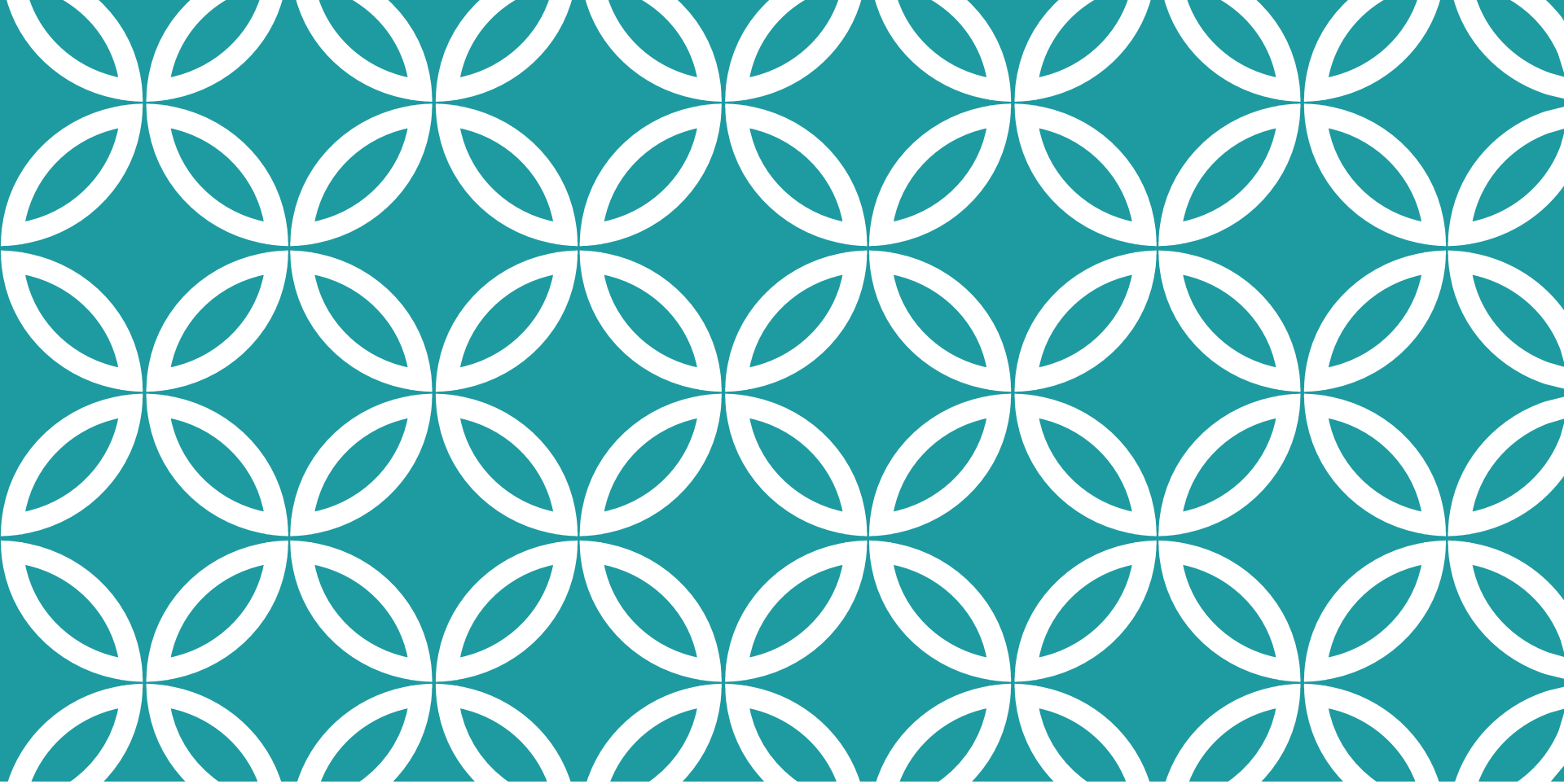


DATABASE SYSTEMS

(Part 1 of 2)

CT102: Information Systems

DATABASE SYSTEMS

fname	minit	lname	ssn	bdate	address	gender	salary	superssn	dno
John	B	Smith	123456789	1975-01-09	731 Fondren, Houston, Tx	Man	55250	333445555	5
Franklin	T	Wong	333445555	1980-12-08	638 Voss, Houston, TX	Man	65000	888665555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	Woman	44183	333445555	5
Ramesh	K	Narayan	666884444	1995-09-15	975 Fire Oak, Humble, TX	Man	60000	333445555	5
James	E	Borg	888665555	1997-11-10	450 Stone, Houston, TX	Man	94199	*NULL*	1
Jennifer	S	Wallace	987654321	1991-06-20	291 Berry, Bellaire, TX	Woman	69240	888665555	4
Ahmad	V	Jabbar	987987987	2000-03-29	980 Dallas, Houston, TX	Man	44183	987654321	4
Alicia	J	Zelaya	999887777	1998-07-19	3321 Castle, Spring, TX	Non-binary	44183	987654321	4

A database system is an Information System that stores and retrieves **structured** data

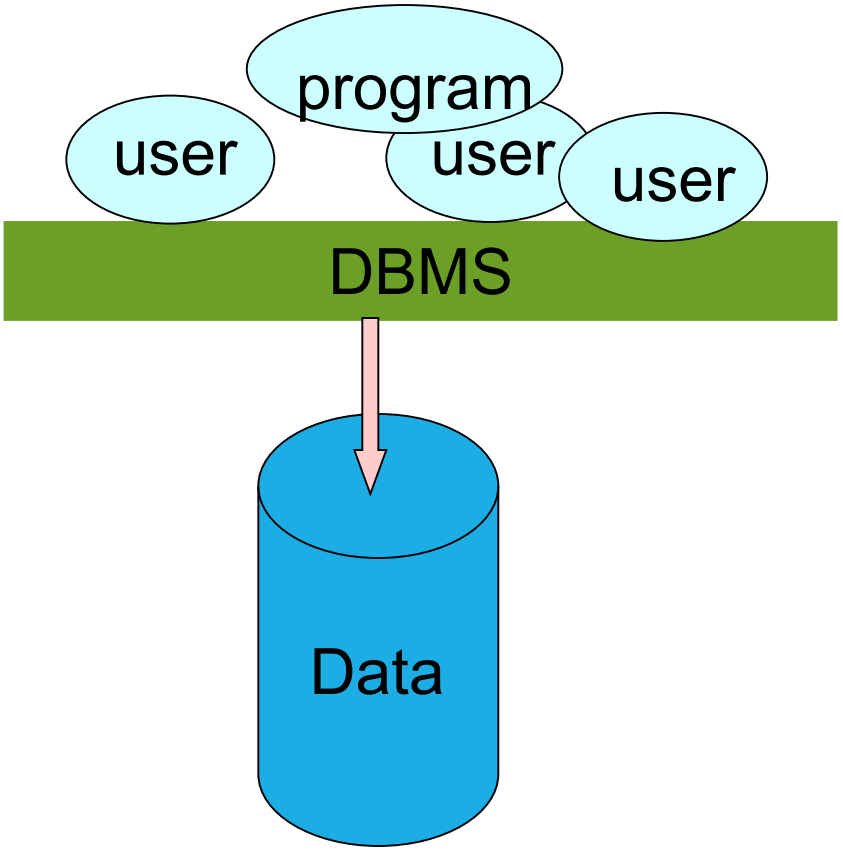
DATABASE DEFINITION

One or more **tables**

- where a table is an ordered collection of **records**
  - where a record consists of data

DATABASE APPROACH

A single repository of data is maintained that is defined once and then accessed by various users/programs through a DBMS

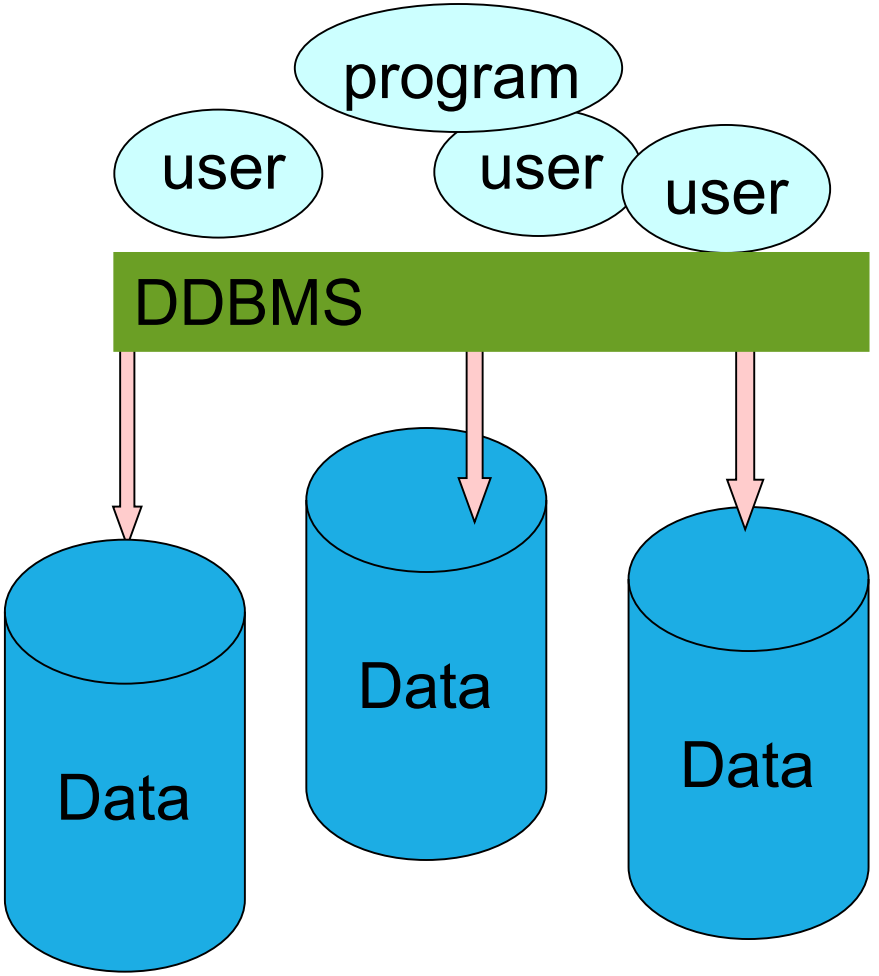


DISTRIBUTED DATABASE APPROACH

Data is defined once and then stored at multiple (**distributed**) sites

However:

Users have the impression of a single repository of data



TYPES OF DATABASE SYSTEMS

- Relational Databases (mySQL, Sybase, Oracle)
- Non-Relational Databases (MongoDB, Redis, Apache Cassandra)
- XML databases (BaseX, eXist, Sedna)
- Blockchain databases

DATABASE MANAGEMENT SYSTEMS (DBMS)

A DBMS is a collection of programs that facilitates the process of **defining, constructing** and **manipulating** databases for various applications.

Microsoft® SQL Server®

Microsoft Office Access

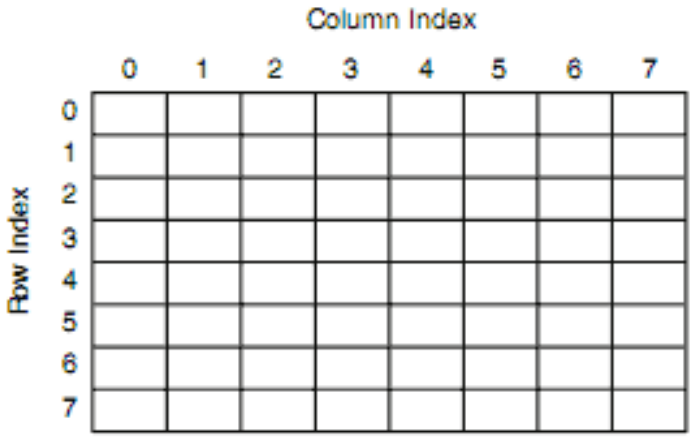
ORACLE®

MySQL®

SYBASE®

The SAP DB Database System

RELATIONAL DATABASES

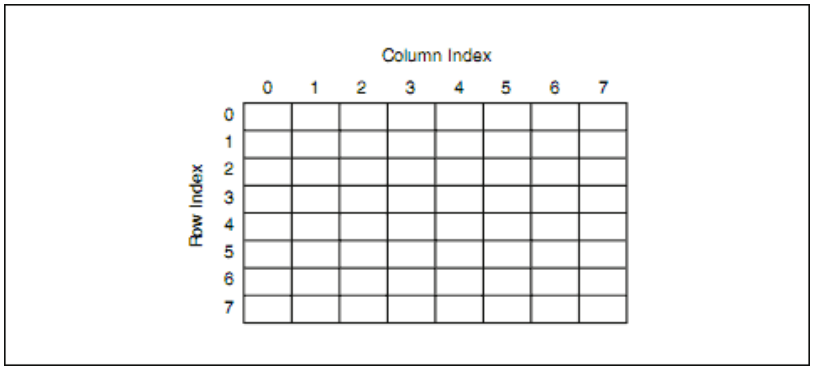


Based on the mathematical theory of relations (originally from Codd, IBM, 1970s)

Can be seen as consisting of "tables and only tables":

- A table is a natural representation of a relation
- A table is a 2-d array

RELATION TABLES



The theory refers to “relations”

The implementation refers to “tables”

Each relation table has a name

The top row contains headings called attributes

An attribute corresponds to a column

Every other row (0 or more) is an **instance** of the relation and is defined by **a tuple** (row) having components (values) corresponding to the attributes

RELATION TABLES

ID	fname	surname	email	courseCode	currYear
16555666	Claire	Cox	c.cox555@nuigalway.ie	GY406	3
17444455	Marc	Bale	m.bale444@nuigalway.ie	GY350	2
17667788	Jack	Carr	j.carr667@nuigalway.ie	GY101	2
17987654	Marie	Berger	m.berger987@nuigalway.ie	GY101	2
17998877	Hugh	Flynn	h.flynn998@nuigalway.ie	GY350	2
18112233	Anna	Chikarovski	a.chikarovski122@nuigalway.ie	GY350	1
18123456	Donal	Nee	d.nee123@nuigalway.ie	GY101	1
18333222	Sadhbh	O'Malley	s.omalley333@nuigalway.ie	GY350	1
18654321	Sean	Lynch	s.lynch654@nuigalway.ie	GY101	1

Recall that mathematical relations do not contain duplicates:

- In relation tables no two tuples can be exactly the same (across all attribute values).
- To ensure this completely, one or more special attributes are chosen (or added) which are called **primary key** attributes which must have unique values for each tuple.
- We use the convention (in writing) that attributes that form the primary key are underlined
- Graphically they are often represented with an image of a key.

RELATIONAL DBMS IN INDUSTRY

90% of industry applications use Relational DBMS or Relational DBMS with extensions.

The majority of industry applications require:

- Correctness
- Completeness
- Efficiency (Complex optimisation techniques and complex Indexing structures)

Relational DBMS provide this

RELATIONAL DBMS HAVE….

1. Design/Structure View where you can see structure of tables – names, data types and constraints
2. Datasheet/Browse View where you can see the database instance – the actual data in the tables
3. Usually a results window
4. Usually a SQL editor (to write code)
5. And many (many) other features ….

SAMPLE DESIGN/STRUCTURE VIEW

	#	Name	Type	C
☐	1	id 🔑	bigint(20)	
☐	2	fname	varchar(50)	la
☐	3	surname	varchar(50)	la
☐	4	email	varchar(50)	la
☐	5	courseCode	varchar(5)	la
☐	6	currYear	int(11)	

Where we specify:

- Attribute (column) names
- Attribute (column) data types
- Primary key

SAMPLE DATASHEET/BROWSE VIEW

ID	fname	surname	email	courseCode	currYear
19555666	Claire	Cox	c.cox555@universityofgalway.ie	GY406	4
20444455	Marc	Anderson	m.anderson22@universityofgalway.ie	GY350	3
23111122	Sean	Kirwan	s.kirwan53@universityofgalway.ie	GY101	1

Where we enter the actual data

TABLE 1: addressbook

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh
Peter Smith	12	Tudor Vale	Oranmore	Galway	Ireland	091888666	085454545
Ali Byrne	31	Station Road	Athenry	Galway	Ireland	091888444	085989811
Cheryl Ainsley	131	Cherry Gardens	Newcastle	Galway	Ireland	091232323	086123123
Chris Nowak		Golf Road	Westport	Mayo	Ireland	098660012	086876543
Ben Okoro	31	Clare's Walk	Ennis	Clare	Ireland	065767676	087123456
Gabe Jones		Dun Mor	Roundstone	Galway	Ireland	095333666	087232323
Jane Doyle		Claremount	Claremorris	Mayo	Ireland	0949367821	087665544

EXAMPLE 1: using table 1

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh
Peter Smith	12	Tudor Vale	Oranmore	Galway	Ireland	091888666	085454545
Ali Byrne	31	Station Road	Athenry	Galway	Ireland	091888444	085989811
Cheryl Ainsley	131	Cherry Gardens	Newcastle	Galway	Ireland	091232323	086123123
Chris Nowak		Golf Road	Westport	Mayo	Ireland	098660012	086876543
Ben Okoro	31	Clare's Walk	Ennis	Clare	Ireland	065767676	087123456
Gabe Jones		Dun Mor	Roundstone	Galway	Ireland	095333666	087232323
Jane Doyle		Claremount	Claremorris	Mayo	Ireland	0949367821	087665544

Number of attributes?

Name of attributes?

Data type of attributes?

Number of rows?

Any duplicates?

Any potential duplicates?

Alternative if not using database system?

For single person applications, relatively easy to store the data from a single table in a text file and write a program, in programming language of choice, to open file and access data.

TABLE 2: appointments

aptID	PatientName	DateOfBirth	ConsultantName	Consult Room	Consult Area	AptDate	Click
1	Peter Murphy	08/06/1989	Prof Keogh	113	ENT	13/11/2023	
2	Ali Byrne	23/06/2001	Dr Lee	201	Gastro	23/11/2023	
3	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	25/01/2024	
4	Chris Nowak	02/06/2004	Prof Keogh	113	ENT	21/01/2024	
5	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	23/11/2023	
6	Jane Doyle	10/06/1985	Mr Gormley	101	ENT	30/11/2023	
7	Ben Okoro	22/05/1969	Mr Comer	107	Ophthalmology	19/01/2024	
8	Ali Byrne	23/06/2001	Mr Gormley	101	ENT	19/01/2024	
9	Gabe Jones	20/06/2006	Dr Garvey	205	Dermatology	01/02/2024	

Description:

Each appointment has a unique ID and associates one patient with one consultant and an appointment date.

Each Consultant has their own room (ConsultRoom) and a speciality ConsultArea).

For each patient, a date of birth is stored.

FUNDAMENTAL CONCEPT IN RELATIONAL DATABASES:

PRIMARY KEY

It is very important that the primary key is unique and unambiguous and remains so even when **new**, yet unseen, data is added to a table.

Repetition in a primary key is ruled out theoretically and also not desirable in practical terms.

Often considerable effort is involved in the choosing, or creation, of a primary key

Examples of good primary keys

- PPS numbers or equivalent (unique within a country)
- Student IDs, Staff IDs (unique within an organisation)
- Bank account numbers (unique within a bank)
- Hospital chart numbers (unique within a hospital)
- Car registration numbers (unique within a country/region)

Others?

- Mobile phone numbers?
- Email addresses?
- Usernames?

CHOOSING A PRIMARY KEY? (1 OF 2)

In general want the simplest primary key possible:

- Not too long if possible – but length dependent on number of keys potentially required
- Chosen from existing attributes rather than having to add new one if possible
- Not too many attributes, one is best if possible
- Not too complex a data type, e.g. integers are easiest!

CHOOSING A PRIMARY KEY? (2 OF 2)

- Some existing attribute may be unique and can be chosen
- Some combination of existing attributes may be unique (in combination) and can be chosen (if not too many and if not too complex)
- Some new ("artificial") attribute can be picked and added (e.g., autonumber datatype).

TABLE 1:
Suitable primary key for the addressbook table?

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh
Peter Smith	12	Tudor Vale	Oranmore	Galway	Ireland	091888666	085454545
Ali Byrne	31	Station Road	Athenry	Galway	Ireland	091888444	085989811
Cheryl Ainsley	131	Cherry Gardens	Newcastle	Galway	Ireland	091232323	086123123
Chris Nowak		Golf Road	Westport	Mayo	Ireland	098660012	086876543
Ben Okoro	31	Clare's Walk	Ennis	Clare	Ireland	065767676	087123456
Gabe Jones		Dun Mor	Roundstone	Galway	Ireland	095333666	087232323
Jane Doyle		Claremount	Claremorris	Mayo	Ireland	0949367821	087665544

TABLE 2:
What is a suitable primary key for the appointments table?

aptID	PatientName	DateOfBirth	ConsultantName	Consult Room	Consult Area	AptDate	Click
1	Peter Murphy	08/06/1989	Prof Keogh	113	ENT	13/11/2023	
2	Ali Byrne	23/06/2001	Dr Lee	201	Gastro	23/11/2023	
3	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	25/01/2024	
4	Chris Nowak	02/06/2004	Prof Keogh	113	ENT	21/01/2024	
5	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	23/11/2023	
6	Jane Doyle	10/06/1985	Mr Gormley	101	ENT	30/11/2023	
7	Ben Okoro	22/05/1969	Mr Comer	107	Ophthalmology	19/01/2024	
8	Ali Byrne	23/06/2001	Mr Gormley	101	ENT	19/01/2024	
9	Gabe Jones	20/06/2006	Dr Garvey	205	Dermatology	01/02/2024	

RECALL: DATABASE DEFINITION

One or more **tables**

- where a table is an ordered collection of **records**
  - where a record consists of data

ONE OR MORE TABLES?

A relational database *could* consist of just one large table (which is common in *nosql databases*)

For many purposes, this would be impractical and inefficient and would be difficult to update (i.e., add, modify or delete tuples or data).

The table would contain a great deal of **redundancy**

DEFINITION: Redundancy

Unnecessary **duplication** of data in a table as a result of data not being split into multiple tables

Duplication:

- If an attribute in a database has two identical values
- Data may be duplicated without being redundant
- Data is duplicated rather than redundant if when deleting or restructuring data in to multiple tables, information is lost

CONSEQUENCES OF REDUNDANCY

- Space is wasted
- Data can become inconsistent (data integrity is lost)
- Problems with update, insert and delete operations

Redundancy in appointments table?

Which, if any, attributes have redundancy?

aptID	PatientName	DateOfBirth	ConsultantName	Consult Room	Consult Area	AptDate	Click
1	Peter Murphy	08/06/1989	Prof Keogh	113	ENT	13/11/2023	
2	Ali Byrne	23/06/2001	Dr Lee	201	Gastro	23/11/2023	
3	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	25/01/2024	
4	Chris Nowak	02/06/2004	Prof Keogh	113	ENT	21/01/2024	
5	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	23/11/2023	
6	Jane Doyle	10/06/1985	Mr Gormley	101	ENT	30/11/2023	
7	Ben Okoro	22/05/1969	Mr Comer	107	Ophthalmology	19/01/2024	
8	Ali Byrne	23/06/2001	Mr Gormley	101	ENT	19/01/2024	
9	Gabe Jones	20/06/2006	Dr Garvey	205	Dermatology	01/02/2024	

Answer: ?

NORMALISATION

All tables in a relational database must satisfy certain desirable properties

A hierarchy of "normal forms" exist that impose increasing restrictions on tables

These normal forms use "functional dependencies"

These normal forms are called:

- 1st, 2nd and 3rd normal forms
- Boyce-Codd (BCNF) normal form
- 4th and 5th normal forms

FUNCTIONAL DEPENDENCY

An attribute Y is functionally dependent on X, if knowing X can uniquely determine Y

e.g., if Y = name and X = studentID ….

The attribute *name* is functionally dependent on the attribute *studentID* as knowing a *studentID* can uniquely determine a *name*

Note: the reverse is **not** true.

Functional dependencies present in appointments table?:

aptID	PatientName	DateOfBirth	ConsultantName	Consult Room	Consult Area	AptDate	Click
1	Peter Murphy	08/06/1989	Prof Keogh	113	ENT	13/11/2023	
2	Ali Byrne	23/06/2001	Dr Lee	201	Gastro	23/11/2023	
3	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	25/01/2024	
4	Chris Nowak	02/06/2004	Prof Keogh	113	ENT	21/01/2024	
5	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	23/11/2023	
6	Jane Doyle	10/06/1985	Mr Gormley	101	ENT	30/11/2023	
7	Ben Okoro	22/05/1969	Mr Comer	107	Ophthalmology	19/01/2024	
8	Ali Byrne	23/06/2001	Mr Gormley	101	ENT	19/01/2024	
9	Gabe Jones	20/06/2006	Dr Garvey	205	Dermatology	01/02/2024	

Functional dependency from ID to PatientName, to ConsultantName and to AptDate
Functional dependency from PatientName to DateOfBirth (assuming no two patients with the same name …. ?)
Functional dependency from ConsultantName to Room* and to Speciality
*assuming consultant always has same room

REMOVING/REDUCING REDUNDANCY

Split data in to multiple tables according to functional dependencies

Important:

- No information should be lost
- Some attributes may exist more than once across multiple tables and this allows tables to be linked and cross-referenced (can be considered *necessary* duplication of data)

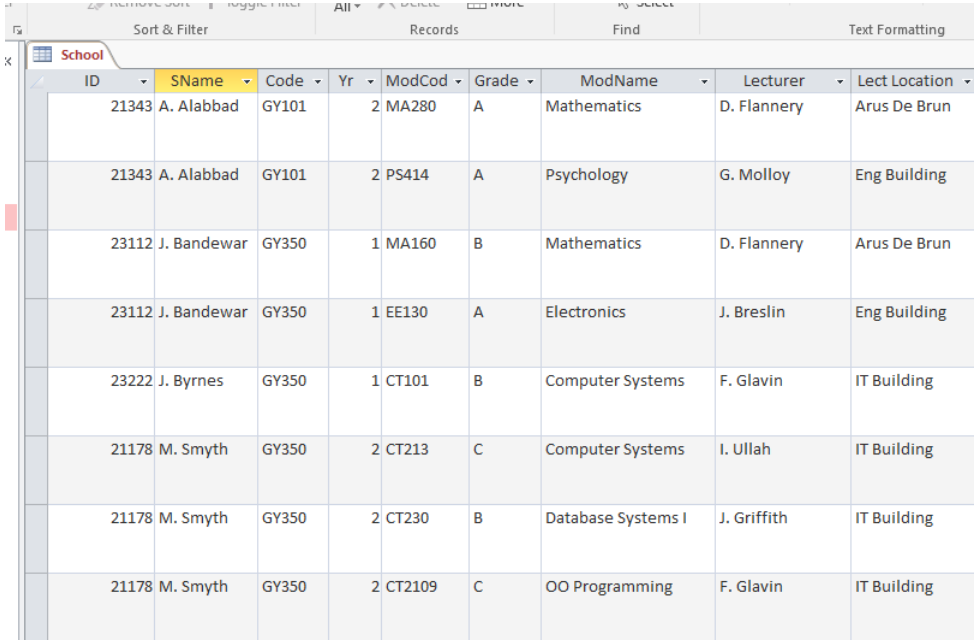
A better ordering of attributes appointments table?
(Using multiple tables)

aptID	PatientName	DateOfBirth	ConsultantName	Consult Room	Consult Area	AptDate	Click
1	Peter Murphy	08/06/1989	Prof Keogh	113	ENT	13/11/2023	
2	Ali Byrne	23/06/2001	Dr Lee	201	Gastro	23/11/2023	
3	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	25/01/2024	
4	Chris Nowak	02/06/2004	Prof Keogh	113	ENT	21/01/2024	
5	Cheryl Ainsley	17/06/1995	Dr Garvey	205	Dermatology	23/11/2023	
6	Jane Doyle	10/06/1985	Mr Gormley	101	ENT	30/11/2023	
7	Ben Okoro	22/05/1969	Mr Comer	107	Ophthalmology	19/01/2024	
8	Ali Byrne	23/06/2001	Mr Gormley	101	ENT	19/01/2024	
9	Gabe Jones	20/06/2006	Dr Garvey	205	Dermatology	01/02/2024	

Answer: 3 tables:

```
patient(pID, PatientName, DateOfBirth)
consultant(cID, ConsultantName, ConsultRoom, ConsultArea)
appointment(aptID, pID, cID, aptDate)
```

YOU TRY … TABLE 3: School



ID	SName	Code	Yr	ModCod	Grade	ModName	Lecturer	Lect Location
21343	A. Alabbad	GY101	2	MA280	A	Mathematics	D. Flannery	Arus De Brun
21343	A. Alabbad	GY101	2	PS414	A	Psychology	G. Molloy	Eng Building
23112	J. Bandewar	GY350	1	MA160	B	Mathematics	D. Flannery	Arus De Brun
23112	J. Bandewar	GY350	1	EE130	A	Electronics	J. Breslin	Eng Building
23222	J. Byrnes	GY350	1	CT101	B	Computer Systems	F. Glavin	IT Building
21178	M. Smyth	GY350	2	CT213	C	Computer Systems	I. Ullah	IT Building
21178	M. Smyth	GY350	2	CT230	B	Database Systems I	J. Griffith	IT Building
21178	M. Smyth	GY350	2	CT2109	C	OO Programming	F. Glavin	IT Building

Description:

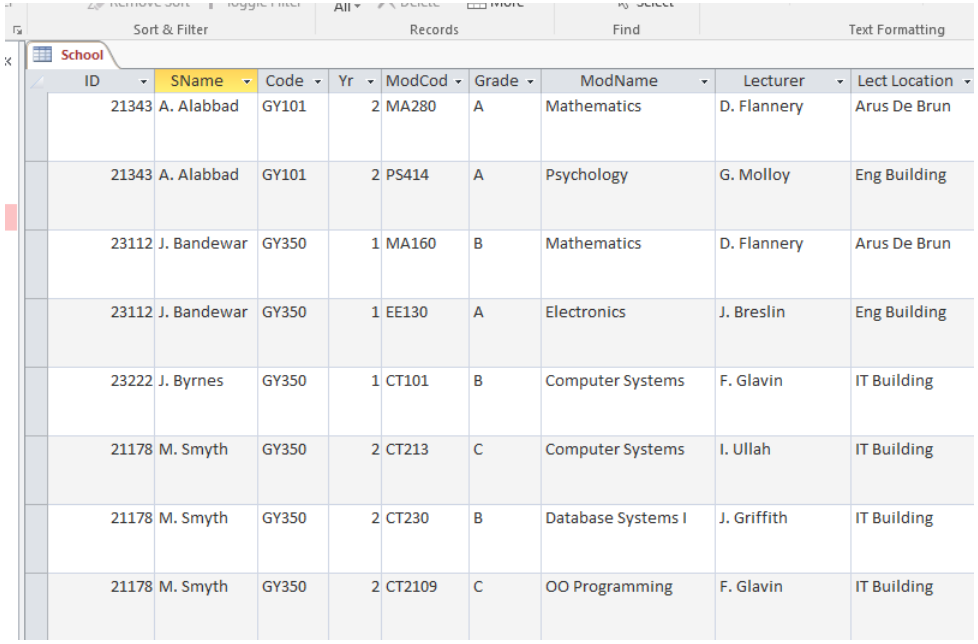
Each student has an associated unique ID, name, course on which they are registered and their current year.

Each student is also registered for a number of modules and receives a grade for that module.

Each module has a code, name and lecturer.

Each lecturer has a location.

YOU TRY … TABLE 3: School



ID	SName	Code	Yr	ModCod	Grade	ModName	Lecturer	Lect Location
21343	A. Alabbad	GY101	2	MA280	A	Mathematics	D. Flannery	Arus De Brun
21343	A. Alabbad	GY101	2	PS414	A	Psychology	G. Molloy	Eng Building
23112	J. Bandewar	GY350	1	MA160	B	Mathematics	D. Flannery	Arus De Brun
23112	J. Bandewar	GY350	1	EE130	A	Electronics	J. Breslin	Eng Building
23222	J. Byrnes	GY350	1	CT101	B	Computer Systems	F. Glavin	IT Building
21178	M. Smyth	GY350	2	CT213	C	Computer Systems	I. Ullah	IT Building
21178	M. Smyth	GY350	2	CT230	B	Database Systems I	J. Griffith	IT Building
21178	M. Smyth	GY350	2	CT2109	C	OO Programming	F. Glavin	IT Building

Choose an appropriate primary key (if possible)

Identify any redundancy in the table

Identify any functional dependencies (based on the description given)

Suggest a better ordering of attributes than that given (potentially in multiple tables)

SUMMARY (PART 1)

- A database stores data in a structured format – having named columns (attributes) and their associated data type
- A primary key is a special attribute that has a unique value for each row of data entered to the table
- A database can have many tables
- Redundant data is often removed/reduced by considering functional dependencies and creating new tables
- A special programming language called SQL is used with relational database systems (and many other database systems use a language similar to SQL) … which we cover next.