Develop a Database

Internal Assessment Resource

Digital Technologies: Level 1

This resource supports assessment against Achievement Standard 92005

Standard title:	Develop a digital technologies outcome
Credits:	5 Credits
Resource title:	**My Database Application**

Authenticity of evidence	Assessor involvement during the assessment event is limited to providing general feedback which suggests sections of student work that would benefit from further development or skills a student may need to revisit across the work. Student work which has received sustained or detailed feedback is not suitable for submission towards this Standard.

Student/Ākonga instructions

Introduction

You are going to make a simple database application using Python and SQL in Visual Studio Code over four weeks.

This application can be your own design or a design can be given to you by your teacher. Please note that **databases should NOT be more than 3 tables.** A single table is acceptable as long as the design is correct and you may have to compromise to ensure the database does not become too complex.

Appendix 1 contains a few examples.

Please ensure you have a sound grasp of SQL and Python before attempting this assessment. You must have completed all the relevant coursework before beginning this assessment as your teacher is NOT allowed to give you significant help beyond basic feedback during the assessment period.

You will be expected to be able to follow basic software engineering and programming conventions during this project.

Conventions include but are not limited to:

- Using an organised file structure for your project files
- Using well named files and folders in your project
- Has appropriate variable and function names that describe their purpose
- Using code comments where appropriate
- Following common language conventions for Python and SQL
- Following Database Design standards (eg normalized data to 3NF- well structured database)
- Using version control including descriptive commit messages

You are encouraged to incrementally develop your application through weekly iterations of “plan-develop-test”, keeping a record of your progress as you go inside of this document. Git Version control should also be used as evidence of this.

If you are unable to use git, multiple files in your source code folder must be made with version number included. Failure to do so will limit your ability to gain a mark better than achieved.

You will submit the working application (stored on github or a zip of your entire source code folder) as well as this document for assessment against the criteria listed above.

Now complete the following questions.

Before Development

What is the purpose of this Application

A quiz game that lets people add their own questions, attempt to answer questions, and vote on which questions are better. It will also track how many questions users have answered and added.

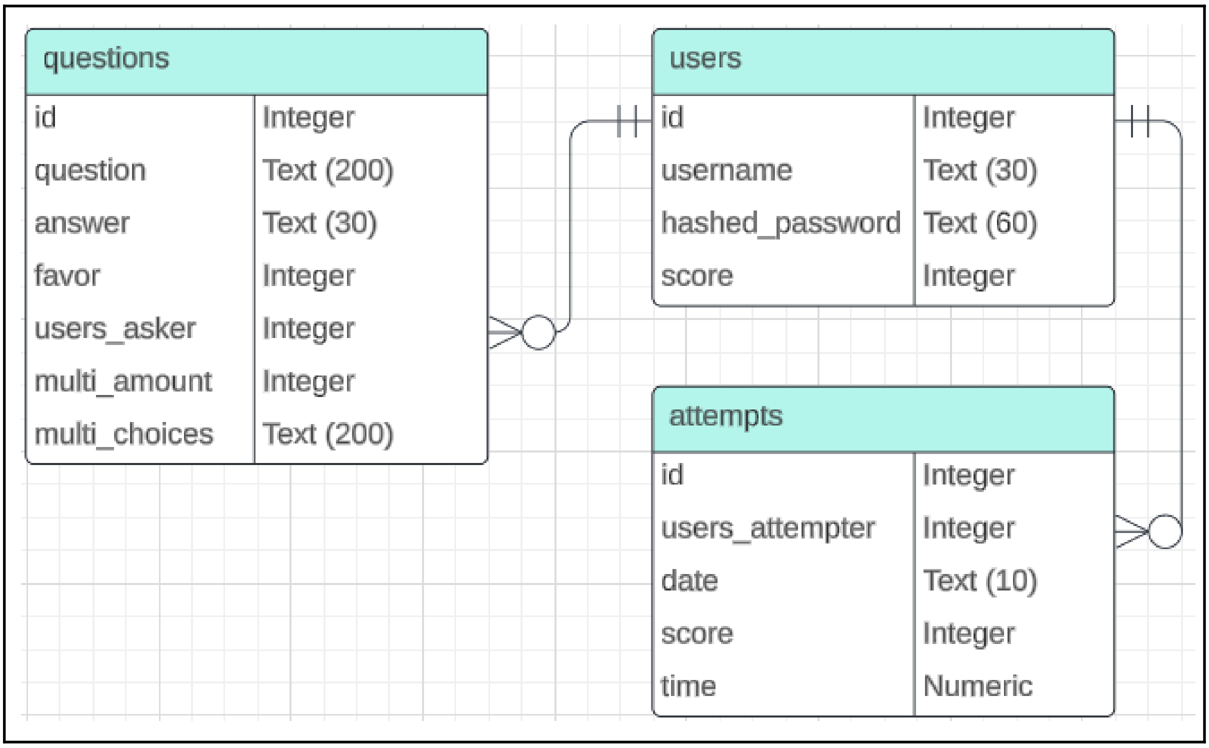
Who might use this Application?

People who like learning trivia or are bored.

What are the requirements and specifications for this Application?

Requirements	**Specifications**
Users can register accounts and log in. Users can add questions and attempt to answer questions. The application will detect whether the user's answer was correct or not and gain the user score. Users can rank questions, higher favoured questions are more likely to be asked.	It uses a questions table to store a question's wording, answer, favour, and author. It uses a users table to store usernames, hashed passwords, score, and questions.

Database Design- Your Entity Relationship Diagram.



questions	
id	Integer
question	Text (200)
answer	Text (30)
favor	Integer
users_asker	Integer
multi_amount	Integer
multi_choices	Text (200)

users	
id	Integer
username	Text (30)
hashed_password	Text (60)
score	Integer

attempts	
id	Integer
users_attempter	Integer
date	Text (10)
score	Integer
time	Numeric

Database Testing Table: SQL Statements

Write down all the "purposes" that you think of for your database and then work out the SQL query that you might use. Test it and make sure it works here before moving on.

Purpose	**SQL Statement**	**Result Success?**
Remove unpopular questions	DELETE FROM questions WHERE favor <= -8;	ya
Get all questions sorted by favor	SELECT * FROM questions ORDER BY favor DESC;	ya
Update question favor after ranking	UPDATE questions SET favor = {favor+FAVOR_CURVE[i]} WHERE id = {id};	ya
Get password from username	SELECT hashed_password FROM users WHERE username = '{username}';	ya
Add question	INSERT INTO questions (question, answer, favor, asker) VALUES ('{question}', '{answer}', 0, '{asker}')	ya
Add user	INSERT INTO users (username, hashed_password, score) VALUES ('{username}', '{hashed_password}', 0)	ya
Add score	UPDATE users SET score = {score+corrects} WHERE id = {id};	ya
Get author name from id	SELECT username FROM users WHERE id = {id};	ya

Purpose	SQL Statement	Result Success?
Remove unpopular questions	DELETE FROM questions WHERE favor <= -8;	ya
Get all questions sorted by favor	SELECT * FROM questions ORDER BY favor DESC;	ya
Update question favor after ranking	UPDATE questions SET favor = {favor+FAVOR_CURVE[i]} WHERE id = {id};	ya
Get password from username	SELECT hashed_password FROM users WHERE username = '{username}';	ya
Get user information	SELECT id, username, score FROM users WHERE username = '{username}';	ya
Add attempt	INSERT INTO attempts (users_attempter, score, time) VALUES ({user_id}, {corrects}, {time_taken});	ya
Get attempts	SELECT score, time FROM attempts WHERE users_attempter = {user_id} ORDER BY {sort} {order};	ya

During Development

Now you are ready to start an iterative process of development. You have one week of development per "Sprint".

At the end of each week/Sprint, you will test the application with other people in your class, your teacher or stakeholders and summarise the feedback that they give you to help you to refine the application further.

Note: **Testing** is ongoing throughout development. Every time you write code and run it to see that it works you are testing. **Trialling** is deliberate, organised testing WITH SEVERAL OTHER PEOPLE. Watch them use your applications without prompting and see what they have difficulty with, what they do and what they break.

Sprint#1 Trialling		**Date: 2024 02 26**
Who	**What did they say?**	**How does this help?**
Mrs Maddaford	If I accidentally hit a character instead of a number e.g. on the how many days in a week, should this be incorrect? If I can't spell, these would be incorrect - could they be multi choice?	Give the user multiple tries, add multiple choice maybe, detect if the guess is close to the answer maybe (but not for numbers)
Mr Rodkiss	Track user attempts, track multiple score types (best, average, sum)	Do those things; also possibly ask more than 5 questions sometimes.
Rick Wang	I thought there were 9 days in week, I now know there are 5. Tanks!	Make more, better questions.

<table>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3"><b>Summary of Testing and Feedback and next steps</b></td></tr>
<tr><td colspan="3">Make user system
- Allow entering questions
Give multiple tries
Multiple choice?
Attempt history</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Sprint#2 Trialling</b></td><td><b>Date:2024 03 15</b></td></tr>
<tr><td><b>Who</b></td><td><b>What did they say?</b></td><td><b>How does this help?</b></td></tr>
<tr><td>Lukas De Wit</td><td>Leaderboard doesnt work when someone hasnt had any attempts yet
When adding a question if the set answer has capitals it cannot be answered</td><td>Fixed the bugs</td></tr>
<tr><td>Rick</td><td>Not much but I watched him test it out</td><td>Various quality of life improvements</td></tr>
<tr><td>Mr Rodkiss</td><td>It was good. The questions were hard. The table was cool. Hashing the passwords was good.</td><td>Confidence boost?</td></tr>
<tr><td colspan="3"><b>Summary of Testing and Feedback and next steps</b></td></tr>
<tr><td colspan="3">Finish things up, test for bugs</td></tr>
</table>

Final Trialling	**Date:2024 04 12**
Give three examples of how and why you improved your outcome	
Example 1. I added multi-choice questions, because the questions needed variety, and needed to be easier.	
Example 2. I made the quiz timed, to encourage fast question answering.	
Example 3. I added user statistics tracking and a leaderboard, so users can compare their results with each other.	

Git Log

Link to Github Repo:
https://github.com/Beeka274/sql1
Screenshot of your commit history.

Final Product

Github repo link or zip file link (through Google Drive or OneNote)

https://github.com/Beeka274/sql1

Appendix 1

GPU Database
A database of modern graphics cards, their speeds and their prices for PC gamers who want to be able to compare cards by manufacturer, price and speed.

Sports Team Tracker
A sports team database for a coach to keep track of players attendance, games and points scored over the season.

Library Catalog Database
A comprehensive catalog system allowing librarians to manage book details, borrower information, and lending status.

Student Grade Tracker
An efficient tool enabling teachers to record and analyze student grades for assignments, quizzes, and tests.

Inventory Management System
A streamlined solution for businesses to monitor product inventory, suppliers, and reordering needs**.**

Study Guide
A program to help students study by asking quick quiz questions and checking their answers. They can customise it by being able to add or delete questions from the database.

Recipe Organizer
A user-friendly platform for culinary enthusiasts to collect and categorize recipes, along with ingredients and cooking steps.

Fitness Progress Database
A fitness tracking tool allowing individuals to log workouts, monitor progress, and set fitness goals.

Task Manager Database
An intuitive task management system for organizing tasks, setting priorities, and monitoring progress.

For Teacher Use

Develop a digital technologies outcome

Domain: Digital Technology 1.2
Standard: AS92005 v1
Credits: 5 (Internal)
Link: https://ncea.education.govt.nz/technology/digital-technologies/1/2?view=standard

Achieved **Develop a digital technologies outcome**	**Comments**	
Identifying the purpose, potential users, requirements, and specifications of the outcome	Complete section 1 correctly	
Using appropriate tools and techniques of a digital technologies domain to produce an outcome that addresses the requirements and specifications	Use Python and SQLite to create a program to interface with a database that functions as expected and hands it in on time. Application has at least read functionality and can output data from the database from users input. Handed in all project code as a github link or as zip file. May not have used github Final project may not look exactly like design and might not have all the intended functionality.	
Testing the outcome to ensure basic functionality.	Teacher observation of testing and the application looks and functions mostly as expected. May not have done sql query table.	
Merit **Refine a digital technologies outcome**		
Following relevant conventions of a digital technologies domain	Meet at least 4 of the following (teacher discretion) Conventions include but are not limited to: ● Using an organised file structure for your project files ● Using well named files and folders in your project ● Has appropriate function names that describe the purpose ● Using code comments where appropriate ● Following common language conventions for Python and SQL ● Following Database Design standards (eg normalized data to 3NF- well structured database) ● Using version control including descriptive commit messages	
Using information from testing to make improvements to the outcome's fitness for purpose.	Did testing during development. This can be observed or exemplified through a git commit log or multiple versions of code with incremental improvements.	
Excellence **Enhance a digital technologies outcome**		
using information from trialling the outcome with others to improve its fitness for purpose	Lots of improvement through at least three versions. Including relevant feedback from others and implementing suggested changes where appropriate. Teachers should set “Checkpoints” with the whole class to help facilitate this.	
applying tools and techniques effectively in the production of a fit-for-purpose outcome.	The final outcome meets the design and specifications and shows an excellent grasp of Python and SQL to create an above average program that follows all conventions. Adds features not included in the tutorial like Login/admin/passwords etc.	

Teacher Comments: