

Базы данных

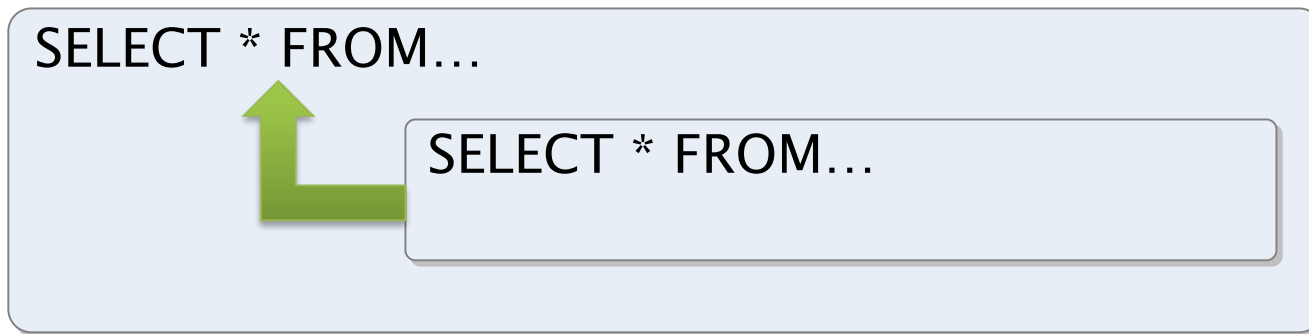
06 | Использование подзапросов и
предложения APPLY

План

- Введение в подзапросы
- Скалярные и многозначные подзапросы
- Замкнутые (self-contained) или коррелированные (correlated)?
- Использование предложения APPLY с табличными функциями

Введение в подзапросы

- Подзапросы являются вложенными запросами: запросы в запросах
- Результаты вложенного запроса передаются внешнему запросу
 - Вложенный запрос выступает в роли выражения с точки зрения внешнего запроса



Скалярные или многозначные?

- Скалярный подзапрос возвращает одно значение во внешний запрос
 - Можно использовать везде, где можно использовать однозначные выражения: SELECT, WHERE и т.д.
- Многозначные подзапросы возвращают многозначные значения в виде одного столбца внешнему запросу
 - Используется с предикатом IN

```
SELECT orderid, productid, unitprice, qty  
FROM Sales.OrderDetails  
WHERE orderid =  
      (SELECT MAX(orderid) AS lastorder  
        FROM Sales.Orders);
```

```
SELECT custid, orderid  
FROM Sales.orders  
WHERE custid IN (  
      SELECT custid  
        FROM Sales.Customers  
        WHERE countryregion = N'Mexico');
```

DEMO

Использование подзапросов

Замкнутые или коррелированные?

- Большинство подзапросов замкнуты, т.е. не имеют взаимосвязи с внешним запросом помимо передачи результата
- Коррелированные подзапросы ссылаются на сущности (столбцы таблиц) внешнего запроса
 - Зависят от внешнего запроса, нельзя исполнять отдельно
 - Поведение подзапроса: один запуск для каждой строки внешнего запроса
 - Могут возвращать скалярное или множественное значение

```
SELECT orderid, empid, orderdate
FROM Sales.Orders AS O1
WHERE orderdate = (SELECT MAX(orderdate)
                  FROM Sales.Orders AS O2
                  WHERE O2.empid = O1.empid)
ORDER BY empid, orderdate;
```

DEMO

Создание коррелированного запроса

Использование предложения APPLY с табличными функциями

- CROSS APPLY применяет табличное выражение справа к каждой строке в левой таблице
 - Концептуально похоже на CROSS JOIN между двумя таблицами, но может сопоставлять данные между источниками

```
SELECT S.supplierid, s.companyname, P.productid, P.productname, P.unitprice  
FROM Production.Suppliers AS S  
CROSS APPLY dbo.fn_TopProductsByShipper(S.supplierid) AS P
```

- OUTER APPLY добавляет строки со значениями NULL для правой таблицы
 - Концептуально похоже на LEFT OUTER JOIN между двумя таблицами

DEMO

Использование APPLY с табличными выражениями

Изучено

- Введение в подзапросы
- Скалярные и многозначные подзапросы
- Замкнутые (self-contained) или коррелированные (correlated)?
- Использование предложения APPLY с табличными функциями