PERCONA

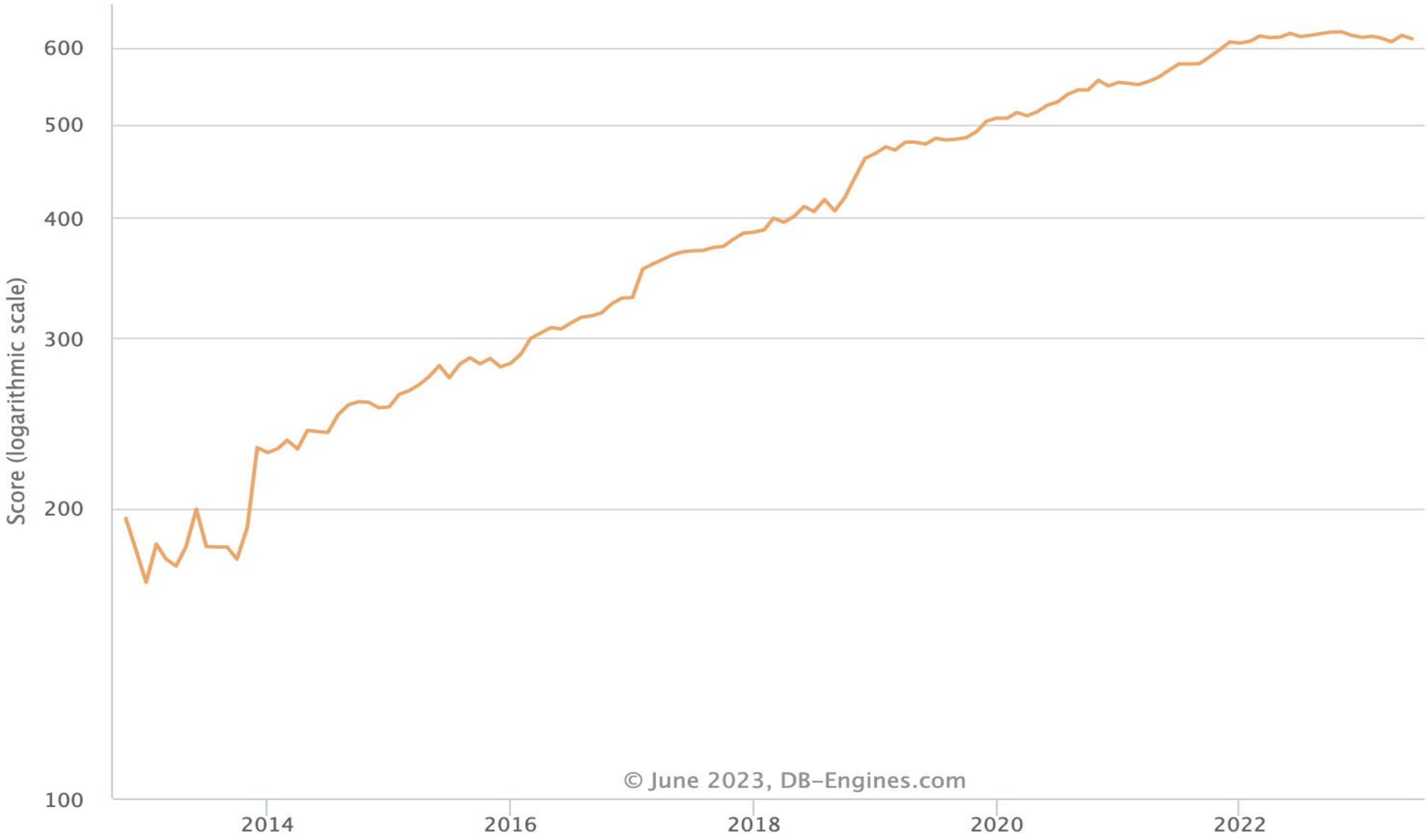
Why PostgreSQL is Becoming a Migration Target in Enterprises

Jobin Augustine
PostgreSQL Escalation Specialist

Agenda

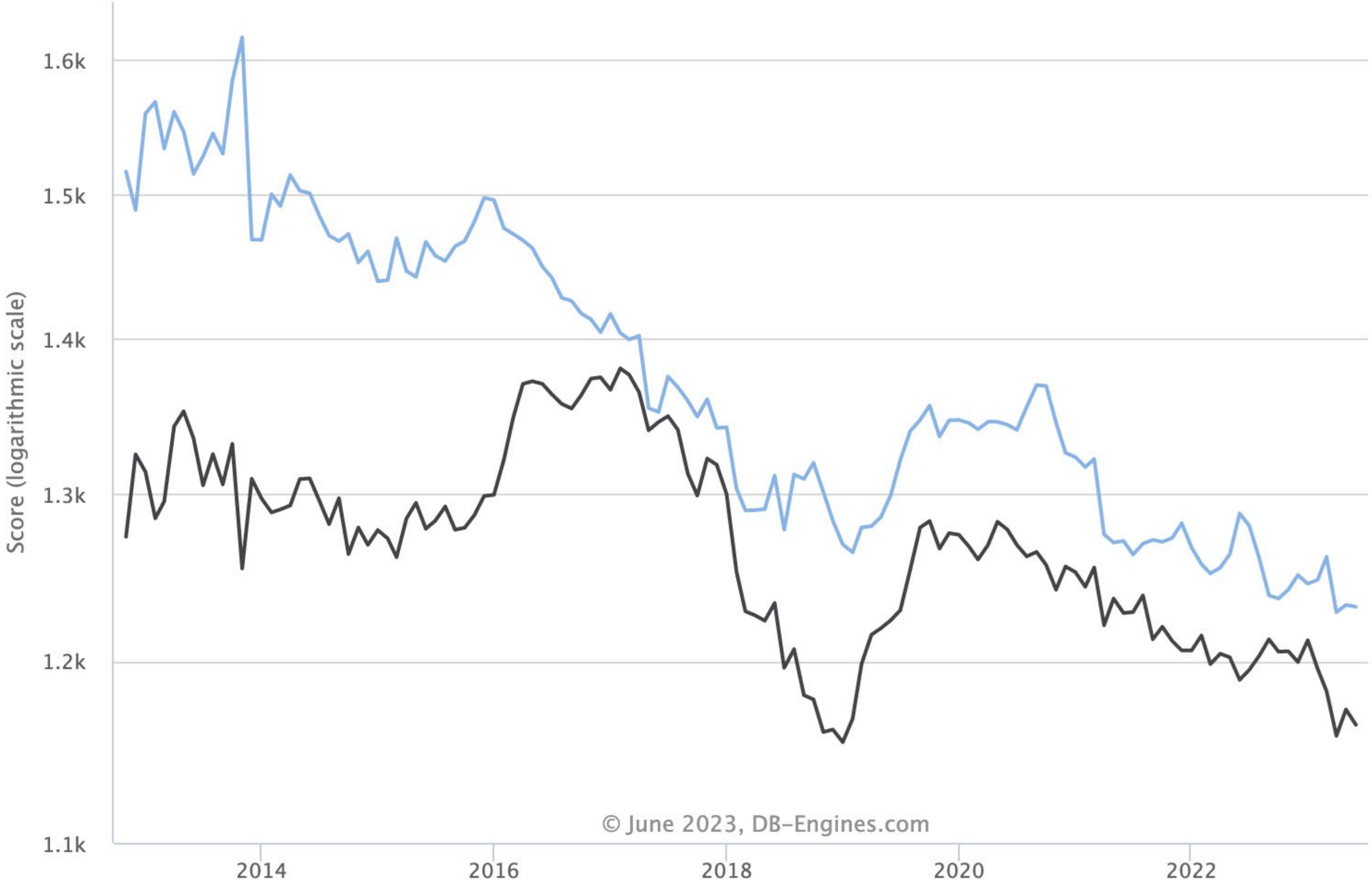
- Major Trends
- Reasons:
    - Executive/management
    - Developers
    - Operations
    - Architects
- Compatibility and Migrations
- Security
- Warning

What's Happening?



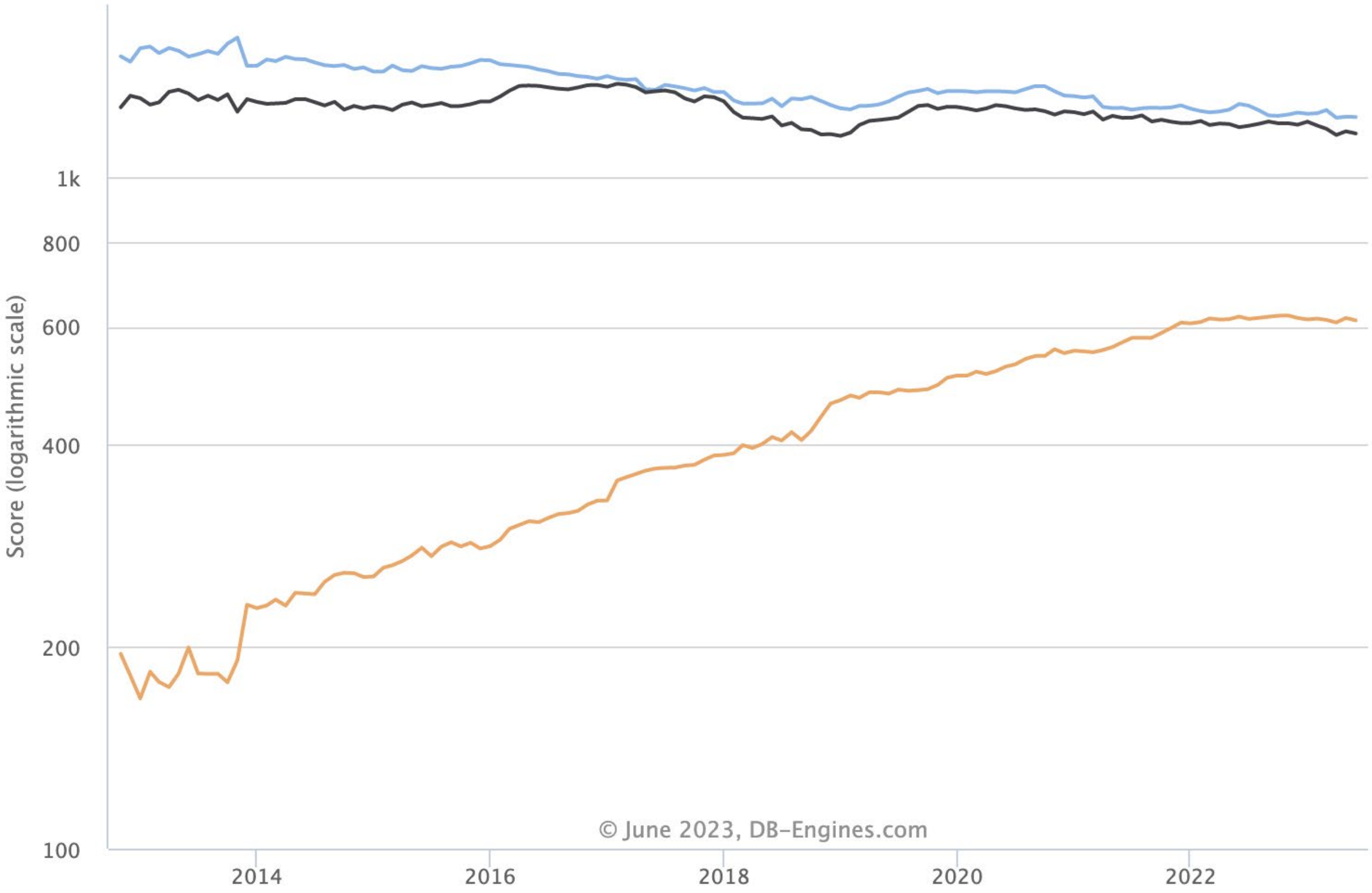
https://db-engines.com

What's Happening?



https://db-engines.com

What's Happening?



https://db-engines.com

Executive Management’s

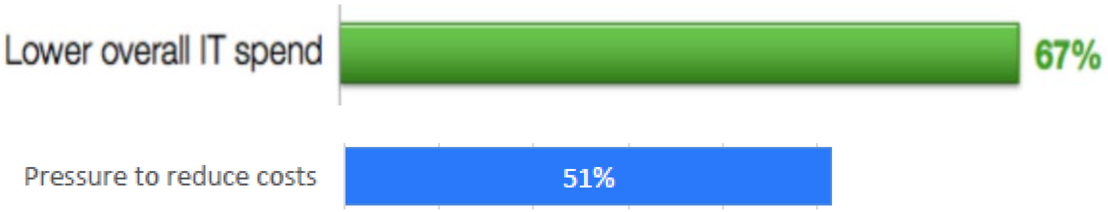
Reasons to choose PostgreSQL

What's happening

- Large number of migrations from mainly proprietary RDBMS
- Migrations from NoSQL / Document Stores
- **PostgreSQL as the default choice of database**
- Push from cloud vendors

TCO - Total Cost of Ownership

Primary driving reason in 50-70 % of various survey results



“They offer much less total cost of ownership (TCO)”
“Migrating to PostgreSQL can allow organizations to remove database licensing costs altogether from their budgets.”

Courtesy : Gartner, Stratoscale, EnterpriseDB

Mergers and Acquisitions

- Breaks the Operational challenges

  Expanding the existing team and retraining.

- Rolling the efficiency across the Organization
- Lesser Outage / Incident numbers

Cloud Migration Strategy

aws

HEROKU

Google Cloud

Azure

IBM Cloud

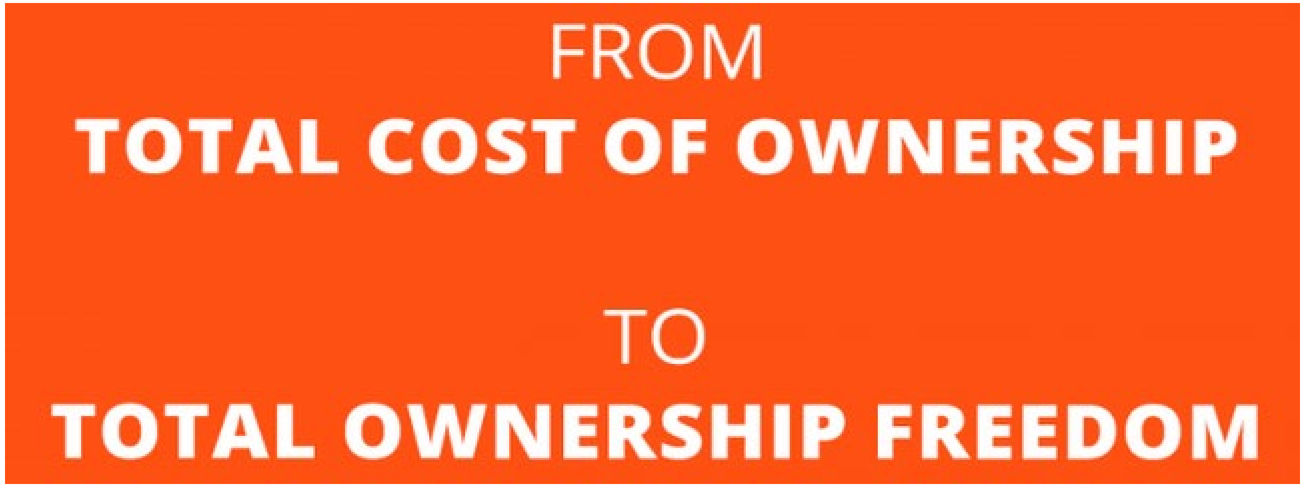
1. Large Number of Cloud Providers supporting PostgreSQL
2. Database as a Service from Multiple Vendors
3. Integration with Cloud storages - Aurora

We own it!

PostgreSQL cannot be brought out -
Open Source Products Vs Open Source Projects

No more fear of

- licence changes
- sell offs / spin offs



FROM

TOTAL COST OF OWNERSHIP

TO

TOTAL OWNERSHIP FREEDOM

Courtesy : splendiddata.com

Support with No lock-in

- Absolute Zero vendor lock-in and dependency
- 55+ Support companies in North America

  Support companies headquartered across globe
- 96+ companies in Europe

Intellectual Property Rights

- Intellectual Property protection

  No Copyrights assignments

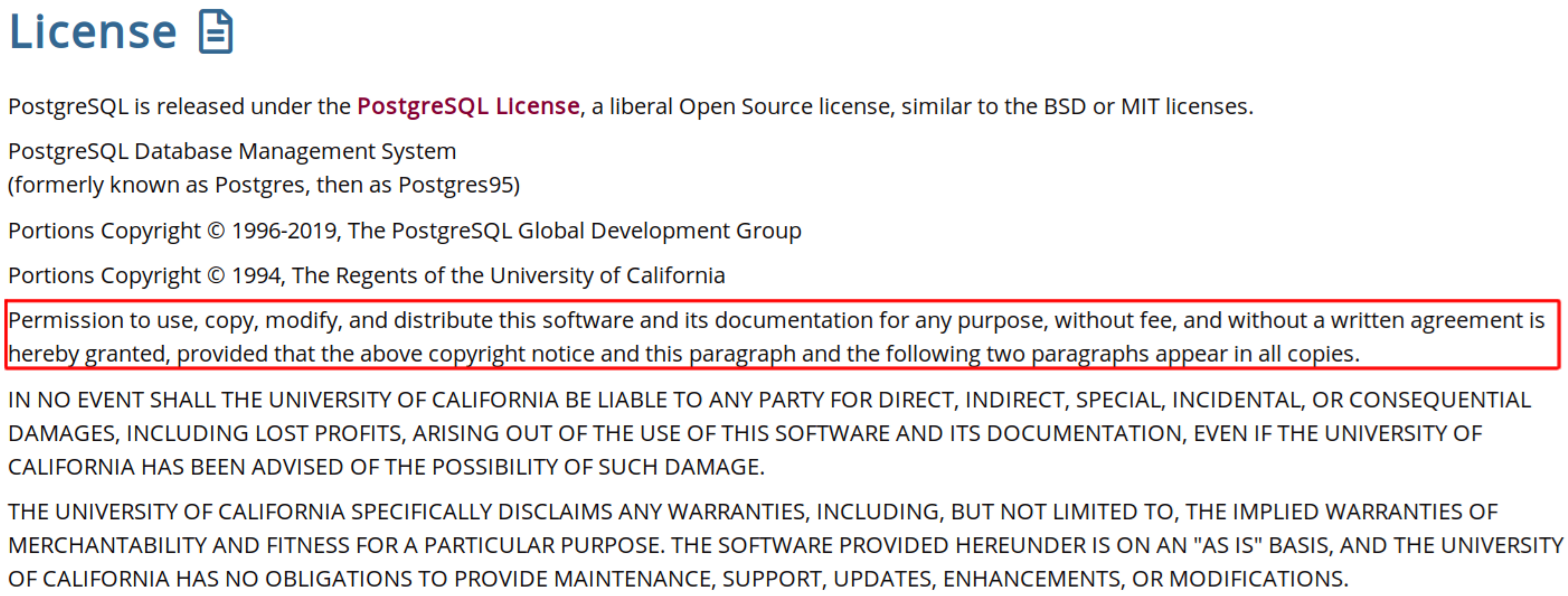
- Impact on M&A

amazon REDSHIFT

Timescale

GREENPLUM DATABASE

Licence, Integration and Packaging



License

PostgreSQL is released under the **PostgreSQL License**, a liberal Open Source license, similar to the BSD or MIT licenses.

PostgreSQL Database Management System
(formerly known as Postgres, then as Postgres95)

Portions Copyright © 1996-2019, The PostgreSQL Global Development Group

Portions Copyright © 1994, The Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Integration with application and shipping

https://www.postgresql.org/about/licence/

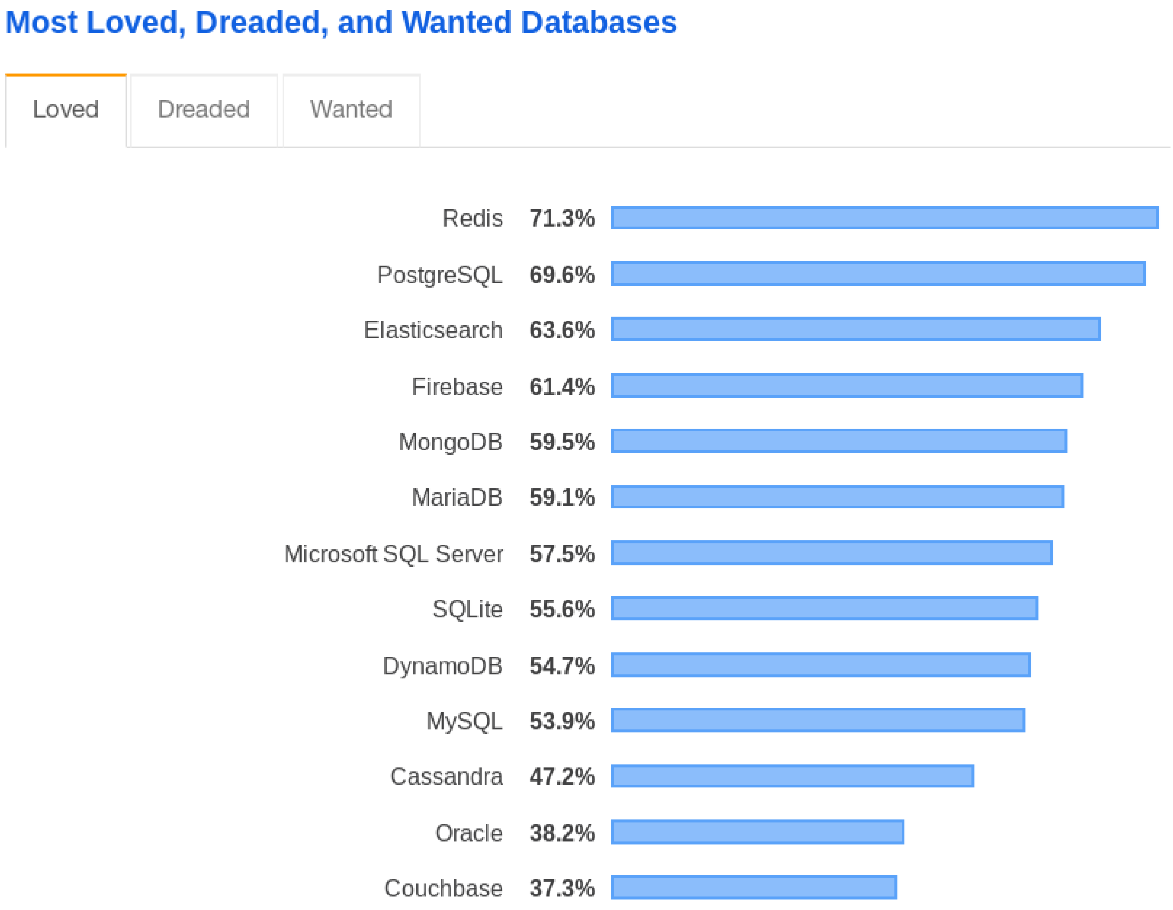
Comfort of PostgreSQL

- Most Powerful - Feature rich RDBMS
- Least TCO
- No Vendor lock-in
- Freedom to move - Cloud adoption
- No Intellectual Property, License concerns
- Certainty

Developer’s

Reasons to choose PostgreSQL

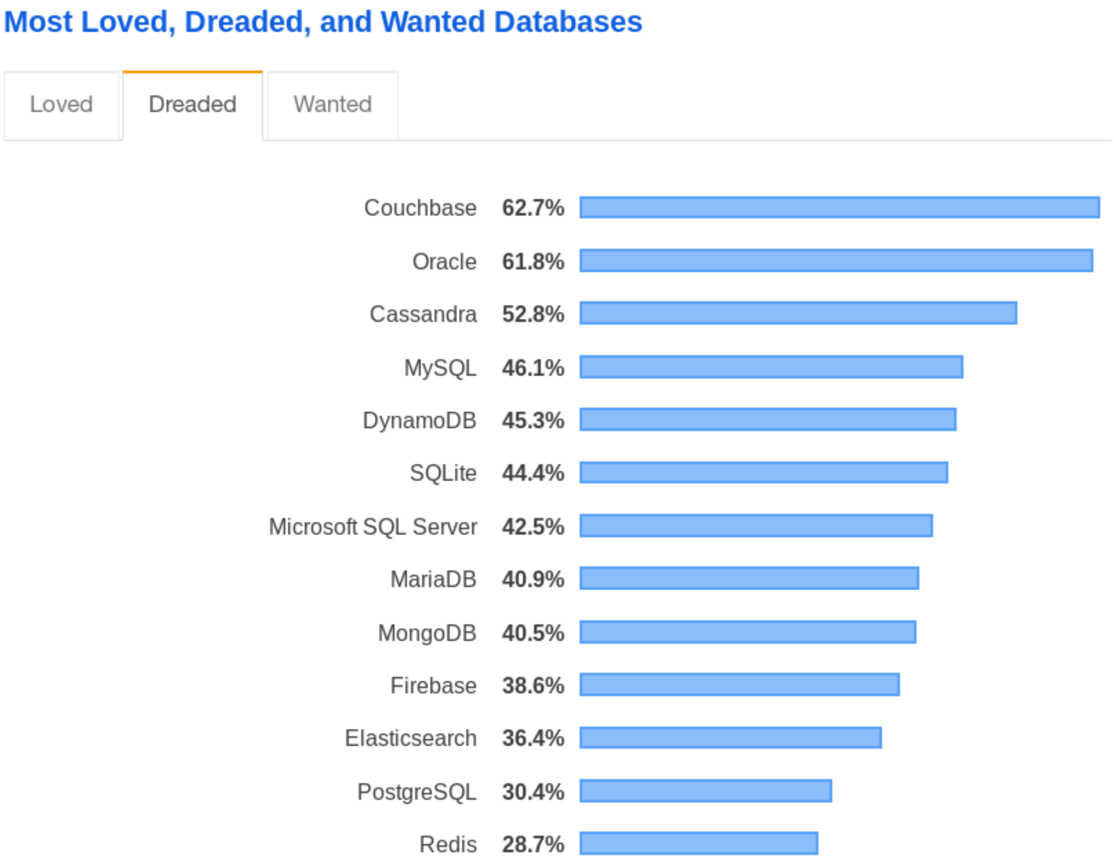
What's happening - 2019



Most Loved, Dreaded, and Wanted Databases

Loved | Dreaded | Wanted

Redis	71.3%
PostgreSQL	69.6%
Elasticsearch	63.6%
Firebase	61.4%
MongoDB	59.5%
MariaDB	59.1%
Microsoft SQL Server	57.5%
SQLite	55.6%
DynamoDB	54.7%
MySQL	53.9%
Cassandra	47.2%
Oracle	38.2%
Couchbase	37.3%



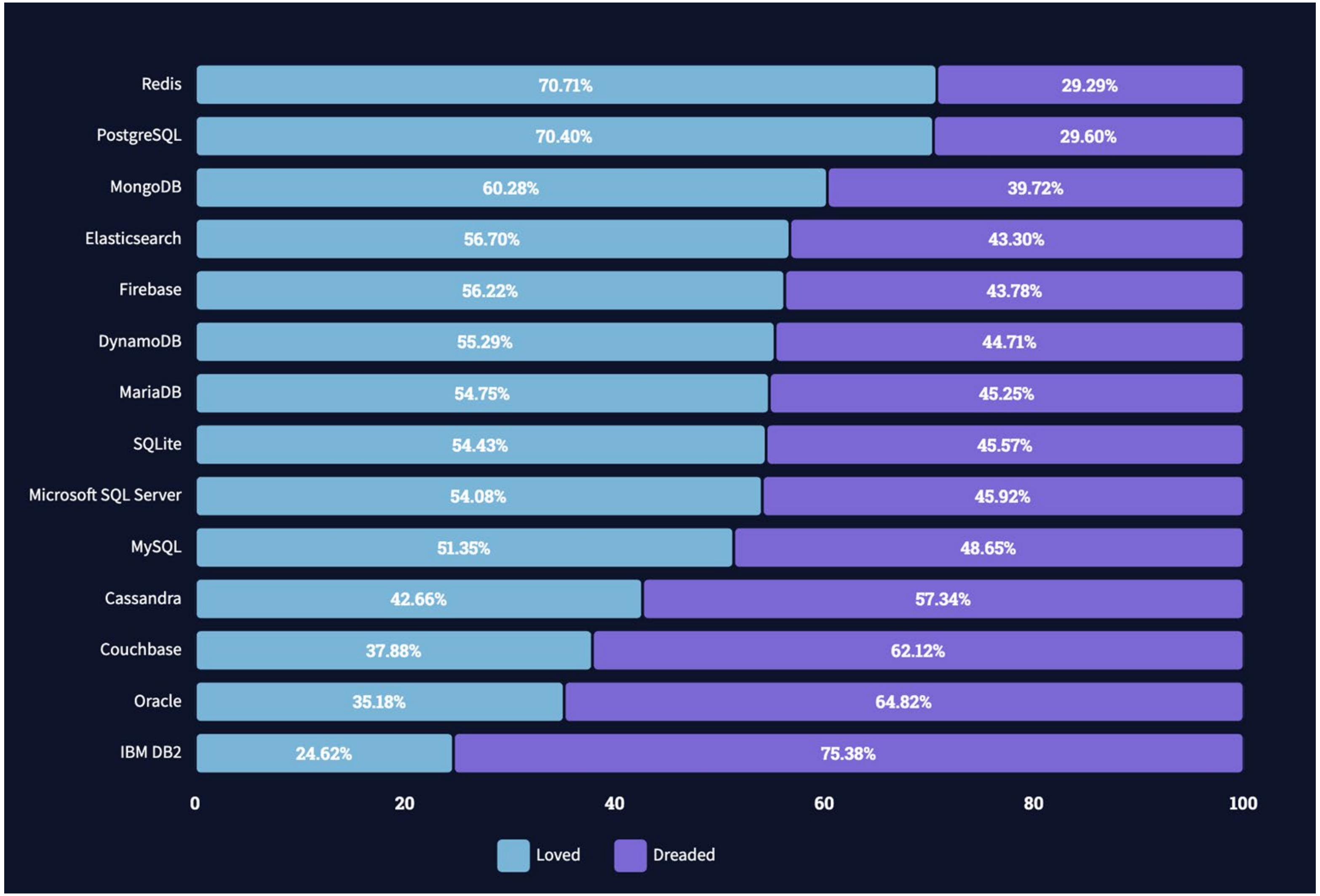
Most Loved, Dreaded, and Wanted Databases

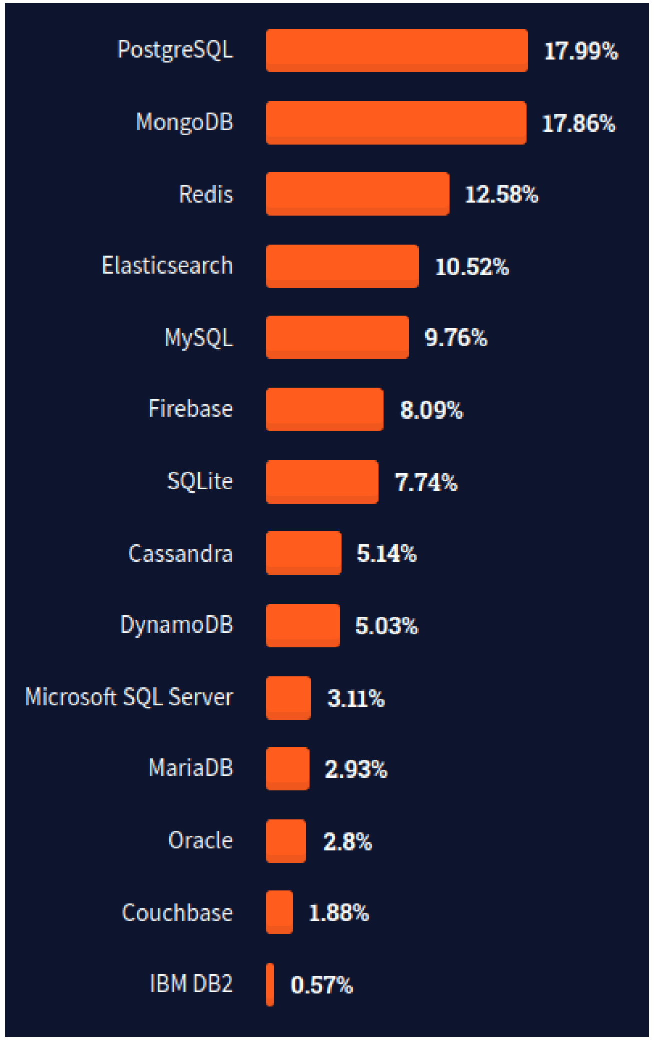
Loved | Dreaded | Wanted

Couchbase	62.7%
Oracle	61.8%
Cassandra	52.8%
MySQL	46.1%
DynamoDB	45.3%
SQLite	44.4%
Microsoft SQL Server	42.5%
MariaDB	40.9%
MongoDB	40.5%
Firebase	38.6%
Elasticsearch	36.4%
PostgreSQL	30.4%
Redis	28.7%

https://insights.stackoverflow.com/survey/2019#most-loved-dreaded-and-wanted

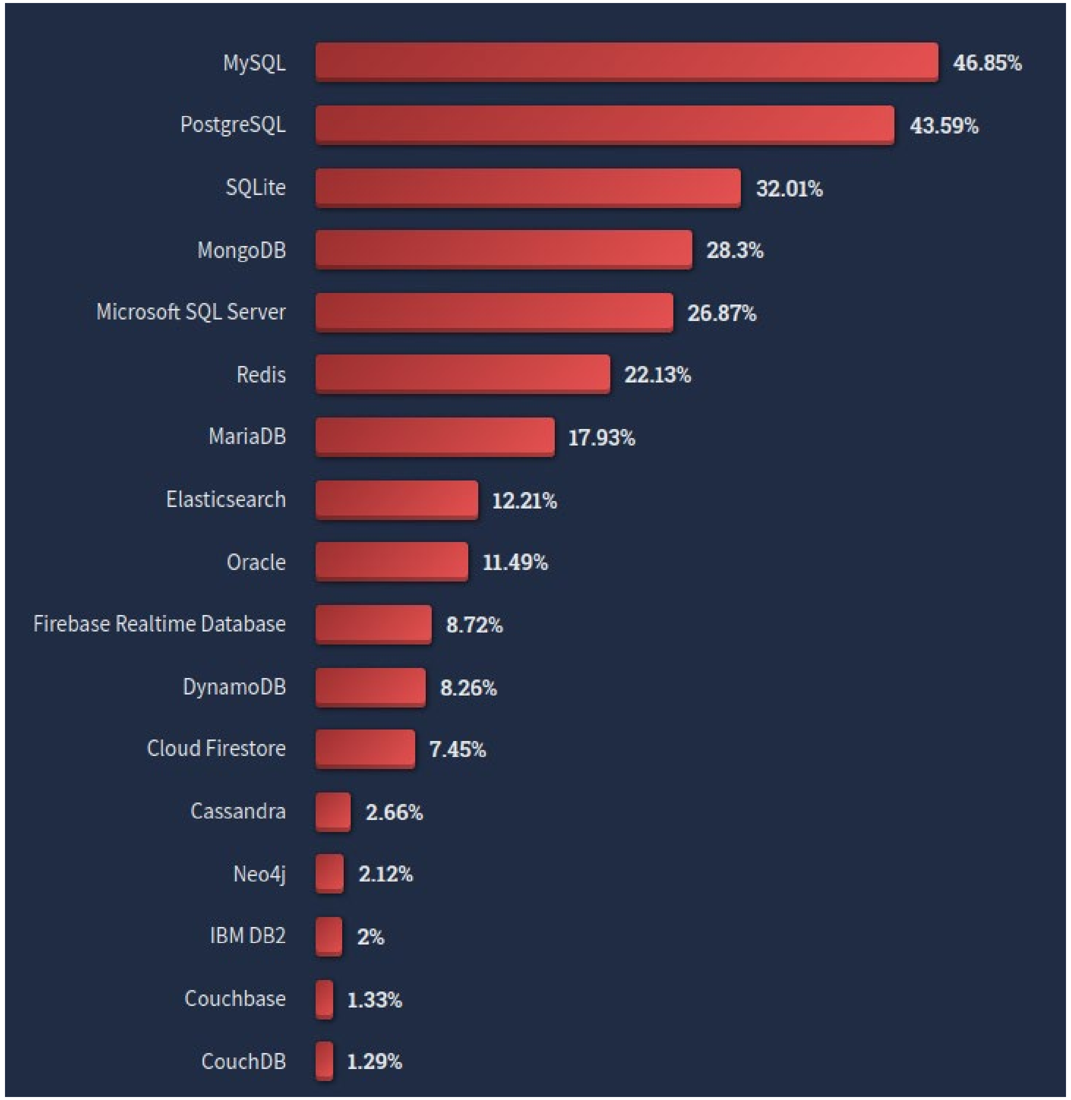
2021

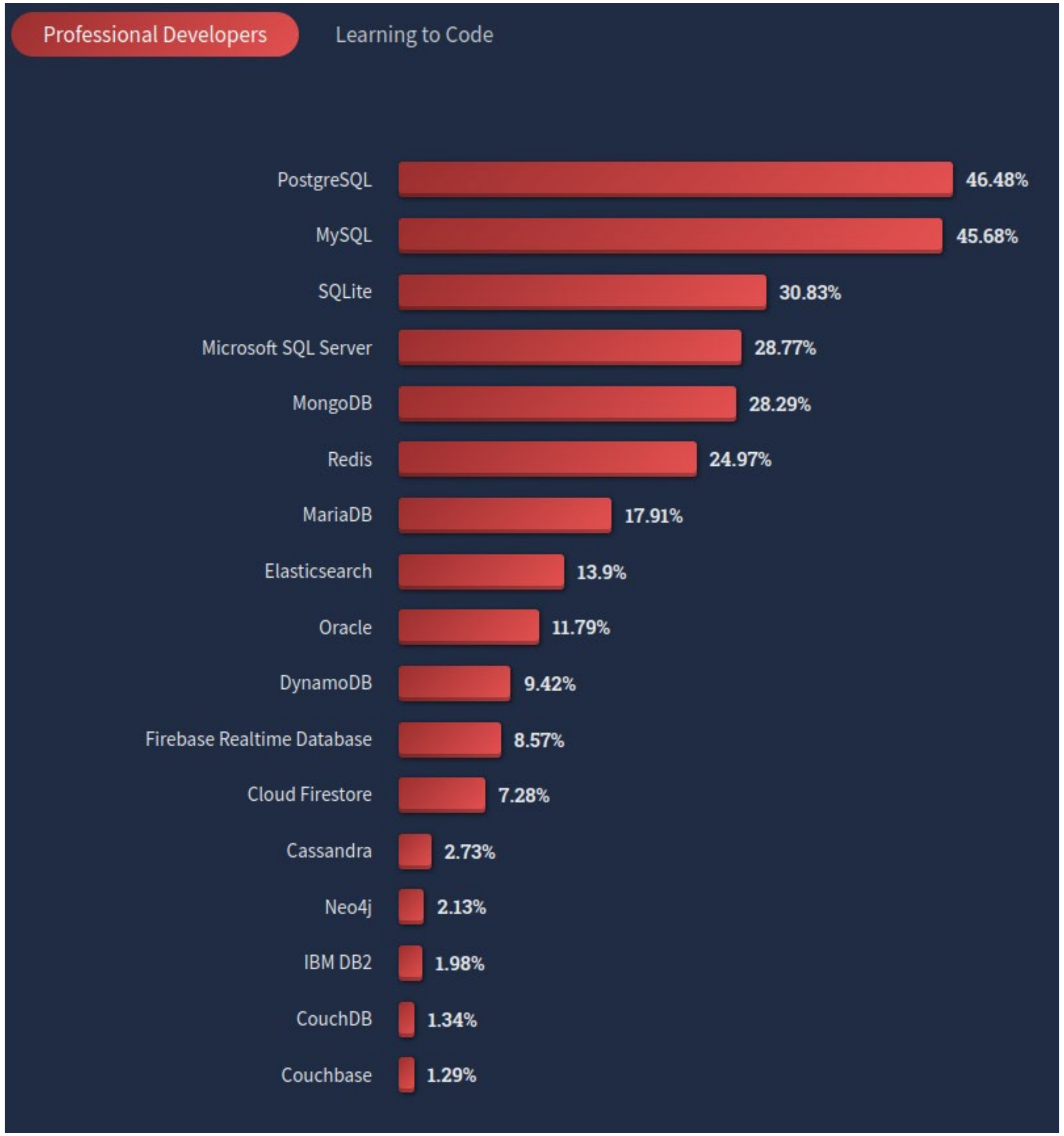




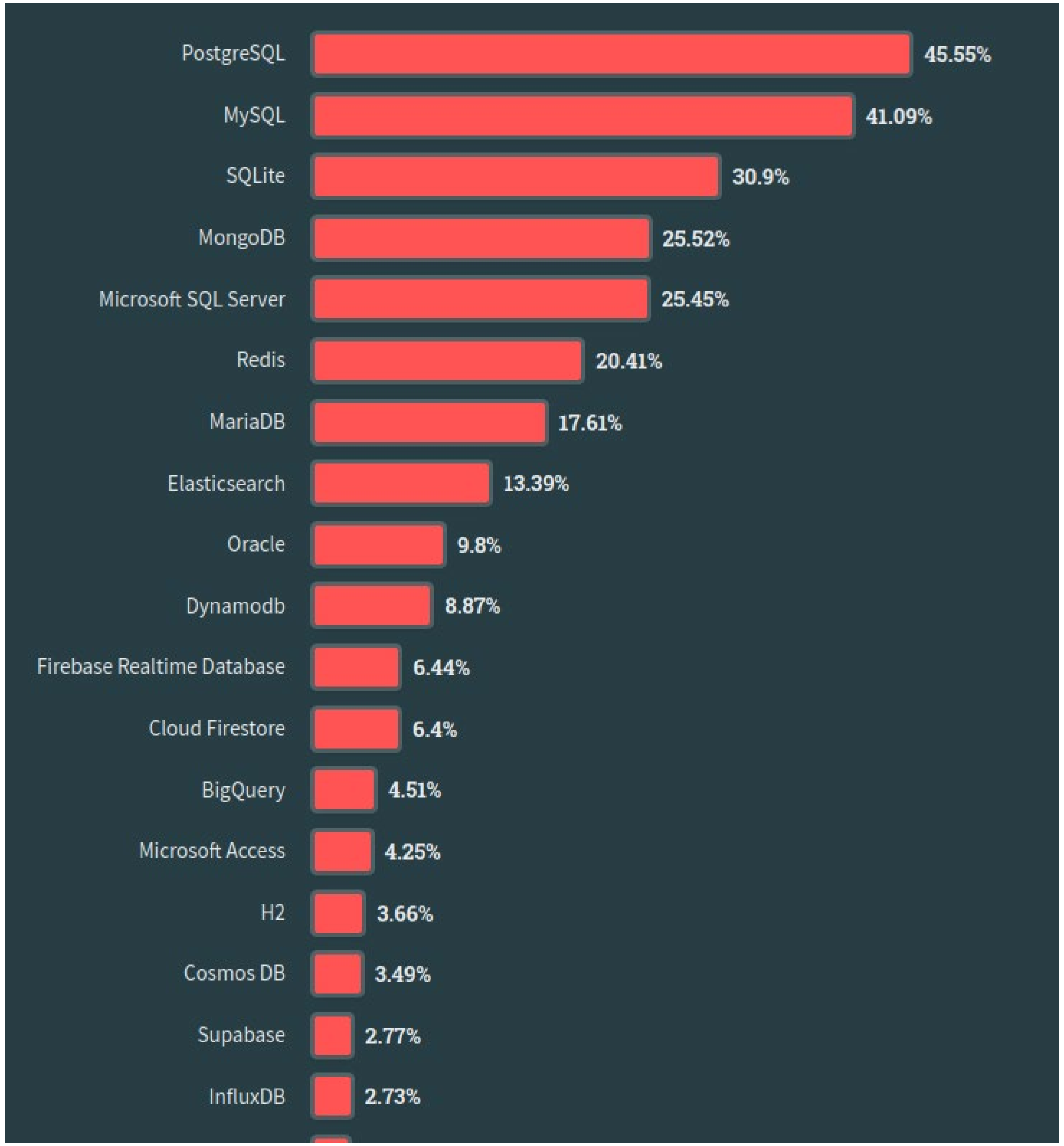
https://insights.stackoverflow.com/survey/2021#most-loved-dreaded-and-wanted-database-love-dread

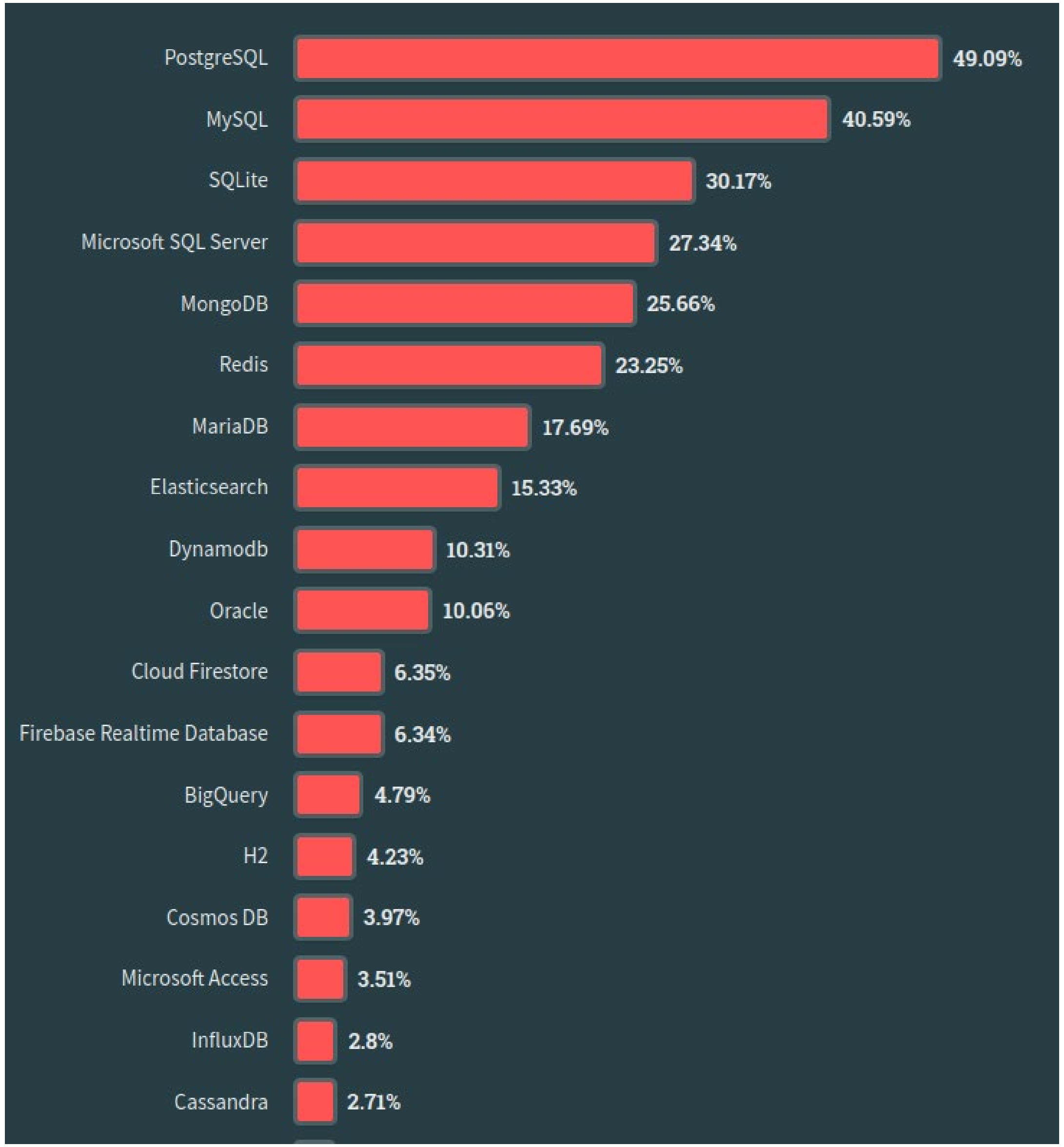
2022


MySQL 46.85%
PostgreSQL 43.59%
SQLite 32.01%
MongoDB 28.3%
Microsoft SQL Server 26.87%
Redis 22.13%
MariaDB 17.93%
Elasticsearch 12.21%
Oracle 11.49%
Firebase Realtime Database 8.72%
DynamoDB 8.26%
Cloud Firestore 7.45%
Cassandra 2.66%
Neo4j 2.12%
IBM DB2 2%
Couchbase 1.33%
CouchDB 1.29%


Professional Developers
Learning to Code
PostgreSQL 46.48%
MySQL 45.68%
SQLite 30.83%
Microsoft SQL Server 28.77%
MongoDB 28.29%
Redis 24.97%
MariaDB 17.91%
Elasticsearch 13.9%
Oracle 11.79%
DynamoDB 9.42%
Firebase Realtime Database 8.57%
Cloud Firestore 7.28%
Cassandra 2.73%
Neo4j 2.13%
IBM DB2 1.98%
CouchDB 1.34%
Couchbase 1.29%

2023


PostgreSQL 45.55%
MySQL 41.09%
SQLite 30.9%
MongoDB 25.52%
Microsoft SQL Server 25.45%
Redis 20.41%
MariaDB 17.61%
Elasticsearch 13.39%
Oracle 9.8%
Dynamodb 8.87%
Firebase Realtime Database 6.44%
Cloud Firestore 6.4%
BigQuery 4.51%
Microsoft Access 4.25%
H2 3.66%
Cosmos DB 3.49%
Supabase 2.77%
InfluxDB 2.73%


PostgreSQL 49.09%
MySQL 40.59%
SQLite 30.17%
Microsoft SQL Server 27.34%
MongoDB 25.66%
Redis 23.25%
MariaDB 17.69%
Elasticsearch 15.33%
Dynamodb 10.31%
Oracle 10.06%
Cloud Firestore 6.35%
Firebase Realtime Database 6.34%
BigQuery 4.79%
H2 4.23%
Cosmos DB 3.97%
Microsoft Access 3.51%
InfluxDB 2.8%
Cassandra 2.71%

SQL Standard

ISO/IEC 9075:2016 (SQL:2016)
SQL:2011, SQL:2008, SQL:2006, SQL:2003, SQL:1999, and SQL-92

- Most of the “Core” features are implemented with minor missings and deviations.

  Out of 177 mandatory features required for full Core conformance, PostgreSQL conforms to at **least 170**
- **Upcoming PostgreSQL 16 uses latest SQL:2023** (ISO/IEC 9075:2023) as the reference.

Foreign Data Wrappers:

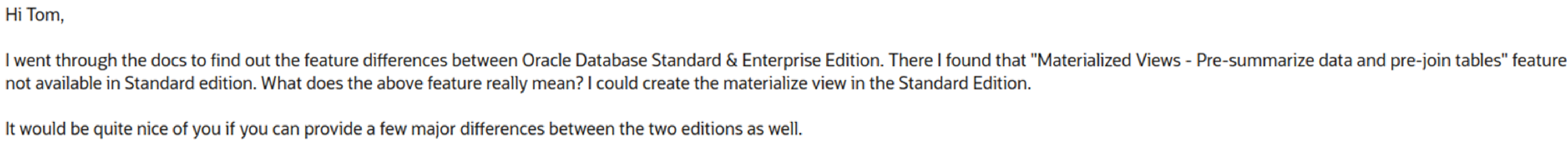
- ISO/IEC 9075-9 Management of External Data (SQL/MED)

Deviation from standards are discouraged

- No Hints
- No Non-standard statements

No Crippleware

There are NO 'Editions'



Hi Tom,

I went through the docs to find out the feature differences between Oracle Database Standard & Enterprise Edition. There I found that "Materialized Views - Pre-summarize data and pre-join tables" feature not available in Standard edition. What does the above feature really mean? I could create the materialize view in the Standard Edition.

It would be quite nice of you if you can provide a few major differences between the two editions as well.

Materialized views, Replication, Specialized Indexes etc.

There are **Distribution** and **Derived Softwares**

- https://www.postgresql.org/download/products/8-postgresql-derived-servers/
- https://wiki.postgresql.org/wiki/PostgreSQL_derived_databases

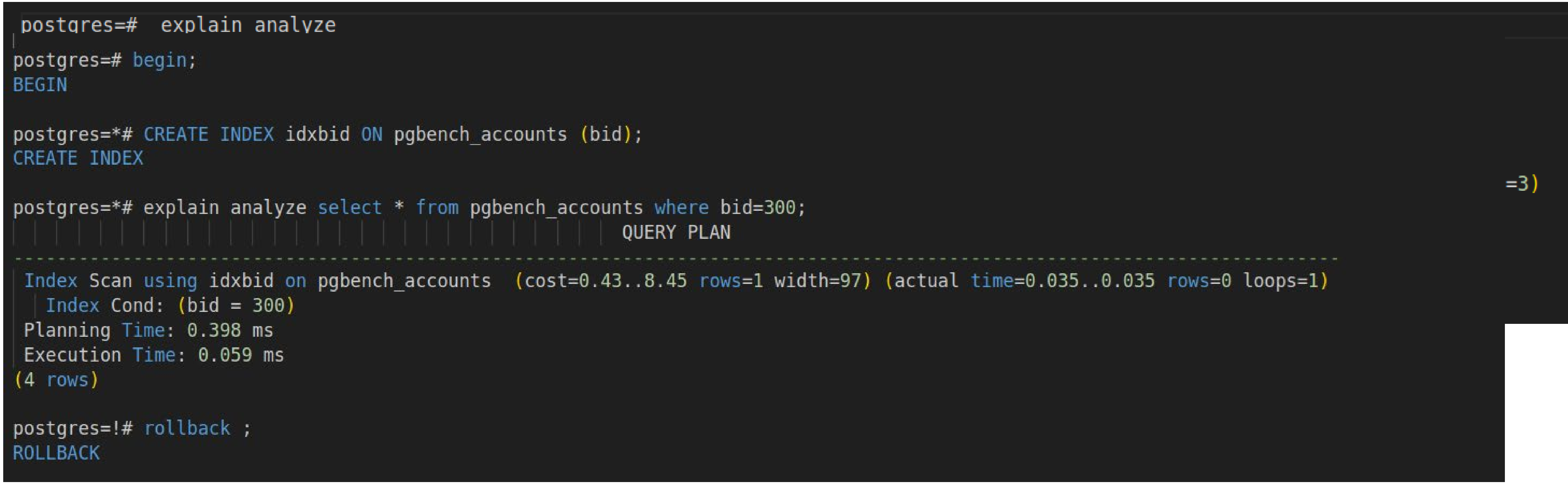
Best In class MVCC

- Snapshot Isolation
- Transactional DDLs

Isolation Level	Dirty Read Anomaly	Non-Repeatable Read Anomaly	Phantom Read Anomaly	Serialization Anomaly
Read Committed	Not Possible			
Repeatable Read	Not Possible	Not Possible	Not Possible in Postgres	
Serializable	Not Possible	Not Possible	Not Possible	Not Possible

PostgreSQL Capable of Serialisable Snapshot Isolation (SSI)

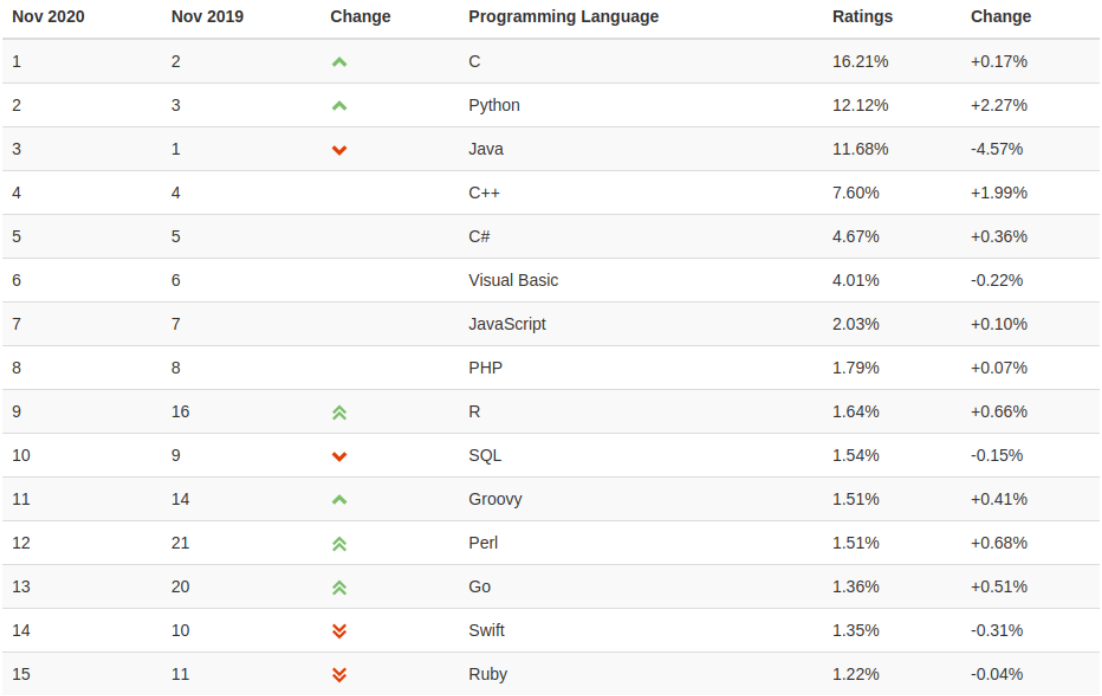
Best In class MVCC + ACID



```
 postgres=#  explain analyze

postgres=# begin;
BEGIN

postgres=*# CREATE INDEX idxbid ON pgbench_accounts (bid);
CREATE INDEX
                                                                                                                                     =3)
postgres=*# explain analyze select * from pgbench_accounts where bid=300;
                                                        QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------
 Index Scan using idxbid on pgbench_accounts  (cost=0.43..8.45 rows=1 width=97) (actual time=0.035..0.035 rows=0 loops=1)
   Index Cond: (bid = 300)
 Planning Time: 0.398 ms
 Execution Time: 0.059 ms
(4 rows)

postgres=!# rollback ;
ROLLBACK
```

Languages and Framework

- PL/pgSQL
- PL/Tcl
- PL/Perl
- PL/Python
- PL/Java
- PL/Lua
- PL/R
- PL/sh
- PL/v8
- C/C++
- …



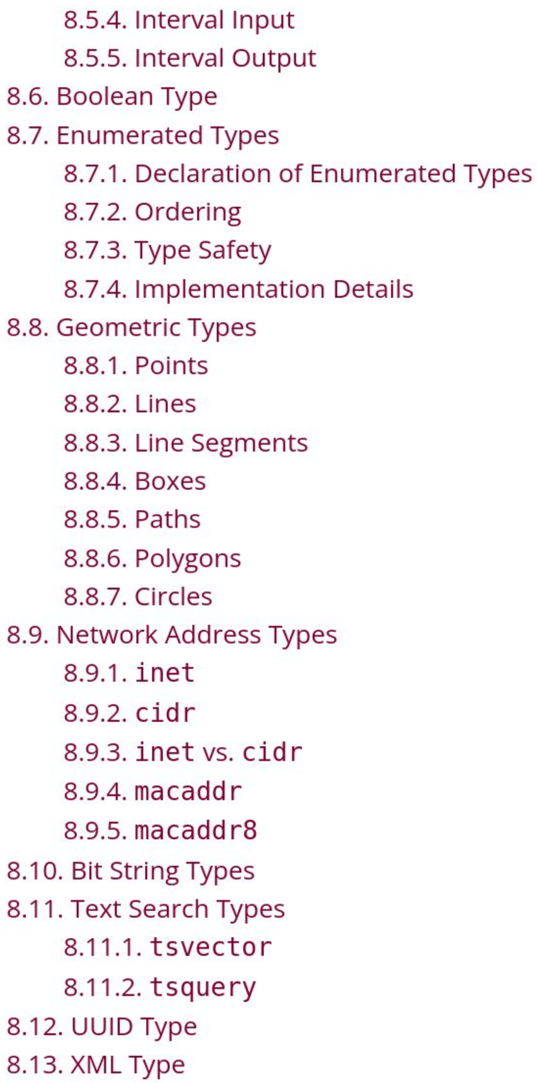
Nov 2020	Nov 2019	Change	Programming Language	Ratings	Change
1	2	⌃	C	16.21%	+0.17%
2	3	⌃	Python	12.12%	+2.27%
3	1	⌄	Java	11.68%	-4.57%
4	4		C++	7.60%	+1.99%
5	5		C#	4.67%	+0.36%
6	6		Visual Basic	4.01%	-0.22%
7	7		JavaScript	2.03%	+0.10%
8	8		PHP	1.79%	+0.07%
9	16	⇈	R	1.64%	+0.66%
10	9	⌄	SQL	1.54%	-0.15%
11	14	⌃	Groovy	1.51%	+0.41%
12	21	⇈	Perl	1.51%	+0.68%
13	20	⇈	Go	1.36%	+0.51%
14	10	⇊	Swift	1.35%	-0.31%
15	11	⇊	Ruby	1.22%	-0.04%

PostgREST

Serve a RESTful API from any Postgres database

https://www.tiobe.com

Data types



- 8.5.4. Interval Input
- 8.5.5. Interval Output
- 8.6. Boolean Type
- 8.7. Enumerated Types
  - 8.7.1. Declaration of Enumerated Types
  - 8.7.2. Ordering
  - 8.7.3. Type Safety
  - 8.7.4. Implementation Details
- 8.8. Geometric Types
  - 8.8.1. Points
  - 8.8.2. Lines
  - 8.8.3. Line Segments
  - 8.8.4. Boxes
  - 8.8.5. Paths
  - 8.8.6. Polygons
  - 8.8.7. Circles
- 8.9. Network Address Types
  - 8.9.1. `inet`
  - 8.9.2. `cidr`
  - 8.9.3. `inet` vs. `cidr`
  - 8.9.4. `macaddr`
  - 8.9.5. `macaddr8`
- 8.10. Bit String Types
- 8.11. Text Search Types
  - 8.11.1. `tsvector`
  - 8.11.2. `tsquery`
- 8.12. UUID Type
- 8.13. XML Type

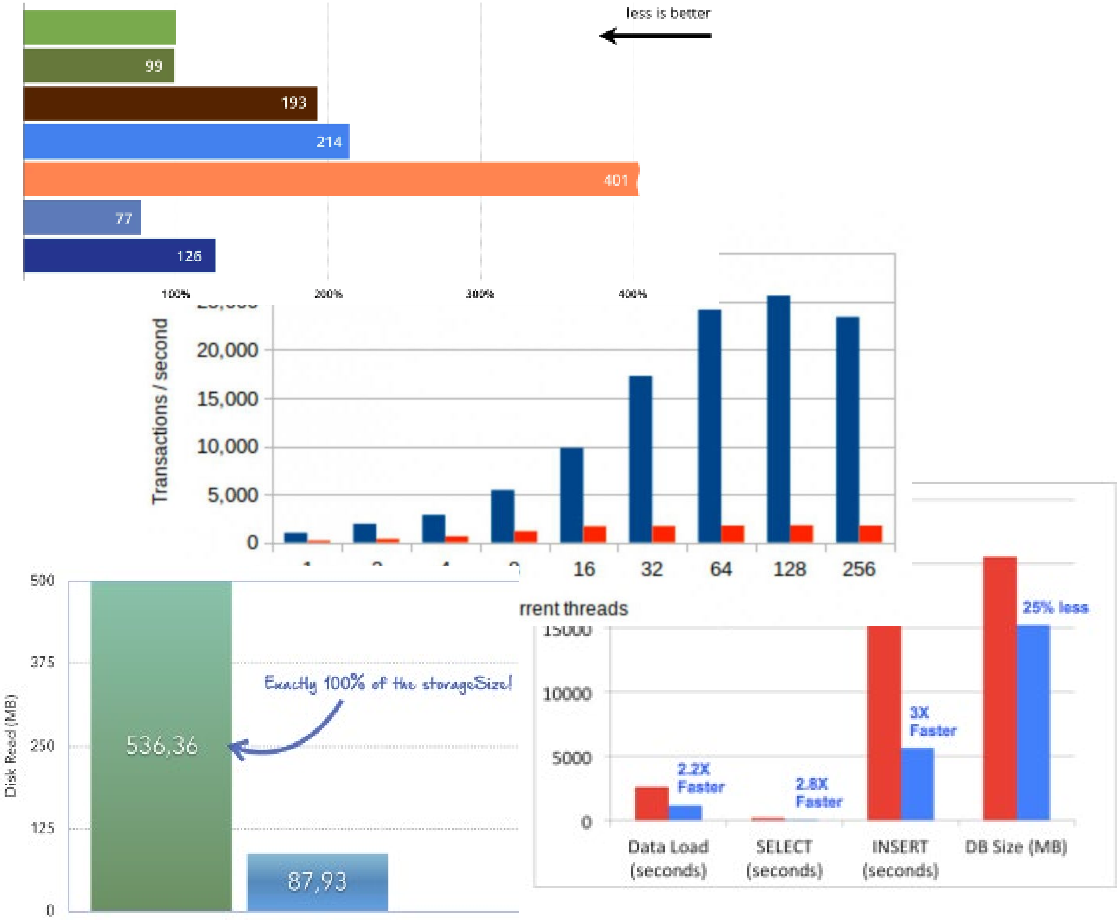
bool bytea char name int8 int2 int2vector int4 regproc text oid tid xid cid oidvector json xml xid8 point lseg path box polygon line float4 float8 unknown circle money macaddr inet cidr macaddr8 aclitem bpchar varchar date time timestamp timestamptz interval timetz bit varbit numeric refcursor regprocedure regoper regoperator regclass regcollation regtype regrole regnamespace uuid pg_lsn tsvector gtsvector tsquery regconfig regdictionary jsonb jsonpath txid_snapshot int4range numrange tsrange tstzrange daterange int8range int4multirange nummultirange tsmultirange tstzmultirange datemultirange int8multirange record cstring any anyarray void trigger event_trigger language_handler internal anyelement anynonarray anyenum fdw_handler anyrange anycompatible anycompatiblearray anycompatiblenonarray anycompatiblerange anymultirange anycompatiblemultirange

JSON document Store

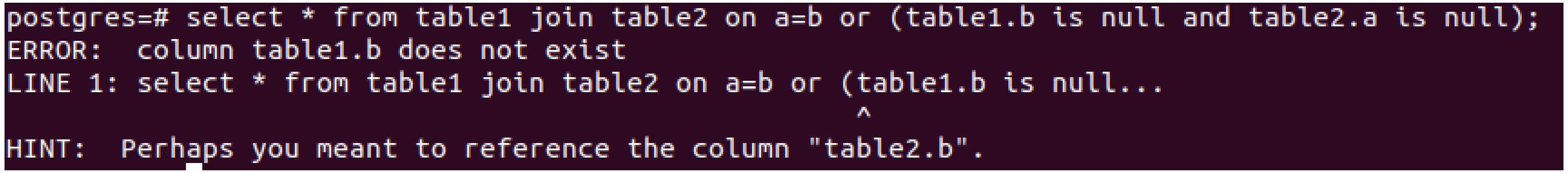
- JSON and JSONB data types
- Combining the power of SQL with Document store
- JSONB decomposes the JSON to binary
  - Efficient, Fast, Indexable.

SQL 2016 standards - SQL/JSON features

- json[b]_to_tsvector() improvments
- jsonb_set() improvements
- jsonpath .datetime()
  - Timezone aware output

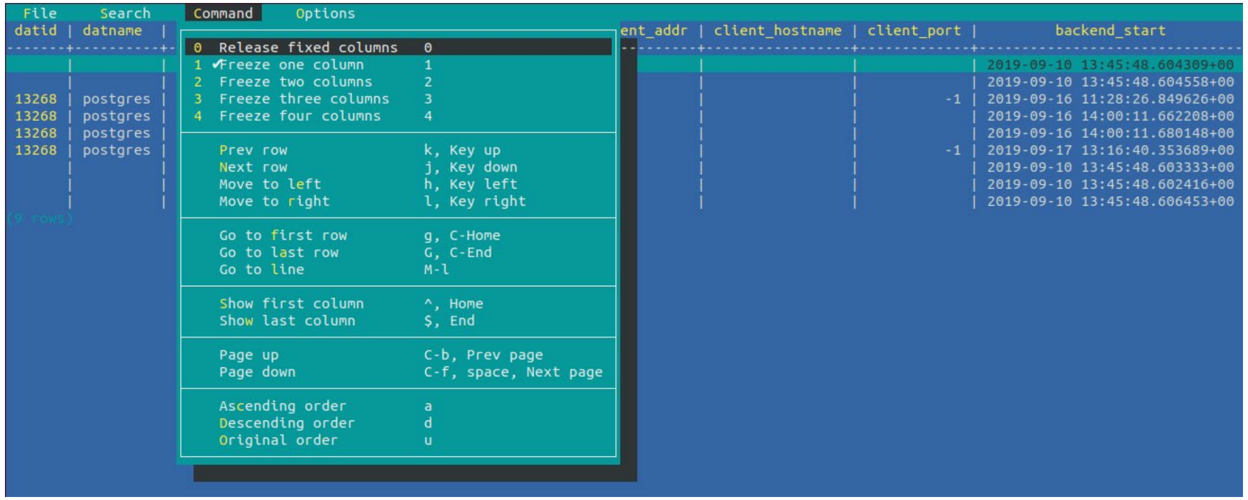


PSQL and Clients



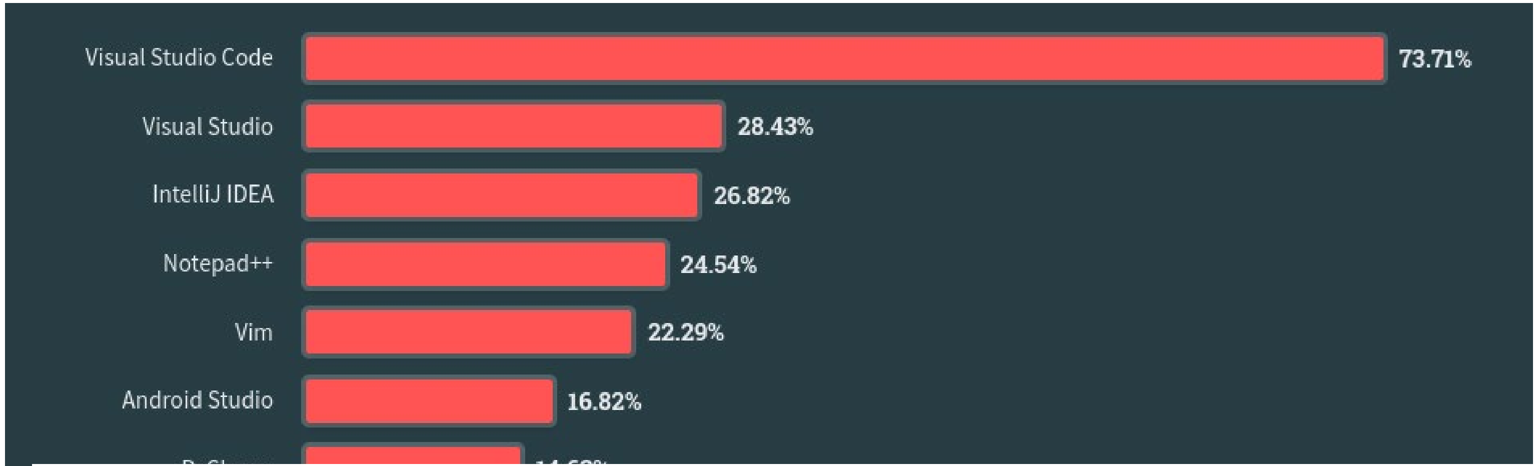
```
postgres=# select * from table1 join table2 on a=b or (table1.b is null and table2.a is null);
ERROR:  column table1.b does not exist
LINE 1: select * from table1 join table2 on a=b or (table1.b is null...
                                                    ^
HINT:  Perhaps you meant to reference the column "table2.b".
```

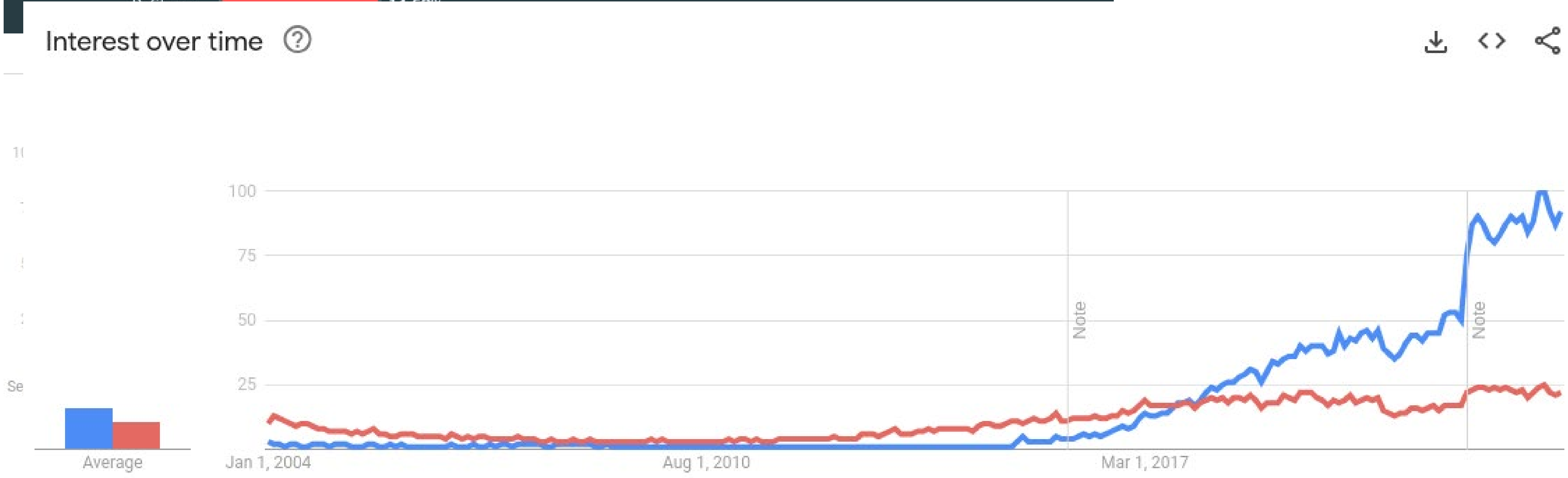
1. Autocompletion
2. User-friendly ERROR/HINTS messages
3. Shortcuts
4. Environment Variables / files (.pgpass .psqlrc)
5. Add-on Pagers : (eg: **pspg**)



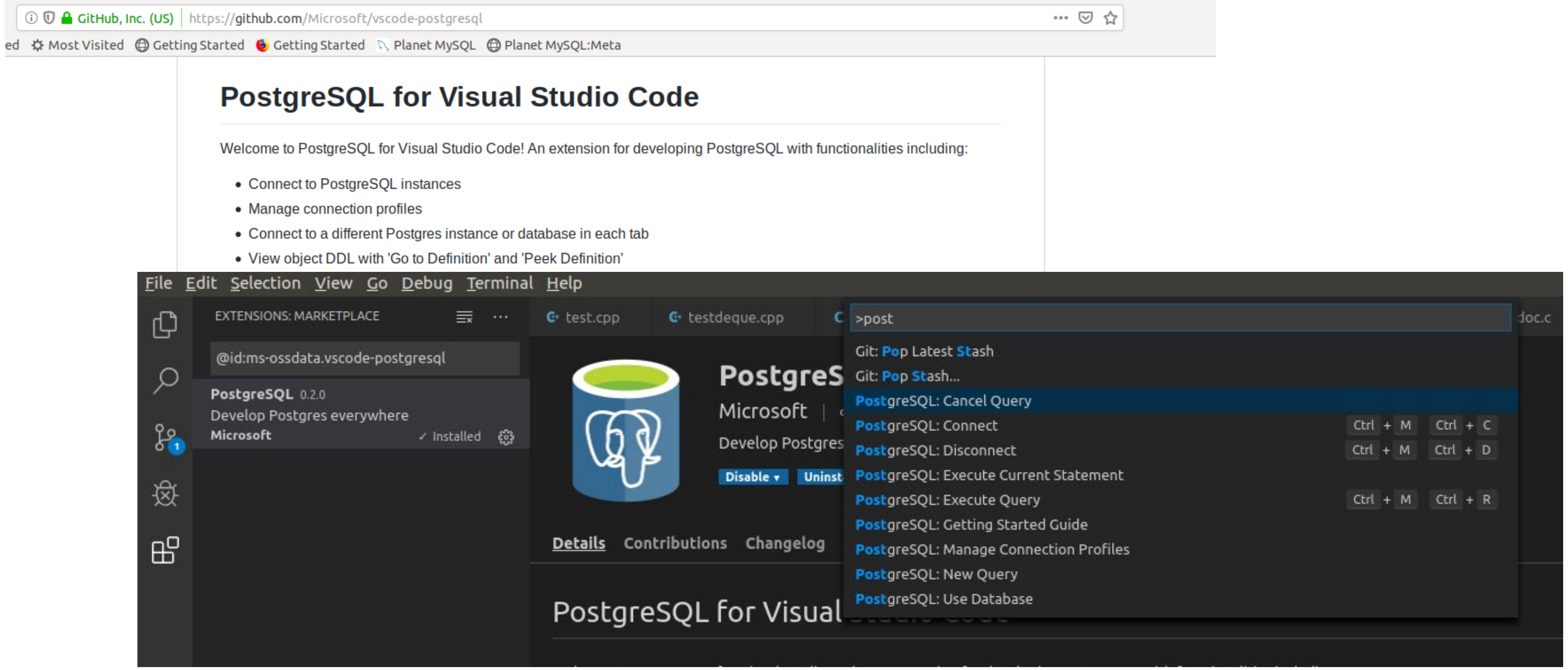
```
datid | datname  |      ent_addr | client_hostname | client_port |        backend_start
------+----------+-    ----------+-----------------+-------------+-----------------------------
      |          |               |                 |             | 2019-09-10 13:45:48.604309+00
      |          |               |                 |             | 2019-09-10 13:45:48.604558+00
13268 | postgres |               |                 |          -1 | 2019-09-16 11:28:26.849626+00
13268 | postgres |               |                 |             | 2019-09-16 14:00:11.662208+00
13268 | postgres |               |                 |             | 2019-09-16 14:00:11.680148+00
13268 | postgres |               |                 |          -1 | 2019-09-17 13:16:40.353689+00
      |          |               |                 |             | 2019-09-10 13:45:48.603333+00
      |          |               |                 |             | 2019-09-10 13:45:48.602416+00
      |          |               |                 |             | 2019-09-10 13:45:48.606453+00
(9 rows)
```

Developer tools





Visual Studio Code Example of vibrant community


GitHub, Inc. (US) | https://github.com/Microsoft/vscode-postgresql
ed Most Visited Getting Started Getting Started Planet MySQL Planet MySQL:Meta

PostgreSQL for Visual Studio Code

Welcome to PostgreSQL for Visual Studio Code! An extension for developing PostgreSQL with functionalities including:

- Connect to PostgreSQL instances
- Manage connection profiles
- Connect to a different Postgres instance or database in each tab
- View object DDL with 'Go to Definition' and 'Peek Definition'

File Edit Selection View Go Debug Terminal Help
EXTENSIONS: MARKETPLACE
test.cpp
testdeque.cpp
doc.c
Disable ▾
Uninst
Details
Contributions
Changelog

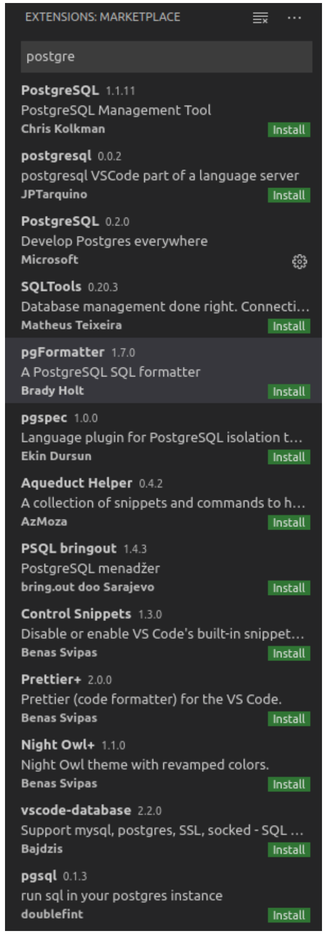
@id:ms-ossdata.vscode-postgresql

PostgreSQL 0.2.0
Develop Postgres everywhere
Microsoft ✓ Installed

PostgreS
Microsoft |
Develop Postgres

PostgreSQL for Visual

```
>post
Git: Pop Latest Stash
Git: Pop Stash...
PostgreSQL: Cancel Query
PostgreSQL: Connect                          Ctrl + M  Ctrl + C
PostgreSQL: Disconnect                       Ctrl + M  Ctrl + D
PostgreSQL: Execute Current Statement
PostgreSQL: Execute Query                    Ctrl + M  Ctrl + R
PostgreSQL: Getting Started Guide
PostgreSQL: Manage Connection Profiles
PostgreSQL: New Query
PostgreSQL: Use Database
```



postgre

PostgreSQL 1.1.11
PostgreSQL Management Tool
Chris Kolkman Install

postgresql 0.0.2
postgresql VSCode part of a language server
JPTarquino Install

PostgreSQL 0.2.0
Develop Postgres everywhere
Microsoft

SQLTools 0.20.3
Database management done right. Connecti…
Matheus Teixeira Install

pgFormatter 1.7.0
A PostgreSQL SQL formatter
Brady Holt Install

pgspec 1.0.0
Language plugin for PostgreSQL isolation t…
Ekin Dursun Install

Aqueduct Helper 0.4.2
A collection of snippets and commands to h…
AzMoza Install

PSQL bringout 1.4.3
PostgreSQL menadžer
bring.out doo Sarajevo Install

Control Snippets 1.3.0
Disable or enable VS Code's built-in snippet…
Benas Svipas Install

Prettier+ 2.0.0
Prettier (code formatter) for the VS Code.
Benas Svipas Install

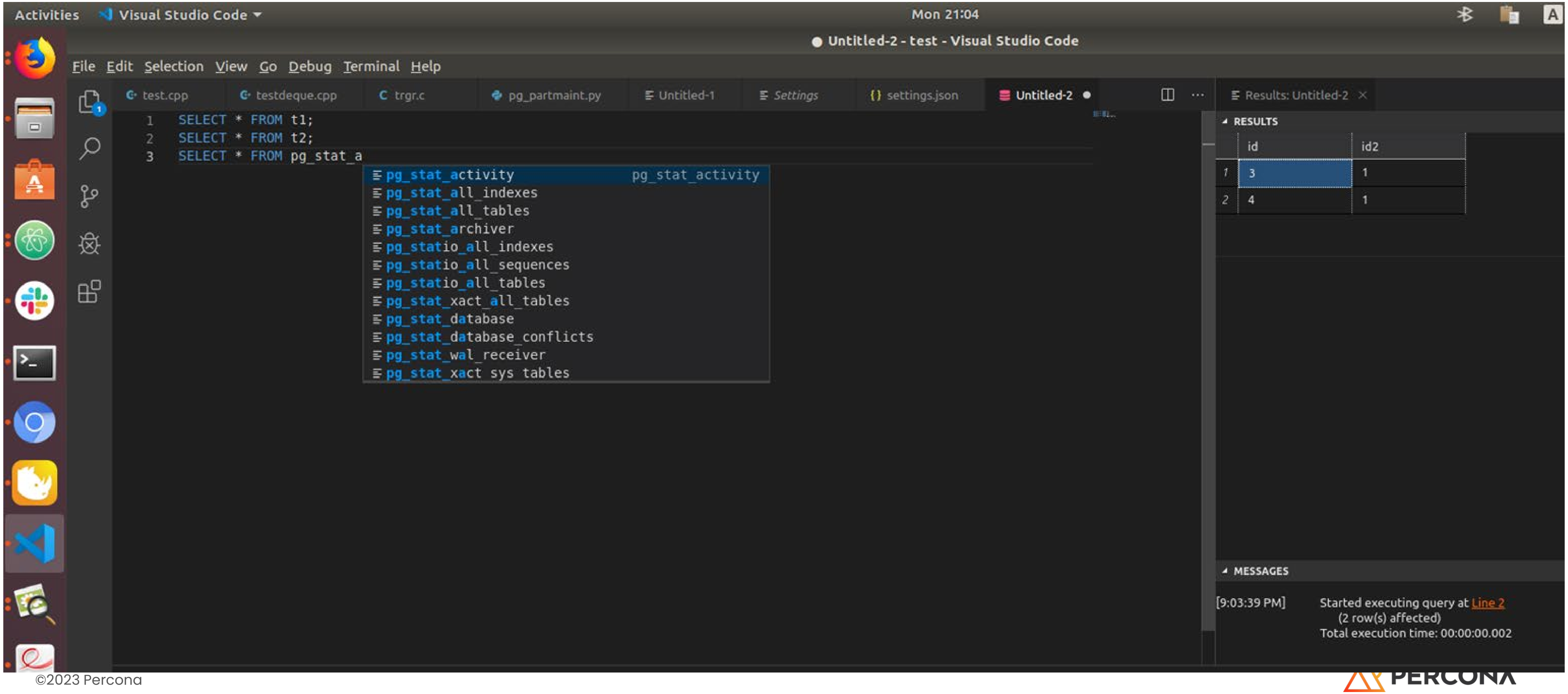
Night Owl+ 1.1.0
Night Owl theme with revamped colors.
Benas Svipas Install

vscode-database 2.2.0
Support mysql, postgres, SSL, socked - SQL …
Bajdzis Install

pgsql 0.1.3
run sql in your postgres instance
doublefint Install

- Github Repository
- MIT Licence
- Add-on

Visual Studio Code Example of vibrant community



```
SELECT * FROM t1;
SELECT * FROM t2;
SELECT * FROM pg_stat_a
```

```
pg_stat_activity                pg_stat_activity
pg_stat_all_indexes
pg_stat_all_tables
pg_stat_archiver
pg_statio_all_indexes
pg_statio_all_sequences
pg_statio_all_tables
pg_stat_xact_all_tables
pg_stat_database
pg_stat_database_conflicts
pg_stat_wal_receiver
pg_stat_xact sys tables
```

RESULTS

	id	id2
1	3	1
2	4	1

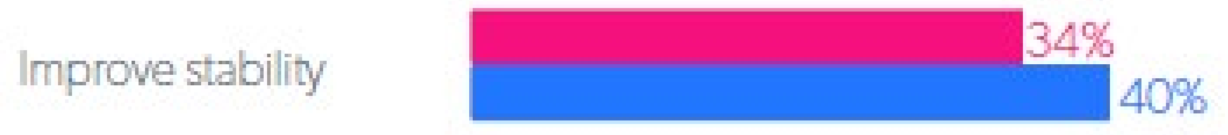
MESSAGES

```
[9:03:39 PM]    Started executing query at Line 2
                  (2 row(s) affected)
                Total execution time: 00:00:00.002
```

Operations’

Reasons to choose PostgreSQL

Legendary Stability

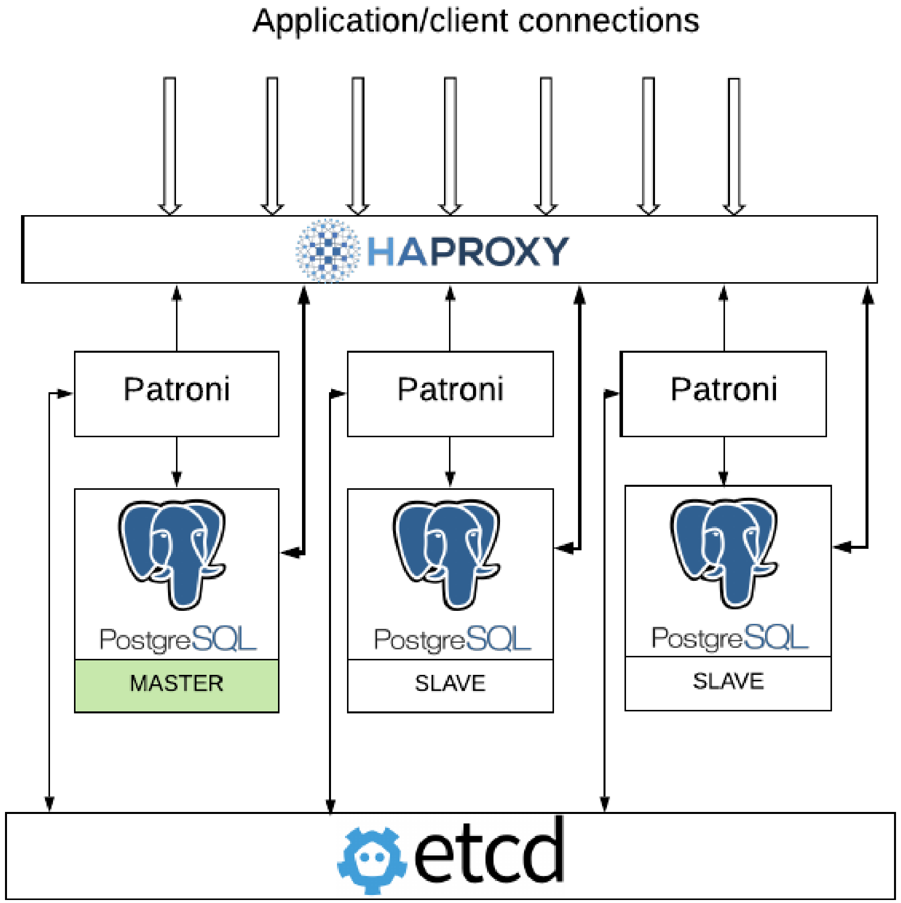


“The contributors have stayed true to its core of prizing stability and data integrity over flashy enhancements”

https://www.stratoscale.com

Multitude of HA options

- Patroni
- Londiste
- PAF
- repmgr
- pg_auto_failover
- ...

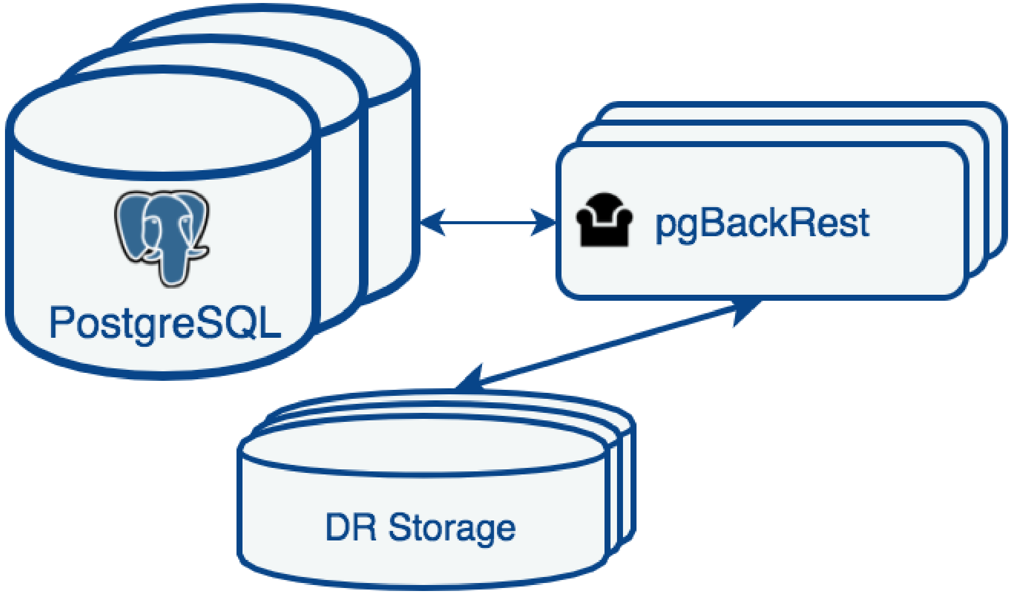


Enterprise class Backup tools

pg_basebackup

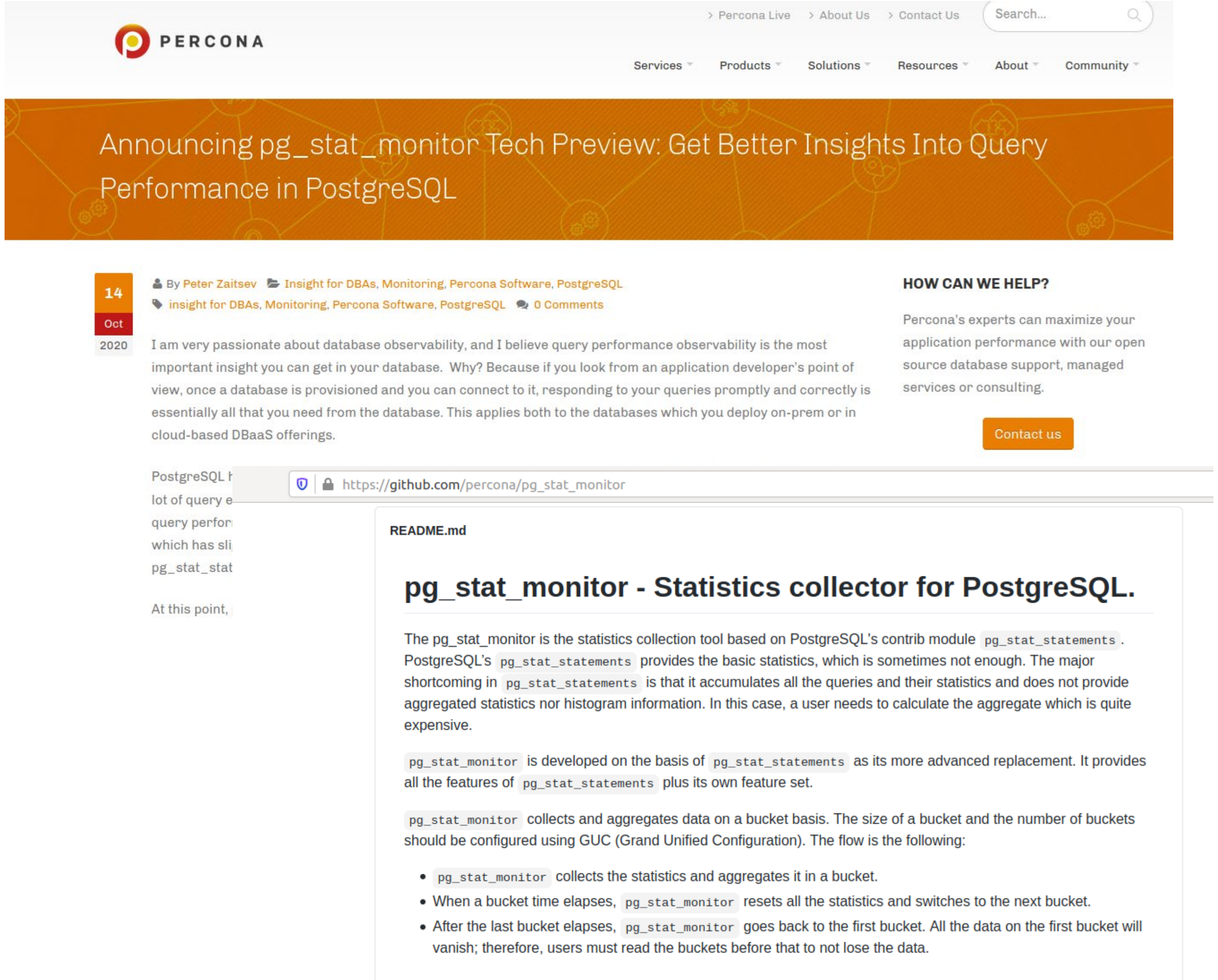
pgBackRest
Reliable PostgreSQL Backup & Restore

Barman
Backup and recovery
manager for PostgreSQL



Plethora of tools

- Monitoring tools
- Troubleshooting tools
- Extensions for DBAs
- Community Scripts



PERCONA
› Percona Live › About Us › Contact Us
Search...
Services Products Solutions Resources About Community

Announcing pg_stat_monitor Tech Preview: Get Better Insights Into Query Performance in PostgreSQL

14 Oct 2020

By Peter Zaitsev Insight for DBAs, Monitoring, Percona Software, PostgreSQL
insight for DBAs, Monitoring, Percona Software, PostgreSQL 0 Comments

I am very passionate about database observability, and I believe query performance observability is the most important insight you can get in your database. Why? Because if you look from an application developer's point of view, once a database is provisioned and you can connect to it, responding to your queries promptly and correctly is essentially all that you need from the database. This applies both to the databases which you deploy on-prem or in cloud-based DBaaS offerings.

PostgreSQL h
lot of query e
query perfor
which has sli
pg_stat_stat

At this point,

HOW CAN WE HELP?

Percona's experts can maximize your application performance with our open source database support, managed services or consulting.

Contact us

https://github.com/percona/pg_stat_monitor

README.md

pg_stat_monitor - Statistics collector for PostgreSQL.

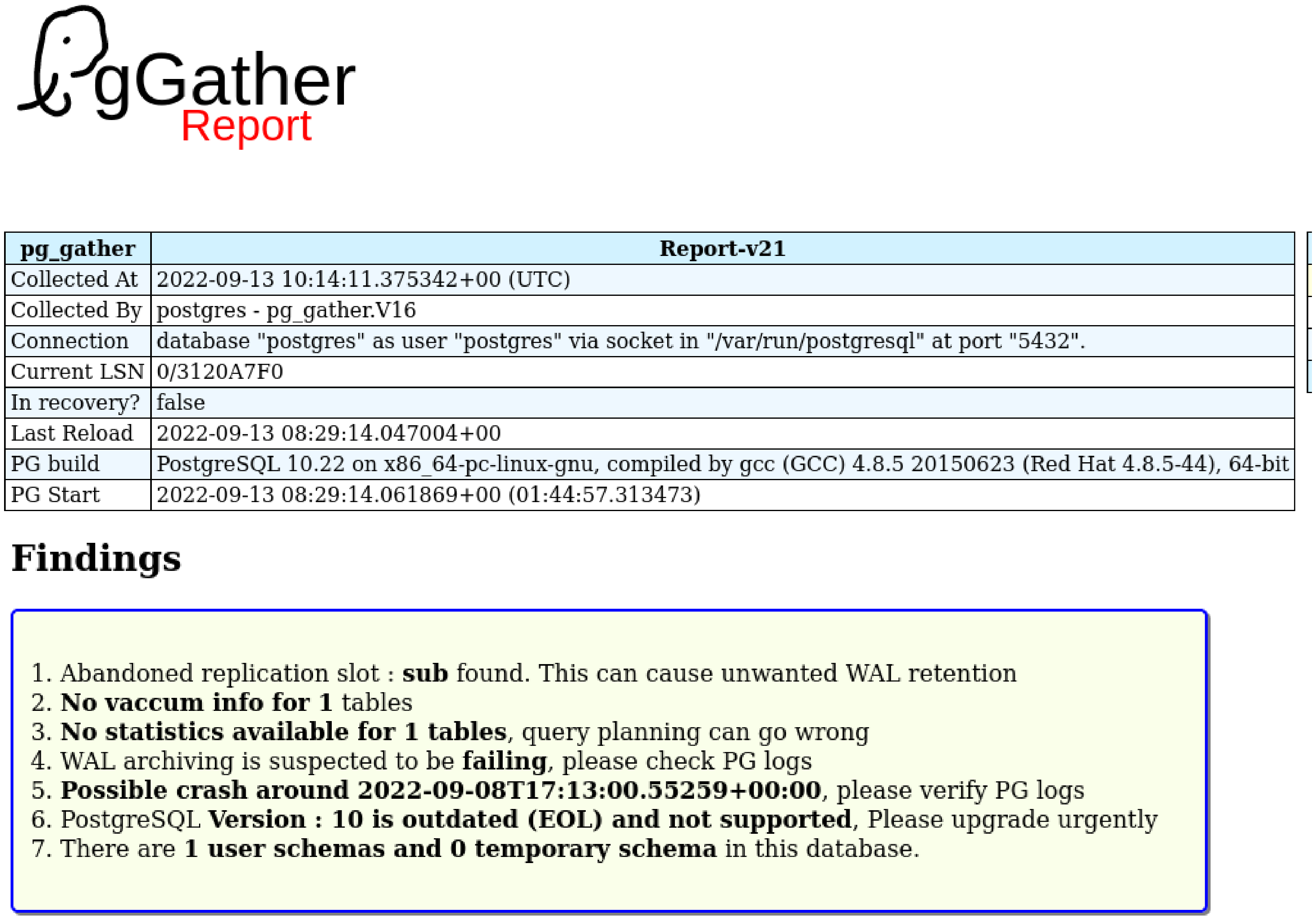
The pg_stat_monitor is the statistics collection tool based on PostgreSQL's contrib module `pg_stat_statements`. PostgreSQL's `pg_stat_statements` provides the basic statistics, which is sometimes not enough. The major shortcoming in `pg_stat_statements` is that it accumulates all the queries and their statistics and does not provide aggregated statistics nor histogram information. In this case, a user needs to calculate the aggregate which is quite expensive.

`pg_stat_monitor` is developed on the basis of `pg_stat_statements` as its more advanced replacement. It provides all the features of `pg_stat_statements` plus its own feature set.

`pg_stat_monitor` collects and aggregates data on a bucket basis. The size of a bucket and the number of buckets should be configured using GUC (Grand Unified Configuration). The flow is the following:

- `pg_stat_monitor` collects the statistics and aggregates it in a bucket.
- When a bucket time elapses, `pg_stat_monitor` resets all the statistics and switches to the next bucket.
- After the last bucket elapses, `pg_stat_monitor` goes back to the first bucket. All the data on the first bucket will vanish; therefore, users must read the buckets before that to not lose the data.

pg_gather - The scanner



pg_gather	Report-v21
Collected At	2022-09-13 10:14:11.375342+00 (UTC)
Collected By	postgres - pg_gather.V16
Connection	database "postgres" as user "postgres" via socket in "/var/run/postgresql" at port "5432".
Current LSN	0/3120A7F0
In recovery?	false
Last Reload	2022-09-13 08:29:14.047004+00
PG build	PostgreSQL 10.22 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-44), 64-bit
PG Start	2022-09-13 08:29:14.061869+00 (01:44:57.313473)

Findings

1. Abandoned replication slot : **sub** found. This can cause unwanted WAL retention
2. **No vaccum info for 1** tables
3. **No statistics available for 1 tables**, query planning can go wrong
4. WAL archiving is suspected to be **failing**, please check PG logs
5. **Possible crash around 2022-09-08T17:13:00.55259+00:00**, please verify PG logs
6. PostgreSQL **Version : 10 is outdated (EOL) and not supported**, Please upgrade urgently
7. There are **1 user schemas and 0 temporary schema** in this database.

Architects’

Reasons to choose PostgreSQL

Functions and Procedures

Supports Procedures from PostgreSQL 11 onwards

- Procedure can have COMMITS and ROLLBACK
- Multiple transaction blocks

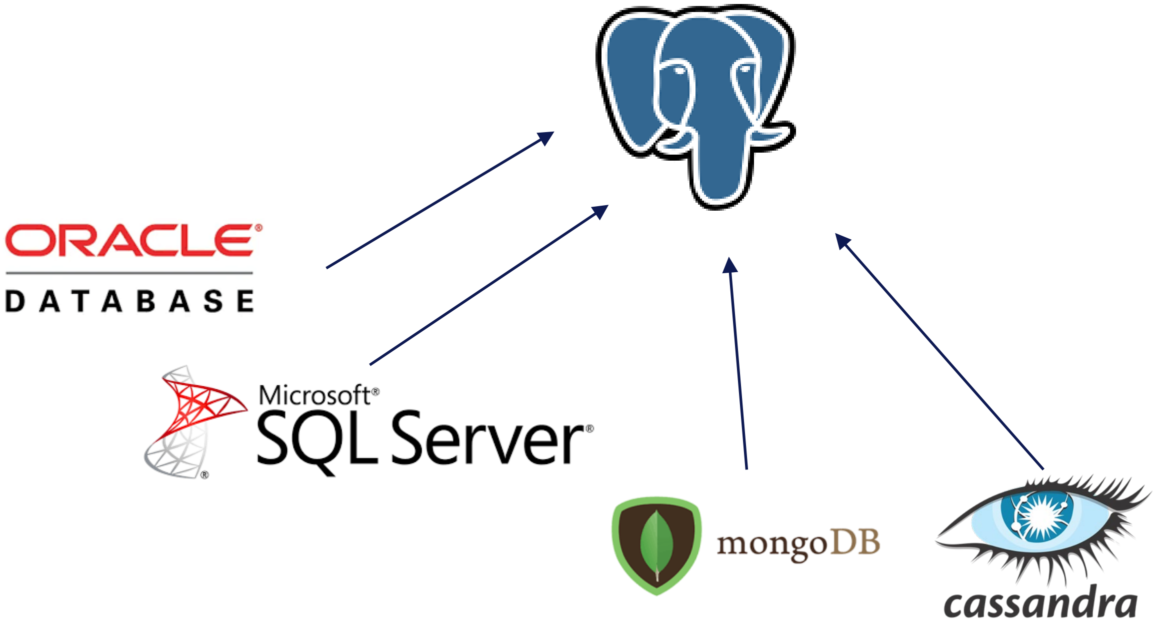
High conversion rates from Other databases to PostgreSQL

- PL/Python
  - plpy.commit()
  - plpy.rollback()
- PL/Tcl
  - Commit
  - rollback
- PL/Perl
  - spi_commit()
  - spi_rollback()

Interoperability

Challenge of migrating one system at a time.

Foreign Data Wrappers
Replication
Change Data Capture



Sharding

Native Sharding Capabilities

```
CREATE FOREIGN TABLE [ IF NOT
EXISTS ] table_name
  PARTITION OF parent_table [ (
  { column_name [ WITH OPTIONS ]
[ column_constraint [ ... ] ]
    | table_constraint }
    [, ... ]
) ] partition_bound_spec
  SERVER server_name
[ OPTIONS ( option 'value' [, ...
] ) ]
```

- Predicate pushdown
- Aggregate pushdown
- Join pushdown
- Partition Wise join

Extensions

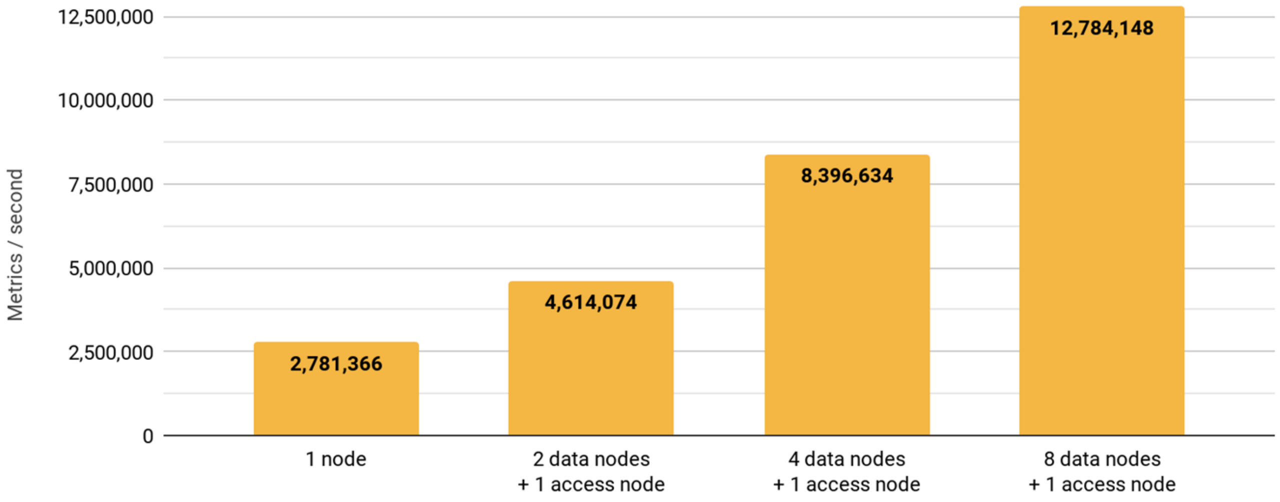
citusdata

Time Series data

Timescale

- 2 million metrics per second on single node.
- ~~Sharding~~ Chunking

Insert performance as cluster size increases



PostGIS-geospatial

- postgis
- postgis_topology
- postgis_sfcgal
- address_standardizer
- fuzzystrmatch
- postgis_tiger_geocoder

```
SELECT num, street, city, state, zip
 FROM parse_address('1 Devonshire Place PH301, Boston, MA 02109');
```

PostGIS

pgRouting

OGR FDW

Pointcloud

Essential features

- SEQUENCES
- USER DEFINED DATATYPES
- TRIGGERS statement and system events
- MATERIALIZED VIEWS
- DML ON VIEWS
- matured STORED FUNCTIONS and PROCEDURES
- PARALLEL EXECUTION

  Parallel Hash Join, Merge Join, Bitmap heap scans, Index scans

Partitioning

- Best among all Open Source Databases
- Native, Declarative Partitioning feature from PG 10
- LIST, RANGE and HASH partitioning
- ATTACH and DETACH partitioning

PG11

- Row migration
- Default Partition
- Automatic Index
- Have foreign key
- Unique Index
- Hash Partition
- Partition wise aggregation

PG12

- Intelligent Planner
- Less locking, faster Inserts
- Bulk copy performance.
- Avoid unwanted [Merge]Append nodes
- Concurrent ATTACH
- Built-in functions
- Expressions as boundaries
- Huge Performance Improvements

PG13

- Performance Improvements
- More cases of partition pruning
- Improved Partition Wise joins
- BEFORE triggers in partitioned tables
- Logical replication of top level table.
  Publisher and Subscriber

Powerful Indexing

- B-Tree
- Hash
- GIN
- GiST
- SP-GiST
- BRIN
- …

- Partial Indexes
- Expression Indexes
- Additional columns in unique indexes
- Full text indexing
- Spatial indexing
- …

Parallel execution of query

- Introduced in PG 9.6
- Partitioning wise parallel joins

PG10 Improvements

- Parallelism by default
- max_parallel_workers
- min_parallel_table_scan_size and min_parallel_index_scan_size
- Parallel B-Tree Index scan
- Parallel Bitmap heap scan
- Parallel Merge joins
- Parallel non-correlated subqueries
- Parallel workers return presorted data
- Parallel query within Procedure languages
- pg_stat_activity shows parallel execution

PG11 Improvements

- CREATE INDEX in Parallel
- CREATE TABLE … AS in Parallel
- CREATE MATERIALIZED VIEW in Parallel
- UNION in Parallel**
- Improved Parallel Hash join
- Improved Parallel Sequential scan
- Partition scans in Parallel
- LIMIT clause to Parallel workers
- WHERE clause aggregate query in parallel
- Functions in target list in parallel
- parallel_leader_participation
- parallel workers' sort activity in EXPLAIN

PG12 Improvements

- Parallel query even in SERIALIZABLE isolation
- Edge case fixes
- Improved parallel pg_dump
- …

Compatibility and migrations

Reasons to choose PostgreSQL

Highest code conversion rates

- Postgres target gets the highest code conversion rates

  Thanks to powerful Procedural language and great list of features
  Thanks to Standards and Versatile features

orafce - Pluggable Oracle compatibility extension

sysdate()
to_date()
add_months()
date + integer
...

nls_date_format
select xxx from dual
...

dbms_output
dbms_output.putline
...

left()
substr()

utl_file
dbms_pipe
dbms_alert
...

oracle.user_tables
oracle.user_tab_columns
oracle.user_cons_columns
oracle.user_constraints
oracle.product_componenent_version
oracle.user_objects
oracle.dba_segments

PERCONA

Schema Migration

ora2pg

aws Schema Conversion tool

migVisor for Google Cloud

Data Migration

ora2pg

ORACLE FDW

ETL Tools
Migration Services
Cross-Database Replication

alooma striim Informatica pentaho ...

Oracle_FDW

- Oracle Instant Client libraries
- Foreign INSERTs, UPDATEs, DELETEs

Oracle FDW

```
CREATE FOREIGN TABLE public.t1 (
        id integer OPTIONS ( key 'true') NOT NULL
)
SERVER xe OPTIONS
  ( schema 'PG', "table" 'T1');
```

- **Updates and Deletes won’t work if key is not defined**

```
postgres=# update t1 set id=3 where id=2;
ERROR:  no primary key column specified for foreign Oracle table
DETAIL:  For UPDATE or DELETE, at least one foreign table column must be marked as
primary key column.
```

- **Intelligent IMPORT FOREIGN SCHEMA**

```
postgres=# IMPORT FOREIGN SCHEMA "PG" FROM SERVER xe INTO public;
```

- Automatically converts numeric(6,0) to INT datatype
- Primary key definition on the Oracle side to to Key definition

Migration using Oracle_FDW

Schema Migration

```
CREATE TABLE t11 ( LIKE t1);
```

Data Migration

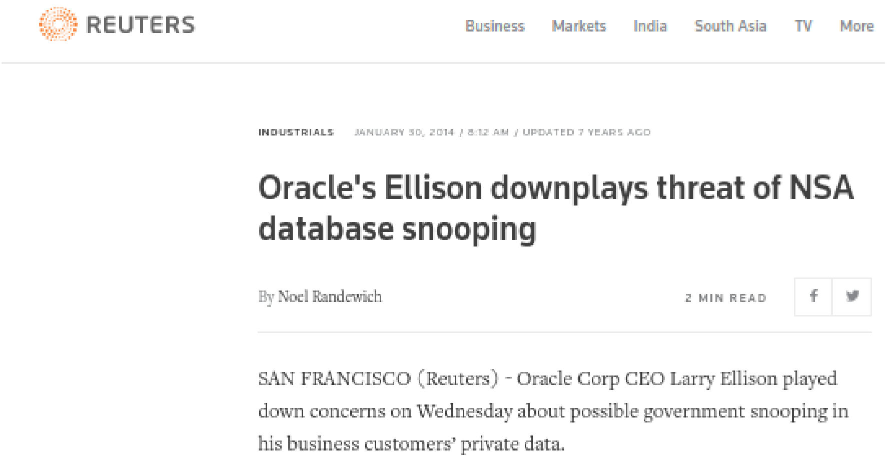
```
INSERT INTO t11 SELECT * FROM t1;
```

Security

Reasons to choose PostgreSQL

Security

- TLS / SSL over Network
- Certificate authentication
  - Certificate Verification at different levels.
- Row Level Security
- Different authentication plugins
- Built-in encryption for table columns
- Built-in Host Based Authentication
- Auditing features
- Audit Extensions



INDUSTRIALS JANUARY 30, 2014 / 8:12 AM / UPDATED 7 YEARS AGO

Oracle's Ellison downplays threat of NSA database snooping

By Noel Randewich 2 MIN READ

SAN FRANCISCO (Reuters) - Oracle Corp CEO Larry Ellison played down concerns on Wednesday about possible government snooping in his business customers' private data.

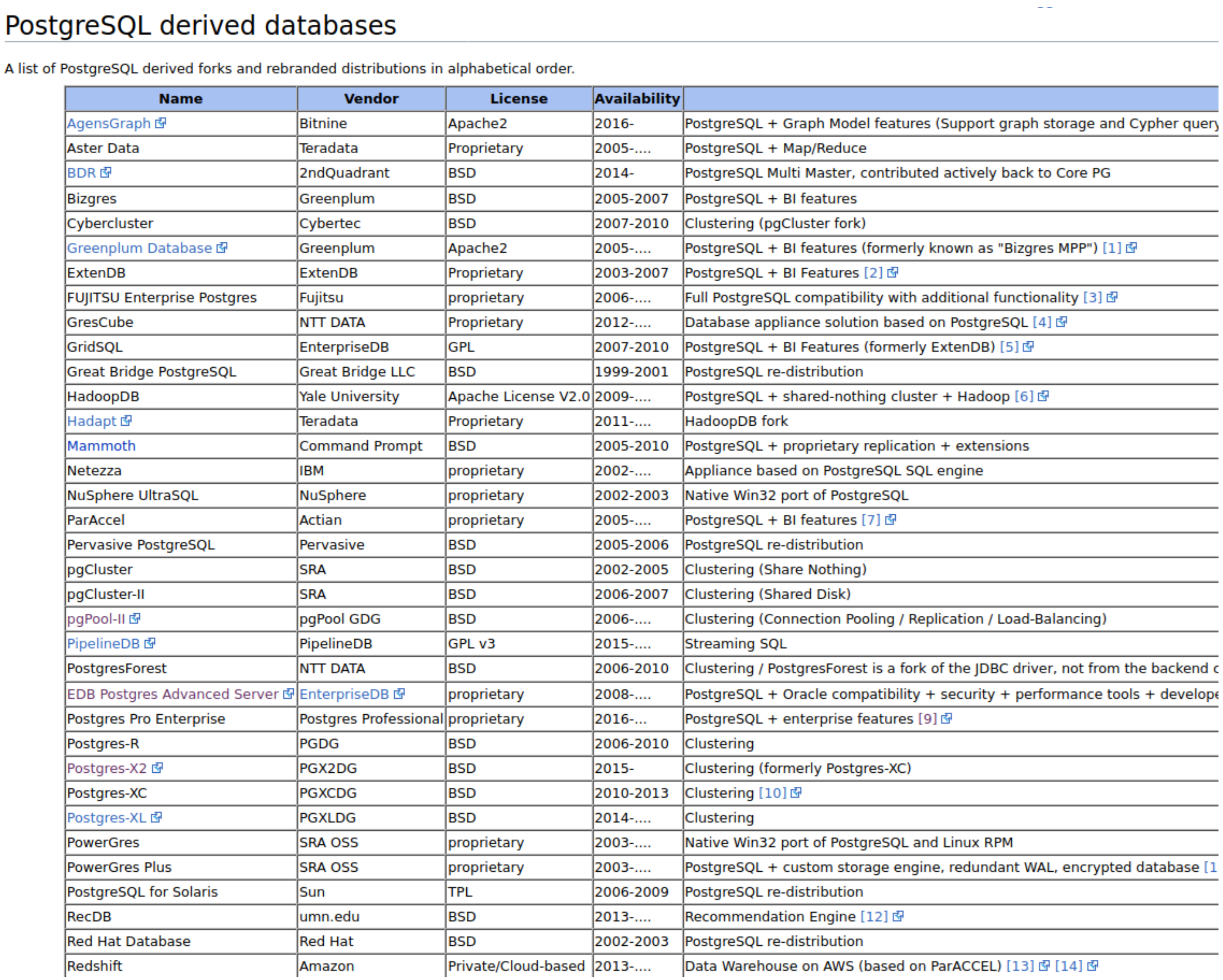
IMPORTANCE OF

- Public auditing of the Source Code
- World wide review
- Source code is the primary means of distribution

Being secure is an attitude & culture, Not just a technology solution.

Closing remarks

Beware!



PostgreSQL derived databases

A list of PostgreSQL derived forks and rebranded distributions in alphabetical order.

Name	Vendor	License	Availability	
AgensGraph	Bitnine	Apache2	2016-	PostgreSQL + Graph Model features (Support graph storage and Cypher query
Aster Data	Teradata	Proprietary	2005-....	PostgreSQL + Map/Reduce
BDR	2ndQuadrant	BSD	2014-	PostgreSQL Multi Master, contributed actively back to Core PG
Bizgres	Greenplum	BSD	2005-2007	PostgreSQL + BI features
Cybercluster	Cybertec	BSD	2007-2010	Clustering (pgCluster fork)
Greenplum Database	Greenplum	Apache2	2005-....	PostgreSQL + BI features (formerly known as "Bizgres MPP") [1]
ExtenDB	ExtenDB	Proprietary	2003-2007	PostgreSQL + BI Features [2]
FUJITSU Enterprise Postgres	Fujitsu	proprietary	2006-....	Full PostgreSQL compatibility with additional functionality [3]
GresCube	NTT DATA	Proprietary	2012-....	Database appliance solution based on PostgreSQL [4]
GridSQL	EnterpriseDB	GPL	2007-2010	PostgreSQL + BI Features (formerly ExtenDB) [5]
Great Bridge PostgreSQL	Great Bridge LLC	BSD	1999-2001	PostgreSQL re-distribution
HadoopDB	Yale University	Apache License V2.0	2009-....	PostgreSQL + shared-nothing cluster + Hadoop [6]
Hadapt	Teradata	Proprietary	2011-....	HadoopDB fork
Mammoth	Command Prompt	BSD	2005-2010	PostgreSQL + proprietary replication + extensions
Netezza	IBM	proprietary	2002-....	Appliance based on PostgreSQL SQL engine
NuSphere UltraSQL	NuSphere	proprietary	2002-2003	Native Win32 port of PostgreSQL
ParAccel	Actian	proprietary	2005-....	PostgreSQL + BI features [7]
Pervasive PostgreSQL	Pervasive	BSD	2005-2006	PostgreSQL re-distribution
pgCluster	SRA	BSD	2002-2005	Clustering (Share Nothing)
pgCluster-II	SRA	BSD	2006-2007	Clustering (Shared Disk)
pgPool-II	pgPool GDG	BSD	2006-....	Clustering (Connection Pooling / Replication / Load-Balancing)
PipelineDB	PipelineDB	GPL v3	2015-....	Streaming SQL
PostgresForest	NTT DATA	BSD	2006-2010	Clustering / PostgresForest is a fork of the JDBC driver, not from the backend c
EDB Postgres Advanced Server	EnterpriseDB	proprietary	2008-....	PostgreSQL + Oracle compatibility + security + performance tools + develope
Postgres Pro Enterprise	Postgres Professional	proprietary	2016-...	PostgreSQL + enterprise features [9]
Postgres-R	PGDG	BSD	2006-2010	Clustering
Postgres-X2	PGX2DG	BSD	2015-	Clustering (formerly Postgres-XC)
Postgres-XC	PGXCDG	BSD	2010-2013	Clustering [10]
Postgres-XL	PGXLDG	BSD	2014-....	Clustering
PowerGres	SRA OSS	proprietary	2003-....	Native Win32 port of PostgreSQL and Linux RPM
PowerGres Plus	SRA OSS	proprietary	2003-....	PostgreSQL + custom storage engine, redundant WAL, encrypted database [1
PostgreSQL for Solaris	Sun	TPL	2006-2009	PostgreSQL re-distribution
RecDB	umn.edu	BSD	2013-....	Recommendation Engine [12]
Red Hat Database	Red Hat	BSD	2002-2003	PostgreSQL re-distribution
Redshift	Amazon	Private/Cloud-based	2013-....	Data Warehouse on AWS (based on ParACCEL) [13] [14]

24th September 2020: PostgreSQL 13 Released!

PostgreSQL: The World's Most Advanced Open Source Relational Database

Beware :

- Vendor lock-ins
- Cloud lock-ins
- Bugs and Security Vulnerabilities
- Data directories, Binaries, Features won’t be compatible for PostgreSQL Derived , Feature compatible softwares

Key takeaways

1. PostgreSQL is getting more and more popular for different workloads

   It's been growing exponentially over the years

1. PosgreSQL is loved and wanted by developers

   It's in the top of the independent rankings

1. Any group in the organization can benefit form Postgres, not only devs

   Management, Architects, Operations, Security, and more

1. PostgreSQL is stable, mature, and the ecosystem is rich - great target for Oracle (and other DBs) migrations

   Its features will please any seasoned database engineer and make migration possible with reasonable efforts

1. Be aware of traps - stay with open source

   Alternative licenses and Postgres derivative projects come with benefits but also certain risks

PERCONA

Databases run better with Percona