

RBD LAB

Aggregate Functions

Table 12.25 Aggregate Functions

Name	Description
<u>AVG ()</u>	Return the average value of the argument
<u>BIT AND ()</u>	Return bitwise AND
<u>BIT OR ()</u>	Return bitwise OR
<u>BIT XOR ()</u>	Return bitwise XOR
<u>COUNT ()</u>	Return a count of the number of rows returned
<u>COUNT (DISTINCT)</u>	Return the count of a number of different values
<u>GROUP CONCAT ()</u>	Return a concatenated string
<u>JSON ARRAYAGG ()</u>	Return result set as a single JSON array
<u>JSON OBJECTAGG ()</u>	Return result set as a single JSON object
<u>MAX ()</u>	Return the maximum value
<u>MIN ()</u>	Return the minimum value
<u>STD ()</u>	Return the population standard deviation
<u>STDDEV ()</u>	Return the population standard deviation
<u>STDDEV POP ()</u>	Return the population standard deviation
<u>STDDEV SAMP ()</u>	Return the sample standard deviation
<u>SUM ()</u>	Return the sum
<u>VAR POP ()</u>	Return the population standard variance
<u>VAR SAMP ()</u>	Return the sample variance
<u>VARIANCE ()</u>	Return the population standard variance

Examples:

```
mysql> SELECT student_name, AVG(test_score)
      FROM student
      GROUP BY student_name;
```

```
mysql> SELECT COUNT(DISTINCT results) FROM student;
```

```
mysql> SELECT student.student_name, COUNT(*)
      FROM student, course
      WHERE student.student_id=course.student_id
      GROUP BY student_name;
```

Union & Union All

UNION ALL command is equal to UNION command, except that UNION ALL selects all the values. The difference between Union and Union all is that **Union all will not eliminate duplicate rows**, instead it just pulls all the rows from all the tables fitting your query specifics and combines them into a table.

Examples:

```
mysql> SELECT 1, 2;
```

```
+---+---+
```

```
| 1 | 2 |
```

```
+---+---+
```

```
| 1 | 2 |
```

```
+---+---+
```

```
mysql> SELECT 'a', 'b';
```

```
+---+---+
```

```
| a | b |
```

```
+---+---+
```

```
| a | b |
```

```
+---+---+
```

```
mysql> SELECT 1, 2 UNION SELECT 'a', 'b';
```

```
+---+---+
```

```
| 1 | 2 |
```

```
+---+---+
```

```
| 1 | 2 |
```

```
| a | b |
```

```
+---+---+
```

Mysql references:

<https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html>

<https://dev.mysql.com/doc/refman/8.0/en/union.html>

Polecenia

Przygotować tabele z danymi

KIEROWCA

id	imie	nazwisko	kategoria	wiek	<i>miasto</i>
1	jan	kowal	B	28	<i>Gdańsk</i>
2	damian	dowbor	B	25	<i>Przejdowo</i>
3	justyna	lis	B	30	<i>Gdynia</i>
4	pawel	cis	C	40	<i>Gdańsk</i>
5	paulina	wilk	A	50	<i>Wejcherowo</i>
6	pawel	lis	B	20	<i>Reda</i>

POJAZD

id	marka	model	nr_rej	id_kierowcy
1	vw	polo	GD 1234s	1
2	vw	golf	GD 1884s	2
3	audi	q7	GDA 652mn	1
4	mazda	cx-3	GA LOPEK	3
5	renault	clio	GD 1234s	NULL

PASAZER

id	imie	nazwisko	id_pojazdu	<i>miasto</i>
1	kaziu	kowal	1	<i>Gdańsk</i>
2	gawel	cis	2	<i>Warszawa</i>
3	ola	bodo	3	<i>Gdynia</i>
4	<i>ala</i>	<i>kowalska</i>	1	<i>Gdańsk</i>
5	<i>pawel</i>	<i>lis</i>	2	<i>Reda</i>

Przygotować zapytania SQL odpowiadające na pytania”

1. Ilu jest kierowców?
2. Ilu kierowców ma kategorię B?
3. Średni wiek kierowców?
4. Średni wiek kierowców z kategorią B?
5. Średni wiek kierowców per kategoria prawa jazdy?
6. Średni wiek kierowców per marka pojazdu?
7. Liczba pasażerów per marka pojazdu?
8. Liczba pasażerów per marka pojazdu i kierowca?
9. Ile łącznie lat mają kierowcy z poszczególnych miast?
10. Najstarszy kierowca per miasto?
11. Najmłodszy kierowca?
12. Którzy kierowcy wiozą pasażerów z tego samego miasta co miasto kierowcy i ilu ich wiozą?
13. Którymi markami pojazdów kierowcy wożą więcej niż jednego pasażera?
14. Pełna lista kierowców i pasażerów (imie, nazwisko miasto).
15. Unikalna lista kierowców i pasażerów (imie, nazwisko miasto).