multiverse

We're on a mission

Data Analysis Concepts Revision Material Pack

Welcome to your Data Fellowship BCS Data Analysis Concepts Revision Pack.

This Revision Pack has been created by Multiverse to help you prepare for your BCS Data Analysis Concepts exam - the first exam you'll take as part of your data fellowship.

We have intentionally left lots of extra space to allow you to add your own annotations and notes.

The format for the examination is a one-hour multiple-choice examination consisting of 40 questions. The examination is closed book (no materials can be taken into the examination room).
The pass mark is 26/40(65%).

Explore the different types of data

Data on its own does not tell us anything – it is only when we process or analyse the data that we're able to get information, and later knowledge from this information.

The difference between Data, Information, Knowledge.
Data – Raw and unorganised facts
Information – Processed data to make it useful
Knowledge – Understanding Information

Data Formats

There are hundreds, if not thousands, of different ways to store digital data. As a data analyst, there are some common types that you are more likely to encounter.

File Type	Characteristics	Benefits	Limitations
.CSV Comma Separated Values	Tabular data. Separated by a Comma. Is a raw text file, does not contain any metadata	Lightweight file. Easily read by many applications.	Commas within the data need to be "text qualified" so the interpreter knows they are part of the data and not delimiters.
.XML eXtensible Markup Language	A hierarchy based markup language that uses user defined keywords to tag data	Computers can read it easily. Portable to many different systems.	Hard for humans to read. Heavy in terms of size due to repeated markup
.RTF Rich Text Format	A file that is stored as Raw text but has Mark up language to denote basic formatting such as bold, underline etc.	Fairly lightweight. Not suitable for holding actual data, is more for documents etc	Rarely used any more. Hard for humans to read as an actual file Pretty much only used by wordpad
.TXT Text	Raw Text. Can be delimited by anything.	Very flexible. Very lightweight. Easily Read.	Can easily 'break' and needs text qualification the same as CSV
.XLSX Excel File later versions	Proprietary spreadsheet file format created by Microsoft Excel	Many users are comfortable with this format, widely used.	Large file size. Specialist software required to view/edit. Hard for external applications to read.

Delimiter. The character, or set of characters, used to separate values in data format that uses a delimited (such as CSV or Txt), Comma delimiters and TAB delimiters are the most common.

ETL || Organise, Structure & process || Concept, mapping, matching

To work with Data in a structured way, we typically need to store it in a database. Generally, if we have files as above, we need to:

➔ **Organise**
Where are our files? Which files are included? Are they in the right order? Can we rename them to make it clearer etc.

➔ **Structure**
Are our files consistent and structured properly. Do they have headings that match in all files, in the same order etc.

➔ **Process**
Take each file and process it i.e. putting it into the database

ETL Process
BCS tests on a specific ETL Process. Weirdly, the one used by company Informatica in their ETL software. This process can change per software/flow etc

Source > Target > Mapping > Session > WorkFlow

Define the source - Where is the data you are adding to the database? What columns does it have?

Define the Target - Where is the data going? What columns are there?

Mapping - Which Column from Source goes into which column in target

Session - The set of instructions that tells the system what to do. covers any transformations that might happen.

Workflow - The actual running of the process. This can often be automated to save time.

Data Sources

Source of Data	Definition
Open/Public	Data that can be freely used, reused and redistributed.
Research (or purchased)	Data from a third party that is made available to you under a licence agreement
Proprietary	Data that is owned and stored within an organisation. Two categories - see below
Operational	Proprietary data that is produced by your organisations day to day operations. Things like customer, inventory, and purchase data fall into this category.
Administrative	Proprietary data that is required to run your organisation but not part of the day-to-day organisations. Such as HR, admin, payroll etc.
Client	A third party's proprietary data.

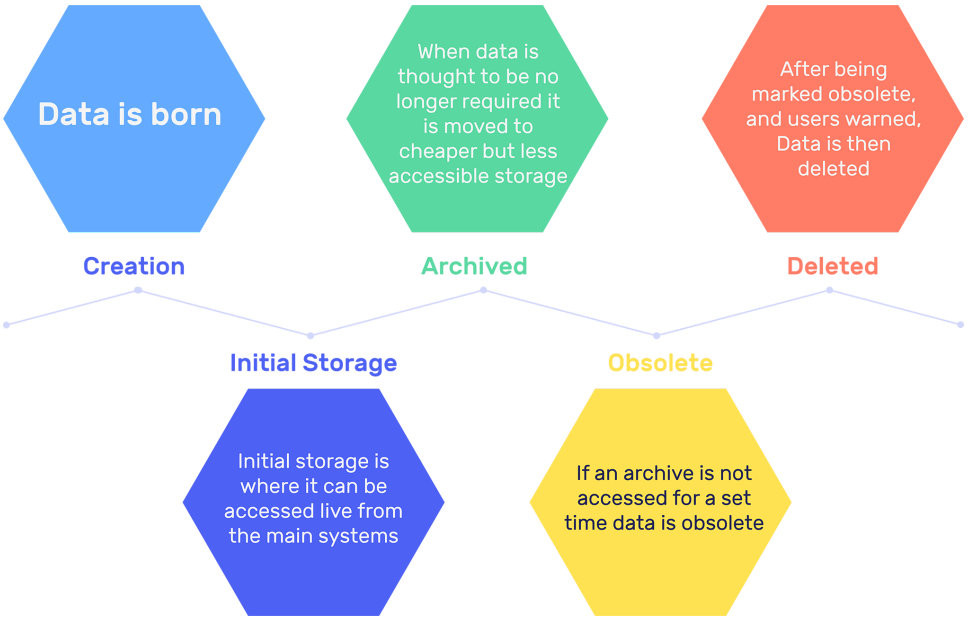
Note Space

The Data Lifecycle

Note: There are two Life Cycles we may think of within data.

Data LifeCycle: The flow of data from its original creation through to its eventual deletion.

Data Project LifeCycle: The flow of a data project from planning through to implementation.

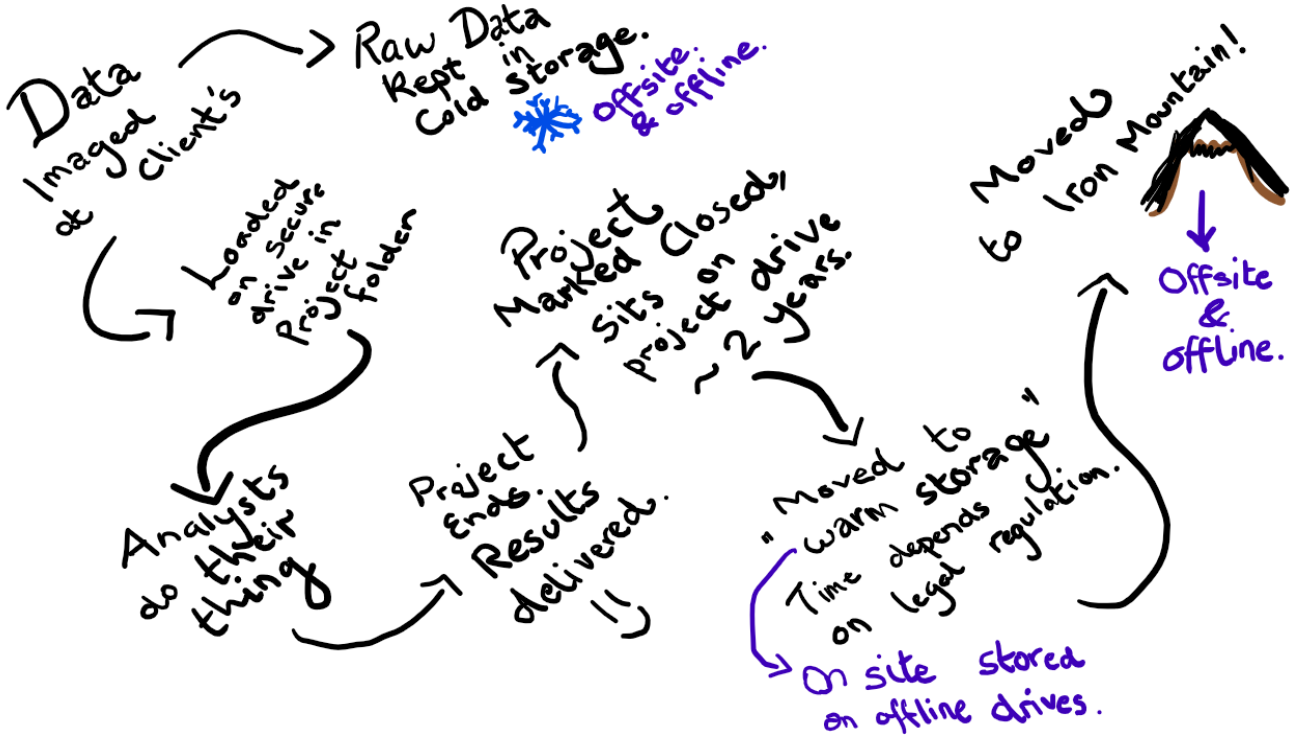


Imagine that we have a database that manages the customers accounts and subscriptions for a large broadcasting company – and that the company has been operating for over 20 years. Over this length of operation, the database will have processed and held a vast volume of information, likely at a granular level.

Storage of data in its 'live' and 'ready' format (immediately available to access) can be expensive both in terms of the physical storage but also the processing cost of searching/finding the correct records.

As data ages, we might find that no longer need nor are capable of storing data in the same way as its initial creation. For example, our large broadcasting company might originally store an itemised version of each customer's bills in the system. After 2 years, the company may not need (nor legally be required) to hold that level of detail any more. They may choose to change the level of detail they hold for older data – perhaps only the bills date and total amount, rather than the itemised level.

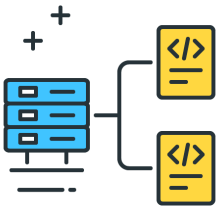
Deloitte Forensic Example



Note Space

Structured and Unstructured Data

Structured Data:



Structured data is information which can be ordered and processed by data analysis tools.

Examples:

- ➔ Tables stored within a database management system.
- ➔ Extensible Markup Language (XML)
- ➔ Serially or sequentially stored data files (such as text or CSV)

Unstructured Data:

Unstructured data is everything else -

Examples:

Word processor, spreadsheet and PowerPoint files. CSV and Text files that are not serial or sequential.

- ➔ Audio
- ➔ Video
- ➔ Sensor and log data
- ➔ External data (such as social media feeds)
- ➔ Paper-based documents

When data looks structured but isn't – Spreadsheets

Spreadsheets (such as Excel), may look 'structured' and you may have designed it with structure in mind – but spreadsheets are defined as unstructured. This is because spreadsheets can not be easily ordered and processed by data analysis tools. Spreadsheets contain no rules around the usage within the spreadsheet – for example, you are under no obligation to define headers, you could have multiple tables on one sheet, you do not need to have only one data type in a column. So while some spreadsheets might look structured, they are typically considered unstructured.

Spreadsheet data is typically stored in proprietary formats (remember opening the XLSX file with an unzipping tool – XLSX files are zip files with XML files within them).

Classifying Data

Quantitative Data	Qualitative Data
Associated with numbers	Associated with details
Implemented when data is numerical	Implemented when data can be segregated into well-defined groups
Collected data can be statistically analyzed	Collected data can just be observed and not evaluated
Examples: Height, Weight, Time, Price, Temperature, etc.	Examples: Scents, Appearance, Beauty, Colors, Flavors, etc.

Quantitative Data

Discrete - Numeric variables that have a countable number of values between any two values. These are always numeric. Examples are number of phone calls made or number of T-shirts in stock. They are countable numbers.

Continuous - Numeric variables that have an infinite number of values between any two values. A continuous variable can be numeric or date/time. For example, the length of a part or the date and time a payment is received. Time series data is continuous.

Qualitative Data

Categorical - Contains a finite number of categories or groups. Such as Gender or payment method. They are quantitative data types where a category is assigned a defined value.

Categorical Data can be further split into these categories -

Nominal - Categorical data where the order doesn't matter. For instance, hair colour, gender etc.

Ordinal - Categorical data where the order does matter. For instance, scales from 1 to 5. Year based Class of students.

Binary (binomial in BCS material)- Categorical data where there are only two options. For instance, yes or no, 0 or 1, Pass or fail.

Note Space

Analysing Unstructured Data

While structured data is generally easily analysed with data analytics tools, unstructured data tends to be more complicated (especially when it is in large quantities). A large part of analysing unstructured data is putting it into a structured format.

Unstructured data can still be analysed statistically - it may just need some tweaking in order to get it into a processable format.

Unstructured and structured data can be used together to deliver rich insight - the unstructured data can enhance the analysis of the structured data.

Example:
A Data Analyst conducts a piece of Customer Insight Analysis, using their structured web analytics database. It shows that a lot of customers are browsing the chalkboard page on their website but ultimately going on to purchase nothing and exiting the website without purchase. The Data Analyst can identify when the behaviour is happening, and how the behaviour is happening and what behaviour is happening. However, using the data they are unable to identify why.

The Data Analyst enlists the assistance of the Customer Experience team who host a small focus group to explore their customers' views around the chalkboard page. They gather a large amount of feedback - unstructured data. This data is then codified and the customer experience team collaborate with the data analyst to identify the 'why' using the input from the customer. This joint and enhanced data can then be used to present a compelling argument on what changes are required for better Chalkboard sales.

Competitive Advantage

In some senses, data can be considered a perishable entity. As time passes, the value of data decreases. Being able to quickly act on data, both unstructured and structured, can give a serious competition advantage.

The value of knowing the winning lottery numbers of a particular draw

Time	Value
The day before the draw.	Highly valuable - you can literally win the lottery.
Just after the draw	Useful - you can determine whether you have won or not.
A week after the draw	Useful if you forgot to check before.
After the closing date of possible redemptions for that draw	Literally useless.

Note Space

Requirements for Data Analysis

Data is not a universal ubiquitous solution to all of a business' problems. Without analysis, data has limited usage. While data can produce a lot of valuable insights after analysis, data does not contain all of the answers – it can only display the best picture of the data based on the data itself and the analysis an analyst decide to do.

Project Scoping

Project Scoping is an important step where you will be gathering all of the project requirements (whether customer/client, internal client, own team etc), and obtaining all of the relevant information.

There are different types of requirements that will influence a project:

- ➔ Business Policies & Standards
- ➔ Stakeholder requirements
- ➔ Technical requirements
- ➔ Budget, resource & scoping constraint

Validation vs Verification

Source: https://www.toolsqa.com/software-testing/difference-between-verification-and-validation/

Validation *is the process of checking whether the specification captures the customer's needs. "Did I build what I said I would?"*

Verification *is the process of checking that the software meets the specification. "Did I build what I need?"*

What is Verification?

Definition: *The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.*

Verification is a static practice of verifying documents, design, code and program. It includes all the activities associated with producing high quality software: inspection*, design analysis and specification analysis. It is a relatively objective process. Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful. Verification is concerned with whether the system is well-engineered and error-free.*

What is Validation?

Definition: *The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.*

Validation is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product.

Requirements elicitation process

The elicitation process is when you gather requirements for the project from various stakeholders.

Methods of Elicitation

- ➔ Apprenticing - Getting the stakeholder to teach you what they know and how they do something. And then try it yourself.
- ➔ Observe - Watching the stakeholders carry out a task.
- ➔ Recount (aka interview)- Asking a stakeholder to recount their experiences and what they do.
- ➔ Enact - Recreate the task and carry it out yourself to make observations.
- ➔ Documents - Documentation can be an input and an output. Will you be using existing documentation to guide parts of the project? This may be in the form of data dictionaries etc.

Explicit vs. tacit knowledge.
In the world of Information & Knowledge Management, knowledge is broadly split into 2 categories - Explicit and tacit. Explicit knowledge is fact and figure based - it is easier to communicate and be captured in a variety of ways. Tacit knowledge is more subjective and harder to communicate and capture. Different elicitation techniques are applicable for different types of knowledge.

Source:
https://www.tlu.ee/~sirvir/Information%20and%20Knowledge%20Management/Key_Concepts_of_IKM/tacit_and_explicit_knowledge.html

Tacit knowledge (knowing-how): knowledge embedded in the human mind through experience and jobs. Know-how and learning embedded within the minds of people. Personal wisdom and experience, context-specific, more difficult to extract and codify. Tacit knowledge Includes insights, intuitions.

Explicit knowledge (knowing-that): knowledge codified and digitized in books, documents, reports, memos, etc. Documented information that can facilitate action. Knowledge what is easily identified, articulated, shared and employed.

Thus, explicit (already codified) and tacit (embedded in the mind).

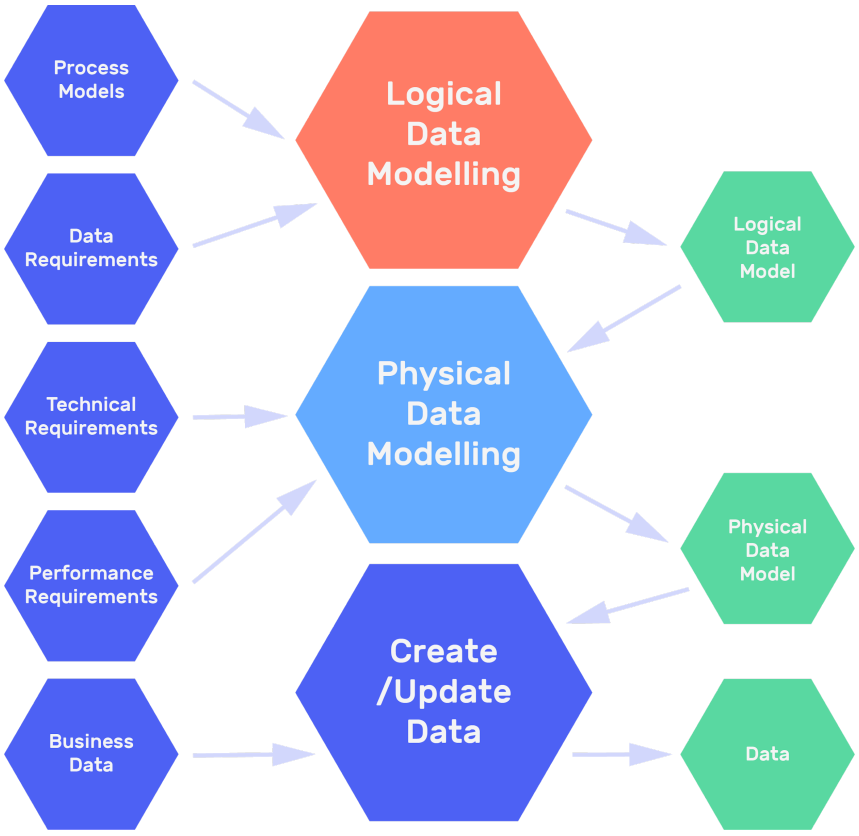
Note Space

Data Modelling

There are mainly three different levels of data modelling:

Source: https://www.guru99.com/data-modelling-conceptual-logical.html

1. **Conceptual:** This Data Model defines **WHAT** the system contains. This model is typically created by Business stakeholders and Data Architects. The purpose is to organise, scope and define business concepts and rules.
2. **Logical:** Defines **HOW** the system should be implemented regardless of the Database Management System (DBMS). This model is typically created by Data Architects and Business Analysts. The purpose is to develop a technical map of rules and data structures.
3. **Physical:** This Data Model describes **HOW** the system will be implemented using a specific DBMS system. This model is typically created by DBA and developers. The purpose is actual implementation of the database.



Quality Issues for Data Analysis (Quality Control)

Good quality data can be the make or break of any data project. Why should you make sure that your Data, and your analysis, is good quality?

Legal and regulatory compliance
We have a legal & compliance duty to ensure data is good quality – GDPR states that a data subject has rights to have data kept about them to be up to date.

Commercial and intellectual property
There is value in data. Bad data has less value. Ensure good data quality protects your data assets.

Confidentiality, integrity, and availability
Good data, and good data management, ensures your company is able to operate well. For instance, imagine you're an award winning apprenticeships provider. You might want to make claims about how successful your company has been - such as we placed XYZ amount apprentices in 2018. Without strong quality control and good data standards, such claims could be unsubstantiated. (Multiverse has great data FYI!)

Common Source Errors

- ➔ Completeness - Is it all there? Have we lost anything?
- ➔ Uniqueness - Do we only have one source of the truth?
- ➔ Timeliness - Is this up to date?
- ➔ Accuracy - Is this right?
- ➔ Consistency - Are we measuring things now in the same as we used to?

Methods to mitigate these errors:

- ➔ **Entry / Transcription**
  How are we putting data into the system? Can we introduce extra checks into this system?
- ➔ **Process**
  What can we do to ensure we're setting ourselves up for data success? How do we create processes that reduce the likelihood of introducing error?
- ➔ **Identification**
  Can we identify when data is bad? How can we know when an error has occurred?
- ➔ **Usage**
  Using the data can be a great way to identify any errors - running tests can ensure that your data is correct.
- ➔ **Structure**
  Having a defined structure means that you can identify problems easier.

Issues from Bad Data

Major issues are:

- ➔ Cost - Dealing with dirty data can be expensive. Analyst time is spent fixing things.
- ➔ Accuracy - You might be making business decisions based on inaccurate data.
- ➔ Inconsistency - Your business operations might fluctuate due to bad data interrupting processes.
- ➔ Cleanliness - Dirty data can make a mess in the system, and cause many issues down the line.

Good quality data that is well defined by an organisation with a strategy for data creation and storage will mean that data analytics is easier and more time can be spent on work that produces value and improved business decision making, rather than tasks such as data cleansing, standardising and verification.

Data Analysis Tasks

While each data analytics project is likely to contain its own unique journey and challenges, there are typical activities or tasks that are likely to occur:

Problem hypothesis

- ➔ Defining the hypothesis as well as null and alternative hypotheses
  - The (null) hypothesis is that there is no significant difference between specified populations, any observed difference being due to sampling or experimental error.
  - A hypothesis test is a statistical test that is used to determine whether there is enough evidence in a sample of data to infer that a certain condition is true for the entire population.
- ➔ Understanding the subject area for analysis
- ➔ Finding similar previous analysis and exploring existing definitions & assumptions

Identifying what to measure

- ➔ Select the data sources
- ➔ Select the aggregation and / or summarisation level

Collect data

- ➔ Understand the size, nature and content of the data
- ➔ Identification of the data security and accessibility
- ➔ Complete data extraction
- ➔ Complete data transfer

Cleanse data

- ➔ Filtering (filtering out irrelevant data)
- ➔ Interpolation (finds the missing value when given two known values across each side of the missing value)
- ➔ Transformation (helps convert the data in the appropriate format)
- ➔ Masking (modifies or hides data to protect sensitive information)
- ➔ Blending (used to combine data from multiple sources into a single location)

Model data

- ➔ Provide assumptions made of data
- ➔ Train the model
- ➔ Data preparation
- ➔ Model definition
- ➔ Validation / verification of predictive models
- ➔ Troubleshooting
- ➔ Validation testing
- ➔ Strategy for improving model performance
- ➔ Identifying and selecting an appropriate model

Visualise data

- ➔ Understand which type of visual data is suitable for the customer:
- ➔ Types of charts (such as line graph, column and bar charts, pie chart,scatter plot, histogram, radar / spider chart, waterfall chart)
- ➔ Geospatial distributions such as heat maps, bubble maps
- ➔ Time series such as time plot, Gantt chart
- ➔ Unstructured data such as Word Cloud

Analyse data

- ➔ Reconcile and compare with other sources

Interpret results

- ➔ Understand the relationship between variables
- ➔ Show and compare the results in terms of real world objects

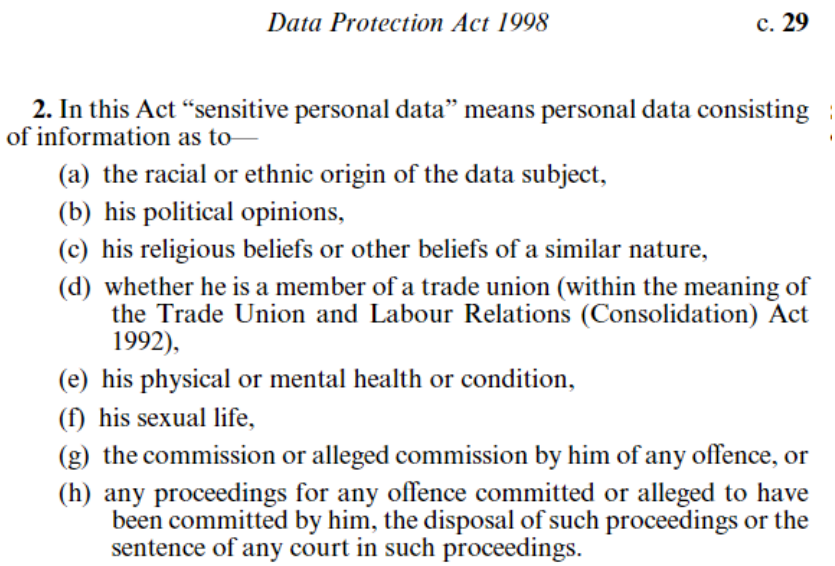
Document and communicate results

- ➔ List the models and assumptions
- ➔ Understand your customer and stakeholders needs and communication style

Note Space

Data Protection & Legal Issues

As of 11/06/2019, the BCS exam has not been updated to reflect the obsoletion of the Data protection act 1998 UK by the introduction of the EU GDPR. **The Data Protection Act was overseen in the UK by the Information Commissioners' Office (ICO) - they also oversee GDPR.**



Data Protection Act 1998 c. **29**

2. In this Act "sensitive personal data" means personal data consisting of information as to—

(a) the racial or ethnic origin of the data subject,

(b) his political opinions,

(c) his religious beliefs or other beliefs of a similar nature,

(d) whether he is a member of a trade union (within the meaning of the Trade Union and Labour Relations (Consolidation) Act 1992),

(e) his physical or mental health or condition,

(f) his sexual life,

(g) the commission or alleged commission by him of any offence, or

(h) any proceedings for any offence committed or alleged to have been committed by him, the disposal of such proceedings or the sentence of any court in such proceedings.

The 8 principles of the Data Protection Act.

There are eight principles of good information handling outlined in the act that state that data must be:

- ➔ Fairly and lawfully processed
- ➔ Processed for limited purposes
- ➔ Adequate, relevant and not excessive
- ➔ Accurate
- ➔ Not kept for longer than is necessary
- ➔ Processed in line with your rights
- ➔ Secure
- ➔ Not transferred to other countries without adequate protection

Personal data is becoming increasingly valuable and the collectors and users of data have responsibilities under the act, such as asking a data subject's permission to use the data.

There are three groups referred to in the act: Data Subjects, Data Users, Data Controllers

Data Structures

There are many different ways that a computer based system can store data.

These Data structures refer to the non-primitive data structures, those storing a collection of values in various formats.

Files

A **computer file** is used for storing information. Computer programs can use files to read the information that needs to be processed and to write the results of the processing. The data inside a file is typically organized in smaller packets of information, which are often referred to as 'records' or 'lines.'

For example, a file can contain the payroll information for each employee, with each employee represented by a line. A computer program can read this information line by line and perform some type of payroll-related operation, such as calculating benefits for a certain pay period. The results could be added to the existing file or written to a new file. Files make it possible for different programs to share information, as one file can be passed onto the next.

Lists

A **list** contains elements of one particular data type. For example, a list could contain strings. An example would be the names of all the players on a soccer team. Each name is a string, but when you organize all the names together, they form a list.

For example, a list of strings could look like this:
('John,' 'Paul,' 'George,' 'Ringo')

A list of numbers could look like this:
(67, 84, 92, 52, 81, 75)

Array

An **array** is a data structure where the elements are identified by one or more indices. An array is similar to a list, but an array can have multiple dimensions. A one-dimensional array is the same as a list: a linear sequence of elements that are all of the same type.

In a two-dimensional array, the elements are organized in two dimensions, which you can think of as the rows and columns of a table. This type of array uses two indices: one for rows and one for columns. The unique combination of two index values represents a unique cell in the table. Each cell corresponds to an element, which can be a string, a number or some other type of data.

A two-dimensional array is also called a matrix. A three-dimensional array can be represented by a cube and uses three indices. Arrays can have more dimensions, but they are more difficult to visualize.

Arrays make it possible to organize data in an efficient manner because the indices make it possible to retrieve any element. They are relatively easy to implement. However, all the elements have to be of the same type, and searching through a large array can be time consuming if it is not sorted.

Records

A **record** is a collection of data relating to one entity. Similar to a single row of data in a table.

Trees

A hierarchical collection of data - expressed through parents and children nodes.

Tables

How data is stored within a database, with structured labelled columns and defined data types

Database Design, Implementation, and Maintenance

Forms of Databases

Some definitions sourced from:
https://www.c-sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/

Type	Examples	Description
Relational	MsSQL, Postgres, mySQL	This is the most common form of database. Defined by having tables that contain data and can be joined to other tables through their relations. Normalisation is a key part of relational databases.
Hierarchical	IBM Information Management System (IMS) Windows Registry	Data is stored in a parent-children relationship nodes, in a tree like structure. The data is stored as a collection of fields where each field contains only one value. The records are linked to each other via links into a parent-children relationship. In a hierarchical database model, each child record has only one parent. A parent can have multiple children. To retrieve a field's data, we need to traverse through each tree until the record is found.
Network	Data Store (IDS), IDMS (Integrated Database Management System), Raima Database Manager, TurboIMAGE	Network database management systems (Network DBMSs) use a network structure to create relationship between entities. Network databases are mainly used on a large digital computers. Network databases are hierarchical databases but unlike hierarchical databases where one node can have one parent only, a network node can have relationship with multiple entities. A network database looks more like a cobweb or interconnected network of records.
Object-oriented	TORNADO, Gemstone, ObjectStore, GBase	Object-oriented databases use small, recyclable separated chunks of data called objects. The objects themselves are stored in the object-oriented database. Each object contains two things: the object itself, and the metadata that explains that object, it's purpose, what it is and where it fits in

NoSQL		The 'No' stands for 'Not Only' SQL - these database seek to improve on standard SQL based DBMS by enhancing the forms of analytics available and changing the way that data is stored (no longer relational.)

NoSQL

While you're likely familiar with relational databases, NoSQL databases are a strong contender and basically really cool - so they get their own section.

No fixed schema. While relational databases are all about control and knowing the data, NoSQL databases allow for a flexible schema.

Avoids Joins. Relational databases use relations to avoid data repetition and wasting space. NoSQL databases are designed in a time of cheap storage and fast processing - so data redundancy is no longer a huge issue. NoSQL databases encourage storing data in single huge tables rather than multiple tiny tables.

All About size and scale. Often distributed across multiple servers and nodes, NoSQL databases are all about using what they need and being able to get hold of more processing power or space when required. With no defined limits, they are able to just scale and scale according to the resources available.

There are two major types of NoSQL Database: Document Store and Graph.

Examples are Neo4J (graph) and MongoDB (document store)

Graph is essentially like the hierarchical format of old, but done really well and in an amazing way.

This is an amazing resource to find out more:
https://www.guru99.com/nosql-tutorial.html

Implementation of design

The Conceptual, Logical and Physical models of database design do not capture all of the requirements and business rules that may exist. Database modelling looks at an idealistic world, and does not hugely take easy of use into account.

Redundancy free & Unambiguous
Has the data design been normalised and designed in a way that eliminates the same data being stored in multiple places? Each data point should be stored once. Are columns, tables, IDs etc unambiguous. Often data dictionaries and documentation are required to clarify data.

Flexible / extensible
Have future requirements been taken into account? Does the database model account for growth or changes in the future? Does it take into account the outliers and various challenges that the end user might come across? Dirty Data is often a result of data that has been entered into an inflexible system and left the user with no choice.

Introduction of derivable data (cumulative values, flags / status values)
Data normalisation serves an important purpose, but sometimes user friendliness is lost in a complex design. Sometimes, features need to be built back in to allow for easier querying.

Take for instance a table of customer addresses which exists to allow for a customer to have multiple addresses over time. If you have a start_date, you can write a query to gather the latest address for each customer. This is a surprisingly complex query when you are getting the current address for all customers. To make things simpler, a flag such as 'is_current' would allow for easier querying. It would introduce a new process - when a new address is inserted, the flag on the old address would need to be updated to 0, and the new one would need to set itself to 1.

Splitting logical data structures, Combining logical data structures or introducing redundant relationships
The modelling stage of database design can be very idealistic and does not take into account technical feasibility or database usage. Sometimes it is necessary to add in additional structures or to combine existing ones to allow the database to actually be used.

For example, a large broadcasting company may want one customer table, that holds all of the information for all customers. However, due to their success, they might find that this becomes unmanageable due to the sheer volume of customers. It might be necessary to split the logical structures into something like customer regions, so you might have London_Customer, South_East_Customer etc, and have a querying step that determines which region a customer should belong to. Sometimes it is also necessary to introduce redundant relationships. Often a commonly used data piece might be many logical table steps away from the data we need to join it to - resulting in

long queries with multiple steps. Sometimes it is necessary to create artificial foreign keys and redundant relationships back to main data subjects.

Database Maintenance

Database maintenance is an activity designed to keep databases running smoothly otherwise databases can become sluggish and lose functionality.

Log file maintenance

The log file is a file that will contain data on various different operations that have happened in the database, depending on the individual settings and the volume of usage of the database, log files can grow rapidly. This can slow down the database as it struggles to read and write to the log file. Log files should be trimmed down regularly, they should also be checked to ensure there are no warning messages.

Data compaction

When data is originally written into a database, the database will be focusing on the speed and efficiency. This may include duplication of data or storing in the rawest formats with no compression. Data compaction ensures that data is stored in the most efficient way.

Defragmentation

A hard disk can be thought of as a grid of blocks. When anything is written to a hard disk, the computer finds a section (that has a section of blocks that are big enough), and will then write the file to it. The computer prioritises speed – this may mean blocks are not efficiently stored (think of a game of tetris with lots of gaps and holes!). Defragmentation is a process whereby the hard disk is reorganised in the most efficient way. This makes reading and writing to the hard disc a lot easier.

Integrity Check

When we design a database there are a series of rules we put into our relationships - from “Must” to “May”, to “one and only one” to “one or more”. An integrity check ensures that these relationships have been observed, and that we do not have things such as orphaned child entries in our data tables. For example -

Each CUSTOMER **may** have **one or more** ADDRESSES
Each ADDRESS **must** belong to **one and only one** CUSTOMER

Address is a dependency of customer. If a Customer is removed from the system without the associated addresses, the left addresses become Orphaned Child entries.

Cascading deletion and properly defined and enforced foreign keys prevents this from happening. Integrity checks ensure that it has not happened.

Backing Up

We have all lost an important document or suffered a power cut deleting our work. Mistakes happen - so data should be backed up! You never know when a graduate is going to accidentally drop the live full database.

Data Architecture

Every organisation will have a different approach to their data architecture.

There are often things in place to govern data such as:

- ➔ Rules
- ➔ Policies
- ➔ Standards
- ➔ Models

and they are likely to govern how data is -

Stored

- ➔ How and where data is stored
- ➔ How long it is stored for

Managed

- ➔ Who owns the data?
- ➔ How often is it backed up?
- ➔ What happens when new data is gained?
- ➔ Who controls access to the data?

Used

- ➔ Who is allowed access to the data?
- ➔ Which departments can use the data?
- ➔ What can specific data be used for?
- ➔ How confidential is data? Can it be used internally and externally?

Integrated (multiple connected systems or siloed systems)

- ➔ How do different systems connect together?
- ➔ How is data transferred across systems?
- ➔ How can we be sure that there is data integrity across systems? If a customer is assigned a customer id in one system, is it the same in another system?

Examples of how Data Storage regulations may be implemented across a business:

Rules:
All client confidential data must be stored in the user restricted tables only and can only be accessed by the client team.

Policies:
Client data is held for 2 years after the close of a project.

Standards:
Client data is stored using the naming convention of {CLIENTNAME} - {PROJECTNAME}

Models:
The physical, logical and conceptual database models.

Data Architecture functions

When we think of any organisation, it is likely that data plays a huge part. Understanding how data sits in its infrastructure, is used by applications and different functions is a vital part of managing an organisation. You can see why proper mapping and understanding of an organisations data is important:

Nightmare Scenario that could happen without good data architecture:
Imagine you work in an IT department and maintain a database containing user details on all of the employees. When a new starter comes, they are added to the system, and leavers are marked as leaving. You use this to manage user permissions on various different IT systems.

Your IT department is migrating from one method of user authentication to a new one that will no longer require this database, so your team carries out the migration and deletes that database.

The end of that month, no one in the organisation gets paid. Payroll is in a panic and is unable to process any payments to employees. It turns out that when HR and the finance team were setting up their new payroll system, your database of employees turned out to be the most accurate and up-to-date, so was the backbone of their entire system.

Key Terms:
Data migration
How are we going to move data from one system to another system. What will happen to the old data - does anyone rely on it? Will the new system hold all of the same information, are we going to lose any? How do we effectively manage this process?

Data modelling
What does our data look like? How much? Where? Who owns it? How is it stored? Data modelling show be able to tell us these answers. Good data modelling also shows you how changes might impact existing data.

Data integration
Company merges are a brilliant way to visualise this one. What happens when our company buys up another company and is integrating their employees and clients into ours. How do we make sure the data is compatible?

Data warehousing
Data warehousing is best thought of as a super data library that is the source of all data in the organisation. Data creation, processing, updating etc may occur elsewhere, but ultimately data ends up in the data warehouse. Data warehouses are typically read-only and do not contain live processed data. They are more like a library or an archive of all of the data, brought together in a standardised and 'single source of truth' model.

The Domain Context for Data Analytics

Data analysis is not ubiquitous across all industries or even organisations within the same industries. A highly successful analyst in company A would not be able to immediately hit the ground running in company B - they would first need the Domain Context. The tools and techniques may be similar, but their implementation and how they are interpreted are likely to be very different.

Scenario:
A bank is identifying customers that held a certain type of account between 2005 and 2008 due to regulatory breaches and the need to compensate those customers.

A fashion e-tailer is identifying customers that bought a certain type of shoe between 2005 and 2008 due to the launch of a related product, and they would like to market to those customers.

The analysis is essentially the same: Find customers that did an activity during a time period.

However, the fashion e-tailer is likely to not require the same level of accuracy and completeness that the bank will require. The bank is likely to need an exact customer group with no exceptions - no customers should be incorrectly included or excluded. The fashion e-tailer is less likely to require the same thing - complete accuracy could be sacrificed in favour of speed and efficiency.

If an analyst moved from the fashion e-tailer to the bank, while they are doing incredibly similar analysis, the execution and quality control is likely to be very different. The fashion e-tailer might spend hours or even minutes on such a task, whereas the bank could spend months.

The different types of Analytics

Source: http://www.gurobi.com/resources/prescriptive-analytics

	Tools used	Limitations	When to use
Decision Analytics **Systematic, quantitative and visual approach for business decisions**	• Variety of tools • Psychology, management, & economics techniques • Decision trees	• Information overload, leading to analysis paralysis	• When risk, capital investments, and strategic business decisions are needed
Descriptive Analytics What happened and why?	• Data aggregation • Data mining	• Snapshot of the past • Limited ability to guide decisions	• When you want to summarize results for all/part of your business
Predictive Analytics What might happen?	• Statistical models • Simulation	• Guess at the future • Helps inform low complexity decisions	• When you want to make an educated guess at likely results
Prescriptive Analytics What should we do?	• Optimization models • Heuristics	• Most effective where you have more control over what is being modeled	• When you have important, complex or time-sensitive decisions to make

Notes Space