

sqlalchemy IS NOT NULL select

Asked 8 years, 11 months ago Modified 1 year, 3 months ago Viewed 171k times

How can I add the filter as in SQL to select values that are NOT NULL from a certain column ?

```
172 SELECT *
FROM table
WHERE YourColumn IS NOT NULL;
```


How can I do the same with SQLAlchemy filters?

```
select = select(table).select_from(table).where(all_filters)
```


python sqlalchemy

Share Edit Follow

edited Feb 14, 2014 at 16:49

 Alexander
3,078 2 18 32

asked Feb 14, 2014 at 16:44

 salamey
3,493 10 36 70

What is `all_filters` here? Why the `select_from` ? – Martijn Pieters ♦ Feb 14, 2014 at 16:45

And do you have a table definition or do you need to use column literals? – Martijn Pieters ♦ Feb 14, 2014 at 16:45

3 Answers

Sorted by:

Highest score (default) ▾

`column_obj != None` will produce a [IS NOT NULL constraint](#):

214 In a column context, produces the clause `a != b`. If the target is `None`, produces a `IS NOT NULL`.

or use `is_not()`:

Implement the `IS NOT` operator.

Normally, `IS NOT` is generated automatically when comparing to a value of `None`, which resolves to `NULL`. However, explicit usage of `IS NOT` may be desirable if comparing to boolean values on certain platforms.

Demo:

```
>>> from sqlalchemy.sql import column
>>> column('YourColumn') != None
```

```
<sqlalchemy.sql.elements.BinaryExpression object at 0x10f81aa90>
>>> print(column('YourColumn') != None)
"YourColumn" IS NOT NULL
>>> column('YourColumn').is_not(None)
<sqlalchemy.sql.elements.BinaryExpression object at 0x11081edf0>
>>> print(column('YourColumn').is_not(None))
"YourColumn" IS NOT NULL
```

You can't use `is not None` here, because the [is not object identity inequality test](#) can't be overloaded the way `!=` can; you'll just get `True` instead as a `ColumnClause` instance is not the same object as the `None` singleton:

```
>>> column('YourColumn') is not None
True
```

*) The method was formerly named `isnot()` and was renamed in SQLAlchemy 1.4. The old name is still available for backwards compatibility.

Share Edit Follow

edited Sep 3, 2021 at 10:00

answered Feb 14, 2014 at 16:46



[Martijn Pieters](#) ♦

1.0m 283 3974
3297

Starting in version 0.7.9 you can use the filter operator `.isnot` instead of comparing constraints, like this:

```
query.filter(User.name.isnot(None))
```

This method is only necessary if pep8 is a concern.

source: [sqlalchemy documentation](#)

Share Edit Follow

edited Sep 2, 2016 at 17:16

answered May 12, 2016 at 20:30



[Josh](#)

1,266 2 19 28



[Filipe Spindola](#)

1,795 1 14 17

5 Beyond making pep8 happy, I think this is a better solution because `NULL` is not a valid as the RHS of `!=` in SQL and using `isnot` better conveys your intentions for what you want the generated statement to look like. – [Josh](#) Sep 2, 2016 at 16:04

2 @Josh: SQLAlchemy won't emit `!= NULL` however, even if you use `column != None` on the Python side; you get `IS NOT NULL`. Using `.isnot()` lets you *force* `IS NOT` on other types however (think `.isnot(True)` against boolean columns, for example). – [Martijn Pieters](#) ♦ Sep 15, 2016 at 6:23

In case anyone else is wondering, you can use `is_` to generate `foo IS NULL`:

```
>>> from sqlalchemy.sql import column
>>> print(column('foo').is_(None))
foo IS NULL
>>> print(column('foo').isnot(None))
```



foo IS NOT NULL



Share Edit Follow

edited Sep 10, 2021 at 16:48

answered Apr 15, 2017 at 3:54



Ahmed

2,764 1 24 39



Matthew Moisen

15.7k 26 115 219

7 Thanks, this is what I was looking for but Google sent me here! – Sinister Beard Jan 30, 2020 at 10:43
