Aggregate Functions

CS 2550 Database Design 2023
Cody Henrichsen

1

Definition

- Aggregate functions operate on an entire COLUMN's worth of values compared to the values in individual cells
- The column type must 'fit' the function
  - You can MIN a date or text
  - You can't AVG a text or a date

id	first_name	last_name	account_id	age	birth_year	address_id
	sdf	fg	354	32	1990	2
5	sdgd	dfgdvbd	3	12	2010	325
546	gved	uhfyu	45	43	1979	122
4	bfhg	tby	34	74	1948	34
456	jyg	sgrdf	5	33	1989	12
45	dsfgsd	sdf-dsf	346	39	1983	32
4564	nmgj	dfsdf	54	24	1998	5
4555	sdf	dfgdf	752	2	2020	5
544	sd	ssdre	345	27	1995	5

id	first_name	last_name	account_id	age	birth_year	address_id
	sdf	fg	354	32	1990	2
5	sdgd	dfgdvbd	3	12	2010	325
546	gved	uhfyu	45	43	1979	122
4	bfhg	tby	34	74	1948	34
456	jyg	sgrdf	5	33	1989	12
45	dsfgsd	sdf-dsf	346	39	1983	32
4564	nmgj	dfsdf	54	24	1998	5
4555	sdf	dfgdf	752	2	2020	5
544	sd	ssdre	345	27	1995	5

id	first_name	last_name	account_id	age	birth_year	address_id
	sdf	fg	354	32	1990	2
5	sdgd	dfgdvbd	3	12	2010	325
546	gved	uhfyu	45	43	1979	122
4	bfhg	tby	34	74	1948	34
456	jyg	sgrdf	5	33	1989	12
45	dsfgsd	sdf-dsf	346	39	1983	32
4564	nmgj	dfsdf	54	24	1998	5
4555	sdf	dfgdf	752	2	2020	5
544	sd	ssdre	345	27	1995	5

2

Which functions?

- COUNT
  - The number of rows that match
- AVG
  - The mean average numeric value
- MIN
  - The smallest value specified
- MAX
  - The largest value specified
- SUM
  - The total of all values in the column

3

How?

- All the aggregate functions take a single parameter and return a single value
- AVG/MAX/MIN/SUM each take the column, the results of which are analyzed or operated on.
- COUNT takes **ONLY** the * as its parameter as it returns the number of rows in the the query that are returned
  - On its own just rows in table
  - Especially useful with the GROUP BY clause for group counts

4

Extra Clauses

GROUP BY

- **REQUIRED** for ALL columns not linked to an aggregate function
- Columns separated by commas
- Extra columns invalidate results
- Order is not required but follow the SELECT clause
- Written after the FROM/WHERE clauses
- Executes BEFORE the SELECT clause

HAVING

- Optional
- Used to limit rows based on results of an aggregate function (like WHERE)
- Written after the GROUP BY clause
- Executes before the SELECT clause

5

When?

- If more columns than what the aggregate function is called on, the required GROUP BY clause is going to execute **AFTER** the FROM (of course) and WHERE clauses, but **BEFORE** the SELECT clause since the coalesced column values need to be grouped together (lol)
- The optional HAVING clause also occurs **BEFORE** the SELECT clause since this is how the now coalesced results are then filtered before becoming available to the SELECT clause

6

Execution order

1. FROM
   - Pick the table(s)
2. WHERE
   - Limit results based on other value(s)
3. GROUP BY
   - Identify and isolate based on column values
4. HAVING
   - Limit results based on result of aggregate
5. SELECT
   - Pick the columns to display

7

Execution order of a query

```
4 SELECT
      section_id,
      COUNT(*) AS enroll_count
1 FROM
      enrollment
2 GROUP BY
      section_id
3 HAVING
      COUNT(*) > 4
5 ORDER BY
      COUNT(*),
      section_id
  ;
```

8

Demo Query

Query

```
Worksheet  Query Builder
1 SELECT
2     section_id
3 FROM
4     enrollment;
```

Results

```
SECTION_ID
----------
        99
       101
        99
       101
        99
       101

226 rows selected.
```

9

Limit query with DISTINCT

Distinct Query

```
Worksheet  Query Builder
 6 SELECT DISTINCT
 7       section_id
 8 FROM
 9       enrollment;
10
```

Distinct Results

```
SECTION_ID
----------
       102
       100
       147
       117
       123
        92
       150
       138
        91

64 rows selected.
```

10

Use Aggregate Function

Aggregate Query

```
Worksheet  Query Builder
11 SELECT
12       section_id,
13       COUNT(*) AS enroll_count
14 FROM
15       enrollment
16 GROUP BY
17       section_id;
18
```

Aggregate Results

```
SECTION_ID ENROLL_COUNT
---------- ------------
       148            5
       149            1
       150            3
       151            2
       152            4
       153            3
       154            4
       155            5
       156            8

64 rows selected.
```

11