

CSCI235 Database Systems

Distributed Relational Database Systems

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Distributed database ? What is it ?

Distributed database system (DDBS) is a collection of multiple logically related databases distributed over a computer network

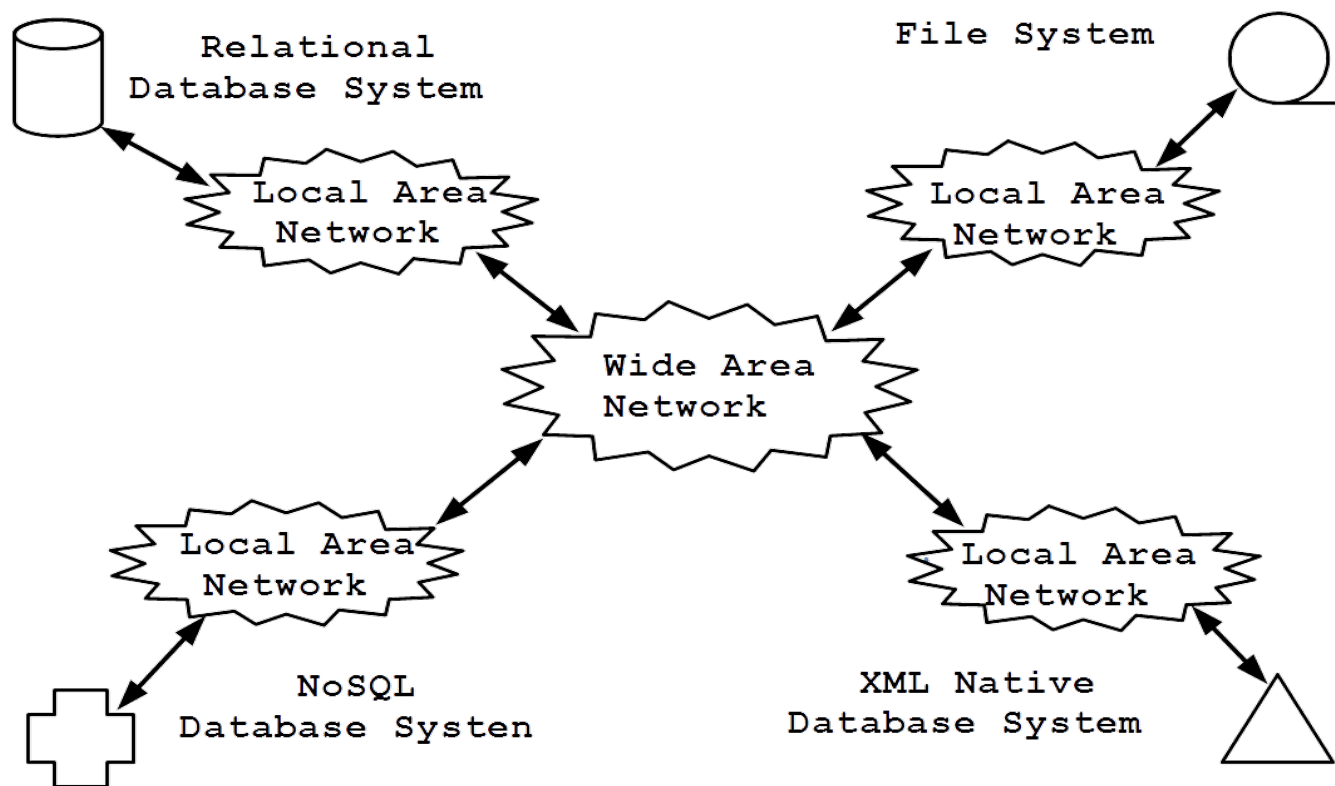
Distributed database management system (DDBMS) is a collection of database systems together with software providing a required set of operations on data and management features

Homogeneous DDBS is a collection of identical database systems distributed over a computer network, e.g. a collection of Oracle database systems

Heterogeneous DDBS is a collection of different database systems distributed over a computer network, e.g. a collection of Oracle + MySQL + DB/2 + MongoDB + XML native database systems + Excel spreadsheets + ... , systems

Distributed database ? What is it ?

A sample organization of **heterogeneous distributed** database system



[In HTML view press 'p' to see the lecture notes](#)

[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

4/37

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

[Autonomy](#)

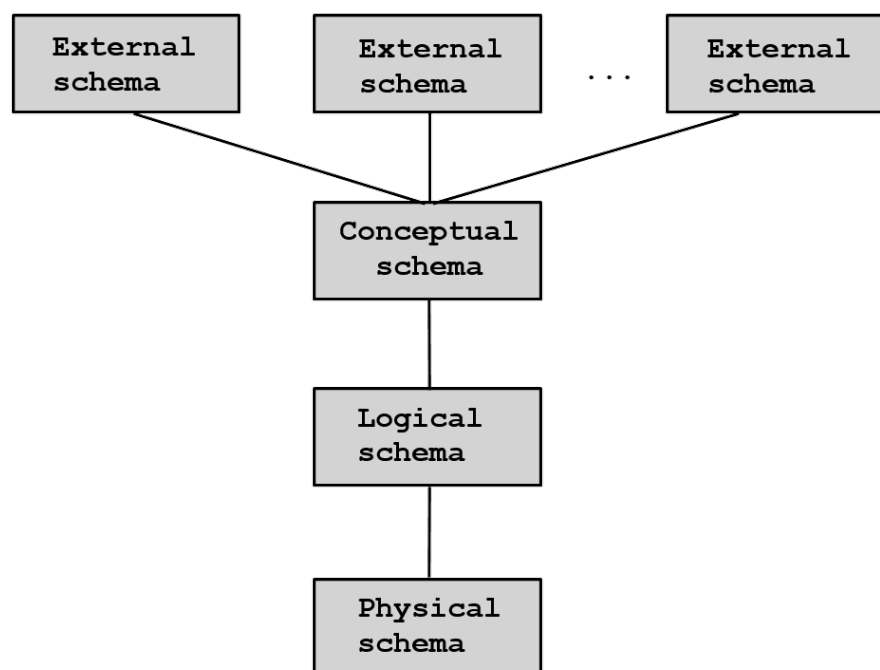
[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Centralized database schema model

In a **centralized database schema model** the users are provided their **personal external schemas** (views) of data

The **external schemas** are integrated into a **single conceptual schema** later on transformed into a **logical schema** and implemented as a **physical schema**



Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

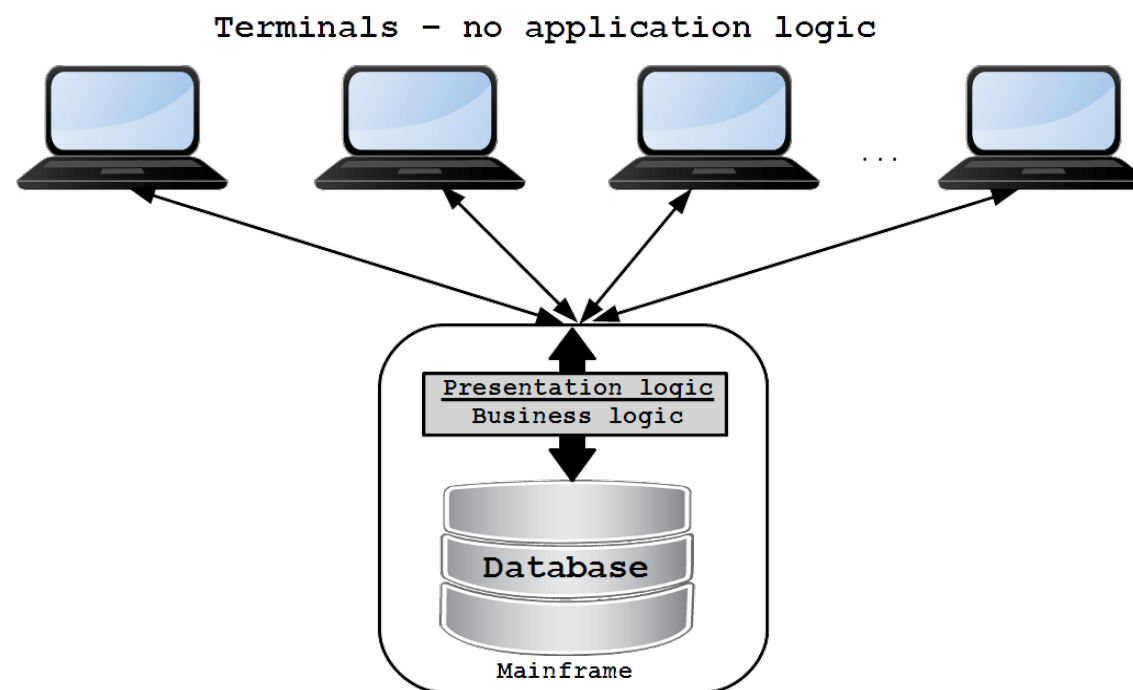
[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Mainframe application architecture

In **mainframe application architecture** "dumb" terminals are connected to a **single database server**



In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

8/37

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

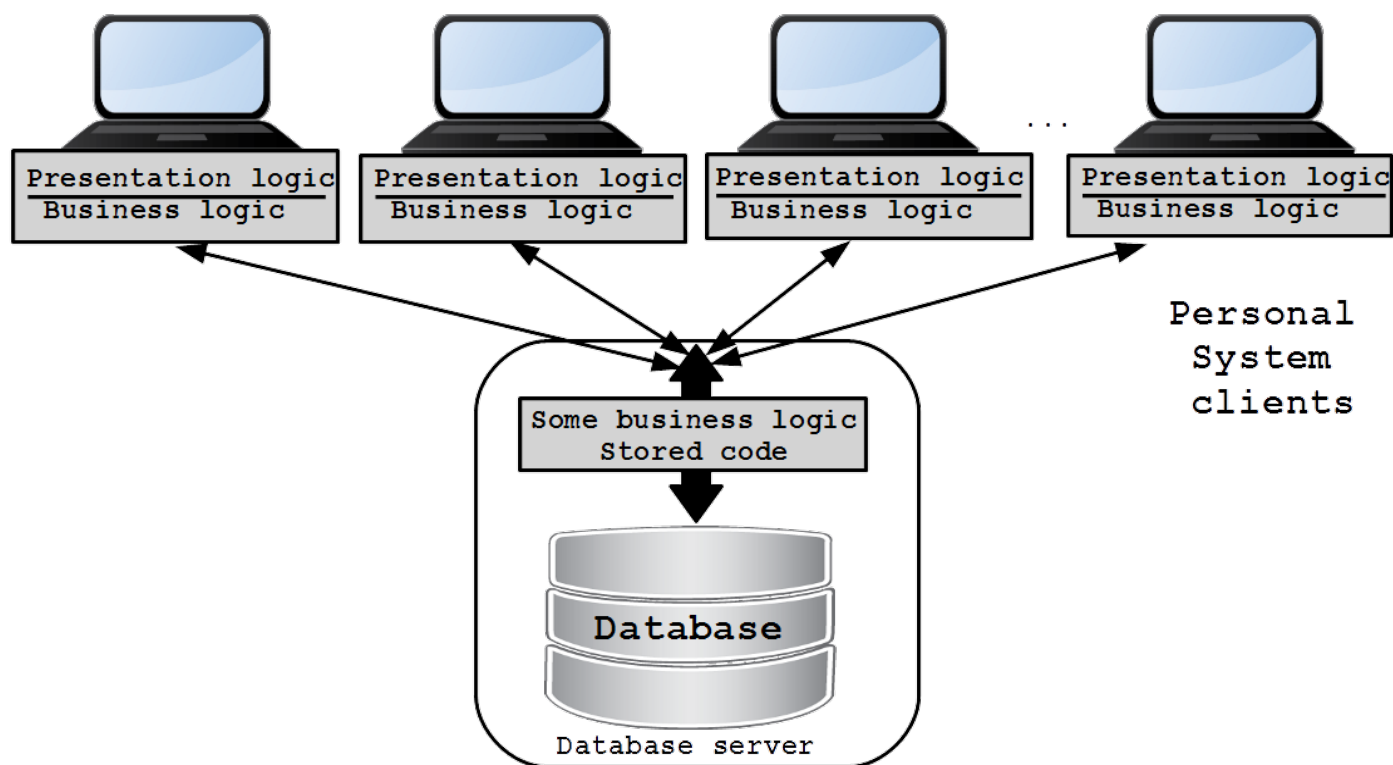
[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Client-server architecture

In **client-server architecture** personal systems communicate with a **single database server**



Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

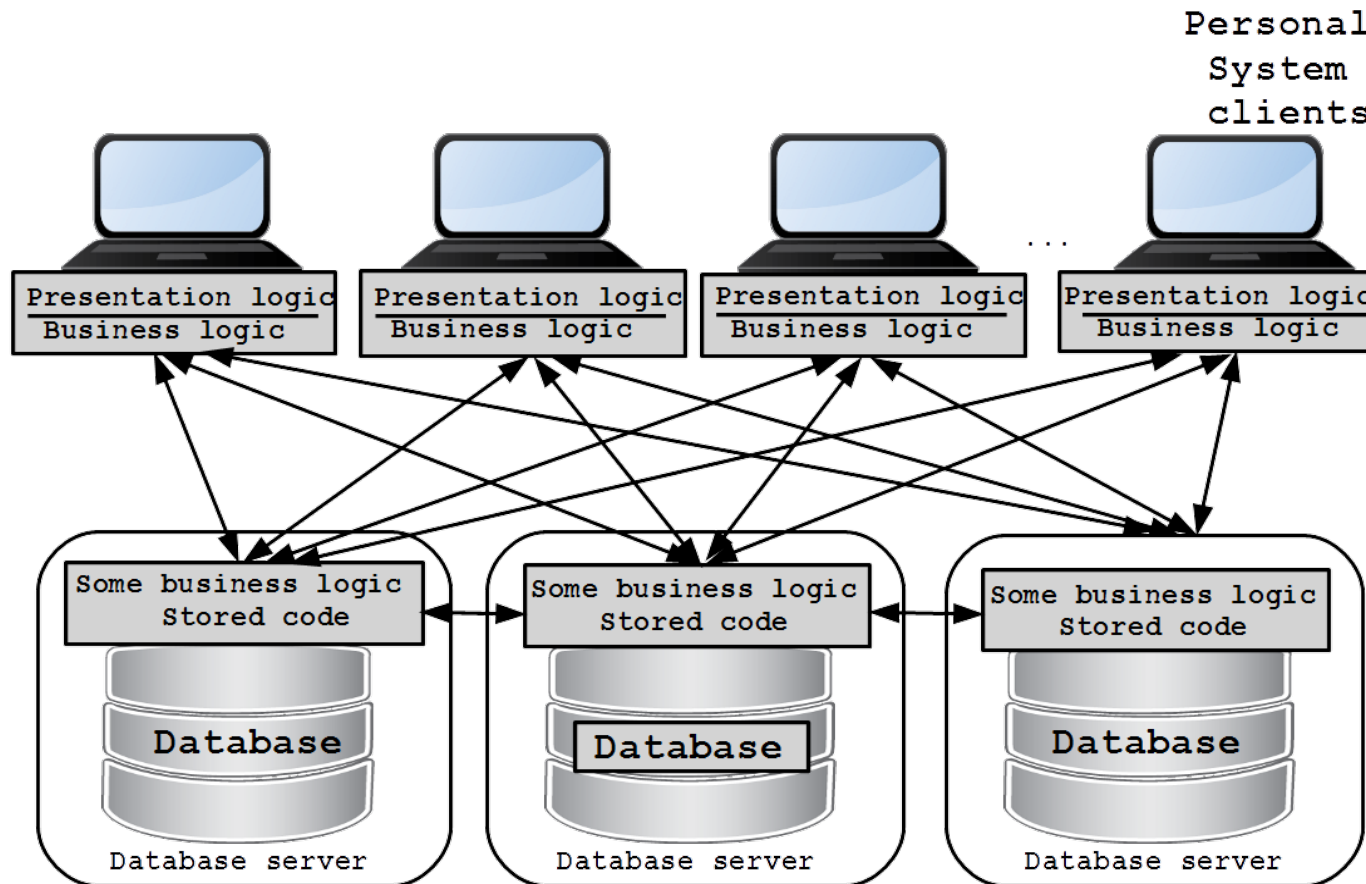
[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Distributed client-server architecture

In **distributed client-server architecture** personal systems communicate with **many database servers**



[In HTML view press 'p' to see the lecture notes](#)
[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

12/37

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

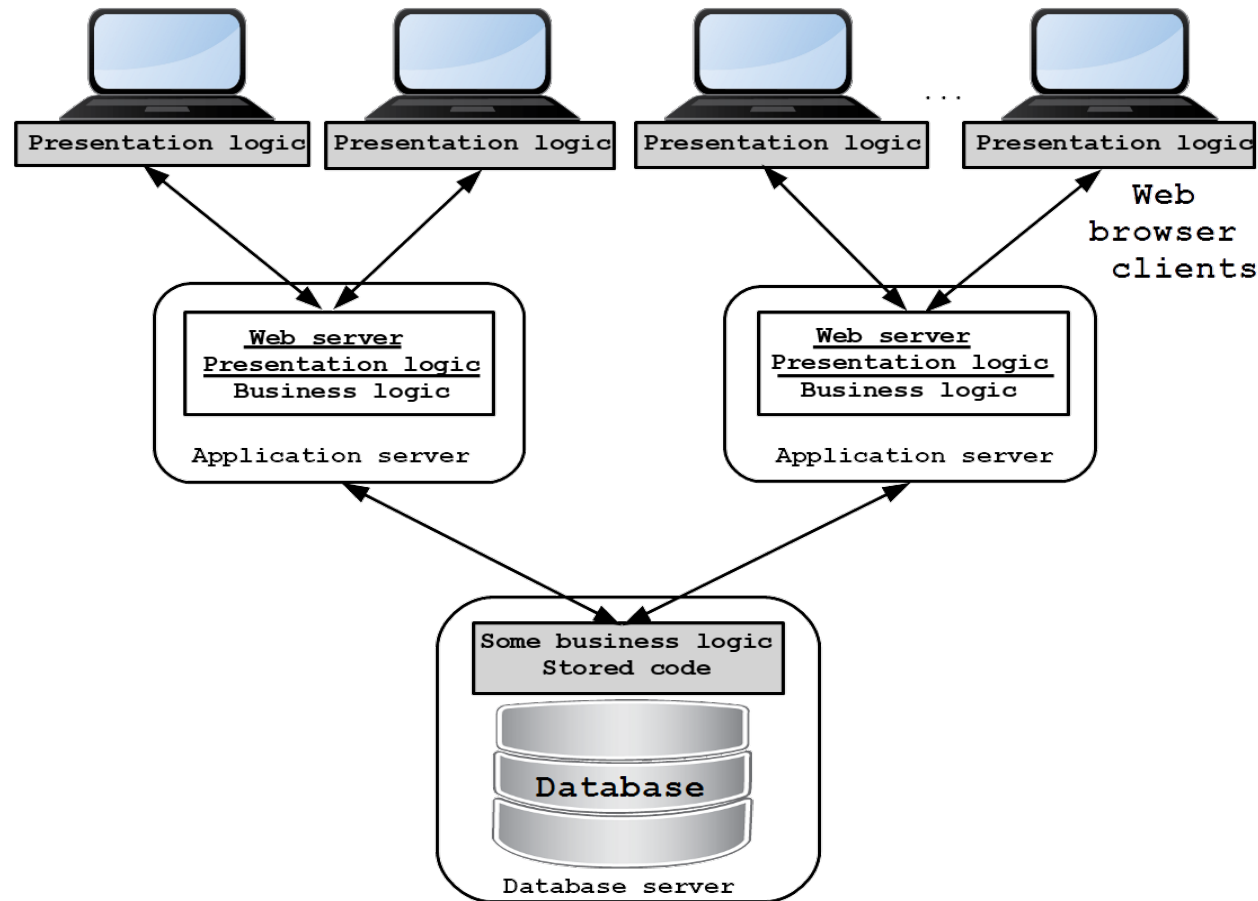
[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Web based architecture

In **Web based architecture** personal systems communicate with the Web servers that communicate with a **single database server**



In HTML view press 'p' to see the lecture notes
[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

14/37

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

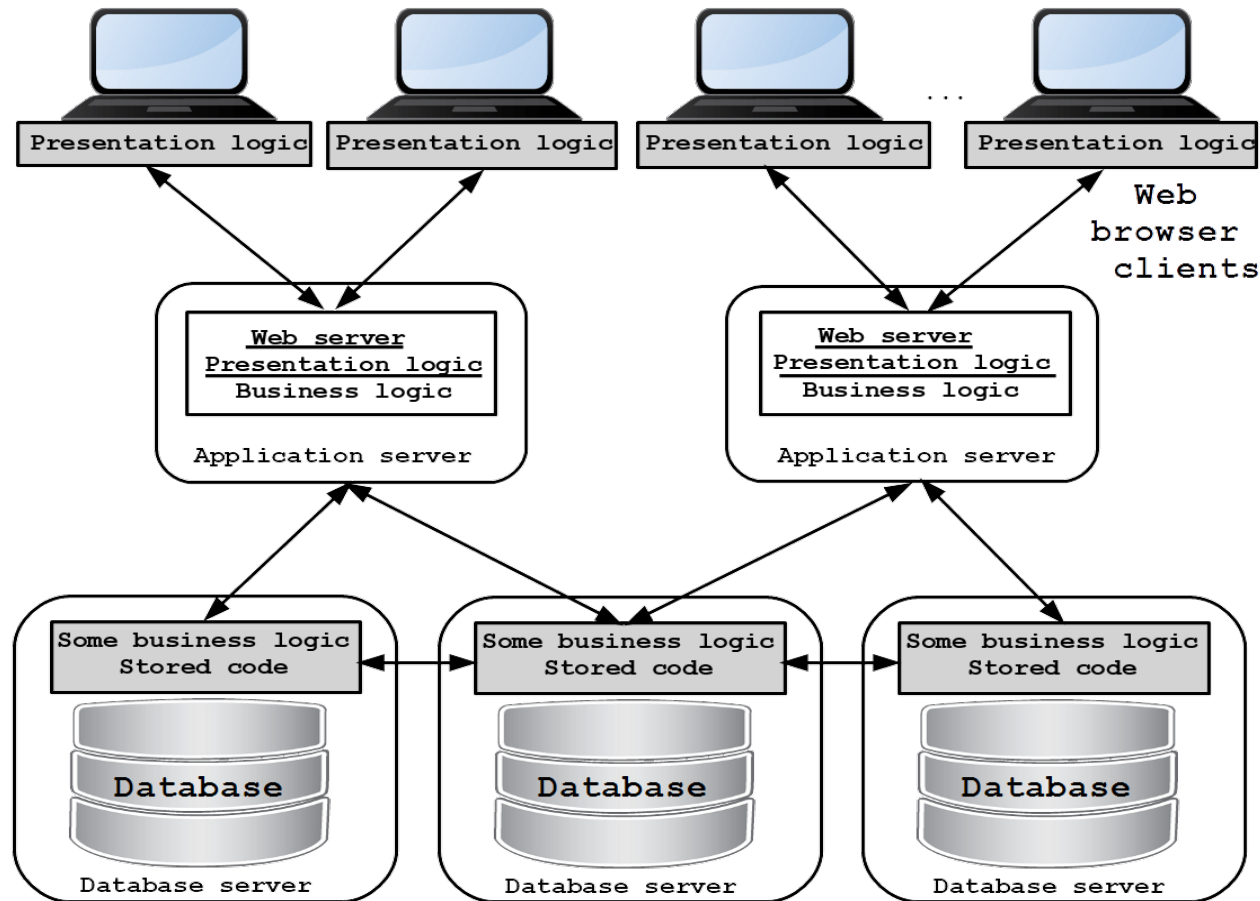
[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Distributed Web based architecture

In **Distributed Web based architecture** personal systems communicate with the Web servers that communicate with **many database servers**



[In HTML view press 'p' to see the lecture notes](#)
[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

16/37

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

[Autonomy](#)

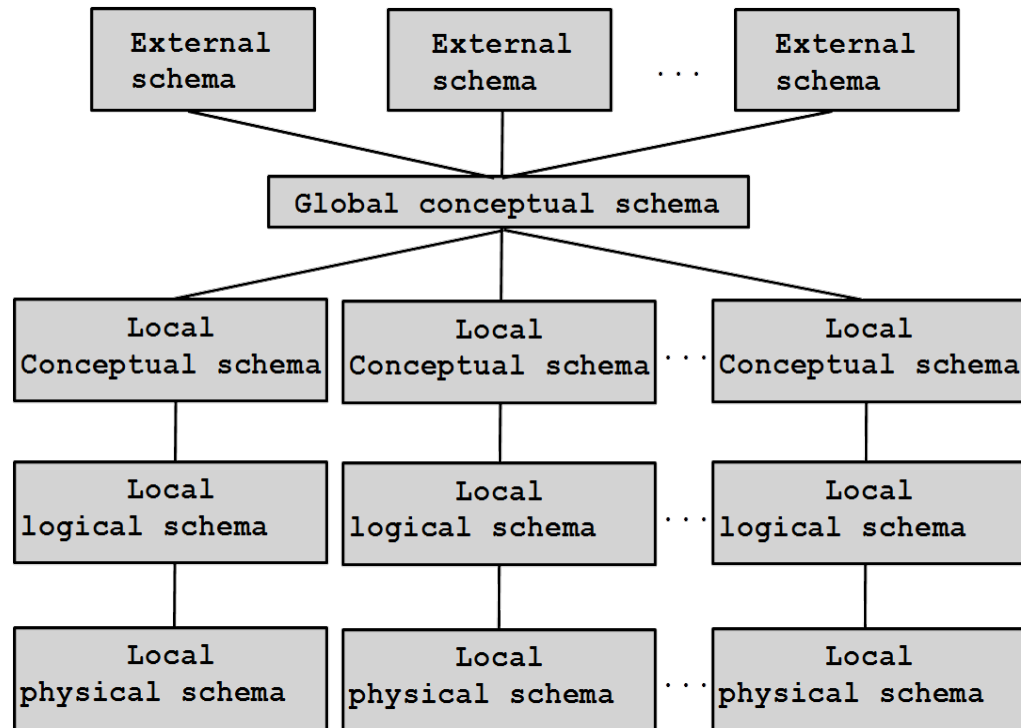
[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Distributed database schema architecture

In **distributed database schema architecture** a **global conceptual schema** hides distribution from the users

The users can see a **distributed database system** as a **single monolithic database system**



[In HTML view press 'p' to see the lecture notes](#)

[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

18/37

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Transparency

Transparency means hiding information from end users

Data organization (distribution or network) transparency means hiding network related information and data placement information; it is either **location** or **naming transparency**

Naming transparency allows for global naming of data objects

Location transparency allows the operations to be independent on the locations of data objects

Replication transparency means that users are unaware of the existence of multiple copies of the same data objects

Fragmentation transparency means that users are unaware of data fragmentation over many sites; it includes **vertical** and **horizontal fragmentation**

Transparency

Design transparency means that users are unaware of how distributed database was designed

Execution transparency means that users are unaware of how database transactions are processed

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Autonomy

Autonomy determines a level of independence of individual nodes in distributed database system

High degree autonomy is required for flexibility and customized maintenance of distributed database system

Design autonomy means a level of independence of data model usage and transaction management technique between the nodes

Communication autonomy means a level of independence to which a node can share information with other nodes

Execution autonomy means a level of independence to which users act as they please

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Advantages and disadvantages of DDBS

Higher level of **reliability** and **availability**

Improved **ease** and **flexibility** of application development

Improved **performance**

Easier **expansion**

Advantages and disadvantages of DDBS

Keeping track of data distribution

Distributed query processing

Distributed transaction management

Replicated data management

Distributed database recovery

Security

Distributed directory (catalog) management

Distributed Relational Database Systems

Outline

[Distributed database ? What is it ?](#)

[Centralized database schema model](#)

[Mainframe application architecture](#)

[Client-server architecture](#)

[Distributed client-server architecture](#)

[Web based architecture](#)

[Distributed Web based architecture](#)

[Distributed database schema architecture](#)

[Transparency](#)

[Autonomy](#)

[Advantages and disadvantages of DDBS](#)

[Distributed database system in PostgreSQL](#)

Distributed Database System PostgreSQL

Implementation of **distributed database system** requires access with SQL to data located at a **remote database system** from outside a **host database system**

SQL/Management of External Data (MED) allows for access to data located at an external database system through implementation of **foreign data wrappers (FDW)**

A module `postgres_fdw` in PostgreSQL provides the foreign data wrapper `postgres_fdw`

A module `postgres_fdw` replaces the older `dblink` module

Distributed Database System PostgreSQL

A procedure to access to data located at an external/remote database with a module `postgres_fdw` consists of the following steps:

- install an extension `postgres_fdw`
- create a `foreign server` to determine a location of data stored in an external PostgreSQL database system
- create a `user mapping`, to identify the user/role that will be used on a foreign/remote server
- create the `foreign tables` to access the relational tables located on a foreign/remote server

Distributed Database System PostgreSQLn

As an example, we consider a database that can be created through processing of the following **CREATE TABLE** statements

Relational table DEPARTMENT

```
CREATE TABLE DEPARTMENT(  
  name          VARCHAR(50)      NOT NULL,  
  code          CHAR(5)        NOT NULL,  
  total_staff_number DECIMAL(2)  NOT NULL,  
  chair         VARCHAR(50)    NULL,  
  budget        DECIMAL(9,1)   NOT NULL,  
  CONSTRAINT dept_pkey PRIMARY KEY(name),  
  CONSTRAINT dept_cke1 UNIQUE(code),  
  CONSTRAINT dept_cke2 UNIQUE(chair),  
  CONSTRAINT dept_cke1 CHECK (total_staff_number BETWEEN 1 AND 50) );
```

Relational table COURSE

```
CREATE TABLE COURSE(  
  cnum          CHAR(7)        NOT NULL,  
  title         VARCHAR(200)   NOT NULL,  
  credits       DECIMAL(2)     NOT NULL,  
  offered_by   VARCHAR(50)    NULL,  
  CONSTRAINT course_pkey PRIMARY KEY(cnum),  
  CONSTRAINT course_cke1 CHECK (credits IN (6, 12)),  
  CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)  
    REFERENCES DEPARTMENT(name) ON DELETE CASCADE );
```

[In HTML view press 'p' to see the lecture notes](#)
[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

30/37

Distributed Database System PostgreSQL

A user `xyz007` creates the relational tables `DEPARTMENT` and `COURSE` on a database server `db1teach.adeis.uow.edu.au`

A user `xyz007` loads data into the relational tables `DEPARTMENT` and `COURSE` on a database server `db1teach.adeis.uow.edu.au`

A user `xyz007` creates the relational tables `DEPARTMENT` and `COURSE` on a database server `db2teach.adeis.uow.edu.au`

A user `xyz007` would like to copy information about a department `Computer Science` from a database server `db1teach.adeis.uow.edu.au` to a database server `db2teach.adeis.uow.edu.au`

To do so a user `xyz007` must create the **foreign relational tables** with the names `REMOTE_DEPARTMENT` and `REMOTE_COURSE` on a database server `db1teach.adeis.uow.edu.au` and to **map** such tables into the relational tables `DEPARTMENT` and `COURSE` on a database server `db2teach.adeis.uow.edu.au`

[In HTML view press 'p' to see the lecture notes](#)

[TOP](#)

Created by Janusz R. Getta, CSCI235 Database Systems, Autumn 2024

31/37

Distributed Database System PostgreSQL

Then a user `xyz007` can remotely operate on the relational tables `DEPARTMENT` and `COURSE` on a database server `db2teach.adeis.uow.edu.au` without connecting to the server

The **foreign relational tables** `REMOTE_DEPARTMENT` and `REMOTE_COURSE` created on a database server `db1teach.adeis.uow.edu.au` represent the respective relational tables `DEPARTMENT` and `COURSE` on a database server `db2teach.adeis.uow.edu.au`

If in a general case we would like to access a relational table `R` located on a database server we are not connected to then we must create a **foreign relational table** `REMOTE_R` on a database server we are connected to and we must **map** a table `REMOTE_R` into a table `R` on a database server we are not connected to

A name of a **foreign relational table** is up to us but it must be different from the already existing relational tables or foreign relational tables

Distributed Database System PostgreSQL

To create a **foreign relational table** a user `xyz007` creates an extension `postgres_fdw` (note, that a privilege to create the extension has been already granted to a user `xyz007`)

Creating an extension postgres_fdw

```
CREATE EXTENSION postgres_fdw;
```

Next, a user `xyz007` creates a foreign server using `CREATE SERVER` statement to a database server on a host `db2teach.adaeis.uow.edu.au` listening on port `5432`

The database to which the connection is made is named `xyz007` on the remote server

Creating a foreign server

```
CREATE SERVER foreign_server FOREIGN DATA WRAPPER postgres_fdw  
  OPTIONS (host 'db2teach.adaeis.uow.edu.au', port '5432', dbname 'xyz007');
```

Distributed Database System PostgreSQL

Next, a user `xyz007` creates a **user mapping** with `CREATE USER MAPPING` statement to identify a user that will be used on the remote server

Creating a user mapping

```
CREATE USER MAPPING FOR xyz007 SERVER foreign_server
  OPTIONS (user 'xyz007', password 'xyz007');
```

Next, a user `xyz007` creates a **foreign relational table** `REMOTE_DEPARTMENT` with `CREATE FOREIGN TABLE` statement

Creating a foreign relational table REMOTE_DEPARTMENT

```
CREATE FOREIGN TABLE REMOTE_DEPARTMENT (
  name          VARCHAR(50)      NOT NULL,
  code          CHAR(5)         NOT NULL,
  total_staff_number DECIMAL(2)  NOT NULL,
  chair         VARCHAR(50)     NULL,
  budget        DECIMAL(9,1)    NOT NULL )
  SERVER foreign_server
  OPTIONS (schema_name 'public', table_name 'department');
```

Distributed Database System PostgreSQL

Next, a user `xyz007` creates a **foreign relational table** `REMOTE COURSE` with `CREATE FOREIGN TABLE` statement

```
CREATE FOREIGN TABLE REMOTE_COURSE(  
  cnum          CHAR(7)          NOT NULL,  
  title         VARCHAR(200)     NOT NULL,  
  credits       DECIMAL(2)       NOT NULL,  
  offered_by   VARCHAR(50)      NULL )  
  SERVER foreign_server  
  OPTIONS (schema_name 'public', table_name 'course');
```

Creating a foreign relational table REMOTE_COURSE

Next, a user `xyz007` copies information about Computer Science department from `DEPARTMENT` table to `REMOTE_DEPARTMENT` table

```
INSERT INTO REMOTE_DEPARTMENT(  
  SELECT *  
  FROM DEPARTMENT  
  WHERE name = 'Computer Science');
```

Copying information from DEPARTMENT to REMOTE_DEPARTMENT

Distributed Database System PostgreSQL

Next, a user `xyz007` copies information about the courses offered by Computer Science department from `COURSE` table to `REMOTE_COURSE` table

Copying information from COURSE to REMOTE_COURSE

```
INSERT INTO REMOTE_COURSE(  
  SELECT *  
  FROM COURSE  
  WHERE offered_by = 'Computer Science');
```

References

T. Connolly, C. Begg, Database Systems, A Practical Approach to Design, Implementation, and Management, Chapter 24 Distributed DBMs - Concepts and Design, Chapter 25.7 Distribution in Oracle, Pearson Education Ltd, 2015

[PostgreSQL 16.1 Documentation, F.38. postgres_fdw — access data stored in external PostgreSQL servers](#)