DIGIT~Bio~TECH

LO5 OPEN ACCESS SCIENTIFIC RESOURCES: DIGITAL DATABASES

Basic level

AUTHOR:
FABIANO CHALHOUB & ZOI GEORGIOU

Contents

Open access scientific resources

AN INTRODUCTION TO «OPEN ACCESS» RESOURCES

An Open access or OA is a set of principles and a range of practices through which research outputs are distributed online, free of cost or other access barriers. With open access strictly defined (according to the 2001 definition), or libre open access, barriers to copying or reuse are also reduced or removed by applying an open license for copyright.

The main focus of the open access movement is "peer reviewed research literature". Historically, this has centered mainly on print-based academic journals. Whereas conventional (non-open access) journals cover publishing costs through access tolls, such as subscriptions, site licenses or pay-per-view charges, open-access journals are characterized by funding models which do not require the reader to pay to read the journal's contents. Open access can be applied to all forms of published research output, including peer-reviewed and non-peer-reviewed academic journal articles, conference papers, theses, book chapters, monographs and images.

However, when it comes to define “free” access, one has to distinguish “gratis” from “libre”.

In order to reflect real-world differences in the degree of open access, the distinction between *gratis* open access and *libre* open access was added in 2006 by Peter Suber and Stevan Harnad, two of the co-drafters of the original Budapest Open Access Initiative (BOAI) definition of open access publishing. *Gratis* open access refers to online access free of charge and *libre* open access refers to online access free of charge plus some additional re-use rights. *Libre* open access is equivalent to the definition of open access in the BOAI, the Bethesda Statement on Open Access Publishing and the Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities. The re-use rights of *libre* OA are often specified by various specific Creative Commons licenses; these almost all require attribution of authorship to the original authors.

The document released in February 2002 by the BOAI contains the following very widely used definition:

- By "open access" to this literature, we mean its free availability on the public internet, permitting any users to read, download, copy, distribute, print, search, or link to the full texts of these articles, crawl them for indexing, pass them as data to software, or use them for any other lawful purpose, without financial, legal or technical barriers other than those inseparable from gaining access to the internet itself. The only constraint on reproduction and distribution and the only role for copyright in this domain, should be to give authors control over the integrity of their work and the right to be properly acknowledged and cited.

In light of the information above, the use of open source scientific resources must follow the rules commonly adopted. Publishing open source scientific resources must also clearly mention if they are *libre* or *gratis* and must be attributed to the original author.

An introduction to data

WHAT ARE "DATA"

According to the Merriam-Webster dictionary, there are three different definitions of *data*:

1. Factual information, such as measurements or statistics, used as a basis for reasoning, discussion, or calculation
2. Information in digital form that can be transmitted or processed
3. Information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful

In this document we will cover most of the three definitions.

A BRIEF HISTORY OF DATA

Since humans began to communicate, they have experienced the need to retain information for the long term. Keeping information was necessary for our ancestors to ensure their survival. Transmitting information across generations allowed them to keep track of potential dangers, but also to have an inventory of the best places to collect food, the best spots for fishing, the most interesting animals to hunt and where to find the best shelters. All of this information was transmitted orally. With the evolution of knowledge and the invention of writing, they began to store information on indelible media.

Without going into detail about the evolution of the representation of information, some significant examples will be provided that have helped in the structuring of thought, which lead to the discovery of the computer tools we use daily.

DATA PRIOR TO THE INVENTION OF COMPUTERS

As human societies emerged, collective motivations for the development of writing were driven by pragmatic exigencies. These include organizing and governing societies through the formation of legal systems, contracts, deeds of ownership, taxation, trade agreements, treaties, census records, keeping history, maintaining culture, keeping track of scientific discoveries, codifying knowledge through curricula and lists of texts that are artistically exceptional or deemed to contain foundational knowledge, and many other needs.

For example, around the 4th millennium BC, the complexity of trade and administration in Mesopotamia outgrew human memory, and writing became a more dependable method of recording and presenting transactions in a permanent form.

Figure 1. Cuneiform writing

Cuneiform was one of the earliest systems of writing, invented by Sumerians in ancient Mesopotamia. It is distinguished by its wedge-shaped marks on clay tablets, made by means of a blunt reed for a stylus, as demonstrated in Fig. 1.

Over time, the development of knowledge, the multiplication of information, the limitation of human memory, the necessity of writing and keeping record of huge quantities of information has become essential. However, despite keeping record of almost every kind of information or data on various media, it became more and more complex to retrieve it in a simple manner. One had to read tens of reports and books to be able to synthesize on a subject.

DATA IN MODERN AGE

Today, the quantity of data produced every year and kept digitally, e.g., to-do lists, recipes, reminders, logbooks, maps, photos, e-mails, scientific data, political reports, videos, etc. is so exponential that it creates the need to structure the way we can retrieve these phenomenal quantities.

Computers gained popularity and became cost effective to use by individuals and private companies in the early 80's. However, the 60's can be considered as the new era in the field of databases. The introduction of the term "database" coincided with the availability of direct-access storage or DAS, from the mid-60s onward. This new technology represented a contrast with the past punch cards and the tape-based systems, allowing shared interactive use rather than daily batch processing. Two main data models were developed – network model "CODASYL" (Conference on Data System Language) and hierarchical model "IMS" (Information Management System).

The first generation of database systems was "navigational, in opposition to the sequential access due to the previous technologies used to store data, i.e. tapes and punch cards. Applications typically accessed data by following pointers from one record to another. Storage details depended on the type of data to be stored.

Adding an extra field to a database required rewriting the underlying access/modification scheme. Emphasis was on records to be processed, not the overall structure of the system. A user would need to know the physical structure of the database in order to query for information. One database that proved to be a commercial success was the "SABRE" system that was used by IBM to help American Airlines manage its reservations data. This system is still utilized by the major travel services for their reservation systems.

In modern Information technology, confusion has always existed among users between databases and internet web search engines accessed by browsers. A database usually contains structured data, in contrast to the World Wide Web (www), which usually contains unstructured data. Even if retrieving information from both databases and "www" are seamless and look similar, the content and the way queries are addressed are completely different. Structured and unstructured data will be explained later in this document.

Understanding the basic vocabulary

TERMINOLOGY

Like any other science, computer science has its own language. In order to fully comprehend the information that will be provided in this document, it is essential to become familiar with the vocabulary related to this topic.

Moreover, the communication with a DBA (Database Administrator) will be eased. When a Biochemist will have to express his needs in terms of structuring or managing data in a Database, he will be tempted to use his own technical language. Then the DBA will have to understand the request and transform it into a computer language, which will be understandable by biochemists.

What are “Data” in the computer age

As mentioned in section 2.1, according to the domain that is being referred to, *data* might have different meanings. In the case of computing and databases, *data* is defined as any sequence of one or more symbols. *Data* requires interpretation to become information. In information technology the “bit” is the smallest quantity of data. A bit is binary. Binary numbers are a representation of numbers using only two digits, 0 and 1 (Fig. 2). It is a base-2 numeral system, i.e.:

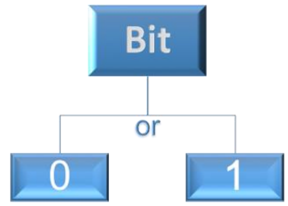


Figure 2. A bit can be 0 or 1

- 0 0 0 1 = numerical value 2^0
- 0 0 1 0 = numerical value 2^1
- 0 1 0 0 = numerical value 2^2
- 1 0 0 0 = numerical value 2^3

A sequence of “bits” constitutes a “Byte”. Bytes are made of a multiple of 4 bits (a byte of 4 bits is called a Nibble) as in the example above. Today, the byte is a unit of digital information that most commonly consists of eight bits. Historically, the byte was the number of bits used to encode a single character of text in a computer. With a byte of eight bits the maximum decimal number is 256. Historically, the byte is also the unit of computer information or data-storage capacity used to measure the quantity of data (Table 1).

Table 1

Abbreviation	Unit	Value	Size in bytes
b	bit	0 or 1	1/8 of a byte
B	bytes	8 bits	1 byte
KB	kilobytes	10^{3} bytes	1,000 bytes
MB	megabyte	10^{6} bytes	1,000,000 bytes
GB	gigabyte	10^{9} bytes	1,000,000,000 bytes
TB	terabyte	10^{12} bytes	1,000,000,000,000 bytes
PB	petabyte	10^{15} bytes	1,000,000,000,000,000 bytes
EB	exabyte	10^{18} bytes	1,000,000,000,000,000,000 bytes
ZB	zettabyte	10^{21} bytes	1,000,000,000,000,000,000,000 bytes
YB	yottabyte	10^{24} bytes	1,000,000,000,000,000,000,000,000 bytes

Table 2. ASCII Table

BINARY BYTE OF 8 BIT								DECIMAL	HEXADECIMAL	CHARACTER
0	0	0	0	0	0	0	0	0	00	NULL
0	0	0	0	0	0	0	1	1	01	SOH
0	0	0	0	0	0	1	0	2	02	STX
0	0	0	0	0	0	1	1	3	03	ETX
0	0	0	0	0	1	0	0	4	04	EOT
0	0	0	0	0	1	0	1	5	05	ENQ
0	0	0	0	0	1	1	0	6	06	ACK
0	0	0	0	0	1	1	1	7	07	BELL
0	0	0	0	1	0	0	0	8	08	BS
0	0	0	0	1	0	0	1	9	09	TAB
0	0	0	0	1	0	1	0	10	0A	LF
0	0	0	0	1	0	1	1	11	0B	VT
0	0	0	0	1	1	0	0	12	0C	FF
0	0	0	0	1	1	0	1	13	0D	CR
0	0	0	0	1	1	1	0	14	0E	SO
0	0	0	0	1	1	1	1	15	0F	SI
0	0	0	1	0	0	0	0	16	10	DLE
0	0	0	1	0	0	0	1	17	11	DC1
0	0	0	1	0	0	1	0	18	12	DC2
0	0	0	1	0	0	1	1	19	13	DC3
0	0	0	1	0	1	0	0	20	14	DC4
0	0	0	1	0	1	0	1	21	15	NAK
0	0	0	1	0	1	1	0	22	16	SYN
0	0	0	1	0	1	1	1	23	17	ETB
0	0	0	1	1	0	0	0	24	18	CAN
0	0	0	1	1	0	0	1	25	19	EM
0	0	0	1	1	0	1	0	26	1A	SUB
0	0	0	1	1	0	1	1	27	1B	FSC
0	0	0	1	1	1	0	0	28	1C	FS
0	0	0	1	1	1	0	1	29	1D	GS
0	0	0	1	1	1	1	0	30	1E	RS
0	0	0	1	1	1	1	1	31	1F	US

BINARY BYTE OF 8 BIT								DECIMAL	HEXADECIMAL	CHARACTER
0	0	1	0	0	0	0	0	32	20	SPACE
0	0	1	0	0	0	0	1	33	21	!
0	0	1	0	0	0	1	0	34	22	"
0	0	1	0	0	0	1	1	35	23	#
0	0	1	0	0	1	0	0	36	24	$
0	0	1	0	0	1	0	1	37	25	%
0	0	1	0	0	1	1	0	38	26	&
0	0	1	0	0	1	1	1	39	27	'
0	0	1	0	1	0	0	0	40	28	(
0	0	1	0	1	0	0	1	41	29	)
0	0	1	0	1	0	1	0	42	2A	*
0	0	1	0	1	0	1	1	43	2B	+
0	0	1	0	1	1	0	0	44	2C	,
0	0	1	0	1	1	0	1	45	2D	-
0	0	1	0	1	1	1	0	46	2F	.
0	0	1	0	1	1	1	1	47	30	/
0	0	1	1	0	0	0	0	48	31	0
0	0	1	1	0	0	0	1	49	32	1
0	0	1	1	0	0	1	0	50	33	2
0	0	1	1	0	0	1	1	51	34	3
0	0	1	1	0	1	0	0	52	35	4
0	0	1	1	0	1	0	1	53	36	5
0	0	1	1	0	1	1	0	54	37	6
0	0	1	1	0	1	1	1	55	38	7
0	0	1	1	1	0	0	0	56	39	8
0	0	1	1	1	0	0	1	57	3A	9
0	0	1	1	1	0	1	0	58	3B	:
0	0	1	1	1	0	1	1	59	3C	;
0	0	1	1	1	1	0	0	60	3D	<
0	0	1	1	1	1	0	1	61	3E	=
0	0	1	1	1	1	1	0	62	3F	>
0	0	1	1	1	1	1	1	63	40	?

BINARY BYTE OF 8 BIT								DECIMAL	HEXADECIMAL	CHARACTER
0	1	0	0	0	0	0	0	64	41	@
0	1	0	0	0	0	0	1	65	42	A
0	1	0	0	0	0	1	0	66	43	B
0	1	0	0	0	0	1	1	67	44	C
0	1	0	0	0	1	0	0	68	45	D
0	1	0	0	0	1	0	1	69	46	E
0	1	0	0	0	1	1	0	70	47	F
0	1	0	0	0	1	1	1	71	48	G
0	1	0	0	1	0	0	0	72	49	H
0	1	0	0	1	0	0	1	73	4A	I
0	1	0	0	1	0	1	0	74	4B	J
0	1	0	0	1	0	1	1	75	4C	K
0	1	0	0	1	1	0	0	76	4D	L
0	1	0	0	1	1	0	1	77	4E	M
0	1	0	0	1	1	1	0	78	4F	N
0	1	0	0	1	1	1	1	79	50	O
0	1	0	1	0	0	0	0	80	51	P
0	1	0	1	0	0	0	1	81	52	Q
0	1	0	1	0	0	1	0	82	53	R
0	1	0	1	0	0	1	1	83	54	S
0	1	0	1	0	1	0	0	84	55	T
0	1	0	1	0	1	0	1	85	56	U
0	1	0	1	0	1	1	0	86	57	V
0	1	0	1	0	1	1	1	87	58	W
0	1	0	1	1	0	0	0	88	59	X
0	1	0	1	1	0	0	1	89	5A	Y
0	1	0	1	1	0	1	0	90	5B	Z
0	1	0	1	1	0	1	1	91	5C	[
0	1	0	1	1	1	0	0	92	5D	\
0	1	0	1	1	1	0	1	93	5E	]
0	1	0	1	1	1	1	0	94	5F	^
0	1	0	1	1	1	1	1	95	60	_

BINARY BYTE OF 8 BIT								DECIMAL	HEXADECIMAL	CHARACTER
1	0	1	0	0	0	0	0	96	96	`
1	0	1	0	0	0	0	1	97	97	a
1	0	1	0	0	0	1	0	98	98	b
1	0	1	0	0	0	1	1	99	99	c
1	0	1	0	0	1	0	0	100	9A	d
1	0	1	0	0	1	0	1	101	9B	e
1	0	1	0	0	1	1	0	102	9C	f
1	0	1	0	0	1	1	1	103	9D	g
1	0	1	0	1	0	0	0	104	9E	h
1	0	1	0	1	0	0	1	105	9F	i
1	0	1	0	1	0	1	0	106	A0	j
1	0	1	0	1	0	1	1	107	A1	k
1	0	1	0	1	1	0	0	108	A2	l
1	0	1	0	1	1	0	1	109	A3	m
1	0	1	0	1	1	1	0	110	A4	n
1	0	1	0	1	1	1	1	111	A5	o
1	0	1	1	0	0	0	0	112	A6	p
1	0	1	1	0	0	0	1	113	A7	q
1	0	1	1	0	0	1	0	114	A8	r
1	0	1	1	0	0	1	1	115	A9	s
1	0	1	1	0	1	0	0	116	AA	t
1	0	1	1	0	1	0	1	117	AB	u
1	0	1	1	0	1	1	0	118	AC	v
1	0	1	1	0	1	1	1	119	AD	w
1	0	1	1	1	0	0	0	120	AE	x
1	0	1	1	1	0	0	1	121	AF	y
1	0	1	1	1	0	1	0	122	B0	z
1	0	1	1	1	0	1	1	123	B1	{
1	0	1	1	1	1	0	0	124	B2	|
1	0	1	1	1	1	0	1	125	B3	}
1	0	1	1	1	1	1	0	126	B4	~
1	0	1	1	1	1	1	1	127	B5	DEL

An example of usage is the ASCII (American Standard Code for Information Interchange) table of characters commonly used for alphabetical characters (Table 2). The first 32 characters are called control characters. Initially, they were not designed to represent printable information, but to control devices that use ASCII code, such as printers, or to provide meta-information about data streams, e.g., those stored on magnetic tape.

What is "Metadata"

Metadata, or, put simply, meta-information, is used to reference the data about the data. Having data is not enough to simply put them online. Data are not usable until they can be explained in a manner that both humans and computers can process.

Figure 3. Picture taken in Greece

Metadata may be implied, specified or given. It includes data relating to physical events, or processes, and will also have a temporal component. In almost all cases this temporal component is implied. It may be slightly tricky to understand, however, the following example will provide a clearer explanation of this term.

Imagine that you are traveling with your favorite smartphone in some paradisiac island. You start taking pictures (Fig. 3) to keep nice records of your trip. A week later, your trip reaches its end and you have to go back home.

Back home, you invite your best friends for a party and want to share with them the beauties you have seen during your trip. You start showing the pictures, but you cannot recall which day, at what time and where some of them were taken. This is where the metadata of the pictures can help. In a few words, it is the description of the data. In this example, the picture is the data and the description of the picture is the metadata (Fig. 4).

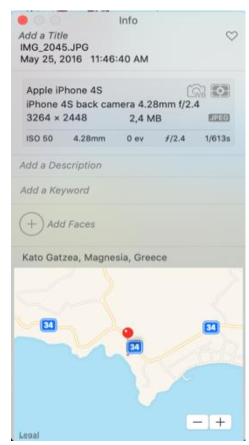


Figure 4. Metadata of the picture

In Biotechnology, one must understand that metadata are by far more important than data. It is very simple to understand the reason why metadata are a crucial component directly related to data. Imagine an experiment that will lead to a specific result. This experiment, to be valid, must be documented. This documentation should include all the conditions, under which the experiment was conducted. This might include the description of the kind of raw material used, its source, in which conditions it was collected, the types of machines to process the experiment, temperature, date, time, etc. For the result of this experiment to be comparable to other results of similar experiments, all the conditions must be similar. Raw data without metadata are useless.

The biggest challenge in Biotech, and any other science, is to standardize metadata. In most of the Biotech databases, this is not respected. One must absolutely be conscious of this phenomenon and thoroughly respect the standards.

What is a "Database"

In general, a database is defined as a collection of data items, such as phone books, price lists, inventory lists, customer's addresses, etc. Nonetheless, in technical terms, a database is referred to as "a self-describing collection of integrated records". It implies computer technology, completed with a specific computer language, such as SQL (Structured Query Language).

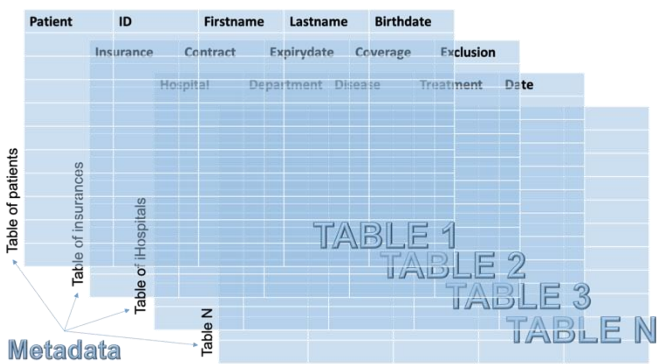


Figure 5. Database partial structure

A database consists of multiple tables (Fig. 5) and of both data and metadata. Metadata is the data that describes the structure of the data within a database. If you know how your data is arranged, then you can retrieve it. Since the database contains a description of its own structure, it is referred to as *self-describing*. The database is integrated because it includes not only data items but also the relationships among them.

The database stores metadata in an area called the data dictionary, which describes the tables, columns, indexes, constraints and other items that make up the database.

Because a flat file system i.e. "Spreadsheet" has no metadata, applications written to work with flat files must contain the equivalent of the metadata as part of the application program.

What are "Tables" in a database

A table is a collection of related data held in a table format composed of columns and rows within a database. It resembles a spreadsheet (Fig. 6).

What are "Columns" in a database

A column is a set of data values, all of a single type, in a table. Columns define the data in a table. Most databases allow columns to contain complex data like images, whole documents or even video clips. Therefore, a column allowing data values of a single type does not necessarily mean it only has simple text values. Some databases go even further and allow the data to be stored as a file on the operating system, while the column data only contains a pointer or link to the actual file. This is done for the purpose of keeping the overall database size manageable – a smaller database size means less time taken for backups and less time required to search for data within the database.

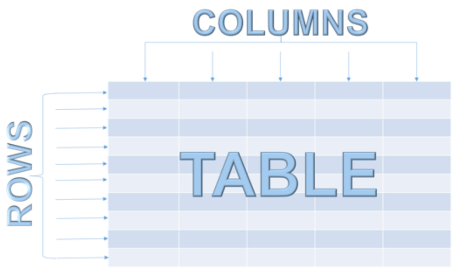


Figure 6. Table with rows and columns

In a table, each column is typically assigned a data type and other constraints, which determine the type of value that can be stored in that column. For example, one column might accept email addresses, and another might accept phone numbers with a constraint of 10 digits.

What is a "Record"

A record is a representation of a physical or conceptual object. Say, for example, that you want to keep track of the customers of a business. You assign a record for each customer. Each record has multiple attributes, such as name, address, and telephone number. Individual names, addresses and so on are the data.

What are “Indexes”

Structured data are stored in the form of records in a database. Every record has a key field, which helps it to be recognized uniquely, i.e. the ID of a patient. No other patient can have the same ID number, but another patient may have the same first name and last name.

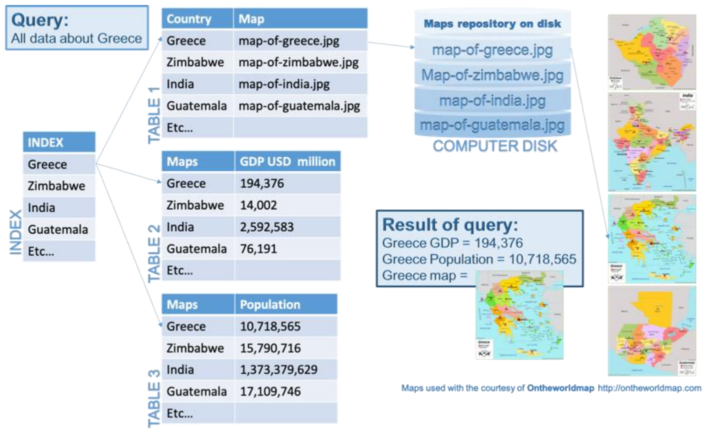


Figure 7. Example of index

Indexing a database is a technique to efficiently retrieve records from the database files, based on some attributes on which the indexing has been performed. To make it simple, indexing in database systems is similar to what we usually see in books. At the beginning or the end of a book, an index may be found (which is different from a table of contents), which provides all the page numbers for a specific topic. For example, an Atlas may be divided into chapters containing maps, chapters containing data on population and chapters dedicated to countries production or agricultural data. If you are looking for a specific country and you would like to have an overview of all the data regarding this specific country, the index might be very helpful as it will show you the page related to that country in each chapter (Fig. 7).

What is an “Object”

In computer science, an *object* can be a variable, a data structure, a function, or a method and, as such, is a value in memory referenced by an identifier. In the relational model of database management, an object can be a table or column, or an association between data and a database entity, such as relating a person's age to a specific person.

Structured data

According to SNIA (Storage Networking Industry association), *structured data* is defined as:

“Data that is organized and formatted in a known and fixed way. The format and organization are customarily defined in a schema. The term structured

Structured data

First Name	Last Name	Citizenship	City	Etc...
Fyodor	Dostoevsky	Russian	Moscow	...
Albert	Camus	French	Mondovi	...
Dante	Alighieri	Italian	Florence	...
Nikos	Kazantzakis	Greek	Kandiye	...

Figure 8. Structured data

data is usually taken to mean data generated and maintained by databases and business applications."

Three conditions are needed to describe data as structured:

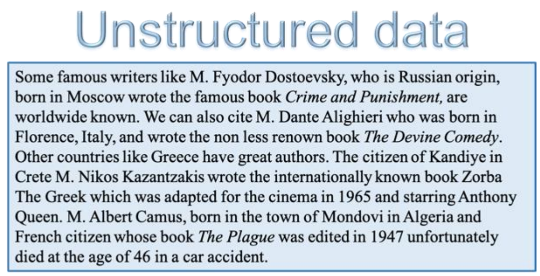
- It must conform to a data model,
- It must have a well define structure,
- It must follow a consistent order and can be easily accessed and used by a person or a computer program.

Structured data is usually stored in well-defined schemas such as Databases. It is generally tabular with columns and rows that clearly define its attributes (Fig. 8).

SQL (Structured Query language) is often used to manage structured data stored in databases.

Unstructured data

Information that is not organized in a predefined model is called *unstructured data* or *unstructured information*. In computer science, files like text files, photos, video files, audio files and presentations are considered unstructured files. Typically, a PDF file contains unstructured data (Fig. 9).


Unstructured data

Some famous writers like M. Fyodor Dostoevsky, who is Russian origin, born in Moscow wrote the famous book *Crime and Punishment,* are worldwide known. We can also cite M. Dante Alighieri who was born in Florence, Italy, and wrote the non less renown book *The Devine Comedy*. Other countries like Greece have great authors. The citizen of Kandiye in Crete M. Nikos Kazantzakis wrote the internationally known book Zorba The Greek which was adapted for the cinema in 1965 and starring Anthony Queen. M. Albert Camus, born in the town of Mondovi in Algeria and French citizen whose book *The Plague* was edited in 1947 unfortunately died at the age of 46 in a car accident.

Figure 9. Extract of a PDF file

It is estimated that 80 to 90% of the worldwide total dematerialized data is unstructured. Usual query algorithms are unable to simply and efficiently extract the required information from an unstructured file, such as in the example of Fig. 9. The same information contained in Fig. 9 can easily be retrieved with a query. However, today, *unstructured data analytics tool*s powered by *artificial intelligence* (AI) are available, which were specifically created to access the insights available from *unstructured data (see 3.1.12 Analytics)*.

Big data

According to SNIA (Storage Networking Industry association), *big data* is defined as:

"A characterization of datasets that are too large to be efficiently processed in their entirety by the most powerful standard computational platforms available."

In other words, *Big Data* refers to huge quantities of structured or unstructured data that cannot be processed by usual software as traditional database query language or any other kind of fetching engine.

Confusion exists concerning the current usage of the terms *Big Data* and *Analytics*. *Big Data* is the information, while *Analytics* is the way to extract the desired information from huge quantities of available information.

Analytics

In computer technology, *Analytics* is a method to extract value from big data.

In the field of healthcare, *Big Data Analytics* has led to many improvements by providing personalized medicine and predictive analytics. As the volume of data is dramatically increasing, traditional databases and search engines will not be able to handle and retrieve specific information. Patient data is generated by MRI's, X-rays, blood tests machines, monitoring sensors and many more sources of data complex to process. Extensive information in healthcare is now in electronic form; it fits under the *big data* umbrella as most of it is unstructured and difficult to use.

Big data in health research is particularly promising in terms of exploratory biomedical research, as data-driven analysis can move forward more quickly than hypothesis-driven research. Subsequently, trends seen in data analysis can be tested in traditional, hypothesis-driven follow-up biological research and eventually clinical research.

Repository

A data repository or data warehouse is a centralized place to store and maintain data. A data repository can consist of one or more *structured data* files, such as databases or *unstructured data* files, which can be distributed over a network and preserved over the long-term.

Basic structure of a database

This section is dedicated to the overview of the main building blocks constitutive of a database.

INTRODUCTION

Since the invention of computers, the amount of data stored and managed electronically has increased drastically. It is estimated that the quantity of data will reach 175 zettabytes (10^{21} Bytes) by 2025 growing from a few petabytes (10^{15} Bytes) in the year 2000. One common way of simplifying the lives of users and making the most of their resources is by storing and retrieving it more efficiently. For example, while a flat file works just fine for storing your personal data, such as an address book or some recipes, it is not as suitable for storing a city phone directory or, more precisely, the genomic data in the Biotech field. In addition, if you want to store several genomic species worth of data, it is very difficult to search and retrieve data from a flat file. Databases offer a solution to this problem by making the storage, handling and retrieval of data much easier.

The software used to manage a database is called a database management system (DBMS). This specialized software acts as a go between to help end users access the database. Usually, users do not interact directly with a database because this may result in its disorganization. Instead, they use a DBMS that reads data from or writes data to the database.

The growing complexity of big quantities of data required some companies to use data management tools based on the relational model, such as the classic RDMBS. RDBMS stands for Relational Database Management System. Nevertheless, major Internet companies, such as Google, Yahoo and Amazon, or all the popular Social Media, each faced challenge in dealing with huge quantities of data in real-time, something that conventional RDBMS solutions could not cope with. That explains the soaring popularity of NoSQL database systems that sprang up alongside.

NoSQL systems are distributed, non-relational databases designed for large-scale data storage and for massively-parallel, high-performance data processing across a large number of commodity servers. They arose out of a need for agility, performance and scale, and can support a wide set of use cases, including exploratory and predictive analytics in real-time. Built by top internet companies to keep pace with the data deluge, NoSQL databases scale horizontally and are designed to scale to hundreds of millions and even billions of users performing updates as well as reads.

Some of the common applications of NoSQL databases are social media, large scale e-mail providers and governmental healthcare systems.

Usually, a social application can scale from zero to millions of users in a few weeks and to better manage this growth, one needs a DB that can manage a massive number of users and data, but can also easily scale horizontally.

In this course, we will focus on DBMS and RDBMS only. These are the two kinds of Databases commonly used in the Biotech world up to date.

OVERVIEW OF A DATABASE ARCHITECTURE

Databases can store all kinds of information, from numbers and text, to email, web content, phone records, biological, geographical data, etc. Databases are officially classified according to how they store this data. Relational databases store data in tables. Object oriented databases store data in object classes and subclasses. We are going to focus on relational databases, as they are most commonly used. However, most of the basic topologies of databases need to have backend servers in order to host the database management system, a storage system attached to the servers to store the structure and the data of the database and, of course, computers, laptops, desktops or terminals as an interface to allow users to access the database, its management system and its content. Also required is a network to exchange between all the hardware components and a Cloud attachment to allow remote users to access the database. Fig. 10 summarizes in a simple way the minimum required to run a database.

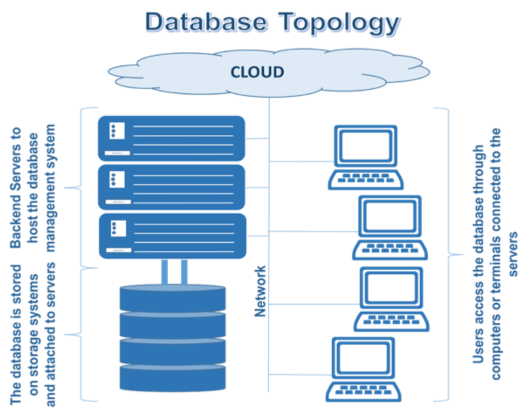


Figure 10. Hardware architecture for a database

Another basic way to describe it, is to show the three level architecture of a database. It is a virtual view of the necessary layers to make a database function properly. Fig. 11 demonstrates the three-level view architecture. It is called the ANSI-SPARC model. Nonetheless, despite the fact that this model never became a formal standard, it presents the idea of logical data independence that has been widely adopted.

Information stored inside a relational database is contained within tables. These tables are composed of rows of data and each row contains fields or columns. In a well-designed database definition, called a schema, only similar data is stored within each table and duplication of columns is kept to a minimum. Developers can connect, or join, data from two tables to link different types of information to each other.

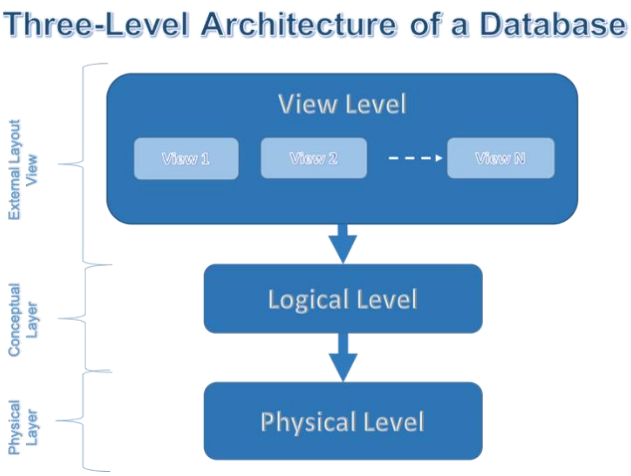


Figure 11. Three-level view architecture

Indexes can be created on fields in the database table to make it easier for the DBMS to retrieve data. Indexes are usually configured for frequently searched columns, like a person's name or a date value. The drawback to using indexes is that they take up storage disk space and can slow things down, if too many of them are maintained, because every time a row in the database is updated, the index also has to be updated.

Most databases support Structured Query Language (SQL), a standard language for interacting with information contained in a database. SQL allows users and applications to interact with specific subsets of data from one or more tables using several statements as SELECT, INSERT, UPDATE and DELETE.

Relational databases also provide a layered approach to storage, allowing the definition of what database objects reside in specific data files and where those data files are placed within the operating system's file structure. On top of managing the physical storage location of database objects, many database systems give some control over how the data is stored within the data files.

COMMON DATABASE TERMS

Certain database terms derive from ways that databases automate write actions. Database developers often automate writing to certain fields or other tables, such as writing a copy of the row being inserted – along with a timestamp or username – to a history or audit table. Most DBMS systems provide several ways to automatically manage database write actions.

Database triggers are the most common method of taking action on data as it is being written to the database. Triggers are usually associated with a particular table and configured to execute at a certain point during a specific write action, such as before or after an update, or after a row is inserted. Triggers can be used to format data, populate a column with data derived from existing information, or even write to another table based on the row being inserted or updated.

A stored procedure is another way of interacting with a relational database. Stored procedures are more complex than triggers and are not tied to a single specific table. Typically created by a developer, they use a combination of SQL and a programming language, such as Java or SQL (depending on the database platform). Stored procedures provide developers a lot of control over how data is validated or massaged by an application. A stored procedure could be used to manage how a user logs in to an application. The procedure might first validate the username and password, then log the success or failure of the attempt to another table, along with other information, including the computer name and a timestamp. An alert could even be sent to the user informing them that their password has expired and must be changed.

Functions are simpler than stored procedure, and can sometimes even be used from within SQL queries. Functions are usually used in a database to perform a set of actions that return one or more values, such as calculating the sum of a column for rows that match a certain condition. While these actions can be performed using SQL, building them into a function can make them easier to use in other code. Both functions and stored procedures can perform common actions in a streamlined and consistent manner, easing the workload for database administrators and developers.

WHAT IS THE DIFFERENCE BETWEEN MAJOR DBMS SYSTEMS?

The DBMS is generally driven by what the user applications need to support. That said, here is a brief comparison of the three most widely used platforms.

Microsoft SQL Server is widely used in enterprise applications and integrates easily with other Microsoft tools. Microsoft SQL Server 2019 Express is the latest version of Microsoft's free offering and is often bundled with applications that use SQL Server.

MySQL has been a favorite for open source developers for the better part of two decades. Often used as a back end for open-source blog or content management systems, MySQL has a massive installed base across the globe. In 2008, MySQL AB was acquired by Sun Microsystems, which was itself acquired by Oracle Corp. in 2009, bringing MySQL under the umbrella of one of its largest competitors. However, the MySQL Community Edition remains free and is well supported by the community. MySQL is available for numerous operating systems, including Linux, UNIX, Mac OS X and Windows.

Oracle Database is considered by many to be the standard in enterprise-level database platforms and supports numerous enterprise applications. Oracle Database Express Edition is available free of charge and is also free to distribute (though it is not technically free software), making it another popular option for developers or hobbyists on Windows or Linux.

Now that you have learned the fundamental database terms and concepts, you are that much closer to speaking the same language as your organization's database developers.

Databases in the scientific word

This part deals with the basics of databases used in the scientific world

INTRODUCTION TO EXISTING DATABASES DEDICATED TO SCIENCE

This section is dedicated to the overview of the most common open access databases used in science.

Continuous developments in the fields of biotechnology and information technology have led to the exponential growth of data. Studies conducted by researchers at the European Bioinformatics Institute (EMBL-EBI) have demonstrated that this growth of information is doubling approximately every year. These extensive amounts of data are stored, organized and constantly updated in scientific databases, where they are readily available for scientists, including biologists and bio-informaticians, to use for research purposes. The information available in biological databases is obtained from a range of scientific fields, including metabolomics, microarray gene expression and proteomics. Apart from storing, organizing and sharing huge volumes of data, the main goal of biological databases is to offer web application programming interfaces (APIs) for computers to exchange and integrate data from many different database resources via an automated method.

Figure 12. Example of names of Biotech databases

Biological databases can be defined as data collections, which are structured in such a way making their contents easy to explore, handle and update. Examples of such databases are presented in Fig. 12. In 1972, the first protein structure database, known as the Protein Data Bank (PDB), was created. This database originally contained only 10 entries, which has now expanded to contain more than 10,000 entries, signifying the rapid growth of biological data. A biological database may contain several types of data, including protein sequences, textual descriptions, attributes, and tabular data. Generally, they can be divided into primary, secondary, and composite databases. Primary databases include data about the sequence or structure alone, whereas secondary databases include data originating from the primary database. Data, such as the conserved sequence and active site residues of protein families, can be found in secondary structure databases. Furthermore, entries of the PDB, which is a primary database, can be found in secondary structure databases, stored in an organized way.

Broadly speaking, biological databases can be categorized into sequence, structure, and pathway databases:

- Sequence databases: The most commonly used biological databases. These include protein and nucleotide sequence databases, which contain wet lab results and are the main source for experimental results. GenBank and EMBL are examples of sequence databases.
- Structure databases: These databases contain information regarding protein structure and molecular interactions. PDB is an example of a structure database.
- Pathway databases: These databases are based on data derived from the comparative study of metabolic pathways. The Kyoto Encyclopedia of Genes and Genomes (KEGG) and Biocyc are two indicative pathway databases.

A typical search in a nucleotide sequence database may, for example, generate data concerning the scientific name of the source organism from which it was isolated, contact name, the input sequence with details of the molecule type and, frequently, literature citations related to the sequence.

Certain tools have been developed to facilitate scientists in data processing and retrieval from biological databases. These tools, which are termed bioinformatics tools, are software programs created for the extraction of meaningful data from the vast number of biological databases and for conducting sequence or structural analysis. Bioinformatics tools are used to obtain data from genomic sequence databases and for the visualization, analysis and retrieval of date from proteomic databases. These tools are largely divided into:

- Homology and similarity tools: These tools are used for the detection of similarities between the sequences of unknown structural and functional sequences, whose function and structure are already known.

- Protein function analysis tools: Programs applied for the comparison of one protein sequence to a secondary (or derived) protein, which permit the estimation of the biochemical function of a query protein.
- Structural analysis tools: These tools allow the comparison of structures with the known structure databases and the establishment of the 2D/3D structure of a protein.
- Sequence analysis tools: Programs used for the additional, more comprehensive assessment of a query sequence, involving evolutionary analysis and identification of mutations.

Biological databases may also be categorized, based on the scope of data coverage, into:

- Comprehensive databases: These databases comprise various types of data from a number of species. Examples of comprehensive databases are GenBank and EMBL.
- Specialized databases: These databases include particular types of data or data from particular organisms. An example of specialized databases is WormBase, which contains information on nematode biology and genomics.

In relation to the level of biocuration, which is defined as the activity of organizing, demonstrating and making biological information readily available to both humans and computers, biological databases are classified as primary and secondary or derivative databases. Primary databases consist of raw data as archival repository, while secondary or derivative databases consist of curated information as added value. In regard to the method employed for curating the data, biological databases may be further classified as expert-curated databases or community-curated databases, which are curated in a co-operative way by numerous researchers.

Additional categorization of biological databases can also be made based on data type. The data types that accordingly classify databases include DNA, RNA, protein, expression, pathway, disease, nomenclature, literature and standard and ontology. Some of the most important and widely used biological databases are the following: *GenBank*, the *UCSC* Genome Browser and Ensembl, which are sequence databases/portals; *WormBase* and The Arabidopsis Information Resource (TAIR), which are model organism databases; and the *PDB*, Online Mendelian Inheritance in Man (OMIM), *MetaCyc* and *KEGG*, which are characterized as non-sequence-centric databases.

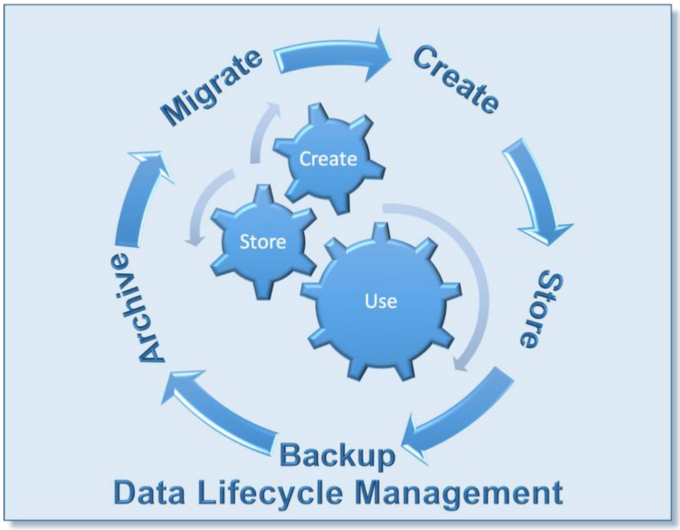
Data manipulation is an essential part of the experimental process of all studies, regardless of their scale. The online availability of biological data combined with the decreasing costs of automated genome sequencers have made it possible for small biology laboratories to become big-data generators. Even if a laboratory is not equipped with such instruments, it can still become a big-data user by gaining access to public repositories containing biological data, such as the US National Center for Biotechnology Information in Bethesda. A large part of the construction in big-data biology is virtual, based on cloud computing, in which data and software are located in massive, off-site centers that can be accessed on demand. Therefore, it is not necessary for users to purchase their own hardware. The cloud computing system allows potential users to create virtual spaces for data, software and results that are freely accessible by everyone, or to keep spaces locked up behind a firewall permitting access to a chosen group of collaborators.

The use of biological databases can be advantageous in several research areas. For example, databases may aid experimental design by allowing the automatic analysis and easy processing of experimental data and making the examination of experimental results simple and quick. Drug discovery is another area that may be simplified by using databases. In this specific area, databases can be scanned in order to find new candidates for drugs by training a classifier on a dataset where functioning and non-functioning drugs have been identified. Moreover, machine learning techniques may be applied to design virtual assays that are able to identify promising new drugs, which can subsequently be analyzed in a laboratory setting. (REF. 4) And most importantly, new scientific experiments can be carried out and new results generated by analyzing existing data sets.

Without the existence of databases, sharing and integration of large quantities of data would be virtually impossible. Although many life scientists have advanced computational skills, a large percentage are not familiar with developing or adapting the relevant software. Nonetheless, the involvement of life scientists in this process is crucial, since they can provide feedback to computer science specialists focusing on different needs and approaches to science. The ability to have access to the actual data sets originally used in a specific study provides researchers with the opportunity to reproduce and expand on such study. This is why it is important for data to become freely available to scientists at any time without restrictions, a notion supported by Open Science and numerous related initiatives. One of these initiatives is known as ELIXIR, a project designed to help scientists across Europe safeguard and share their data and to reinforce current resources, including databases and computing facilities, in individual countries.

Although the creation of biological databases has brought about many benefits, such as the promotion of scientific quality production enabled by networking, they still require improvement in terms of knowledge optimization. It is crucial to manage transdisciplinary knowledge in such a way that will lead to an increase in its quality and quantity. Data heterogeneity is another common issue faced in biological data integration. In the field of biology, several different methods exist for the representation of similar data. This complicates data integration and processing, which, in turn, makes it harder to acquire unified views of such data. An example of this problem is the use of various alternate names when referring to genes, regardless of the existence of full guidelines issued in 1979 proposing the adoption of gene nomenclature standard, leading to difficulties in data sharing. The implementation of standards enables the re-use of data, however, their absence causes significant loss of productivity and contributes to a decrease in data accessible by researchers. Therefore, it is imperative to find a solution to this matter in order to eliminate the challenges faced by scientists when using biological databases to conduct their research.

FINAL THOUGHTS



Dealing with data implies a drastic discipline to keep access on a long term to the stored information. Technology evolves, which means that the hardware and software used today is not the standard of tomorrow. This means that to be able to read any data written today we will have to execute two different kinds of migrations. A logical migration and a technological migration. Logical migration is related to the kind of format in which the data is stored. Technological migration is related to the kind of hardware used. As an example, if you try to open a Word file written in 1993 with Word version 6 with the latest version Word 2019, it will not work. This example shows a lack of logical compatibility. To avoid this issue and keep an ascended compatibility, the file should have been migrated by the time to the latest version in order to keep it up to date and readable with the latest versions of software.

The same thing applies to hardware, i.e. servers, storage, networks, etc… Another example could be the kind of server and operating system used to run a database. In case you decide to change your hardware and to migrate from, let's say, Windows to UNIX, a different kind of hardware will be needed to run UNIX and a different version of database to run on UNIX. Windows run on Intel based platforms (and Intel like) and Unix runs on SPARC based platforms, which means that you will have to migrate to a UNIX - SPARC compatible version of the database.

Keeping in mind this constant evolution of hardware, operating systems, software and formats, performing the appropriate logical and technological migrations on time could save you a lot of time and troubles.

Last but not least, it is important to keep backing up your data. Once every three to six months, perform a restore test to see if you are capable of retrieving your backups. This is crucial for two reasons:

- It will keep you up to date on how to restore your data
- It is the best testing method to see if your data was properly backed up

References

Baxevanis AD, Bateman A. 2015. The importance of biological databases in biological discovery. Curr Protoc Bioinformatics., 50(1):1.1.1-1.1.8.

Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. 2014. GenBank. Nucleic Acids Res., 42:D32–D37.

Brooksbank C, Bergman MT, Apweiler R, Birney E, Thornton J. 2014. The European Bioinformatics Institute's data resources 2014. Nucleic Acids Res., 42:D18–D25.

Caspi R, Billington R, Ferrer L, Foerster H, Fulcher CA, Keseler IM, et al. 2016. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Res., 44(D1):D471-80.

Figueiredo MSN, Pereira AM. 2017. Managing knowledge – the importance of databases in the scientific production. Procedia Manuf., 12:166–73.

Harris TW, Baran J, Bieri T, Cabunoc A, Chan J, Chen WJ. 2014. WormBase 2014: new views of curated biology. Nucleic Acids Res., 42:D789–D793.

Howe D, Costanzo M, Fey P, Gojobori T, Hannick L, Hide W, et al. 2008. Big data: The future of biocuration: Big data. Nature., 455(7209):47–50.

Kanehisa M, Furumichi M, Sato Y, Ishiguro-Watanabe M, Tanabe M. 2021. KEGG: integrating viruses and cellular organisms. Nucleic Acids Res., 49(D1): D545–51.

Karp PD, Billington R, Caspi R, Fulcher CA, Latendresse M, Kothari A, et al. 2019. The BioCyc collection of microbial genomes and metabolic pathways. Brief Bioinform., 20(4):1085–93.

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. 2002. The human genome browser at UCSC. Genome Res., 12(6):996-1006.

Lapatas V, Stefanidakis M, Jimenez RC, Via A, Schneider MV. Data integration in biological research: an overview. J Biol Res (Thessalon). 2015;22(1):9.

Marx V. 2013. Biology: The big challenges of big data: Biology. Nature., 498(7453):255–60.

Nature Structural Biology 10, 980. 2003; doi: 10.1038/nsb1203-980

Oliveira AL. 2019. Biotechnology, big data and artificial intelligence. Biotechnol J., 14(8):e1800613.

Razvi SRH, Rampogu S. 2016. Bioinformatics in the present day. MOJ proteom bioinform [Internet]., 3(1):11–2. Available from: http://dx.doi.org/10.15406/mojpb.2016.03.00073

Toomula N, Kumar A, Kumar D S, Bheemidi VS. 2012. Biological databases- integration of life science data. J Comput Sci Syst Biol., 04(05):087-092. Available from: http://dx.doi.org/10.4172/jcsb.1000081

Yates AD, Achuthan P, Akanni W, Allen J, Allen J, Alvarez-Jarreta J, et al. 2020. Ensembl 2020. Nucleic Acids Res., 48(D1): D682–8.

Zou D, Ma L, Yu J, Zhang Z. 2015. Biological databases for human research. Genomics Proteomics Bioinformatics., 13(1):55–63.

Web sources:

https://en.wikipedia.org/wiki/Airline_reservations_system

https://en.wikipedia.org/wiki/CODASYL

https://en.wikipedia.org/wiki/Database_administrator

https://en.wikipedia.org/wiki/IBM_Information_Management_System

http://www.redbooks.ibm.com/abstracts/sg245352.html

https://en.wikipedia.org/wiki/Navigational_database

https://en.wikipedia.org/wiki/SQL

https://omim.org/

https://www.ascii-code.com

https://www.budapestopenaccessinitiative.org/

https://www.merriam-webster.com/dictionary/data https://www.snia.org/education/online-dictionary/term/big-data

https://www.snia.org/education/online-dictionary/term/structured-data

www.arabidopsis.org/aboutarabidopsis.html

Project website: www.digit-biotech.eu

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.