

**Internal Assessment Resource**

**Digital Technologies | Hangarau Matihiko**

**Level 3**

This resource supports assessment against Achievement Standard 91902 and 91906

**Standard title:** Use complex techniques to develop a database

Use complex programming techniques to develop a computer program

**Credits:** 10 Credits

**Resource title: My Data**

| 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**

This resource supports assessment against Achievement Standard 91902 and 91906

**Standard title:** Use complex techniques to develop a database

Use complex programming techniques to develop a computer program

**Credits:** 10 Credits

**Resource title: My Data**

**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. Research and decide on a purpose and end user for a data project
2. Design the database structure with an ERD.
3. Create the database in SQLite Studio. This could include writing code to parse data from another source (eg. a CSV file)
4. Write a Python program to Create, Read, Update and Delete data in the database
5. Address any relevant implications

| **The code and supporting evidence is due in Term 1** |
| --- |

### Part 1- Research and Decide

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:

* Data about crime, health or education for large datasets such as:
  + <https://www.pewresearch.org/internet/datasets/>
  + <https://data.gov/>
  + <https://catalogue.data.govt.nz/dataset>
  + <https://www.stats.govt.nz/>
* 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 researched and 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. This MUST contain at least one many to many relationship.

| **Add the Entity Relationship Diagram here** |
| --- |
|  |

**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. You may need to write code to parse data from large external datasets into your database.

Once this is done, the database should be tested to ensure it functions as expected with several sql queries that select, sort, filter, insert, update and delete entries. Decide on a number of queries that you will probably need for this data and write and test the SQL queries on your database in SQLite Studio. Add them to the table below.

#### 

| SQL Query Testing Table | | |
| --- | --- | --- |
| **Purpose** | **Query** | **Pass/Fail** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Part 3- Create a 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

If you have not already done so, create a local folder for your code and **publish it to github.**

Start small with simple functionality. Test regularly and improve the program by adding in more features. Commit regularly with good commit messages to your github repo.

The github repo will be your final submission and your commit history will help show how much it improved.

| **Submit a link to your github repo below:** |
| --- |
|  |

### Evidence of Program Testing

Your final program must now be tested thoroughly before submission. Think of all the possible inputs that the program can be given and test them all. Write the tests down in the table below in a format that best suits your program. You may wish to test small parts of the program (specific functions) or the whole program.

Don’t forget to include expected, boundary and invalid data inputs. Try to break your code!!

| **Program Testing Table** | | | |
| --- | --- | --- | --- |
| **What are you testing** | **Input Value(s)** | **Expected Result(s)** | **Pass/Fail** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Part 4- Relevant Implications

In your report explain how you addressed 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.

| **Implications** | **How I addressed it (if relevant)** |
| --- | --- |
| social |  |
| cultural |  |
| legal |  |
| ethical |  |
| intellectual property |  |
| privacy |  |
| accessibility |  |
| usability |  |
| functionality |  |
| aesthetics |  |
| sustainability and future-proofing |  |
| end-user considerations |  |
| health and safety. |  |

**Teacher Checklists:**

## AS91906 - Use complex programming techniques to develop a computer program

**Credits:** 6 (Internal)

**NZQA:**  <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91906.pdf>

| **Achieved** | **Evidence** |  |
| --- | --- | --- |
| writing code for a program that performs a specified task |  |  |
| using complex techniques in a suitable programming language |  |  |
| setting out the program code clearly and documenting the program with comments |  |  |
| testing and debugging the program to ensure that it works on a sample of expected cases. |  |  |
| **Merit** |  |  |
| documenting the program with appropriate names and organised comments that describe code function and behaviour |  |  |
| following conventions for the chosen programming language |  |  |
| testing and debugging the program effectively to ensure that it works on a sample of both expected cases and relevant boundary cases. |  |  |
| **Excellence** |  |  |
| ensuring that the program is a well-structured, logical response to the specified task |  |  |
| making the program flexible and robust |  |  |
| comprehensively testing and debugging the program. |  |  |

## 

## AS91902- Use complex techniques to develop a database

**Credits:** 4 (internal)

**NZQA:**  <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91902.pdf>

| **Achieved- develop a database** | **Evidence** | ✔ |
| --- | --- | --- |
| designing the structure of the data |  |  |
| using appropriate tools and advanced techniques to organise, query and present data for a purpose and end users |  |  |
| applying appropriate data integrity and testing procedures |  |  |
| addressing relevant implications. |  |  |
| **Merit- Develop and informed database** |  |  |
| using information from testing procedures to improve the quality of the outcome |  |  |
| structuring, organising and querying the data logically |  |  |
| **Excellence- develop a refined database** |  |  |
| iterative improvement throughout the design, development and testing process |  |  |
| using efficient tools and techniques in the outcome’s production |  |  |
| presenting the data effectively for the purpose and end users. |  |  |

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