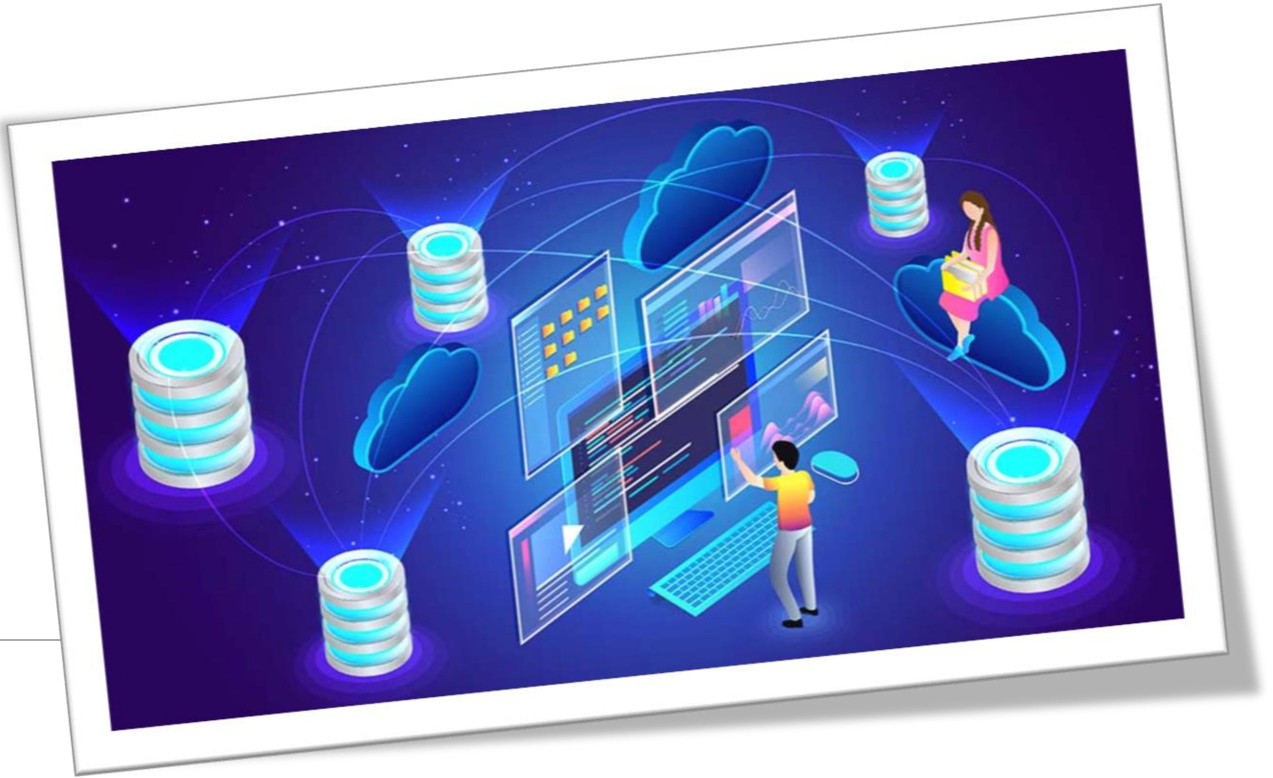
Advanced Database

COMP412

CHAPTER:01

Introduction



RDMS: RELATIONAL DATABASE MANAGEMENT SYSTEM

What are Database Management Systems

DBMS is a system for providing:

- EFFICIENT,
- CONVENIENT,
- SAFEMULTI-USER storage of and access to MASSIVE amounts of PERSISTENT data

Example: Banking System

- Data
  - Information on accounts, customers, balances, current interest rates, transaction histories, etc.
- MASSIVE
  - Many gigabytes at a minimum for big banks, more if keep history of all transactions, even more if keep images of checks -> Far too big to fit in main memory
- PERSISTENT
  - Data outlives programs that operate on it

Example: Banking System

- SAFE:
  - from system failures
  - from bad users
- CONVENIENT:
  - simple commands to debit account, get balance, write statement, transfer funds, etc.
  - also unpredicted queries should be easy
- EFFICIENT:
  - don't search all files in order to get balance of one account, get all accounts with low balances, get large transactions, etc.
  - massive data! -> DBMS's carefully tuned for performance

Multi-user Access

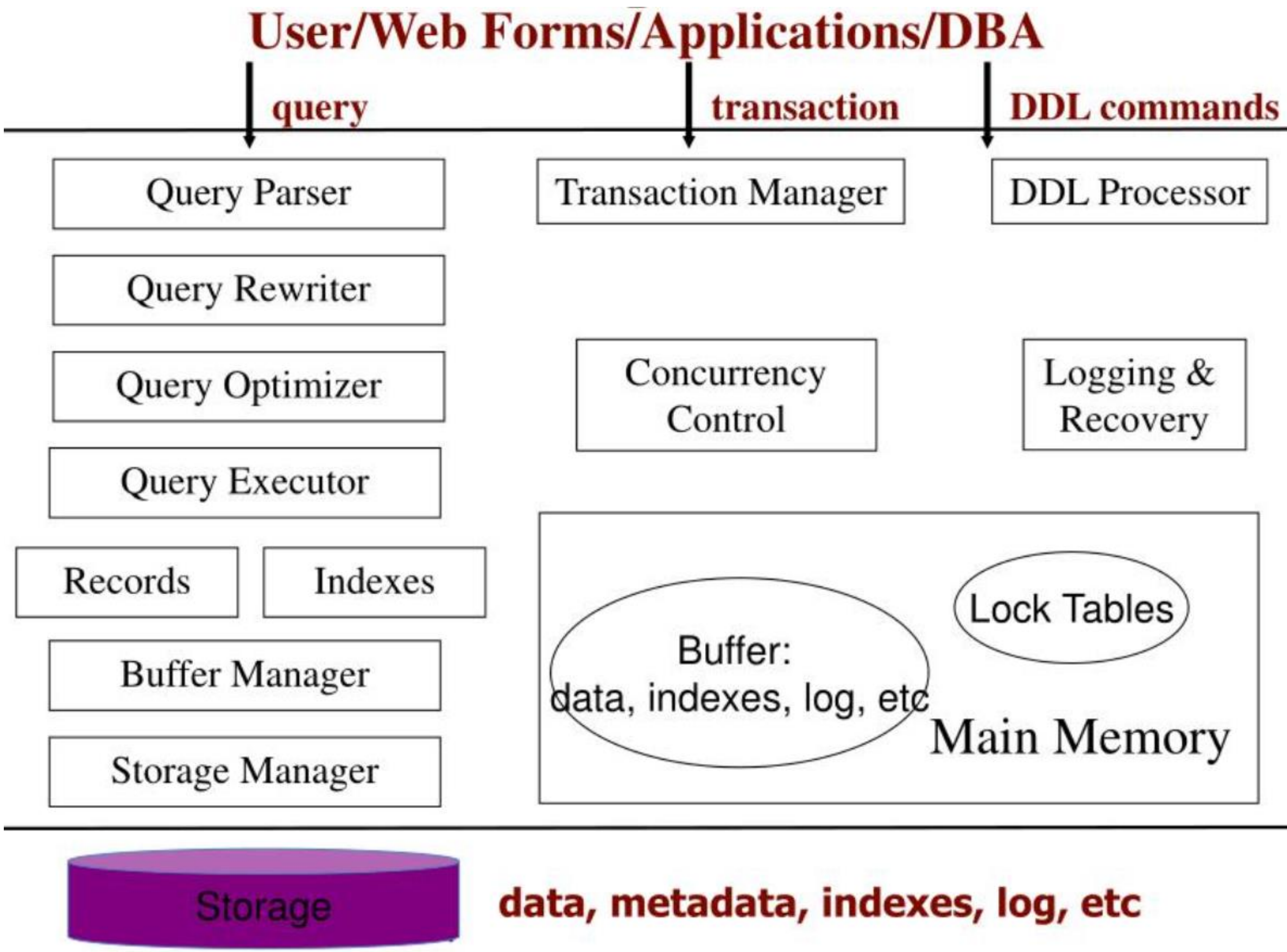
- Many people/programs accessing same database, or even same data, simultaneously -> Need careful controls
  - Alex @ ATM1: withdraw $100 from account #007
    - get balance from database;
    - if balance >= 100 then balance := balance - 100;
    - dispense cash;
    - put new balance into database;
  - Bob @ ATM2: withdraw $50 from account #007
    - get balance from database;
    - if balance >= 50 then balance := balance - 50;
    - dispense cash;
    - put new balance into database;
  - Initial balance = 120. Final balance = ??

Why File Systems Won't Work

- Storing data: file system is limited
  - size limit by disk or address space
  - when system crashes we may loose data
  - Password/file-based authorization insufficient
- Query/update:
  - need to write a new C++/Java program for every new query
  - need to worry about performance
- Concurrency: limited protection
  - need to worry about interfering with other users
  - need to offer different views to different users (e.g. registrar, students, professors)
- Schema change:
  - entails changing file formats
  - need to rewrite virtually all applications

That's why the notion of DBMS was motivated!

DMS Architecture

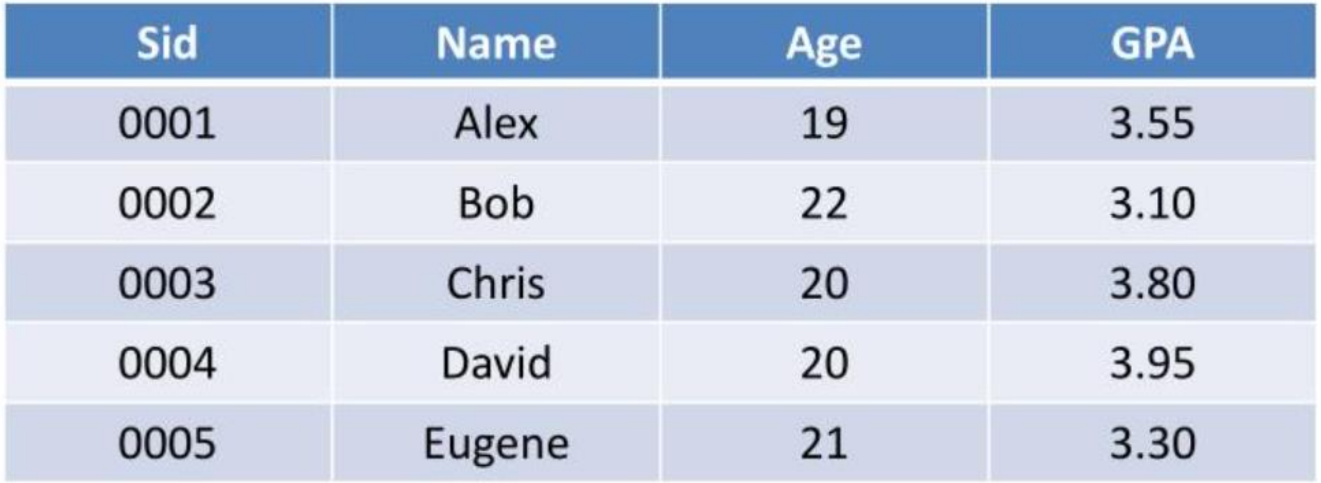


Data Structuring: Model, Schema, Data

- Data model
  - conceptual structuring of data stored in database
  - ex: data is set of records, each with student-ID, name, address, courses, photo
  - ex: data is graph where nodes represent cities, edges represent airline routes
- Schema versus data
  - schema: describes how data is to be structured, defined at set-up time, rarely changes (also called "metadata")
  - data: actual "instance" of database, changes rapidly
  - vs. types and variables in programming languages

Schema vs. Data

- Schema: name, name of each field, the type of each field
  - Students (Sid:string, Name:string, Age: integer, GPA: real)
  - A template for describing a student
- Data: an example instance of the relation



Sid	Name	Age	GPA
0001	Alex	19	3.55
0002	Bob	22	3.10
0003	Chris	20	3.80
0004	David	20	3.95
0005	Eugene	21	3.30

Data Structuring: Model, Schema, Data

- Data definition language (DDL)
  - commands for setting up schema of database

- Data Manipulation Language (DML)
  - Commands to manipulate data in database:
    - RETRIEVE, INSERT, DELETE, MODIFY
  - Also called "query language"

PostgreSQL

THE WORLD'S MOST ADVANCED OPEN SOURCE RELATIONAL DATABASE

PostgreSQL

- PostgreSQL is the world’s most advanced open source database and the fourth most popular database. In development for more than 30 years, PostgreSQL is managed by a well-organized and highly principled and experienced open source community.

- PostgreSQL databases provide enterprise-class database solutions and are used by a wide variety of enterprises across many industries, including financial services, information technology, government and media and communications.

Administration Tools

Administration Tools

- Four tools widely used with PostgreSQL are:
  - psql,
  - pgAdmin,
  - phpPgAdmin,
  - Adminer.

psql

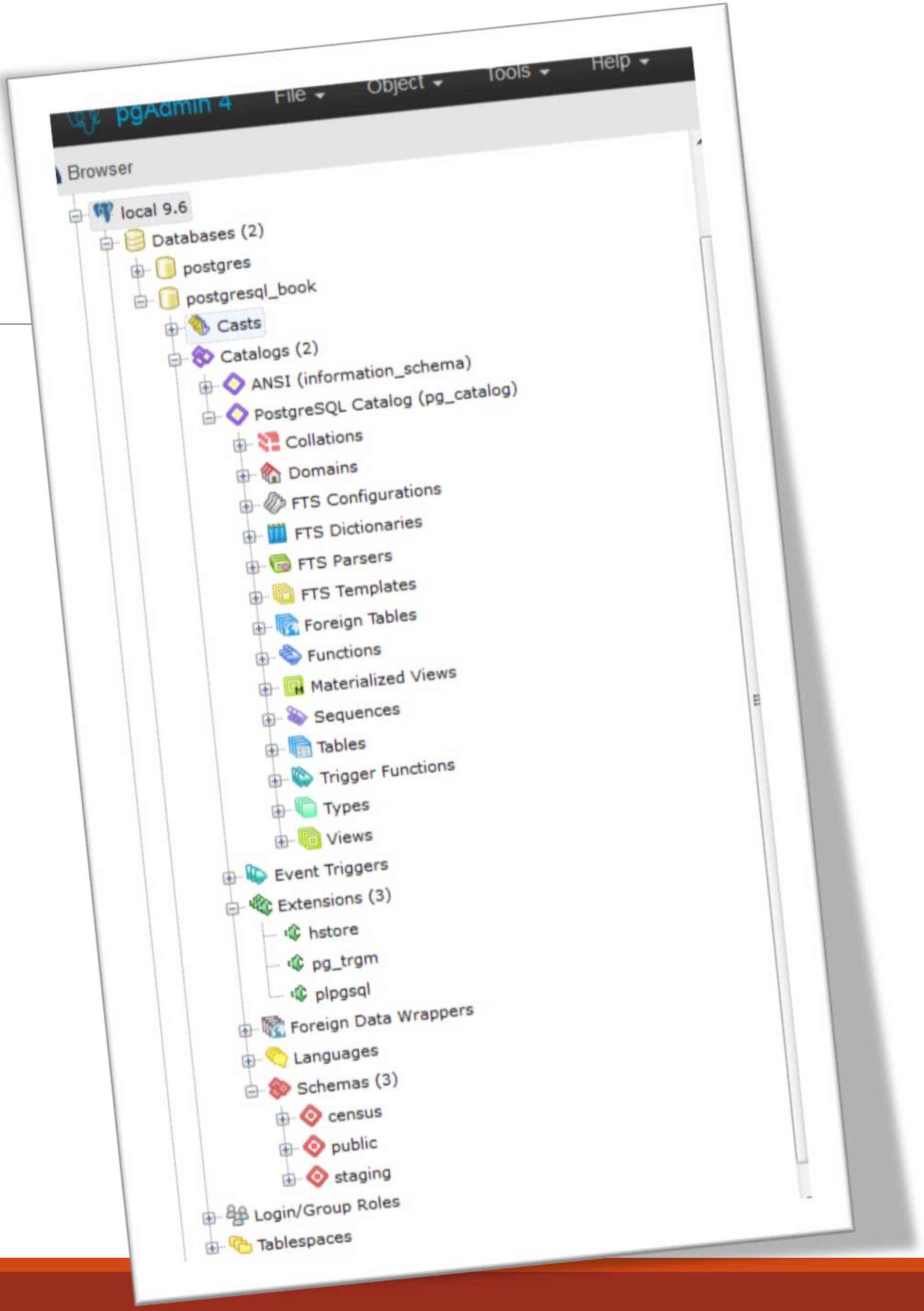
- psql is a command-line interface for running queries and is included in all distributions of PostgreSQL

- Is the tool of choice for many expert users, for people working in consoles without a GUI, or for running common tasks in shell scripts.

- psql has some unusual features, such as an import and export command for delimited files (CSV or tab), and a minimalistic report writer that can generate HTML output

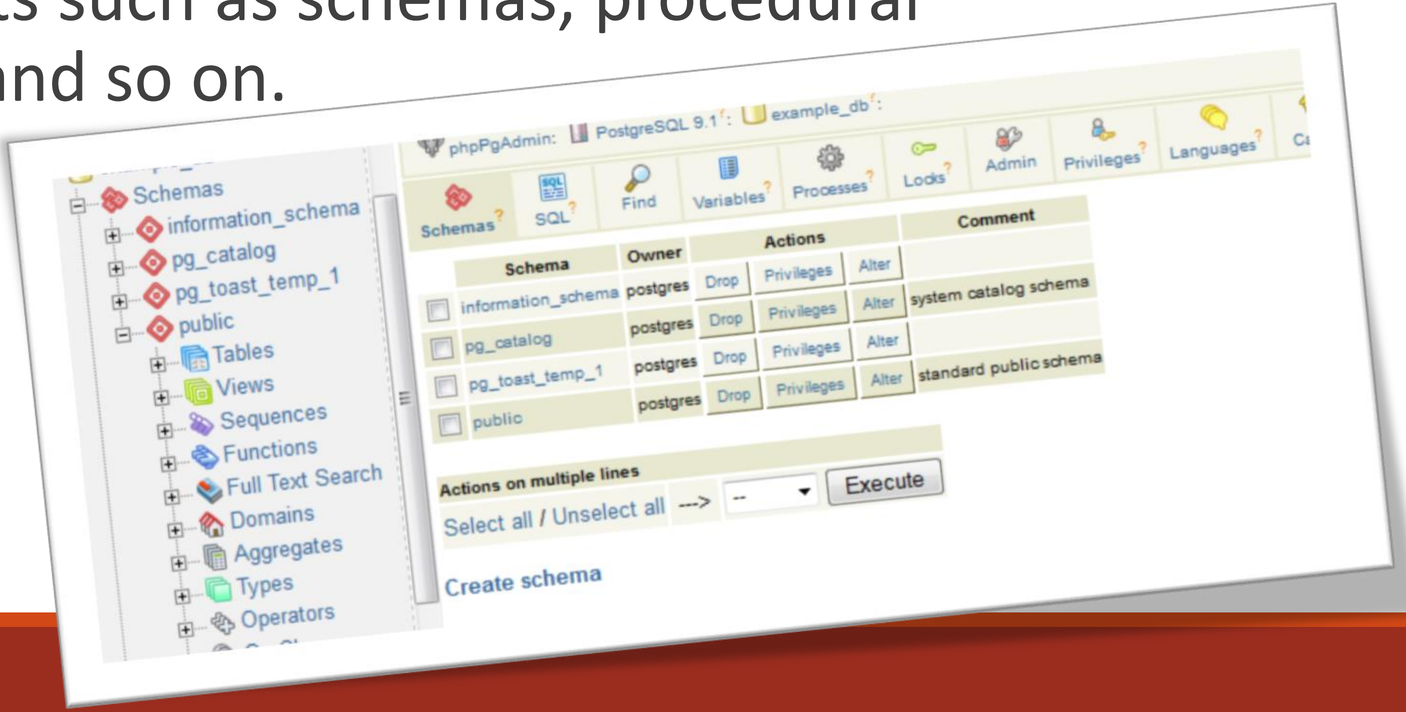
pgAdmin

- pgAdmin is a popular, free GUI tool for PostgreSQL.
- Download it separately from PostgreSQL if it isn't already packaged with your installer
- pgAdmin recently entered its fourth release, dubbed pgAdmin4.
  - pgAdmin4 is a complete rewrite of pgAdmin3 that supports a desktop as well as a web server application version utilizing Python.



phpPgAdmin

- phpPgAdmin is a free, web-based administration tool patterned after the popular phpMyAdmin.
- phpPgAdmin differs from phpMyAdmin by including ways to manage PostgreSQL objects such as schemas, procedural languages, casts, operators, and so on.

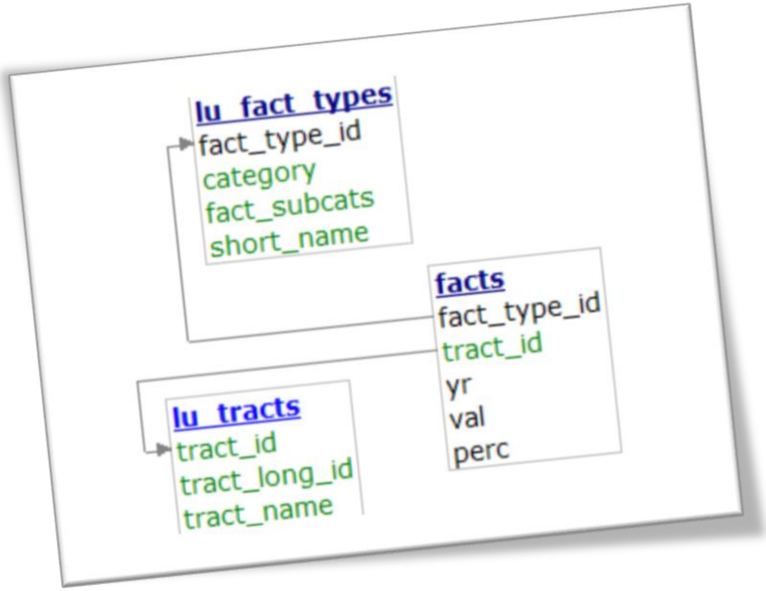


	Schema	Owner	Actions			Comment
□	information_schema	postgres	Drop	Privileges	Alter	
□	pg_catalog	postgres	Drop	Privileges	Alter	system catalog schema
□	pg_toast_temp_1	postgres	Drop	Privileges	Alter	
□	public	postgres	Drop	Privileges	Alter	standard public schema

Actions on multiple lines
Select all / Unselect all ---> -- Execute
Create schema

Adminer

- If you manage other databases besides PostgreSQL and are looking for a unified tool, Adminer might fit the bill.
- Adminer is a lightweight, open source PHP application with options for PostgreSQL, MySQL, SQLite, SQL Server, and Oracle, all delivered through a single interface.
- One unique feature of Adminer we’re impressed with is the relational diagramme that can produce a schematic layout of your database schema.



PostgreSQL Database Objects

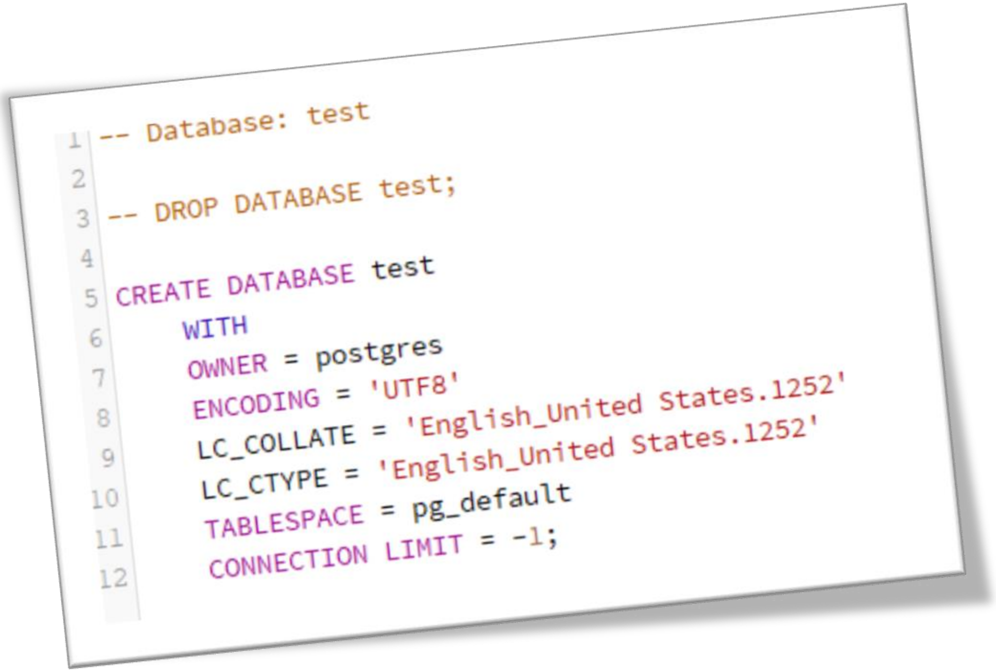
PostgreSQL Database Objects

- PostgreSQL has more database objects than most other relational database products

- We limit our quick overview to the most important objects that you should be familiar with.

Databases

- Each PostgreSQL service houses many individual databases.



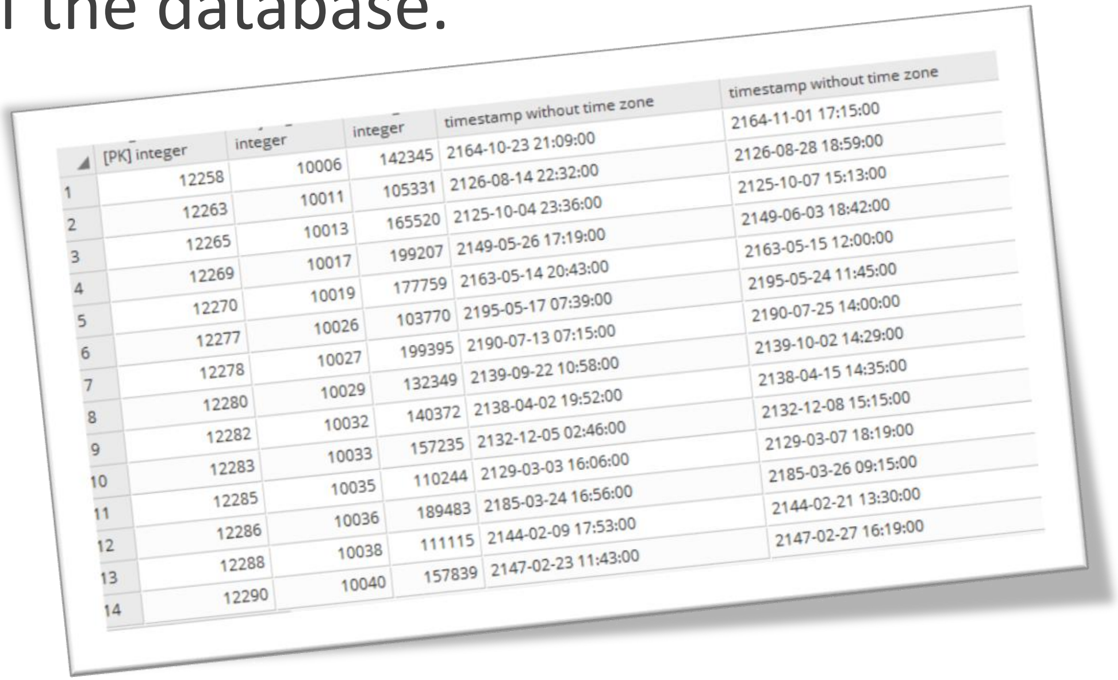
```sql
-- Database: test

-- DROP DATABASE test;

CREATE DATABASE test
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'English_United States.1252'
    LC_CTYPE = 'English_United States.1252'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1;
```

Schemas

- Schemas are part of the ANSI SQL standard.
- They are the immediate next level of organization within each database.
  - If you think of the database as a country, schemas would be the individual states.
- Most database objects first belong to a schema, which belongs to a database.
- When you create a new database, PostgreSQL automatically creates a schema named public to store objects that you create.
- If you have few tables, using public would be fine.
  - But if you have thousands of tables, you should organize them into different schemas.

Tables

- Tables are the workhorses of any database.
- In PostgreSQL, tables are first citizens of their respective schemas, which in turn are citizens of the database.



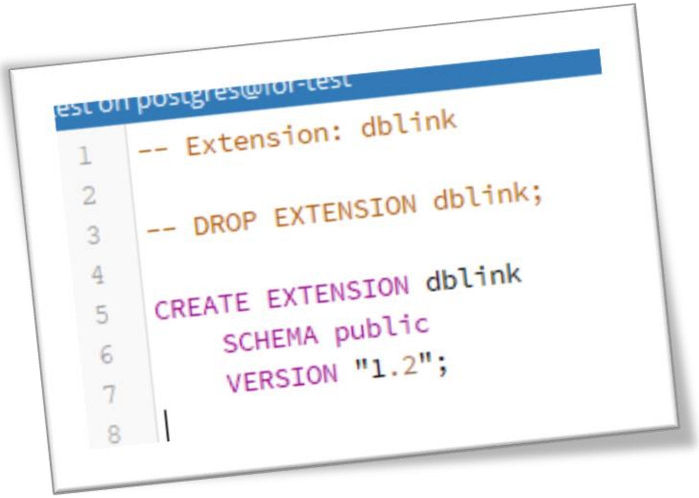
	[PK] integer	integer	integer	timestamp without time zone	timestamp without time zone
1	12258	10006	142345	2164-10-23 21:09:00	2164-11-01 17:15:00
2	12263	10011	105331	2126-08-14 22:32:00	2126-08-28 18:59:00
3	12265	10013	165520	2125-10-04 23:36:00	2125-10-07 15:13:00
4	12269	10017	199207	2149-05-26 17:19:00	2149-06-03 18:42:00
5	12270	10019	177759	2163-05-14 20:43:00	2163-05-15 12:00:00
6	12277	10026	103770	2195-05-17 07:39:00	2195-05-24 11:45:00
7	12278	10027	199395	2190-07-13 07:15:00	2190-07-25 14:00:00
8	12280	10029	132349	2139-09-22 10:58:00	2139-10-02 14:29:00
9	12282	10032	140372	2138-04-02 19:52:00	2138-04-15 14:35:00
10	12283	10033	157235	2132-12-05 02:46:00	2132-12-08 15:15:00
11	12285	10035	110244	2129-03-03 16:06:00	2129-03-07 18:19:00
12	12286	10036	189483	2185-03-24 16:56:00	2185-03-26 09:15:00
13	12288	10038	111115	2144-02-09 17:53:00	2144-02-21 13:30:00
14	12290	10040	157839	2147-02-23 11:43:00	2147-02-27 16:19:00

Views

- Relational database products offer views as a level of abstraction from tables.
- In a view, you can query multiple tables and present additional derived columns based on complex calculations.
- Views are generally read-only, but PostgreSQL allows you to update the underlying data by updating the view, provided that the view draws from a single table.
- Version 9.3 introduced materialized views, which cache data to speed up commonly used queries at the sacrifice of having the most up-to-date data.

Extension

- Extensions allow developers to package functions, data types, casts, custom index types, tables, attribute variables, etc., for installation or removal as a unit.



```
-- Extension: dblink

-- DROP EXTENSION dblink;

CREATE EXTENSION dblink
    SCHEMA public
    VERSION "1.2";

```

Functions

- You can program your own custom functions to handle data manipulation, perform complex calculations, or wrap similar functionality.
  - Create functions using PLs.
- PostgreSQL comes stocked with thousands of functions, which you can view in the postgres database that is part of every install.
- PostgreSQL functions can return scalar values, arrays, single records, or sets of records.

Foreign tables

- Foreign tables are virtual tables linked to data outside a PostgreSQL database.
- Once you’ve configured the link, you can query them like any other tables.
- Foreign tables can link to CSV files, a PostgreSQL table on another server, a table in a different product such as:
  - SQL Server or Oracle, a NoSQL database such as Redis, or even a web service such as Twitter or Salesforce.

Triggers and trigger functions

- Triggers detect data-change events.
- When PostgreSQL fires a trigger, you have the opportunity to execute trigger functions in response.
- A trigger can run in response to particular types of statements or in response to changes to particular rows, and can fire before or after a data-change event.
- Trigger functions are often used to write complex validation routines that are beyond what can be implemented using check constraints.

Catalogs

- Catalogs are system schemas that store PostgreSQL builtin functions and metadata.
- Every database contains two catalogs:
  - pg_catalog, which holds all functions, tables, system views, casts, and types packaged with PostgreSQL;
  - information_schema, which offers views exposing metadata in a format dictated by the ANSI SQL standard.

Types

- Data type such as: integers, characters, arrays, blobs, etc.
- PostgreSQL has composite types, which are made up of other types. Think of complex numbers, polar coordinates, or vectors as examples.

Full text search

- Full text search (FTS) is a natural language–based search.
- This kind of search has some “intelligence” built in.

- Unlike regular expression search, FTS can match based on the semantics of an expression, not just its syntactical makeup.
  - For example, if you’re searching for the word running in a long piece of text, you may end up with run, running, ran, runner, jog, sprint, dash, and so on.

Sequences

- A sequence controls the auto incrementation of a serial data type.
- You can easily change the initial value, step, and next available value.
- More than one table can share the same sequence object.
  - This allows you to create a unique key value that can span tables.

Versions of PostgreSQL

Versions of PostgreSQL

- Every September a new PostgreSQL is released.
- With each new release comes greater stability, heightened security, better performance and grade features.
- The upgrade process itself gets easier with each new version.

Database Drivers

Database Drivers

- PostgreSQL works with free drivers for many programming languages and tools:
  - For **PHP**: most PHP distributions include at least one PostgreSQL driver: the old pgsql driver or the newer pdo_pgsql
  - For **Java** developers, the JDBC driver keeps up with latest PostgreSQL versions.
  - For .**NET** (both Microsoft or Mono), you can use the Npgsql driver. Both the source code and the binary are available for .NET Framework, Microsoft Entity Framework, and Mono.NET.

Database Drivers

- If you need to connect from Microsoft **Access**, **Excel**, or any other products that support Open Database Connectivity (ODBC), download drivers from the PostgreSQL ODBC drivers site.
- **Python** has support for PostgreSQL via many database drivers. At the moment, psycopg2 is the most popular.
- **Node.js** is a JavaScript framework for running scalable network programs. There are two PostgreSQL drivers currently: Node Postgres with optional native libpq bindings and pure JS (no compilation required) and Node-DBI.