VCE Applied Computing: Units 3 and 4 Data Analytics: Software tools and functions and outcome-specific requirements

The VCE Applied Computing Study Design (From 2025) mandates software tools and functions and outcome-specific requirements that students are to follow in Unit 3 and 4 Data Analytics. For 2025, schools must use these software tools and functions and outcome-specific requirements in the study.

Government schools are advised to refer to the Department of Education’s Generative Artificial Intelligence policy at: [Generative Artificial Intelligence: Policy | education.vic.gov.au](https://www2.education.vic.gov.au/pal/generative-artificial-intelligence/policy).

Catholic and independent schools should refer to their sector authorities for advice on generative artificial intelligence.

Teachers of VCE Applied Computing should note that the Software tools and functions and outcome-specific requirements may be revised for 2026 and notification will be published in the [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/news-and-events/bulletins-and-updates/bulletin/Pages/index.aspx).

Software tools and functions

The following software tools and functions are outlined in the VCE Applied Computing Study Design (From 2025) for Unit 3 and 4 Data Analytics:

* Unit 3 Area of Study 1 – Data analytics (Page 37)
  + Students are required to both study and use
    - database software, spreadsheet software and data visualisation software.
    - an appropriate tool for running Structured Query Language (SQL) queries.
* Unit 3 Area of Study 2 – Data analytics: analysis and design (Page 40)
  + Students are required to use, but are not required to study:
    - an appropriate tool for documenting and modifying project plans and
    - appropriate tools for ideation and generating designs.
* Unit 4 Area of Study 1 – Data analytics: development and evaluation (Page 44)
  + Students are required to both study and use
    - database software, spreadsheet software and data visualisation software.
    - an appropriate tool for running Structured Query Language (SQL) queries.
  + Students are required to use, but not required to study:
    - an appropriate tool for documenting and modifying project plans.

Software functions

The following is a list of software functions for each of the software tools that are studied and used, and that students are expected to be able to apply. Note that this list is not exhaustive; learning does not have to be confined to the functions listed.

For assessment purposes, students must be familiar with all of the listed functions for the mandated software tools.

Database software

Students are expected to apply functions that provide the ability to:

* create tables
* create relationships between tables (for RDBMS) using SQL
* use a range of data types
* validate data
* create, edit and use queries using SQL
* search and filter records using SQL and Boolean operators where necessary
* sort records or index on different fields using SQL
* import and export data
* secure data.

Spreadsheet software

Students are expected to apply functions that provide the ability to:

* create worksheets
* create links between worksheets
* use a range of data types
* validate data
* create, edit and use charts
* calculate descriptive statistics using appropriate functions
* calculate Pearson’s correlation coefficient (*r*)
* use LOOKUP functions (XLOOKUP, VLOOKUP, HLOOKUP)
* apply conditional formatting
* sort and filter data
* import and export data
* secure data.

Statistical analysis

The following techniques are used to statistically analyse data to identify trends, relationships and patterns:

1. Descriptive statistics (average, median, minimum, maximum, range, standard deviation, count/frequency, sum).

Data analysis involves the processes and tools that allow organisations to acquire and/or extract data in various forms, update erroneous or incomplete data, analyse the data to identify trends, relationships and patterns, draw inferences about the data and present findings using visual methods that provide clear and unambiguous conclusions.

As such meaningful data needs to be collected and analysed. Descriptive statistics are a collection of measures that can be used to provide a summary of a data set. These can be referred to as measures of central tendency (average, median) or measures of variability and spread (maximum, minimum, range, frequency, standard deviation).

Descriptive statistics calculation formulas used in common spreadsheet software are used below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Excel | Google Sheets | Numbers |
| average (arithmetic mean) | AVERAGE | AVERAGE | AVERAGE |
| median | MEDIAN | MEDIAN | MEDIAN |
| minimum | MIN | MIN | MIN |
| maximum | MAX | MAX | MAX |
| range | No formula directly calculates range. Calculated using MAX-MIN | No formula directly calculates range. Calculated using MAX-MIN | No formula directly calculates range. Calculated using MAX-MIN |
| standard deviation (of sample) | STDEV.S | STDEV.S | STDEV |
| count (frequency) | COUNT | COUNT | COUNT |
| sum (subtotal) | SUM | SUM | SUM |

Note:

Further refinement of formulas can be completed, for example, including all values including non-numerical would use the formula AVERAGEA, MINA, MAXA, COUNTA.

Formulas that include IF/IFS (multiple IF) statements built in and are listed with descriptive statistics include: AVERAGEIF, AVERAGEIFS, MINIFS\*, MAXIFS\*, COUNTIF, COUNTIFS, SUMIF, SUMIFS.

\*The MINIFS and MAXIFS formulas requires a version later than 2019, or Microsoft 365 version for Microsoft Excel. Available in Google Sheets. Not available in Numbers.

1. Pearson’s correlation co-efficient (*r*).

When working with multiple points of data, linear regression is a method that can be used to describe the impact of one variable on another. By calculating Pearson’s correlation coefficient (*r*), a description can be made about whether the relationship is perfect/strong/weak positive, perfect/strong/weak negative, or if there is no correlation at all between the values for each variable.

Calculation formula for Pearson’s correlation co-efficient (*r*):

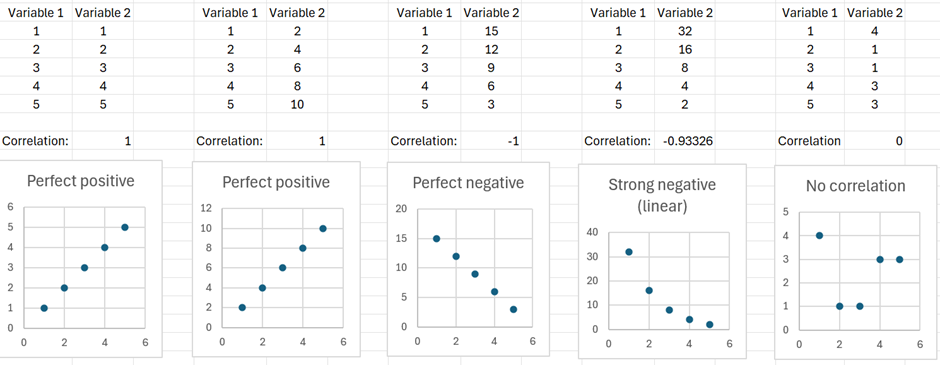
|  |  |  |  |
| --- | --- | --- | --- |
|  | Excel | Google Sheets | Numbers |
| *r* | PEARSON or CORREL | PEARSON or CORREL | CORREL |

Input parameters are the ranges for the variables to be compared.

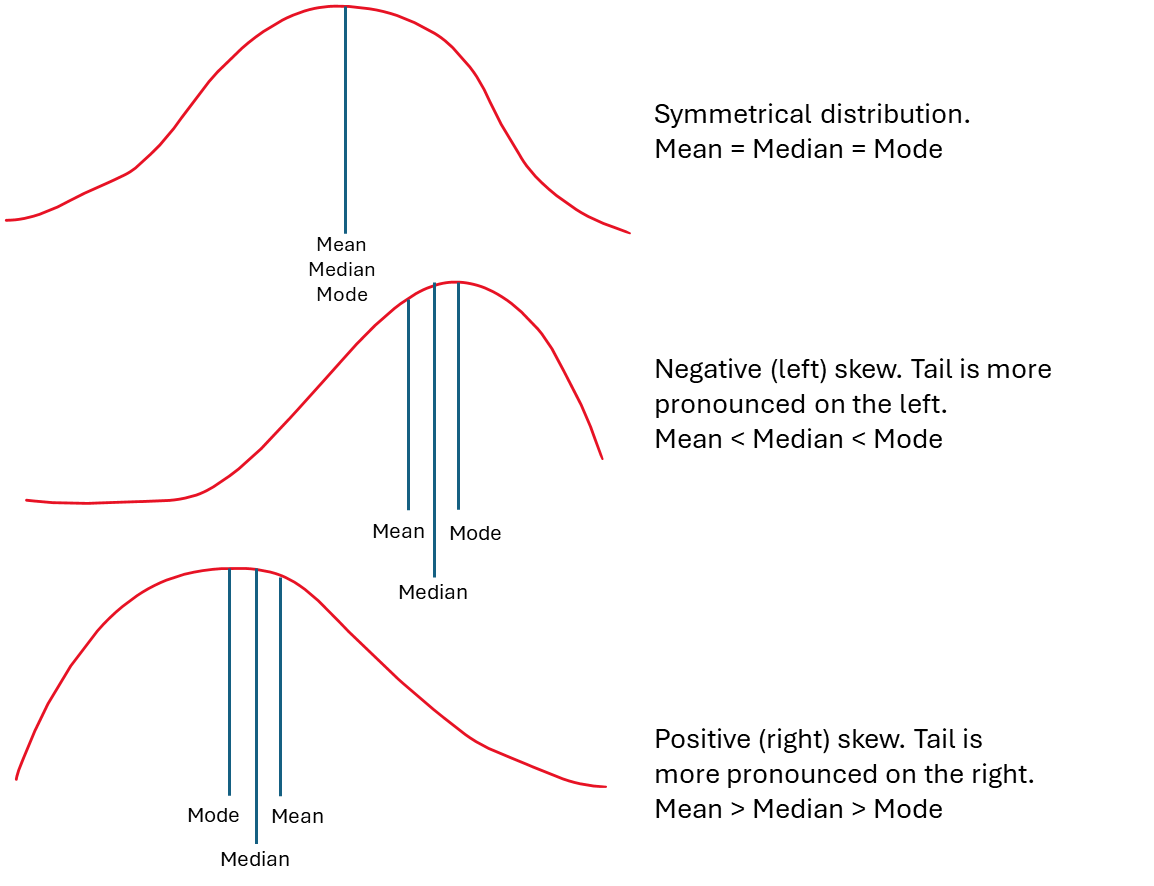
Pearson’s correlation coefficient, *r* is a measure that describes the strength and direction of a linear relationship between two numerical variables. It can be used as a descriptive statistic that shows when one variable changes, the impact that occurs on the other variable.

A perfect positive linear correlation occurs when the value of *r* is equal to 1.

A perfect negative linear correlation occurs when the value of *r* is equal to -1.

No linear correlation occurs when the value of *r* is equal to 0.

1. The shape and skew of data.

When working with distributions of data it is important to be able to describe the shape and skew of the data, from being normally or symmetrically distributed around the mean, or positively or negatively skewed.

Data visualisation software

Students are expected to apply functions that provide the ability to:

* create/select a range of shapes
* create/select a range of chart types
* use symbols/images/charts
* show trends, relationships and patterns
* enter, edit and format text and other content
* use colour/shading
* use of animation in charts (showing movement or changes in response to user input).

In the development of dynamic data visualisations, the chosen data visualisation software tool should provide users with the ability to interact with the data visualisation in order to identify meaning.

Types of data visualisations

The following is a list of the types of data visualisations:

1. Infographics (series or long-form, static).

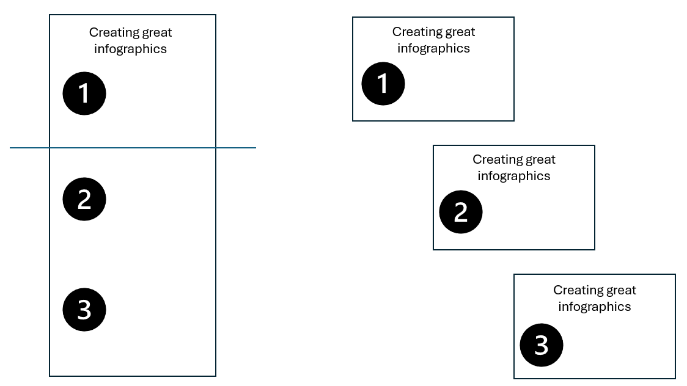
Infographics are graphical representations of complex data or information that provide an overview of a topic or area of interest. They rely on combining visual elements to communicate data patterns or trends quickly and clearly or information as data visualisations. These include complementary colour schemes, easy-to-read text fonts, headings, multiple graphs, simple charts and statistics.

A series of infographics allow different viewpoints of data (for example the same topic, but for different time periods) to be shown in sequence.

A long-form infographic combines charts, images and appropriate text below the scroll area of a screen and can be used to the same effect as a series of infographics, but within the same core document.

Infographics are designed to be static, without data changes or interactive capability. They can be printed physically and displayed for public viewing, but they can also be designed for online view.

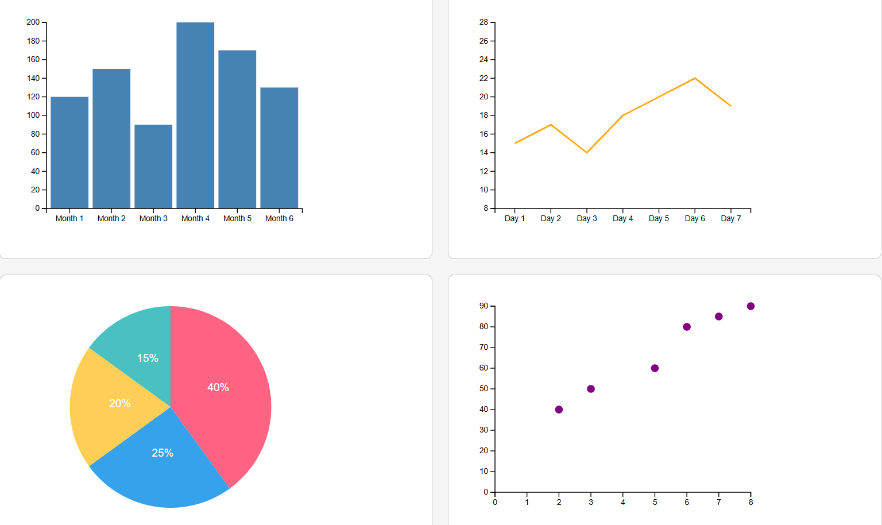
On the left is an example of a long form infographic, with multiple sections, designed for a reader to scroll past the bottom of the page (indicated by a horizontal line).

On the right is an example of a series of infographics on the same topic.

1. Dashboards (interactive, static or live data).

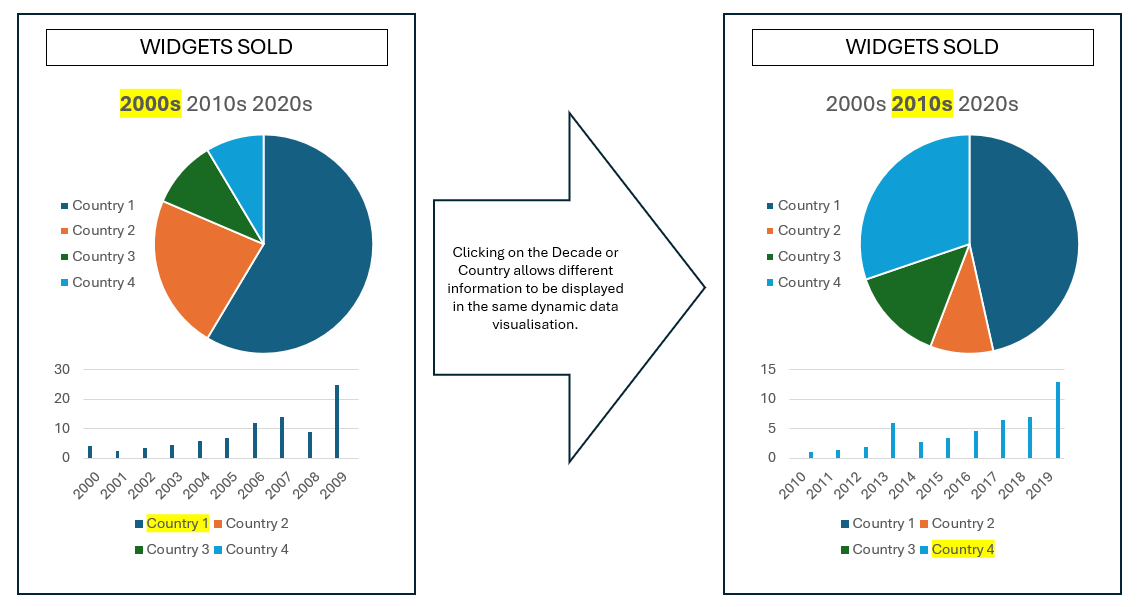
A dashboard is an information management data visualisation that can display various types of data in one screen or page. Data is often linked with either a database or spreadsheet.

Dashboards can have interactive elements where users can interact with the data being displayed, with an example being the use of filters. However, dashboards need not require interaction.

The data used by dashboards can either be static, informed by a point-in-time, or being updated by live data. This can be completed in either an automated or manual fashion.

1. Dynamic data visualisations (interactive, live data).

Dynamic data visualisations are graphical representations of complex data or information. They allow the exploration of data in an interactive way.

In contrast to a static visualisation or chart, a dynamic data visualisation contains data that can change in response to user interaction or the addition of live data.

Outcome-specific requirements

The Outcome-specific requirements for Unit 3 and 4 Data Analytics provide specifications and scope for the listed key knowledge dot points in this document.

Unit 3: Emerging trends in data analytics using AI

The following is a list of emerging trends in data analytics using artificial intelligence:

1. Integration of artificial intelligence features into software tools.

Software tools can include artificial intelligence that uses generative functions to locate and analyse data, write information and create images.

Some examples:

* chatbots and virtual assistants
* use of voice recognition systems
* content creation and design tools embedded into products
* code generation and debugging.

1. Generating data visualisations through the writing and refinement of prompts.

A prompt can be given as a natural language text input, or instruction to a large language model through a platform and is answered using artificial intelligence. By refining a prompt or using a sequence of instructions, users can create data visualisations quickly.

There are disadvantages in the use of prompts, which can include errors in the interpretation of instructions, the need to fine-tune answers to meet specific user needs and the usability of end solutions.

Some examples:

* use of natural language text input
* real-time changes with refined prompts
* automation of insights.

An example of a prompt that could create a template for a data visualisation could be:

*Create a data visualisation using d3.js that looks like an online dashboard with four charts in each corner, a column chart, a line chart, a pie chart and a scatter chart. Start with random values.*

1. Machine learning and statistical modelling for making predictions, decisions and recommendations.

Machine learning is a process within the artificial intelligence field that aims to develop and refine statistical algorithms that learn and perform tasks without direct instructions. The output of previous instructions can be used to inform future learning.

Statistical modelling is a mathematical process that aims to use hypothesis testing and correlations to be able to make predictions of the output that would occur from future samples of data.

Both machine learning and statistical modelling can be used to predict events that have not occurred, but for which there is data that can be analysed. The way this is done is either with limited supervision (machine learning) or explicit supervision (statistical modelling).

Some examples:

* self-learning systems in robotics and automotive industries
* decision support systems in finance and healthcare
* the use of urban planning data to manage traffic flow.

Unit 3: Ideation techniques and tools

The following is a list of ideation techniques and tools for generating design ideas:

1. Mood boards.

Mood boards are used to collate imagery, colours, typography and code samples to explore and inspire the overall direction of the design of the software solution.

Samples may be annotated to indicate potential applicability to the solution, specific features to be/not to be included, as well as reasoning why identified elements should/shouldn’t be applied within the final design.

A sample mood board is found below:

A diagram of a computer program

Description automatically generated with medium confidence

1. Brainstorming.

Brainstorming results in an unstructured collation of related ideas, based around a central theme/concept.

Brainstorming can be documented as a list of ideas, but should not be restricted in format, nor requiring correct punctuation or proper sentences.

The aim of brainstorming is to write as many ideas as come to mind on the central theme. These ideas are kept on a single page and can allow any of these ideas to be built on in future.

Examples of brainstorming are found below:

|  |  |
| --- | --- |
| Design   * Design principles * colour * alignment * storyboard * interactivity * Design tools * mock-ups * query designs * usability |  |

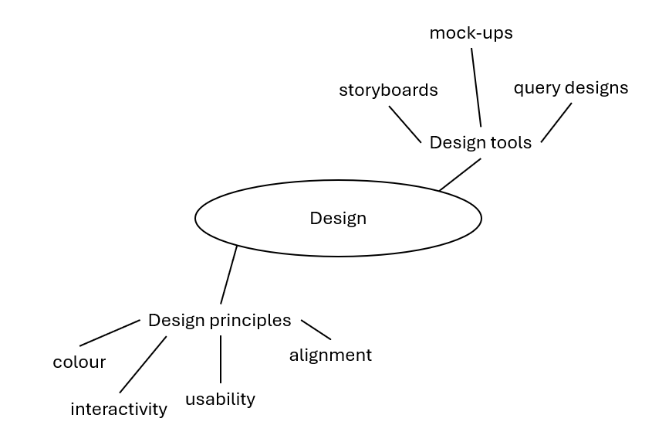
1. Mind maps.

Mind maps are a structured collation of related ideas, based around a central theme/concept, represented as a diagram. Mind maps can build on the ideas created using brainstorming. They can help to visually organise information.

The central theme/concept is represented using a rectangle, cloud-shape or ellipse.

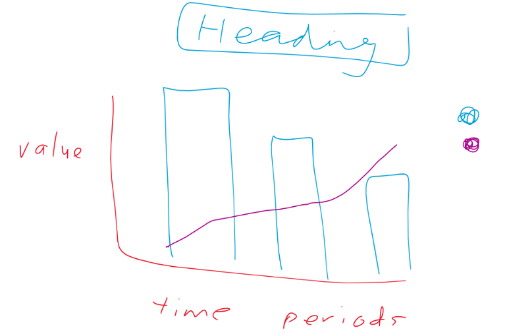
Related ideas (as words) are connected to the central theme/concept using straight lines. Ideas branch out further as required.

A sample mind map is found below:



1. Sketches.

Sketches are unrefined hand-drawn illustrations that briefly depict a visualisation. They can be simply created with a pencil and paper, or could include colour, particularly when considering chart elements.

A sample sketch is shown below:

1. Annotations.

Annotations are attached to a layout diagram, mock-up, storyboard or sketch to provide details or clarification. Annotations commonly include labels, arrows and explanatory notes. These can be added digitally or handwritten.

Annotations may also be used to explain how design principles have been incorporated, or to justify design choices that have been made.

Unit 3: Design tools for representing databases and spreadsheets

The following is a list of design tools for representing databases and spreadsheets:

1. Data dictionaries.

Data dictionaries are used to design the data requirements for a database table. This helps to setup fields in a database table for the inclusion of data as records. There should be a separate data dictionary created for each database table.

Data dictionaries are represented using a table with the following fields included as a minimum:

* + Name
  + Data type
  + Description

Additional fields may be included, as required, such as:

* + Field size
  + Validation rules, or a list of acceptable values
  + Key field information
  + Data sample

A sample data dictionary is found below:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| customer\_name | String | Given name of a customer |
| customer\_address | String | Address of the customer |
| customer\_age | Integer | Age of the customer, in years. |

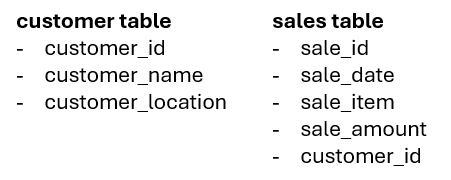
1. Query designs.

Query designs help to define the method to extract data from a structured database. They can be created using text (Query design examples 1 and 2 – below) or a tabular format (Query design examples 3 and 4 – below) and will contain at minimum the data that needs to be retrieved (fields), and where the data needs to be retrieved from (tables).

Optionally, query designs also contain clauses that define criteria that the data needs to meet (which can include Boolean operators – AND/OR/NOT – as well as the order that the final data is sorted by). Query designs can also contain joins that link more than one database table together by specifying the linking fields within each table.

Query designs do not need to be created in SQL as this will be created in Unit 4 Area of Study 1 as part of the development of the database solution.

Query design examples:

Database structure:

Query design 1 – Extract names of all customers

Fields: customer\_name

Table/s: customer

Criteria: N/A

Sort Order: N/A

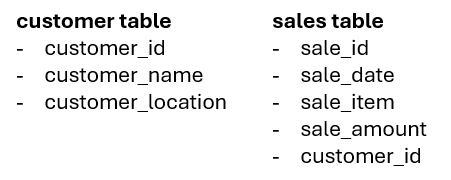
Query design 2 – Extract customer details from all customers from Victoria

Field/s: customer\_id, customer\_name, customer\_location

Table/s: customer

Criteria: customer\_location = “Victoria”

Sort Order: N/A

Database structure:

Query design 3 – Extract sales details for hammers ordered by customer ascending.

|  |  |
| --- | --- |
| Field/s: | sale\_id, sale\_date, sale\_item, sale\_amount, customer\_id |
| Table/s: | sales |
| Criteria: | sale\_item = “hammers” |
| Sort order: | customer\_id ASC |

Query design 4 – Extract all sales of hammers from customers in Victoria or Queensland after 20/01/2025. Order these sales items by customer (ascending), then by sales amount (descending).

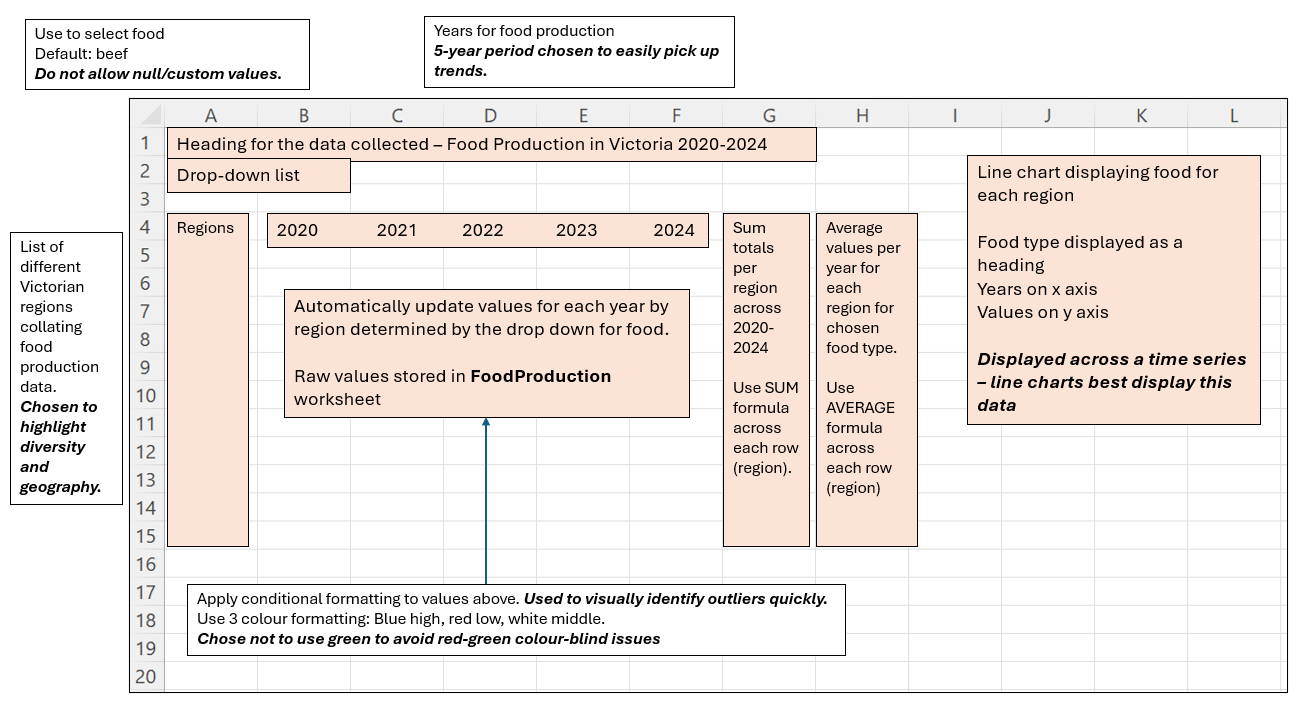
|  |  |
| --- | --- |
| Field/s: | customer\_id, customer\_location (customer table), sale\_date, sale\_item, sale\_amount (sales table) |
| Table/s: | customer, sales |
| Criteria (1): | customer\_location = “Victoria” OR customer\_location = “Queensland” |
| Criteria (2): | sale\_item = “hammers” |
| Criteria (3): | sale\_date > “20/01/2025” |
| Sort order: | customer\_id ASC |
| Join | customer and sales table on customer\_id field |

1. Layout diagrams.

A layout diagram is a detailed diagram that demonstrates how different elements of a spreadsheet will be displayed. A key identifying feature of a layout diagram is not including the precise or expected values but including elements as placeholders on a page.

Annotations are used to describe appearance (font, text colour, text size, alignment/justification, location details) and structure (required functionality including formulas to be used, default values).

Annotations may also be used to explain how design principles have been incorporated, or to justify design choices that have been made.

An example of a layout diagram for a spreadsheet is shown below:

1. Input-process-output (IPO) charts.

IPO charts provide a brief overview of the inputs and outputs of a task/process, and outline the transformations (processes) required to derive the output/s from the input/s.

IPO charts can be represented in a table, with columns for Input, Processes and Output.

Separate IPO charts should be used when multiple tasks are being represented.

Processing tasks could include:

* + Data cleansing – remove duplicate values, account for missing records.
  + Coding data – grouping qualitative data for qualitative analysis.
  + Merging data – linking data tables.
  + Data analysis – calculation of descriptive statistics for as part of exploring the data.
  + Reformatting – transforming data to work in a spreadsheet/data visualisation.

A sample IPO chart is found below:

Calculate the average test score for all students in a year level, grouped by class.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Student test score  Student class | Use AVERAGEIF function to calculate scores per class | A list of the average scores for all students in each class |

Unit 3: Design tools for representing data visualisations

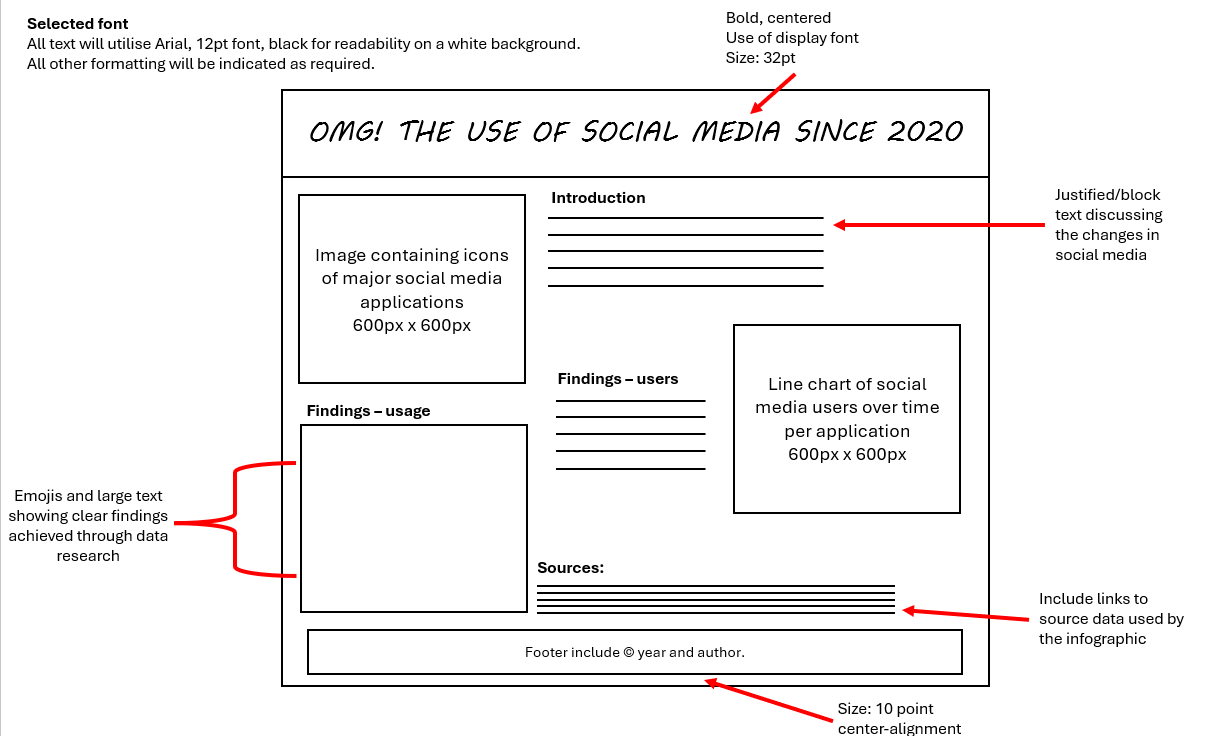
The following is a list of design tools for representing data visualisations:

1. Mock-ups.

A mock-up is a detailed diagram that illustrates what a software solution, in this case a data visualisation, will look like. These mock-ups should display the headings, text, charts, images and objects that are to be included in the solution, along with their properties.

Annotations are used to describe appearance (font, text and background colours, text size, alignment/justification, images to be sourced, chart properties) and structure (interactive features, visual navigation tools, default values). Annotations may also be used to explain how design principles, or to justify design choices that have been made.

A sample mock-up is shown below:



1. Storyboards.

Storyboards contain multiple drawings or images that are sequenced. They are a design tool used to capture change when viewing or interacting with a software solution. For dynamic data visualisations, storyboards can capture the initial state of the data visualisation and how the data visualisation will change.

Optionally, the interaction that can be launched by the user can also be displayed as part of a storyboard.

A screenshot of a graph

Description automatically generatedAn example of a storyboard for a dynamic data visualisation where the data is moving over time across the screen is shown below:

Units 3 and 4: Key legislation

The following is a list of key legislation:

1. *Copyright Act 1968* (Cwlth)

For more information go to: Federal Register of Legislation: <https://www.legislation.gov.au/C1968A00063/2019-01-01/text>.

1. Health Records Act 2001
   * Unit 3 Area of Study 2: HPP 1, 2, 4
   * Unit 4 Area of Study 2: HPP 1, 2, 4, 5

For more information go to: Health Records Act:

<https://www.health.vic.gov.au/legislation/health-records-act>.

1. *Privacy Act 1988* (Cwlth)
   * Unit 3 Area of Study 2: APP 3, 4, 5, 6, 11
   * Unit 4 Area of Study 2: APP 1, 3, 4, 5, 6, 7, 11

For more information go to: Federal Register of Legislation:

<https://www.legislation.gov.au/C2004A03712/2019-08-13/text>.

1. Privacy Amendment (Notifiable Data Breaches) Act 2017
   * Unit 4 Area of Study 2: Sections 26WE, 26WF, 26WH, 26WK, 26WL, 26WR

For more information go to: Federal Register of Legislation:

<https://www.legislation.gov.au/C2017A00012/latest/text>.

1. Privacy and Data Protection Act 2014
   * Unit 3 Area of Study 2: IPP 1, 2, 4, 10
   * Unit 4 Area of Study 2: IPP 1, 2, 4, 5, 10

For more information go to: Your privacy rights:

<https://ovic.vic.gov.au/privacy/for-the-public/your-privacy-rights/>.

Unit 4: Methods to manipulate data using software tools

The following is a list of effective and efficient methods to manipulate data using software tools:

1. Use of templates.

Templates are a method that can be used to create data visualisations more quickly. It is important for users to remember to not simply copy a template and someone else’s work, but to use the template to help structure their own work to meet their own specific requirements.

Templates can provide guidance on complementary colour schemes, ensure that fonts are easy to read, and demonstrate layouts that allocate space appropriately to help make data clearer to interpret. They may include a theme for which individual properties can be adjusted.

As users become more experienced, their use of templates reduces as their initial designs are written precisely to satisfy the needs of their target audience.

Examples of templates include specific chart templates, timelines, processes, comparisons, reports, social media posts and initial background images.

1. Software functions.

Software functions can be as simple as sorting or filtering data, or more advanced by removing or highlighting duplicate values.

Software functions can be pre-defined by the software tool itself, and the use of these functions is regulated. These can be accessed by clicking on a menu item and choosing the function that is needed, for example applying conditional formatting to a section of a spreadsheet. They can also be initiated by writing a formula which runs a specific pre-written program across the data. A SUM function is commonly used to calculate a subtotal of values within a specific range of a spreadsheet application.

Software functions can also be created by a user to achieve a set purpose and therefore be customised, for example the creation of stored procedures or macros. These functions can be used for repetitive tasks to ensure efficient and effective processes.

1. Use of artificial intelligence tools to represent data and information.

Increasingly, data can be analysed using artificial intelligence tools to extract meaning.

Spreadsheet applications are providing opportunities to explore and analyse highlighted data, by providing statistical calculations of key values such as frequency, average, median and adding totals, as well as creating charts automatically, or suggesting recommended charts for the data that is highlighted within the application.

Unit 4: Emerging trends in cyber security

The following is a list of emerging trends in cyber security:

1. The use of artificial intelligence to protect data.

Artificial intelligence can be used to identify threats that will impact on the data integrity of a person or organisation. This can be achieved by analysing if system data contains similar properties or values to what has occurred in a previous threat. Automated responses are provided to users who have made recent authentication changes, credit card usage can trigger a response that blocks a card and protects a user.

Each of the following cyber security areas have established practices which are being updated by artificial intelligence: authentication procedures, threat detection and response, analytics of user behaviour and privacy protection.

1. Authentication procedures.

Authentication procedures prove the identity of a user or process.

Trends:

* Moving away from the use of written passwords.
* Use of Short Message Service (SMS) or one-time password (OTP).
* Use of geolocation in authentication.

1. Threat detection and response.

Threat detection represents the identification of security risks. Responses to threat detection can include blocking users, detecting and stopping malware and malicious behaviour occurring on a network. These responses can be manual or automated.

Trends:

* Identifying patterns (through artificial intelligence) more than using rule-based systems.
* Designing cloud-native security solutions.
* Threat hunting – taking a proactive approach.

1. Analytics of user behaviour.

User behaviour represents the different interactions that a user can have within a system. Analysis of this behaviour can help identify behaviours that are abnormal and can flag a threat within a system.

Trends:

* Collection of user data to build expected behaviour profiles.
* Detection of insider threats.
* Use of machine learning to detect anomalies in user behaviour.

1. Privacy protection.

Privacy protection can be defined as processes and policies that prevents the identification of a person, their activities, and their beliefs.

Trends:

* Regulations for data collection increasingly require transparency.
* Inclusion of biometric data in personally identifiable information.
* Greater resources placed on protecting children’s right to privacy, age limits.

Unit 4: Cryptographic techniques

The following is a list of cryptographic techniques to protect data at rest and in transit:

1. Use of ciphers.

Ciphers are used in the process of encoding data and information from a plain text format. This is done to protect data and information from being compromised. A cipher is an algorithm that is used to either encrypt (scramble) or decrypt (make readable, translate) data.

Two common types of ciphers are substitution (e.g. Caesar and random substitution) and polyalphabetic (e.g. Vigenère). These types of ciphers can be deciphered with the assistance of a computer and should not be used in everyday programming or data protection.

Examples of these ciphers are:

* Caesar cipher

Shifts letters in the message by a set number of positions down the alphabet.

The word: HELLO moved one position would be encrypted as IFMMP. H moved one position = I, E moved one position = F, L moved one position = M, and O moved one position = P.

Decryption is made knowing the number of positions moved (1 to 25), or through brute force viewing of all alternative messages.

* Random substitution cipher:

Substitutes letters for a random letter of the alphabet. Each letter does not map sequentially as in a Caesar cipher but is substituted randomly to the same letter consistently through the message.

The word: HELLO could use these substitutions: H=A, E=Z, L=C, O=Y and be encrypted as AZCCY.

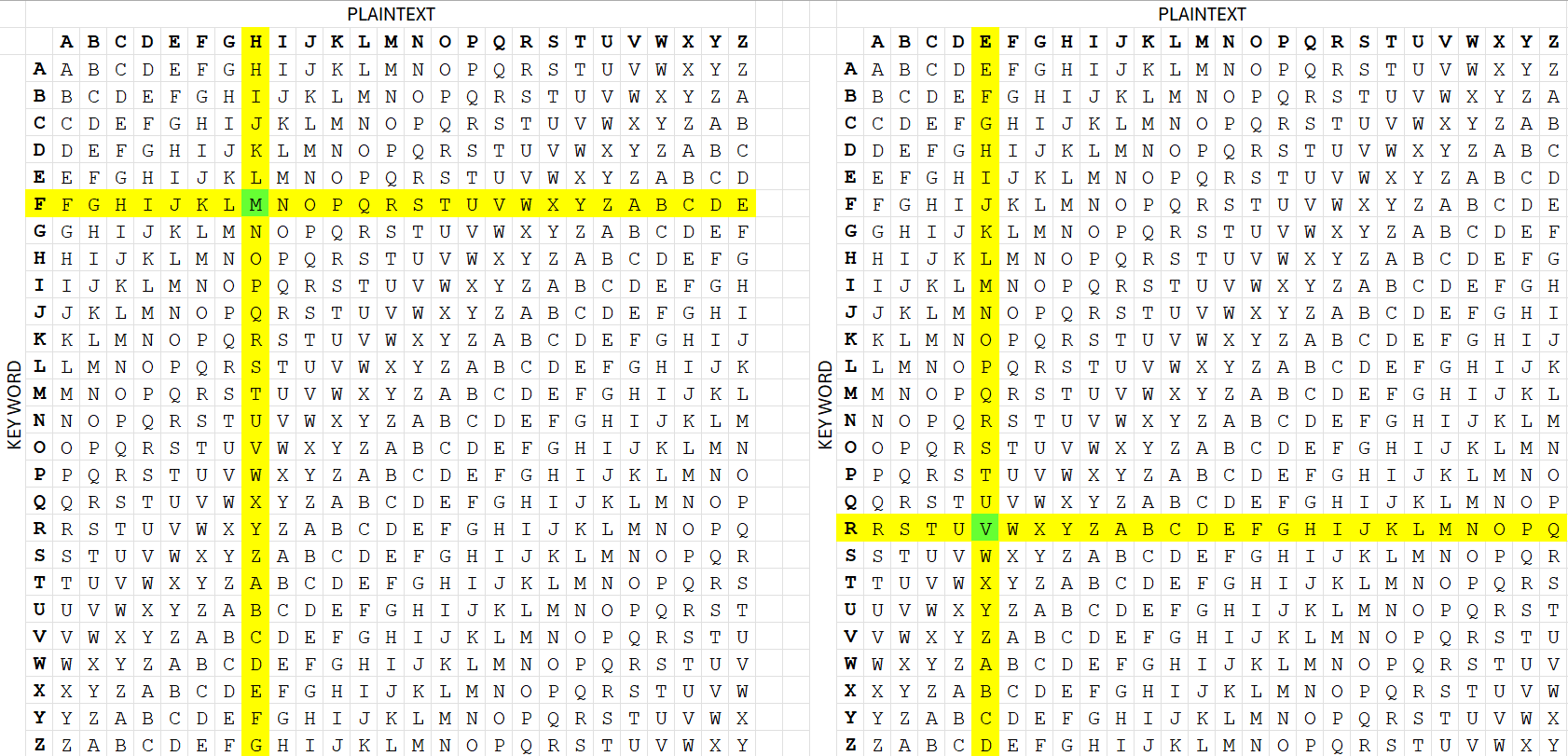
Decryption is made knowing all of the different substitutions for each letter, or by using frequency analysis particularly with larger messages using the theory that the more frequently a letter is used in a language, that letter should also appear in the encrypted text similarly frequently.

* Vigenère cipher:

Shifts letters with the use of a key word. This cipher uses a different Caesar cipher for each letter of the original text by referring to a key word and using a table to reference a letter.

An example of the encryption and decryption process follows:

The phrase HELLO WORLD could be encoded using a key word of FRIEND.



|  |  |
| --- | --- |
| Plaintext | HELLO WORLD |
| Key word | FRIEN DFRIE |
| Encryption | MVTPB ZTITH |

Decryption can occur if the key word is known, using the table, reversing the above process.

Use the letter F (the first letter of the key word FRIEND) as the key word row, find the letter M (encrypted word first letter) and follow the column up to the top to find the plaintext letter H.

Use the letter R (second letter of the key word FRIEND) as the key word row, find the letter V (encrypted word second letter) and follow the column up to the top to find the plaintext letter E.

Vigenère ciphers are hard to decrypt without knowledge of the key word, or without the use of computing power.

1. Symmetric and asymmetric keys.

Symmetric encryption involves the use of a single key for the encryption and decryption of data. A common standard is Advanced Encryption Standard (AES).

Asymmetric encryption involves the use of a public key for the encryption of data and information and a private key for the decryption of the data and information. A common standard is Rivest-Shamir-Adleman (RSA).

These encryption methods ensure the security of data, helping to protect privacy and safeguard sensitive information in everyday activities, such as online banking, secure email, and use of the internet.

1. Use of hashing functions.

A hashing function is used in cryptography to take input of any length and use an algorithm to change the input to a binary string of a fixed length of bytes. The same input will result in the same output and the process is expected to be one-way, with the resulting hash value not being able to be reversed.

A common use of a hashing function is to store user passwords. The password is entered in plaintext, hashing functions are used to convert the password into a format that is not only unreadable, but not able to be used if a hacker was able to access the password. To verify that a user has the correct password, the same hashing function is used, and if the output is the same, then the user login attempt is viewed as successful. Hashing functions help to ensure that data breaches are less common, as a user’s authentication passwords are not stored in plaintext.