

Structured Query Language

TABLE EXPRESSIONS
& PREDICATES

Table Expressions (Clauses)

- ▶ Table expressions are used to specify a table or a grouped table.
- ▶ Table expressions are:
 - FROM
 - WHERE
 - GROUP BY
 - HAVING
 - ORDER BY
- ▶ The From clause is necessary to a SQL query; the WHERE, GROUP BY, HAVING, and ORDER BY clauses are all optional.

FROM Clause

- ▶ The FROM clause specifies a table or tables from which the desired rows are to be retrieved.
- ▶ For example:
 - `SELECT *`
 - `FROM STUDENT`

 - `SELECT STUDENT_NO, ADDRESS`
 - `FROM STUDENT`
- ▶ If attributes are required from more than one table, then the FROM clause must name all the tables involved in any order.

WHERE Clause

- ▶ The WHERE clause specifies the row or rows retrieved from a table meet the WHERE clause specification.
- ▶ The syntax is:
 - SELECT column1, column2, ..., columnN
 - FROM tablename
 - WHERE condition;
- ▶ An example:
 - SELECT Student_No, Student_Name
 - FROM Student
 - WHERE Course_No = "HNC34";

GROUP BY Clause

- ▶ The result of the GROUP BY clause partitions the result of the FROM clause into a set of groups so that for each group of more than one row, the values in the grouping attribute are identical.
- ▶ The syntax is:
 - SELECT column1, column2, ..., columnN
 - FROM tablename
 - GROUP BY groupin_column;
- ▶ an example:
 - SELECT lecturer_no, campus
 - FROM Lecturer
 - GROUP BY Campus, Lecturer_no;

GROUP BY Clause

Microsoft Access example:

```
SELECT Lecturer_No, Name, Campus
```

```
FROM Lecturer
```

```
GROUP BY Campus, Name, Lecturer_No;
```

HAVING Clause

- ▶ The HAVING clause is used to specify the quality a group must possess for it to be returned in the output.
- ▶ HAVING compares a property of the group with a constant value.
- ▶ It perform the same function for groups that WHERE performs for individual rows.
- ▶ HAVING must be used in conjunction with GROUP BY.
- ▶ The syntax is:
 - SELECT column1, column2, ..., columnN
 - FROM tablename
 - GROUP BY groupin_column
 - HAVING specified_group_prpoerty;

HAVING Clause

► An example:

- SELECT Campus, COUNT(*)
- FROM Lecturer
- GROUP BY Campus
- HAVING COUNT(*) > 1;

► Microsoft Access example:

- SELECT Lecturer_No, Name, Campus
- FROM Lecturer
- GROUP BY Campus, Name, Lecturer_No
- HAVING Campus = "Mayals" OR Campus = "Town";

ORDER BY Clause

- ▶ The ORDER BY clause allows you to specify the order in which rows will appear in the retrieval.
- ▶ If the ORDER BY clause is used, it must be the last clause in a SELECT command.
- ▶ The syntax is:
 - SELECT column1, column2, ..., columnN
 - FROM tablename
 - ORDER By column1, column2, ..., columnN;
- ▶ An example:
 - SELECT Lecturer_No, Name, Campus
 - FROM Lecturer
 - ORDER BY Campus, Name;

Predicates

- ▶ Predicates are conditions that are stated in the WHERE clause of a SQL query.
- ▶ The predicates supported by SQL are:
 - Comparison
 - BETWEEN
 - IN
 - LIKE
 - NULL
 - Quantified
 - EXISTS

Comparison Predicate

- ▶ A comparison predicate specifies a comparison of two values.
- ▶ It consists of a value expression followed by a comparison operator followed by either another value expression, or a single-valued subquery.
- ▶ The data types of the two value expressions, or the value expression of the subquery, must be compatible.
- ▶ The comparison operators supported by SQL are:
 - equals =
 - not equal to <>
 - less than <
 - greater than >
 - less than or equal to <=
 - greater than or equal to >=

BETWEEN Predicate

- ▶ The BETWEEN predicate specifies a range comparison.
- ▶ The syntax is:
 - BETWEEN AND
 - Or
 - NOT BETWEEN AND
- ▶ An example:
 - SELECT Part_No, Part_Name, Price, Supplier
 - FROM Supplier
 - WHERE Price BETWEEN 50 and 75;

BETWEEN Predicate

An example:

- SELECT Part_No, Part_Name, Price, Supplier
- FROM Supplier
- WHERE Price NOT BETWEEN 150 and 275;

IN Predicate

The IN (or NOT IN) predicate specifies a quantified comparison.

It lists a set of values and tests for whether or not a candidate value is in the list.

The list must be enclosed in brackets.

An example

- `SELECT Part_No, Part_Name, Price, Supplier`
- `FROM Supplier`
- `WHERE Price IN(100, 120, 180);`

IN Predicate

- ▶ You could write the query using OR instead of IN

- SELECT Part_No, Part_Name, Price, Supplier
- FROM Supplier
- WHERE Price = 100
- OR Price = 120
- Or Price = 180;

- ▶ The query can also be written using ANY

- SELECT Part_No, Part_Name, Price, Supplier
- FROM Supplier
- WHERE Price = ANY (100, 120, 180);

LIKE Predicate

- ▶ The LIKE predicate specifies a pattern-matching comparison in which an underscore (_) is used to represent a single character in the pattern, and a percent sign (%) is used to represent a character string of arbitrary length (including zero).
- ▶ The form of the LIKE predicate is:
 - `column_name LIKE character_string_constant`
- ▶ An example: Select all Lecturers whose Name starts with the letter 'R'
 - `SELECT Lecturer_No, Name, Campus`
 - `FROM Lecturer`
 - `WHERE Name Like "R%";`

LIKE Predicate

Microsoft Access example

- `SELECT Lecturer.Lecturer_No, Lecturer.Name, Lecturer.Campus`
- `FROM Lecturer`
- `WHERE Name Like "R*";`