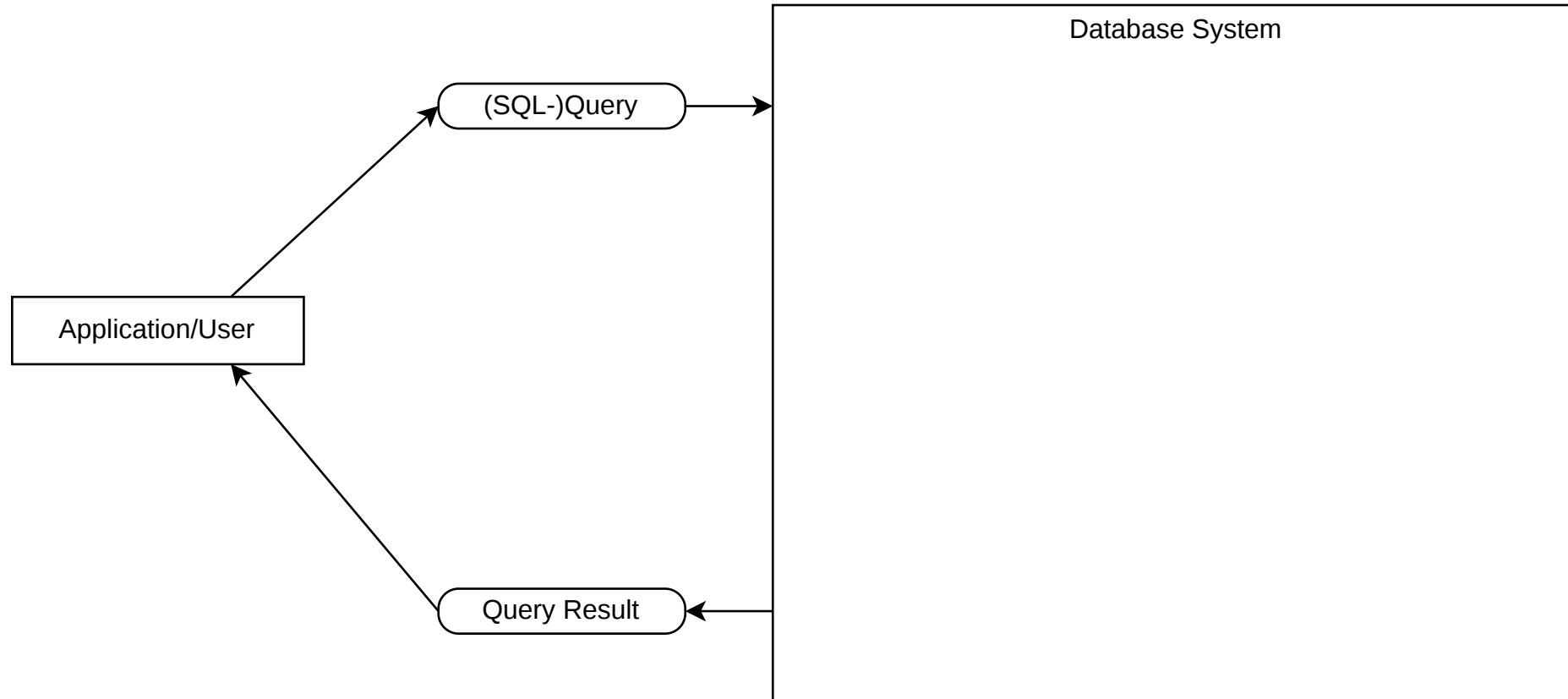




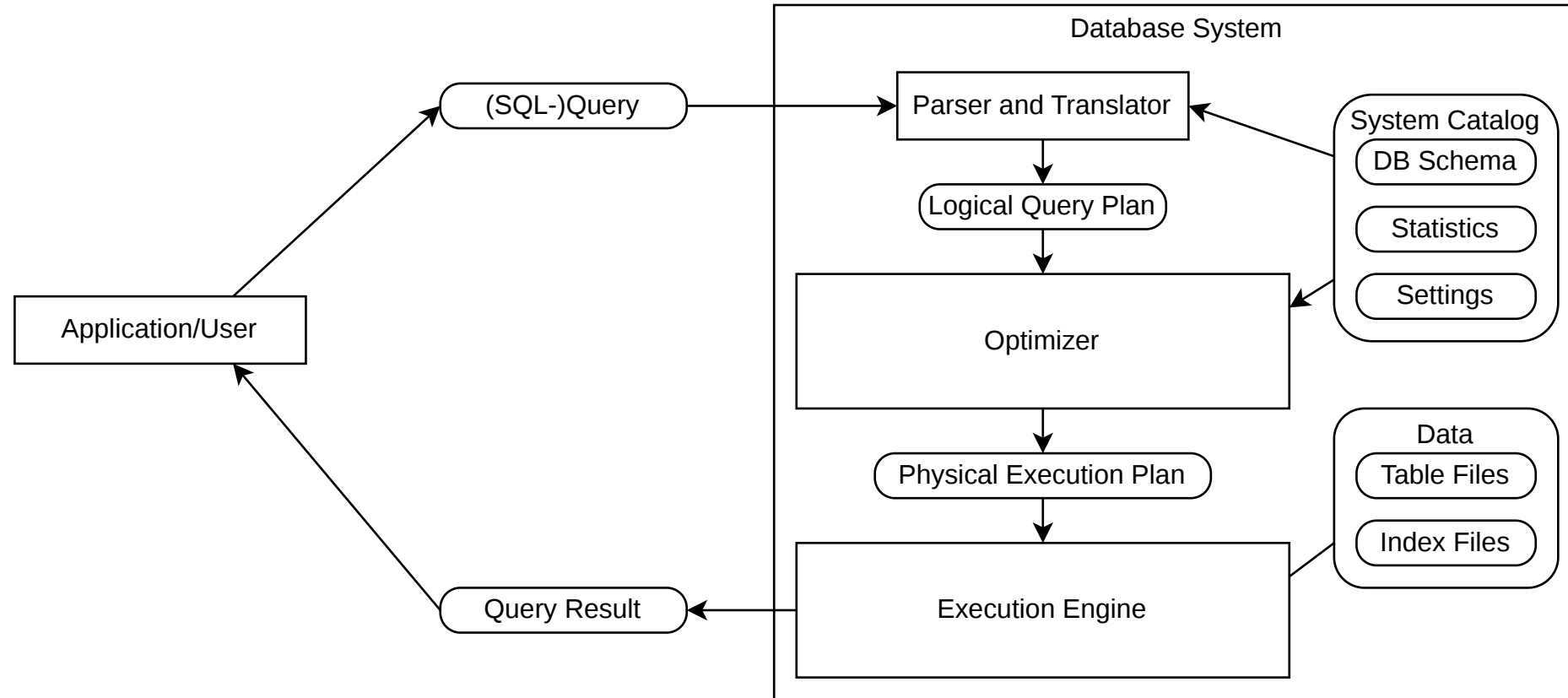
# A Collection of Demonstrations with PostgreSQL for Teaching and Learning Database System Internals

Stefan Halfpap, Daniel Hristov, Volker Markl

# Database Systems



# Database System Internals



# Learning Database System Internals is Essential



Database systems are omnipresent

Database systems include timeless patterns for performant system programming

→ If you can write code for a database system,  
you can write code for almost any software system

ADVANCED DATABASE SYSTEMS

Course Overview & Logistics

00 Andy Pavlo CMU 15-721 Spring 2024 Carnegie Mellon University

WHY YOU SHOULD TAKE THIS COURSE

DBMS developers are in demand and there are many challenging unsolved problems in data management and processing.

If you are good enough to write code for a DBMS, then you can write code on almost anything else.

And people will pay you lots of money to do it...

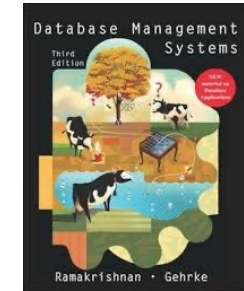
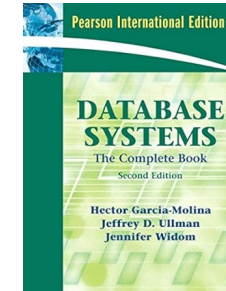
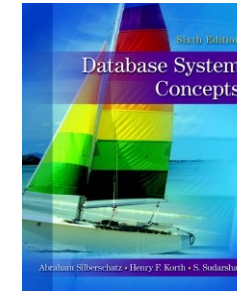
©CMU DB

The slide features a background image of a large orange moon over a dark landscape with a body of water reflecting the moon and a dark cylindrical structure. The text is white and black on a dark background.

# How to Learn Database Systems Internals?

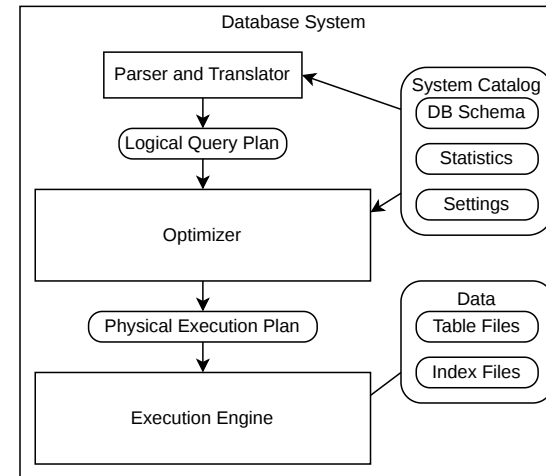


Database system lectures and books cover concepts



Main topics:

- Storage, indexing, query optimization, query execution



Learn theoretical concepts

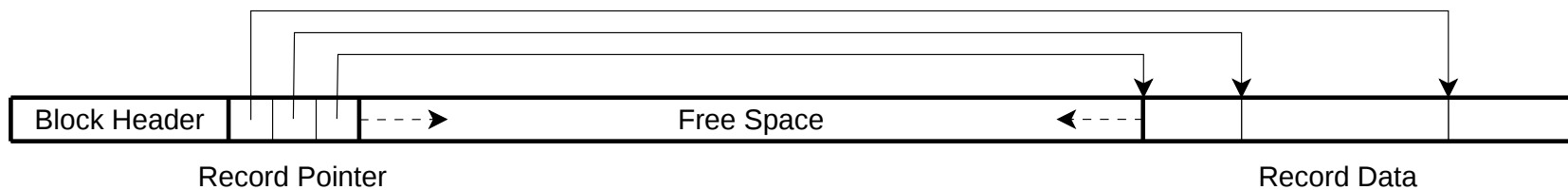


Learn practical system design and implementation

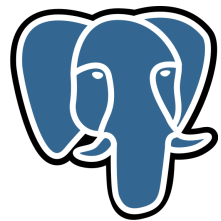
# How Database Systems Store Tables



- Collection of files, split into fixed-size pages
- Slotted page structure

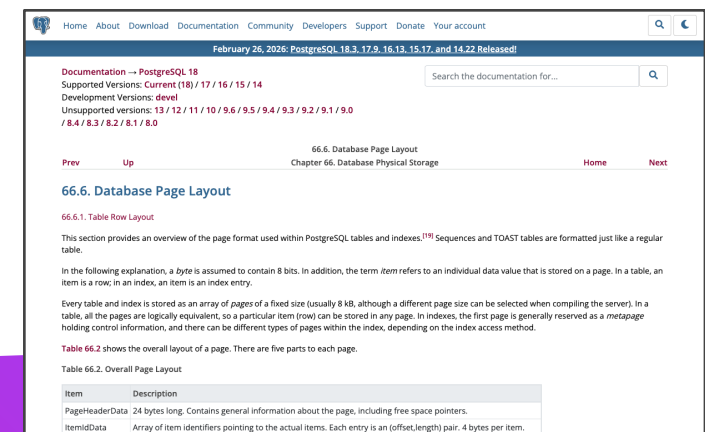
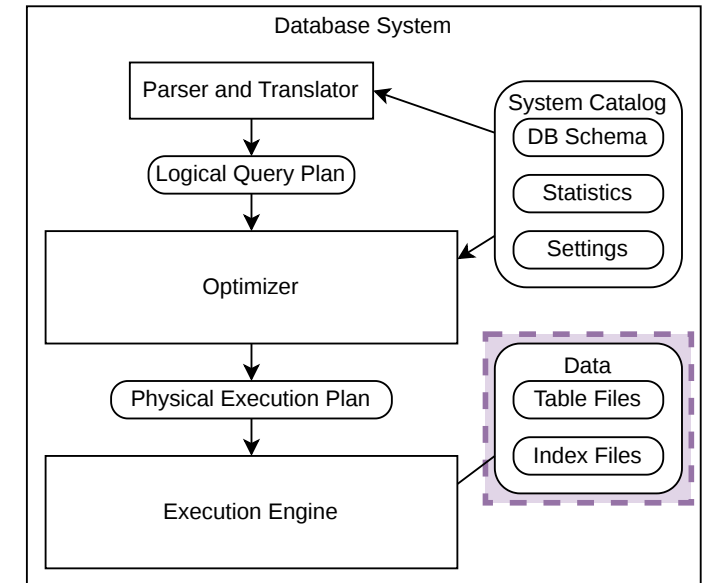


- Concrete file organization and format are system-specific



## Demonstrate PostgreSQL's implementation

<https://www.postgresql.org/docs/current/storage-page-layout.html>



# Demonstrations



[https://github.com/klauck/demo\\_dbs\\_internals](https://github.com/klauck/demo_dbs_internals)



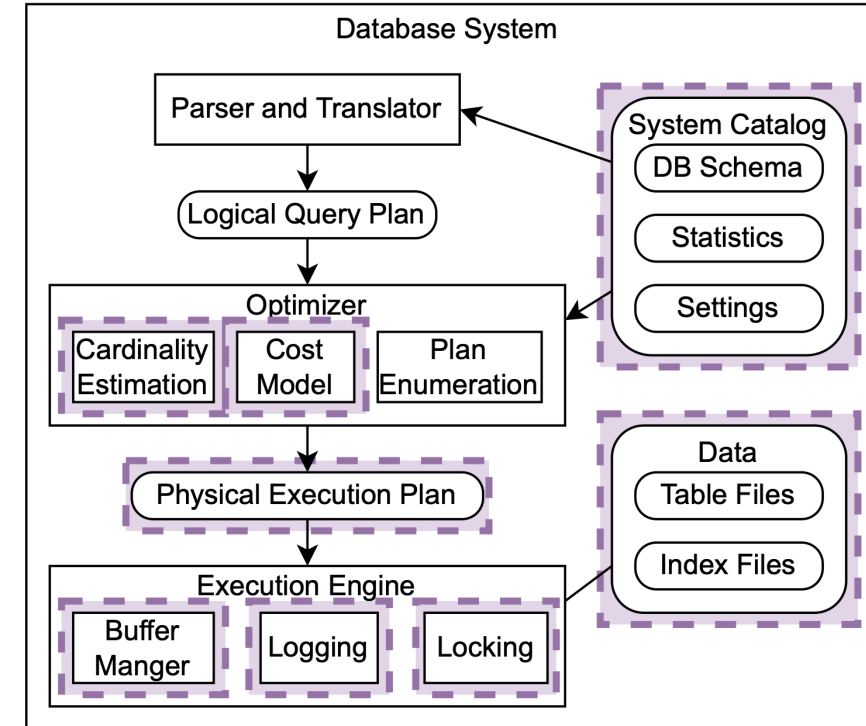
Broad range of fundamental topics

Demonstrations per course topic

Step-by-step instructions



Course Topic	Suitable Demonstrations
Database Representation	File organization (Slotted) Page structure Tuple representation
Indexing	Index utilization Index representation on disk
Query Optimization	Statistics Cardinality estimation Query plan
Query Execution	Query execution plan Buffer usage
Concurrency Control	Running transactions MVCC columns Row locking
Recovery	Running transactions Write-ahead log

```
Database Representation on Disk
Settings
The view pg_settings provides access to PostgreSQL's configuration settings, including parameter names (name), current values (setting), and units (units). We can use the SHOW and SET commands to display and change individual (e.g., SHOW name) or all (e.g., SHOW ALL) configuration parameters.
File organization
Show the directory in which PostgreSQL stores its configuration and data (e.g., tables, indexes) files.
SELECT setting FROM pg_settings WHERE name = 'data_directory';
-----
setting
/opt/homebrew/var/postgres
Within this directory, there is a subdirectory for every database, which groups a set of tables. Each database has an assigned OID (Object Identifier), which can be queried:
SELECT oid FROM pg_database WHERE datname = 'demo_db_internals';
-----
oid
17843524
```



# Running the Demonstrations



- Existing PostgreSQL installations
-  **docker** -based setup
- 

### PGLite Playground REPL

A REPL that you can use to try out PGLite with a database persisted in your browser's IndexedDB.

**Tip:** The `psql \d[. .]` commands are available, and there is autocomplete based on your schema.

#### Enabled Extensions

- `pgvector` Open-source vector ...
- `amcheck` Verify the logical con...
- `auto_explain` Log execution pla...
- `bloom` Index access method b...
- `btree_gin` GIN operator classes...
- `btree_gist` GIST operator class...
- `citext` Case-insensitive charac...
- `cube` Multidimensional cubes ...
- `earthdistance` Calculate great ...
- `fuzzystrmatch` Determine simil...
- `hstore` Key/value pairs data ty...
- `intarray` Operators for manipul...
- `dict_xsyn` Example synonym fu...
- `pageinspect` Low-level inspect...
- `dict_int` Example full-text sear...
- `unaccent` A text search diction...
- `pg_surgery` Perform low-level ...
- `pgtap` pgTAP
- `pg_uuid7` Use the new v7 UUI...
- `pg_walinspect` Low-level WAL i...
- `pg_visibility` Visibility map infor...
- `pg_freespacemap` Examine the...
- `pg_buffercache` Inspect Postgr...
- `file_fdw` Access data files in the

```
> CREATE EXTENSION IF NOT EXISTS pageinspect;
DROP TABLE IF EXISTS student;
CREATE TABLE student (
  student_id INTEGER PRIMARY KEY,
  name VARCHAR(50),
  phone_number VARCHAR(20)
);
INSERT INTO student VALUES(1, 'Sarah', '0815');
SELECT ctid, student_id, name, phone_number FROM student;
```

< null

< null

< null

< null

ctid	student_id	name	phone_number
(0,1)	1	Sarah	0815

1 rows

```
> UPDATE student SET phone_number = '1704' WHERE student_id = 1;
```

< null

```
> UPDATE student SET phone_number = '42' WHERE student_id = 1;
```

< null

```
> SELECT * FROM student;
```

student_id	name	phone_number
1	Sarah	42

1 rows

```
> SELECT ctid, student_id, name, phone_number FROM student;
```

ctid	student_id	name	phone_number
(0,3)	1	Sarah	42

1 rows

```
> SELECT lp, t_xmin, t_xmax, t_ctid, t_data
FROM heap_page_items(get_raw_page('student', 0));
```

lp	t_xmin	t_xmax	t_ctid	t_data
1	741	742	(0,2)	{\0":1,"1":0,"2":0,"3":0,"4":13,"5":83,"6":97,"7":114,"8":97,"9...
2	742	743	(0,3)	{\0":1,"1":0,"2":0,"3":0,"4":13,"5":83,"6":97,"7":114,"8":97,"9...
3	743	0	(0,3)	{\0":1,"1":0,"2":0,"3":0,"4":13,"5":83,"6":97,"7":114,"8":97,"9...




3 rows

1

Clear Playground Database

# Running the Demonstrations



- Existing PostgreSQL installations
-  **docker** -based setup
-  **PGlite**
-  **jupyter** notebooks



## Setup environment, connect to PostgreSQL, and load data

```
[1]: %load_ext sql
      %sql postgres://postgres:@localhost:5432/demo_db_internals
```

```
[10]: %%sql
      DROP TABLE IF EXISTS student;
      CREATE TABLE student (
        student_id SERIAL PRIMARY KEY,
        name VARCHAR(50),
        phone_number VARCHAR(20)
      );
      INSERT INTO student VALUES(1, 'Sarah', '0815');
      SELECT ctid, student_id, name, phone_number FROM student;

      * postgres://postgres:***@localhost:5432/demo_db_internals
      Done.
      Done.
      1 rows affected.
      1 rows affected.
```

```
[10]: ctid student_id name phone_number
      (0,1)          1 Sarah          0815
```

## PostgreSQL Demonstrations for Teaching Database System Internals

### Database Representation on Disk

#### Settings

The view `pg_settings` provides access to PostgreSQL's configuration settings, including parameter names ( `name` ), current values ( `setting` ), and units ( `unit` ). We can use the `SHOW` and `SET` commands to display and change individual (e.g., `SHOW name` ) or all (e.g., `SHOW ALL` ) configuration parameters.

#### File organization

Show the directory in which PostgreSQL stores its configuration and data (e.g., tables, indexes) files.

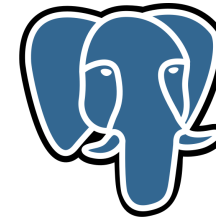
```
SELECT setting FROM pg_settings WHERE name = 'data_directory';

      setting
      -----
      /opt/homebrew/var/postgres
```

```
[2]: %%sql
      SHOW data_directory;

      * postgres://postgres:***@localhost:5432/demo_db_internals
```

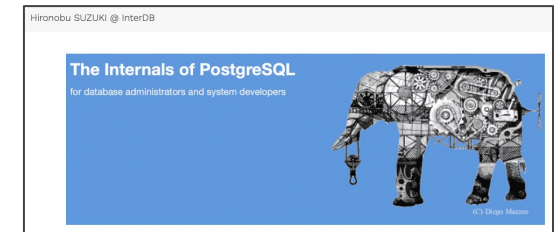
# Why PostgreSQL



- Popular, commonly used, and mature system
- Offers extensive capabilities for inspecting internals  
e.g., [system catalogs](#), [pageinspect](#), [pg\\_buffercache](#), [pgrowlocks](#), [pg\\_walinspect](#)

- Great documentation  
(including books, blog posts, ...)

<b>PostgreSQL 17.5 Documentation</b> The PostgreSQL Global Development Group Copyright © 1996–2025 The PostgreSQL Global Development Group <b>Legal Notice</b> <b>Table of Contents</b> Preface 1. What is PostgreSQL? 2. A Brief History of PostgreSQL 3. Conventions 4. Further Information 5. Bug Reporting Guidelines I. Tutorial 1. Getting Started 2. The SQL Language 3. Advanced Features II. The SQL Language 4. SQL Syntax 5. Data Definition
--

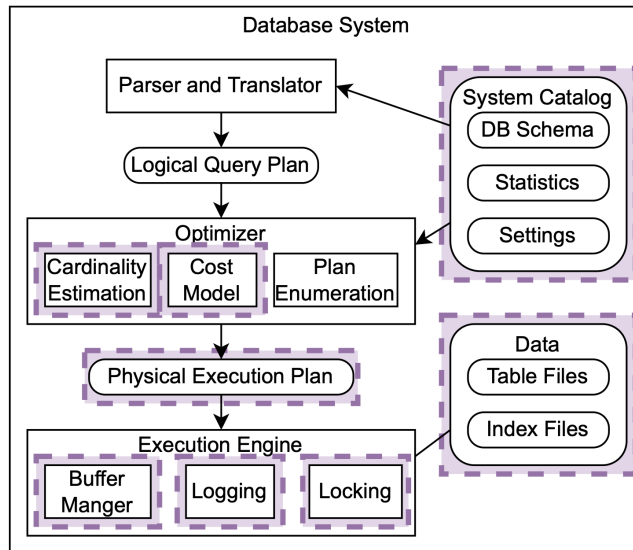


- Open-source

# A Collection of Demonstrations with PostgreSQL for Teaching and Learning Database System Internals



## Broad range of concepts



## Specific examples

```
README
```

### Database Representation on Disk

Settings

The view `pg_settings` provides access to PostgreSQL's configuration settings, including parameter names ( `name` ), current values ( `setting` ), and units ( `unit` ). We can use the `SHOW` and `SET` commands to display and change individual (e.g., `SHOW name`) or all (e.g., `SHOW ALL`) configuration parameters.

File organization

Show the directory in which PostgreSQL stores its configuration and data (e.g., tables, indexes) files.

```
SELECT setting FROM pg_settings WHERE name = 'data_directory';
```

setting
/opt/homebrew/var/postgres

Within this directory, there is a subdirectory for every database, which groups a set of tables. Each database has an assigned OID (Object Identifier), which can be queried:

```
SELECT oid FROM pg_database WHERE datname = 'demo_db_internals';
```

oid
17843524

## Flexible demonstration setups

- Existing PostgreSQL installations
- docker** -based setups
- PGLite**
- jupyter notebooks**



[https://github.com/klauck/demo\\_dbs\\_internals](https://github.com/klauck/demo_dbs_internals)

Stefan Halfpap  
halfpap@tu-berlin.de

# Image Attribution



## Book Covers:

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database System Concepts* (6th ed.). McGraw-Hill Education.

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database Systems: The Complete Book* (2nd ed.). Upper Saddle River, NJ: Pearson Education.

Ramakrishnan, R., & Gehrke, J. (2002). *Database Management Systems* (3rd ed.). McGraw-Hill Education.

Rogov, E. (2023). *PostgreSQL 14 Internals*. Moscow: Postgres Professional. ISBN 978-5-6045970-4-0.

## Screenshots:

Andy Pavlo: Advanced Database Systems

Retrieved from <https://15721.courses.cs.cmu.edu/spring2024/slides/00-introduction.pdf> on March 24, 2026.

The PostgreSQL Global Development Group: PostgreSQL 17 Documentation

Retrieved from <https://www.postgresql.org/docs/17/index.html> on May 9, 2025.

Hironobu Suzuki: The Internals of PostgreSQL

Retrieved from <https://www.interdb.jp/pg/> on May 9, 2025.

All images used under fair use for educational purposes.