Introduction to Database

AY2019-20, Spring semester

COMP1041: Database and Interface

Week 3

Contents

- Database introduction
- Relational model
- Relational keys and constraints

What is Data?

- A sequence of bits in CPU register.
  - Used by CPU instructions to carry out calculations.
  - `lw $s0, 12($fp)`
  - `addi $s0, $s1, 1`
  - `sw $s0, 12($fp)`
- An array of integers/float…
  - Used by functions to support the purpose of these functions.
  - `ids[0] = prompt("Input your ID","001");`
  - `alert("added id " + ids[0]);`

What is Data?

We can conclude that:

- Data is only meaningful under its designed scenario.
  - E.g. An array created for storing user input is only used by functions that record and process user input.
- Must have ways to create/modify data.
- Must have ways to access data.

Database and Database Management System

Database:

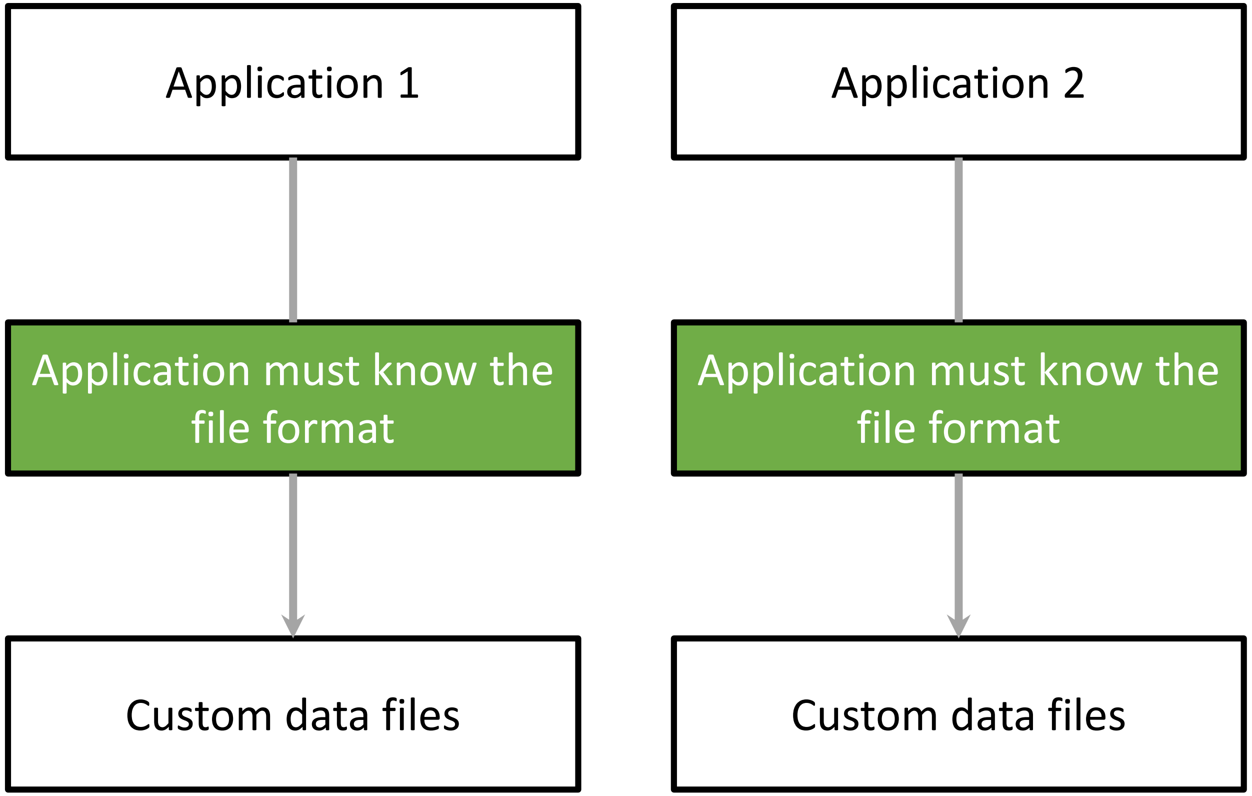
Organised collection of data. Structured, arranged for ease and speed of search and retrieval.

Database Management System (DBMS):

- Software that is designed to enable users and programs to store, retrieve and update data.
- A software must have a set of standard functions to be called DBMS.
    - Otherwise excel + excel files can also be called DBMS.
- But how did the management of data evolve?

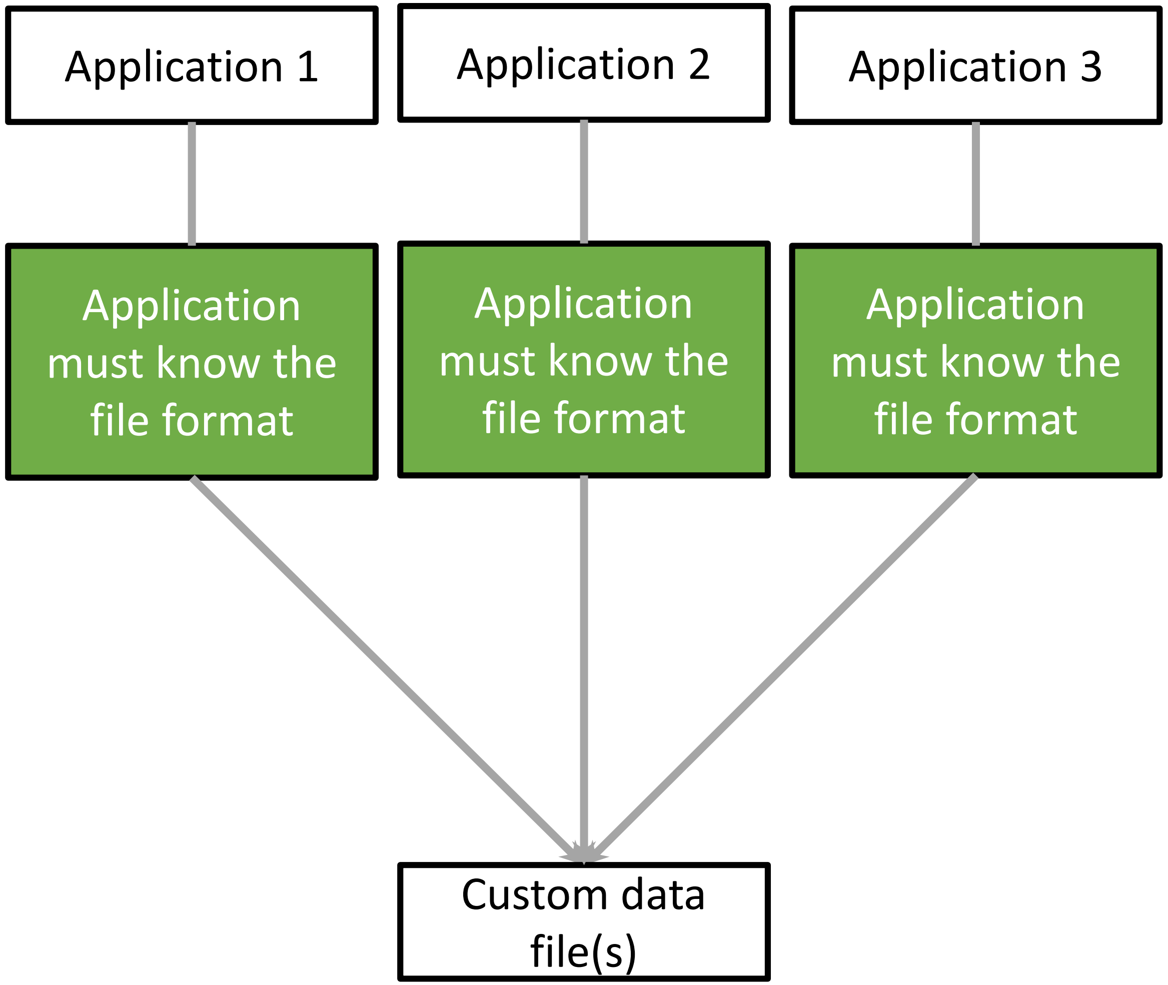
Pre-DBMS Methods

- 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.
  - MS Word -> ".docx" files -> LibreOffice -> rendering inconsistency.


Application 1
Application 2
Application must know the file format
Application must know the file format
Custom data files
Custom data files

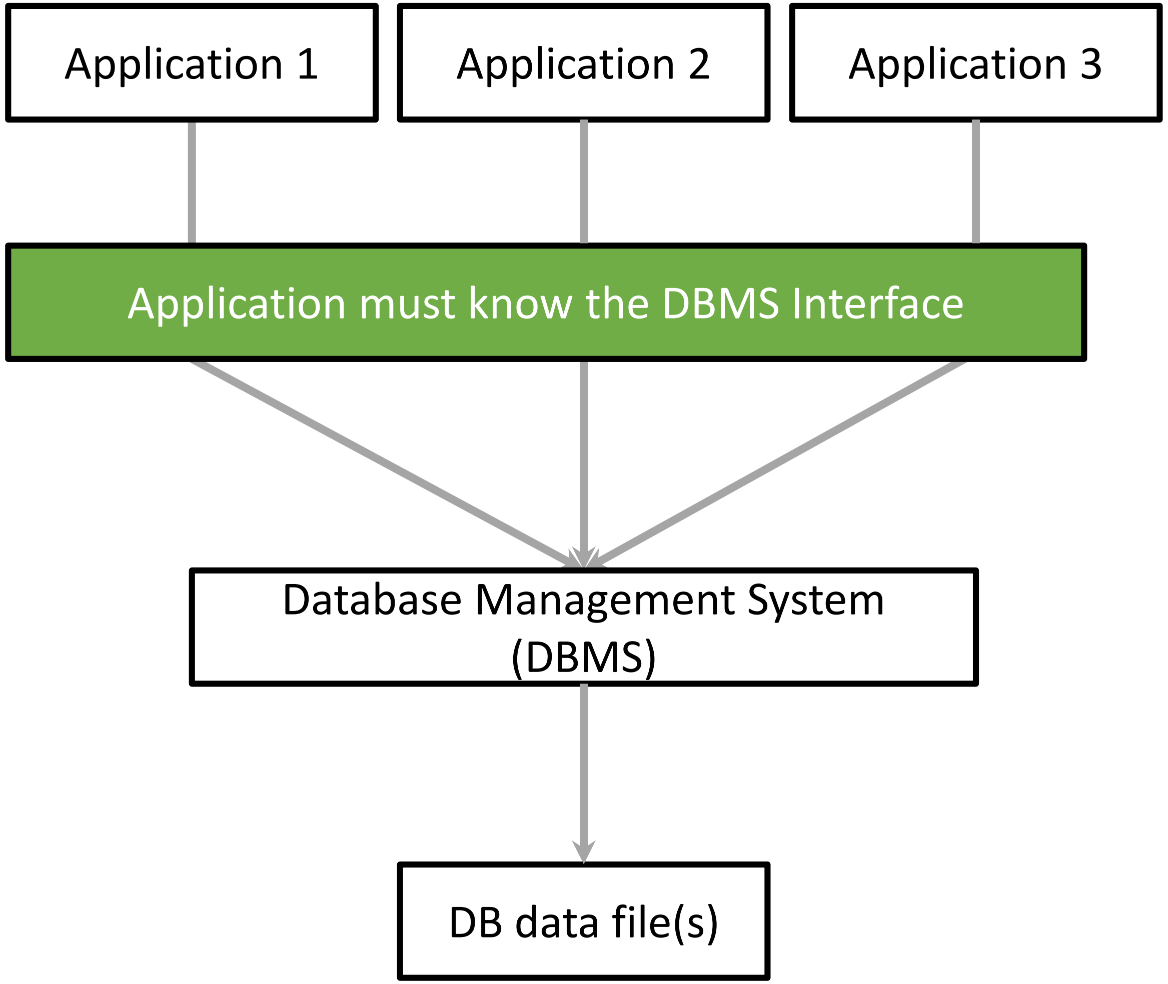
Pre-DBMS Methods

- How about 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.


Application 1
Application 2
Application 3
Application must know the file format
Application must know the file format
Application must know the file format
Custom data file(s)

DBMS Approach

- Work as a delegate for this common collection of data.
- Applications use a common API for database.
  - A header file in C, plus the dynamic/static library file.
  - A Java Interface, plus a set of class files or a single ".jar" file


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

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

Cases of Using Databases

- Chain store: member card.

Phone No.	Name	Points
233333	Vincent	1000
233334	Matt	1231

- Bank service: account balance.

Card ID	Holder ID	Name	Balance
0933 1223 0001 4321	12360	Daryl XXXX	-50
0963 1245 0291 0177	78799	Jessie XXXX	1000000000

Commonly Seen DBMS

- Oracle
- DB2
- MySQL
- Microsoft SQL Server, Ingres, PostgreSQL, MS Access, etc.
- Mostly Relational DB

- Hive
- MangoDB
- Cassandra
- Neo4J
- And many more…
- Big data DBs, NoSQL.
- A new module offered at Year 4 in UNNC themed “Big Data” taught by Dr. Saeid. Stay Tuned!

Relational Model

The Relational Model

- The relational model is an approach to managing data
  - An earlier model is called navigational model.
- Originally Introduced by E.F. Codd in his paper
  - "A Relational Model of Data for Large Shared Databanks", 1970
- The model uses a structure and language that is consistent with
  - First-order predicate logic
- Provides a declarative method for specifying data and queries
- Chapter 4 of the DB book.

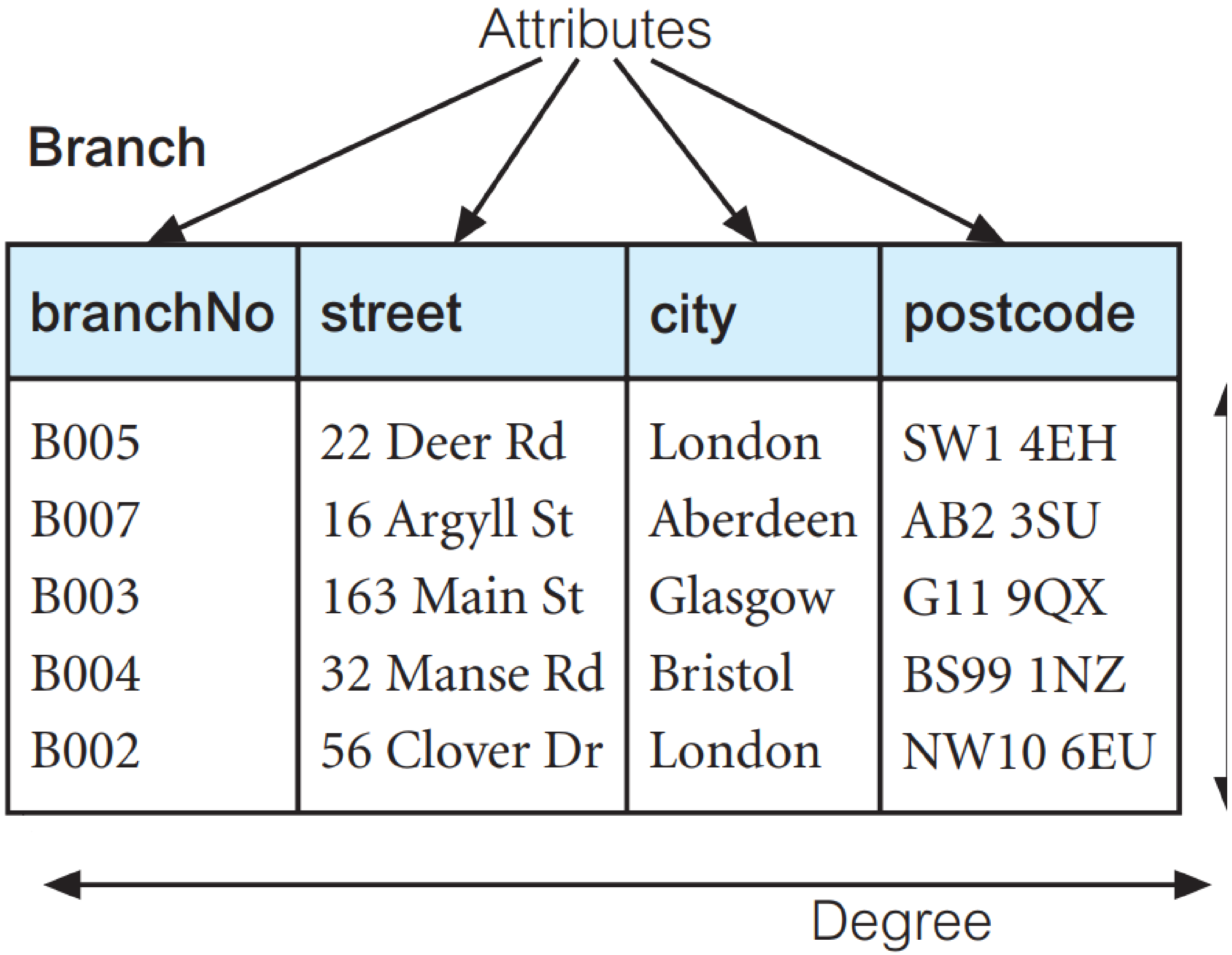
Relational Model: Terminologies

- “Designed usage scenario” of relational database:
  - Dealing with information that can be expressed as tables.

- A relation is a mathematical concept. (see Chapter 4.2.2)
- The physical form of a **relation** is a table with columns and rows.
- An **attribute** is a named column of a relation.
- A **domain** is the set of allowable values for attributes. (value constraints)
  - Age must be positive integers
  - Postcodes have length limit

Terminologies

- Attribute
- Domain
- **Tuple**: a tuple is a row of a relation. (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.



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”
- **Relational database schema:**
  - A set of relation schemas, each with a distinct name

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

Alternative Terminology

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

Table 4.1 in the textbook

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

The cardinality of the relation is 5

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) }

Relation: Properties

- A relation has the following properties:
  - Its name is unique in the relational database scheme.
  - Each cell contains exactly one atomic value.
  - Each attribute has a distinct name.
  - The values of an attribute are from the same domain.
  - No duplicate tuples.
  - The order of attributes has no significance.
  - The order of tuples has no significance.

Relational Keys

Relational Keys: Super Key

- **Super key**: one or more attributes that uniquely identifies a tuple within a relation.

Passport ID	Student ID	Name

- {Passport ID}
- {Passport ID, Name}
- {Student ID}
- {Student ID, Name}
- {Passport ID, Student ID}
- {Passport ID, Student ID, Name}

{Name} alone does not uniquely identify a tuple. Because there are people with the same name.

Relational Keys: Candidate Key & Primary Key

- **Candidate key**: a super key such that no proper subset is a super key within the relation
  - Every tuple has a unique value for that set of attributes: uniqueness
  - No proper subset of the set has the uniqueness property: minimality
- **Primary key**: The candidate key that is selected to identify tuples uniquely within the relation.

Available choices:

- {Passport ID}
- ~~{Passport ID, Name}~~
- {Student ID}
- ~~{Student ID, Name}~~
- ~~{Passport ID, Student ID}~~
- ~~{Passport ID, Student ID, Name}~~

Passport ID	Student ID	Name

Choosing Candidate Keys

- 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
  - Resturant booking number is reset to 1 after a larger number.
- You must use knowledge of the real-world to help.

Choosing Candidate Keys

- What are the candidate keys of the following relation?

OfficeID	Name	Country	Postcode	Phone
O1001	Headquarters	UK	W1 1AA	0044 20 1545 3241
O1002	R&D Labs	UK	W1 1AA	0044 20 1545 4984
O1003	US West	USA	94130	001 415 665981
O1004	US East	USA	10201	001 212 448731
O1005	Telemarketing	UK	NE5 2GE	0044 1909 559862
O1006	Telemarketing	USA	84754	001 385 994763

The candidate keys are : {OfficeID}, {Phone}, {Name, Postcode/Zip}, {Name, Country}

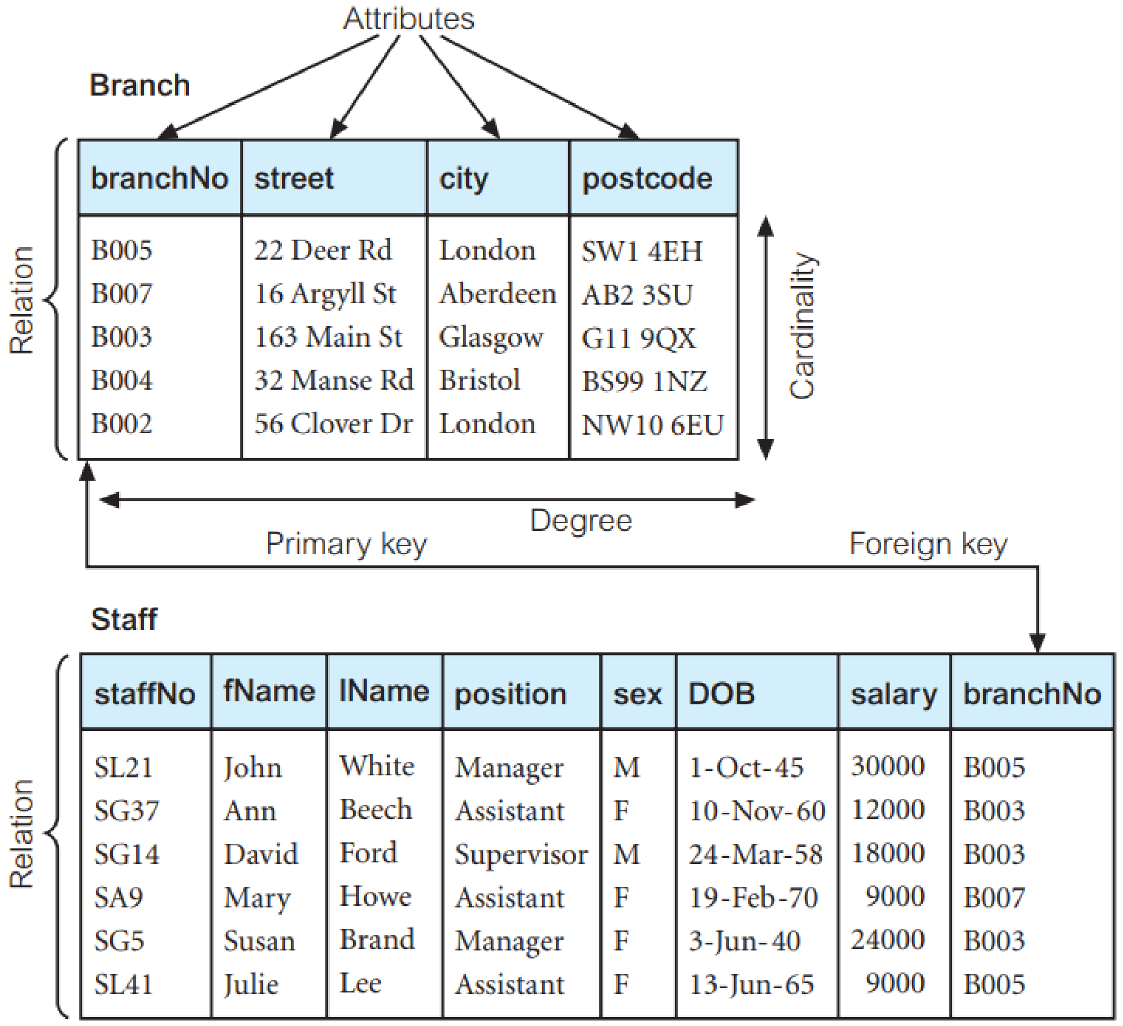
- Note: Keys like {Name, Postcode/Zip, Phone} satisfy uniqueness, but not minimality

OfficeID	Name	Country	Postcode	Phone
O1001	Headquarters	UK	W1 1AA	0044 20 1545 3241
O1002	R&D Labs	UK	W1 1AA	0044 20 1545 4984
O1003	US West	USA	94130	001 415 665981
O1004	US East	USA	10201	001 212 448731
O1005	Telemarketing	UK	NE5 2GE	0044 1909 559862
O1006	Telemarketing	USA	84754	001 385 994763

Relational Keys: Foreign Key

- Foreign key:
  - One or more attributes within one relation that matches the candidate key of some relation.

- In the example on the right:
  - 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

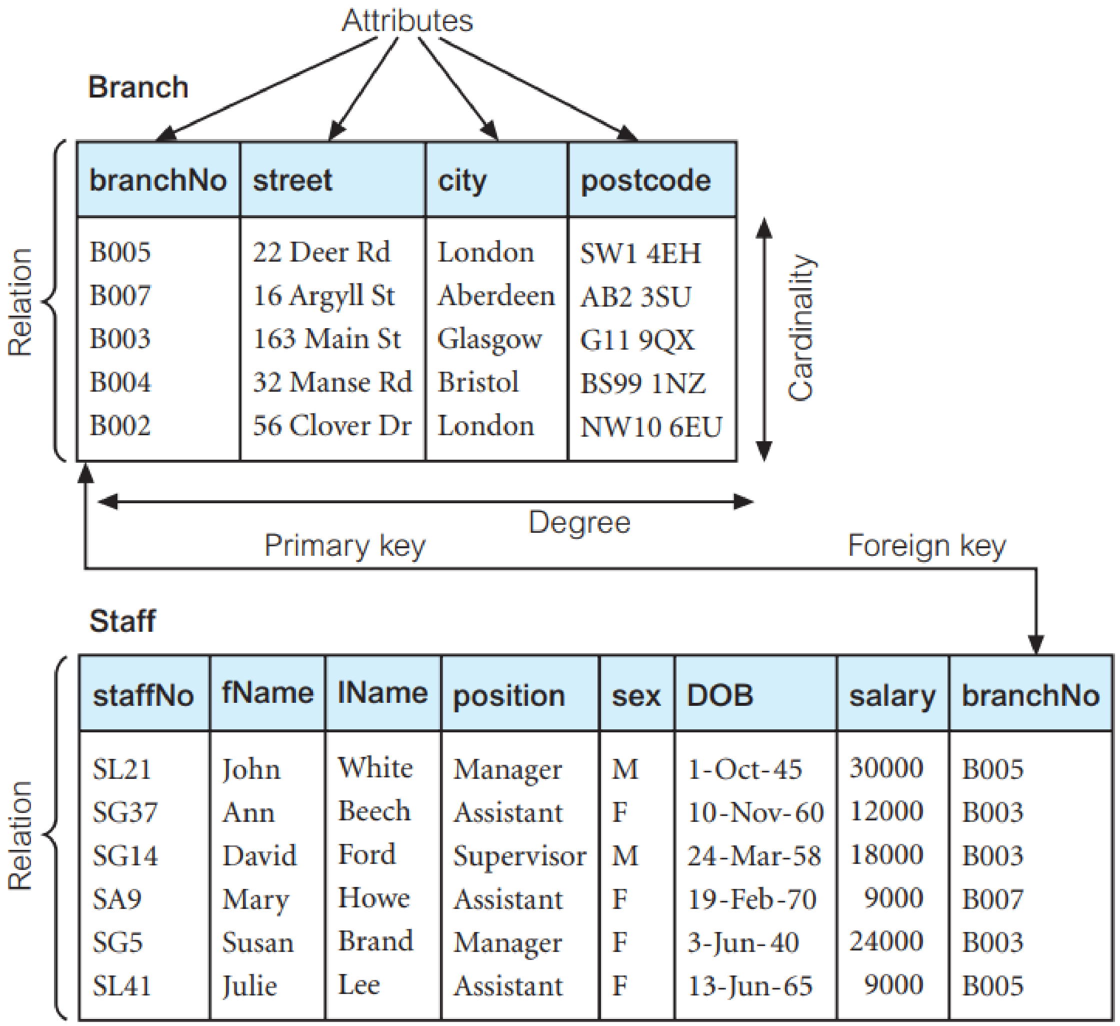
staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee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

Integrity Constraints

- A data model needs to satisfy constraints to ensure the correctness/accuracy/timeliness of data.
- Domain constraint
  - Restrictions on the values allowed for the attributes of a relation.
  - Age = -50, Sex = A, etc.
- Integrity rules: entity integrity and referential integrity
  - constraints or restrictions that apply to all instances of the database
  - Closely related to the concept of Null values.

Null

- Represents a state for an attribute that is currently unknown or is not applicable for this tuple.
  - Nulls are a way to deal with incomplete or exceptional data.
  - Null is not a value. Unlike 0 or -1.
- A new staff is just added, but hasn't been decided which branch he belongs to.
  - Or use a false branchNo? Then you will have to add this branchNo to Branch relation too. May cause extra problems.
- Ultimately, it is <u>your decision</u>.



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

Relational Keys and Integrity Constraints

- Primary Key enforces **entity integrity**
  - In a relation, no attribute of a primary key can be **null**.
- Foreign Key enforces **referential integrity**
  - Either the foreign key value must match a value in referred candidate key, or the foreign key must be **null**.
- Keys are constraints applied to relations.
- DBMS will enforce these constraints when relations are modified
  - E.g. Trying to add duplicated rows to a table with candidate key will never succeed.

Integrity Constraints: Small Practice

Consider following tables (underlined are Primary Keys)

- Hotel (hotelNo, hotelName, city)
- Room (roomNo, hotelNo, type, price)
- Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)
- Guest (guestNo, guestName, guestAddress)

Identify the foreign keys in this schema, Explain how the entity and referential integrity rules apply to these relations.

Assignment

- Read “Database Systems: A Practical Approach …” by Connolly and Begg, 6th Ed.

- Go through Chapter 1 to Chapter 4 quickly.

- Focus more on chapter 4.