



**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 WhiteHat 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	3
The Data Lifecycle	5
Structured and Unstructured Data	7
Requirements for Data Analysis	11
Quality Issues for Data Analysis (Quality Control)	15
Data Analysis Tasks	17
Data Protection & Legal Issues	19
Data Structures	20
Database Design, Implementation, and Maintenance	22
Data Architecture	25
The Domain Context for Data Analytics	27

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

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:

1. Organise (or Concept)
 - Where is the data you are adding to the database? Where are our files? Which files are included? Are they in the right order? What columns does it have? Can we rename them to make it clearer etc. In other words, define the source of your data.
2. Structure (or Mapping)
 - Map the source data to the target (new) data structure. Define which column from source goes into which column in target. Are our files consistent and structured properly? Do they have headings that match in all files, in the same order etc.
3. Process (or Matching)
 - Take each file and process it (match it to target) i.e. putting it into the database.

Data Sources

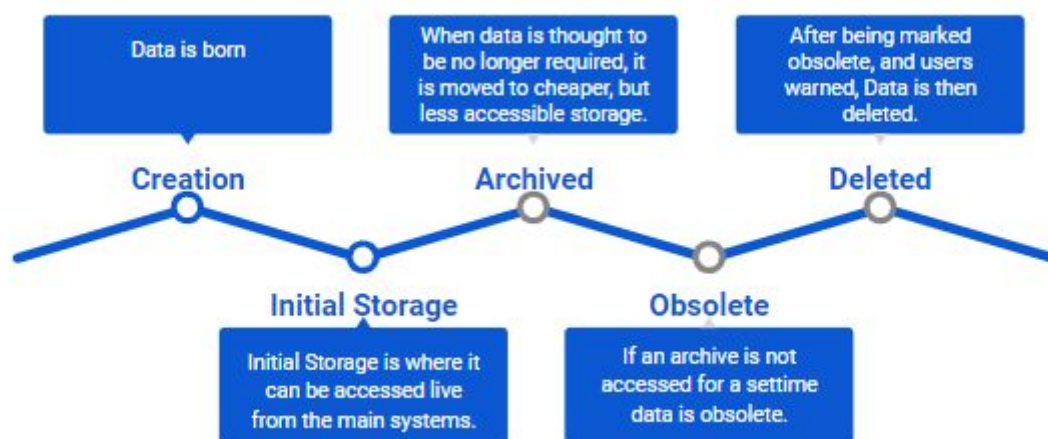
Source of Data	Definition
Public	Data that can be freely used, reused and redistributed, although hard to change or modify
Open	It is a subset of public data but has some fundamental differences. For example, open data are smaller in volume, more likely to be structured, more likely to be open licensed and well maintained and importantly more likely to be reliable as they come from sanctioned portals. Some open data require the user to pay a nominal fee. They can be freely used, reused and redistributed by anyone with no existing local, national or international legal restrictions on access or usage.
Research (or purchased)	Data from a third party that is made available to you under a licence agreement or data that data has been collected, generated or created to validate original research findings. Examples include observational data, simulation data, derived data.
Proprietary	Data that is owned and stored within an organisation. Proprietary data may be protected by patents, copyrights/trademarks or trade laws. Two subcategories of proprietary data are operational and administrative data.
Operational	Proprietary data that is produced by your organisation's 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

Data life cycle: It tells the story of data: from its original creation through to its eventual deletion. Pay attention to the following diagram which outlines what each stage involves.



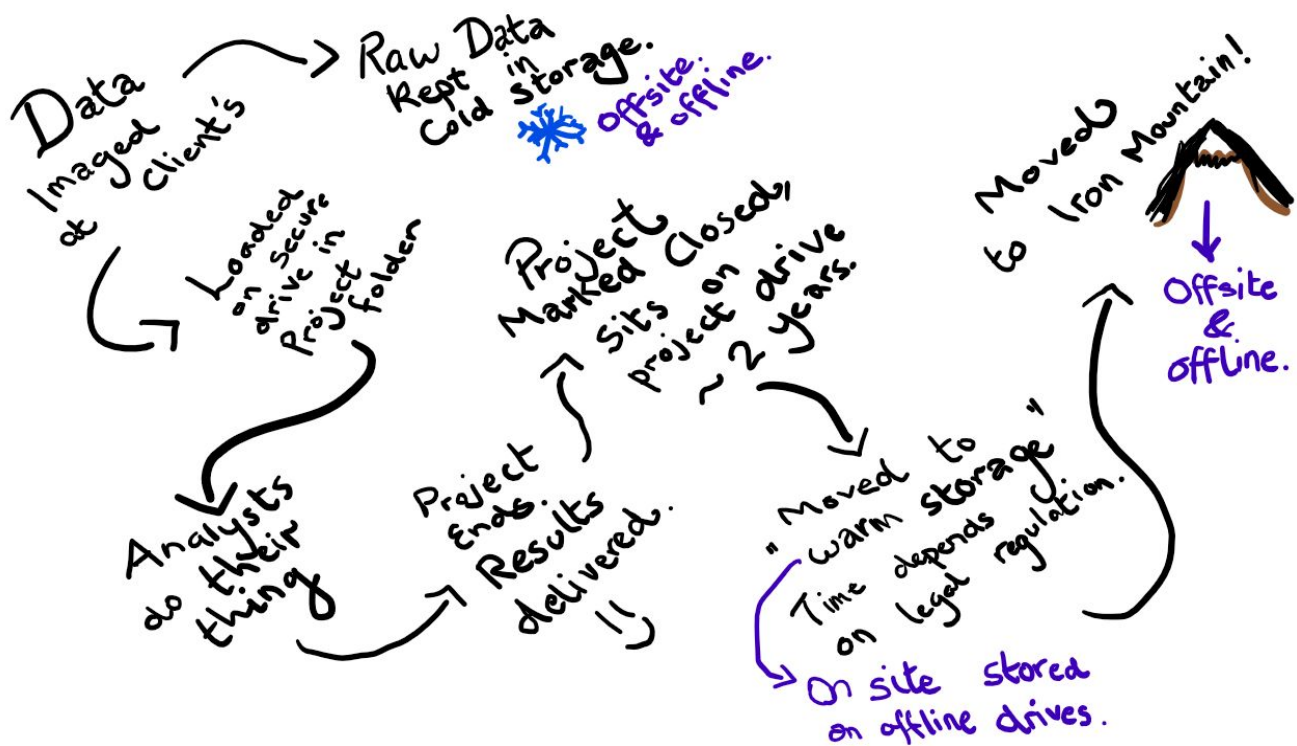
Imagine that we have a database that manages the customers accounts and subscriptions for a large broadcasting company – and that the company has been operating for over 20 years. Over this length of operation, the database will have processed and held a vast volume of information, likely at a granular 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. Once data move into the archival phase they are no longer 'live'.

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.

Note: the data lifecycle is different than the data *analytics* lifecycle which you were taught in Module I. The data analytics lifecycle is not examined and refers to the stages that you go through when completing a data analytics lifecycle.

Deloitte Forensic Example



Note Space

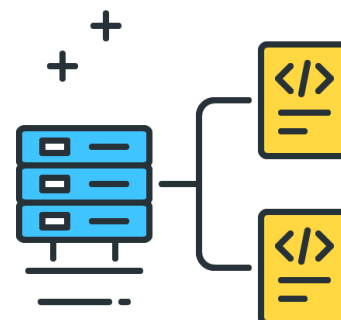
Structured and Unstructured Data

Structured Data:

Structured data is information which can be ordered and processed by data analysis tools. Structured data consist of clearly defined data types whose pattern makes them easily searchable and that typically they reside in fixed fields (e.g. in tables).

Examples:

- Tables stored within a database management system.
- Extensible Mark-up Language (XML)*
- Serially or sequentially stored data files

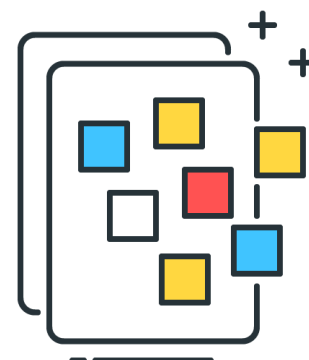


Unstructured Data:

Unstructured data is everything else -



Examples:

- Word processor and PowerPoint files. Any files that are **not** serially or sequentially stored.
- Audio
- Video
- Sensor and log data
- External data (such as social media feeds)
- Paper-based documents



*Extensible Markup Language is generally regarded as semi-structured data but for the purposes of BCS exam you should regard it as structured data.

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 attendees in an event or the number of T-shirts in stock. They are countable numbers. Always a whole number.

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. Or you can have one meter and two centimeters of tape.

Qualitative Data

Categorical - Contains a finite number of categories or groups. Examples of categorical data include Gender (e.g. male, female, other) or payment method (e.g. cash, credit card, debit card, etc). They are qualitative 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, country etc.

Ordinal - Categorical data where the order does matter. For instance, clothing sizes ranging from XS, S, M, L, XL, or the size of a coffee: small, medium, large.

Binominal - 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 however 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.

The combination of unstructured data and structured data is known in the industry as Big Data Analytics.

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	Literally useless.

redemptions for that draw	
---------------------------	--

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** (aka *general requirements* e.g. complying with GDPR)
- **Stakeholder requirements**
- **Technical requirements** (like technical specifications: e.g. using a specific DBMS or software)
- **Budget, resource & scoping constraint**

What is Verification?

Definition : *The process of evaluating a (data) project to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. It happens throughout the duration of a project.*

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

Verification is the process of checking that the (data) project meets the specification. *'Did I build the project right?'*

What is Validation?

Definition: *The process of evaluating a (data) project 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 project meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product. Validation is often used **after** training (or building) a project (model).

Validation is the process of checking whether the (data) project captures the customer's needs. 'Did I build the right project'?

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

An easy way to remember this is that **developers** (people who build the project/product) **verify** during the creation and **people who test** the final project/product, **validate**.

Examples: <https://www.slideshare.net/saad95/validation-vs-verification>

Requirements elicitation process

The elicitation process is a set of methods (techniques) that allow you to gather requirements for a project from various stakeholders.

Various methods for Elicitation requirements include:

- **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** - Use existing documentation to guide parts of the project or gather requirements. This may be in the form of data dictionaries etc (past question)

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.

Tacit knowledge (knowing-how, knowledge of experience): 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.

Note Space

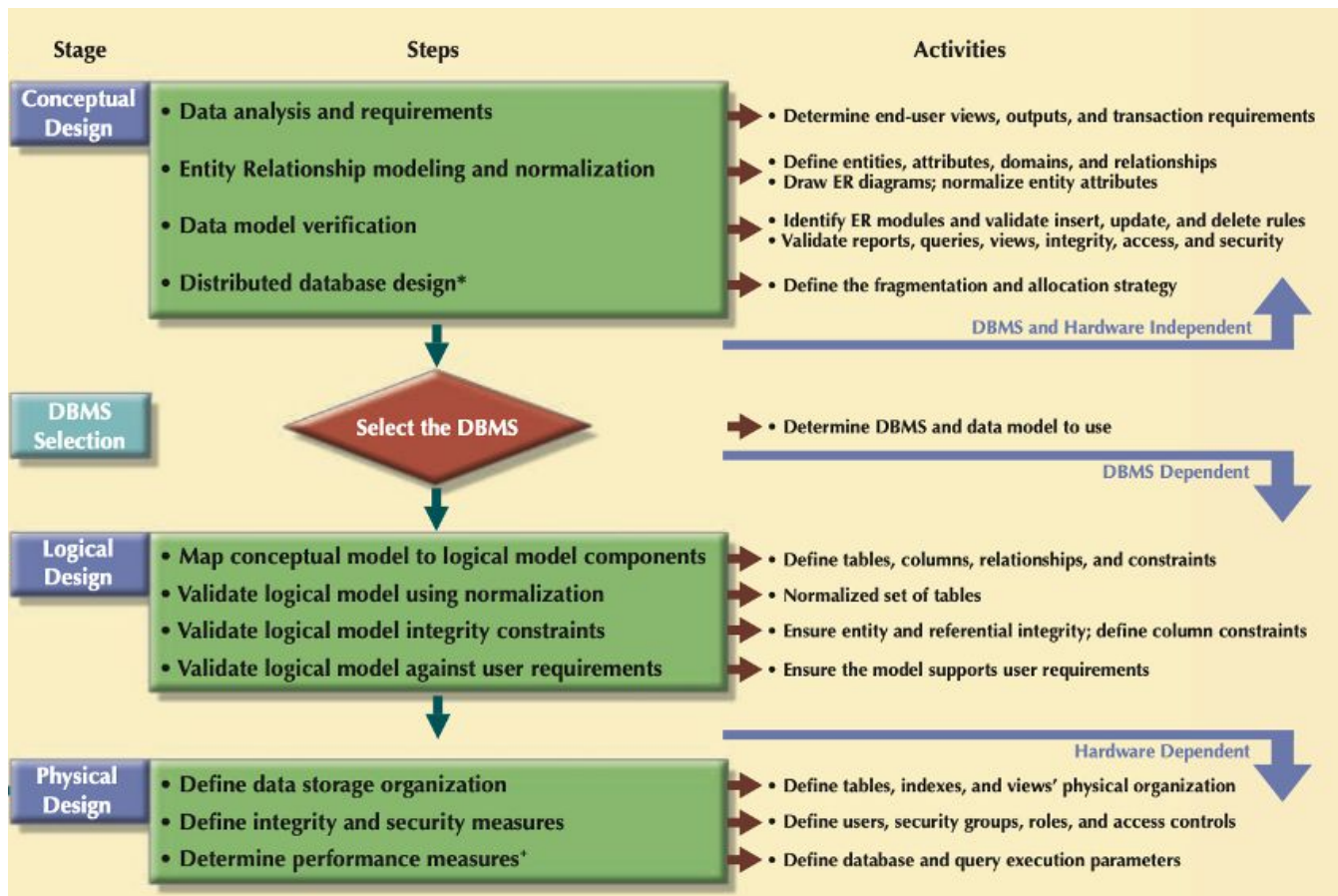
Data Modelling

Data modelling is the most common method of documenting database designs. The data model also provides an invaluable resource when maintaining or modifying a database as business requirements change. At the highest level, data modeling is simply a pictorial representation of database tables and their relations. There are mainly three different levels of data modelling:

(Source: <https://www.guru99.com/data-modelling-conceptual-logical.html>)

1. **Conceptual design (or Conceptual model):** 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 design (or Logical model):** 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 design (or Physical model):** 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.*

Here's a nice diagram that outlines the key stages of data modelling, how they related to database design and the activities involved at each stage (Source: *Coronel, Carlos, and Steven Morris. Database systems: design, implementation, & management. Cengage Learning, 2016.*):



Note Space

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.

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 way 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.

Note Space

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 in most project:

- **Problem hypothesis**

- Defining the hypothesis as well as null and alternative hypotheses
- Hypothesis is a *testable statement* associated to the problem under study. A null hypothesis assumes that your testable statement is positive (I.e. correct) whereas an alternative hypothesis is used when your testable statement negative (I.e. wrong). The hypothesis formulation is an important measure of success for our data project. It helps frame the problem and the expected outcome.

An example of hypothesis:

- **Null hypothesis:** there is **no** relationship between height and weight.
- **Alternative hypothesis:** there **is** a relationship between height and weight.

Here's some more formal definitions for hypothesis which essentially mean the same thing but in a more formal statistics language.

- 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, line graph, 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 obsolescence of the Data protection act 1998 UK by the introduction of the EU GDPR. After the United Kingdom left the European Union, the Data Protection Act 2018 is the UK's implementation of the General Data Protection Regulation (GDPR is the EU equivalent).

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 (individuals who their data is concerned), Data Users, Data Controllers.

Data subject refers to any individual person who can be identified, directly or indirectly, via an identifier such as a name, an ID number, location data, or via factors specific to the person's physical, physiological, genetic, mental, economic, cultural or social identity.

Data user is any individual who uses personal data as part of their job or role.

Data controller is a person, company, or other body that determines the purpose and means of personal data processing

Source: www.atinternet.com

Important data that are under protection

- **Personally Identifiable Information:** data that can be used to identify an individual. Such data are usually encrypted to ensure their safeguarding. Examples include biometric information, medical information, financial information and unique identifiers such as passport and Social Security numbers.
- **Protected Health Information:** data in a medical record that can be used to identify an individual, and which were created, used, or disclosed in the course of providing a health care service, e.g. during diagnosis or treatment. It may also include billing information and any patient information that a health insurance company's holds for example.

Who regulates and enforces the Data Protection Act in UK?

It is regulated and enforced by an independent authority, the Information Commissioner's Office. It is a non-departmental public body which reports to Parliament.

Note Space

Data Structures

Data structures are 'objects' that contain data. Computers use these 'objects' to store data. Each type of data structure (i.e. each type of these objects) stores data in a different way and some data structures are better suited for some applications than others. It is out of the scope of this course to know in detail how computers use data structures to store data in their memory. For this course it should suffice to know the main types of data structures and what are some of their key characteristics, given below.

Lists

A **list** contains elements of different data types. For example, a list could contain strings and numbers together. For example

For example,

('John, 10, 'George,' 21)

Array

An **array** is a data structure similar type of data which are identified by one or more indices. An array can have multiple dimensions. 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 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.

Records

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

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.

Trees

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

Tables

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

Note Space

Database Design, Implementation, and Maintenance

Forms of Databases

- Some definitions sourced from:
<https://www.c-sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/>
- A good refresher [video](#) on the types of DB

Type of DB	Examples	Description
Relational	MsSQL, Postgres, mySQL	This is the most common form of database. It contains tables that 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 <i>tree</i> 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, <i>each child record has only one parent</i> . 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 large digital computers. Network databases are hierarchical databases but unlike hierarchical databases where one node can have one parent only, <i>a network node can have relationship with multiple entities</i> . 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.) These databases are ideal for storing unstructured data (past exam question)
-------	--	--

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)

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

Normalisation and De-normalisation:

*"NORMALIZATION is a database **design** technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships".* Read more about normalisation/de-normalisation [here](#).

Normalisation enables the design of database tables that are well structured, free of data anomalies and easier to extend/scale.

A side effect of normalisation is the creation of many different tables in order to meet the various rules of normalisation. As a result, a database query may require longer to run causing delays in processing. De-normalisation is the opposite of normalisation – where instead of having many different tables, we have the data stored in larger tables. However, this may lead to more difficult data indexing, data redundancies and less efficiency. During de-normalisation logical data structures are rearranged by splitting or combining.*.

**Splitting logical data structures, Combining logical data structures or introducing redundant relationships : Meaning through an 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 prevent 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.

[Note Space](#)

Data Architecture

Data Architecture pertains to the set of:

- **Rules**
- **Policies**
- **Standards**
- **Models**

that are used within an organisation and 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 design 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.

Important processes relating to data architecture, include:

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 should 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.

Note Space

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

		 Tools used	 Limitations	 When to use
 Descriptive Analytics	<i>What happened and why?</i>	<ul style="list-style-type: none"> › Data aggregation › Data mining 	<ul style="list-style-type: none"> › Snapshot of the past › Limited ability to guide decisions 	<ul style="list-style-type: none"> › When you want to summarize results for all/part of your business
 Predictive Analytics	<i>What might happen?</i>	<ul style="list-style-type: none"> › Statistical models › Simulation 	<ul style="list-style-type: none"> › Guess at the future › Helps inform low complexity decisions 	<ul style="list-style-type: none"> › When you want to make an educated guess at likely results
 Prescriptive Analytics	<i>What should we do?</i>	<ul style="list-style-type: none"> › Optimization models › Heuristics 	<ul style="list-style-type: none"> › Most effective where you have more control over what is being modeled 	<ul style="list-style-type: none"> • When you have important, complex or time-sensitive decisions to make

In addition to the above three types of analytics, you should familiarise yourself with **Decision analytics** for the purposes of the exam.

Decision analytics ([source](#)):

- Decision analysis is a systematic, quantitative, and visual approach to making strategic business decisions.
- Decision analysis uses a variety of tools and also incorporates aspects of psychology, management techniques, and economics. Decision trees and influence diagrams are visual representations that help in the analysis process. **(Tools used)**
- Risk, capital investments, and strategic business decisions are areas where decision analysis can be applied. **(When to use)**
- Critics argue that decision analysis can easily lead to analysis paralysis and, due to information overload, the inability to make any decisions at all. **(Limitations)**

Notes Space