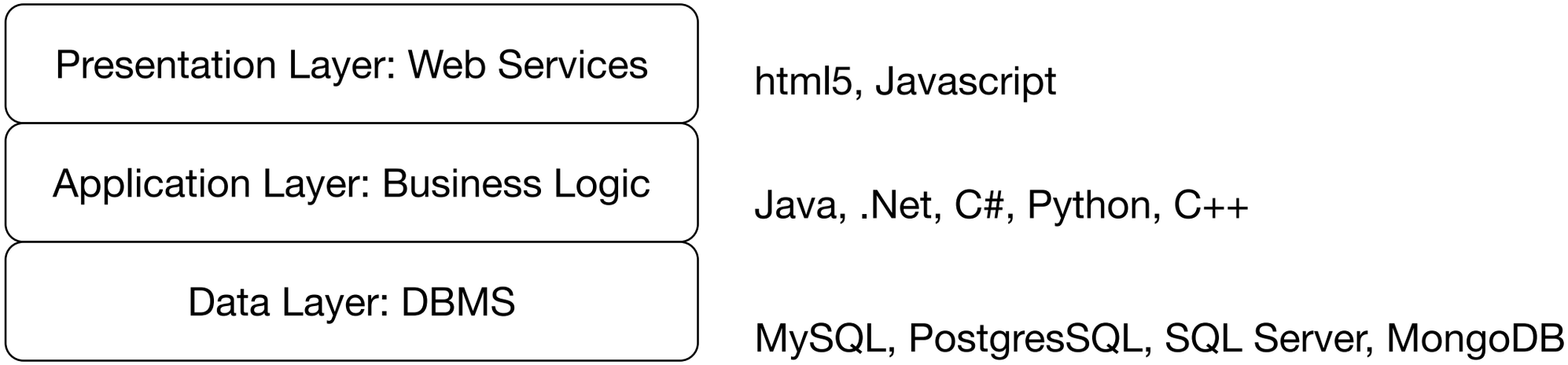
Database System Interactions

Thomas Schwarz, SJ

Three Tier Web Server Architecture

- Standard architecture for e-commerce sites
  - Tiered / layered architecture around since the THE operating system 1965



Three Tier Web Server Architecture

- Web Services Layer:
  - User interact with the site using a web browser
  - Forms, scripts, …
  - Requests are being routed to the application layer
    - Simple example: Embed PHP scripts into a web server
      - Download: LAMP / WAMP / XAMPP etc. With Apache, MySQL, PHP, Perl, …
      - Embed PHP script in HTML: <?php … ?>

Three Tier Web Server Architecture

- Application Tier
  - Simple system: Bypass application tier by directly translating web requests to database requests
  - Normally:
    - Integrate different databases
    - Implement business logic

Three Tier Web Server Architecture

- Database Tier:
  - Executes queries (including updates and inserts)

Integrating SQL with Application Layer

- Application layer uses languages like PHP, Python, Java, …
- Needs to interact with an application programming environment

SQL Environment

- SQL environment
  - Schemas: Tables, views, assertions, triggers, stored procedures, character sets, grant statements (for rights) maintained by a catalog
  - Servers / Clients
    - Clients need to connect to a server
    - Client/server connection is divided into Sessions
      - Each session selects a catalog and a schema

Integrating SQL with Application Layer

- Impedance mismatch problem
  - All languages / environment are Turing complete
  - Standard SQL is not:
    - Not everything that a computer can do can be done with SQL
      - E.g. cannot compute factorial with SQL
- Need to use both SQL (to interact with database) AND application level program

Integrating SQL with Application Layer

- Program sets up a connection to a database and closes it at the end
  - which might be automatic

Integrating SQL with Application Layer

- Central idea is the 'cursor'
  - Basically a pointer into the result table of an SQL query
  - Usually:
    - Can get result table row by row
    - Can get result table all at once
      - Could be hard on memory resources
    - Can get result table in tranches

Integrating Python with MySQL

- Solutions differ widely according to application tier environment and
  - Here: look at how to connect Python with MySQL
    - There are a variety of Python packages that will do that
      - I chose SQL-connector

Python 3 SQL connector

- Needed: Python 3
- Install MySQL Connector
    - Install with pip
        - Be careful for which Python you install
            - E.g. Mac has a Python 2.7 installed as part of the OS
- You will need to know your MySQL password
    - If necessary, just re-install MySQL

Python 3 MySQL Connector

- You can use
  - https://www.mysqltutorial.org/python-mysql/

Python 3 MySQL Connector

- Write a Python 3 program that
- Task 1:
    - finds all employees with a given last name
- Task 2:
    - inserts an employee with a given first and last name and emp_no 600000.
- Task 3:
    - changes the hiredate of the employee with emp_no 600000 to today
- Task 4:
    - deletes the employee with emp_no 600000

Homework

- Write a Python program that connects to your databate and
    - Insert an employee "John Adams" into the database working as a Senior Engineer and earning 200000 as of today in the Research department
    - Find all information on all employees called "John Adams"
    - Delete the newly inserted record from the database
- (If you are not familiar with Python, you can use any other language that has a connector)