

BiJoe - Bug #47766

ProgrammingError: syntax error at or near "{"

16 Oct 2020 08:51 AM - sentry io

Status:	Solution proposée	Start date:	16 Oct 2020
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Planning:	No
Patch proposed:	Yes		

Description

<https://sentry.entrouvert.org/entrouvert/publik/issues/5804/>

```
ProgrammingError: syntax error at or near "{"
LINE 1: ...ormdata.receipt_time = "receipt_time".date where ({{fact_tabl...
                                                ^

(7 additional frame(s) were not displayed)
...
File "rest_framework/views.py", line 492, in dispatch
  response = handler(request, *args, **kwargs)
File "bijoe/visualization/views.py", line 334, in get
  x_axis, grid = visualization.table_1d()
File "bijoe/visualization/utils.py", line 209, in table_1d
  data = self.data()
File "bijoe/visualization/utils.py", line 162, in data
  [self.measure]))
File "bijoe/engine.py", line 441, in query
  cursor.execute(sql)
```

History

#1 - 16 Oct 2020 08:51 AM - Lauréline Guerin

- Project changed from *Suivi des traces* to *BiJoe*

#2 - 05 Nov 2020 04:49 PM - Nicolas Roche

Sans chercher plus pour l'instant,

est-ce que ça ne viendrait pas de là : `/bijoe/engine.py`

```
class SchemaJSONDimension(schemas.Dimension):
    '''Generated dimensions for JSON fields keys'''
    ...
    def __init__(self, json_field, name):
    ...
        sql = ('SELECT DISTINCT {json_field}->>\'%s\' AS v, {json_field}->>\'%s\' AS v'
              ' FROM {{fact_table}} WHERE ({{json_field}->>\'%s\' ) IS NOT NULL ORDER BY v' %
              (self.name, self.name, self.name))
        ...
        self.filter_expression = ('{{fact_table}}.id IS NULL ' ...
```

où l'on voit que `fact_table` est d'abord utilisé avec double moustache, puis où l'on retrouve la chaîne du message d'erreur avec `'{fact_table}'` ?

#3 - 05 Nov 2020 06:18 PM - Nicolas Roche

Mince, je n'ai rien dit : on utilise ensuite 'a{var}b'.format(var='foo') et donc c'est normal d'avoir des simples moustaches.

Mais du coup, je ne comprend pas comment la chaîne sql a pu être exécutée sans avoir été remplacée.

(le remplacement dans la chaîne est fait par un appel juste avant le plantage, et la fonction appelée ne me semble pas pouvoir esquiver l'instruction de remplacement)

```
def sql_query(self, filters, drilldown, measures, **kwargs):
    ...
    sql = sql.format(fact_table=self.cube.fact_table, ...
    return sql

def query(self, filters, drilldown, measures, **kwargs):
    sql = self.sql_query(filters=filters, drilldown=drilldown, measures=measures, **kwargs)
    cursor.execute(sql)
```

Par exemple il y a ce test qui me semble rejouer cela :

```
$ tox -e py3-dj22 -- tests/test_schema1.py::test_json_dimensions
```

#4 - 05 Nov 2020 06:50 PM - Benjamin Dauvergne

La requête fonctionne maintenant, <https://statistiques.services.metzmetropole.fr/visualization/120/json/> je ne passerai pas trop de temps là dessus.

#5 - 05 Nov 2020 08:37 PM - Benjamin Dauvergne

Bon ça vient de là :

```
sql = sql.format(fact_table=self.cube.fact_table,
                 table_expression=table_expression,
                 where_conditions=where_conditions)
```

where_condition peut contenir {fact_table}, et peu de requêtes utilisent where_condition récursivement ; normalement il est déjà intégré dans sql un peut avant :

```
if where:
    where_conditions = ' AND '.join(where)
    sql += ' WHERE %s' % where_conditions
```

ça vient de ce schéma sur metz :

```
{
  "expression": "case (select count({fact_table}.id) from {table_expression} where {where_conditions})
when 0 then null else count({fact_table}.id) * 100. / (select count({fact_table}.id) from {table_expression}
where {where_conditions}) end",
  "label": "pourcentage des demandes",
  "name": "percent",
  "type": "percent"
},
```

les pourcentages n'ont jamais du fonctionner associé à un filtrage sur un champ JSON.

Il faut vraisemblablement ajouter un `where_condition = where_condition.format(fact_table=self.cube.fact_table)`.

Possible de faire un test en choisissant une dimension json et une mesure de pourcentage (voir `test_schema1.py` utiliser `measure=percent` et `drilldown_x=a`).

#6 - 06 Nov 2020 02:27 PM - Nicolas Roche

- File `0001-engine-format-where-sql-condition-string-47766.patch` added
- Status changed from *Nouveau* to *Solution proposée*
- Patch proposed changed from *No* to *Yes*

Merci Benjamin, je suis encore passé à côté.

Files

0001-engine-format-where-sql-condition-string-47766.patch	2.96 KB	06 Nov 2020	Nicolas Roche
---	---------	-------------	---------------