IN, OR, AS, BETWEEN AND Operators

Dr. Sambit Bakshi

NIT Rourkela

March 31, 2020

Outline

In Operator

IN operator allows you to determine if a specified value matches any value in a set of values or returned by a subquery.

In Operator

- Syntax:

SELECT
column1,column2
FROM
table_name
WHERE
(expr—column_1) IN ('value1','value2');

In Operator

Example: A **Regional_Offices** table contains the entries for the following attributes.

- Office_Code
- City
- AddressLine
- State
- County

In Operator

Example: We need to find out the offices those locate in India and China.

- Query :

SELECT Office_Code, City, Country
FROM Offices
WHERE Country IN ('India', 'China');

Not In Operator

NOT operator combines with **IN** operator to determine the values in a list or a subquery. Example: **SELECT** Select office_code, country from Offices
Where NOT IN ('India', 'China');

- The above query produces a table containg all the office_codes and the corresponding countries except INDIA AND CHINA.

OR Operator

To determine the offices which are present in France or USA.

- Example:

SELECT office_code, country
FROM Offices
WHERE Country = 'India' OR Country = 'China';

Aliases

- **To have a better understand of the columns in a query output alias has been used**
- **Syntax :**

SELECT [column_1 — expression] **AS** descriptive_name **FROM** table_name;

Aliases

- **Example :**

SELECT First_name , Last_name
AS FULL_NAME
FROM Student_Details;

Between And operator

- **Between And operator is used to check whether a value is in range or not.**
- **Often used in where clause of select, update, delete statement.**
- Syntax :

expr [NOT] **BETWEEN** begin_expr **AND** end_expr;

Between And operator

- Product (P_code, P_name, Price)
- Find out the products having price between 90 to 100;

SELECT P_Code, P_name, Price
FROM Product **WHERE** Price **BETWEEN** 90 **AND** 100;;

SQL

Group By, Having, Order By Clause

GROUP BY

- Groups a set of rows into a set of summary rows by values of columns or expressions
- Returns one row for each group.
- Often use with aggregate functions such as **SUM**, **AVG**, **MAX**, **MIN** and **COUNT**.

GROUP BY

- Syntax :

SELECT c1, c2,..., cn, aggregate_function(ci) **FROM** table **WHERE** where_conditions

GROUP BY

- It must appear after FROM and WHERE clauses
- Evaluation order of GROUP BY clause as follows:
- FROM WHERE SELECT GROUP BY

GROUP BY

- Example:
- Order (o_no, product_code, quantity_order, price)
- Suppose we need to find out the order numbers and the corresponding price for that order number.

SELECT o_no, SUM(quantity_order * price) **AS** Total **FROM** Order **GROUP BY** o_no;

HAVING Clause

- Having clause is used to filter conditions for a group of rows or aggregates.
- Syntax:

SELECT select_list **FROM** Table_name
WHERE Search_condition **GROUP BY** group_expressions **HAVING** group condition;

HAVING Clause

- Evaluation order of HAVING clause is done in the following sequence.
- FROM WHERE SELECT GROUP BY HAVING:
- Exam : Order (O_no, Product_Code, Quantity_ordered, Price_each)

HAVING Clause

- To get order numbers and total sales for each order from the Order table.

SELECT o_no, SUM (quantity_ordered * PriceEach)
AS Total_Sales
FROM Order **GROUP BY** o_no;

Suppose the output of the order is as follows:

O_no	Total_Sales
1	1200
2	700
3	1050

HAVING Clause

- Now if it has been asking to findout order_nos and total_sales which are greater than 1000.

SELECT o_no, SUM (quantity_ordered * PriceEach)
AS Total_Sales
FROM Order **GROUP BY** o_no;
Having Total_Sales ¿ 1000

O_no	Total_Sales
1	1200
3	1050

Order By Clause

- The output of SELECT Statement is not a sorted one.
- To sort them, Order By clause is being used.
- Evaluation Sequence of Order By clause is as follows.

FROM WHERE SELECT GROUP BY HAVING ORDER BY

Order By Clause

- The output of SELECT Statement is not a sorted one.
- To sort them, Order By clause is being used.
- Evaluation Sequence of Order By clause is as follows.

SELECT select_list
FROM Table_name **ORDER BY** Column1 [ASC—DSC], Column2 [ASC—DSC];

Order By Clause

- Order (O_no, Product_Code, Quantity_ordered, Price_each).
- Find the o_no and the product code from high to low order of the Price.
- Evaluation Sequence of Order By clause is as follows.

SELECT O_no, Product_Code, Quantity_ordered * Price_each **AS** Table **FROM** Order **ORDER BY** Total DESC;