Project of Software Development

Lecture 6 - 13Feb

The Data tier: Data and Database Management

PDSOFT S14 | Prof. Olivier Carneiro

ISEG Lisbon School of Economics & Management Universidade de Lisboa

U LISBOA | UNIVERSIDADE DE LISBOA

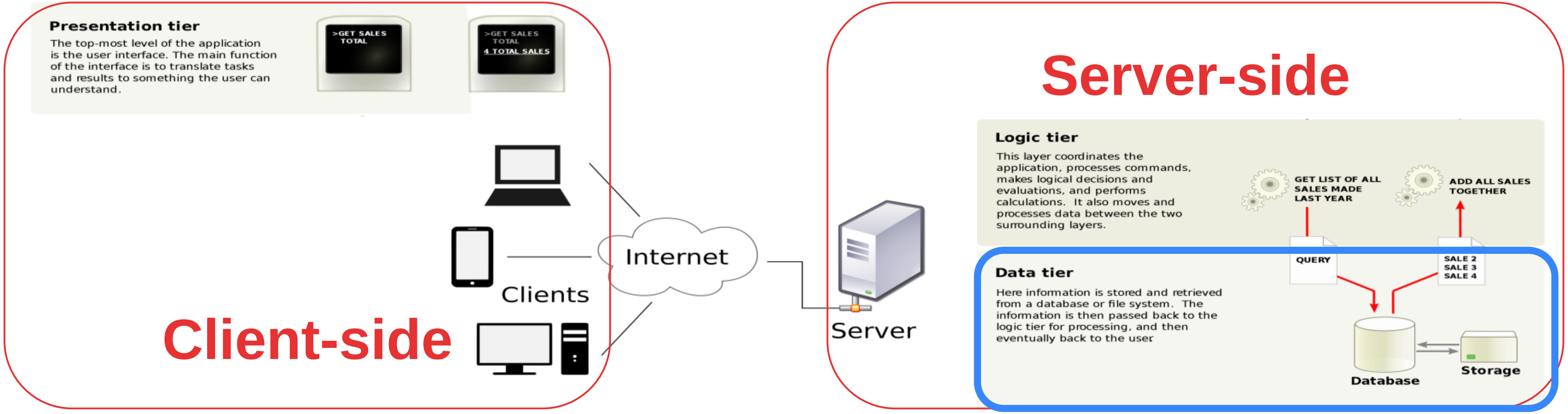
Lecture Topic

The Data tier: Data and Database Management

Bibliography:

- Hoffer, J. A., Ramesh, V., & Topi, H. (2020). Modern Database Management (13th ed.), Harlow: Pearson Education Limited, UK:
  - Chapter 1: All chapter's content **up to and including** "Components of the Database Environment"
  - Chapter 2: Introduction; data names and definitions; modeling Entities and attributes; modeling relationships. Examples are optional
  - Chapter 4: All chapter's content **up to and including** "Third normal form"
  - Chapter 5: All chapter's **except** "Defining a database in SQL"
  - Chapter 6: All chapter's content **up to and including** "self-join"

The Data tier: Data and Database Management



Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Clients
Client-side
Internet
Server
Server-side
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user
Database
Storage

Why does data and data management matter?

Remind the BBC video “How does the internet work? (@BBC Click - 2019)” (first 35 seconds)

Why does data management matter?

How do you store and access your data?

Apple Notes

Files (text, CSV, etc.)

Google Docs

Tuesday 12

Apple calendar

OneDrive

Google Sheets

Excel

Google Keep

Google Drive

Dropbox

Google Calendar

31

TeraBox

Relational Databases: concepts

Database: an organized collection of logically related data

Data: **facts concerning objects and events** that could be recorded and stored on computer media. A stored representation of objects and events that **have meaning and importance** in the user's environment. Examples: customer name, address, telephone number, order, product, price, images, documents,

The most important **structured data** types are numeric, character, and dates. Structured data are stored in **tabular form** (in tables, relations, arrays, spreadsheets, and so forth) and are most commonly found in traditional databases and data warehouses.

Data versus Information: **Information** is data that has been processed in such a way that the knowledge of the person who uses the data is increased. **Information provides meaning to data**.

Relational Databases: data vs. information examples

Temperature Readings:

Data: Consider raw temperature readings collected from various sensors around a city, such as 32°C, 35°C, 38°C, etc. These readings by themselves are just numbers without context.
Information: When these readings are processed to find an average temperature for the city or to compare the current day's temperature with historical data to identify a trend, they become information. For example, "The average temperature in the city today is 35°F, which is 5 degrees above the average for this time of the year."

Customer Feedback Surveys:

Data: Responses to a customer feedback survey where customers rate their satisfaction on a scale of 1 to 5 for various services. The responses collected are raw data, such as a series of numbers: 4, 3, 5, 2, etc.
Information: When these responses are analyzed to determine the overall customer satisfaction level or to identify areas needing improvement, they provide information. For example, "85% of customers rated their satisfaction level as 4 or above, indicating high satisfaction with our services."

Website Traffic:

Data: Raw logs of website traffic that include details like IP addresses, timestamps, page requests, and durations of visits. This data is a collection of numerous bits of information that, on their own, don't provide actionable insights.
Information: When this data is analyzed to identify patterns, such as the most visited pages, average visit duration, or traffic sources, it becomes valuable information. For instance, "The product page receives 30% more visits than any other page, with visitors spending an average of 5 minutes, suggesting high interest in our products."

Relational Databases: concepts

Data model: graphical systems used to capture the nature and relationships among data.

Entity or table: is like a **noun** in that it describes a person, a place, an object, an event, or a concept in the user environment about which the organisation wishes to maintain data. Example: Customer, Order, User, Product, Review, Payment, Event, Volunteer, etc.

Instance or record or row: the information about each individual element of an entity. Examples: The customer John Doe with email a@a.com and 31 years old; an Order of 2 products at the price of 5€ each; the payment of 35€ on the 10th of January for the order nº71, etc.

Attribute (or field or column): the data to be collected about the entity. Example: Customer's name, email, date of birth, shipping address; Order's date, products, prices, shipping address, payment status, shipping status, invoice; Review's nr of stars, comments, order to which it refers to; etc.

Primary Key: an attribute (or a combination of attributes) that uniquely identifies each instance (or record or row) in a relation

Relational Databases: concepts

Relationship between entities: describes how entities relate to other entities. Most relationships are one-to-many (1:M) or many-to-many (M:N). one-to-one (1:1) is a particular case of the one-to-many (1:M).

- one-to-many (1:M) or one-to-one (1:1) relationships are implemented via an attribute
- many-to-many (M:N) relationship is implemented via an entity

Foreign Key: a foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. It is a link between two tables in a database

Examples of relationships:

- A company has multiple employees. **What type of relationship is this?**
- Each employee has a User in the App. **What type of relationship is this?**
- A order has multiple products ordered. **What type of relationship is this?**

Relational Databases: example

You are tasked with designing a simplified database schema for a School Management System. This system needs to manage information about teachers, students, courses, and classrooms. Requirements:

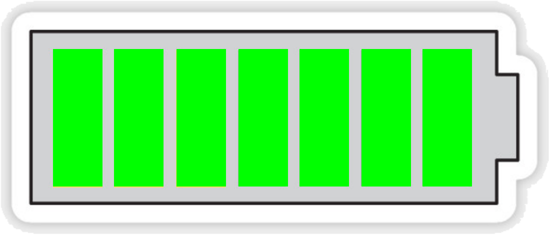
- Teachers: Each teacher can teach multiple courses, but each course is taught by only one teacher.
- Students: Each student can enroll in multiple courses, and each course can have multiple enrolled students.
- Courses: Courses are taught in one classroom, but each classroom can host multiple courses at different times.
- Classrooms: we need to collect room number, floor, and building.

Which are the Entities (or tables)?

And the Entities' attributes?

And the Entities' relationships?

5 minutes break



Relational Databases: School Management System

- Teachers: Each teacher can teach multiple courses, but each course is taught by only one teacher.
- Students: Each student can enroll in multiple courses, and each course can have multiple enrolled students.
- Courses: Courses are taught in one classroom, but each classroom can host multiple courses at different times.
- Classrooms: we need to collect room number, floor, and building.

Entities and attributes:

Teacher: TeacherID (Primary Key), Name, Email

Student: StudentID (Primary Key, Name, EnrollmentYear

Course: CourseID (Primary Key), Title, TeacherID, ClassroomID

Classroom: ClassroomID (Primary Key), RoomNumber, Floor, Building

Relational Databases: School Management System

- Teachers: Each teacher can teach multiple courses, but each course is taught by only one teacher.
- Students: Each student can enroll in multiple courses, and each course can have multiple enrolled students.
- Courses: Courses are taught in one classroom, but each classroom can host multiple courses at different times.
- Classrooms: we need to collect room number, floor, and building.

Relationships:

- 1-to-M relationship between Teachers and Courses:
  Courses: CourseID (Primary Key), Title, **TeacherID (Foreign Key),** ClassroomID
- 1-to-M relationship between Courses and Classrooms:
  Courses: CourseID (Primary Key), Title, TeacherID (Foreign Key), **ClassroomID (Foreign Key)**
- M-to-N relationship between Students and Courses:
  StudentCourse (new entity to represent the relationship): **StudentID (Foreign Key), CourseID (Foreign Key)**

Database Commands: CRUD

Insert (or Create): Adds new data

Select (or Get or Read): Retrieves data

Update: Modifies existing data

Delete: Removes data

Database Commands: SELECT

The SELECT statement allows to retrieve just the data we're interested in. Joins enable us to combine data from multiple tables, providing a powerful tool for data analysis.

Type of joins:
Inner Join (or **Only With**) : Returns rows when there is at least one match in both tables.
Left Join (or **With or Without**): Returns all rows from the left table, and the matched rows from the right table.

Example:

Retrieve all students enrolled in a course?

SELECT * FROM StudentCourse INNER JOIN Student ON StudentCourse.StudentID = Student.StudentID WHERE StudentCourse.CourseID = 53

Retrieve all courses and their classrooms, if defined?

SELECT * FROM Course LEFT JOIN Classroom ON Course.ClassroomID = Classroom.ClassroomID

Advanced SQL: LIKE Operator

The LIKE operator, combined with wildcards, allows to perform pattern matching in queries.

Wildcards: are special characters used with the LIKE operator to perform pattern matching in SQL queries:

- %: matches any sequence of characters
- _: matches any single character

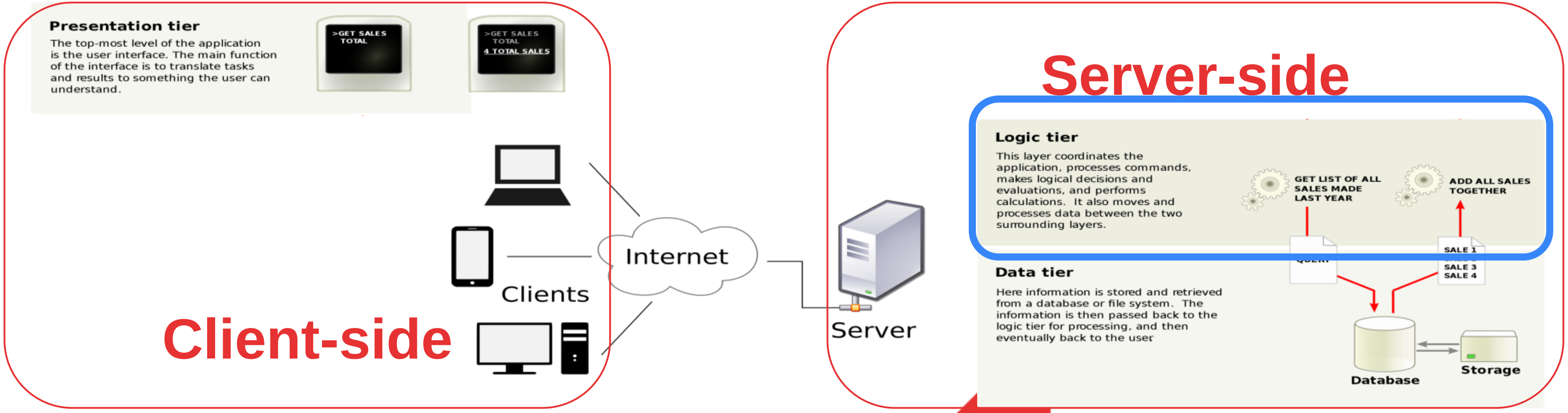
Example:

Retrieve all students with last name “Ronaldo”?

SELECT * FROM Student WHERE Name LIKE “%Ronaldo”

OutSystems Online Training: Becoming a Web Developer

Short review and Q&A: Logic and actions & exercise solution



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user

Learning goals

- Relational Databases:
  - Entities, attributes
  - Relationships: primary key, foreign key
- Define, manipulate and search data in a Relational Database Management System:
  - Insert, update, delete, select
  - Left join, inner join, like
  - Wildcards: % _

Next time, we will start programming!

The real fun starts in the next time 😊! Bring your laptop or sit in front of a PC as we will start implementing a WebApp in-class

⚠️ You will need to be familiar with all in-class content and all homework up to and including Lecture 6 (today's lecture). Please make sure you are prepared in order to keep up with the pace.

Homework outsystems

OutSystems Online Training: Becoming a Web Developer
https://learn.outsystems.com/training/journeys/web-developer-662

Modeling Data (45 minutes)
https://learn.outsystems.com/training/journeys/web-developer-662/modeling-data/o11/372

1. Modeling Data, 2. Database Entities, 3. Static Entities

After having successfully finished the exercise of item “#2. Database Entities”, please take a screenshot of your full screen, showing the data previewer and the Employee Entity content visible (equal to the one in the exercise) **but also having ServiceStudio behind it visible in the screenshot**. Then, create a WebApp with a WebPage containing this screenshot:

1. Add the “anonymous” role to all your Web Screens:
    a. Click on the Web Screen (widget tree on the right side)
    b. The attributes area will open. Click on “Anonymous”
    c. Repeat (a) and (b) for each Web Screen you created
    d. Publish and test
2. Submit the address of your Web Application by following up on the email

Expected total effort: 45 to 60 minutes