# Relational databases – deeper look into PostgreSQL

Michał Zaborowski

# Michał Zaborowski

- **For about 18 years Developer, Engineer.**
- **For last 3 consultant.**

https://github.com/TeXXaS/postgresql

www.linkedin.com/in/texxas

michal@zaborowski.info.pl

michal.zaborowski@gmail.com

# Agenda

- Intro
- Proceses
- Data storage
- Transactions
- Indices
- Locks
- Query execution process
- Replication

# Database – what for?

- **Data storage as well defined problem**
- **Relational algebra**
- **ACID**

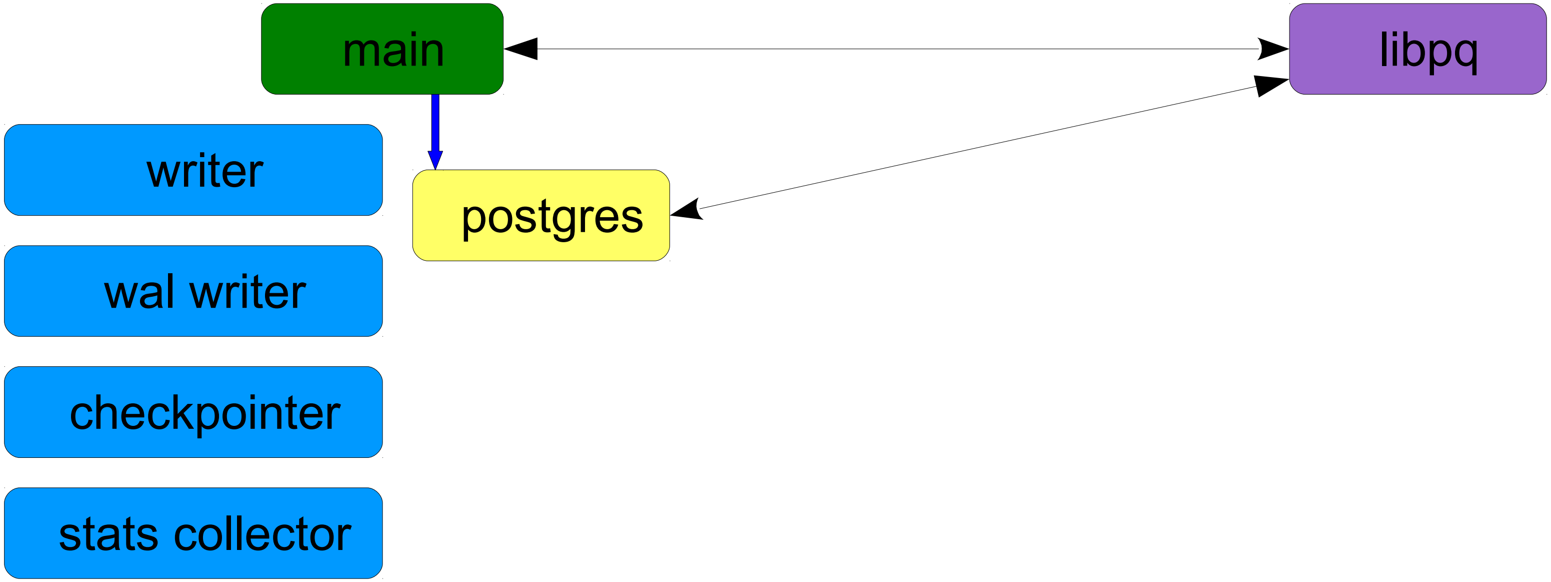
- **More-or-less common interface:**
  - Structured Query Language.

# Database – what for?

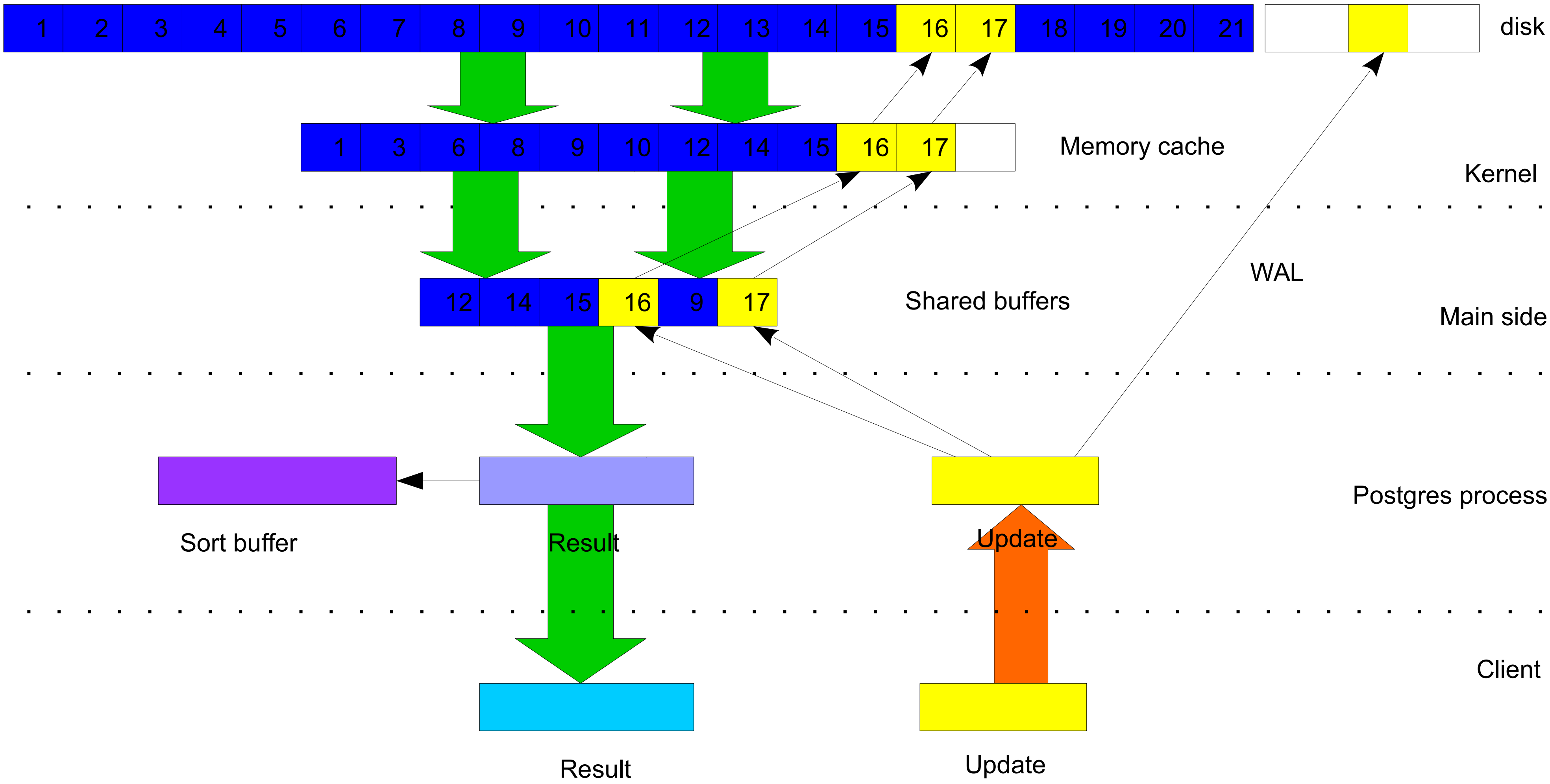
- **Data storage as well defined problem**
- **Relational algebra**
- **ACID**

- **More-or-less common interface:**
  - Structured Query Language.

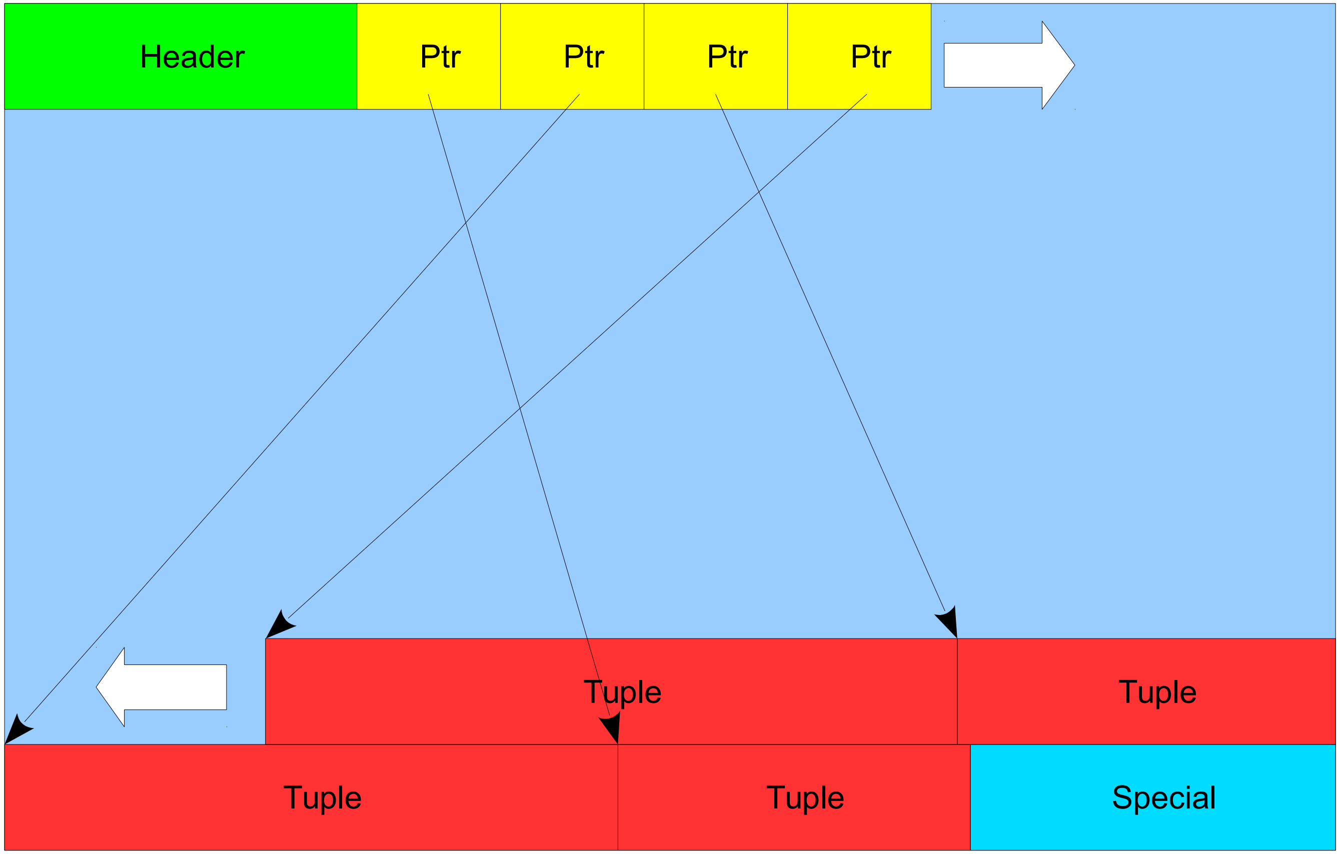
# Processes


main
libpq
writer
postgres
wal writer
checkpointer
stats collector

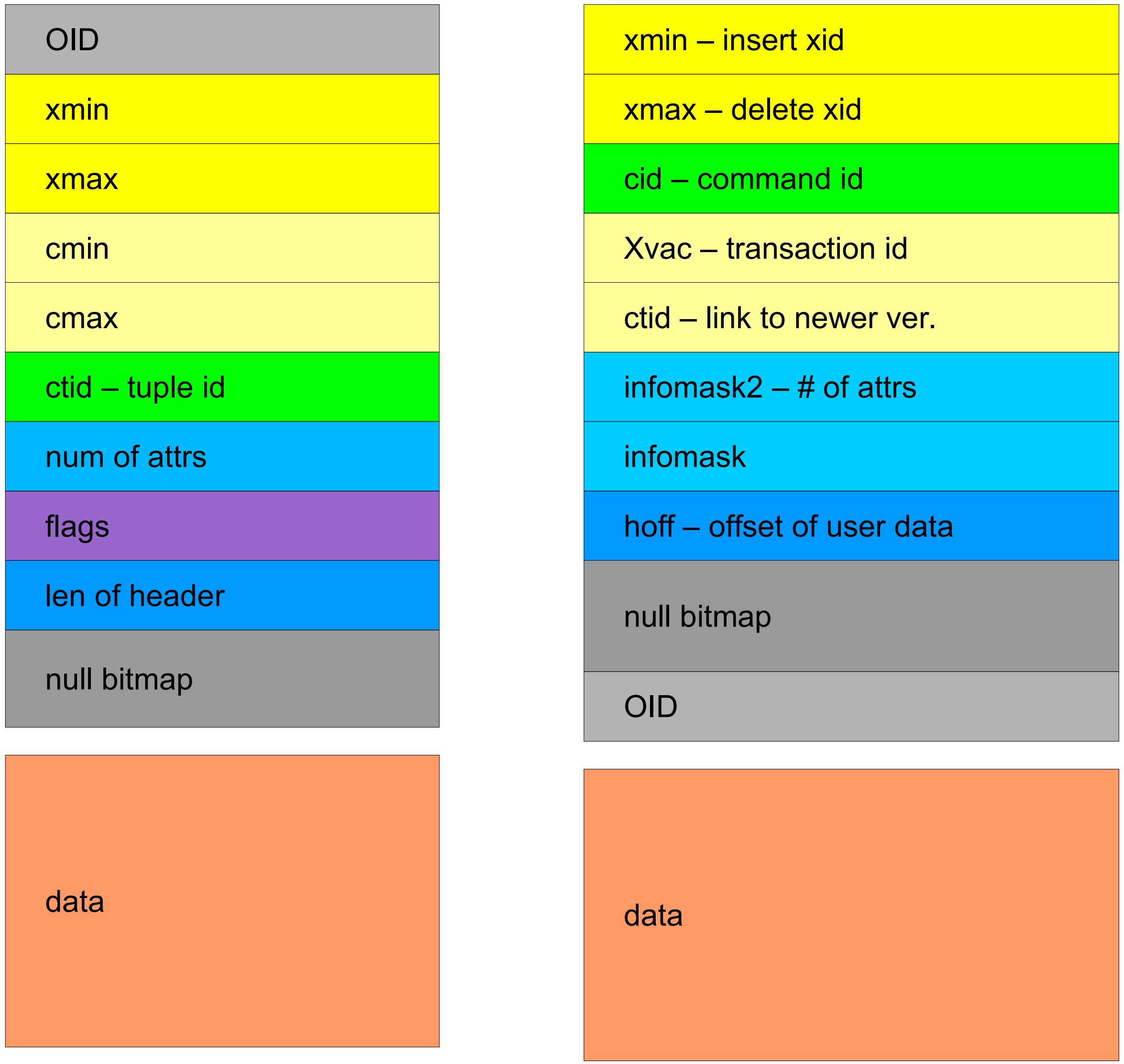
# Memory management



# Data storage – The Page



# Tuple



OID

xmin

xmax

cmin

cmax

ctid – tuple id

num of attrs

flags

len of header

null bitmap

data

xmin – insert xid

xmax – delete xid

cid – command id

Xvac – transaction id

ctid – link to newer ver.

infomask2 – # of attrs

infomask

hoff – offset of user data

null bitmap

OID

data

# MVCC

- No override on tuple level
- Every change is wrote in new tuple, as a new version
- DB has ”Snapshot” of transactions

# WAL

- **C-log**
- **X-log**
- **Chceck points**

# Table

- **How it looks on disk.**
- **TID**
  - Page id
  - Tuple id
- **TOAST**

# Vacuum

- **Space releasing**
- **Transactions management**

# Vacuum – modern way

- **Space releasing**
  - Free Space Map
  - FSM per table
- **Vacuum map**
- **Transactions management**

# Transaction

- **Change management**
  - MVCC
  - Redo / undo logs
- **Multi phase transactions**
- **Transactions and stored procedures**

# ACID

- **Atomic – whole transaction appears, instantly**
- **Consistent – all constraints applied**
- **Isolated – transactions do not interfere**
- **Durable – if job is done, it is done forewer**

# Transaction isolation

| Level | Dirty read | Non repetable Read | Phantom read |
|---|---|---|---|
| Serializable | Not possible | Not Possible | Not Possible |
| Repetable reads | Not possible | Not Possible | Possible |
| Read committed | Not possible | Possible | Possible |
| Read uncommited | Possible | Possible | Possible |

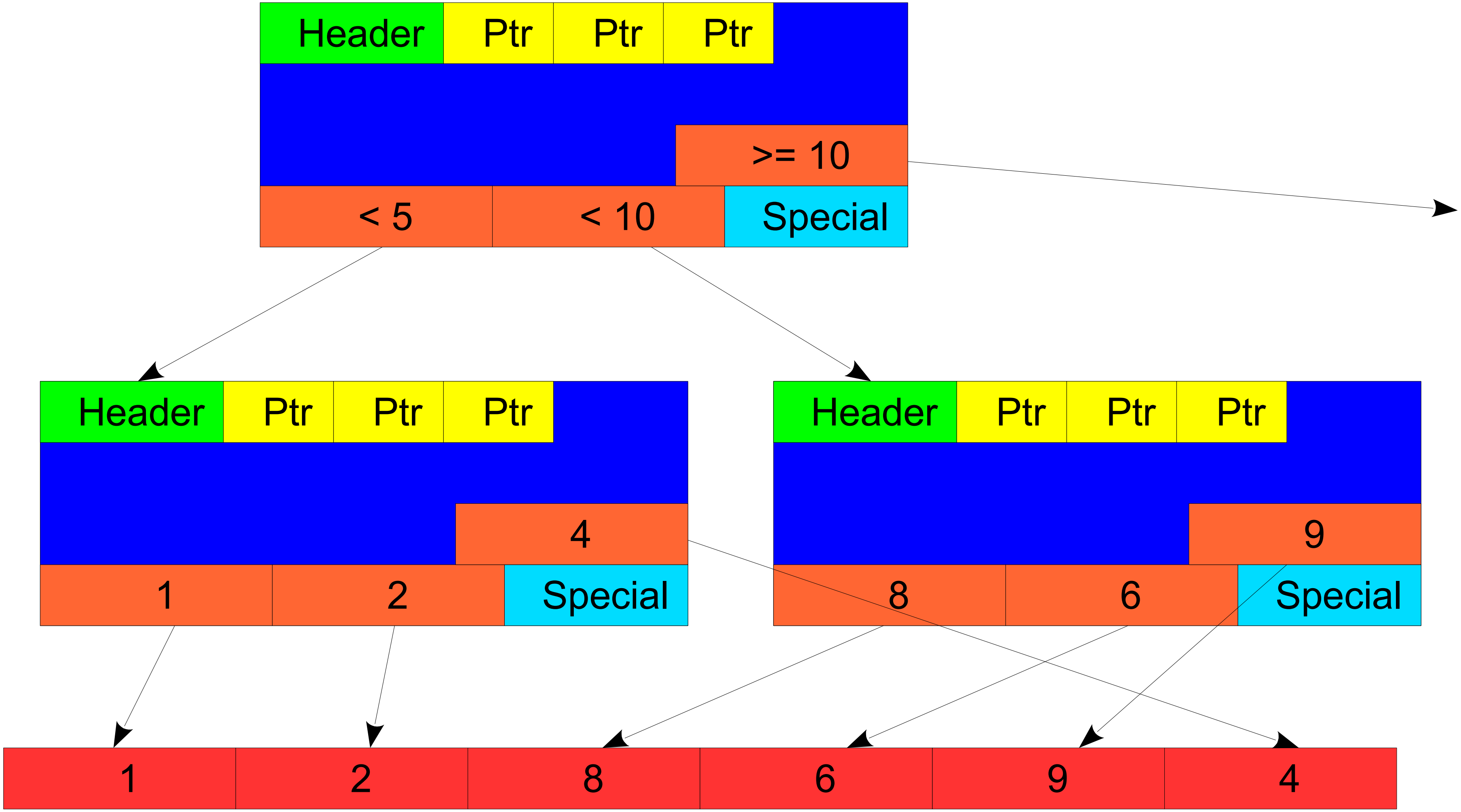
# Indices

- **How to index:**
  - B-tree
  - Hash,
  - Bitmap,
  - BRIN
- **Storing data in indices.**

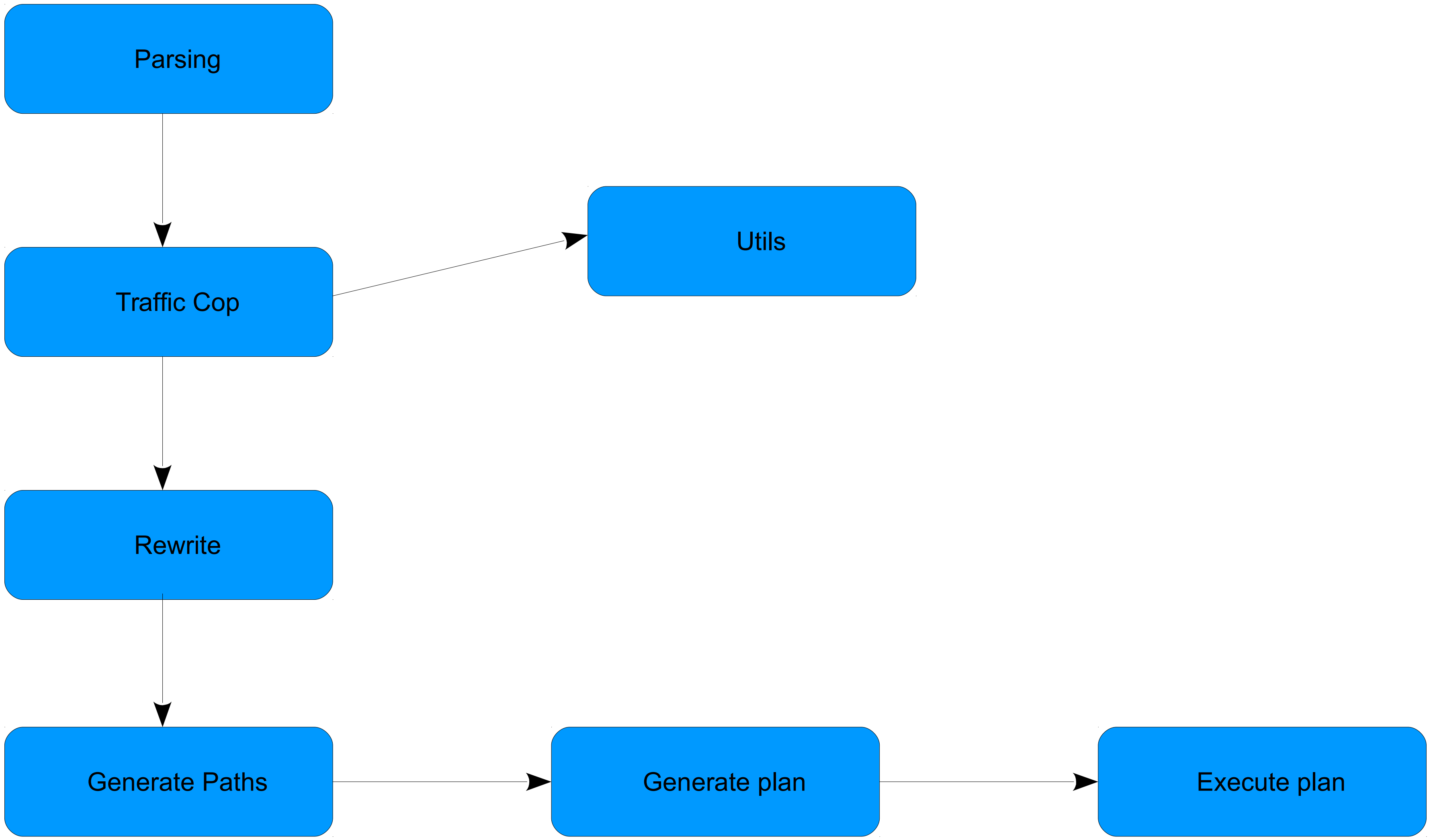
# Indices II

- **How to index:**
  - Partial index
  - Functional index
- **Full text search**
- **GIST**
- **GIN**

# Index storage



# Query Execution



# Types and operators

- **Types:**
  - Length of data can vary.
  - varchar vs text.
- **Comparison operators:**
  - ==, !=, <>, <, >, <=, >=
- **Other:**
  - @>, <@
- **Be sure that types match**

# Locks

| Mode | Code example |
|---|---|
| Access Share | SELECT |
| Row Share | SELECT … FOR UPDATE |
| Row Exclusive | INSERT, UPDATE, DELETE |
| Share | CREATE INDEX |
| Share Row Exclusive | EXCLUSIVE MODE, allows ROW SHARE |
| Exclusive | Locks ROW SHARE |
| Access Exclusive | ALTER TABLE, DROP TABLE, VACUUM, LOCK TABLE |

Deferred

# Locks II aka Row Level Locks

| Requested | Current Lock Mode | | | |
|---|---|---|---|---|
| | For Key Share | For Share | For No Key Update | For Update |
| For Key Share | | | | X |
| For Share | | | X | X |
| For No Key Update | | X | X | X |
| For Update | X | X | X | X |

# Query Execution II

- **Data accessing methods:**
  - Seqscan
  - Index scan / Index only scan
  - Bitmap scan
- **Table joining methods:**
  - Nestled loop
  - Hash join
  - Merge join

# Replication

- **Clustering – shared storage**
- **Multimaster**
- **Master / slave**

# Replication – PG way

- **Old way**
  - Query replication - slony
- **Log shipping / streaming**
  - Master + standby
  - Master + slaves

# Point In Time Recovery

- WAL Archiving
- SELECT pg_start_backup('label');
- Copy DB files
- SELECT pg_stop_backup();

# Pro Tips

- Update highly used table
- Processing order vs locking
- Admin vs developer - monitoring
- Data mining / long operations
- Denormalizations
- Partitioning
- Values

# Update highly used table

- Alter table user add column phone varchar(10) not null default ’112’;
- Long exclusive lock…

# Update highly used table II

- **Change only metadata**
  - Alter table user add column phone varchar(10);
- **Fix data (in blocks)**
  - Update user set phone = ’112’;
- **Fix metadata**
  - Alter table user alter column phone set not null;

# Processing order

- **Problem description:**
  - Forums, with threads, posts.
  - Counters on forums, thereads.
  - Moving posts between threads sometimes fails.

# Admin vs. Developer

- **Developer**
  - Trace bug
  - Fixing issues
  - Checking processing
- **Admin**
  - Log scanning
  - Top 10 longest queries
  - Forecasting

# Data mining

- **The longer operation takes, the more thinking before...**

# Denormalization

- **Arrays**
  - List of keys
- **H-store**
  - Key-Value inside column, indexable

# Partitioning

- **Types of partitions:**
  - Round-robin,
  - Range,
  - List.

# Future reading / bibliography

- **PostgreSQL docs:**

  http://www.postgresql.org/docs/current/static/index.html

- **depesz blog, projects**
  - https://www.depesz.com/
- **Bruce Momjian writings**
  - http://momjian.us/main/writings/pgsql/
- **Docs from source code**
  - http://doxygen.postgresql.org/index.html