



**Australian Government**

**Department of Education, Employment and Workplace Relations**

# **ICAGAM419A Build a database to support a computer game**

**Release: 1**

## ICAGAM419A Build a database to support a computer game

### Modification History

Release	Comments
Release 1	This Unit first released with <i>ICAI1 Information and Communications Technology Training Package version 1.0</i>

### Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to build a database to support a computer game.

### Application of the Unit

This unit applies to game programmers or junior analyst programmers responsible for the development and implementation of a database to support state management for a computer game.

This unit relates to the specific requirements for designing and implementing a relational database required for the management of persisted state of a computer game. The game may be an online or networked game or a stand-alone game where player actions are preserved or persisted between sessions.

This unit does not address specialist programming skills required for games programming which are detailed in other units of competency.

### Licensing/Regulatory Information

No licensing, legislative, regulatory or certification requirements apply to this unit at the time of endorsement but users should confirm requirements with the relevant federal, state or territory authority.

### Pre-Requisites

Not applicable.

### Employability Skills Information

This unit contains employability skills.

## Elements and Performance Criteria Pre-Content

<b>Element</b>	<b>Performance Criteria</b>
<i>Elements describe the essential outcomes of a unit of competency.</i>	<i>Performance criteria describe the performance needed to demonstrate achievement of the element. Where bold italicised text is used, further information is detailed in the required skills and knowledge section and the range statement. Assessment of performance is to be consistent with the evidence guide.</i>

## Elements and Performance Criteria

<p>1. Verify database model</p>	<p>1.1 Verify that <i>game architecture</i> and <i>system implementation requirements</i> have been addressed by the proposed <i>database architecture</i></p> <p>1.2 Correlate <i>data model</i> entities and their attributes with the <i>game artefacts</i> and <i>game design requirements</i></p> <p>1.3 Verify that <i>entity relationships</i> defined in the data model implement game design and game-play rules</p> <p>1.4 Verify estimates of amounts of data storage required and data types defined by the data model</p> <p>1.5 Document design inconsistencies</p> <p>1.6 Review database model and address and correct inconsistencies</p> <p>1.7 Define database test data</p>
<p>2. Create physical database design</p>	<p>2.1 Implement conceptual data model design as a physical database design</p> <p>2.2 Determine default database file size according to the data model storage estimates</p> <p>2.3 Determine game data <i>input and output workload</i> estimates</p> <p>2.4 Determine reliability and performance requirements</p> <p>2.5 Define minimum hardware requirements to support input and output workload and reliability or performance requirements</p> <p>2.6 Configure the database management system to use minimum hardware requirements</p> <p>2.7 Refine and confirm database test data</p>
<p>3. Implement physical database prototype</p>	<p>3.1 Write structured query language (SQL) scripts to create physical database files according to design requirements</p> <p>3.2 Write SQL scripts to create database tables</p> <p>3.3 Implement primary key constraints</p> <p>3.4 Implement foreign key constraints</p> <p>3.5 Define and script stored procedures for selecting, inserting, updating and deleting data</p> <p>3.6 Define and script Insert statements for test data</p> <p>3.7 Execute SQL scripts and database, including test data</p>

4. Test the database and document results	4.1 Document entity integrity tests and results 4.2 Document referential integrity tests and results 4.3 Document input and output performance tests and results 4.4 Document stored procedures tests and results
5. Review, evaluate and correct database performance	5.1 Review test result deficiencies 5.2 Define, evaluate and test deficiencies and ensure corrective measures are implemented 5.3 Implement corrections to database

## Required Skills and Knowledge

*This section describes the skills and knowledge required for this unit.*

### Required skills

- analytical skills to:
  - analyse and assess quantitative requirements for data storage and data throughput
  - analyse game architecture and game system implementation requirements in the context of the proposed database architecture
  - assess how well the implementation of the data model maps to the game artefacts, the game-play rules and the game design requirements
- communication skills to:
  - identify and discuss game-play rules with game designers
  - define and describe data throughput requirements and discuss these with game designers
  - negotiate with project managers to determine database implementation resource requirements, project timeframes and deliverables
  - propose data structures to implement game-play functionality and system requirements
- initiative and enterprise skills to:
  - assess proposed database architectures within the context of existing enterprise technologies, infrastructure and industry standards
  - recommend solutions to problems or deficiencies associated with the implementation of the database
- learning skills to:
  - acquire the skills and knowledge required to overcome current limitations in data throughput and query execution times
  - stay at the cutting edge of industry developments, industry standards and innovative applications of relational database management systems to problems of persisting state in computer games
- literacy skills to:
  - document physical design inconsistencies
  - interpret documentation describing game design requirements and game-play rules
  - interpret related game industry standards and associated database management system standards and query language standards
  - write requirement specifications for physical database implementation
- planning and organising skills to:
  - organise resources, documentation and infrastructure elements required for the implementation of a relational or object-oriented database management system
  - plan an effective implementation to a relational or object-oriented database management system
- problem-solving skills to:
  - debug query scripts, identify script errors and resolve all query scripting issues
  - identify and implement solutions to problems with the initial conceptual database model

- identify problems with physical model and implement solutions
- research skills to:
  - enhance knowledge of industry standards and industry trends related to computer game state management techniques and relational or object-oriented database management systems
  - identify and locate sources of information that provide solutions to technical problems
  - identify sources of information and documentation required to build a database to support a computer game
- technical skills to:
  - execute data and query analysis tools and evaluate data input and output performance statistics
  - redefine data structures to improve the efficiency of data throughput
  - use an integrated development environment to build, modify and manipulate relational or object-oriented data structures
  - write code to implement a physical database design.

### Required knowledge

- computer hardware specifications and hardware performance expectations
- creation, use and testing of stored procedures:
  - defining input parameters
  - defining output parameters
- database stress testing and stress testing software
- database testing methodologies
- object-oriented database management systems
- relational database management systems:
  - IBM DB2
  - Microsoft Access
  - Microsoft SQL (MS SQL) server
  - MySQL
  - Oracle
  - Postgre Structured Query Language (Postgre SQL)
- relational database modelling building blocks and theory:
  - attributes
  - entity relationship modelling
  - normalisation of database tables
  - relationship cardinality and connectivity:
    - 1:1
    - 1:M
    - M:N
- SQL, such as:
  - data definition language
  - data manipulation language

- evolution of relational database management systems and their relationship to the flat file, hierarchical and network data storage legacy of games
- important functions of a relational database management system within the context of a computer game, such as:
  - controlling data access for multiple users
  - interpreting data access languages, such as SQL and LINQ
  - interpreting data application programming interfaces (APIs), such as ADO.Net
  - managing data integrity
  - managing disaster recovery and backups
  - managing the data dictionary
  - managing the data storage
  - managing the security requirements of the data
  - transforming and presenting data
- importance of data models
- different types of data models, such as Chen and Crow's Foot models
- relationship between game-play rules and the modelling of the database structure.



## Evidence Guide

*The evidence guide provides advice on assessment and must be read in conjunction with the performance criteria, required skills and knowledge, range statement and the Assessment Guidelines for the Training Package.*

<b>Overview of assessment</b>	
<b>Critical aspects for assessment and evidence required to demonstrate competency in this unit</b>	<p>Evidence of the ability to:</p> <ul style="list-style-type: none"> <li>• interpret a conceptual database model and create a physical database design from the conceptual model</li> <li>• create a database that allows for the persistence of state data for a computer game</li> <li>• evaluate and improve the performance of a database built to support a computer game.</li> </ul>
<b>Context of and specific resources for assessment</b>	<p>Assessment must ensure access to:</p> <ul style="list-style-type: none"> <li>• computer aided software engineering and modelling tools</li> <li>• relational or object-oriented database management system</li> <li>• performance testing and analysis software for the selected database management system</li> <li>• game architecture specifications</li> <li>• game design and requirements documentation</li> <li>• client or server hardware and networking infrastructure sufficient to simulate the game architecture specifications</li> <li>• appropriate learning and assessment support when required</li> <li>• modified equipment for people with special needs.</li> </ul>
<b>Method of assessment</b>	<p>A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit:</p> <ul style="list-style-type: none"> <li>• written or verbal evaluation of knowledge of data modelling techniques</li> <li>• written or verbal evaluation of knowledge of scripting languages, such as SQL and LINQ</li> <li>• evaluation of the implementation of a conceptual model as a physical model</li> <li>• review of a database created to support a computer game</li> <li>• review of a database performance measurement process and evaluation techniques.</li> </ul>
<b>Guidance information for assessment</b>	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate.</p> <p>Assessment processes and techniques must be culturally</p>

	<p>appropriate, and suitable to the communication skill level, language, literacy and numeracy capacity of the candidate and the work being performed.</p> <p>Indigenous people and other people from a non-English speaking background may need additional support.</p> <p>In cases where practical assessment is used it should be combined with targeted questioning to assess required knowledge.</p>
--	---

## Range Statement

The range statement relates to the unit of competency as a whole. It allows for different work environments and situations that may affect performance. Bold italicised wording, if used in the performance criteria, is detailed below. Essential operating conditions that may be present with training and assessment (depending on the work situation, needs of the candidate, accessibility of the item, and local industry and regional contexts) may also be included.

<p><b><i>Game architecture</i></b> may include:</p>	<ul style="list-style-type: none"> <li>• networked, persistent games, including massively multi-player online (MMO) games, such as:             <ul style="list-style-type: none"> <li>• World of Warcraft</li> <li>• Second Life implemented in a client (PC), server architecture using web-based protocols</li> </ul> </li> <li>• networked proprietary console game architectures implemented on hardware, such as:             <ul style="list-style-type: none"> <li>• PlayStation</li> <li>• Xbox</li> <li>• Wii</li> <li>• Nintendo</li> <li>• iPod Touch</li> <li>• Zone 40</li> <li>• where game state is persisted between sessions</li> </ul> </li> <li>• non-networked persistent games.</li> </ul>
<p><b><i>System implementation requirements</i></b> may include:</p>	<ul style="list-style-type: none"> <li>• games that persist current game state and consequently must manage an authoritative store of state data using a client-server architecture</li> <li>• non-networked stand-alone games that persist game state between sessions.</li> </ul>
<p><b><i>Database architecture</i></b> may include:</p>	<ul style="list-style-type: none"> <li>• client-server based architectures that use:             <ul style="list-style-type: none"> <li>• relational database management systems</li> <li>• object-oriented database management systems.</li> </ul> </li> </ul>
<p><b><i>Data model</i></b> may include:</p>	<ul style="list-style-type: none"> <li>• Chen Entity relationship diagrams</li> <li>• Crow's Foot entity relationship diagrams</li> <li>• data dictionaries</li> <li>• object data models</li> <li>• relational data models.</li> </ul>
<p><b><i>Game artefacts</i></b> may include:</p>	<ul style="list-style-type: none"> <li>• characters</li> <li>• maps</li> <li>• objects, weaponry, machinery and prizes</li> <li>• players</li> <li>• regions, geographical locations and game servers.</li> </ul>

<p><b><i>Game-design requirements</i></b> may include:</p>	<ul style="list-style-type: none"> <li>• design rules and limitations</li> <li>• level customisations and game level design</li> <li>• level definitions</li> <li>• level progression requirements</li> <li>• play rules.</li> </ul>
<p><b><i>Entity relationships</i></b> may include:</p>	<ul style="list-style-type: none"> <li>• assessed and implemented cardinality requirements</li> <li>• assessed and implemented connectivity requirements</li> <li>• consideration of degree of normalisation of tables and referential integrity constraints</li> <li>• consideration of entity integrity constraints.</li> </ul>
<p><b><i>Input and output workload</i></b> may include estimates of:</p>	<ul style="list-style-type: none"> <li>• index read, modify and rebuild rates</li> <li>• number of database transaction log entries generated</li> <li>• number of random disk read writes</li> <li>• number of sequential disk read writes.</li> </ul>

## Unit Sector(s)

Game development