SQL In 10 Minutes

Summarizing & Manipulating Data - Vibhu Sapra

Manipulate & Sum

Overview

- Extract the specific data you want
  - Not very portable
  - Each DBMS uses different syntax
  - Text, Date, and Numeric Manipulation
- Summarize data
  - Actually pretty consistent across DBMS
  - Avg, Count, Max, Min, Sum, etc.

Data Manipulation

Inconsistency 🤦

General Overview

Data Manipulation

- Manipulate text / dates / numbers
- Every DBMS uses different syntax
- Can still accomplish the same tasks, just different code
  - Text Functions
  - Numeric Functions
  - Date & Time Functions
  - Formatting Functions
  - System Functions

Syntax Issues

Data Manipulation

- Every DBMS uses different syntax

Extract part of a string	DB2, Oracle, PostgreSQL, and SQLite use `SUBSTR()`. MariaDB, MySQL, and SQL Server use `SUBSTRING()`.
Datatype conversion	Oracle uses multiple functions, one for each conversion type. DB2, PostgreSQL, and SQL Server use `CAST()`. MariaDB, MySQL, and SQL Server use `CONVERT()`.
Get current date	DB2 and PostgreSQL use `CURRENT_DATE`. MariaDB and MySQL use `CURDATE()`. Oracle uses SYSDATE. SQL Server uses `GETDATE()`. SQLite uses `DATE()`.

Text Manipulation

Text

Text Manipulation

- Saw RTRIM( ) last time
- Not case sensitive
  - upper(), UPPER(), Upper(), substr(), SUBSTR(), SubStr() all work the same
- Many more text manipulation functions
- Let’s start with Upper() -> Uppercase text

Text Manipulation

Text

```
SELECT vend_name, UPPER(vend_name) AS vend_name_upcase
FROM Vendors
ORDER BY vend_name;
```

Output ▼

Click here to view code image

```
vend_name                 vend_name_upcase
------------------------  ------------------------
Bear Emporium             BEAR EMPORIUM
Bears R Us                BEARS R US
Doll House Inc.           DOLL HOUSE INC.
Fun and Games             FUN AND GAMES
Furball Inc.              FURBALL INC.
Jouets et ours            JOUETS ET OURS
```

Text

Common Functions

Function	Description
`LEFT()` (or use substring function	Returns characters from left of string
`LENGTH()` (also `DATALENGTH()` or `LEN()`)	Returns the length of a string
`LOWER()`	Converts string to lowercase
`LTRIM()`	Trims white space from left of string
`RIGHT()` (or use substring function)	Returns characters from right of string
`RTRIM()`	Trims white space from right of string
`SUBSTR()` or `SUBSTRING()`	Extracts part of a string (as noted in Table 8.1)
`SOUNDEX()`	Returns a string's `SOUNDEX` value
`UPPER()`	Converts string to uppercase

Soundex

Soundex Fn

- Fix typos in names and stuff
- Convert text to alphanumeric pattern
  - Not in PostgreSQL
  - Not default in SQLite
- Customer named “Michelle Green”
  - Typo and should be “Michael Green”

Soundex

Soundex Fn

```
SELECT cust_name, cust_contact
FROM Customers
WHERE cust_contact = 'Michael Green';
```

Output ▼

Click here to view code image

```
cust_name                      cust_contact
-----------------------------  ------------
```

Soundex

Soundex Fn

Click here to view code image

```
SELECT cust_name, cust_contact
FROM Customers
WHERE SOUNDEX(cust_contact) = SOUNDEX('Michael Green');
```

Output ▼

Click here to view code image

```
cust_name                      cust_contact
-----------------------------  -----------------------------
Kids Place                     Michelle Green
```

Date & Time Manipulation

Date + Time

Date & Time

- Values sorted in special format so they can be indexed quickly
  - Internal format = unreadable
- Most inconsistent between DBMS :(
- Basic querying

Click here to view code image

```
SELECT order_num
FROM Orders
WHERE DATEPART(yy, order_date) = 2020;
```

Output

```
order_num
-----------
20005
20006
20007
20008
```

Date + Time

Date & Time

- Values sorted in special format so they can be indexed quickly
  - Internal format = unreadable
- Most inconsistent between DBMS :(
- Basic querying - DATEPART ( )
  - Only for SQL Server
- Every DBMS is different…
  - Know the one you need

Date + Time

SQL Version

Click here to view code image

```
SELECT order_num
FROM Orders
WHERE DATEPART(yy, order_date) = 2020;
```

Output ▼

```
order_num
-----------
20005
20006
20007
20008
```

Date + Time

Oracle Version

Input ▼

Click here to view code image

```
SELECT order_num
FROM Orders
WHERE EXTRACT(year FROM order_date) = 2020;
```

Date + Time

PostgreSQL Version

Input ▼

Click here to view code image

```
SELECT order_num
FROM Orders
WHERE DATE_PART('year', order_date) = 2020;
```

- PostgreSQL also can do datepart() and extract

Date + Time

Date & Time

- DB2, MySQL, and MariaDB have all sorts of date manipulation functions, but not DATEPART()
  - Users can use a function named YEAR()
- SQLite is also different

SQLite is a little trickier:

Input ▾

Click here to view code image

```
SELECT order_num
FROM Orders
WHERE strftime('%Y', order_date) = '2020';
```

Numeric Manipulation

Numeric

Numeric Functions

- Uniform and consistent!
  - Algebraic, Trig, Geometric, etc. transformations

Function	Description
ABS()	Returns a number's absolute value
COS()	Returns the trigonometric cosine of a specified angle
EXP()	Returns the exponential value of a specific number
PI()	Returns the value of PI
SIN()	Returns the trigonometric sine of a specified angle
SQRT()	Returns the square root of a specified number
TAN()	Returns the trigonometric tangent of a specified angle

Summary

- Very inconsistent
- Still good to use - fast & scalable
- Relevant in the next section on Summarizing Data
- Questions break?

Data Summarization

Consistency 😍

General Overview

Data Summarization

- Summarize data w/o extracting it all
  - More efficient to pull what you need
  - SQL is fast
- Aggregation functions are pretty universal between DBMS
- Very Straightforward and “easy”

Aggregation Fn ( )

Common Fn()

Table 9.1 SQL Aggregate Functions

Function	Description
AVG()	Returns a column's average value
COUNT()	Returns the number of rows in a column
MAX()	Returns a column's highest value
MIN()	Returns a column's lowest value
SUM()	Returns the sum of a column's values

Average

Average Fn ( )

- Returns the average of all columns

Input

Click here to view code image

```
SELECT AVG(prod_price) AS avg_price
FROM Products;
```

Output

```
avg_price
-------------
6.823333
```

Input

Click here to view code image

```
SELECT AVG(prod_price) AS avg_price
FROM Products
WHERE vend_id = 'DLL01';
```

Output

```
avg_price
-----------
3.8650
```

Average

Average Fn ( )

- Returns the average of all columns
- Can only do the average of one column at once
- Null Values = ignored

Average

Average Fn ( )

- Returns the average of all columns

Input

Click here to view code image

```
SELECT AVG(prod_price) AS avg_price
FROM Products;
```

Output

```
avg_price
-------------
6.823333
```

Input

Click here to view code image

```
SELECT AVG(prod_price) AS avg_price
FROM Products
WHERE vend_id = 'DLL01';
```

Output

```
avg_price
-----------
3.8650
```

Count

Count Fn ( )

- Determine number of rows in a table that meet a specific criteria
- Count (*) = number of all rows regardless of null
- Count (column) = number of rows that have a value in a column ignoring null

Count

Count Fn ( )

- Count (*) = total number of customers
- Count (cust_emails) = total number of customers that gave their emails
  - Ignores null values

Count

Count Fn ( )

Input ▼

```
SELECT COUNT(*) AS num_cust
FROM Customers;
```

Output ▼

```
num_cust
--------
5
```

Input ▼

Click here to view code image

```
SELECT COUNT(cust_email) AS num_cust
FROM Customers;
```

Output ▼

```
num_cust
--------
3
```

Min / Man

Min & Max Fn ( )

- Returns the Max / Min values
- Ignore Null values
- * Max can be used with non-numeric but not Min *

```
SELECT MAX(prod_price) AS max_price
FROM Products;
```

Output

```
max_price
----------
11.9900
```

```
SELECT MIN(prod_price) AS min_price
FROM Products;
```

Output

```
min_price
----------
3.4900
```

Sum

Sum Fn ( )

- Returns the sum (total) of all values
- Can be used to calc total (price * total quantity (ignore null)

```
SELECT SUM(quantity) AS items_ordered
FROM OrderItems
WHERE order_num = 20005;
```

Output

```
items_ordered
----------
200
```

```
SELECT SUM(item_price*quantity) AS total_price
FROM OrderItems
WHERE order_num = 20005;
```

Output

```
total_price
----------
1648.0000
```

Tips

General Tips

- To perform calculations on all rows, specify the **ALL** argument
  - On by default
- To include only unique values, specify the **DISTINCT** argument.
  - Useful for things like average price of items

```
SELECT AVG(DISTINCT prod_price) AS avg_price
FROM Products
WHERE vend_id = 'DLL01';
```

Output ▾

```
avg_price
-----------
4.2400
```

Tips

Combine Fns ( )

Click here to view code image

```
SELECT COUNT(*) AS num_items,
       MIN(prod_price) AS price_min,
       MAX(prod_price) AS price_max,
       AVG(prod_price) AS price_avg
FROM Products;
```

Output ▼

Click here to view code image

```
num_items    price_min        price_max        price_avg
----------   ---------------  ---------------  ---------
9            3.4900           11.9900          6.823333
```

THANKS!

Questions?