Наиболее простые условия задаются при помощи операторов сравнения и равенства (<, >, =), а также ключевого слова IS. Условий может быть несколько, тогда они перечисляются с использованием ключевого слова AND. Запросы для выбора строк имеют следующий синтаксис:

SELECT ИМЯ_СТОЛБЦА FROM ИМЯ_ТАБЛИЦЫ WHERE УСЛОВИЕ

5) Выберем из таблицы Staff строки, в которых содержатся данные только о сотрудниках, которые работают в 38-м отделе (на MS SQL Server - с предваряющей конструкцией USE company1;):

SELECT Dept, Name, Job FROM Staff WHERE Dept=38

6) В предыдущем примере мы выбирали строки из таблицы только по значению одного столбца - DEPT. Пусть теперь нужно выбрать данные о сотрудниках, которые работают в 38-м отделе и должность которых - служащий (Clerk). Для этого в секции WHERE соответствующие значения нужно перечислить с использованием слова AND (на MS SQL Server - с предваряющей конструкцией USE company1;):

SELECT Dept, Name, Job FROM Staff WHERE Job='Clerk' AND DEPT=38

7) Пусть нужно выбрать из таблицы Staff идентификаторы и имена тех сотрудников, размер комиссии которых - неопределённый. Для этого в секции WHERE перед указанием значения столбца Comm - NULL нужно ставить не знак равенства, а слово IS (на MS SQL Server - с предваряющей конструкцией USE company1;):

SELECT ID, Name FROM Staff WHERE Comm IS NULL

8) Выберем из таблицы имена, размеры заработной платы и число лет, проработанных в фирме, сотрудников, которые работают в фирме более девяти лет (на MS SQL Server - с предваряющей конструкцией USE company1;):

SELECT Name, Salary, Years FROM Staff WHERE Years>9

Задания:

Проделать операции по ходу работы.

Сделать скриншоты.

Прислать работу на почту преподавателя.