**Queries:**

* Substring(name of string , index*(positive for start and negative for end)*, length*)*
* Limit query can be used only once
* Replace function can be used for integer values as well as string values
* For reserved keywords used as column use backticks **EX:** `select` , `from`
* **rank() vs dense\_rank() ->**  rank function is used to assign a rank while skipping the rank if the corresponding values are same. While dense\_rank() function is used to assign continuous value and same rank for same values.
* **Syntax:** rank() over (order by column\_name asc/desc)
* **Syntax:** dense\_rank() over (order by column\_name asc/desc)
* **Limit** and **offset** are used with each other- **Limit** is used to limit the amount of rows returned and **offset** is used to specify the number of rows to skip before starting.

**Syntax:** SELECT column1, column2 FROM table **LIMIT** number\_of\_rows **OFFSET** offset\_value;

* **Having** clause is used because aggregate functions cannot be used in **where** clause
* **SQL query execution:**

1. From
2. Where
3. Group by
4. Having
5. Select
6. Order by

* To group a date by month and year together,

**Syntax:** select **date\_format( column\_name , '%Y-%m')** as month from table\_name group by **month**

* For performing cumulative\_sum over a specific column

**Syntax**: select SUM(**column\_name**) OVER (ORDER BY **order\_column**) AS **cumulative\_sum**

**Practice Questions:**

180. Consecutive Numbers

1204. Last Person to Fit in the Bus