Chapters

Queries

A query is a formalized instruction to a database to either return a set of records or perform a specified action on a set of records as specified in the query. For example, the following SQL query statement returns records:

SELECT [Company Name] FROM Publishers WHERE State = "NY"

Queries are expressed with Structured Query Language (SQL) statements like the one shown above. The most common query type is the **SELECT** query.

Some Common SQL Keywords

SQL keyword	**Class Wizard and database classes use it ...**
SELECT	To identify which tables and columns in the data source are to be used.
WHERE	To apply a filter that narrows the selection.
ORDER BY	To apply a sort order
INSERT	To add new records
DELETE	To delete records
UPDATE	To modify the fields of a record.

SQL

Structured Query Language (SQL) is a way to communicate with a relational database that lets you define, query, modify, and control the data. Using SQL syntax, you can construct a statement that extracts records according to criteria you specify.

SQL statements begin with a keyword “verb” such as **CREATE** or **SELECT**. SQL is a very powerful language; a single statement can affect an entire table.

• Simple Queries in SQL Server

Select query

A query that returns rows into a result set from one or more tables. A Select query can contain specifications for those columns to return, the rows to select, the order to put the rows in, and how to group (summarize) information.

Examples

1. Select * from Com_m_Customer where Cust_Code='C50239'
2. Select * from td_t_pay_in_slip where slip_no='036954465'

Insert query

A query that copies specific columns and rows from one table to another or to the same table.

Examples

1. Insert into td_t_ac_open_cust (branch_code, appl_no, cust_code, authorised)
Values ('01','0321456','C50236','Y')
2. Insert into td_t_ac_open_cust (branch_code, appl_no, cust_code, authorised)
Select branch_code, appl_no, cust_code, authorised from td_t_ac_open_hd where Comp_Name = 'Server2K'

Update query

A query that changes the values in columns of one or more rows in a table.

Examples

1. Update td_party_ledger_hd set Post_status = 0 where ac_no= '23569' and ac_type = 'FD' and branch_code='01'
2. Update JV_Main set Approval_code ='G' where Voc_no='03569875'

Delete query

A query that removes rows from one or more tables.

Examples

1. Delete ac_t_journal_dtl where Voc_no='0369785' and branch_code = '01'
2. Delete from ac_t_journal_dtl where Voc_no='0369785' and branch_code = '01'

• Advanced Queries in SQL Server

Subquery

A subquery is a SELECT query that returns a single value and is nested inside a SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery. A subquery can be used anywhere an expression is allowed. In this example a subquery is used as a column expression named **MaxUnitPrice** in a SELECT statement.

SELECT Ord.OrderID, Ord.OrderDate,(SELECT MAX(OrdDet.UnitPrice) FROM Northwind.dbo.[Order Details] AS OrdDet WHERE Ord.OrderID = OrdDet.OrderID) AS MaxUnitPrice FROM Northwind.dbo.Orders AS Ord

A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.

Many Transact-SQL statements that include subqueries can be alternatively formulated as joins. Other questions can be posed only with subqueries. In Transact-SQL, there is usually no performance difference between a statement that includes a subquery and a semantically equivalent version that does not. However, in some cases where existence must be checked, a join yields better performance. Otherwise, the nested query must be processed for each result of the outer query to ensure elimination of duplicates. In such cases, a join approach would yield better results. This is an example showing both a subquery SELECT and a join SELECT that return the same result set:

/* SELECT statement built using a subquery. */

SELECT ProductName FROM Northwind.dbo.Products WHERE UnitPrice = (SELECT UnitPrice FROM Northwind.dbo.Products WHERE ProductName = 'Sir Rodney''s Scones')

/* SELECT statement built using a join that returns the same result set. */

SELECT Prd1.ProductName FROM Northwind.dbo.Products AS Prd1 JOIN Northwind.dbo.Products AS Prd2 ON (Prd1.UnitPrice = Prd2.UnitPrice) WHERE Prd2.ProductName = 'Sir Rodney''s Scones'

A subquery nested in the outer SELECT statement has the following components:

• A regular SELECT query including the regular select list components.
• A regular FROM clause including one or more table or view names.
• An optional WHERE clause.
• An optional GROUP BY clause.
• An optional HAVING clause.

Subquery Rules

A subquery is subject to a number of restrictions:

• The select list of a subquery introduced with a comparison operator can include only one expression or column name (except that EXISTS and IN operate on SELECT * or a list, respectively).
• If the WHERE clause of an outer query includes a column name, it must be join-compatible with the column in the subquery select list.
• The **ntext**, **text** and **image** data types are not allowed in the select list of subqueries.
• Because they must return a single value, subqueries introduced by an unmodified comparison operator (one not followed by the keyword ANY or ALL) cannot include GROUP BY and HAVING clauses.

- The DISTINCT keyword cannot be used with subqueries that include GROUP BY.
- The COMPUTE and INTO clauses cannot be specified.
- ORDER BY can only be specified if TOP is also specified.
- A view created with a subquery cannot be updated.

Unions

The UNION operator allows you to combine the results of two or more SELECT statements into a single result set. The result sets combined using UNION must all have the same structure. They must have the same number of columns, and the corresponding result set columns must have compatible data types

UNION is specified as:

Select_statement UNION [ALL] *select_statement*

Any number of UNION operators can appear in a Transact-SQL statement, for example:

SELECT * FROM TableA UNION SELECT * FROM TableB UNION SELECT * FROM TableC UNION SELECT * FROM TableD

Some Complex Queries

1. Selection of records from more than 1 table

- Select com_m_customer.cust_Name from com_m_cust_ac, com_m_customer where com_m_cust_ac. Cust_Code=com_m_customer. Cust_Code and com_m_cust_ac.ac_no = '235689'

2. Locking of Tables

- Select * from ac_t_journal_hd (ROWLOCK HOLDLOCK) where Voc_No ='0356985'

3. Selection of a string in the field

- Select max (voc_no) from ac_t_journal_hd WHERE SUBSTRING (VOC_NO, 1,2) ='03' and branch_code = '01'

4. Updation of a table by linking with another table

- Update period_pay_table set pay_name=p.name, one_time_status=p.one_time, pay_category=p.category, pay_code=p.code, type='PC', tax_implicable=p. Tax_Imp from period_pay_table pt inner join pay_types p on pt.pay_id=p.id
- Update period_pay_table set emp_code=e.emp_no, emp_name=e.emp_name, stop_salary_status=e.stop_status, emp_branch=e.branch, resignation_status=e.resig_status from period_pay_table pt inner join employees e on pt.emp_id=e.id

5. Storing the result of addition or subtraction or one or more record in a single row

- Update Staff_Table set Col2 = (Select Sum (amt) from Tempd_pay_Table where Pay_Code = 'BAS' and Pay_Category = 'ALLOWANCE' and Period = 'AUG2003' and Staff_Table.Gen_Id = 2 and Staff_Table.Col1 = Tempd_pay_Table .emp_id) where Gen_Id = 2 and Comp_Name = 'G'
- Update Staff_Table set Col3 = Col3 - (Select Sum (amt) from Tempd_pay_Table where Pay_Code = 'DA' and Pay_Category = 'ALLOWANCE' and Period = 'AUG2003' and Staff_Table.Gen_Id
= 2 and Staff_Table.Col1 = Tempd_pay_Table .emp_id) where Gen_Id = 2 and Comp_Name = 'G'