Aggregations

Consider the case of sports tournaments like cricket. Players' performances are analysed based on their batting average, maximum number of sixes hit, the least score in a tournament, etc.

We perform aggregations in such scenarios to combine multiple values into a single value, i.e., individual scores to an average score.

Let's learn more about aggregations to perform insightful analysis using the following database.

Database

The database consists of a player_match_details table that stores the information of players' details in a match like name, match, score, year, number of fours and sixes scored.

Schema

SQL

```sql
player_match_details (
  name VARCHAR(250),
  match VARCHAR(10),
  score INTEGER,
  fours INTEGER,
  sixes INTEGER,
  year INTEGER
);
```

Aggregation Functions

Combining multiple values into a single value is called aggregation. Following are the functions provided by SQL to perform aggregations on the given data:

Aggregate Functions	Description
COUNT	Counts the number of values
SUM	Adds all the values
MIN	Returns the minimum value
MAX	Returns the maximum value

Aggregate Functions	Description
AVG	Calculates the average of the values

Syntax

SQL

```sql
SELECT
  aggregate_function(c1),
  aggregate_function(c2)
FROM
  TABLE;
```

Note

We can calculate multiple aggregate functions in a single query.

Examples

1. Get the total runs scored by "Ram" from the `player_match_details` table.

name	match	score	fours	sixes	year
Ram	CSK vs DD	55	4	2	2011
Joseph	MI vs RR	58	4	2	2012
Lokesh	RCB vs KKR	55	3	4	2013
David	MI vs RR	63	4	2	2014
Viraj	KKR vs SRH	72	6	4	2010
Shyam	SRH vs RCB	62	3	2	2011
Stark	RR vs SRH	54	5	1	2012
Ram	CSK vs SRH	59	4	4	2013
Joseph	SRH vs RR	52	3	2	2014

SQL

```sql
SELECT
```

```sql
  SUM(score)
FROM
  player_match_details
WHERE
  name = "Ram";
```

Output

SUM(score)
221

2. Get the highest and least scores among all the matches that happened in the year 2011.

SQL

```sql
SELECT
  MAX(score),
  MIN(score)
FROM
  player_match_details
WHERE
  year = 2011;
```

Output

MAX(score)	MIN(score)
75	62

COUNT Variants

- Calculate the total number of matches played in the tournament.

Variant 1

SQL

```sql
SELECT COUNT(*)
FROM player_match_details;
```

Variant 2

SQL

```sql
SELECT COUNT(1)
```

```
2    FROM player_match_details;
```

Variant 3

SQL

```sql
SELECT COUNT()
FROM player_match_details;
```

Output of Variant 1, Variant 2 and Variant 3

All the variants, i.e.,

`Variant 1` , `Variant 2` and `Variant 3` give the same result: `18`

Special Cases

- When `SUM` function is applied on non-numeric data types like strings, date, time, datetime etc., `SQLite` DBMS returns `0.0` and `PostgreSQL` DBMS returns `None` .
- Aggregate functions on strings and their outputs

Aggregate Functions	Output
MIN, MAX	Based on lexicographic ordering
SUM, AVG	0 (depends on DBMS)
COUNT	Default behavior

- `NULL` values are ignored while computing the aggregation values
- When aggregate functions are applied on only

  `NULL` values:

Aggregate Functions	Output
MIN	NULL
MAX	NULL
SUM	NULL
COUNT	0
AVG	NULL

Alias

Using the keyword

`AS` , we can provide alternate temporary names to the columns in the output.

Syntax

SQL

```sql
SELECT
  c1 AS a1,
  c2 AS a2,
  ...
FROM
  table_name;
```

Examples

- Get all the names of players with column name as "player_name".

SQL

```sql
SELECT
  name AS player_name
FROM
  player_match_details;
```

Output

player_name
Ram
Joseph

- Get the average score of players as "avg_score".

SQL

```sql
SELECT
  AVG(score) AS avg_score
FROM
  player_match_details;
```

Output

avg_score
60

Try it Yourself!

- Get the average score of "Ram" in the year 2011.
- Get the least score among all the matches.
- Get the highest score among the scores of all players in 2014.
- Get the total number of sixes hit as `sixes_hit`