

# Aggregate Functions

CS 2550 Database Design 2022  
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	email	phone_number	hire_date	job_id	salary	commission_pct
1	Jeffrey	Deane	jeff@oracle.com	515.122.4444	1997-09-01	AD_ASST	3000	0
2	Timothy	Gietz	tim@oracle.com	515.122.6055	1997-09-01	AD_ASST	2400	0
3	Alexander	Toth	alex@oracle.com	515.122.5588	1997-09-01	AD_ASST	3000	0
4	Bruce	Smith	bruce@oracle.com	515.122.3456	1997-09-01	AD_ASST	2400	0
5	David	Adams	david@oracle.com	515.122.9876	1997-09-01	AD_ASST	2400	0
6	John	Chung	john@oracle.com	515.122.7890	1997-09-01	AD_ASST	2400	0
7	John	Chung	john@oracle.com	515.122.7890	1997-09-01	AD_ASST	2400	0
8	John	Chung	john@oracle.com	515.122.7890	1997-09-01	AD_ASST	2400	0
9	John	Chung	john@oracle.com	515.122.7890	1997-09-01	AD_ASST	2400	0
10	John	Chung	john@oracle.com	515.122.7890	1997-09-01	AD_ASST	2400	0

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
- ▶ Order is not required but follow the SELECT clause
- ▶ Written after the FROM/WHERE clauses
- ▶ Executes **BEFORE** the WHERE and/or SELECT clauses

#### HAVING

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

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 **BEFORE** the WHERE/SELECT clauses since the coalesced column values need to be grouped together (lol)
- ▶ The optional HAVING clause also occurs **BEFORE** the WHERE/SELECT clauses since this is how the now coalesced results are then filtered before becoming available to the SELECT/WHERE clauses

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

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

---

---

---

---

---

---

---

---