# QUERIES

**A query specifies a result table. A query is a component of certain SQL statements. There are three forms of a query:**

- **A subselect**
- **A full-select**
- **A select-statement**

**A subselect is a subset of a fullselect and a fullselect is a subset of a select-statement.**

**There is another SQL statement called SELECT INTO.**

Figure 2.1 Queries

# SUBSELECT

**The subselect is a component of the fullselect, the CREATE VIEW statement and the INSET statement.**

**It is also a component of certain predicates, which in turn are components of a subselect.**

**A subselect that is a component of a predicate is called a sub query.**

**A subselect specifies a result table derived from the result of its first FROM clause.**

**The derivation can be described as a sequence of operations in which the result of each operation in which the result of each operation is input for the next.**

**The sequence of the operation is**

1. **FROM clause**
2. **WHERE clause**
3. **GROUP BY clause**
4. **HAVING clause**
5. **SELECT clause.**

Figure 2.2 Subselect

# SELECT-CLAUSE

**The SELECT clause specifies the columns of the final result table.**

**The column values are produced by the application of the select list.**

**The select list is a list of names and expressions specified in the SELECT clause.**

Figure 2.3 SELECT-clause

**Notes:**

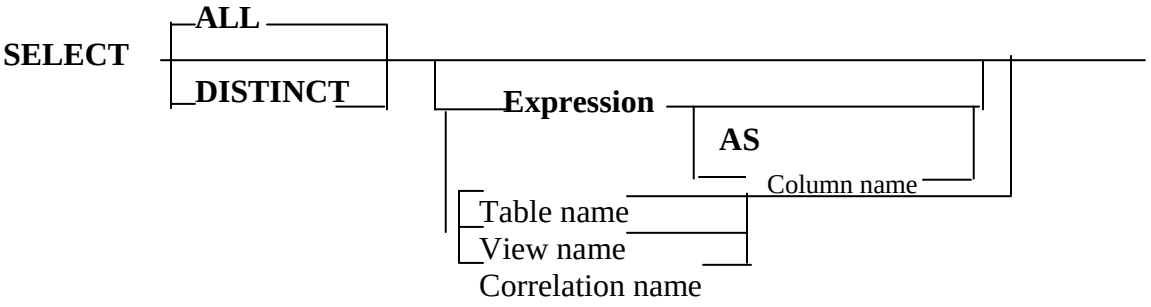
ALL
Retains all rows of the final result table and does not eliminate redundant duplicates. This is the default.

DISTINCT
Eliminates all but one of each set of duplicate rows of the final result table. DISTINCT must not be used more than once in a subselect, with the exception of its use with a column function whose expression is a column.

The same DISTINCT column function with the same column expression can be referred to more than once in a subselect.

# SELECT-CLAUSE (contd….)

## Syntax:




## ALL
## DISTINCT
## DUPLICATE ROWS
## AS column name

Figure 2.4 SELECT-clause

**Notes:**

This restriction includes SELECT DISTINCT and the use of DISTINCT in a column function of the select list or HAVING clause.

It does not include occurrences of DISTINCT in subqueries of the subselect.

Duplicate rows
Two rows are duplicates of one another only if each value in the first row is equal to the corresponding value in the second row. For determining duplicate rows, two null values are considered equal.

AS column name
Names or renames the result column. The name must not be qualified and does not have to be unique.

# FROM-CLAUSE

**The FROM clause specifies an intermediate result table, R.**

**If a single table-spec is specified, R is the result of that table-space.**

**If more than one table-spec is specified, R consists of all possible combinations of the rows of the result of each table-spec concatenated with a row from the result of the second table-spec, concatenated with a row from the result of the third table-spec, and so on.**

**The number of rows in R is the product of the number of rows in the result of each table-spec.**

**Thus if the result of any table-spec is empty, R is empty.**

**Syntax:**

**FROM table-spec**

**Table spec**

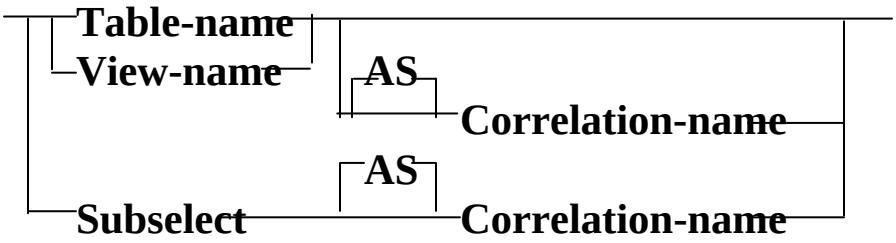



Figure 2.5 FROM-clause

# WHERE-CLAUSE

**The WHERE clause specifies an intermediate result table that consists of those rows of R when the search condition is true.**

**R is the result of the FROM clause of the subselect.**

**Syntax:**

**WHERE search condition**

Figure: 2.6 WHERE-clause

**Notes:**

The search condition must conform to the following rules:

Each column name must unambiguously identify a column of R or be a correlated reference.

A column name is a correlated reference if it identifies a column of a table or view identified in an outer subselect.

A column function must not be specified unless the WHERE clause is specified in a subquery of a HAVING clause and the argument of the function is a correlated reference to a group.

# GROUP-BY-CLAUSE

**The GROUP BY clause specifies an intermediate result table that consists of a grouping of the rows of R.**

**R is the result of the previous clause.**

**Each column-name must unambiguously identify a column of R other than a long string column.**

**Each identified column is called a grouping column.**

**Result:**

**The result o a GROUP BY is a set of groups of rows.**

**In each group of more than one row, all values of each grouping column are equal and all rows with the same set of values of the grouping column are equal and all rows with the same set of values of the grouping columns are in the same group.**

**For grouping, all null values with in a grouping column are considered equal.**

**Syntax:**
**GROUP BY column name**

Figure 2.7 GROUPBY-clause

**Notes:**

Because every row of a group contains the same value of any grouping column, the name of a grouping column can be used in a search condition in a HAVING clause or an expression in a SELECT clause.

In each case, the reference specifies only one value for each group.

However, if the grouping column contains varying-length strings with trailing blanks, the values in the group and differs in the number of trailing blanks and may not all have the same length. In that case, a reference to the grouping column still specifies only one value for each group, but the value for a group is chosen arbitrarily from the available set of values.

Thus, the actual length of the result value is unpredictable.

# HAVING-CLAUSE

**The HAVING clause specifies an intermediate result table that consists of those groups of R for which the search-condition is true.**

**R is the result of the previous clause. If this clause is not GROUP BY.**

**R is the result of the previous column.**

**Syntax:**
**HAVING search condition**

Figure 2.9 HAVING-clause

**Notes:**

Each column-name is search-condition must:

Unambiguously identify a grouping column of R or

Be specific within a column function or

Be a correlated reference.

A column-name is a correlated reference if it identifies a column of a table or view identified in an outer subselect.

A group of R to which the search condition is applied supplies the argument for each function in the search condition, except for any function whose argument is a correlated reference.

If search condition contains a subquery, the subquery can be thought of as being executed each time the search condition is applied to a group of R, and the results used in applying the search condition. In actuality, the subquery is executed for each group only if it contains a correlated reference.

A correlated reference to a group of R must either identify a grouping column or be contained within a column function.

The HAVING clause must not be used in a subquery of a basic predicate.

When HAVING is used without GROUP BY any column name in the select list must appear within a column function.

# EXAMPLES OF SUBSELECTS

**Example1:**

```
SELECT * FROM DSN8510.EMP
```

**Example2:**

```
EXEC SQL
    SELECT JOB, MAX (SALARY), MIN (SALARY)
        FROM DSN8510.EMP
        GROUP BY JOB
    HAVING COUNT (*) > 1 AND MAX (SALARY) > 50000
END-EXEC.
```

**Example3:**

```
EXEC SQL
    SELECT EMPNO, ACTNO, CHAR (EMSTDATE, USA)
        FROM DSN8510.EMPPROJACT
        WHERE EMPNO IN
            (SELECT EMPNO FROM DSN8510.EMP
             WHERE WORKDEPT = 'E11'
END-EXEC.
```

Figure 2.11 Examples Of Subselect

**Notes:**

Example1: Shows all rows of the table DSN8510.EMP.

Example2: Show the job code, maximum salary, and minimum salary for each group of rows of DSN8510.EMP with the same job cod, but only for groups with more than one row and with a maximum salary greater than 50000.

# EXAMPLES OF SUBSELECTS (Contd….)

**Example4:**

```
EXEC SQL
    SELECT WORKDEPT, MAX (SALARY)
        FROM DSN8510.EMP
        GROUP BY WORKDEPT
    HAVING MAX (SALARY) < (SELECT AVG (SALARY)
                           FROM DSN8510.EMP)
END-EXEC.
```

**Example5:**

```
EXEC SQL
    SLEECT WORKDEPT, MAX (SALARY)
        FROM DSN8510.EMP Q
        GROUP BY WORKDEPT
    HAVING MAX (SALARY) <
        (SELECT AVG (SALARY)
          FROM DSN8510.EMP
          WHERE NOT WORKDEPT = Q.WORKDEPT
END-EXEC.
```

Figure 2.12 Examples Of Subselect (Contd….)

**Notes:**

Example3: For each employee in department E11, get the following information from the table DSN8510.EMPPROJACT: employee number, activity number, activity start date, and activity end date. Using the CHAR function, convert the start and end dates to their USA formats. Get the needed department information from the table DSN8510.EMP

# EXAMPLES OF SUBSELECTS (Contd….)

**Example6:**

```
EXEC SQL
    SELECT HIREYEAR, AVG (SLARY)
    FROM
    (SELECT (YEAR (HIREDATE) AS HIREDATE, SALARY
        FROM DSN8510.EMP) AS NEWMP
        GROUP BY HIREYEAR
END-EXEC.
```

Figure 2.13 Examples Of Subselect (Contd….)

**Notes:**

Example4: Show the department number and maximum departmental salary for all departments whose maximum salary is less than the average salary for all employees. In this example the subquery will be executed only once.

Example5: Show the department number and maximum departmental salary for all departments whose maximum salary is less than the average salary for employees in all other departments.

Example6: For each group of employees hired during the same year, show the year-of-hire and current average salary.

# FULLSELECT

**Fullselect specifies a result table. If UNION is not used, the result of the fullselect is the result of the specified subselect.**

**Syntax:**
**Subselect or fullselect UNION or UNION ALL subselect or fullselect**

**Description:**

**UNION or UNION ALL**
**Derives a result table by combining two other result tables, R1 and R2.**
**If UNION ALL is specified, the result consists of all rows in R1 and R2.**
**If UNION is specified without the ALL option, the result is the set of all rows in either R1 or R2, with duplicate rows eliminated.**

**Example:**

1. **EXEC SQL**
   **SELECT * FROM DSN8510.EMP**
   **END-EXEC.**
2. **EXEC SQL**
   **SELECT EMPNO FROM DSN8510.EMP**
   **WHERE WORKDEPT LIKE 'D%'**
   **UNION**
   **SELECT EMPNO FROM DSN8510.EMPPROJACT**
   **WHERE PROJNO LEKE 'AD%'**
   **END-EXEC.**

Figure 2.14 Fullselect

**Notes:**
Example1: Shows all the rows from DSN8510.EMP

Example2: Using tables DSN8510.EMP and DSN8510.EMPPROJACT, list the employee numbers of all employees for which either of the following statements are true:

Their department numbers begin with ‘D’.

They are assigned to projects whose project numbers begin with ‘AD’.

The result is the union of two result tables, one formed from the table DSN8510.EMP, the other formed from the table DSN8510.EMPPROJACT.

The result-a-column-table is a list of employee numbers. Because UNION, rather than UNION ALL, was used the entries in the list are distinct.

Instead UNION ALL was used, certain employee numbers would appear in the list more than once.

These would be the numbers for employees in departments that begin with ‘D’ while their projects begin with ‘AD’.

# SELECT-STATEMENT

**The select-statement can be issued interactively using SPUFI causing a result table to be displayed at the terminal.**

**Syntax:**
**fullselect order-by-clause**

**ORDER-BY-CLAUSE**
**The ORDER BY clause specifies an ordering of the rows of the result table.**

**Syntax:**
**ORDER BY column-name or integer ASC or DSC**

**Description:**

**Column-name: Must unambiguously identify a column of the result table.**
**Integer: Must be greater than 0 and not greater than the number of columns in the result table.**
**ASC: Uses the values of the column in ascending order. This is the default.**
**DESC: Uses the values of the column in descending order.**

Figure 2.16 SELECT statement

# EXAMPLES OF SELECT STATEMENT

**Example1:**

```
EXEC SQL
SELECT * FROM DSN8510.EMP
END-EXEC.
```

**Example2:**

```
EXEC SQL
SELECT * FROM DSN8510.EMP
ORDER BY HIREDATE
END-EXEC.
```

**Example3:**

```
EXEC SQL
SELECT WORKDEPT, AVG (SALARY)
FROM DSN8510.EMP
GROUP BY WORKDEPT
ORDER BY 2
END-EXEC.
```

Figure 2.17 Examples of SELECT statement

**Notes:**

Example1: Select all the rows from DSN8510.EMP

Example2: Select all the rows from DSN8510.EMP, arranging the result table in chronological order by date of hiring.

Example3: Select the department number (PORKDEPT) and average departmental salary (SALARY) for all departments in the table DSN8510.EMP. Arrange the result table in ascending order by average department salary.