<u>Table & column aliases</u> – describes how to use table and column aliases in the query.

### PostgreSQL as clause statement syntax

## Column alias

```
SELECT column name AS alias name FROM table;
```

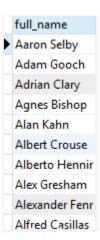
In this syntax, the <code>column\_name</code> is assigned an alias as <code>alias\_name</code>. The <code>AS</code> keyword is optional so you can skip it as shown below:

```
SELECT column_name alias_name FROM table;
```

### Example:

To make the query short and its output more meaningful, you can use the column aliases as follows:

SELECT first\_name ||''|| last\_name AS full\_name FROM customer ORDER BY full\_name
;



<u>Table & column aliases</u> – describes how to use table and column aliases in the query.

## Table alias

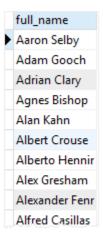
The following illustrates the syntax of the table alias:

```
SELECT column list FROM table name AS alias name;
```

# Example:

To make the query short and its output more meaningful, you can use the column or table aliases as follows:

SELECT c.first\_name | | ' ' | | c.last\_name AS full\_name FROM customer c ORDER BY c.full\_name;



[note: you can also use aliases with multiple table or with inner or etc. join ]