# IN1013 Databases

Vitaly Aksenov

# Discussion Zone

CITY UNIVERSITY LONDON
EST 1894

CUoL-2024-2025. IN1013. Databases. | General | Microsoft Teams

# Practice Overview

CITY UNIVERSITY LONDON
EST 1894

- There is a Tutorial after the lecture
- Several examples are explained in class and others are homework
- The answers for homework will be explained next week
- For homeworks create a public github repo, e.g., IN1013, and submit its link to the form https://forms.office.com/e/Dmiy6ppfbe
- For each homework, create a folder “HW #” (for example, “HW 1”) and the answers for each task put in a pdf file “Task #.pdf” (for example, “Task 3.pdf”)
- The number of the homework and the task numbers are given in Moodle
- Each homework costs 1 point in total
- Each task costs points proportional to the number of tasks

# Grading


CITY UNIVERSITY LONDON

- Homeworks
  - Each homework costs 1 in total
  - Each task is proportional to the number of tasks in HW
- Two courseworks:
  - Database Design - 47 points in total (+3 as HWs)
  - Database Operations - 46 points in total (+4 as HWs)

# Overview of Database Technology

# Why are ‘Databases on the Web’ Important?

CITY UNIVERSITY LONDON
EST 1894

- Databases are established technology for managing large amounts of data
- The Web is a good way to present information
- Separating data management from presentation improves efficiency
  - updating
  - finding information

# Marking Scheme


CITY UNIVERSITY LONDON

The marking scheme is a little bit complicated.

There two courseworks: the first one is 46 points, the second one is 47 points.

There are 7 homeworks

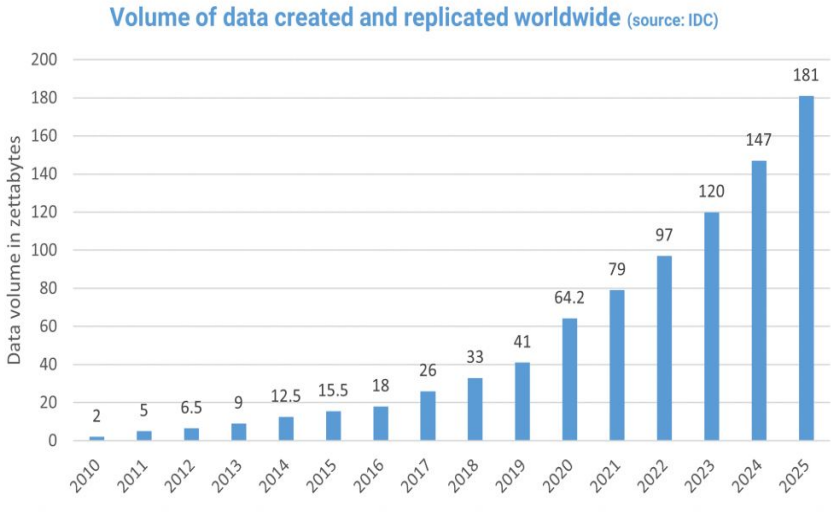
# Businesses use database technology


CITY UNIVERSITY LONDON

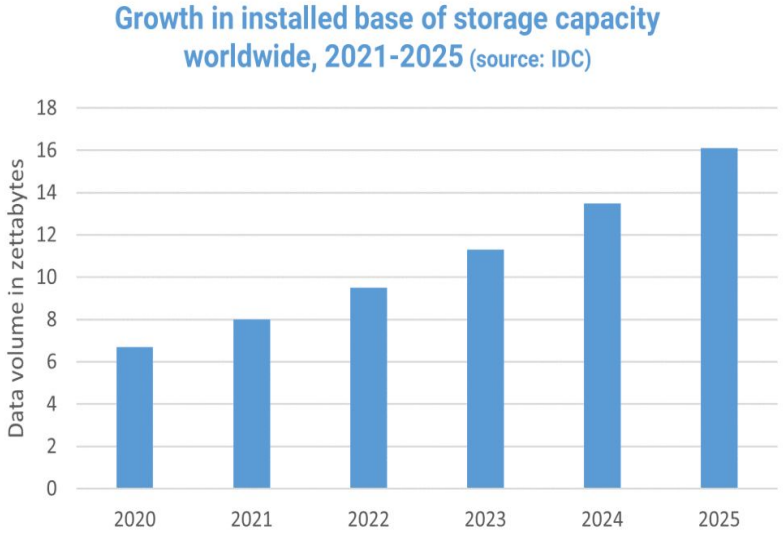





# Usage vs storage




1 zettabyte = 1 million million gigabytes




Source: [url]

5.35 billion users while one creates 15.87TB daily ([url])

# Examples of Websites Using Databases


CITY UNIVERSITY LONDON

- Organizational information services
  - employee directories
- Booking & scheduling
  - airlines, university courses signups
- Electronic commerce
- Website automation
  - www.google.com
  - www.webmonkey.com

# Databases

CITY UNIVERSITY LONDON
EST 1894

- Database
  - an organized collection of data
  - paper-based
- DBMS (database management system)
  - software to enable user to create and maintain databases
- Relational database
  - organizes data into tables
  - RDBMS

# Database Development


CITY UNIVERSITY LONDON

- Text-based
- File systems
- Hierarchical
- Relational
- NoSQL

# Simple Text-based Databases


CITY UNIVERSITY LONDON

- Structure
  - Name of columns
  - Data as text
- Example:

```
Name,Subject,Mark
Kam,Databases,86
Kam,Programming in Java,70
Olga,Databases,91
Olga,Programming in Java,65
```

# Structured Text-based Databases


CITY UNIVERSITY LONDON

- Structure
  - Name of columns, types, and lengths
  - Data as text of the same type
- Example:

```
Name            Subject              Mark
String, 10      String, 30           Number, 2
Kam             Databases            86
Kam             Programming in Java  70
Olga            Databases            91
Olga            Programming in Java  65
```

# Advantages and Disadvantages

CITY UNIVERSITY LONDON
EST 1894

- Advantages:
  - Simple to read
- Disadvantages:
  - Hard to search
  - Hard to operate with
  - Hard to use different formats for an attribute
  - No specific checking
- Examples:
  - Excel
  - dBase

# File System Databases

CITY UNIVERSITY LONDON
EST 1894

Representation:

- One file - one record
- Filepaths are the dependent records

Example:

```
Kam/Data - Full name, address, etc.
Kam/Mark/Databases - 86
Kam/Mark/Programming in Java - 70
Olga/Data - Full name, address, etc.
Olga/Mark/Databases - 91
Olga/Programming in Java - 65
```

# Advantages and Disadvantages

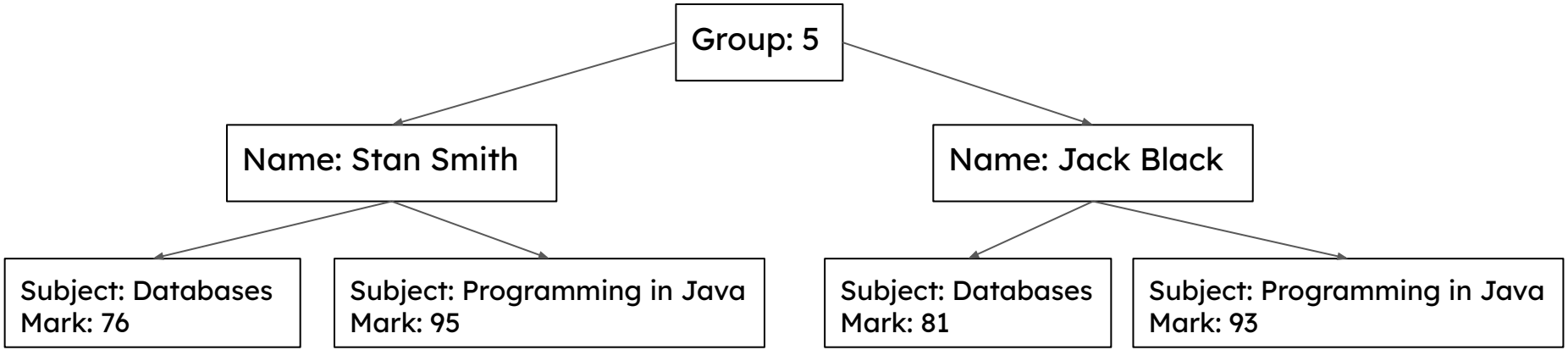
CITY UNIVERSITY LONDON
EST 1894

- Advantages
  - Data is structured
  - Simple to implement
- Disadvantages
  - Hard to get the data
  - No checks of consistency
  - Too many files
- Examples:
  - FATx, NTFS

# Hierarchical Databases

CITY UNIVERSITY LONDON
EST 1894

- Proposed and implemented by IBM in 1966
- Representation:
  - The tree of relations
  - Relation: parent-child



# Advantages and Disadvantages

CITY UNIVERSITY LONDON
EST 1894

- Advantages
  - Checks for the consistency and relations
- Disadvantages
  - Only tree structure
  - No many-to-many
- Examples:
  - IBM DBOMP
  - Windows Registry

# Relational Databases

CITY UNIVERSITY LONDON
EST 1894

- Proposed by E. Codd in 1969
- Structure:
  - Data is in tables
  - Checks for consistency and relations
  - Relations are set through the queries

| Id | Name |
|---|---|
| 1 | Stan Smith |
| 2 | Jack Black |

| Student Id | Subject Id | Mark |
|---|---|---|
| 1 | 1 | 73 |
| 1 | 2 | 95 |
| 2 | 1 | 83 |
| 2 | 2 | 91 |

| Id | Name |
|---|---|
| 1 | Programming in Java |
| 2 | Databases |

# Advantages and Disadvantages


CITY UNIVERSITY LONDON

- Advantages
  - Allow all relations
  - **Mathematical** model
- Disadvantages
  - Complex implementation
  - Hard to represent hierarchical data
  - Hard to make efficient queries

    (A Join B) Join C or A Join (B Join C)
- Examples:
  - Oracle Database
  - MySQL

# NoSQL Databases

CITY UNIVERSITY LONDON
EST 1894

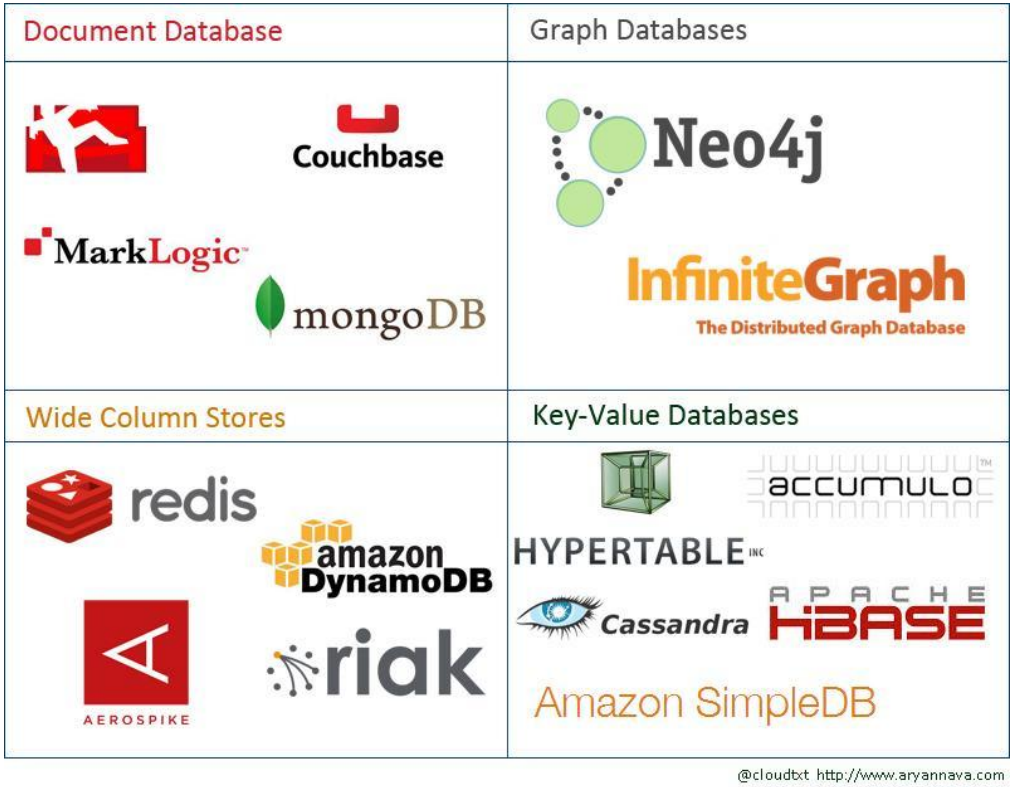
Proposed around 1998

Types:

- Document Database (Json, XML)
- Key-value storage
- Graph database
  - Graph of objects
  - Data in nodes and on edges
- Wide-column Store
  - Format can change depending on a row

# NoSQL Databases

CITY UNIVERSITY LONDON
EST 1894

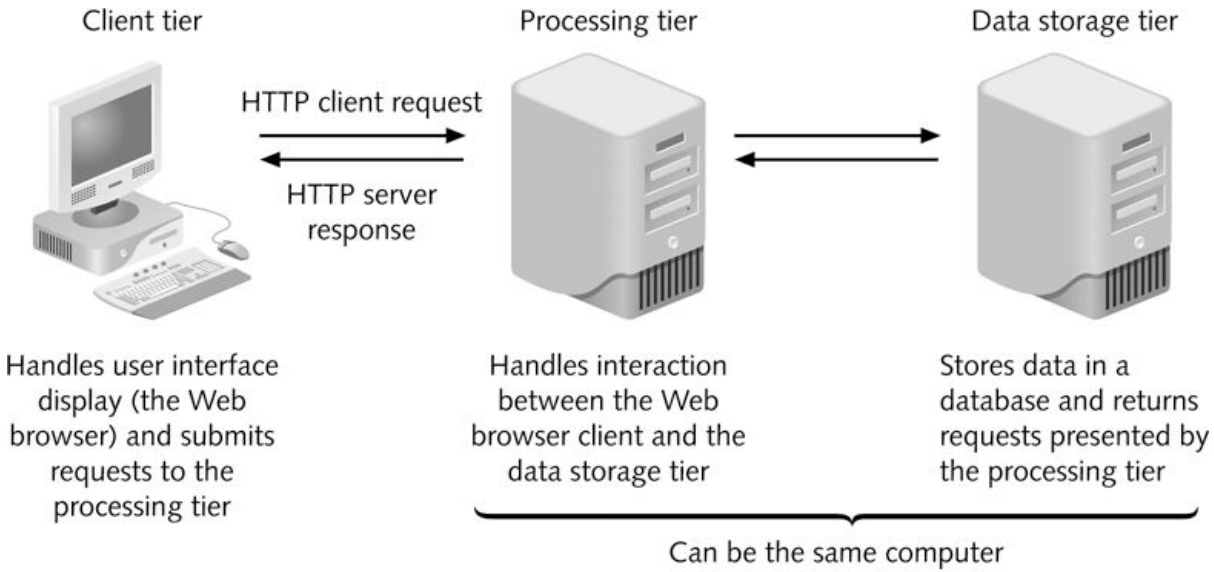


# Advantages and Disadvantages


CITY UNIVERSITY LONDON

- Advantages
  - Big choice depending on the problem
  - Flexible
  - Fast
- Disadvantages:
  - Most things are in code
  - No optimizer
  - Very easy to make a mistake

# Client/Server Architecture




**The design of a three-tier client/server system**

# How do Databases integrate with the Web?

CITY UNIVERSITY LONDON
EST 1894

- Databases
  - MS Access
  - MySQL (MariaDB), mSQL
  - Oracle, Sybase, MS SQL Server
- Integration tools
  - Java EE, PHP or CGI, Servlets, JSP, ASP etc.
  - “Middleware”: e.g. ColdFusion
- Browsers (Firefox, Chrome etc)
  - HTML
  - CSS
  - Javascript

# Web-enabled database technology

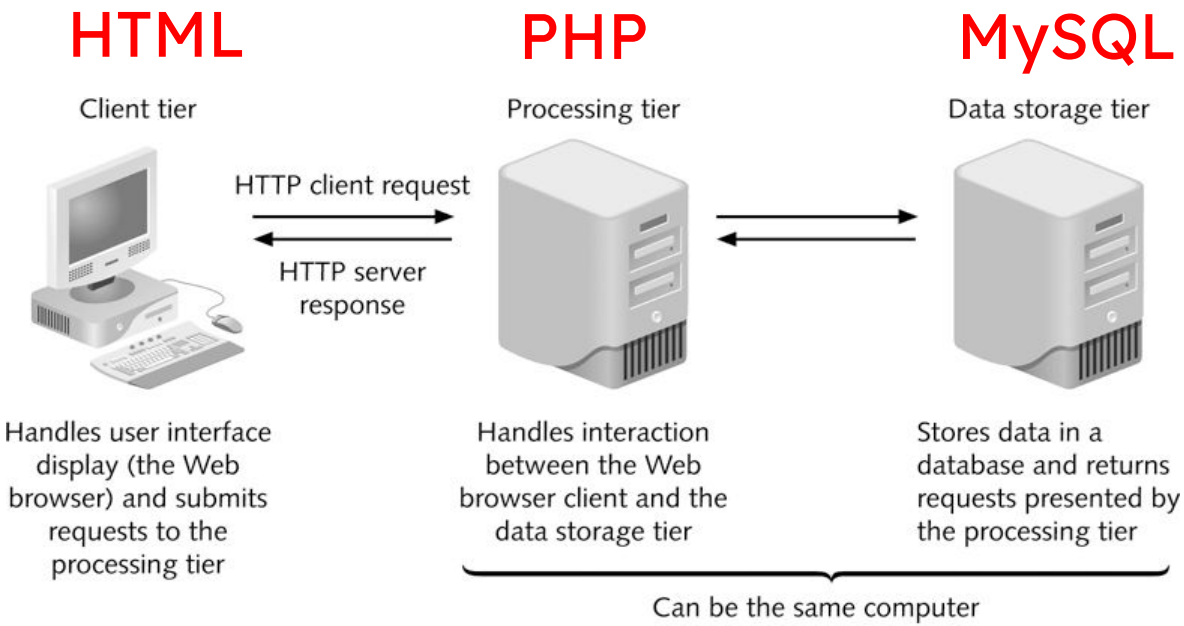
CITY UNIVERSITY LONDON
EST 1894

Browser, presentation of data

Integration layer, logic and functionality

Database, storage of data

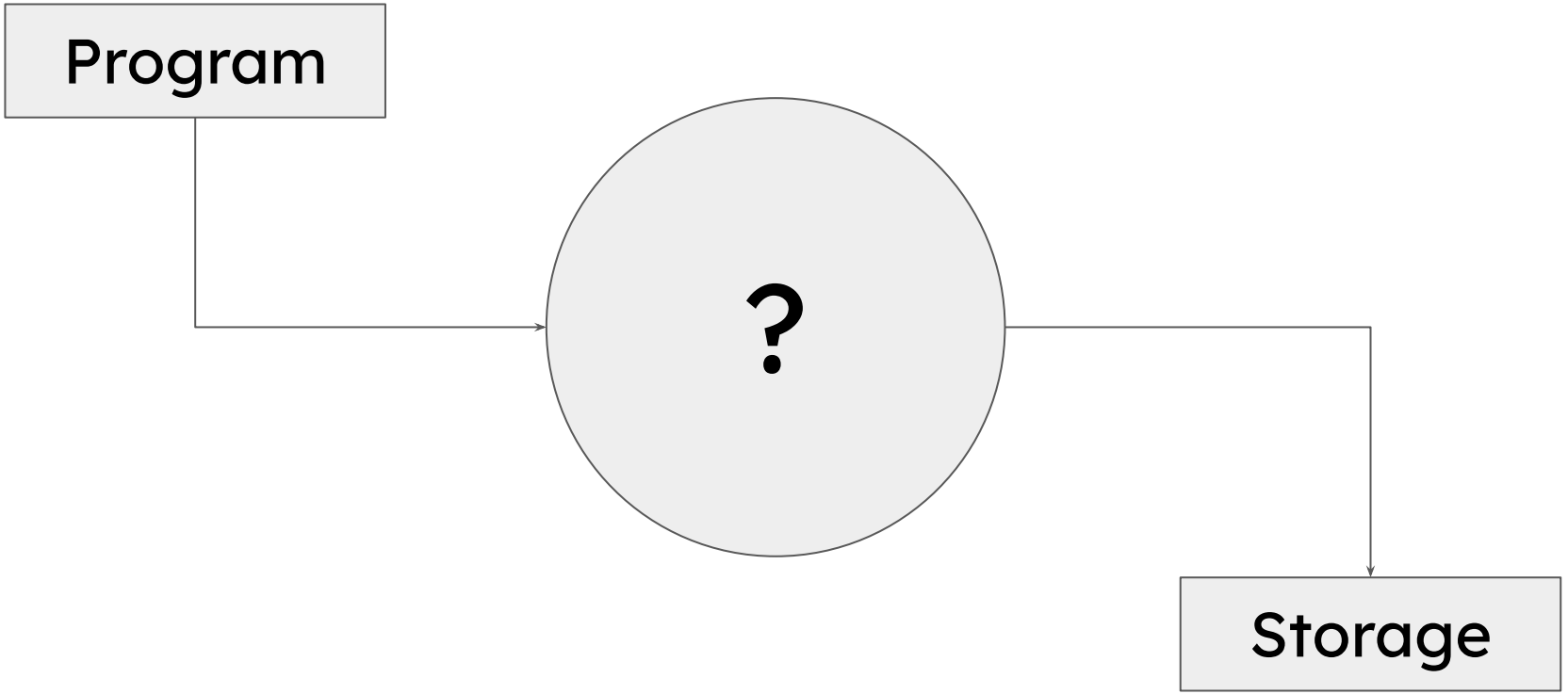
# Client/Server Architecture




**The design of a three-tier client/server system**

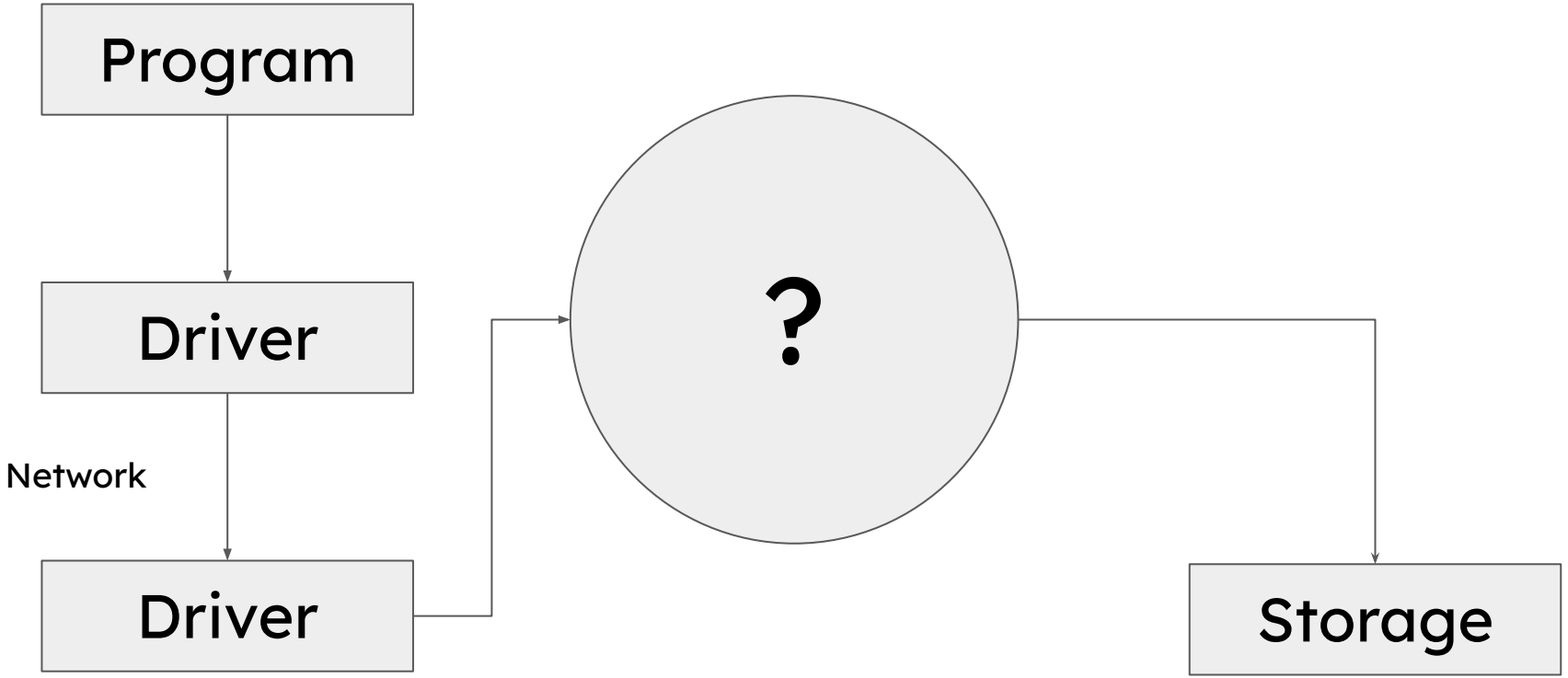
# Work with RDBMS

CITY UNIVERSITY LONDON
EST 1894



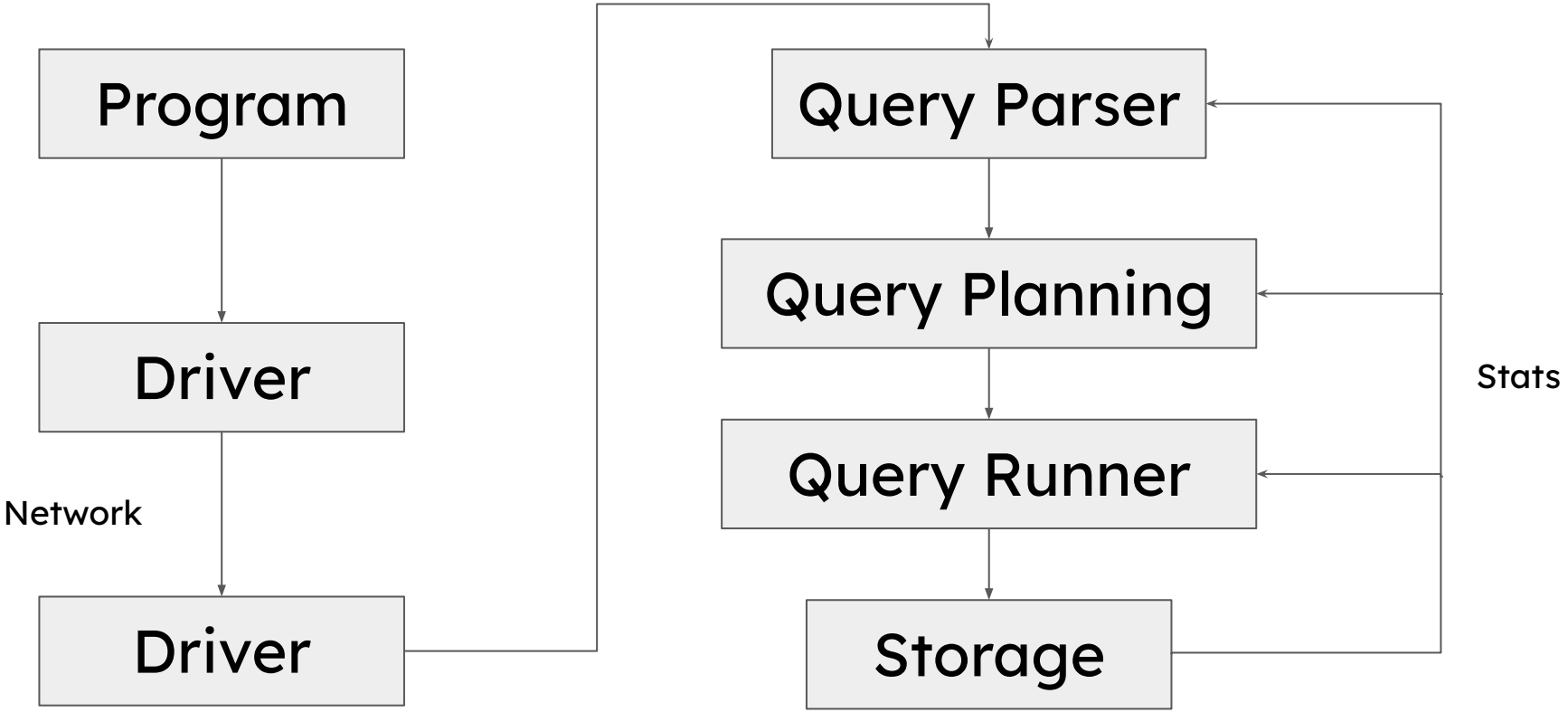
# Work with RDBMS

CITY UNIVERSITY LONDON
EST 1894



# Inside RDBMS

CITY UNIVERSITY LONDON
EST 1894



## Why Relational Databases are so widely used

CITY UNIVERSITY LONDON
EST 1894

- **Organization:** Easy to categorize and store data
- **Extensible:** Easy to extend and not reliant on physical organization
- **Accuracy:** Data is stored just once
- **Flexibility:** Complex queries are easy for users to carry out
- **Collaboration:** Multiple users can access the same database
- **Trust:** Relational database models are mature and well-understood
- **Security:** Data in tables within a RDBMS can be limited to allow access by only specific users

# Core Concepts of Relational Databases

# Recap

CITY UNIVERSITY LONDON
EST 1894

- Database – a collection of data that is logically coherent
- DBMS – Database Management System
  - defines, creates, and maintains a database
  - Allows users controlled access to data in the database
- Relational Database – data is organized into two dimensional tables

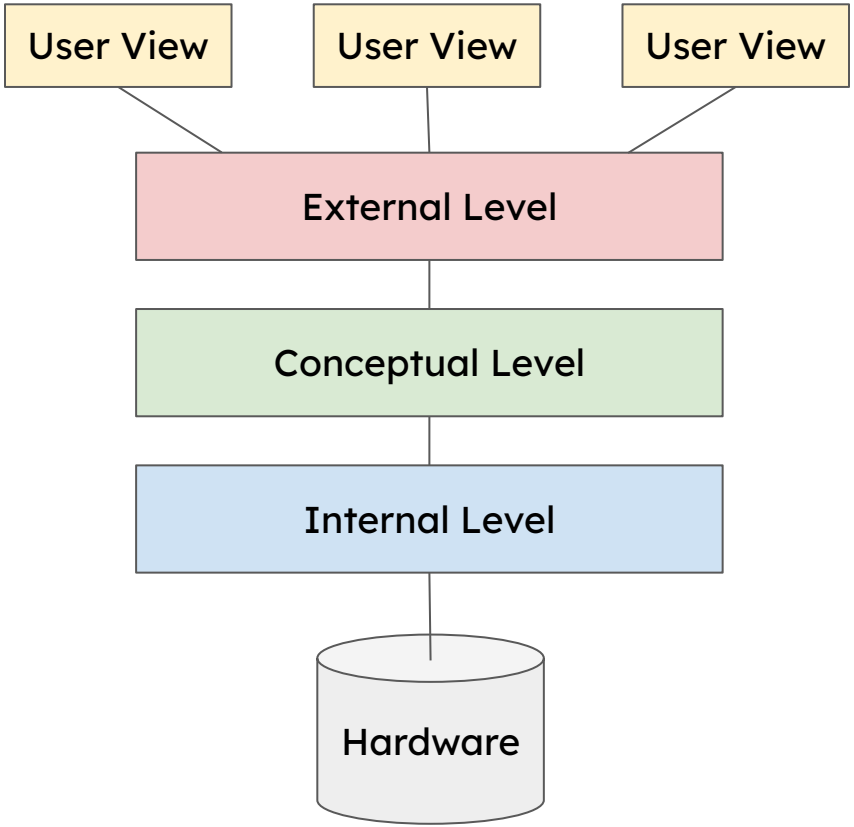
# Relational model

CITY UNIVERSITY LONDON
EST 1894

- Tables are the external view
  - The data are represented as a set of relations
  - A relation is a two-dimensional table
- This doesn't mean that data are stored as tables; the physical storage of the data is independent of the way the data are logically organized.

# DBMS Architecture

CITY UNIVERSITY LONDON
EST 1894



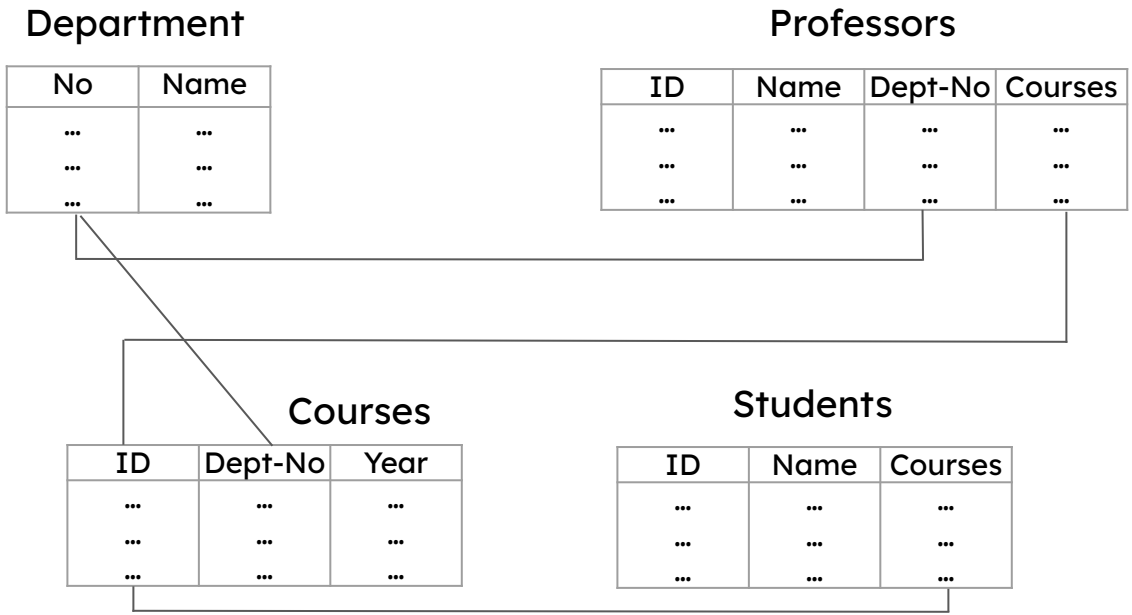
# Architecture

CITY UNIVERSITY LONDON
EST 1894

- Internal level
  - Determines where data are actually stored on the storage device
  - Low-level access method
- Conceptual level
  - Defines the logical view of the data
  - The main functions of DBMS are in this level
- External level
  - Interacts directly with the user
  - Change the data coming from the conceptual level to a format and view that are familiar to the users.

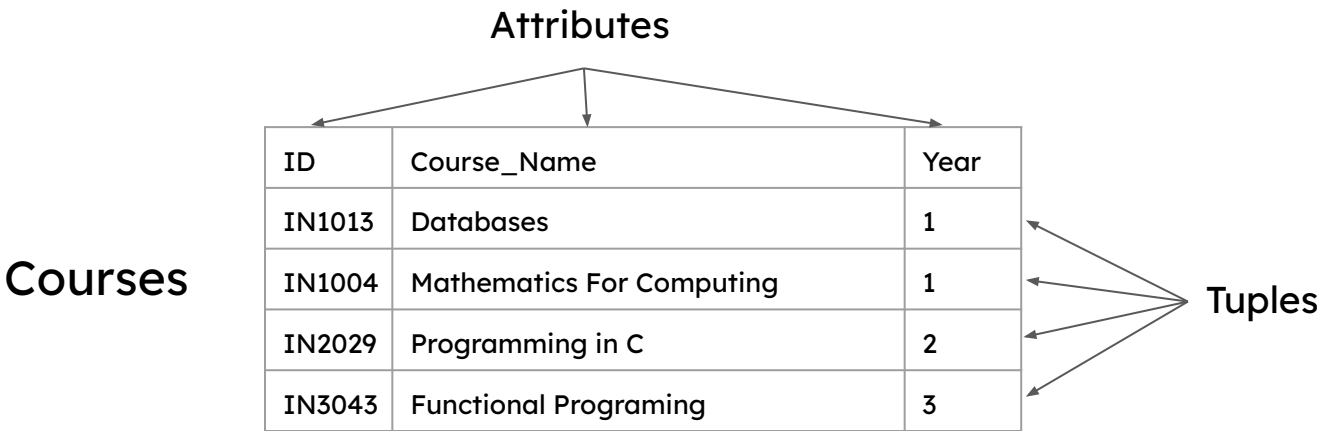
# Relational model

- Data are organized in two-dimensional tables called relations.
- The tables are related to each other.
- The most popular model.



# Relation

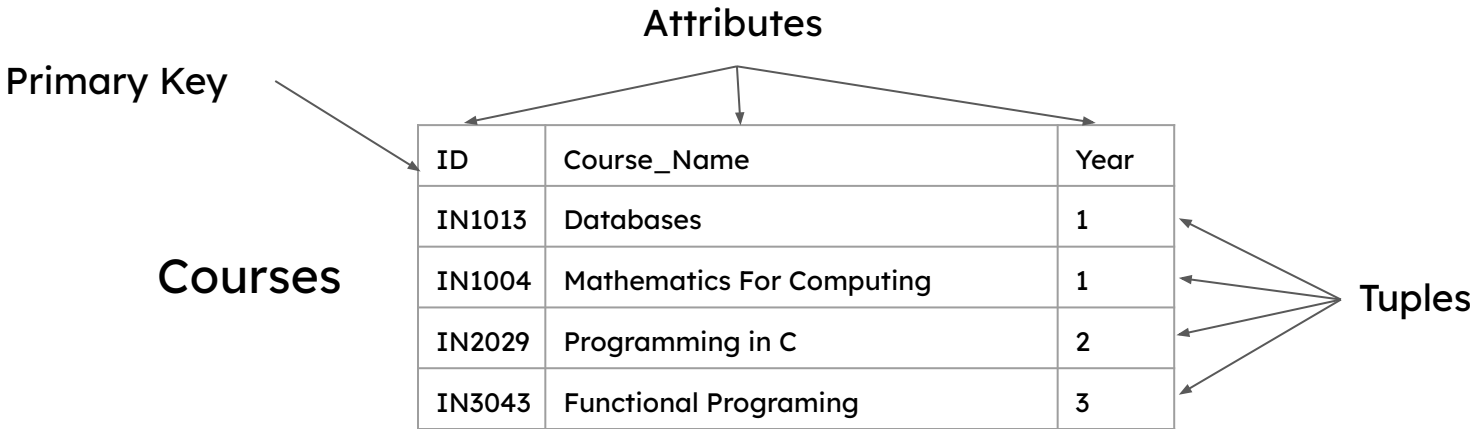
- Name – each relation in a relational database should have a name that is unique among other relations.
- Attribute – each column in a relation.
  - The degree of the relation – the total number of attributes for a relation.
- Tuple – each row in a relation.
  - The cardinality of the relation – the total number of rows in a relation.



# Relational Database

CITY UNIVERSITY LONDON
EST 1894

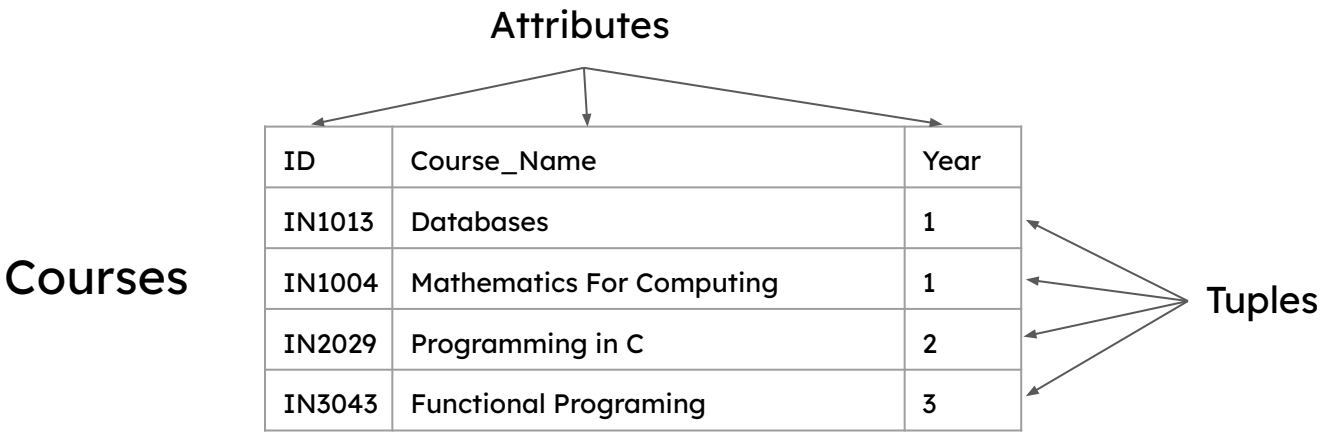
- Definition:
  - Data stored in tables that are associated by shared attributes (keys).
  - Any data element (or entity) can be found in the database through the name of the table, the attribute name, and the value of the primary key.




| ID | Course_Name | Year |
|---|---|---|
| IN1013 | Databases | 1 |
| IN1004 | Mathematics For Computing | 1 |
| IN2029 | Programming in C | 2 |
| IN3043 | Functional Programing | 3 |

# The Relational Model

- Each attribute has a unique name within an entity
- All entries in the column are examples of it
- Each row is unique
- Ordering of rows and columns is unimportant
- Each position (tuple) is limited to a single entry




| ID | Course_Name | Year |
|---|---|---|
| IN1013 | Databases | 1 |
| IN1004 | Mathematics For Computing | 1 |
| IN2029 | Programming in C | 2 |
| IN3043 | Functional Programing | 3 |

# Database Table Example

CITY UNIVERSITY LONDON
EST 1894

## Students

Primary Key Column

Students II : Table

| StudentID | Lastname | Firstname | DOB | Street address | Suburb | Postcode |
|---|---|---|---|---|---|---|
| 89120 | Glasnost | Adam | 1/3/84 | 11 Price Lane | Spline | 2988 |
| 90109 | Dubai | Clarissa | 12/11/82 | 123 Dress Dr | Cranton | 2987 |
| 91082 | Smith | James | 5/9/82 | 27 Lexington Dr | Ester | 2980 |
| 93007 | Smith | James | 5/9/82 | 5/15 Coventry | Spline | 2988 |

Record: 5 of 5

# Database Table Keys

CITY UNIVERSITY LONDON
EST 1894

**Definition:**
A key of a relation is a subset of attributes with the following attributes:

- Unique identification
- Non-redundancy

# Types of Keys

CITY UNIVERSITY LONDON
EST 1894

PRIMARY KEY

- Serves as the row level addressing mechanism in the relational database model.
- It can be formed through the combination of several items.

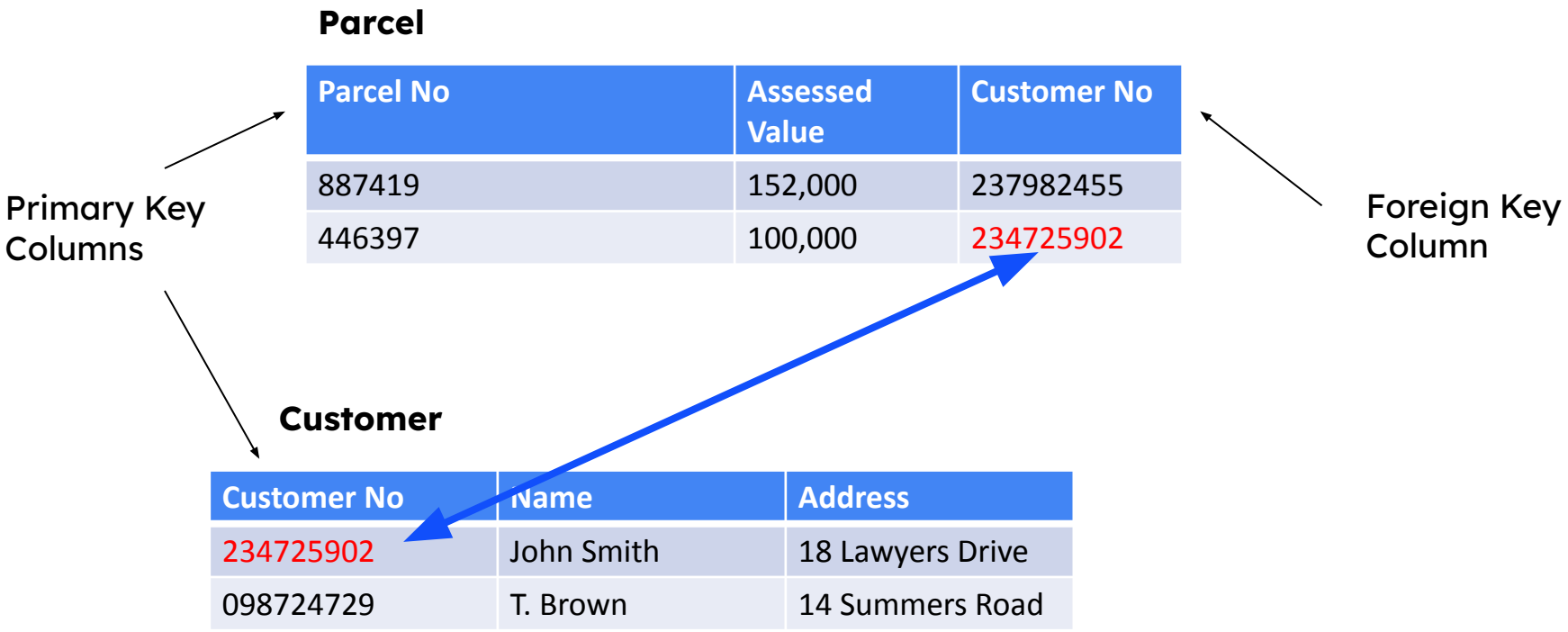
FOREIGN KEY

- A column or set of columns within a table that are required to match those of a primary key of a second table.

These keys are used to form a RELATIONAL JOIN - thereby connecting row to row across the individual tables.

# Relationships and Keys

CITY UNIVERSITY LONDON
EST 1894



# Designing a Relational Database. Part 1.

## Getting started – Entities, attributes and Primary Keys

# Entities


CITY UNIVERSITY LONDON

Entities are the core building blocks of relational database design.

When we implement the database, entities will be mapped to tables

# An Entity

CITY UNIVERSITY LONDON
EST 1894

*A person, place, object, event, or concept about which the organization wishes to maintain data.*

- Must need to store data
- Must have at least two attributes
- Must have at least two records

# Entity Types

CITY UNIVERSITY LONDON
EST 1894

*Classes of people, objects or concepts about which we wish to store data.*

- Become tables in a new computer system
- Instances are rows
- Attributes are columns

# Attribute

CITY UNIVERSITY LONDON
EST 1894

*A description or property of a given entity type.*

- Must depend on the entity key alone
- Must contain information that we explicitly need
- Must have the same data type for all entity occurrences

# Discovering Entities

CITY UNIVERSITY LONDON
EST 1894

- Entities with only one attribute are usually modeled as attributes of another entity.
- Entities that have only one record are usually modeled as a set of parameters and not as files.
- Include only entity types that are needed by a system.  Extra entities require maintenance and space that can add considerably to the cost of a system.

## Converting a text description into an E-R model: Entities


CITY UNIVERSITY LONDON

- Review the conceptual description of the business area for nouns that describe the system
- Each entity type should have more than one potential instance
- Each entity type should have more than one attribute
- Each entity type should be relevant

# Attributes

CITY UNIVERSITY LONDON
EST 1894

- Attributes are properties that describe features of entity types.
- Attributes are usually nouns that describe properties of entity instances (like address for a customer).
- Attributes become fields in a database.

# Keys


CITY UNIVERSITY LONDON

- Candidate keys are any attribute or combination of attributes that uniquely identify a record.  The entire record is a candidate key.
- A Primary Key is one candidate key.  A good primary key is short and does not change over the life of the database.

# Keys

CITY UNIVERSITY LONDON
EST 1894

*Names are normally poor primary keys. They have multiple valid representations. Primary keys:*

1. Should not change values over the life of the instance
2. Should not have null values
3. Should not be "intelligent keys". These are keys that also describe properties of the entity.
4. Should not be large composite keys

# Example

CITY UNIVERSITY LONDON
EST 1894

**Finsbury library** stores data about their book collection, borrowers and loans.

The library can have a number of copies of each book, each copy being identified by its issue_No. The library needs to keep details of the date copies were purchased and the purchase price; purch_date and purch_price. Books are identified by their ISBN and the library also needs to keep details of the Authors name, the Publisher and the book's publication date; author_name, publisher, pub_date.

To help readers the library keeps details of Authors; author_name, date-of-birth, and a short biography.

Borrowers are identified by a borrower_id and details of their name and address are kept.

When a borrower borrows books the date they borrow a book is stored as is the date they return the book.

# Example

CITY UNIVERSITY LONDON
EST 1894

**Finsbury library** stores data about their book collection, borrowers and loans.

The library can have a number of copies of each book, each copy being identified by its issue_No. The library needs to keep details of the date copies were purchased and the purchase price; purch_date and purch_price. Books are identified by their ISBN and the library also needs to keep details of the Authors name, the Publisher and the book's publication date; author_name, publisher, pub_date.

To help readers the library keeps details of Authors; author_name, date-of-birth, and a short biography.

Borrowers are identified by a borrower_id and details of their name and address are kept.

When a borrower borrows books the date they borrow (loan) a book is stored as is the date they return the book.

Candidate entities are shown in red.

# Example

CITY UNIVERSITY LONDON
EST 1894

**Finsbury library** stores data about their book collection, borrowers and loans.

The library can have a number of copies of each book, each copy being identified by its issue_No. The library needs to keep details of the date copies were purchased and the purchase price; purch_date and purch_price. Books are identified by their ISBN and the library also needs to keep details of the Authors name, the Publisher and the book's publication date; author_name, publisher, pub_date.

To help readers the library keeps details of Authors; author_name, date-of-birth, and a short biography.

Borrowers are identified by a borrower_id and details of their name and address are kept.

When a borrower borrows books the date they borrow (loan) a book is stored as is the date they return the book.

Candidate entities are shown in red, candidate keys are in blue.

# Tutorial

CITY UNIVERSITY LONDON
EST 1894

- **Going over an example**
- **Creating a database scenario**
  - First step in the coursework
- **Installing MySQL**

- **Homework!**