Aggregating Functions

Jason P. Browning

ASST. COMMISSIONER OF HIGHER EDUCATION, STATE OF MISSOURI

@jason_from_ky

Overview

Field-level calculations

Aggregate functions

Distinct aggregate functions

Analyzing groups

Using aggregate results to filter data

Numeric Data

Integer	Numeric
INTEGER (or INT)	NUMERIC
Whole numbers	Exact quantities
No fractional component	Can specify precision and scale

Whole number

25

25.714

Precision: 5

Scale: 3

Numeric Data

Floating-point Types	Inexact, variable-precision numeric types
REAL	6 digit decimal precision
DOUBLE PRECISION	15 digit decimal precision
FLOAT	SQL-standard notation

Numeric Data

Use integer type to store whole numbers

Use numeric for monetary amounts or where precision is required

When possible, avoid using floating-point columns in WHERE clause

```
> SELECT 2 + 2;

  '4'

> SELECT 12 / 2;

  '6'

> SELECT 13 / 2;

  '6'

> SELECT 13 / 2::FLOAT;

  '6.5'
```

- **Basic arithmetic operators**
  - + Addition
  - - Subtraction
  - * Multiplication
  - / Division

- **Be aware of data types**

- **By casting number to a floating-point type, the calculation returns a double-precision type**

```
> SELECT 15 % 2;

  '1'

> SELECT 12 ^ 2;

  '144'

> SELECT |/ 36;

  '6'

> SELECT @ (36 - 40);

  '4'

> SELECT ABS(36 – 40);

  '4'
```

Other arithmetic operators

% Modulo

^ Exponent

|/ Square root

@ Absolute value

ANSI SQL compliant operators

ABS() returns absolute value

MOD() returns modulo

POWER(#, *p*) returns number # raised to given power *p*

SQRT() returns square root

Aggregate Functions

COUNT

SUM

AVG

MIN

MAX

```
SELECT AVG(age)    AS avg_age
 FROM person;
```

name	grade_lvl	age
Eliza	Junior	17
Jane	Junior	17
Leslie	Senior	19
Matt	Junior	16
Ned	Freshman	15
Susie	Junior	18

Aggregate Functions

To use an aggregate function, include it in the SELECT clause

The above code returns an average age of 17

Distinct in Aggregate Functions

name	grade_lvl
Eliza	Junior
Jane	Junior
Leslie	Senior
Matt	Junior
Ned	Freshman
Susie	Junior

```
SELECT COUNT(grade_lvl)
  FROM person;
```

6

```
SELECT COUNT(DISTINCT grade_lvl)
  FROM person;
```

3

Analyzing Groups

Aggregate functions can be used for more sophisticated analysis

What is our average age by grade level?

GROUP BY keyword is used to specify groups

```
SELECT grade_lvl,
       AVG(age) AS avg_age
  FROM person
 GROUP BY grade_lvl;
```

grade_lvl	avg_age
Freshman	15
Junior	17
Senior	19

- Aggregate average function
- Group results by grade level

name	grade_lvl	age
Eliza	Junior	17
Jane	Junior	17
Leslie	Senior	19
Matt	Junior	16
Ned	Freshman	15
Susie	Junior	18

Using GROUP BY with Aggregation

Incorrect

```
SELECT grade_lvl,
       MIN(age) AS minimum_age
  FROM person;
```

All non-aggregate fields in the SELECT clause must be represented in the GROUP BY clause

Correct

```
SELECT grade_lvl,
       MIN(age) AS minimum_age
  FROM person
 GROUP BY grade_lvl;
```

Demo

Explore aggregate functions

Filtering Aggregate Results

WHERE

Filter single rows

HAVING

Filter aggregate results

```
SELECT grade_lvl,
       AVG(age) AS avg_age
  FROM person
 GROUP BY grade_lvl
HAVING AVG(age) < 19;
```

grade_lvl	avg_age
Freshman	15
Junior	17

- HAVING clause specifies that we want to filter aggregate values from AVG

name	grade_lvl	age
Eliza	Junior	17
Jane	Junior	17
Leslie	Senior	19
Matt	Junior	16
Ned	Freshman	15
Susie	Junior	18

Demo

Using HAVING to filter results

Summary

Be aware of data types

Aggregate functions perform calculations

- On entire data set
- On groups specified using GROUP BY

HAVING filters aggregate results