pgvocalizer

Vocalizer for Postgres query plan

Installation

Tested on Windows 10 and Python 3.6

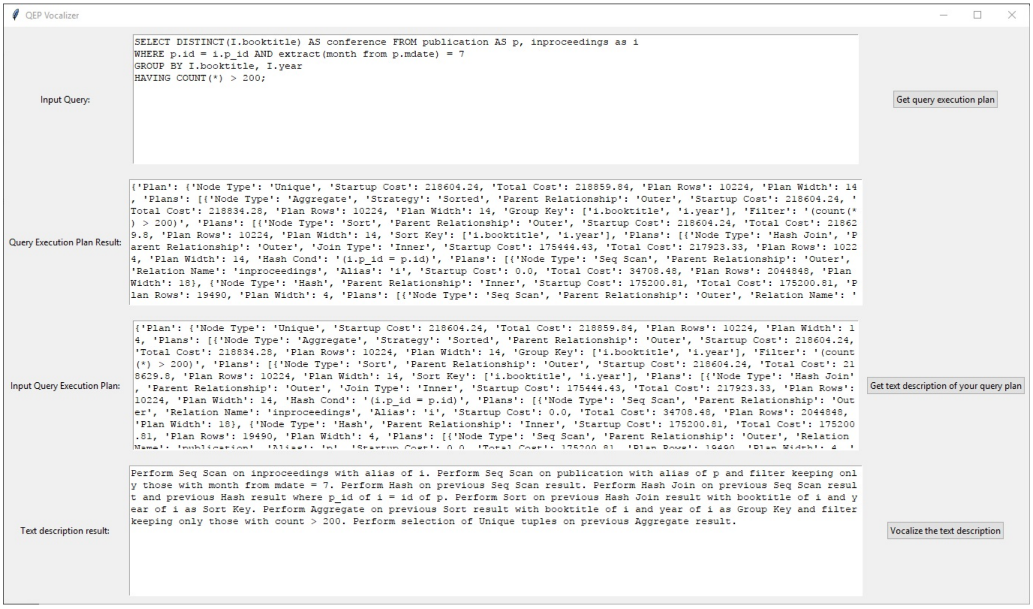
1. Install the pip requirements

```
$ pip install -r requirements.txt
```

2. Edit the .env file and fill in all of database connection details
3. Run ui.py

```
$ python ui.py
```

Usage


QEP Vocalizer
Input Query:
Get query execution plan
Query Execution Plan Result:
Input Query Execution Plan:
Get text description of your query plan
Text description result:
Vocalize the text description

```
SELECT DISTINCT(I.booktitle) AS conference FROM publication AS p, inproceedings as i
WHERE p.id = i.p_id AND extract(month from p.mdate) = 7
GROUP BY I.booktitle, I.year
HAVING COUNT(*) > 200;
```

```
{'Plan': {'Node Type': 'Unique', 'Startup Cost': 218604.24, 'Total Cost': 218859.84, 'Plan Rows': 10224, 'Plan Width': 14
, 'Plans': [{'Node Type': 'Aggregate', 'Strategy': 'Sorted', 'Parent Relationship': 'Outer', 'Startup Cost': 218604.24, '
Total Cost': 218834.28, 'Plan Rows': 10224, 'Plan Width': 14, 'Group Key': ['i.booktitle', 'i.year'], 'Filter': '(count(*
) > 200)', 'Plans': [{'Node Type': 'Sort', 'Parent Relationship': 'Outer', 'Startup Cost': 218604.24, 'Total Cost': 21862
9.8, 'Plan Rows': 10224, 'Plan Width': 14, 'Sort Key': ['i.booktitle', 'i.year'], 'Plans': [{'Node Type': 'Hash Join', 'P
arent Relationship': 'Outer', 'Join Type': 'Inner', 'Startup Cost': 175444.43, 'Total Cost': 217923.33, 'Plan Rows': 1022
4, 'Plan Width': 14, 'Hash Cond': '(i.p_id = p.id)', 'Plans': [{'Node Type': 'Seq Scan', 'Parent Relationship': 'Outer',
'Relation Name': 'inproceedings', 'Alias': 'i', 'Startup Cost': 0.0, 'Total Cost': 34708.48, 'Plan Rows': 2044848, 'Plan
Width': 18}, {'Node Type': 'Hash', 'Parent Relationship': 'Inner', 'Startup Cost': 175200.81, 'Total Cost': 175200.81, 'P
lan Rows': 19490, 'Plan Width': 4, 'Plans': [{'Node Type': 'Seq Scan', 'Parent Relationship': 'Outer', 'Relation Name': '
```

```
{'Plan': {'Node Type': 'Unique', 'Startup Cost': 218604.24, 'Total Cost': 218859.84, 'Plan Rows': 10224, 'Plan Width': 1
4, 'Plans': [{'Node Type': 'Aggregate', 'Strategy': 'Sorted', 'Parent Relationship': 'Outer', 'Startup Cost': 218604.24,
'Total Cost': 218834.28, 'Plan Rows': 10224, 'Plan Width': 14, 'Group Key': ['i.booktitle', 'i.year'], 'Filter': '(count
(*) > 200)', 'Plans': [{'Node Type': 'Sort', 'Parent Relationship': 'Outer', 'Startup Cost': 218604.24, 'Total Cost': 21
8629.8, 'Plan Rows': 10224, 'Plan Width': 14, 'Sort Key': ['i.booktitle', 'i.year'], 'Plans': [{'Node Type': 'Hash Join'
, 'Parent Relationship': 'Outer', 'Join Type': 'Inner', 'Startup Cost': 175444.43, 'Total Cost': 217923.33, 'Plan Rows':
10224, 'Plan Width': 14, 'Hash Cond': '(i.p_id = p.id)', 'Plans': [{'Node Type': 'Seq Scan', 'Parent Relationship': 'Out
er', 'Relation Name': 'inproceedings', 'Alias': 'i', 'Startup Cost': 0.0, 'Total Cost': 34708.48, 'Plan Rows': 2044848,
'Plan Width': 18}, {'Node Type': 'Hash', 'Parent Relationship': 'Inner', 'Startup Cost': 175200.81, 'Total Cost': 175200
.81, 'Plan Rows': 19490, 'Plan Width': 4, 'Plans': [{'Node Type': 'Seq Scan', 'Parent Relationship': 'Outer', 'Relation
Name': 'publication', 'Alias': 'p', 'Startup Cost': 0.0, 'Total Cost': 175200.81, 'Plan Rows': 19490, 'Plan Width': 4, '
```

```
Perform Seq Scan on inproceedings with alias of i. Perform Seq Scan on publication with alias of p and filter keeping onl
y those with month from mdate = 7. Perform Hash on previous Seq Scan result. Perform Hash Join on previous Seq Scan resul
t and previous Hash result where p_id of i = id of p. Perform Sort on previous Hash Join result with booktitle of i and y
ear of i as Sort Key. Perform Aggregate on previous Sort result with booktitle of i and year of i as Group Key and filter
keeping only those with count > 200. Perform selection of Unique tuples on previous Aggregate result.
```

- **Input Query**: You can input your query here to generate the QEP in JSON format
- **Query Execution Plan Result**: The QEP in JSON format from query above will be displayed here
- **Input Query Execution Plan**: Input the QEP in JSON format here, you can copy paste from the **Query Execution Plan Result** *OR* get it from querying Postgres using:

```
EXPLAIN (FORMAT JSON) [your_query_here]
```

- **Text description result**: The natural language translated of the QEP is displayed here.