Database Design and SQL

Chapter 6: Single Table Queries (Part 2)

CPCM-2020 Fall
Sagara Samarawickrama
Term 1 - Group 3

Single Table Queries

Sorting Rows With The ORDER BY Clause:

The Order by clause is used to return the results of a query in a sorted order based on the columns specified with the clause.

```
SELECT    ship_city, customer_name, customer_id
  FROM    customers
  ORDER BY ship_city, customer_name;
```

Results

SHIP_CITY	CUSTOMER_NAME	CUSTOMER_ID
Albany	Ajax Steel Inc.	275978
Albany	Seaworthy	888402
Albany		

The order by clause sorts the result set in ascending order by customer_name within ship_city.

Single Table Queries

The sort order can be ascending or descending . An Order by clause specifies a list of columns with ascending or descending sequence.

Retrieve ship_city, customer_name, **and** customer_id **for all rows in the** CUSTOMERS **table and sequence in** customer_name **within** ship_city **(descending) order.**

```
SELECT     ship_city, customer_name, customer_id
  FROM     customers
  ORDER BY ship_city DESC, customer_name;
```

When descending order is required , the DESC keyword is specified after the column that is to be sorted in descending order

Single Table Queries

LENGTH Function:

The LENGTH function returns the length of a string as a numeric value.

```
SELECT LENGTH ('GoodDay')
  FROM SYSIBM.SYSDUMMY1;

SELECT LENGTH (last_name)      -- Assume last_name is "Barker"
 FROM employees;
```

Results

```
7
6
```

Single Table Queries

AGGREGTE FUNCTIONS:

Aggregate functions produce a value from set of rows Following figure lists the available SQL aggregate functions.

Function	Returns
AVG	The average value of a given column
COUNT	The total number of values in a given column
COUNT(*)	The total number of rows in a table
MAX	The largest value in a given column
MIN	The smallest value in a given column
SUM	The sum of the numeric values in a given column

Single Table Queries

COUNT Function:

The count function returns a single value containing the number of rows in the result sheet.

Retrieve the number of Boston customers in the CUSTOMERS table.

```
SELECT COUNT(*) AS "Count"
  FROM customers
  WHERE ship_city = 'Boston';
```

Results

```
Count
-----
    3
```

Single Table Queries

SUM Function:

The SUM function returns a single value containing the sum(total) for a column in the result set. In example shown below the SUM function returns the accumulated value for all sales in the ORDERS table.

Return the total sales for all orders in the ORDERS table.

```
SELECT  SUM(order_total) AS "Total Sales"
  FROM  orders;
```

Results

```
Total Sales
-----------
    3310.85
```

Single Table Queries

CASE Function:

The CASE function provides a multi-condition test. The query shown below uses CASE to produce a table that categorizes ship_city according the province.

```
SELECT customer_name,
       CASE ship_city
         WHEN 'Albany'     THEN 'New York'
         WHEN 'Boston'     THEN 'Massachusett
         WHEN 'Chicago'    THEN 'Illinois'
         WHEN 'Dallas'     THEN 'Texas'
         WHEN 'Detroit'    THEN 'Michigan'
         WHEN 'Houston'    THEN 'Texas'
         WHEN 'Portland'   THEN 'Oregon'
         WHEN 'San Diego'  THEN 'California'
         WHEN 'Toronto'    THEN 'Ontario'
       END AS state
  FROM customers
  ORDER BY state;
```

Results

```
CUSTOMER_NAME                    STATE
-------------------------------- -------------
Nautilus Mfg.                    California
```

Single Table Queries

GROUP BY and HAVING CLAUSES

The GRO UP BY clause works with aggregate functions to group the data in the result set by columns. The rows in the result set are grouped together based on the columns specified in the GROUP BY clause. In addition , the GROUP BY clause can be used to apply aggregate functions to (sub)groups of the rows in the result set.

The SQL statement shown in the next slide will return one row for each group of customers in a different city.

Single Table Queries

Example 6-66: The GROUP BY clause

Retrieve ship_city **as** "Ship City", count **as** "Count", **and** Average Discount **as** "Average Discount" **for all rows in the** CUSTOMERS **table and group by** ship_city.

```
SELECT     ship_city AS "Ship City",
           COUNT( * ) AS "Count",
           DECIMAL(ROUND(AVG(discount),3), 3,3)
               AS "Average Discount"
  FROM     customers
  GROUP BY ship_city;
```

Results

```
Ship City                 Count Average Discount
------------------------- ----- ---------------
Toronto                       1           0.020
Chicago                       1           0.050
```

Single Table Queries

HAVING CLAUSE

The HAVING clause can be used to restrict rows in the result set after aggregate functions have been applied to grouped rows. The having clause takes a form similar to that of WHERE clause, which selects rows before they are grouped.

Retrieve the average discount of customers in Portland and Boston using a HAVING clause.

```
SELECT    ship_city AS "Ship City",
          DECIMAL (ROUND( AVG( discount ),3 ), 3,3)
              AS "Average Discount"
  FROM    customers
  GROUP BY ship_city
  HAVING   ship_city IN ( 'Portland', 'Boston' );
```

Results

```
Ship City                      Average Discount
------------------------------ ---------------
Boston                                   0.030
Portland                                 0.015
```

Conclusion