Auditing for failed queries

Last updated by | Lisa Liu | Nov 6, 2020 at 10:36 AM PST

One of the customers had a requirement with pgaudit where they wanted to log failed queries as well as the error messages for Azure Database for PostgreSQL. They also had some concerns around object_name/object_type details missing, also missing tableName, schemaName in the JSON logs in the storage account. They were sending the pgaudit logs to an Azure Storage account where they were in the form of JSON files.

First thing to note here is – Any log from pgaudit will be in the “message” column in the JSON files and there should not be any missing info. The logs look the same as on-premises since we use the same extensions, there is nothing that is Azure specific here. Now coming to the table format which has the table name, schema name, it is created to be compatible with Azure Diagnostics which is the log repository for all the logs for SQL/MySQL/PostgreSQL. Hence it is not necessary that it is filled for the pgaudit part.

Refer this doc for more info about the columns: https://docs.microsoft.com/en-us/azure/postgresql/concepts-server-logs

This is what the on-prem logs for postgresql look like for pgaudit which is pretty similar to what is seen on Azure PostgreSQL too

```
...
2019-06-26 14:51:12 UTC:192.0.2.7(44754):test1@postgres:[3547]:LOG:
AUDIT: SESSION,1,1,DDL,CREATE TABLE,,,CREATE TABLE test_table (id
int);,<not logged>

2019-06-26 14:51:18 UTC:192.0.2.7(44754):test1@postgres:[3547]:LOG:
AUDIT: SESSION,2,1,READ,SELECT,,,select * from test_table;,<not
logged>
...
```

This official blog https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/auditing-know-what-s-going-on-in-your-postgres-database/ba-p/885243 has sample data and logs for pgaudit for reference.

The reason why we do not see failed query on pgAudit Logs but only successful queries is that -

"The failure of the query is not an action, the issue of the query is the action, hence this is what is recorded by audit."

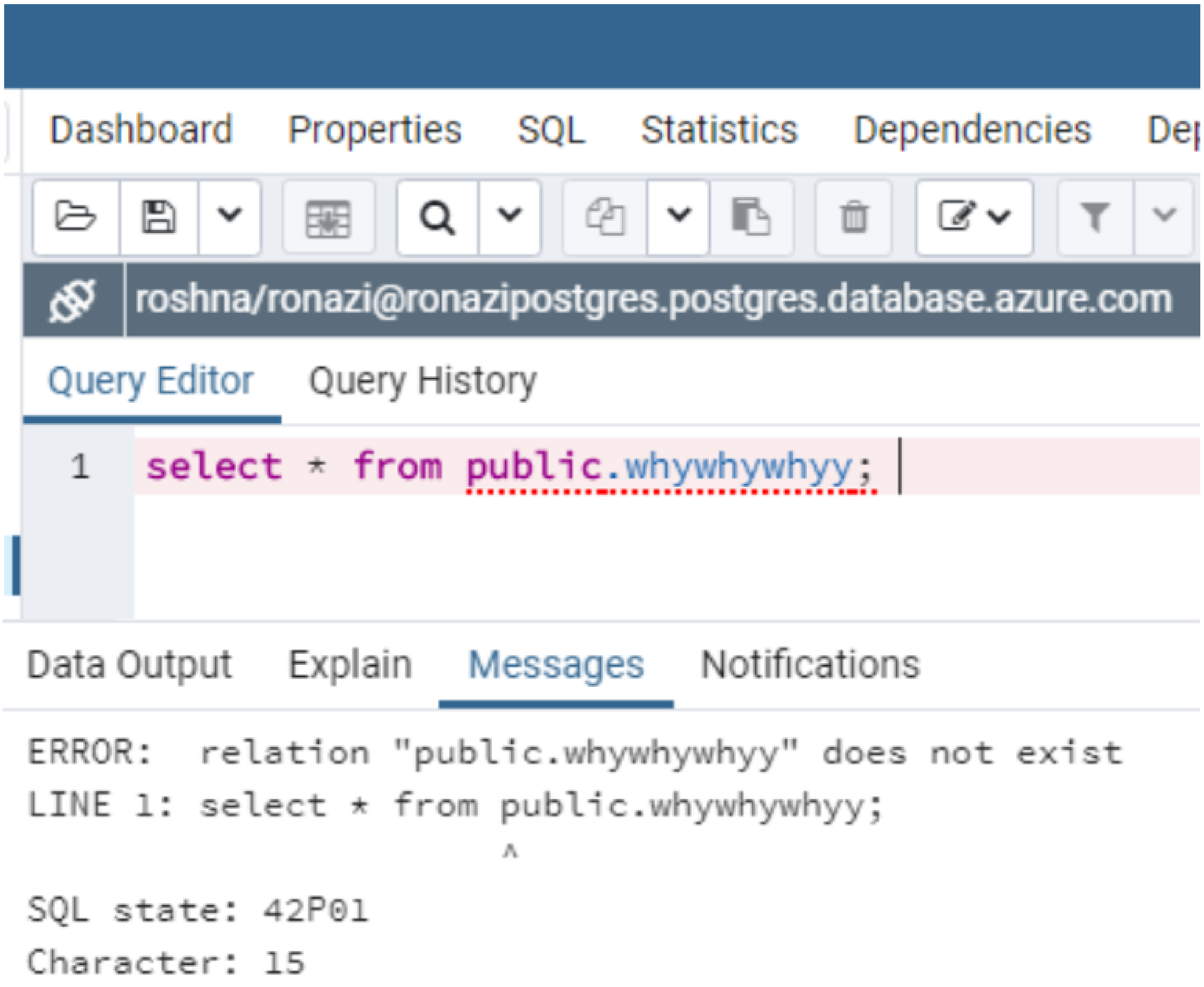
As a workaround for now, we can use PostgreSQL logs to populate the required data.

In addition to enabling the pgaudit extensions, if we want to audit the failed queries, we can modify the below parameters in PostgreSQL logs.

log_min_messages, log_min_error_statement, log_statement

Refer to this blog : https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/how-to-configure-postgres-log-settings/ba-p/1214716 for more details.

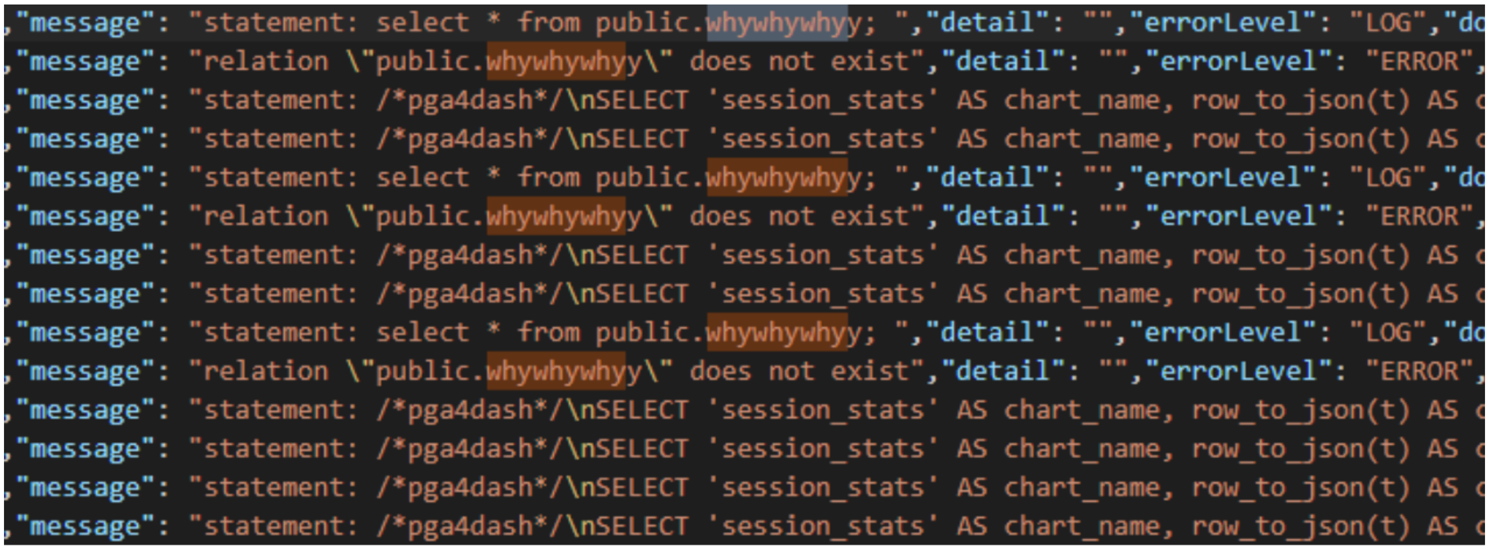
Example for a simple failed query log in JSON format:



```
select * from public.whywhywhyy;
```

```
ERROR:  relation "public.whywhywhyy" does not exist
LINE 1: select * from public.whywhywhyy;
                      ^

SQL state: 42P01
Character: 15
```



```
,"message": "statement: select * from public.whywhywhyy; ","detail": "","errorLevel": "LOG","do
,"message": "relation \"public.whywhywhyy\" does not exist","detail": "","errorLevel": "ERROR",
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
,"message": "statement: select * from public.whywhywhyy; ","detail": "","errorLevel": "LOG","do
,"message": "relation \"public.whywhywhyy\" does not exist","detail": "","errorLevel": "ERROR",
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
,"message": "statement: select * from public.whywhywhyy; ","detail": "","errorLevel": "LOG","do
,"message": "relation \"public.whywhywhyy\" does not exist","detail": "","errorLevel": "ERROR",
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
,"message": "statement: /*pga4dash*/\nSELECT 'session_stats' AS chart_name, row_to_json(t) AS c
```

If you want to reset all the earlier settings and start from scratch, we can do the below:

1)Reset all parameters to default on Portal. 2)Restart the server. 3)Set log_statement = ALL 4)Verify these parameters too: log_min_error_statement --> Error log_min_messages --> WARNING 5)Run the query for “table does not exist” 6)Wait for about 5 min 7)Check the PostgreSQL logs or the JSON logs in the storage account Another thing to note here is - It can also be possible that the query statement is present in a different JSON file than the one with the error. Postgres emits each log as a separate line. The error and the query that caused it are generated by Postgres as separate lines and they can show up as separate json objects.

You will not see a query with a syntax error select * public.koko_table; --causes a syntax error The wrong syntax query won’t appear in the logs as it does not hit the database server.

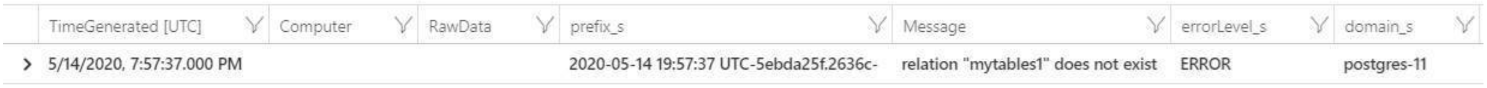
Scenario examples:

Scenario 1 : When running a failed query

I ran for example, a select query on a non-existent table :

select * from mytables1;

From log analytics I can see the below, no records about the failed query and it is not related to the audit logs:

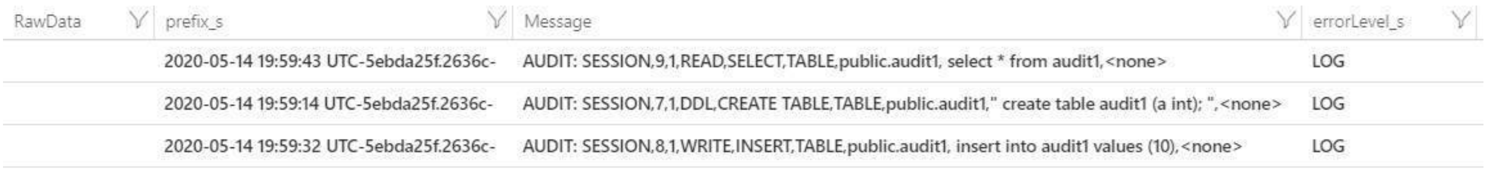


TimeGenerated [UTC]	Computer	RawData	prefix_s	Message	errorLevel_s	domain_s
5/14/2020, 7:57:37.000 PM			2020-05-14 19:57:37 UTC-5ebda25f.2636c-	relation "mytables1" does not exist	ERROR	postgres-11

And the same for any DML/DDL statements.

Scenario 2 : Successful query

Here we can see the queries from AUDIT :

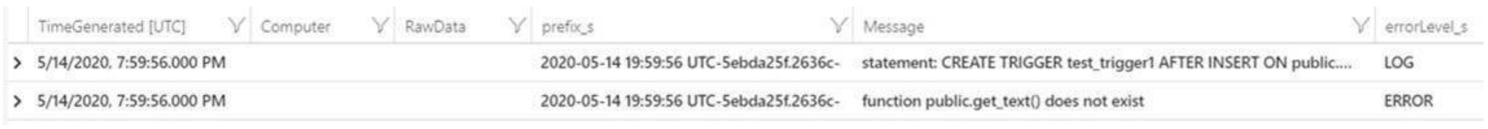


RawData	prefix_s	Message	errorLevel_s
	2020-05-14 19:59:43 UTC-5ebda25f.2636c-	AUDIT: SESSION,9,1,READ,SELECT,TABLE,public.audit1, select * from audit1,<none>	LOG
	2020-05-14 19:59:14 UTC-5ebda25f.2636c-	AUDIT: SESSION,7,1,DDL,CREATE TABLE,TABLE,public.audit1," create table audit1 (a int); ",<none>	LOG
	2020-05-14 19:59:32 UTC-5ebda25f.2636c-	AUDIT: SESSION,8,1,WRITE,INSERT,TABLE,public.audit1, insert into audit1 values (10),<none>	LOG

Hence in addition to enabling the audit extensions , if we want to audit the failed queries, we should modify the below parameters:

log_min_messages log_min_error_statement log_statement

Based on the blog : https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/how-to-configure-postgres-log-settings/ba-p/1214716 , as a test I changed the log_statement to DDL and I can see the failed DDL statement, and it is not generated by the AUDIT, but by the PostgreSQL server logs, so this should server as a workaround for the customer.



	TimeGenerated [UTC]	Computer	RawData	prefix_s	Message	errorLevel_s
>	5/14/2020, 7:59:56.000 PM			2020-05-14 19:59:56 UTC-5ebda25f.2636c-	statement: CREATE TRIGGER test_trigger1 AFTER INSERT ON public....	LOG
>	5/14/2020, 7:59:56.000 PM			2020-05-14 19:59:56 UTC-5ebda25f.2636c-	function public.get_text() does not exist	ERROR

How good have you found this content?