Getting Started with Postgresql

For **Charlotte SQL Saturday on 10/20/2018**

By Ramu Pulipati

Sponsors


SQL Chick


COZYROC


SQLCoop


cardinal


SentryOne


DELL EMC


redgate


SQL Swimmer


Dallas DBAs


IDERA


signature
CONSULTANTS


Quest

About Me

Cofounder / CTO at Botsplash

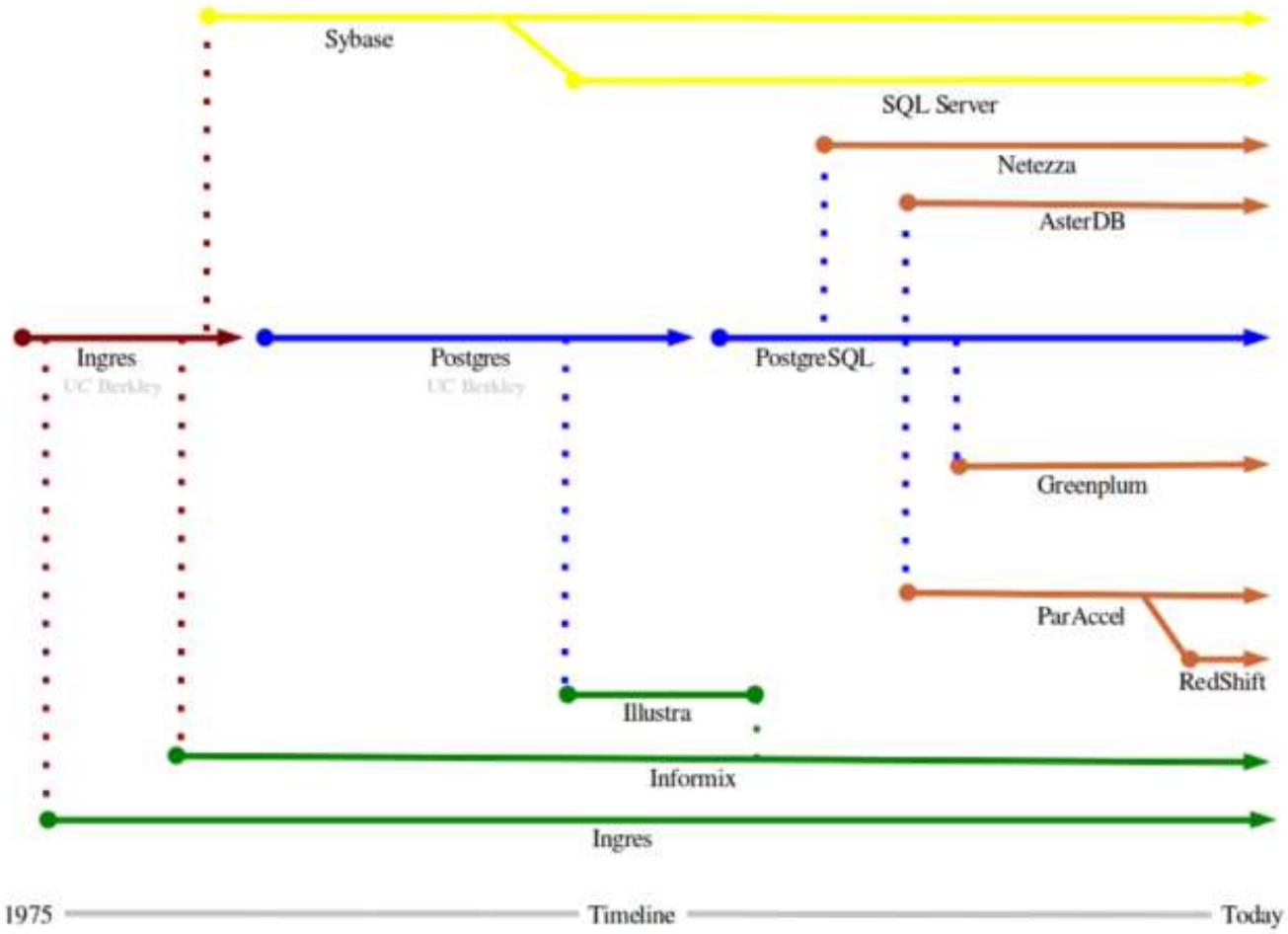
Software-as-a-service omnichannel messaging platform to engage businesses and customers.

botsplash.com

About Postgresql

- Free – Open Source RDMS alternative to Microsoft SQL Server, MySQL, Oracle, Mongo DB or any RDBMS and NoSQL databases
- Battle tested for over 30 years with continuous improvements to support modern applications
- Rich ANSI/ISO Complaint SQL, JSON and Multiple Languages support
- Built-in and extensions for high scalability, geolocation and time series based applications
- Hosted Providers and Commercial Support
- Used for embedded database, web applications and data warehouses

History & What's new

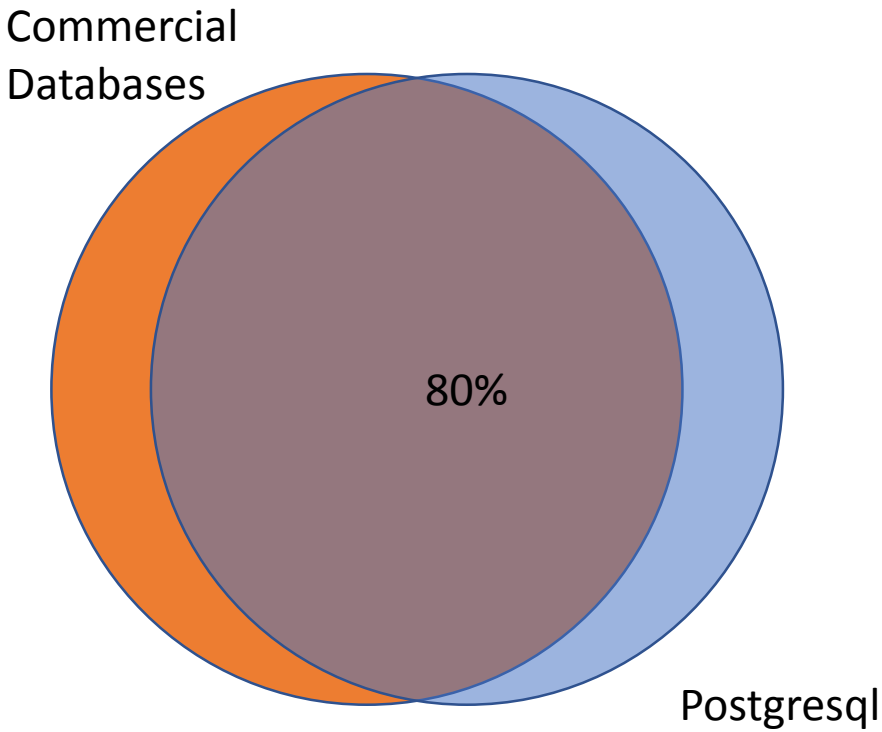


Version	Major Feature
V11 ***10/18/2018***	Partition & Index Improvements Query Parallelism Improvements Just-in-time compilation
V10 *10/05/2017*	Declarative Table Partitions Logical Replication Improved Monitoring and Control
V9.6 *09/29/2016*	Parallel Query Foreign Data Wrappers Replication
V9.5 *07/01/2016*	JSONB Modifying Operators Row level security UPSERT statement
V9.4 *12/18/2014*	Introduce JSONB Materialized View Improvements

Source: [Postgresql Releases]

Capabilities

- Full featured Database
  - ACID Transactions
  - Mature Server Side Programming
  - Hot Standby and High Availability
  - Online backups
  - Point-in-time recovery
  - Native Table Partitioning
  - Spatial Functionality
  - Full Text Search
  - JSON and HStore support



Source: [Introduction to Postgresql Slideshare]

Capabilities (contd …)

- Replication and Read Replicas
- Built-in pub-sub queue
- Multiple schemas with security
- Row level security
- Native SSL Support
- Data Level Encryption
- Extensions
  - Database sharding
  - Timeseries data
- Multi-tenancy
- Extensible and Customizable

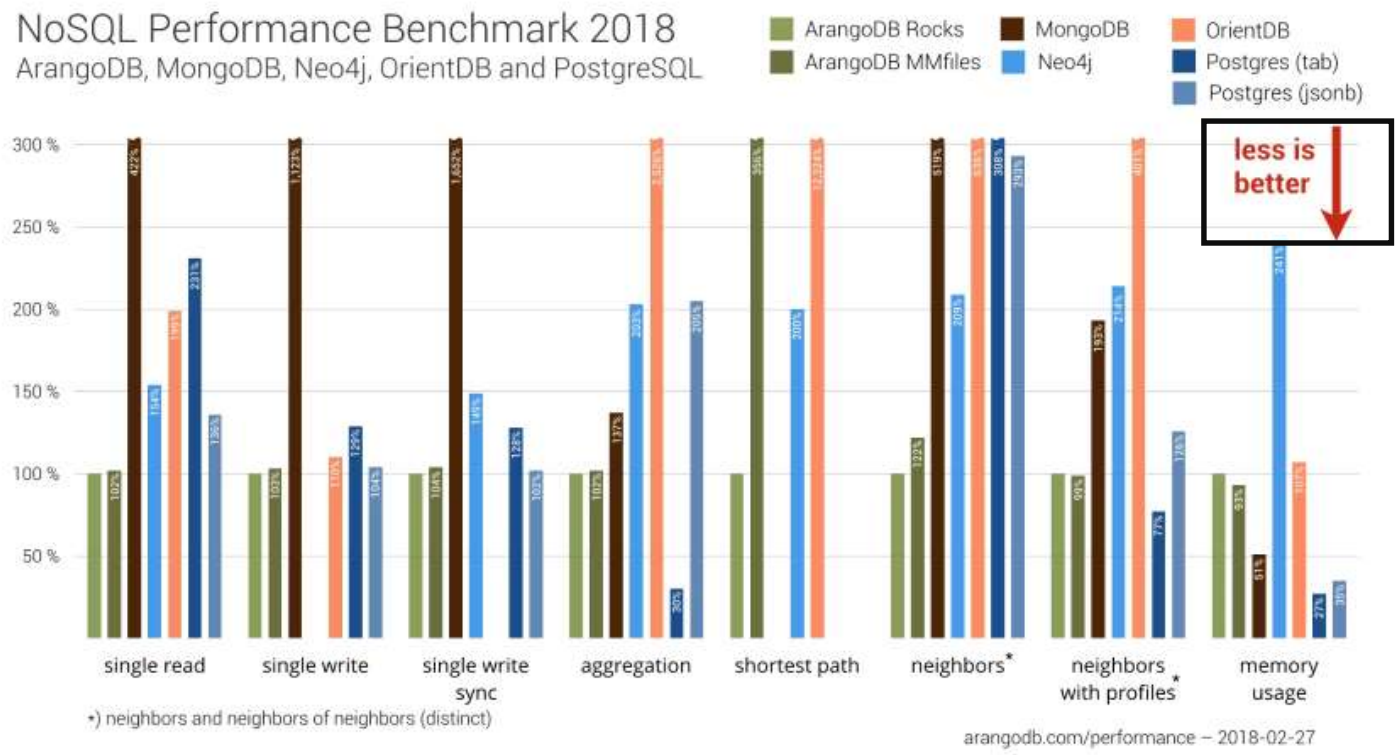
Licensing & Community

- BSD License open for Open Source, Private and Commercial use
- No vendor Lock-in
- Predictable releases
- Faster Bug fixes
- Active Community

Source: postgresql.org

Compare to RDBMS

- MySQL
  - Has better write performance but Postgres is ACID compliant and has better features, reliability and consistency.
- SQL Server
  - Featureful with recently introduced JSON support. Postgres has superset of features and scalable with extensions.
- Oracle
  - Most comprehensive database but very expensive.
- Mongo Db
  - Postgresql outshines read/write for unstructured with JSONB.



Source: [NoSQL Performance Benchmark 02/27/2018]

COMPANIES USING POSTGRESQL

Uber

Instagram

CISCO

skype

Bloomberg

twitch

HEAP

CLOUDFLARE

DERIVED

NETEZZA
an IBM Company

amazon REDSHIFT

Greenplum

TERADATA. ASTER

EXTENSIONS

citusdata

TIMESCALE

EDB
ENTERPRISEDB

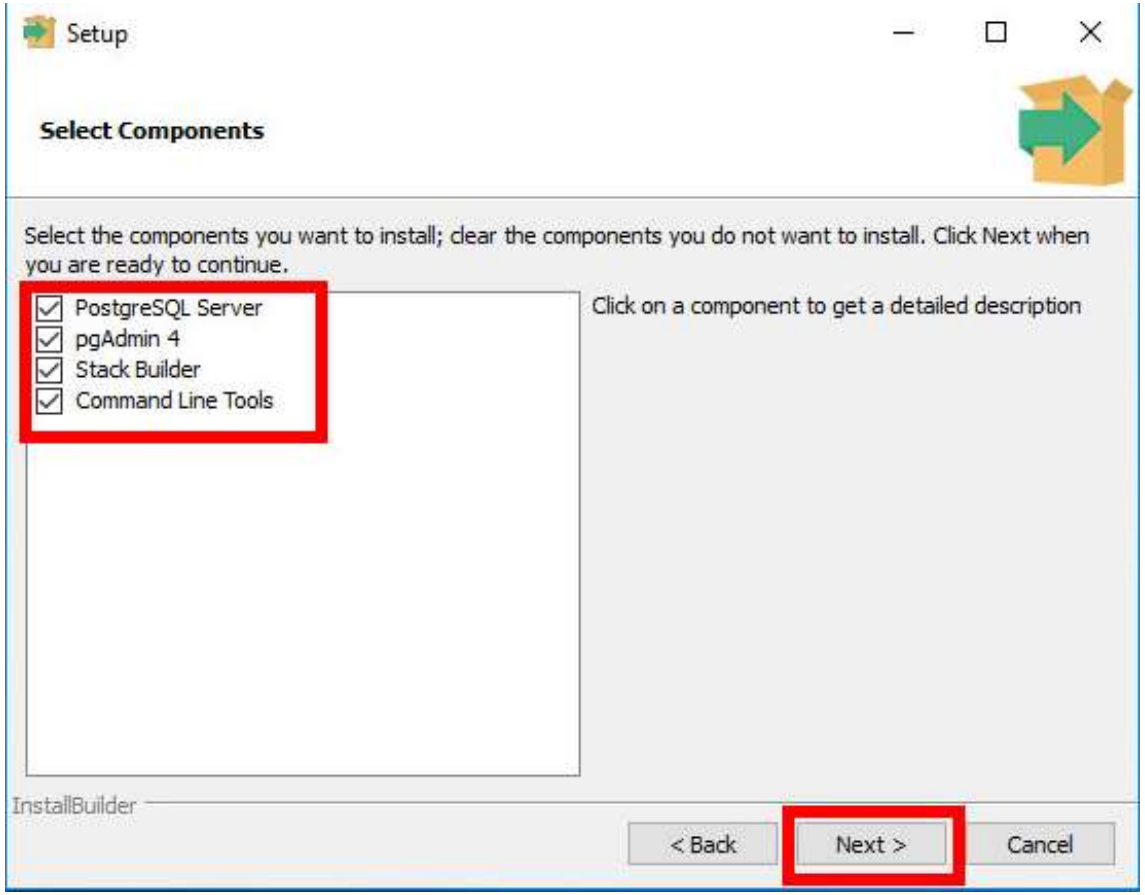
Features

- Affordability
- Technology
- Security
- Flexibility
- Stability
- Extensibility
- Reliability
- Predictability
- Community
- Auditability

Source: Introduction to Postgresql Slideshare

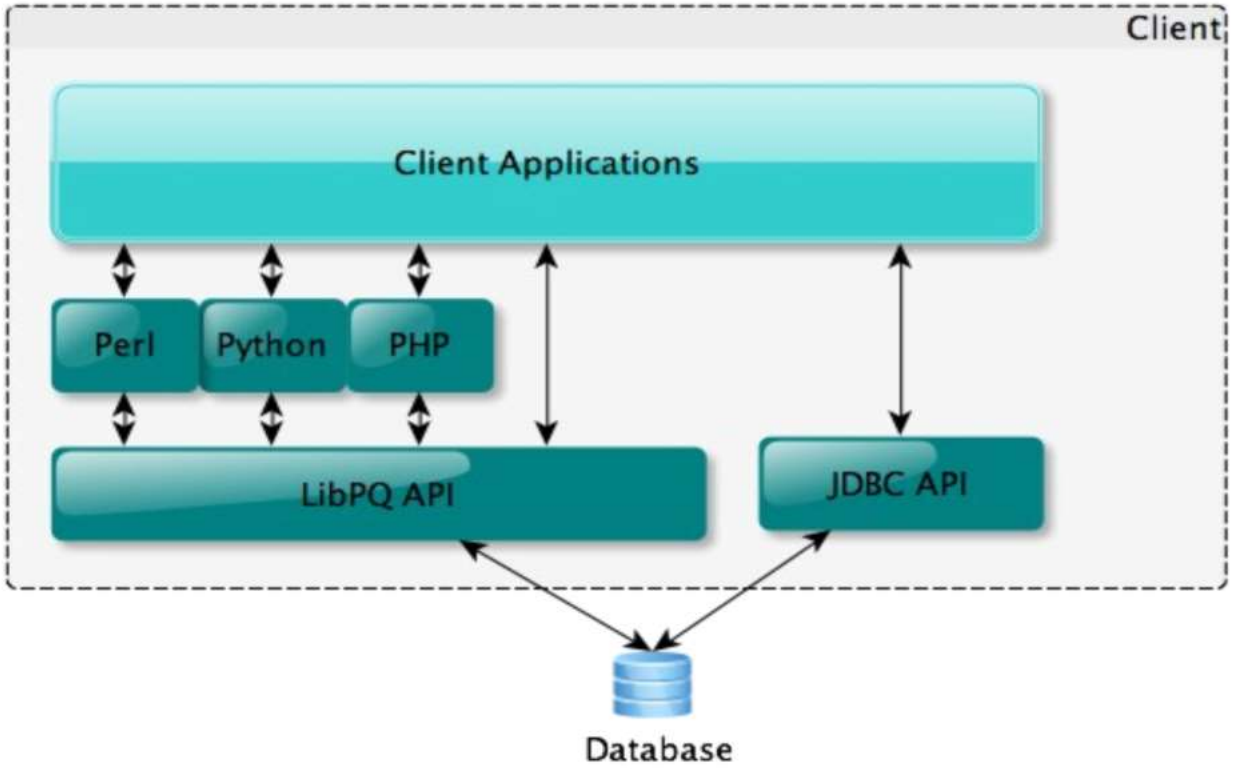
Getting Started

- Download and Install
  - Windows packaged from EnterpriseDb
  - Linux from apt or source
  - Mac installations using homebrew.
- Production recommended on *nix systems. Postgres 11 supports Windows 2012 R2 deployment.
- Cloud hosted databases from Compose, Citus Data, AWS RDS, Goolge CloudSql, Azure Databases are excellent choice.
- Be careful with Docker/Kubernetes deployments.



Postgresql Client

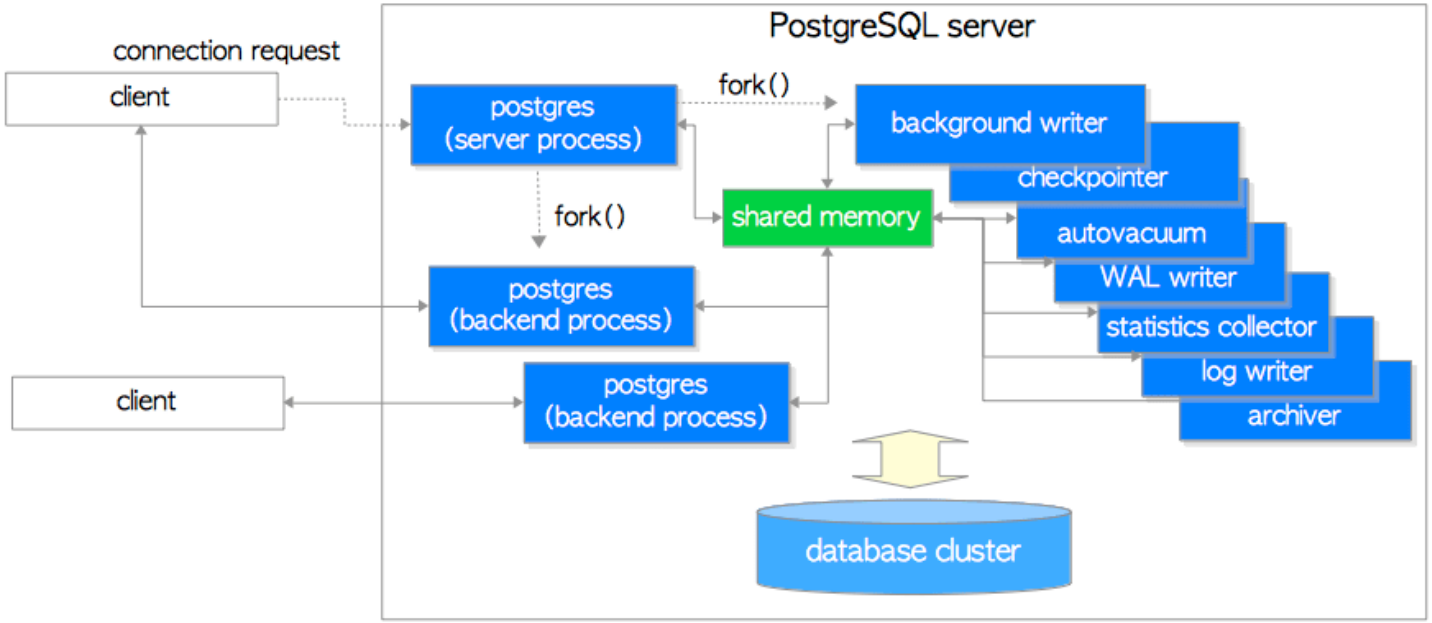
- Instance accessed from TCP/IP Port: 5432 default
- CLI/IDE Clients
  - PSQL, PgAdmin 4, PGCLI, TablePlus, Navicat
- Application Clients
  - Python SqlAlchemy, NodeJs Pg, C# Client, Golang pq



Sample Connection String: **postgres://username:password@127.0.0.1:5432/northwind**

Postgresql Architecture

- Multi-process architecture
  - Primary postmaster
  - Per connection backend process
  - Background maintenanc processes
- Use pgBouncer to load balance large number of connections



Source: [The Internals of Postgresql]

Schema and SQL Support

- Basic and Complex data types
- Complex Queries
- Functions, Operators and Aggregate functions
- Transactional Integrity
- Indexes
- Procedural Language
- Optimized for data storage
- Extend and create custom data types
- Error handling

Data Type	Commonly Used Names
Number	Smallint, integer, Bigint, decimal, Money
Number (auto)	Serial, BigSerial
Character	Char, nchar, Text
Binary	Bytea
Date Time	Timestamp with timezone, Timestamp without Timezone, interval
Collections	Array, JSON, JSONB
Misc	Boolean, UUID, Geo Types, Network Address Mac Address, Custom Types
Full Text Search	TsVector, TsQuery

See details from tutorials point

Demo Sample Table and DDL functions

```sql
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

DROP TABLE IF EXISTS "accountMatches";

CREATE TABLE "accountMatches"
(
    "id"                uuid NOT NULL DEFAULT uuid_generate_v4(),

    "accountId"         uuid NOT NULL REFERENCES "accounts"(id) ON DELETE CASCADE,
    "priority"          smallint NOT NULL,

    "matches"           JSONB NOT NULL DEFAULT '[]',

    "createdAt"         timestamp with time zone NOT NULL DEFAULT now(),
    "createdBy"         bigint NOT NULL,

    PRIMARY KEY ("id")
);
ALTER TABLE "accountMatches" OWNER TO accountdbo;

ALTER TABLE "accountMatches" DROP COLUMN "priority";

ALTER TABLE "accountMatches" ADD COLUMN "ownerId" bigint;

ALTER TABLE "accountMatches"
  ADD CONSTRAINT "accountMatches_owner_fkey"
  FOREIGN KEY ("ownerId") REFERENCES "owners"(id)
  ON DELETE CASCADE;
```

Table DDL

```sql
CREATE OR REPLACE FUNCTION next_member_id(account_id uuid)
RETURNS bigint AS $$
    DECLARE member_key TEXT;
    DECLARE member_key_found INTEGER;
    DECLARE start_position INTEGER;
BEGIN
    member_key = 'member_' || replace(lower(account_id::text), '-', '');
    SELECT COUNT(*) INTO member_key_found FROM pg_class where relname = member_key;
    IF member_key_found = 0
    THEN
        SELECT  (COUNT(*) + 1) INTO start_position
        FROM    "members"
        WHERE   "accountId" = account_id;

        EXECUTE 'CREATE SEQUENCE ' || member_key || ' START ' || start_position::text;
        EXECUTE 'GRANT ALL PRIVILEGES ON SEQUENCE ' || member_key || ' TO public';
    END IF;
    RETURN nextval(member_key);
END;
$$ LANGUAGE PLPGSQL;
```

Sample Function

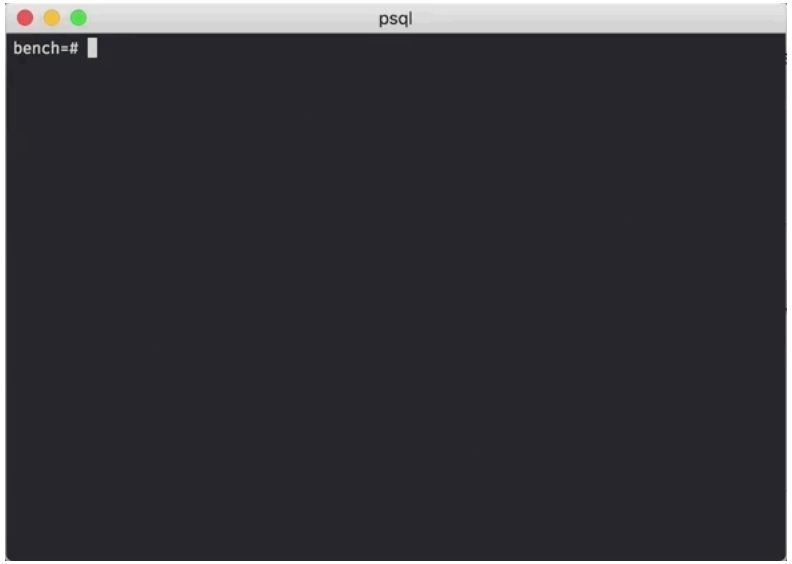
More SQL / NoSQL Support

- Case Sensitive Schema Names
- Case Sensitive Data Storage
- Type casting
- Date Time formatting
- Common Table Expressions
- Array Data Types
- Upserts

- Pivot Tables crosstab function
- Materialized Views
- HStore
- JSONB Structure
  - Rich Functions Support
- EXPLAIN ANALYZE
- Pg_stat

Psql tricks and tips

Command	Description
\l	List of all databases
\c <db>	Change database name
\d	Show all objects in database
\d <item>	Describe database item
\x [auto]	Show one record at a time
\timing	Turn on/off sql timing
\e	Edit in an editor (vi or preferred editor)
\copy …	Save results to CSV file
\h	Help with SQL
Ctrl + R	Search previous SQL command
\ir	Reference SQL file



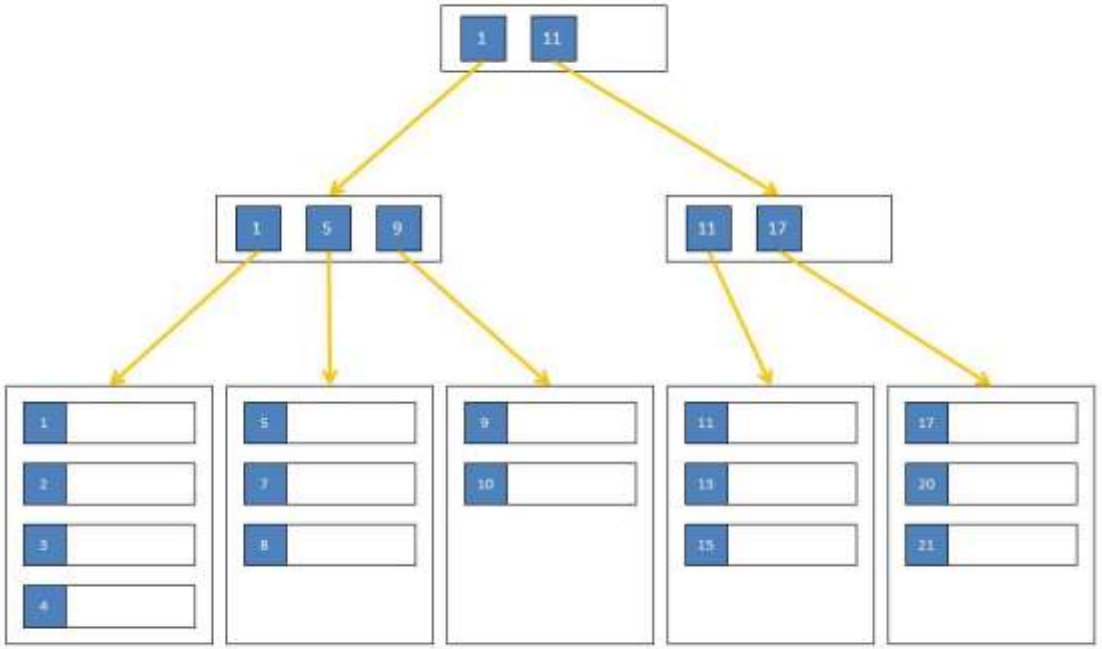
Source: pgdash.io

Concurrency - MVCC

- Postgres uses “Multi-version concurrency control”. More details from postgres internals or interdb.
- All reads are from the snapshot of the database until committed.
- Advantage of this method is reading does not block writing and vice-versa. Isolation at “read-committed” level
- This method supports table level and row level locking as well.
- Other databases that use this technique are Oracle, CouchDB, etc

Indexes

- Index Support
  - Single Column
  - Multi Column
  - Unique
  - Partial Indexes
  - Expression Indexes
  - Implicit Indexes (PK, UK)
- Index Types
  - B-Tree: Balanced Trees used in most databases
  - Generalized Inverted Index (GIN): Useful for full text search
  - Generalized Search Tree (GIST): Useful for geometric and full text search



B-Tree

Disadvantages / Limits

- Database only solution
- Few available tools compared to commercial. PgAdmin is catching up but relatively new.
- CPU bound queries and table inheritance based partitioning before v10.
- Limited Talent pool

Limit	Value
Maximum Database Size	Unlimited
Maximum Table Size	32 TB
Maximum Row Size	1.6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited
Maximum Columns per Table	250 - 1600 depending on column types
Maximum Indexes per Table	Unlimited

Source: [Postgres limits from sreenstepslive.com]

Source: https://www.youtube.com/watch?v=6p2TNPabt6M

Best Practices

- Performance Optimization
  - Do not read from Database at all
  - Use indexes efficiently
- Use pg_stats collector to monitor usage and performance
- Vacuum regularly to clean up storage
- Bench mark hardware and optimize pg parameters
- Always use SSL for connectivity in untrusted networks

References and Follow up

- Introduction to Postgresql slideshare
- The Internals of Postgresql
- Postgresql excercises
- Postgres Blogs: 2nd Quadrant, Percona and Citus Data, Citus - Craig Kerstiens

Questions? Contact me at ramu@botsplash.com