Government of **Western Australia**
School Curriculum and Standards Authority

Technologies: Digital Technologies

Teaching, learning and assessment exemplar

Year 10

Using relational databases

Acknowledgement of Country

Kaya. The School Curriculum and Standards Authority (the Authority) acknowledges that our offices are on Whadjuk Noongar boodjar and that we deliver our services on the country of many traditional custodians and language groups throughout Western Australia. The Authority acknowledges the traditional custodians throughout Western Australia and their continuing connection to land, waters and community. We offer our respect to Elders past and present.

Background

This teaching, learning and assessment exemplar (the exemplar) has been developed by the School Curriculum and Standards Authority (the Authority) as part of the *School Education Act Employees (Teachers and Administrators) General Agreement 2017* (Clause 61.1–61.3).

Disclaimer

Any resources, such as texts and websites, that may be referred to in this document are provided as examples of resources that teachers can use to support their learning programs. Their inclusion does not imply that they are mandated or that they are the only resources relevant to the course. Teachers must exercise their professional judgement as to the appropriateness of any resources they may wish to use.

This resource utilises electronic web-based resources, such as videos and image galleries. Teachers should be present while an electronic resource is in use and close links immediately after a resource, such as a video has played to prevent default 'auto play' of additional videos. Where resources are referred for home study, they should be uploaded through Connect, or an equivalent system, that filters advertising content.

Contents

The Western Australian Curriculum

The *Western Australian Curriculum and Assessment Outline* (the *Outline* – https://k10outline.scsa.wa.edu.au/) sets out the mandated curriculum, guiding principles for teaching, learning and assessment, and support for teachers in their assessment and reporting of student achievement. The *Outline* recognises that all students in Australian schools, or international schools implementing the Western Australian Curriculum, are entitled to be given access to the eight learning areas described in the *Alice Springs* (*Mparntwe*) *Education Declaration*, December 2019.

The Technologies curriculum

The mandated curriculum is presented in the year level syllabus documents.

The Technologies curriculum delivers a sequential and age-appropriate progression of learning with the following key elements:

- a year level description that provides an overview of the context for teaching and learning in the year
- a series of content descriptions, populated through strands and sub-strands, that sets out the knowledge, understanding and skills that teachers are expected to teach and students are expected to learn
- an achievement standard that describes an expected level that the majority of students are achieving by the end of a given year of schooling. An achievement standard describes the quality of learning (e.g. the depth of conceptual understanding and the sophistication of skills) that would indicate the student is well placed to commence the learning required in the next year.

This exemplar

This Technologies exemplar articulates the content in the *Outline* and approaches to teaching, learning and assessment reflective of the Principles of Teaching, Learning and Assessment. This exemplar demonstrates a sequence of teaching and learning, including suggested assessment points, for 16 lessons.

Catering for diversity

This exemplar provides a suggested approach for the delivery of the curriculum and reflects the rationale, aims and content structure of the learning area. When planning the learning experiences, consideration has been given to ensuring that they are inclusive and can be used in, or adapted for, individual circumstances. It is the classroom teacher who is best placed to consider and respond to (accommodate) the diversity of their students. Reflecting on the learning experiences offered in this exemplar will enable teachers to make appropriate adjustments (where applicable) to better cater for students' gender, personal interests, achievement levels, socio-economic, cultural and language backgrounds, experiences and local area contexts.

Using this exemplar

This teaching, learning and assessment exemplar provides suggestions to support the delivery of the mandated curriculum content. The exemplar provides:

- a teaching and learning sequence
- the mandated curriculum content to be taught at each point of the teaching and learning sequence, suggested resources, sample assessment tasks and marking keys
- the number of lessons to deliver the teaching and learning experiences
- learning intentions and support notes that may provide focus questions and additional information and/or examples to assist with the interpretation of curriculum content
- support notes to assist teachers to unpack the content and support teaching and learning experiences
- teaching and learning experiences that outline the structure of the lesson. These explicitly state each activity that the lesson will progress through and the key focus area for that activity.

Links to electronic resources

This sequence of lessons may utilise electronic web-based resources, such as videos and image galleries. Teachers should be present while an electronic resource is in use and close links immediately after a resource, such as a video, has played to prevent default ‘auto play’ of additional videos. Where resources are referred for home study, they should be uploaded through Connect, or an equivalent system, that filters advertising content.

Best practice

Teaching and learning

The teaching and learning opportunities offered in this exemplar are not exhaustive. Thus, teachers are encouraged to make professional decisions about which learning experiences, and the sequence in which they are delivered, are best suited to their classroom context, taking into account the availability of resources and student ability.

This sample may prove a useful starting point for amplifying creativity in the classroom, while presenting the embedded expectations of the Western Australian Curriculum: Technologies.

Teachers may find opportunities to incorporate the General Capabilities and the Cross-curriculum Priorities into the teaching and learning program.

Ways of teaching – teachers can locate additional information on the Ways of teaching from the School Curriculum and Standards Authority (the Authority) website https://k10outline.scsa.wa.edu.au/home/wa-curriculum/learning-areas/technologies/digital-technologies/p-10-digital-technologies-teaching/digital-technologies-ways-of-teaching.

Assessing

Assessment, both formative and summative, is an integral part of teaching and learning. Assessment should arise naturally out of the learning experiences provided to students. In addition, assessment should provide regular opportunities for teachers to reflect on student achievement and progress. As part of the support it provides for teachers, this exemplar includes suggested assessment points. It is the teacher's role to consider the contexts of their classroom and students, the range of assessments required, and the sampling of content descriptions selected to allow their students the opportunity to demonstrate achievement in relation to the year level achievement standard. Teachers are best placed to make decisions about whether the suggested assessment/s are used as formative or summative assessment and/or for moderation purposes.

Ways of assessing – a range of assessment strategies that can enable teachers to understand where students are in their learning is available on the Authority website https://k10outline.scsa.wa.edu.au/home/wa-curriculum/learning-areas/technologies/digital-technologies/p-10-digital-technologies-assessing/digital-technologies-ways-of-assessing.

Reflecting

Reflective practice involves a cyclic process during which teachers continually review the effects of their teaching and make appropriate adjustments to their planning. The cycle involves planning, teaching, observing, reflecting and replanning.

This exemplar supports reflective practice and provides flexibility for teachers in their planning. The exemplar shows how content can be combined and revisited throughout the year. Teachers will choose to expand or contract the amount of time spent on developing the required understandings and skills according to their reflective processes and professional judgements about their students' evolving learning needs.

Using relational databases

In this unit of work, students will learn about how relational databases can be used to store large amounts of data in an efficient way. They will learn how to represent the structure of a relational database using an entity relationship diagram and how to extract useful information from each database using structured query language (SQL).

In designing and creating their own escape room challenges that they will share with the rest of the class, students will demonstrate their understanding of databases and the use of SQL In designing and creating their own escape room challenges that they will share with the rest of the class.

Year level description

In the middle adolescence phase of schooling, teaching and learning programs encourage students to develop an open and questioning view of themselves as active participants in their society and the world.

Digital Technologies further develops student understanding and skills in computational thinking, such as precisely and accurately describing problems, and the use of modular approaches to solutions. Students increase their understanding of the complexity of the natural environment, society and technology. They investigate the potential and problems of increased knowledge and choice of technologies; and an understanding of the relationship between knowledge, technologies and consumer and/or producer values.

In Year 10, students consider how human interaction with networked systems introduces complexities surrounding access to, and the security and privacy of, data of various types. They interrogate security practices and techniques used to compress data. Students explore the role of hardware and software in managing, controlling and securing access to data, in networked digital systems focusing on user or software supply chain vulnerabilities.

Students apply design thinking by using divergent techniques to generate design ideas for user experiences and solutions. They develop and represent documents online as content (text), structure (mark-up) and presentation (styling). Students analyse problems and design, and implement and evaluate a range of solutions, such as database-driven websites, artificial intelligence engines and simulations. They design and implement algorithms involving functions and logical operators and, where appropriate, represent them as flow charts. Students further develop project management skills through developing detailed plans that are considerate of time, production processes, social, ethical, economic and sustainability factors, and legal responsibilities.

Achievement standard

By the end of the year:

Students describe how hardware and software manage, control and secure access to data in networked digital systems with a focus on software supply chain vulnerabilities. They represent documents online as content (text), structure (mark-up) and presentation (styling) and explain the purpose of these distinctions. Students analyse and visualise data interactively using a range of software, including spreadsheets and relational databases, to draw conclusions and make predictions based on identified trends and outliers. They model and query entities and relationships using structured data. Students apply the Australian Privacy Principles to critique systems and manage the digital footprint of individuals.

Students define and decompose real-world problems and use data gathering techniques to create user experiences and user interfaces. They design and prototype the user experience of a digital system and algorithms involving functions, modules and logical operators, and represent them as flow charts and/or pseudocode. Students validate algorithms and programs by comparing their output against a range of test cases. They implement, modify and debug modular programs, applying algorithms and data structures, in a programming language.

In Digital Technologies, students identify the needs of the client or stakeholder to determine the basis for a solution. They develop and critique design briefs for a designed solution. Students investigate components and resources to develop increasingly sophisticated solutions, identifying and considering associated constraints. They apply design thinking, creativity, enterprise skills and innovation to develop, modify and communicate detailed design ideas. Students design possible solutions and analyse designs against criteria, including functionality, accessibility, usability and aesthetics, using appropriate technical terms and technologies. They select, justify, implement and test appropriate technologies and processes to produce designed solutions. They manage projects, using digital technologies with an agile and collaborative approach, while considering time, production processes, social, ethical, economic and sustainable factors, and legal responsibilities.

Lessons 1–16

Acquiring, managing and analysing data

Lesson 1: Database basics

The Western Australian Curriculum content addressed in this lesson is below.

Acquiring, managing and analysing data

- Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers

---

Lesson outline

Learning intentions
Students will: • understand what a database is and its purpose • differentiate between spreadsheets and databases • identify basic components of databases: tables, records, fields.

Preparation for the lesson

- Source real-world examples of databases (social media, school systems, streaming services) to explain their purpose and prevalence.
- A student worksheet example can be found in Appendix E.

Support notes

- A database is a collection of data that has been organised to allow for efficient storage, retrieval and management of information.
- Spreadsheets are used primarily for tabular data and performing data analysis, whereas databases are used primarily for storing large volumes of data to allow efficient access to that data.

Teaching and learning experience

Introduction

- Activate prior knowledge with a class discussion on where students have seen databases in daily life including social media, school systems and streaming services.
- Conduct a brainstorm using the question ‘Why do we need databases?’

Main activities

- Compare spreadsheets and databases visually, discussing the different purposes of each.
- Emphasise the organisation and hierarchical structure of databases: tables made up of records that are made up of rows that are made up of fields.
- Show different view options of a database to help differentiate them from spreadsheets (that is, show the table view, then form showing individual records).
- Students draw a diagram showing the structure of a simple database, including tables, records and fields. They identify the components of a sample database, including tables, individual records and fields.
- Students record their lesson observations in the student worksheet found in Appendix E.

Lesson conclusion

- Students write a definition of a database in their own words.
- Students explain the difference between a spreadsheet and a database in their own words.

Lesson 2: Datatypes

The Western Australian Curriculum content addressed in this lesson is below.

Acquiring, managing and analysing data

- Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers
- Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • explain what data types are and why they are important • identify common data types: integer, float, Boolean, text, data • design a simple table with appropriate data types • define primary key and its role in database management.

Preparation for the lesson

- Worksheet examples for students can be found in Appendix E.

Support notes

- The data type specifies the kind of data that can be stored in a field. The data type determines how the computer interprets that data and what operations can be performed on it.
- A primary key is a field that uniquely identifies a single record in a table. Note: If multiple fields are used to create a unique identifier then this is called a composite key.

Teaching and learning experience

Introduction

- Class discussion: What are data types and why do we need to use them?
  Note: Students should be familiar with this from previous programming experience.
- Students identify data types for fields (e.g. age: integer, first_name: text).

Main activities

- Discuss the importance of choosing the correct data type, taking into account the purpose of data being stored and storage implications.
- Discuss the importance of having a unique way to identify each record in a table, showing examples of duplicate data without primary keys (e.g. two students with the same name).
- Explain the rules of a primary key – it must be unique and not NULL. In a database, NULL represents the absence of a value (unknown or missing data), not zero or an empty string, and it requires special handling in comparisons and calculations.
- Students design a simple database table on a topic of their choice with field names and data types.
- Students peer review each other's table designs and modify tables based on peer feedback.
- Class discussion: How do we uniquely identify each record in a table? Explain the role of a primary key giving examples (e.g. student ID number, driver's licence number, tax file number).
- Add primary keys to tables designed previously.

Lesson conclusion

- Students summarise why choosing correct data types is important and describe the purpose of different data types completing the worksheet example in Appendix E.

Lesson 3: SQL

The Western Australian Curriculum content addressed in this lesson is below.

Acquiring, managing and analysing data

- Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers
- Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • define structured query language (SQL) and its purpose • write basic SELECT queries • retrieve data from a single table.

Preparation for the lesson

- Source appropriate database tools for the lesson. These could include:
  - DB Browser for SQLite (https://sqlitebrowser.org) – free, open-source desktop application
  - SQLite Online (https://sqliteonline.com) – free online database tool.
- A student worksheet example can be found in Appendix E.

Support notes

- SQL (structured query language) is a standardised language that is used to interact with a relational database. It allows the user to manage a database and extract useful information from the database.

Teaching and learning experience

Introduction

- Class discussion: How do we efficiently extract information from a database? Definition of SQL.

Main activities

- Demonstrate how to open database files and run queries.
- Emphasise the basic structure of a SELECT query is always the same (SELECT fields FROM table).
- Encourage students to use capital letters for key works (such as SELECT) and multiple lines for queries to improve readability.
- Introduce students to a database tool (appropriate tools include DB Browser for SQLite).
- Demonstrate a basic SQL query.
- Students write and execute a basic query. They write queries to retrieve specific fields from a table.
- Students should predict query results before running each query.

Lesson conclusion

- Students explain what a SELECT query does and the basic structure of a SELECT query.
- Students make notes on this lesson based on the questions from the worksheet in Appendix E.

Lessons 4–5: SQL 2

The Western Australian Curriculum content addressed in these lessons is below.

Acquiring, managing and analysing data

- Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers
- Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • filter SQL queries using WHERE with conditional operators • apply ORDER BY to sort results of an SQL query • combine conditions using AND/OR • use logical operators with WHERE clause to filter data • use wildcard characters with LIKE operator.

Preparation for the lesson

- A student worksheet example can be found in Appendix E.

Support notes

- Conditional operators in SQL are:

Operator	Meaning
=	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

- Logical operators in SQL include:

Operator	Meaning
AND	Combine multiple logical operations. True if all conditions are true.
OR	Combine multiple logical operations. True if one condition is true.
NOT	Negates a logical operation. True if the logical condition is false.
IN	Checks to see if a value is in a list of values. For example, colour IN [“blue”, “yellow”, “red”]
BETWEEN	Checks to see if a value is within the range between two values. For example, age

	BETWEEN 10 AND 18 NOTE: this is an *inclusive* operation.
LIKE	Checks to see if a value matches a specific pattern. Uses wildcard characters % (multiple letters) and _ (single character). For example, last_name LIKE "%ert%"

Teaching and learning experience

Introduction

- Class discussion: How can we filter the results of our queries to make them more meaningful?

Main activities

- Discuss importance of being able to find specific information in a table and filter information using a WHERE clause.
- Expand upon basic structure of SELECT statement from previous lesson (SELECT fields FROM table WHERE condition). Emphasise that students will always use the same structure for all SELECT queries.
- Demonstrate use of WHERE to filter data from a table with different conditional operators: =, !=, >, <, >=, <= (students should be familiar with these from previous programming experience).
- Demonstrate use of ORDER BY to sort results.
- Demonstrate use of logical operators IN, BETWEEN and NOT.
- Demonstrate use of logical operator LIKE in conjunction with wildcard characters % and _
- Students use Appendix E alongside learning activities to record their results and keep focused.
- Students modify basic SQL queries using a WHERE clause with different conditional operators.
- Students write their own SQL queries to find specific information from a table.
- Students write SQL queries to return results sorted based on specific fields.
- Students write SQL queries using various conditional operators to extract specific information from a table.

Lesson conclusion

- Students explain the purpose of different logical operators.
- Students correctly predict results of a query that is given to them.

Lesson 6: SQL 3

The Western Australian Curriculum content addressed in this lesson is below.

Acquiring, managing and analysing data

- Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers
- Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • use aggregate functions in SQL queries.

Preparation for the lesson

- Source multiple examples of how aggregate functions could be used.
- A student worksheet example can be found in Appendix E.

Support notes

- Aggregate functions perform calculations on multiple rows within a database table and return a single summarised value for that calculation. Aggregate functions in SQL are:

Function	Purpose
COUNT	The number of items in a set
SUM	The total sum of all the items in a set
AVG	The average value of all the items in a set
MAX	The maximum value of items in a set
MIN	The minimum value of items in a set

Teaching and learning experience

Introduction

- Describe purpose of each aggregate function and demonstrate using them in a SELECT query.

Main activities

- Class discussion: Purpose of aggregate functions to summarise data.
- Demonstrate use of aggregate functions COUNT, SUM, AVG, MAX, MIN.
- Students write SQL queries using various aggregate functions.
- Students combine previous SQL skills with use of aggregate functions to write more complex queries.

Lesson conclusion

- Students complete the worksheet in Appendix E to check for understanding.

Lesson 7: ERDs

The Western Australian Curriculum content addressed in this lesson is below.

Acquiring, managing and analysing data

- Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • define foreign keys and their purpose • explain one-to-many and many-to-many relationships • create foreign key between two tables.

Preparation for the lesson

- Prepare examples of data that have been unnecessarily repeated in a flat-file database.
- A student worksheet example can be found in Appendix E.

Support notes

- A foreign key is a field in one table that refers to the primary key of another, related table. It is used to ensure that there is a valid relationship between records of two different tables.

Teaching and learning experience

Introduction

- Discuss problems with repeating data, such as need to update data in several locations and the increased risk of errors occurring.
- Emphasise that a foreign key must ALWAYS go on the many sides of a one-to-many relationship.

Main activities

- Class discussion: Problems with storing data in a single table, in particular unnecessary repetition of data. How can we overcome this? Difference between flat-file and relational database.
- Demonstrate creating simple relational database with two tables. Create a foreign key to link two tables.
- Students complete the worksheet in Appendix E.

Lesson conclusion

- Students describe the purpose of a foreign key in a database and/or describe types of relationships: one-to-many and many-to-many as an exit ticket.

Lesson 8: ERDs 2

The Western Australian Curriculum content addressed in this lesson is below.

Acquiring, managing and analysing data

Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • explain what entity relationship diagrams (ERDs) represent • identify entities, attributes and relationships in an ERD • draw an ERD for a simple scenario.

Preparation for the lesson

- A student worksheet example can be found in Appendix E.

Support notes

- An entity relationship diagram (ERD) is a visual representation of a database, showing how entities relate to each other.
- Teachers should use 'crow's foot' notation for ERDs to be consistent with ERD standards. These ERD standards can be found in the Computer Science ATAR Year 11 and Year 12 Additional syllabus support booklet on the Authority website at https://senior-secondary.scsa.wa.edu.au/syllabus-and-support-materials/technologies/computer-science.

Teaching and learning experience

Introduction

- Reinforce the need for a visual way to represent the structure of a database.
- Emphasise the use of correct terminology and symbols.
- Reinforce the link between parts of a database and components of an ERD:
  - An entity is equivalent to a table
  - An attribute is equivalent to a field
  - Relationships show which tables are connected and where the foreign key should go.

Main activities

- Class discussion: How can we represent the structure of a database clearly so that we can find the information we need?
- Class discussion: The purpose and components of an ERD. Explain the relationship between components of an ERD and a relational database (entities/tables, attributes/fields).
- Students complete questions and activities in Appendix E.

Lesson conclusion

- Students peer review and give feedback on each other's ERDs.

Lessons 9–10: JOIN

The Western Australian Curriculum content addressed in these lessons is below.

Acquiring, managing and analysing data

Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers

Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • explain the purpose of JOIN when writing SQL queries • write queries using INNER JOIN to retrieve information from multiple tables • interpret ERDs to aid in the construction of SQL queries.

Preparation for the lesson

- A student worksheet example can be found in Appendix E.

Support notes

- A JOIN is used in SQL to combine data from multiple related tables based on a related field between them.

Teaching and learning experience

Introduction

- Demonstrate the use of a JOIN and discuss why they are useful.
- Emphasise the importance of explicitly defining the relationship between tables as part of the query. Demonstrate what happens when the relationship is not defined as part of the query (i.e. too many records are returned).

Main activities

- Class discussion: How can we retrieve related information from multiple tables?
- Demonstrate how to use INNER JOIN to retrieve information from two tables.
- Students write INNER JOIN queries to retrieve information from two tables.
- Extension: Students use INNER JOIN to retrieve information from three or more tables.
- Students complete the worksheet in Appendix E.

Lesson conclusion

- Students explain purpose of INNER JOIN to a peer as an exit ticket.

Lesson 11: ERD application

The Western Australian Curriculum content addressed in this lesson is below.

Acquiring, managing and analysing data

- Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers
- Model and query entities and their relationships using structured data

---

Lesson outline

Learning intentions
Students will: • practise writing queries • combine multiple conditional and logical operators in queries.

Preparation for the lesson

- Provide students with a new sample database to practise writing queries.
- An ERD example can be found in Appendix E.

Introduction

- Encourage students to challenge each other to come up with more complex queries.

Main activities

- Provide students with the ERD application with focus questions in Appendix E.
- Students use ERDs to complete further SQL queries across multiple tables, incorporating use of WHERE clauses with conditional and logical operators.

Lesson conclusion

- Students write a query combining multiple conditional and logical operators across two tables.

Lessons 12–16: SQL escape room

See Appendix B: Assessment task – SQL escape room

Appendix A

Resources

Resources

Software

Database tools

Students will need to use an appropriate database tool to create and run SQL queries. Appropriate tools include:

- DB Browser for SQLite (https://sqlitebrowser.org) – free, open-source desktop application
- SQLite Online (https://sqliteonline.com) – free online database tool.

There are numerous other tools that could be used, although these are both free and work across different platforms.

Diagram software

Students should also have access to software to allow them to draw diagrams. An appropriate tool can be found at Draw.io (https://app.diagrams.net/). There is also a desktop version of this software that can be downloaded from the Microsoft Store.

Useful websites

Sample data

Sample data for databases can be quickly and easily generated using the following websites:

- Mockaroo (https://mockaroo.com)
- Generate Data (https://generatedata.com).

SQLite tutorials

Further information about using SQLite can be found at:

- SQLite Tutorial (https://www.sqlitetutorial.net/)
- Tutorials Point SQLite Tutorial (https://www.tutorialspoint.com/sqlite/index.htm)
- SQLite (https://sqlite.org/index.html).

Appendix B

Assessment task

SQL escape room

Task details

Title	SQL escape room
Description	Students will design and develop an interactive quiz using an appropriate tool (e.g. Microsoft Forms) that they can share with fellow students. The quiz is intended to simulate an escape room, where answers to each clue will unlock the next part of the quiz. To successfully escape from the room, players will have to find answers to each clue by writing SQL queries against a database. Students can either create their own database or use one that is provided by their teacher.
Type of assessment	Summative
Ways of assessing	Practical tasks, visual representations, work samples
Evidence to be collected	Design document: Interactive escape room with working solutions and complete database
Suggested time	Five one-hour lessons (three lessons to design the quiz and create the database, two lessons to create the quiz)
Differentiation	Teachers should differentiate their teaching and assessment to meet the specific learning needs of their students, based on their level of readiness to learn and their need to be challenged. Where appropriate, teachers may either scaffold or extend the scope of the assessment task.

Content descriptions

Acquiring, managing and analysing data

- Analyse and visualise data interactively using a range of software, including spreadsheets and/or relational databases, to draw conclusions and make predictions by identifying trends and outliers
- Model and query entities and their relationships using structured data

Design thinking skills

Project management

- Manage projects, using suitable technologies, with an agile and collaborative approach. Use project management processes to consider time, production processes, social, ethical, economic and sustainable factors, and legal responsibilities

Investigating and defining

- Develop a design brief for a solution or to innovate an existing product, service or environment

Designing

- Design alternative solutions considering available technologies, functionality, accessibility, usability and aesthetics, using appropriate technical terms

Producing and implementing

- Select, justify, implement and test a range of technologies, techniques and processes to produce solutions and/or prototypes

Evaluating

- Evaluate design processes and solutions against student-developed criteria

Resources

- Access to a sample database (students may use this database if they are unable to create their own)
- Access to software that will allow students to write SQL queries
- Access to software to allow students to create an interactive quiz for their clues (e.g. Microsoft Forms or Google Forms)

Key concepts

- Relational database design, including:
  - tables, records, fields
  - primary key
  - foreign key.
- Structured query language (SQL)
  - use SELECT to extract data, including use of conditional and logical operators
  - use aggregate functions
  - use JOIN to extract data from multiple tables.
- Entity relationship diagrams (ERDs), including:
  - relationship between ERDs and database structure.

Task preparation

- Students are familiar with database concepts and the use of SQL to query a database.
- Students are familiar with how to use and create a quiz using appropriate software (e.g. Microsoft Forms or Google Forms).

Instructions for teacher

- Students should work independently to develop their own escape room; however, they should collaborate to seek feedback from peers and provide constructive feedback to others.
- Students can use any appropriate online tool. It is recommended that students use either Microsoft Forms or Google Forms for ease of use. Alternative implementations of an escape room include:
  - creating a series of PDF documents that are password protected
  - creating sections in OneNote that are password protected.
- Remind students that if they create their own database they need to respect copyright of that data.
- On completion of the task, ask students to share their escape rooms with their peers.
- Extensions to this task include:
  - requiring all students to create their own database
    Note: If they do this, they should include an ERD as part of their escape room to enable players to write effective SQL.
  - creating a class escape room, where each student is allocated part of the final password
  - completing all the escape room activities to find all parts of the password to this final escape room.

Instructions to students

In this task you will use your SQL skills to create your own digital escape room.

While you are required to complete the task individually, you are encouraged to seek from and provide to peers constructive feedback.

- Create a scenario that provides clear context for the escape room, with well-integrated clues and overall design. **(3 marks)**
- Create an entity relationship diagram (ERD) that represents your intended database structure that clearly shows all entities, attributes, and relationships. **(6 marks)**
- Design the clues, including a sample solution in SQL and password for each clue. Use a variety of SQL queries (SELECT, JOIN, aggregate functions, WHERE, etc.). **(4 marks)**
- Create the escape room with all the clues. Make clues more interesting by including some images and/or video to go with the text. **(8 marks)**
- Create the final escape message that is revealed when all the clues have been answered correctly.
- Players will have to use their SQL skills to solve riddles that will give them the password to each section of a quiz. The final riddle will give them the key to get out and escape the room!

Marking key

Description	Marks
Scenario and description	
Scenario provides clear context for the escape room, with well-integrated clues and overall design	3
Scenario provides adequate and supports the escape room, with detailed clues	2
Scenario and/or integration of clues is limited	1
Total	**/3**
Entity relationship diagram (ERD)	
ERD is complete, accurate, and clearly shows all entities, attributes and relationships	5
ERD clear and accurate entities; minor attribute omissions with clear relationships	4
ERD mostly accurate; minor attribute omissions or errors; with relationships	3
ERD contains several errors or missing elements and/or relationships	2
ERD missing or incorrect elements	1
Subtotal	**/5**
Correct use of crow's foot notation	1
Subtotal	**/1**
Total	**/6**
Clues	
Clues are creative, challenging, and directly test SQL skills; well-integrated with the scenario and support learning intentions	4
Clues are mostly relevant with a range of difficulty	3
Clues are basic, sometimes unclear or not fully relevant to SQL skills	2
Clues are missing, irrelevant, or do not support SQL learning	1
Total	**/4**
Structured query language (SQL) statements	
Accurate, efficient, advanced SQL statements and syntax (SELECT, JOIN, aggregate functions, WHERE, etc.)	7–8
SQL statements are correct and relevant; minor errors or inefficiencies	5–6
Limited range of correct SQL statements and/or unclear logic/errors	3–4
SQL statements are mostly incorrect or missing	1–2
Total	**/8**
Task total	**/21**

Appendix C

Task design document

Purpose of task design document

As part of the development process for this task, students should investigate ideas and design a solution. To assist with this process, they should create a task design document using a suitable word processor. The template provided below gives a possible structure for this document, allowing students to focus on creating ideas and developing their database structure. It is suggested that students are provided with a copy of the template that they can work from to develop their ideas and demonstrate their understanding of database structure and structured query language (SQL).

If required, students could also add a section at the end of the design document for an evaluation, where they can reflect on the success of their project and how they might improve in the future.

Structure of task design document

Room name (add title here):

General information

Developer name:	
Overview:	(One to two sentences)

Description

Write a background story to give your escape room some context and make it more interesting to play. For example, you could be a visitor from outer space looking for a way home, or you could be trying to hunt down a thief who has stolen the codes to your favourite game.

Scenario:	(Detailed description of the escape room scenario)

Database structure

Draw an entity relationship diagram (ERD) for your database

Clues

Using your scenario as a starting point, write a series of questions that will need to be answered using SQL. You have been provided with a database file that has several different data sets that you can use. Make sure that you provide enough detail in the question so that the player has enough information to be able to construct a valid SQL statement.

For each clue, you should complete the planning table below. Create a short story to go with each clue to make the game more interesting.

Clue 1

Clue name	
Story	
Clue	
Password	
SQL solution	

Clue 2

Clue name	
Story	
Clue	
Password	
SQL solution	

Clue 3

Clue name	
Story	
Clue	
Password	
SQL solution	

Clue 4

Clue name	
Story	
Clue	
Password	
SQL solution	

Sample design document

Room name: The Cipher Stone

General information

Developer name:	Jimmy Martin
Overview:	The legendary ‘Cipher Stone’ has gone missing and you need to locate it.

Description

Write a background story to give your escape room some context and make it more interesting to play. For example, you could be a visitor from outer space looking for a way home, or you could be trying to hunt down a thief who has stolen the codes to your favourite game.

Scenario:	In the heart of a Perth museum, a mysterious artefact has gone missing from a top-secret exhibit. The artefact is the legendary ‘Cipher Stone’ – an ancient relic believed to contain hidden knowledge from a long-forgotten civilisation. The museum’s lead curator, Dr Elena Rodriguez, has discovered that the thief has hidden the stone in a secret room with a special password. Luckily, she has left a trail of clues that will reveal the password.

Database structure

Draw an entity relationship diagram (ERD) for your database.

Clues

Using your scenario as a starting point, write a series of questions that will need to be answered using structured query language (SQL). You have been provided with a database file that has several different data sets that you can use. Make sure that you provide enough detail in the question/s to provide the player with enough information to be able to construct a valid SQL statement.

For each clue, you should complete the planning table below. Create a short story to go with each clue to make the game more interesting.

Clue 1

Clue name	Weighty stone
Story	The first part of the password can be found inside an image display cabinet in the middle of the entrance foyer. The only problem is it has a weight sensor hidden inside that will destroy the clue if triggered. The only way to retrieve the clue is to replace it with another object of the same weight on the sensor. But how much weight do you need?
Clue	There is a picture of South America with a lizard sitting on it. How heavy are all the reptiles from South America?
Password	510
SQL solution	SELECT SUM(weight_kg) FROM animals WHERE class = "Reptilia" AND region = "South America"

Clue 2

Clue name	Camping delights
Story	The second part of the password is in the Natural Wonders section of the museum, hidden in a display about one of the national parks of Western Australia.
Clue	Find the name of the national park in Western Australia that does not allow camping.
Password	NAMBUNG
SQL solution	SELECT park_name FROM australian_national_parks WHERE state = "WA" AND camping_allowed = "FALSE"

Clue 3

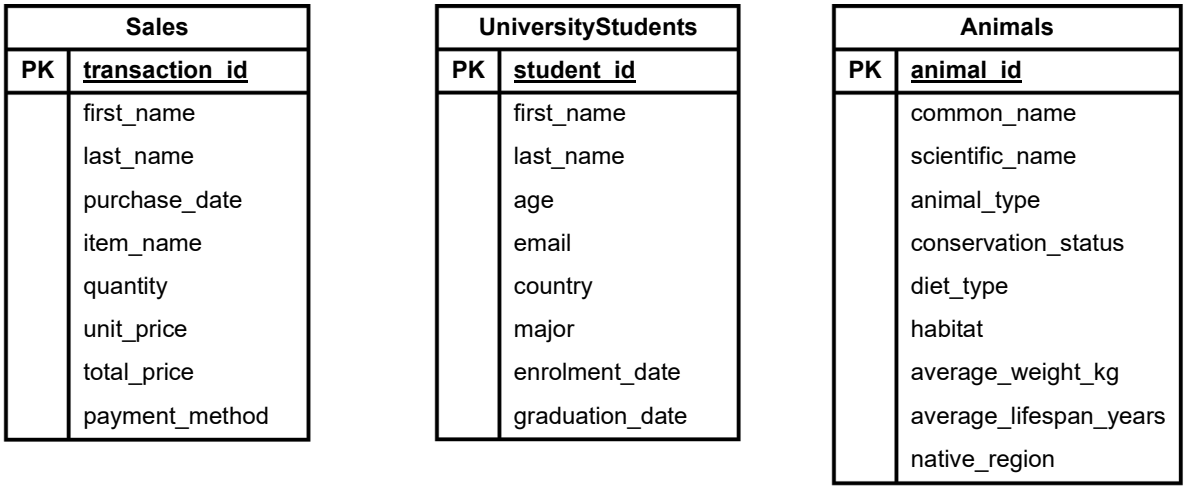
Clue name	Tackling machine
Story	The thief likes people who work hard and help others! The final part of the password is the number of players in local AFL clubs who have made at least seven tackles a game in the first three rounds of the 2025 AFL season.
Clue	Find the number of players from Western Australian clubs who have made at least seven tackles in the first three rounds of the 2025 AFL season.
Password	6
SQL solution	SELECT player, team, tackles FROM afl_2025 WHERE (team = 'FRE' OR team='WCE') AND tackles >= 7

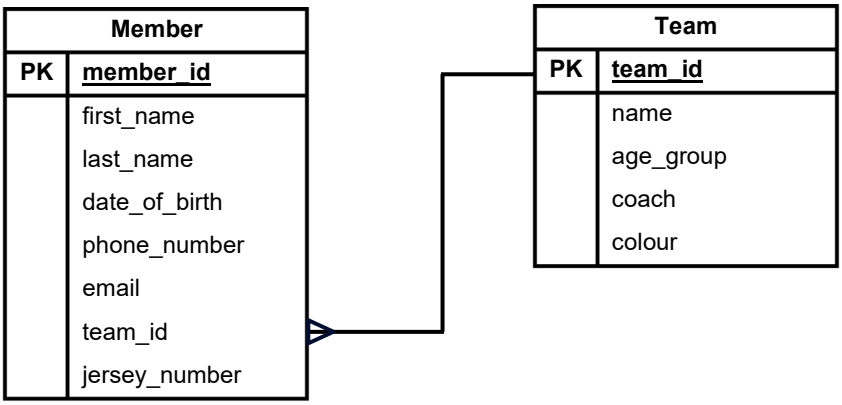
Appendix D

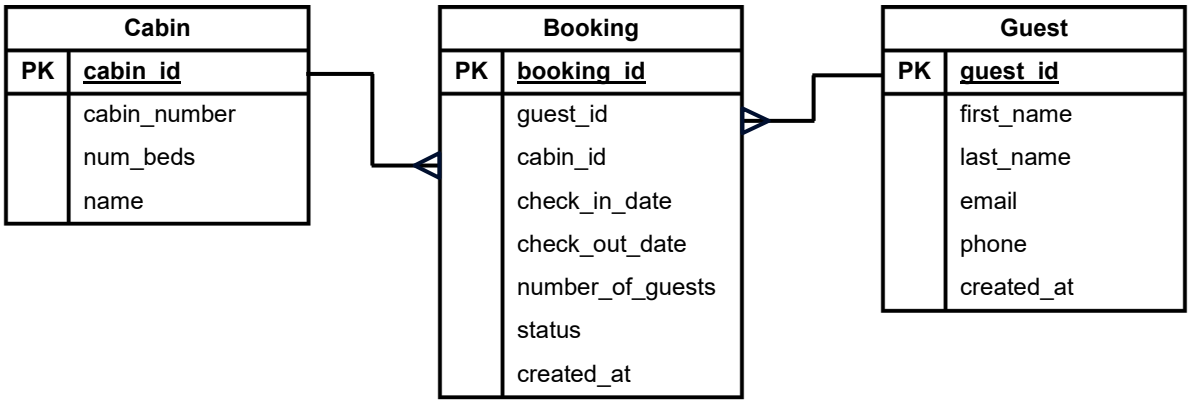
Teaching resources

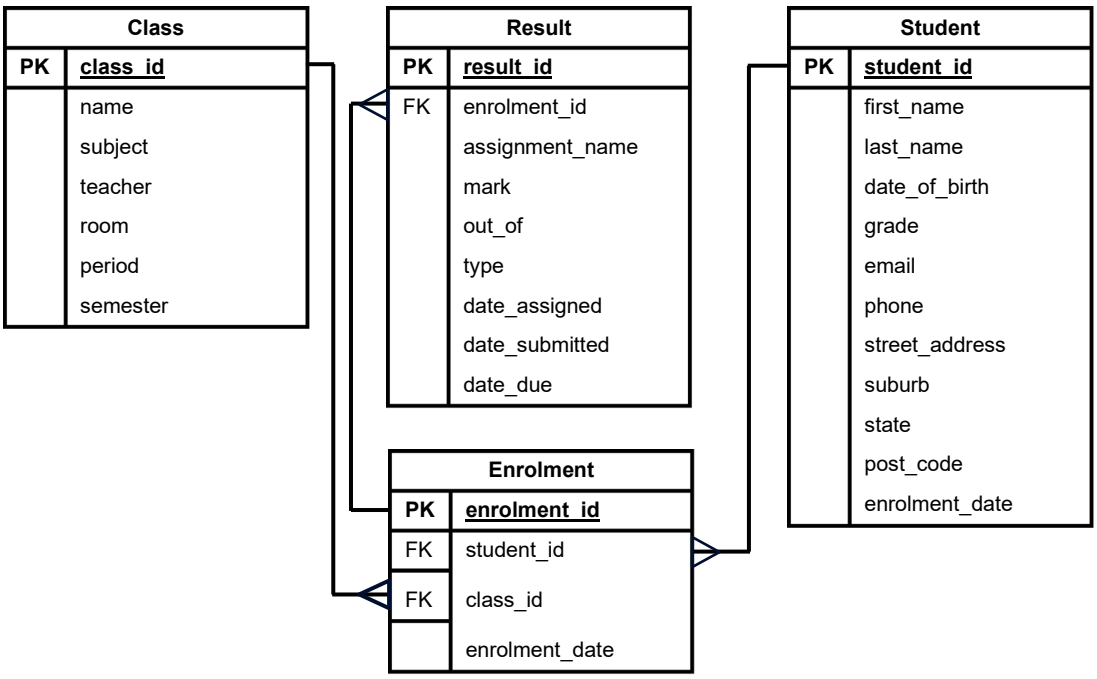
Sample entity relationship diagrams (ERD)

The following ERDs can be used to assist students with the worksheet activities.









SQL to create tables for database activities

The following SQL script can be used to create an empty database that will match the ERDs above.

```
/*################################
# SET UP EMPTY DATABASE
#
# Drop the tables from the database if they exist so we can run the CREATE
statements
# NOTE: If there are foreign keys defined in the database, tables need to be
dropped in the correct order
# to maintain referential integrity
*/
DROP TABLE IF EXISTS Sales;
DROP TABLE IF EXISTS UniversityStudents;
DROP TABLE IF EXISTS Animals;
DROP TABLE IF EXISTS Team;
DROP TABLE IF EXISTS Member;
DROP TABLE IF EXISTS Cabin;
DROP TABLE IF EXISTS Guest;
DROP TABLE IF EXISTS Booking;
DROP TABLE IF EXISTS Student;
DROP TABLE IF EXISTS Class;
DROP TABLE IF EXISTS Enrolment;
DROP TABLE IF EXISTS Result;

/*
# Update the database to allow foreign keys to enforce referential integrity
*/
PRAGMA foreign_keys = ON;
```

```
/*
##############################
# CREATE EMPTY TABLES
#
# Create each individual table in the database.
# NOTE: Tables need to be created in correct order to ensure foreign key
constraints will work
*/

CREATE TABLE Sales (
        transaction_id INTEGER PRIMARY KEY AUTOINCREMENT,
        first_name TEXT NOT NULL,
        last_name TEXT NOT NULL,
        purchase_date DATETIME NOT NULL,
        item_name TEXT NOT NULL,
        quantity INTEGER,
        unit_price REAL,
        total_price REAL,
        payment_method TEXT
        );

CREATE TABLE UniversityStudents (
        student_number INTEGER PRIMARY KEY AUTOINCREMENT,
        first_name TEXT NOT NULL,
        last_name TEXT NOT NULL,
        age INTEGER,
        email TEXT NOT NULL,
        country TEXT,
        major TEXT,
        enrolment_date DATE,
        graduation_date DATE
        );

CREATE TABLE Animals (
        animal_id TEXT PRIMARY KEY,
        common_name TEXT NOT NULL,
        scientific_name TEXT NOT NULL,
        animal_type TEXT NOT NULL,
        conservation_status TEXT NOT NULL,
        diet_type TEXT,
        habitat TEXT,
        average_weight_kg FLOAT,
        average_lifespan_years INTEGER,
        native_region TEXT
        );

-- Create tables for Sports club
CREATE TABLE Team (
        team_id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT NOT NULL,
        age_group TEXT NOT NULL,
        coach TEXT,
        colour TEXT
        );

CREATE TABLE Member (
        member_id INTEGER PRIMARY KEY AUTOINCREMENT,
        first_name TEXT NOT NULL,
        last_name TEXT NOT NULL,
        date_of_birth DATE,
        phone_number TEXT,
```

```
        email TEXT NOT NULL,
        team_id INTEGER,
        jersey_number INTEGER,
        FOREIGN KEY(team_id) REFERENCES Team(team_id)
        );

-- Create tables for Holiday Resort
CREATE TABLE Cabin (
        cabin_id INTEGER PRIMARY KEY AUTOINCREMENT,
        cabin_number INTEGER NOT NULL,
        num_beds INTEGER NOT NULL,
        name TEXT
        );

CREATE TABLE Guest (
        guest_id INTEGER PRIMARY KEY AUTOINCREMENT,
        first_name TEXT NOT NULL,
        last_name TEXT NOT NULL,
        phone_number TEXT,
        email TEXT,
        created_at DATETIME
        );

CREATE TABLE Booking (
        booking_id INTEGER PRIMARY KEY AUTOINCREMENT,
        guest_id INTEGER NOT NULL,
        cabin_id INTEGER NOT NULL,
        check_in_date DATETIME,
        check_out_date DATETIME,
        number_of_guests INTEGER,
        status TEXT,
        created_at DATETIME,
        FOREIGN KEY(guest_id) REFERENCES Guest(guest_id),
        FOREIGN KEY(cabin_id) REFERENCES Cabin(cabin_id)
        );

-- Create tables for School
CREATE TABLE Student (
        student_id TEXT PRIMARY KEY,
        first_name TEXT NOT NULL,
        last_name TEXT NOT NULL,
        date_of_birth DATE NOT NULL,
        grade INTEGER NOT NULL,
        email TEXT NOT NULL,
        phone TEXT NOT NULL,
        street_address TEXT,
        suburb TEXT,
        state TEXT,
        postcode INTEGER,
        enrolment_date DATE
        );

CREATE TABLE Class (
        class_id TEXT PRIMARY KEY,
        name TEXT NOT NULL,
        subject TEXT NOT NULL,
        teacher TEXT,
        room TEXT,
        period INTEGER,
        semester TEXT
        );
```

```
CREATE TABLE Enrolment (
        enrolment_id TEXT PRIMARY KEY,
        class_id TEXT NOT NULL,
        student_id TEXT NOT NULL,
        enrolment_date DATE,
        FOREIGN KEY(class_id) REFERENCES Class(class_id),
        FOREIGN KEY(student_id) REFERENCES Student(student_id)
        );

CREATE TABLE Result (
        result_id TEXT PRIMARY KEY,
        enrolment_id INTEGER NOT NULL,
        assignment_name INTEGER NOT NULL,
        mark INTEGER,
        out_of INTEGER NOT NULL,
        type TEXT NOT NULL,
        date_assigned DATETIME,
        date_submitted DATETIME,
        date_due DATETIME,
        FOREIGN KEY(enrolment_id) REFERENCES Enrolment(enrolment_id)
        );
```

Appendix E

Student worksheets

Lesson 1: Database basics

- Outline where you see databases in daily life. State three examples.
- Spreadsheets versus databases. For each sentence below, identify if it is a feature of a spreadsheet or a database.
  - Best for storing large volumes of data: Spreadsheet/database.
  - Best for quick calculations and charts: Spreadsheet/database.
  - Organised into tables, records, and fields: Spreadsheet/database.
- Explain why we need databases. Include at least two reasons as part of your response.
- Draw the structure of a simple database (table, fields, records).
- Identify components in the sample database below:

Student_ID	First_name	Last_name	Grade
S001	Alex	Brown	10
S002	Mia	Smith	9

Table name: _____

Fields: _____

Records: _____

Extension (optional)

- Research task: Find one real-world example of a database.

  Name of the system: _____

  Describe the kind of data it stores.

Lesson 2: Datatypes

- Describe the purpose of data types.
- Identify the correct data type for each field:
  - Age
  - First_name
  - IsMember (TRUE/FALSE)
  - Date_of_birth
- Design a simple database table on a topic of your choice. Include a table name, field names and data types.
- Create two sample records for your database.
- Peer review: Swap your table with a partner and suggest possible improvements.
- Outline how we can uniquely identify each record in a table. Explain the role of a primary key.
- Add a primary key to your table design (or identify an existing field which could be a primary key).

Extension (optional)

- Research task: Explain the term composite key and how it can be used in database design. Provide an example.

Lesson 3: SQL

- Outline the definition of SQL. Describe why it is useful.
- State what you think a SELECT query will do.

Use an appropriate database tool to run the following queries:

- List all fields of all the animals.
- List all the details of all the sales transactions.
- List all fields of the university students.
- List the first name, last name and country of all the university students.
- List the common name and habitat of all the animals.
- List the purchase date, item name and quantity of all the sales.

Lessons 4–5: SQL 2

- Explain what the WHERE clause does in SQL.
- Describe how ORDER BY can help when viewing the results of a query.

Use an appropriate database tool to run the following queries:

- List the first_name, last_name and transaction date of all sales that have been paid with a credit card.
- List the purchase date, quantity and total price for all sales of potatoes.
- Find the common name and scientific name of all marsupials in the Animals table.
- Find the common name and habitat of the carnivores that are endangered.
- List the first name, last name, email address and country of all the university students from either Canada or United States.
- Find all the university students with an email address that ends with '.com'.
- Find all the university students who will graduate in 2026. List their first name, last name, major and graduation date.
    - What is my common name? I can be found in arid places and eat insects.
    - What is my name? I bought 10 pieces of chicken with cash.
    - How old am I? I am from China and am studying art at university. My last name starts with 'J'.

Extension (optional)

- Write your own SQL puzzle and share it with your classmates. Make sure you can write the solution to your SQL puzzle too!

Lesson 6: SQL 3

- State the purpose of COUNT in SQL.
- Describe how SUM and AVG can help when analysing data.

Use an appropriate database tool to run the following queries:

- Write a query to count the number of university students in the UniversityStudents table.
- Write a query to find the number of herbivores in the Animals table.
- Write a query to find the average life expectancy of all the animals in the Animals table.
- Write a query to find the average weight of all the birds.
- Write a query to find the common name and type of the heaviest animal.
- Write a query to find the average age of all the university students who study Political Science or Anthropology.
- What is my scientific name? I have the shortest life expectancy of any animal!

Extension (optional)

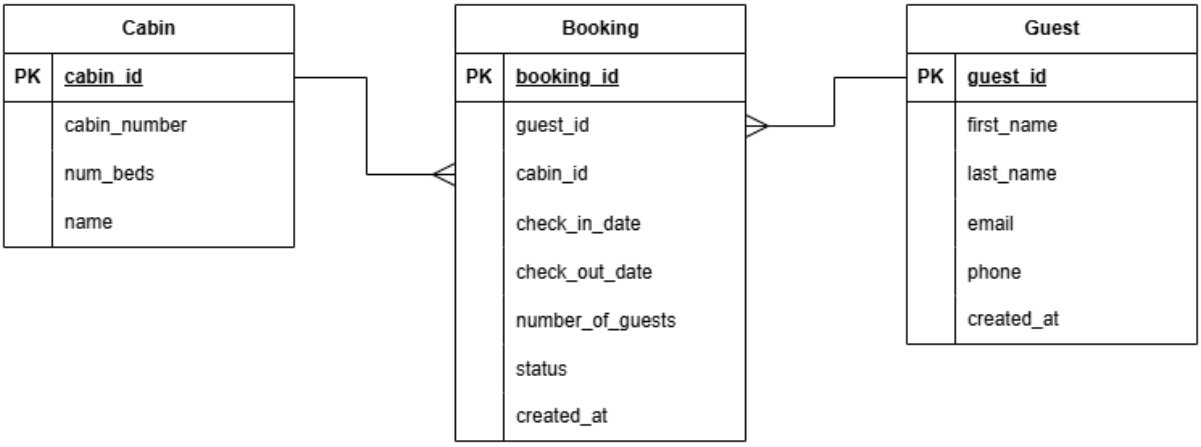
- Write your own SQL puzzle using an aggregate function and share it with your classmates. Make sure you can write the solution too!

Lesson 7: ERDs

- Explain why we need relationships between tables in a relational database.
- Describe some of the problems that could occur if we store all the data in one table. Give an example to help explain your answer.
- Explain what is meant by the term one-to-many relationship. Give an example of a one-to-many relationship.
- Consider the following related tables:
  Customer(customer_ID, first_name, last_name, phone)
  Sales(sale_ID, date, cost, customer_ID)
  - Identify the primary key for each table by underlining it.
  - Identify the foreign key by circling it.
- Design a simple relational database with two tables on a topic of your choice (e.g. a library, with Book and Borrower tables).
- Peer review: Swap your database design with a partner and share possible improvements.

Lesson 8: ERD 2

- State what the letters ERD stand for.
- Describe the purpose of an ERD.
- Identify three components of an ERD.
- Consider the following ERD for a small holiday resort:



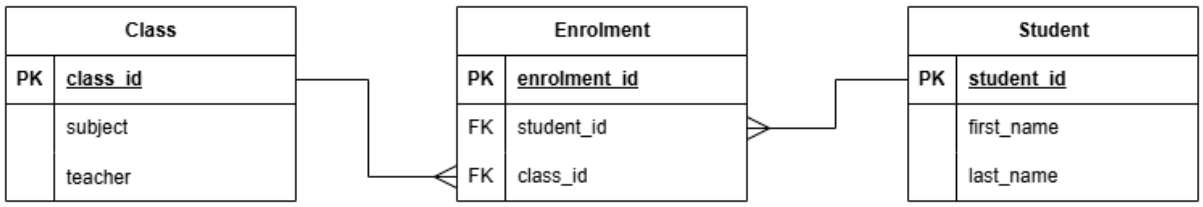
  - State which table contains the foreign key that references cabin.
  - State the primary key for the guest table.
  - State the relationship between a booking and a guest.
  - State the relationship between a cabin and a booking.
  - Consider the relationship between a booking and a guest. Identify what side of the relationship the foreign key is on.
- Draw an ERD for the database that you designed in the previous lesson.

Extension (optional)

- Create a new ERD for a database system that requires at least three tables.

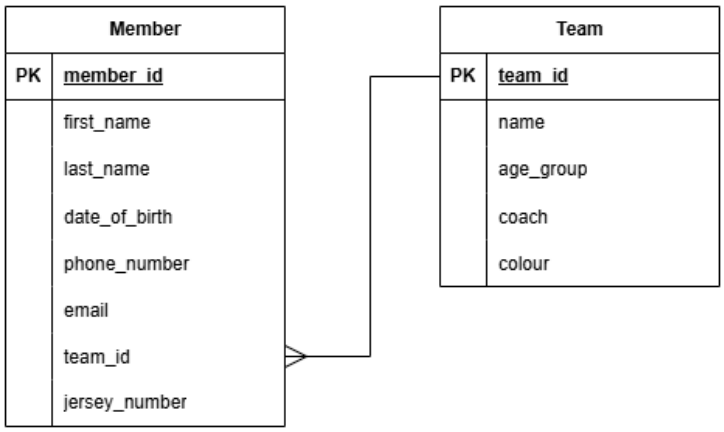
Lessons 9–10: JOIN

- Describe the purpose of a JOIN in SQL.
- Explain why we sometimes need to get information from more than one table.
- Consider the following ERD:
  - State which table acts as a bridge between a student and their class.
  - Identify the foreign keys in Enrolment.



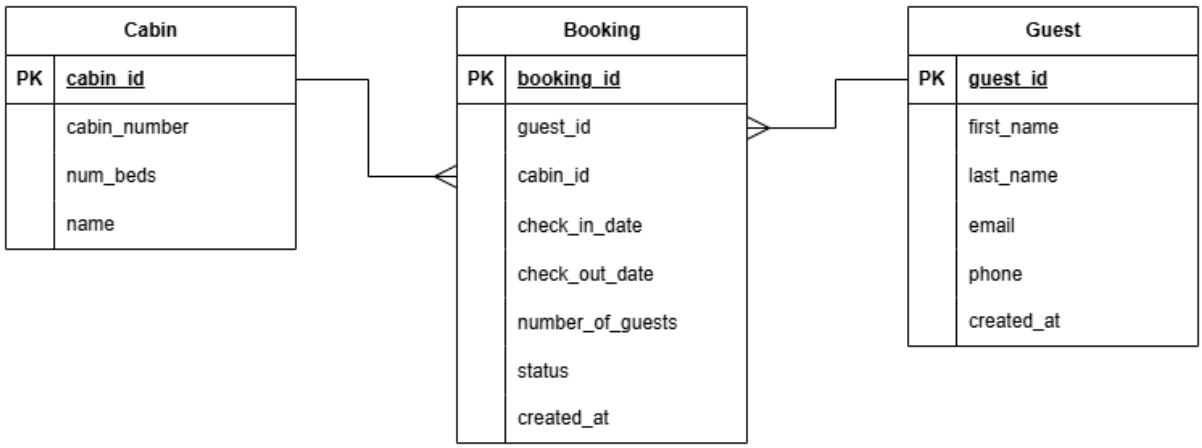
Use an appropriate database tool to run the following queries for each of the ERDs provided.

Sports club



- Write a query to list the name of each member of the club and the team name and colour that they are part of.
- Find all the players who are coached by Laura Wilson. List their full name and phone number.
- Find all the players with a jersey number of 11. List their full name, team name, coach and team colour.

Holiday resort



- Find all the bookings for cabins with at least four beds. List the cabin name, check-in date and status of each booking.
- Find the names of all the guests who have booked the cabin Magpie.
- Find the number of bookings that have been made for the cabin Ibis.
- List all the bookings made by James Taylor.
  - List the check-in and check-out dates and the name of the cabin for each booking.

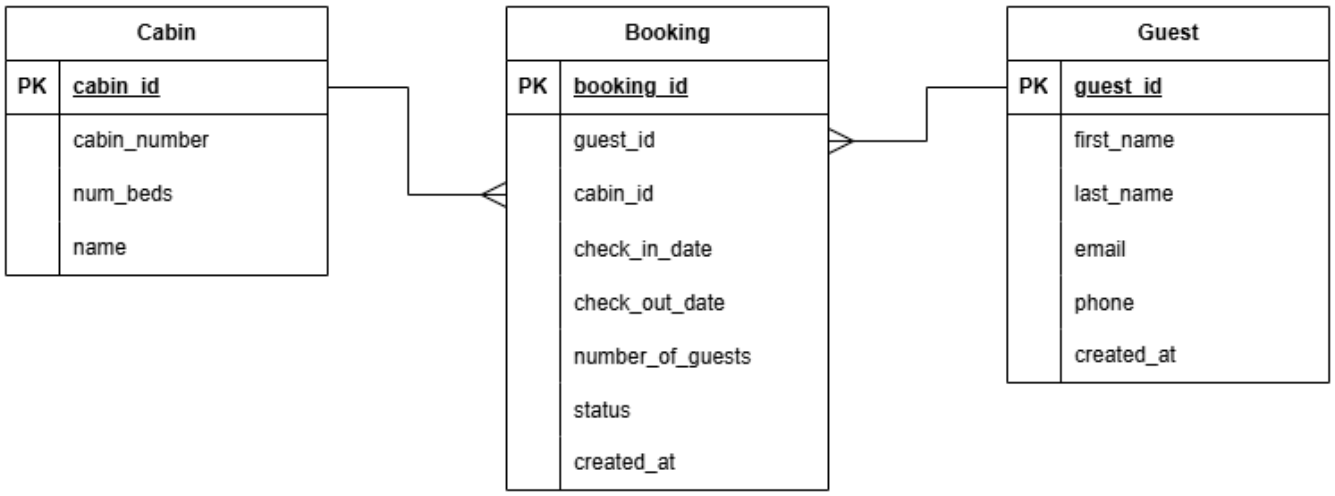
Extension (optional)

- Write your own SQL puzzle using a JOIN and share it with your classmates. Make sure you can write the solution too!

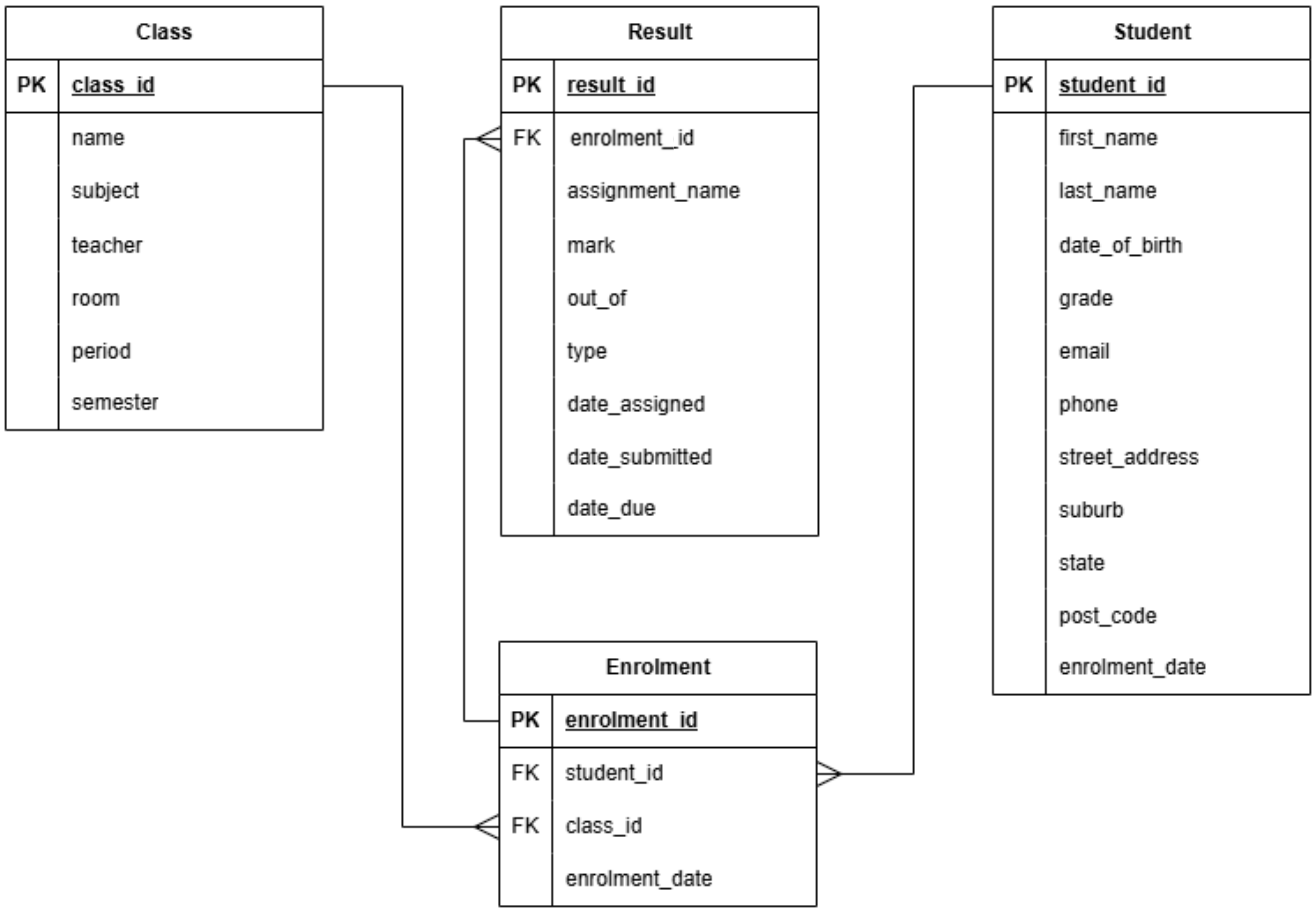
Lesson 11: ERD application

Consider the following ERDs:

Holiday resort



Classes and results



Use an appropriate database tool to run the following queries:

- Find the number of cabins that have five beds.
- Find the number of bookings that have been made for cabins with less than four beds.
- Find all the bookings that have been cancelled. List the name of the guest and the name of the cabin, as well as the dates for the booking.
- Find all the teachers for the student Olivia Harris.
  - List the subject and teacher name.
- Find all the assessments that Jacob Wood has completed. List the name of the class, the assignment name and the mark for each assessment.

Extension (optional)

- Write your own SQL puzzles based on the ERDs above and share them with your classmates. Make sure you can write the solution!