The `query_attributes` component accesses query attributes to implement a `mysql_query_attribute_string()` function. See Section 5.5.4, “Query Attribute Components”.

To uninstall the `query_attributes` component, use this statement:

```
UNINSTALL COMPONENT "file://component_query_attributes";
```

`UNINSTALL COMPONENT` unloads the component, and unregisters it from the `mysql.component` system table to cause it not to be loaded during subsequent server startups.

Because installing and uninstalling the `query_attributes` component installs and uninstalls the `mysql_query_attribute_string()` function that the component implements, it is not necessary to use `CREATE FUNCTION` or `DROP FUNCTION` to do so.

Query Attribute Loadable Functions

- `mysql_query_attribute_string(`*`name`*`)`

  Applications can define attributes that apply to the next query sent to the server. The `mysql_query_attribute_string()` function, available as of MySQL 8.0.23, returns an attribute value as a string, given the attribute name. This function enables a query to access and incorporate values of the attributes that apply to it.

  `mysql_query_attribute_string()` is installed by installing the `query_attributes` component. See Section 9.6, “Query Attributes”, which also discusses the purpose and use of query attributes.

  Arguments:

  - *`name`*: The attribute name.

  Return value:

  Returns the attribute value as a string for success, or `NULL` if the attribute does not exist.

  Example:

  The following example uses the `mysql` client `query_attributes` command to define query attributes that can be retrieved by `mysql_query_attribute_string()`. The `SELECT` shows that retrieving a nonexistent attribute (`n3`) returns `NULL`.

```
mysql> query_attributes n1 v1 n2 v2;
mysql> SELECT
    ->   mysql_query_attribute_string('n1') AS 'attr 1',
    ->   mysql_query_attribute_string('n2') AS 'attr 2',
    ->   mysql_query_attribute_string('n3') AS 'attr 3';
+--------+--------+--------+
| attr 1 | attr 2 | attr 3 |
+--------+--------+--------+
| v1     | v2     | NULL   |
+--------+--------+--------+
```

9.7 Comments

MySQL Server supports three comment styles:

- From a `#` character to the end of the line.
- From a `--` sequence to the end of the line. In MySQL, the `--` (double-dash) comment style requires the second dash to be followed by at least one whitespace or control character (such as a space, tab,