Database Applications
ISYS1102/ 3479 | Semester 1 2025
Assignment 1: Database Design and Optimisation

Assessment Type	Database Design, Database Optimisation, Implementation, and Demonstration Individual Assessment
Due Date (M1)	Week 4, during the lab sessions. Further details will be provided on Canvas.
Due Date (M2)	23:59 Sunday 13 April 2025
Demonstrations (M3)	Week 7 - 8, during the lab sessions. Further details will be provided on Canvas.
Silence Period (M1)	None
Silence Period (M2 & M3)	Starts at 5:00PM Friday 11 April 2025
Weight	Task 1: 50 marks Task 2: 20 marks Task 3: 30 marks
Submission	Online, via Canvas. Submission instructions are provided on Canvas.

1 Overview

1.1 Assessment Criteria

This assessment will determine your ability to:

1. analyse the requirements outlined in the problem description;
2. develop a conceptual model to assist you with the design of the database backend required for the system;
3. use an industry-standard database design documents to communicate database requirements and produce a comprehensive database specifications report;
4. identify and implementing efficient storage strategies for extremely large tables;
5. identify and implementing efficient indexing and partition strategies for extremely large tables;

6. write efficient queries on extremely large tables and describe the query plans that query optimiser would likely be using and explain how a cost-based query optimiser would execute such query plans;
7. write stored T-SQL procedures and functions to automate common tasks in an SQL Server database.

1.2 Learning Outcomes

This assessment will assess how you attained the following course learning outcomes:

CLO 1: apply advanced data analysis and modeling concepts, physical design, integrity, security and transaction management.

CLO 2: create stored procedures and functions to enhance the usability of a database;

CLO 3: apply techniques for efficient storing, accessing, securing, and recovering of data;

CLO 4: build an efficient database application with an emphasis on storage management, indexing, and query optimization.

2 Assessment Details

2.1 Preparation Work

You are required to implement the database backend for the below mentioned application on Microsoft SQL server. In order to successfully complete these tasks you must have completed Week 1 – 5 lab sheets and ensure that your SQL Server account is correctly configured and Azure Data Studio on your laptop is configured correctly.

2.2 Assignment Task Description

Introduction

Victoria's public transport ticketing system (MyKi) is aging and up for a replacement. The governing body – Public Transport Victoria (PTV) – has called for tenders for a replacement system. The multi-national infrastructure development company, in which you just joined, won the contract. You are now in the project team, with your database skills, and tasked to design the database back-end.
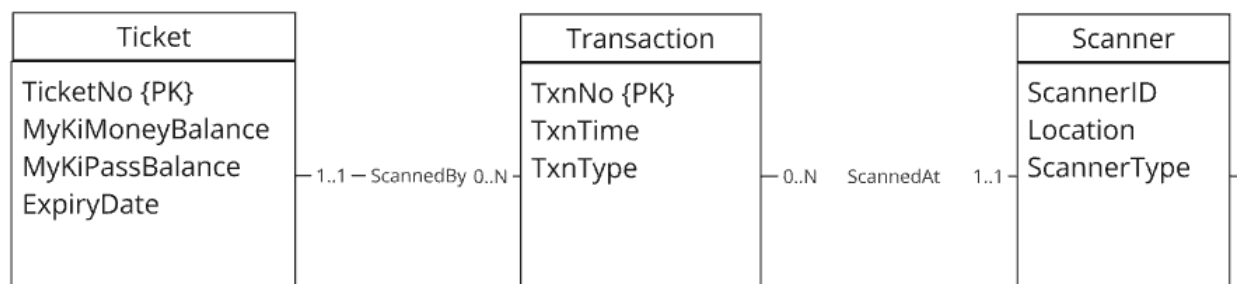
In the Phase 1 of the project, it is not expected to change the basic ticketing guidelines (such as zones, MyKi Money and MyKi Pass options, or the pricing). In this phase, you will simply replace the aging infrastructure, including the substantially slow database infrastructure. In fact, after your initial investigation on the performance issues in the current system, you found out that database infrastructure is one of the bottlenecks in the system and it slows down the operation of the whole system.

Task 1: Backend Database Design: (50 marks)

The MyKi ticketing system is very complex. Its primary role is to administer tickets, however, there are also other tasks such as monitor passenger movements, determine needs for increases in the frequency of trains and budget forecasting.

During the Week 2 Prac discussions, you have come up with a minimalist design just for capturing the basic ticketing functionality (i.e. touch on and touch off of tickets), a simple design as given below.

Ticket	Transaction	Scanner
TicketNo {PK}	TxnNo {PK}	ScannerID
MyKiMoneyBalance	TxnTime	Location
MyKiPassBalance	TxnType	ScannerType
ExpiryDate		

1..1 — ScannedBy 0..N   0..N   ScannedAt   1..1

The complete MyKi system backend is substantially more complicated than the above.

In this task, you are required to produce a complete database design, including the preparation of the database specifications document using the provided template and a **conceptual design** using Entity-Relationship Modeling.

This is an open-ended question. No specifications are provided to you. You are required to analyse the requirements of the current MyKi system, research publicly available technical documents that you can access from internet, and other effective requirement gathering techniques (refer: https://www.linkedin.com/pulse/requirements-gathering-techniques-samgra-malik).
However, it is very important that your requirements must mimic current MyKi system, not the requirements for a hypothetical ticketing system.

You are free to use Generative AI Tools (such as ChatGPT) to gather the requirements and produce the project (i.e. assignment) deliverables. However, if you choose to use such tools, you must submit a separate document listing the prompts you have used with the AI tool. Your assignment work is assessed on how accurately reflect the database requirements for the MyKi system. **Please be aware, if your prompts produced some generic ticketing system requirements, you will be disadvantaged. You won't be penalized for using Generative AI tools, as long as you properly acknowledge their use and supply a "prompts list".**

Requirements for the Entity-Relationship Model:
You are free to use any Entity-Relationship notation, however, UML notation is preferred (most of the pre-req courses at RMIT do use UML notation, so, you must be familiar with this notation. If you need any assistance with the notation, you may refer to: https://vertabelo.com/blog/uml-notation/ UML is a less ambiguous notation compared to other Entity-Relationship notations and has many similarities with the notations used in Object-Oriented Software Design.

Note: The tiny ER model on Page 3 uses the UML notation.

Requirements for the Database Design Requirements document
You must use the template provided with this assignment specifications. You must complete as many sections as possible. However, if you determined that a specific section or a sub-section is not relevant for this project, you may remove them from the document.

You must follow standard technical writing standards and make sure to cite all the external resources used in the report preparation.

The primary learning outcome of this course (this assignment, in particular) is database optimisation. As such, wherever applicable, you should provide sufficient technical depth with your explanations. They may include, but not limited to, your choice of database management system, choice of hardware and software platforms, indexing strategies, query optimization strategies and table partition strategies as applicable to your database design. A substantial portion of marks are allocated to the technical details sections.

Task 2: Implementation (20 marks)

In this task, you are required to build the database schema outlined in the design document. It must be hosted on the MS SQL Server database platform provided to you as part of the course enrolment. If your design document recommended a different database platform, you will be required to undertake required alterations to the database design to get it hosted on MS SQL Server platform.

Your implementation must comply with industry-standards currently applied across relational database system implementations. As a bare minimum, all tables should be in 3NF and referential integrity and entity integrity constraints must be enforced.

Indexing and partitioning strategies recommended in the database design document must also be implemented.

Task 3: Stored Procedures (30 marks)

Write a T-SQL stored procedure – `touchOn(),` to implement the Touch On functionality.

When a MyKi card is touched on, a series of actions are taken place on the back-end database.

It will first check if the ticket expired or not. If it is expired, the touch on fails.

If it is not expired, it checks if this new touch on is within 2 hours of the previous touch on. If that's the case, a new log record is added to the transaction log, but no new payment transactions involved.

If the ticket is not in an active touch on status, It will check if there is an active MyKi Pass in the ticket. If there is, the new touch on is marked as MyKi Pass transaction. If there is no MyKi Pass on the ticket, this new touch on is marked as a MyKi Money transaction. In that case, it checks if there is sufficient funds available. If not, the touch on fails.

The exact database operations would entirely be dependent on your database schema. Your stored procedure must comply with your database implementation.

Task 4: Demonstration

You are required to demonstrate the functionality of your database implementation (Task 2) and T-SQL procedures (Task 3). The demos are held during the week after the assignment is due. Further details of the demonstration process will be provided in due course.

3 Submission

Follow the instructions on Canvas to complete your submission for the project for each Milestone.

3.1 Milestone 1 - Early Design Demonstration

There are no canvas submission requirements for Milestone 1. You are required to demonstrate your interim (partial work) workings to the marker during Week 4.

The marker will check the following elements in your design:

- The draft version of your database design specifications document;

- The conceptual model for the database backend, in the form of an entity – relationship diagram using UML notation. You may use any tool to generate the diagram, Miro Board is one such potential tool;

3.2 Milestone 2 - Submission

Milestone 2 submission (main submission) should include answers to Task 1 and 3. It should comprise of:

1. Database Design Specifications report – in pdf format
2. Entity-Relationship model – in pdf, png or jpg format
3. Stored Procedure – in plain-text as a .sql file
4. [if you used AI tools] Prompts list.

It is your responsibility to make sure the submission is (1) complete; (2) correctly zipped; (3) files are clearly labelled, and (4) not corrupt. Please verify that your submission is correctly submitted by downloading what you have submitted to see if the files include the correct contents.

3.3 Milestone 3 - Demonstration

There are no canvas submission requirements for Milestone 3. You are required to do a demo of your complete application hosted on the school's Microsoft SQL server. These demos will be conducted during the Weeks 8 – 9). You will be provided with same sample data and the tutor will test out the functionality by using these sample data.

Tutors will use a standard testing plan for all submissions and **you will only receive Task 2 and 3 marks if you can demonstrate the functionality of your application in this milestone.**

3.4 Assessment Declaration

When you submit work electronically, you agree to the [RMIT assessment declaration](#).

3.5 Silence Period

For **Milestone 1**, there is no silence period.

For **Milestones 2 & 3**, a silence period will take effect from 5:00PM Friday 11 April 2025.

This means questions about this assignment will be not answered, whether they are asked on Canvas Discussion Forum, by email, or in person. The silence period is in place because staff members are generally unavailable over the weekend. Additionally, to be fair to all students giving presentations, we will not respond to questions about the Milestone 3 presentations during Week 8-9.

Make sure you ask your questions with plenty of time for them to be answered.

3.6 Late Submissions & Extensions

A penalty of 10% per day is applied to late submissions up to 5 business days, after which you will receive zero marks.

Short extensions may be granted by the course coordinator up to 1 business day *before* the due date in accordance with RMIT Assessment Adjustment process. However, extensions are not guaranteed and require suitable documentation. The course coordinator may refer requests to Special Considerations.

Special Consideration *may result in an equivalent assessment*, which may take the form of a timed assessment assessing the same knowledge and skills of the assignment and are generally granted on an individual basis. For more information refer to the [RMIT Special Consideration process](#).

3.7 Supported software for assessment and grading

Your assignment solution must be implemented in SQL Server and hosted on school's SQL Server database. Markers won't mark any other variations of implementations.

4 Marking Guidelines

A detailed breakdown of marks will be provided with the marking rubric.

5 Academic Integrity and Plagiarism (Standard Warning)

Academic integrity is about the honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e., directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites. If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the [RMIT Academic Integrity Website](#).

The penalty for plagiarised assignments includes zero marks for that assignment, or failure for this course. Please keep in mind that RMIT University uses plagiarism detection software.