Xi'an Jiaotong-Liverpool University
西交利物浦大學

Introduction to Database

Welcome to CPT103!

Table of Contents

- Module information
- Introduction to database
- Relational Model
- Relational Keys

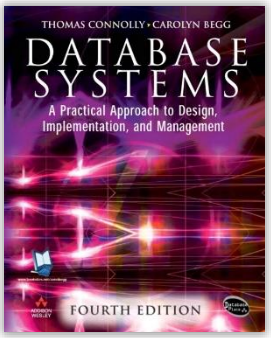
Module Information

About the module instructor, teaching organisation

Teaching Organisation

- Module instructors:

Jianjun Chen	(Jianjun.Chen@xjtlu.edu.cn)	SD541
Jun Qi	(Jun.Qi@xjtlu.edu.cn)	SD461
Yu Liu	(Yu.Liu02@xjtlu.edu.cn)	SD465


THOMAS CONNOLLY • CAROLYN BEGG
DATABASE SYSTEMS
A Practical Approach to Design, Implementation, and Management
FOURTH EDITION

  - Office hours: please check the module page online

- Textbook: “Database Systems: a practical approach to design, implementation, and management”
  by Connolly, Thomas M., Begg, Carolyn E.

Teaching Organisation

- Support:
  - Discussions with classmates enable you to better understand concepts and terminologies.
  - One or more noticeboards will be added on the LearningMall about issues (like coursework, exam, labs) related to this module
  - I will also send notifications to you about any updates. Please check emails frequently.
- Assessments: Please check the information about coursework and exams on e-bridge.
  - You are encouraged to discuss ideas (Not solutions!) with others when doing coursework.
  - But your assignment submissions must be your own works.

What Do You Need?

- Lectures:
  - A piece of note that covers all important knowledge.
  - Laptop or tablets: Some of the questions are on the LearningMall.
  - A friend that you can discuss with during breaks.

- Labs:
  - Someone to discuss with.
  - Questions you want to ask to me.

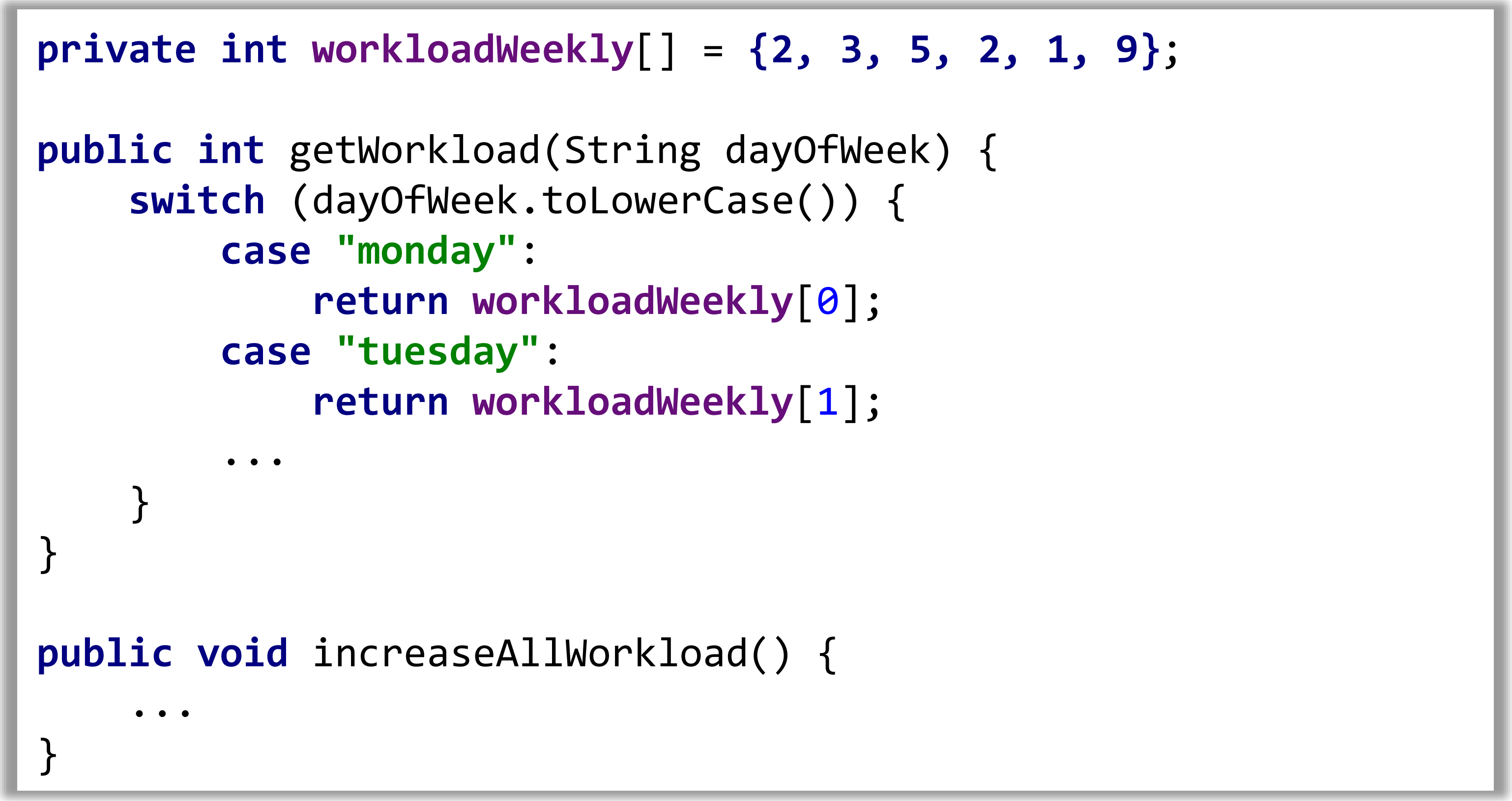
- Coursework:
  - Don’t forget your Java language!

Introduction to Database Systems

What is a database? What is a database management system?

What is Data?

- Example 1: An array that stores some numbers, which can be retrieved sometime later.



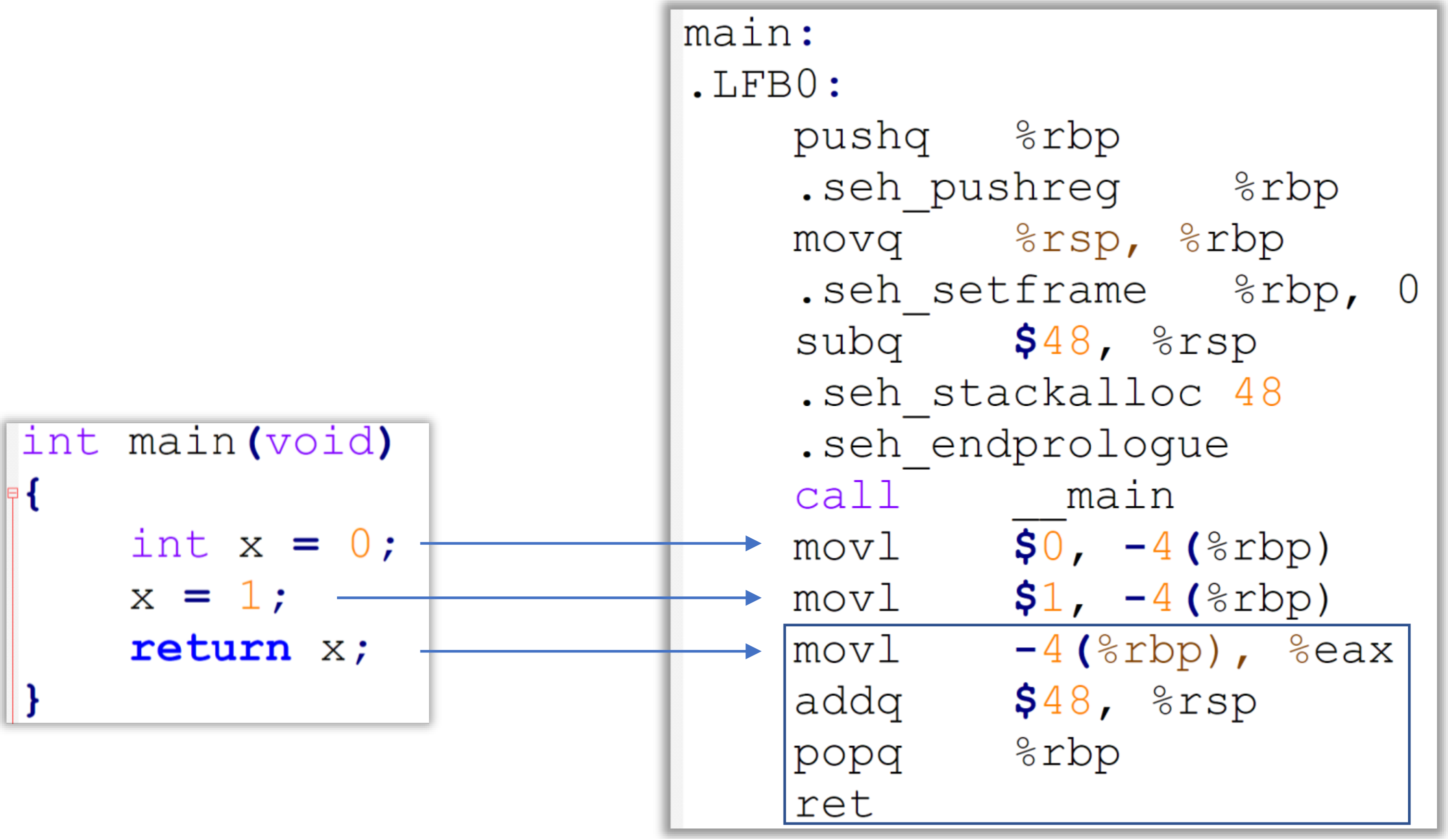
```
private int workloadWeekly[] = {2, 3, 5, 2, 1, 9};

public int getWorkload(String dayOfWeek) {
    switch (dayOfWeek.toLowerCase()) {
        case "monday":
            return workloadWeekly[0];
        case "tuesday":
            return workloadWeekly[1];
        ...
    }
}

public void increaseAllWorkload() {
    ...
}
```

What is Data?

- Example 2: A piece of data inside a CPU register.



```c
int main(void)
{
    int x = 0;
    x = 1;
    return x;
}
```

```
main:
.LFB0:
    pushq   %rbp
    .seh_pushreg    %rbp
    movq    %rsp, %rbp
    .seh_setframe   %rbp, 0
    subq    $48, %rsp
    .seh_stackalloc 48
    .seh_endprologue
    call    __main
    movl    $0, -4(%rbp)
    movl    $1, -4(%rbp)
    movl    -4(%rbp), %eax
    addq    $48, %rsp
    popq    %rbp
    ret
```

"-4(%rbp)" is where the variable x is stored

What is Data?

- Data is only meaningful under its designed scenario.
  - **In example 1:** The array `workloadWeekly` is private, thus cannot be used outside of its class.
  - **In example 2:** The data stored in **“-4(%rbp)”** is invalid once the function returns.
- Must have ways to create/modify data.
- Must have ways to access data.

What is Database?

- **Database**: "Organised collection of data. Structured, arranged for ease and speed of search and retrieval."
- **Database Management System (DBMS)**: Software that is designed to enable users and programs to store, retrieve and update data from database.
  - A software must have a set of standard functions to be called DBMS. We will learn about these functions soon!

What is Database?

> “Organised collection of data. Structured, arranged for ease and speed of search and retrieval.”

- The structure is presented to users as tables with names
  - Example 1: Member cards of a chain store

Phone No.	Name	Points
233333	Vincent	1000
233334	Matt	1231

  - Example 2: Banking service, account balance

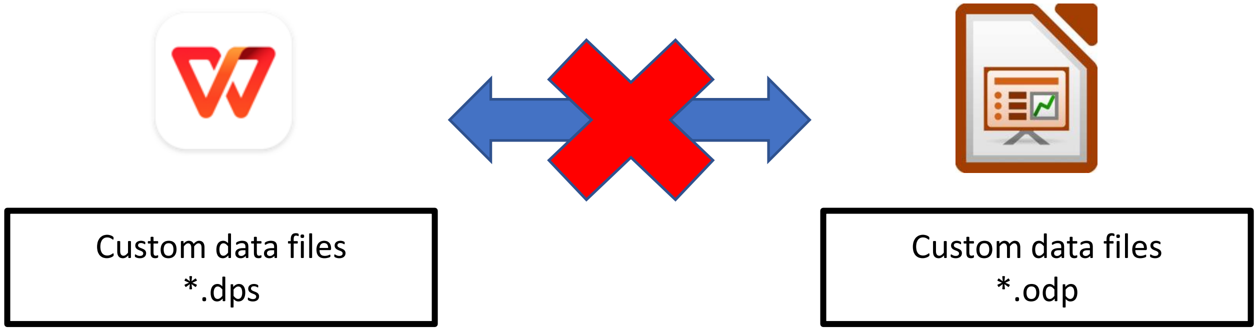
Card ID	Holder ID	Name	Balance
0933 1223 0001 4321	12360	Daryl XXXX	-50
0963 1245 0291 0177	78799	Jessie XXXX	233333

Why Database?

- We will use WPS office, Microsoft office and LibreOffice as an example.
  - PowerPoint uses pptx/ppt as the default format.
  - LibreOffice uses odp as the default format.
  - WPS office supports pptx/ppt and its own format: dps

Pre-DBMS Methods

- Applications store data as files.
  - Each application uses its own format.
- Other applications need to understand that specific format.
  - Leads to duplicated code and wasted effort.
  - Compatibility issues.



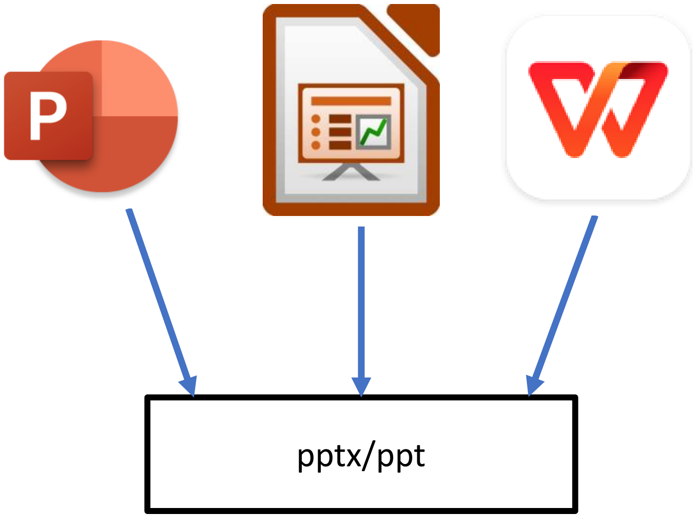
Custom data files
*.dps

Custom data files
*.odp

Pre-DBMS Methods

How about using a common data format?

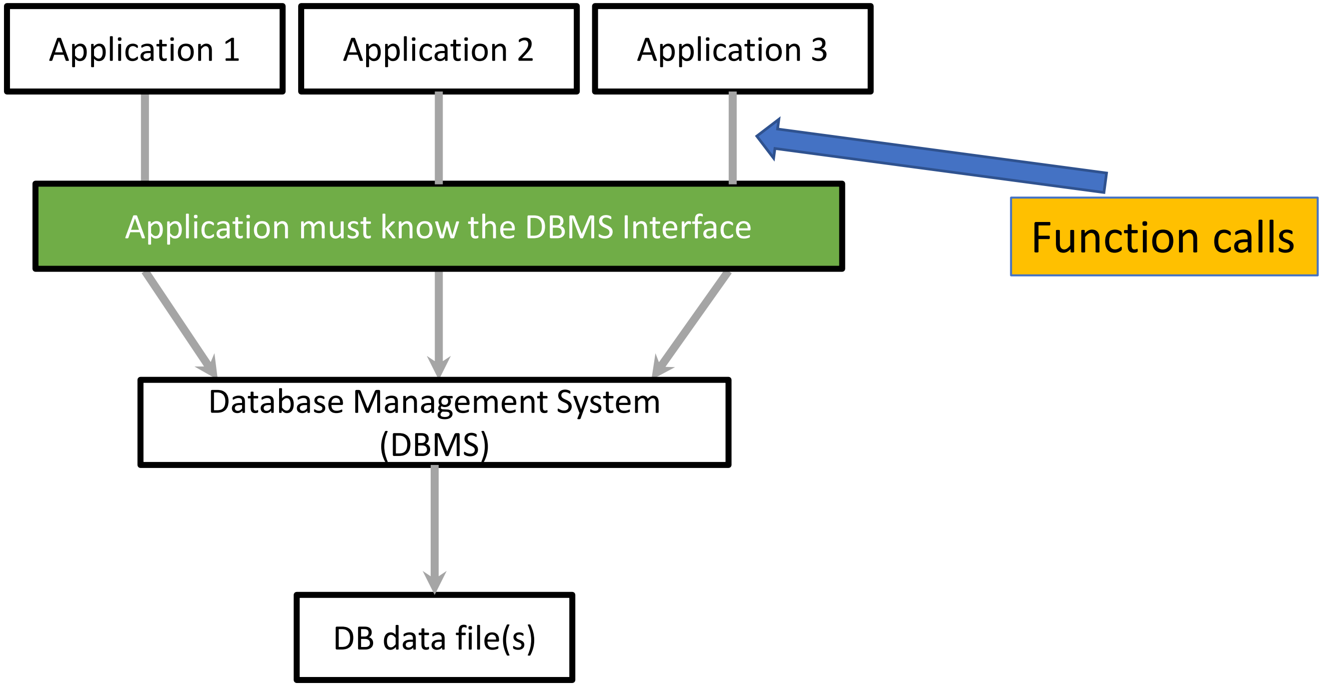
- Still need to write duplicated code for reading this file format.
- Synchronisation issues: Accessed simultaneously?
  - Very hard to coordinate operations from different apps.
- Compatibility issues.



pptx/ppt

DBMS Approach

- Work as a delegate for this common collection of data.
- Applications use a common API for accessing database.
  - The implementation of API is provided by database software companies.
  - All database commands are standardised (SQL language).


Application 1
Application 2
Application 3
Function calls
Application must know the DBMS Interface
Database Management System (DBMS)
DB data file(s)

Commonly Seen DBMS

- Oracle
- DB2
- MySQL
  - MariaDB
- Ingres
- PostgreSQL
- Microsoft SQL Server
- MS Access

DBMS Functions / Must Haves

- Allow users to store, retrieve and update data
- Ensure either that all the updates corresponding to a given action are made or that none of them is made (Atomicity)
- Ensure that DB is updated correctly when multiple users are updating it concurrently
- Recover the DB in the event it is damaged in any way
- Ensure that only authorised users can access the DB
- Be capable of integrating with other software

The Relational Model

And the relational database management systems (RDBMS)

The Relational Model

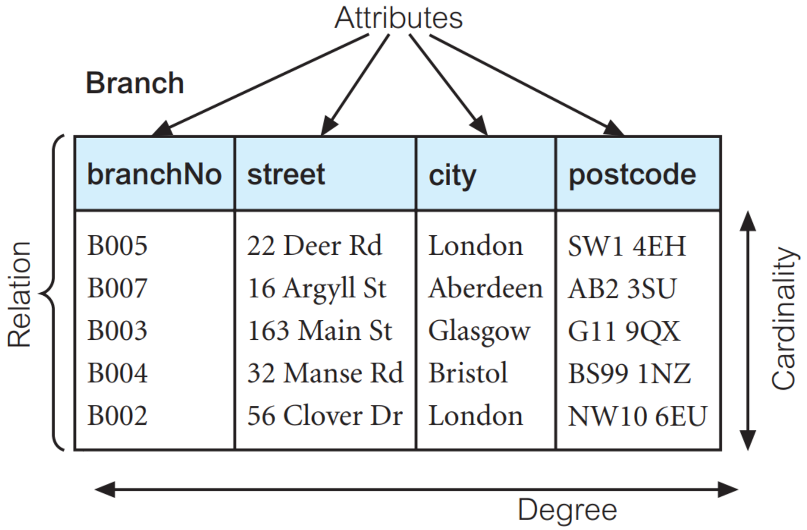
- The relational model is **one approach** to managing data. Originally Introduced by E.F. Codd in his paper “A Relational Model of Data for Large Shared Databanks”, 1970.
  - An earlier model is called the navigational model (https://en.wikipedia.org/wiki/Navigational_database).
- The model uses a structure and language that is consistent with first-order predicate logic
  - Provides a declarative method for specifying data and queries
  - Details are covered in the Chapter 4 of the textbook.
- Relational database management systems (RDBMS) are based on the relational model.
  - Many relational operations are supported.
  - Relational algebra!

Terminologies

- A <u>relation</u> is a mathematical concept. The physical form of a relation is a table with columns and rows.

- An <u>attribute</u> is a named column of a relation.

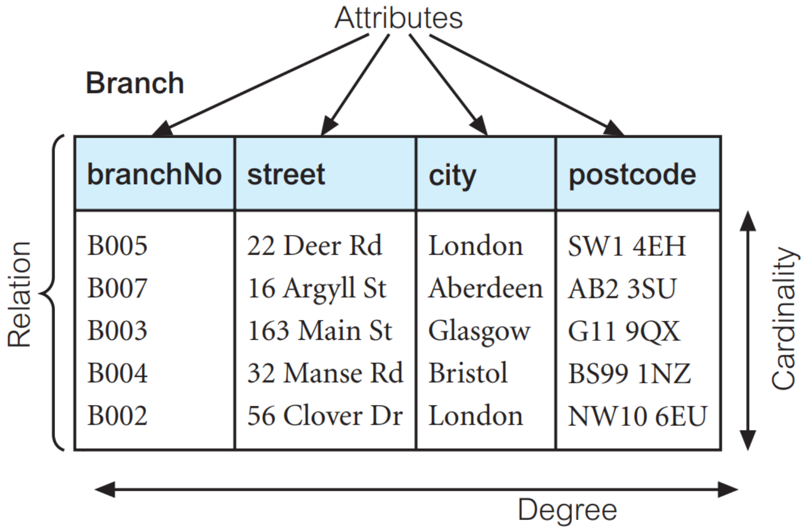
- A <u>domain</u> is the set of allowable values for attributes.
  - Age must be a positive integer.
  - Postcodes have length limit.



branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Terminologies

- Tuple: a tuple is a row of a relation.
  - Mathematically, the order of tuples does not matter.
- The degree of a relation is the number of attributes it contains.
- Cardinality: the number of tuples in a relation.



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Terminologies

- Relation schema: The definition of a relation, which contains the name and domain of each attribute.
  - **Formally** (See Chapter 4.2.3): “A named relation defined by a set of attribute and domain name pairs”

  Table: branch

branchNO	Character: size 4, range B001-B999
street	Character: size 25
city	Character: size 15
postcode	Character: size 8

- Relational database schema:
  - A set of relation schemas, each with a distinct name.
  - Could be understood as a set of table definitions like the above example

Alternative Terminologies

Formal Terms	Alternative #1	Alternative #2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

Question

- Can you point out what the previous terminologies refer to in this table?

Staff

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts

Relation schema:
relation_name(ID: Char, Name: Char, Salary: Monetary, Department: Char)

Attributes are: ID, Name, Salary & Department

The degree of the relation is 4

Staff

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts

Tuples, e.g.
{(ID, A368),
(Name, Jane Brown),
(Salary, 22,000),
(Department, Accounts)}

The cardinality of the relation is 5

Additional Properties of Relations

- Relation’s name is unique in the relational database schema.
- Each cell contains exactly one atomic value.
- Each attribute of a relation must have a distinct name.
- The values of an attribute are from the same domain.
- The order of attributes has no significance.
- The order of tuples has no significance.
- No duplicate tuples

Relational Keys

Super key, candidate key, primary key, foreign key.

Question

- Assume each person’s id is unique.
- Assume that the whole relation is stored as a two-dimensional array, and you want to look for the “Maria” whose age is 22.
- What problem does the relation on the right have?
  - How many rows do you need to check?
- What can be done to improve the efficiency of this search and prevent this from happening?

ID	Name	Age
1	Andrew	34
1	Andrew	34
1	Andrew	34
2	Erick	32
2	Erick	32
3	Thomas	28
4	Paul	33
6	Rodrick	47
7	Maria	55
8	Maria	22

Primary Key

- It is beneficial to let a program to automatically check for and **reject** duplicate values in one or more columns for you when tuples are added.
- This can be done in database systems, by applying a constraint (consider it as a label) called Primary key on the columns of a table.
- Single-column primary key example (Staff table):

ID	Name	Age
1	Jason	12

- Multi-column primary key example (Company with several buildings):

Building Number	Room	Room Size	Has Printer
11	301	96	True

Primary Key

- What will happen if the primary key constraints are applied to:
  - (Name)
  - (Name, Age)

ID	Name	Age
1	Jason	12

staff information table

  - (Room Size)
  - (Room)

Building Number	Room	Room Size	Has Printer
11	301	96	True

building room information table

- Good decisions?

Primary Key: Important Properties

- Columns constrained by a primary key uniquely identifies tuples in a table.
  - For each tuple, the id is always different.
  - This is a core functionality of primary key.
- Each table can only have one primary key.
- `NULL` values are not allowed if a primary key is present.
  - `NULL` will be taught in later weeks.

ID	Name	Age
1	Andrew	34
2	Erick	32
3	Thomas	28
4	Paul	33
6	Rodrick	47
7	Maria	55
8	Maria	22

Primary Key and Entity Integrity

- Primary Key enforces **entity integrity**.
- It helps to maintain the consistency and accuracy of data in a database by preventing duplicate records and ensuring that each record can be uniquely identified.
- Particularly important in applications that rely on a high degree of data integrity:
  - Financial systems
  - Healthcare applications
  - Other mission-critical systems.

Question 2

- If we apply the primary key constraint on (ID, Name).
  - Does (ID, Name) uniquely identifies each tuple in this relation correctly?
  - How will the database program check duplicate values when tuples are inserted?
    - Insert (9, 'Jason', 12) as an example.
- Is this primary key a good idea?

ID	Name	Age
1	Andrew	34
2	Erick	32
3	Thomas	28
4	Paul	33
6	Rodrick	47
7	Maria	55
8	Maria	22

Super Key

- The primary key choice in the previous example is called a super key.
- Super key: using more than enough columns to uniquely identify tuples in a table.
  - In this table, only the constraint (ID) is a primary key.
  - (ID, Name), (ID, Age), (ID, Name, Age) are super keys.
  - (Name), (Name, Age) are bad choices of primary keys. (why bad?)

ID	Name	Age
1	Andrew	34
…	…	…

- Super key is taught so that you can avoid them when choosing the columns to be applied with primary keys.

Important Note

- When I explained how databases check for duplicate values, it was in an linear way. In reality, the check will be faster.

- But even if it is faster, super keys are still bad for performance as more comparisons are needed when checking for duplicate values or looking for a certain value.

- You need a good understanding of data structure to understand the underlying mechanism.
  - If you want to do some research by yourself, start by searching “B-Tree + Primary key”.

Question 3

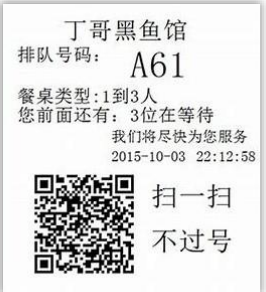
- How many ways you can apply a primary key to the table below?
  - Super keys are not allowed.
  - Bad primary keys that do not work as intended are also unallowed.

- The table below stores staff information.

StaffID	Email	First name	Last name	Passport ID
1	S.Guan@xjtlu.edu.cn	Steven	Guan	P123456
2	J.Woodward@nott.ac.uk	John	Woodward	U543121
3	N.Tubb@bhan.ac.uk	Nathan	Tubb	U998877

Candidate Key

- All these possible primary keys are called Candidate keys.
  - The primay key is just a candidate key chosen by the table designer.
  - There's no definite way to determine which candidate key should be a primary key.
- You can't necessarily infer the candidate keys based solely on the data in your table
  - More often than not, an instance of a relation will only hold a small subset of all the possible values
  - E.g. Restaurants' booking number might reset to 1 after a large number.



丁哥黑鱼馆

排队号码： A61

餐桌类型:1到3人

您前面还有：3位在等待

我们将尽快为您服务

2015-10-03 22:12:58

扫一扫

不过号

Queue No. A31

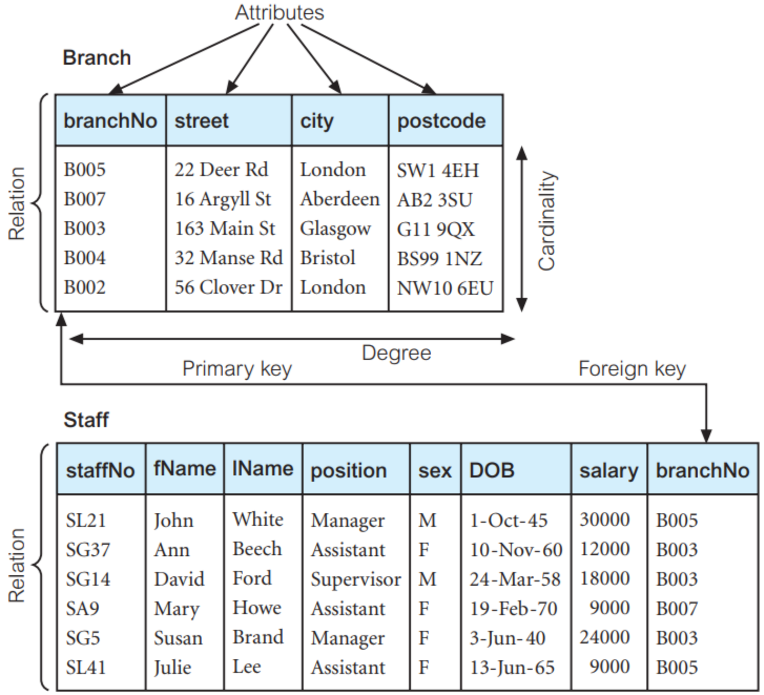
Table size: up to 4

31 People are waiting ahead of you.

A1, A2, A3… A99, A999 -> A1

Foreign Key

- It is also very common that tuples in one relation references data from another relation.
    - As a result, a database should provide such mechanism to ensure correct references.
- This is enforced by something called **foreign key**



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

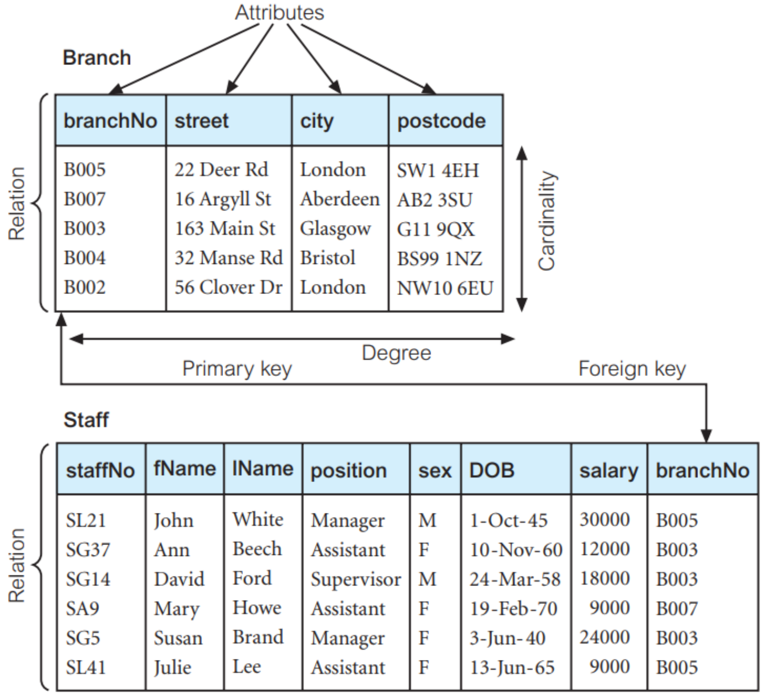
Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Foreign Key

- **Foreign key**:
  - One or more attributes within one relation that **must match** the candidate key of some (possibly the same) relation.

- Example:
  - We want the values of the ‘branchNo’ in relation staff to be one of the ‘branchNo’ in relation Branch.



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

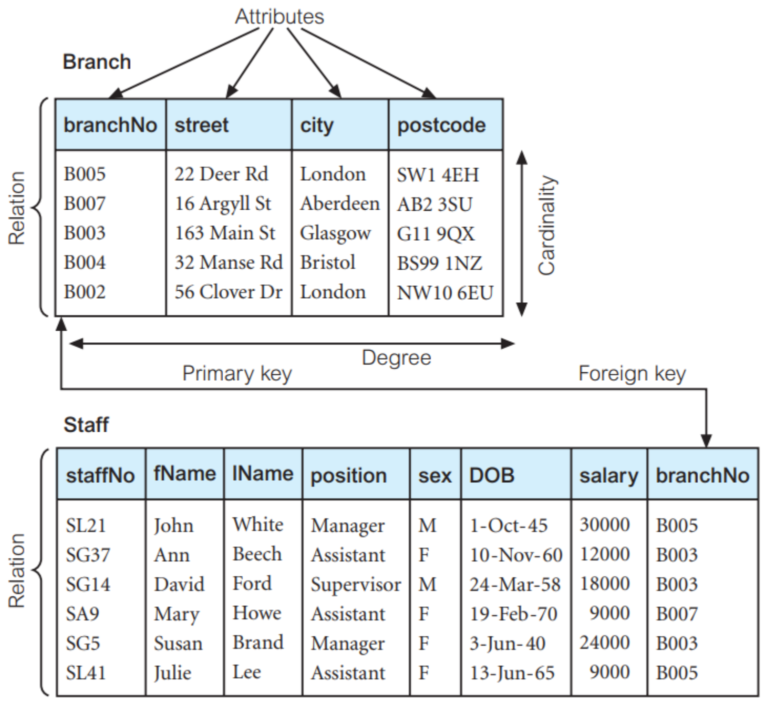
Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Question

- What criteria must be met before a foreign key can work?

- Use the example on the right:
  - if you are to program a database software.
  - need to check whether branch numbers in the Staff table match branch numbers in the Branch table.



Branch

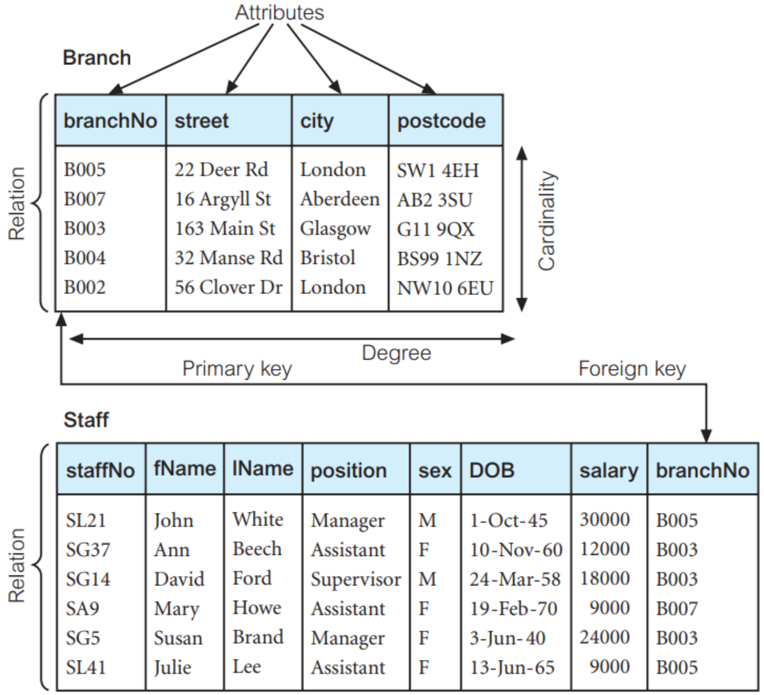
branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Foreign Key

- Data type should be the same.
  - In real databases, sometimes this can be violated.
  - Different data type is not recommended.
- The referenced column must be a candidate key of that table.
  - [Some interesting discussions here](https://stackoverflow.com/questions/8706073/does-foreign-key-always-refe): https://stackoverflow.com/questions/8706073/does-foreign-key-always-refe



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

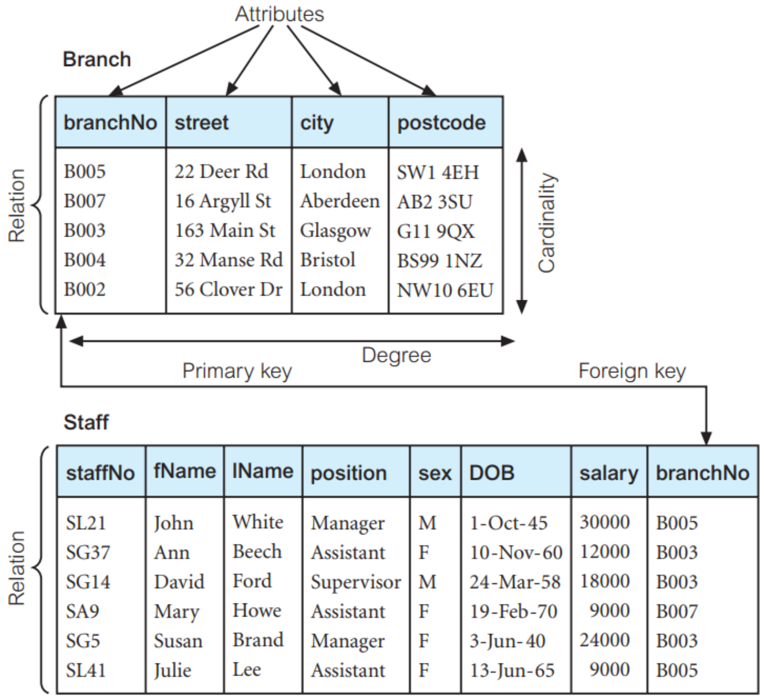
FK and Referential Integrity

- Foreign Key enforces **referential integrity**
  - It ensures that all data in a database remains consistent and up to date.
  - It helps to prevent incorrect records from being added, deleted, or modified.
- Why they are important? Read https://www.techwalla.com/articles/why-are-entity-integrity-referential-integrity-important-in-a-database.
  - You need more lectures to understand that, though.

A Small Challenge

- Assume that the two tables on the right are stored in 2-dimensional arrays:
  - `String[][] Branch`
  - `String[][] Staff`

- How would you find out the postcode of the branch where Julie Lee works in? (In Java code)



Branch

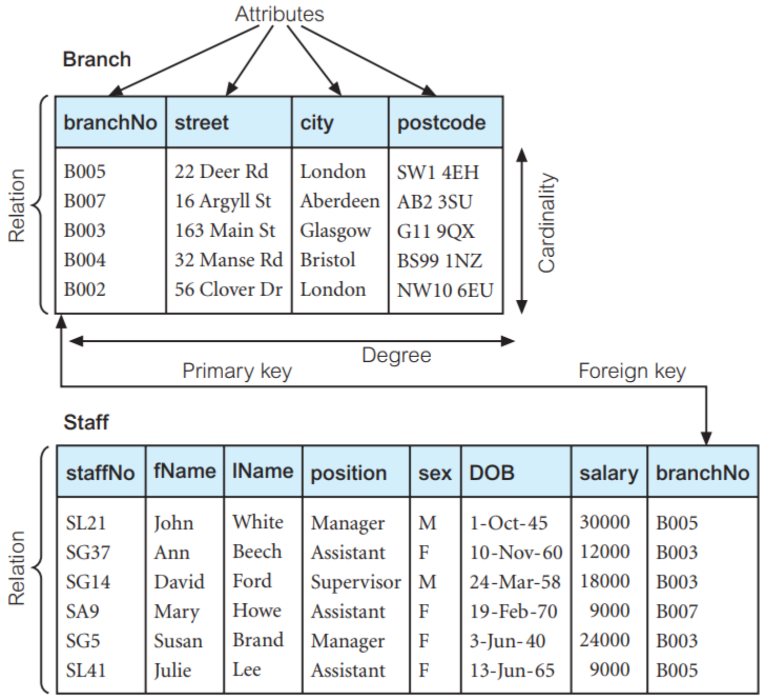
branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

A Small Challenge

- Now, write a function that allows the caller to find out the branch information of any Staff.
- Doing so really helps understand future contents.



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

```
String find(staffID, branchAttributeName)
```