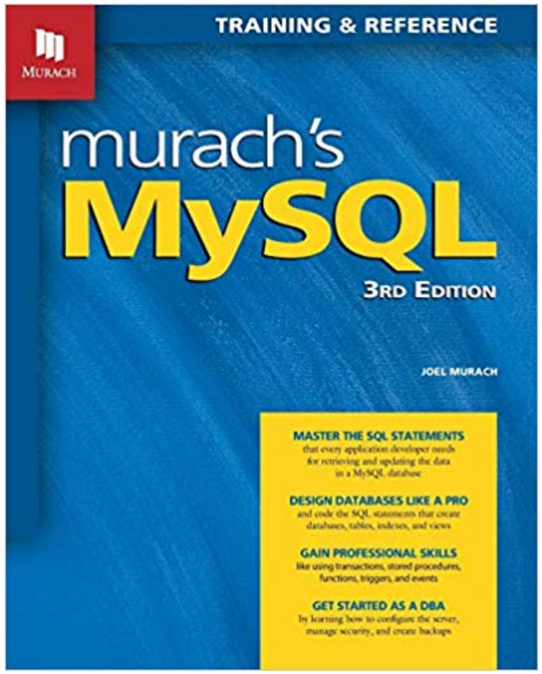
Aggregation using SQL SELECT

Topic 3
Lesson 2 – SQL SELECT Aggregation Queries

Chapter 6 Murach's MySQL


TRAINING & REFERENCE

MURACH

murach's

MySQL

3RD EDITION

JOEL MURACH

MASTER THE SQL STATEMENTS
that every application developer needs
for retrieving and updating the data
in a MySQL database

DESIGN DATABASES LIKE A PRO
and code the SQL statements that create
databases, tables, indexes, and views

GAIN PROFESSIONAL SKILLS
like using transactions, stored procedures,
functions, triggers, and events

GET STARTED AS A DBA
by learning how to configure the server,
manage security, and create backups

Aggregate functions in MySQL

```
AVG([ALL|DISTINCT] expression)
SUM([ALL|DISTINCT] expression)
MIN([ALL|DISTINCT] expression)
MAX([ALL|DISTINCT] expression)
COUNT([ALL|DISTINCT] expression)
COUNT(*)
GROUP_CONCAT([ALL|DISTINCT] expression)
```

An aggregate function can be listed in the field list of the SELECT clause. It will compute the desired aggregation across the tuples qualified by the query.

Aggregate function keywords in SQL

All aggregate functions by default exclude NULL values before working on the data.

The DISTINCT keyword omits duplicates from the results.

The ALL keyword includes even duplicates. If nothing is specified, the ALL is the default behavior.

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

AVG and SUM in MySQL

```
AVG([ALL|DISTINCT] expression)
SUM([ALL|DISTINCT] expression)
```

- AVG and SUM can be applied to numeric expressions.
- AVG returns the average of a column (or an expression).
- SUM returns the sum of a column (or an expression)

EXAMPLE: SELECT AVG(credits_earned) FROM student;

SELECT AVG(credits_earned) AS avg_earned FROM student;

MIN and MAX functions in MySQL

```
MIN([ALL|DISTINCT] expression)
MAX([ALL|DISTINCT] expression)
```

- MIN and MAX can be applied to numeric expressions.
- MIN returns the minimal value of a column (or an expression).
- MAX returns the maximum value of a column (or an expression)

EXAMPLE:
```
SELECT MAX(credits_earned) AS max_earned,
       MIN(credits_earned) AS min_earned
FROM student;
```

COUNT function in MySQL

```
COUNT([ALL|DISTINCT] expression)
COUNT(*)
```

COUNT counts the number of values for a field.

COUNT(*) is a special case, it counts the number of rows. It does not eliminate rows with NULL values. It also counts duplicate rows.

EXAMPLE: SELECT COUNT(*) FROM student;
EXERCISE: create a query to COUNT THE DISTINCT schools in the student table.

GROUP_CONCAT function in MySQL

```
GROUP_CONCAT([ALL|DISTINCT] expression)
```

GROUP_CONCAT is a MYSQL specific aggregate function. It can only be applied to text data, CHAR OR VARCHAR. It returns a comma separated list of the values.

EXAMPLE:
```
SELECT GROUP_CONCAT(school)
                    FROM students;
SELECT GROUP_CONCAT(DISTINCT school)
                    FROM students;
```

Examples: Aggregate functions in MySQL

```
SELECT COUNT(*) FROM student;
SELECT COUNT(*) AS num_students FROM student;
SELECT SUM(credits_earned) Khoury_Earned FROM
  student WHERE school = ‘Khoury’;
SELECT COUNT(DISTINCT school) AS num_schools FROM
  student;
```

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128
3	Li	Khoury	50	120

Solution: Aggregate functions in MySQL

```
SELECT COUNT(*) FROM students;
SELECT COUNT(*)AS num_students FROM students;
SELECT SUM(Credits_Earned) Khoury_Earned FROM
  students WHERE School = 'Khoury';
SELECT COUNT(DISTINCT School) AS num_schools FROM
  students;
```

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

COUNT(*)
3

num_students
3

Khoury_Earned
82

Num_schools
2

Examples 2: Aggregate functions in MySQL

```
SELECT MIN(credits_req) FROM students;
SELECT MAX(credits_earned)AS max_credits FROM
  students;
SELECT MAX(credits_earned)AS max_credits FROM
  students WHERE school = ‘Khoury’;
```

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128
3	Li	Khoury	50	120

Solution: Aggregate functions in MySQL

```
SELECT MIN(credits_req) FROM students;
SELECT MAX(credits_earned) AS max_credits FROM
  students;
SELECT MAX(credits_earned)AS khoury_earned FROM
  students WHERE school = ‘Khoury’;
```

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128
3	Li	Khoury	50	120

MIN(credits_req)
120

Credits_earned
64

Khoury_Earned
50

GROUP BY clause

Sometimes you want to stratify the qualifying tuples by a specific field in the table. Use the GROUP BY clause to define groups of tuples. It produces a result tuple for each combination of the fields' values in the GROUP BY list. The GROUP BY clause has a supporting HAVING clause. The syntax for the SELECT statement with the GROUP BY clause is:

```
SELECT select_list
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_list]
[HAVING search_condition]
[ORDER BY order_by_list]
```

Examples: GROUP BY clause

```
SELECT school FROM students GROUP BY school;
SELECT school, credits_required FROM students
  GROUP BY school, credits_required;
```

GROUP BY can also be used to remove duplicate tuples.

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

Solution: GROUP BY clause

```
SELECT school FROM students GROUP BY school;
SELECT school, credits_required FROM students
  GROUP BY school, credits_required;
```

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

School
Khoury
D'Amore McKim

School	Credits_Req
Khoury	120
D'Amore McKim	128

Combining Aggregation and GROUP BY

We can apply aggregations to specific groups.
Specify a query that returns the largest number of credits earned for each school.

Student

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

School	Max(credits_earned)
Khoury	50
D'Amore McKim	64

Solution query: max credit by school

```
SELECT school, max(credits_earned) FROM student
  GROUP  BY school;
```

Student

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

School	Max(credits_earned)
Khoury	50
D'Amore McKim	64

HAVING clause creates a filter

The HAVING clause applies a filter to the result tuples created by the grouping operation. It is analogous to the WHERE clause

Student

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128
3	Li	Khoury	50	120

Example using the HAVING clause

Write a query that returns the maximum credits_earned for each school in the students table, if the maximum credits_earned > 60.

Student

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128
3	Li	Khoury	50	120

Solution

SELECT school, MAX(credits_earned) FROM students GROUP BY school HAVING MAX(credits_earned) > 60;

Student

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

GROUP BY result

School	Max(credits_earned)
Khoury	50
D'Amore McKim	64

Having result

School	Max(credits_earned)
D'Amore McKim	64

WITH ROLLUP

An extension to the GROUP BY clause, it includes summary rows in the result for each subgroup in your groups. It generates grand totals and subtotals for reports.

It has not been accepted into the SQL Standard but most vendors have implemented it.

EXAMPLE: select school, sum(credits_earned) FROM student GROUP BY school WITH ROLLUP;

EXERCISE: Apply WITH ROLLUP to the aggregate statements written for exercise 1.

Solution

SELECT school, SUM(credits_earned) FROM student GROUP BY school WITH ROLLUP;

Student

id	name	school	credits_earned	credits_req
1	Smith	Khoury	32	120
2	Shah	D'Amore McKim	64	128
3	Li	Khoury	50	120

GROUP BY result

School	SUM(credits_earned)
Khoury	82
D'Amore McKim	64

Rollup result

School	SUM(credits_earned)
Khoury	82
D'Amore McKim	64
	146

Summary

In this module you learned:

- Aggregation functions can be used in the field list as well as in the HAVING clause
- SQL SELECT clauses: GROUP BY and HAVING clause
- Necessary coordination between the field list and the GROUP BY list

MySQL work

Let’s use the MySQL workbench to write Aggregation statements using the ap database

Go to the website, page topic 2, and download the sql_exercises1.sql exercise. Complete the rest of the queries.

SQL work 2

Return the vendor name and vendor contact name as one field

Return the vendor names that start with P

Count the number of vendors

Count the number of invoices per vendor

Calculate the total payment by vendor, return the vendor name and the total

Calculate the total payment by vendor, return the vendor id and the total payment for vendor for vendors who paid more than $100