

**Internal Assessment Resource**

**Digital Technologies | Hangarau Matihiko**

**Level 1**

This resource supports assessment against Achievement Standard 91879

**Standard title:** Develop a digital outcome to manage data

**Credits:** 4 Credits

**Resource title: This is CRUD**

| This resource:   * Clarifies the requirements of the achievement standard * Supports good assessment practice * Should be subjected to the school’s usual assessment quality assurance process * Should be modified to make the context relevant to students in their school/kura environment and ensure that submitted evidence is authentic |
| --- |

| Authenticity of evidence | Teachers/Kaiako must manage authenticity for any assessment from a public source, because students may have access to the assessment schedule or student exemplar material.  Using this assessment resource without modification may mean that students’ work is not authentic. The teacher/kaiako may need to change figures, measurements or data sources or set a different context or topic to be investigated or a different text to read or perform.  All code must be written by the students and a “docstring” at the top is recommended.  Students must write their report in their own words and reference any quotes accordingly. |
| --- | --- |

**Internal Assessment Resource**

**Achievement standard:** 91879

**Standard title:**  Develop a digital outcome to manage data

**Credits:** 4 credits

**Resource title: This is CRUD**

**Student/Ākonga instructions**

**Introduction/Kupu Arataki**

Databases are everywhere. For example, the massive stock databases like those owned and managed by Amazon and the massive amounts of user data gathered by Google every second. Applications have to be able to Create, Read, Update and Delete this data- that is commonly referred to as CRUD. This is usually done by web applications or other types of database applications managed and written by real (highly paid!) people and some elements of this have to be open to users. This brings with it obvious implications around security, privacy, functionality and usability to name just a few.

**Task/Hei Mahi**

You will:

1. Design a database for a chosen purpose and end user
2. Create the database in SQLite Studio
3. Write a Python program to Create, Read, Update and Delete data in the database
4. Explain and hopefully address any relevant implications

| **The code and supporting evidence is due in Term 2** |
| --- |

### Part 1- Design the Database

You must decide on a database purpose and end user. This should be kept fairly simple with between 1 and 3 tables. Examples might include:

* A simple food shop ordering system for customers to make orders (like Dominoes©!)
* A Video Game rating and review database to help game buyers make informed decisions
* A PC Parts database to help buyers create and cost a new PC Build
* A Character Database (eg Marvel Heroes or LoL) to help players find more information and statistics on their favourite characters
* A Quiz program to help students study for a subject at school and make their own questions to help their learning
* Any other similar database purpose and end user

Once you have decided you need to create the **Entity Relationship Diagram**. This will include the tables, fields and data types as well as any Primary Keys or Foreign Keys.

**Add the Diagram to your assignment.**

**CheckPoint #1- Get this design checked off by your teacher!**

### Part 2- Create the Database

You must use the design to create the database. You will create the database file, the tables, the fields and any primary keys or foreign keys you need. This database should be tested to ensure it functions as expected with several sql queries that select, sort, filter, insert, update and delete entries. This will be done using sqlite studio.

**Complete the “Database Testing Table” section in your assignment.**

### Part 3- Create a simple CRUD Program in Python

You now need to plan and **incrementally** develop a program for your end-user (and potentially for the database system administrator) to interface with this database.

This will most likely include the ability for the user to:

* Create entries in the database
* Read, filter, sort and display relevant data from the database
* Update selected data if required
* Delete selected data from the database if required

Start small with simple functionality. Test regularly and improve the program by adding in more features. Keep versions of your code. Take Screenshots and/or short videos of functionality. This will help you complete the final section of the report- the “Showcase”.

You must also submit the final tested and working code as directed by your teacher.

### Part 4- Relevant Implications

In your report describe any relevant implications for your database and program. Take into account the purpose and end user(s) and the nature of data storage and distribution. This should be done at the start to help inform development and at the end to highlight how the implications may have been addressed.

Example of relevant implications include:

● social

● cultural

● legal

● ethical

● intellectual property

● privacy

● accessibility

● usability

● functionality

● aesthetics

● sustainability and future-proofing

● end-user considerations

● health and safety.

**Teacher Checklists:**

**AS91879- Develop a digital outcome to manage data**

**Credits: 4**

**NZQA:** [**https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91879.pdf**](https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91879.pdf)

| **Achieved- Develop a digital outcome to manage data** | **Looks Like….** |  |
| --- | --- | --- |
| using appropriate tools and techniques to structure, organise, query and present data for a purpose and end user | can make an ERD using correct data types, cardinality, naming conventions, Primary keys and Foreign Keys.  Can use SQLite Studio to create a database from an ERD, insert data and write/test queries |  |
| applying appropriate data integrity and testing procedures | Write and test at least 3 queries for specific purposes and ensure the data output from these queries is correct and appropriate (eg. no extra stuff) |  |
| describing relevant implications. | At least 3 relevant implications identified and described (eg. Functionality, Privacy, Usability etc) |  |
| **Merit- Develop an informed digital outcome to manage data** |  |  |
| using information from testing procedures to improve the quality and functionality of the outcome | Refine the data structure, add or remove columns or tables to improve functionality and usefulness of the database based on iterative improvements and testing over the course of the development. |  |
| structuring, organising and querying the data logically | Refined structure following a consistent conventions, using appropriate data types and relational tables to reduce errors and repetition of data (normalised data) |  |
| addressing relevant implications. | Ensure privacy, functionality and usability implications are met. eg. don’t store plain text passwords, applications functions perfectly, allow users to access all data and present it in a clear way to meet the users needs etc. |  |
| **Excellence- Develop a refined digital outcome to manage data** |  |  |
| iterative improvement throughout the development and testing process | Clear evidence or improved functionality throughout development. More tables/columns added to improve functionality. |  |
| presenting the data effectively for the purpose and to meet end-user requirements. | Concise and focussed presentation of the data providing all the information that a user of this database might need. |  |

**Develop a computer program**

**Credits:** 4 (Internal)

**NZQA:** <http://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2018/as91883.pdf>

| **Achieved**  **Develop a computer program** | **Might Look Like…..** |  |
| --- | --- | --- |
| Wrote a program that performs a specific task using a suitable programming language | Wrote a functional program that takes user input and displays information returned from queries of the database in a human readable fashion. |  |
| Set out the program code clearly | Has variables and globals at the top, the functions, then the main program loop. |  |
| Documented the program with comments | Some comments describing code function. |  |
| Tested and debugged to ensure that it works on a sample of expected cases | It works when the user types in what is expected of them. Does not crash, break or have bugs. |  |
| **Merit**  **Develop an informed computer program** |  |  |
| Documented the program with variable names and comments that describe code function and behaviour | Well used comments. |  |
| Following conventions of the chosen programming language | PEP8 followed. Specifically naming conventions, correct use of snake\_case etc. |  |
| Tested and debugged the program in an organised way to ensure it works on expected and relevant boundary cases | As A but also checks for boundary and unexpected input. Eg. if it asks for a number, it should not crash when given letters. Typically uses try->catch or structures to ensure only correct input is processed.  May inform users of errors or incorrect inputs. |  |
| **Excellence**  **Develop a refined computer program** |  |  |
| Ensured the program is a well structured logical solution to the task | Neat concise readable main loop due to useful functions that take parameters and/or return values.  Provides all wanted functionality from the database.  Informs users of errors or incorrect inputs. |  |
| Making the program flexible and robust | Can be easily changed or extended due to good structure and use of functions.  May even write a ‘utility file’ and import it into the main program. |  |
| Comprehensively tested and debugged the program | Will not break or crash no matter what input it is given. |  |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.