# **INFS1200 Assignment 2 – Module 3**

**Code Due: 9 May 2025 @ 3:00 PM AEST**  
**Oral Assessment: Week 12, 19-23 May 2025**

## Weighting: 25%

|  |  |
| --- | --- |
| Full Name | Student ID (8 digits) |
|  |  |

# Overview

The purpose of this assignment is to test your ability to use and apply SQL concepts to complete tasks in a real-world scenario. Specifically, this assessment will examine your ability to use SQL Data Manipulation Language to return specific subsets of information that exist in a database and Data Definition Language to create a new relational schema. The assignment is to be completed **individually.**

# Submission

Assignment 2 is made up of two parts.

* **Part 1** will be submitted through an electronic marking tool called Gradescope, which will also be used for providing feedback.
* **Part 2** is an oral assessment that will be completed during an in-person interview with a tutor during a practical session in Week 12 (after your Gradescope submission).

Details below:

**Part 1:** Answer the questions on this task sheet and submit them through an electronic marking tool called Gradescope.

For this assignment, you will need to submit two types of files to the portal:

* **Query Files**:
  + For each question in Sections A, B and C, and D where indicated, you are required to submit a separate *.sql* or *.txt* file which contains your SQL query solution for that question (submit only one of these file types).
  + Each file should only contain the SQL query(s) and no additional text or comments.
  + Each file should be named as per the *Filename* description in the question.
  + When submitting files to the autograder, select all of your .*sql* or *.txt* files as well as your *.pdf* file.
  + It is recommended to write queries in a SQL/text file and test them in your phpMyAdmin zones before copying them into the Word document.
  + Your queries must compile using **MySQL version 8.0**. This is the same DBMS software as is used on your *phpMyAdmin* zones.
  + You may use any MySQL function that have been used in class in addition to those specified in the questions.
* **Assignment PDF**:
  + Insert your answers for all Sections A-D into the template boxes on the Microsoft Word version of this assignment task sheet where appropriate. Export this document to a PDF and also upload it to the Gradescope autograder portal.
  + Only subsections of Section D will be partially hand-marked from your PDF submission, however this is also a backup for Sections A, B and C in case of autograder failure.
  + For Sections A, B and C, include a screenshot of the output of your query for each question in the space provided. Use your zones to generate the output.
  + **Please name your file ‘Assignment\_2.pdf’.** Please do not alter the format or layout of this document and ensure the name and SID boxes are completed.

**Part 2** is an oral assessment, to verify your understanding of the code you submitted in Part 1 Sections A, B and C.

* This will be an oral critique of your submitted code. In a short meeting with a member of the teaching staff during Week 12 practical sessions, you will explain the work you have submitted in Part 1 and discuss your choices.
* All oral assessments must be given live and will be recorded by the teaching team (i.e. on Zoom) for archiving purposes.

# Marking

Assignment 2 is worth **25 course marks**, and marking is made up of two parts. First, the marks available per section of Part 1 are as follows (note that INFS1200 differs from INFS7900):

|  |  |
| --- | --- |
|  | INFS1200 |
| Section A – SQL DML (SELECT) | 14 marks |
| Section B – SQL DML (UPDATE, INSERT, DELETE) | 3 marks |
| Section C – SQL DDL | 3 marks |
| Section D – Critical thinking | 5 marks |

Given these available marks, **students must also achieve a pass (+/-) in Part 2**, the oral critique, to be eligible to pass Assignment 2. Failure in Part 2 will result in your mark being capped at 12.5 marks (50% for this assignment).

Grading and Autograder feedback:

Sections A, B, C and parts of D of this assignment will be graded via an autograder deployed on Gradescope. However, we reserve the right to revert to hand marking using the PDF submission should the need arise.

When you submit your code, the autograder will provide you with two forms of immediate feedback:

* **File existence and compilation tests**: Your code will be checked to see if it compiles correctly. If it fails one or more compilation tests, the errors returned by the autograder will help you debug. Note that code that fails to compile will receive 0 marks. No marks are given for passing the compilation tests.
* **Column Domain checking:** The autograder will check whether the domain of the projected values for each attribute in your query results matches the expected domain. Make sure to carefully read the 'explanation' section in each question of the assignment, as it provides specific details about the expected domains for attributes in the result.

**Submit your work to Gradescope early so you can use the test results to ensure your queries are compiling and you are on the right track.** You will be able to resubmit to the autograder an unlimited number of times before the deadline.

Materials provided: You will be provided with the database schema and a simple data instance that will yield a result for every correct DML (SELECT) question. Visible test results in the autograder submission portal will check that the query compiles successfully. Because the autograder uses the same DBMS as your zones, you are encouraged to use your zones to develop your assignment answers.

Late penalties: Please consult the course profile for late penalties that apply to this assessment item.

# Plagiarism

The University has strict policies regarding plagiarism. Penalties for engaging in unacceptable behaviour range from loss of grades in a course through to expulsion from UQ. You are required to read and understand the policies on academic integrity and plagiarism in the course profile (Section 6.1). If you have any questions regarding an acceptable level of collaboration with your peers, please see either the lecturer or your tutor for guidance. Remember that ignorance is not a defence!

You are permitted to use generative AI tools to help you complete this assessment task. However, if you do, please provide complete copies of your interactions with the AI tool in the space provided at the end of your submission. Please note that if you use generative AI but fail to acknowledge this by attaching your interaction to the end of the assignment, it will be considered misconduct, as you are claiming credit for work that is not your own.

# Task

For this assignment, you will be presented with the simplified schema of an authorisation application.

**BestTechLtd** have designed and developed a simplistic authorisation management system to provide secure and sufficient access control within their organisational network.

When a new employee joins an organisation using SecureAccess, they are registered and their personal details are stored in the Employee table. Administrative employees, whose additional information is stored in the AdministrativeEmployee table, can then assign appropriate roles to these employees through a role-granting process. Each role, defined in the Role table, comes with specific permissions that determine which company websites and resources the employee can access. This hierarchical permission structure ensures that employees only have access to the resources necessary for their job functions. BestTechLtd manage their authorisation by storing employee and permission data in a relational database management system (RDBMS) with the following schema:

* The **Employee** table records staff-specific information.
* **Administration** employees are a specific type of employee that have authority to grant authorisation roles to employees.
* The **Role** table records typical roles that might exist within a business and is associated with the necessary permissions that a user (Employee) of the role will need.
* The **Permission** table maintains a record of all the websites in a company’s network. A website URI and RoleID associated with the website means that the Role is granted permission to access, retrieve resources and perform operations on the website.

A diagram of a company

AI-generated content may be incorrect.ERD for the BestTechLtd schema:

Relational Schema:

**Employee** [EmployeeID, FirstName, LastName, DOB, PasswordHash, PasswordSalt]

**AdministrativeEmployee** [EmployeeID, Level, Type]

**Role** [RoleID, Name, Description]

**RoleGranting** [EmployeeID, RoleID, AdministrationID, Timestamp]

**Permission** [WebsiteURI, RoleID, GrantType, Description]

**Foreign Key References:**

AdministrativeEmployee.EmployeeID references Employee.EmployeeID

RoleGranting.EmployeeID references Employee.EmployeeID

RoleGranting.RoleID references Role.RoleID

~~RoleGranting.AdministrationID references AdministrativeEmployee.AdministrationID~~ *(correction 23/04/25)*

RoleGranting.AdministrationID references AdministrativeEmployee.EmployeeID

Permission.RoleID references Role.RoleID

For this assignment you will be required to write SQL queries to complete the tasks below.

* Answer the queries using only the information provided in the **Task** box.
* Use the **SQL Solution** box provided to record your answer code. It is recommended to write queries in the SQL/text file and test them in your phpMyAdmin zones before pasting them here.
* Use the **Output Screenshot** box to record the output of your query (generated in your zones before submission).

|  |  |
| --- | --- |
| Example Query | |
| Task | Retrieve all information about all employees. |
| SQL Solution | SELECT \*  FROM Employee; |
| Output Screenshot |  |

# Section A – SQL DML (SELECT)

|  |  |
| --- | --- |
| Question A1 | |
| Task | Return the distinct first name and last name of all employees, in descending alphabetical order of their last name. |
| Filename | A1.sql or A1.txt |
| Columns returned | This query must return two columns; the first name and the last name of employees, in that order. |
| SQL Solution |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A2 | |
| Task | For each website URI, find the number of roles that can access it. |
| Columns returned | This query must return two columns; the website URI, and the number of Roles which access a website URI, in that order. |
| Filename | A2.sql or A2.txt |
| SQL Solution |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A3 | |
| Task | For every role in the database, find the number of websites they have access to. |
| Columns returned | This query must return two columns; the RoleID, and the number of permissions they have, in that order. |
| Filename | A3.sql or A3.txt |
| SQL Solution |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A4 | |
| Task | Return the distinct role ID, name and descriptions of all roles which permit access to commercial websites (where the URI ends with “.com”). |
| Columns returned | This query must return three columns; the role ID, the role’s name, and the role’s Description, in that order. |
| Filename | A4.sql or A4.txt |
| SQL Solution |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A5 | |
| Task | Return all information about all employees that have not been granted a role from an Administrator with type ‘ProductEngineer’.  **Restriction:** You must use a sub-query to answer this question. |
| Columns returned | This query must return all fields from the employee table. |
| Filename | A5.sql or A5.txt |
|  |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A6 | |
| Task | Return the Date of Birth (DOB), first and last name of the youngest employee(s) |
| Columns returned | This query must return three columns; the date of birth, the first name, and the last name of the employee(s), in that order. |
| Filename | A6.sql or A6.txt |
| SQL Solution |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A7 | |
| Task | Find and return the first name and last name of all employees that have access to at least all the website URIs that Elena Popov has access to.  You may assume only one Elena Popov exists in the database. |
| Columns returned | This query must return two columns; the first name and the last name of the employees, in that order. |
| Filename | A7.sql or A7.txt |
|  |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A8 | |
| Task | Return the EmployeeID of the Administrator(s) that have granted the most permissions of any other Administrator.  **Restriction:** You must use one or more views in your answer |
| Columns returned | This query must return two columns; the employeeID(s) and the number of permissions granted for the employee(s). |
| Filename | A8.sql or A8.txt |
| Note | If one Administrator granted the same role twice, then all the permissions of that role must be counted twice. |
| SQL Solution |  |
| Output Screenshot |  |

|  |  |
| --- | --- |
| Question A9 | |
| Task | The co-founders of bestTechLtd can be identified as either:   1. Administrative Employees with a ‘LegacyEngineer’ type; 2. Employees who have been granted the “DBA” role; or 3. Permissions to access **only one** website URI, which is <https://besttechltd/financialperformance.tech>   Return the EmployeeID, first and last name of all co-founders.  **Restriction:** You must use at least one **set operation** in your answer. |
| Columns returned | This query must return three columns; the employeeID, the first name and the last name of employee(s), in that order. |
| Filename | A9.sql or A9.txt |
| Hint | You may want to use one or more views in your answer. |
| SQL Solution |  |
| Output Screenshot |  |

# Section B – SQL DML (UPDATE, DELETE, INSERT)

|  |  |
| --- | --- |
| Question B1 | |
| Task | Revoke the role that was granted to John Stevens by Mei Chen at any time during the day of 22nd July, 2024. You can assume such a role exists, and only one John Stevens and Mei Chen exist in the database. |
| Filename | B1.sql or B1.txt |
| SQL Solution |  |

|  |  |
| --- | --- |
| Question B2 | |
| Task | A new employee has been onboarded to BestTechLtd. The employee (ID E0024) is named James Moran, Born 21 November 2001, with a password hash e7cf3ef8d8aac2c1c93963e7a58b7b62ade24d0d0ba2c8ae0f7fb6c8b0aa0332 and passwordSalt D;%yL9TS:5PalS/d  James Moran is granted all the roles that employee Sofia Gonzalez has, and is granted them by the admin E0001.  Insert new records with James’ details and roles as given above. You can assume only one Sofia Gonzales exists in the database. |
| Filename | B2.sql or B2.txt |
| SQL Solution |  |

# Section C – SQL DDL

|  |  |
| --- | --- |
| Question C1 | |
| Task | Before BestTechLtd created their authorisation system, employees were assigned only a single role in their company, in addition to a unique access token which granted access for that role. To maintain compatibility with their older technologies, a new relation must be made to track legacy employees and their tokens.  Create a new relation **LegacyEmployee**, which is a *specialised* type of employee which maintains additional characteristics of an **Employee** and must be stored in a separate relation. The details are as follows:   * ***LegacyEmployeeID*: A unique reference to the employeeID stored in the authorisation database.** * ***GrantAccessToken***: an [SHA-256 token](https://securiti.ai/glossary/secure-hash-algorithm-sha-256-bit/) to grant access to old websites. * ***RoleID***: A reference to the original role of the legacy employee. This reference cannot be null.   Write a SQL DDL query to implement the relation **LegacyEmployee. Make sure to follow the naming syntax exactly and add all required constraints.** |
| Filename | C1.sql or C1.txt |
| SQL Solution |  |

|  |  |
| --- | --- |
| Question C2 | |
| Task | Add a constraint to ensure that the highest administration level is 10. Name the constraint *AdministrationMax*. (You can assume that no existing employees have an admin level higher than 10.) |
| Explanation | The following resources may be useful when answering this question:  [Check constraints](https://www.w3schools.com/sql/sql_check.asp) |
| Filename | C2.sql or C2.txt |
| SQL Solution |  |

# Section D – Critical Thinking

In this section, you will receive theoretical situations related to the UoD mentioned in the task description. Your task is to offer strategies to tackle the situation and write SQL queries to execute the approaches.

|  |  |
| --- | --- |
| Question D.1 | |
| Task | The database architects at BestTechLtd wish to change the authorisation relational schema and have proposed a new design to improve security management. In this new design, password credentials will be migrated from employee data into a separate relation called **Credential**. A new ERD is proposed to illustrate these changes:  A diagram of a company  AI-generated content may be incorrect.  The database architects have provided the following instructions for implementing both structural changes and the migration of existing data:  **Required Operations:**   * Create a new table called **Credential** with appropriate fields as outlined in the ERD. These fields can be in any order in the table; * Migrate all existing credential data from the Employee table to the new Credential table, where:   + Each *CredentialID* should be formed by prepending 'C' to the corresponding EmployeeID (e.g., EmployeeID E001 has *CredentialID* CE001)   + A reference to the employee must be added to the **Credential** relation and named *EmployeeID*   + The *PasswordSalt* must be transferred from the original passwordSalt in the **Employee** table   + The *PasswordHash* must be transferred from the original passwordHash in the **Employee** table   + The *Timestamp* is the date and time when these new records are immediately migrated into the database   + Ensure the resulting relational schema matches the proposed ERD. *Hint:* Pay particular attention to the structure and constraints given in the new ERD.     **Task:** Write sequence of SQL statements that perform both the structural changes to the relations and the data migration. Your solution must ensure that no credential data is lost during this transition.  Paste your SQL solution into D1.sql and the first text box below.  In the second text box, briefly describe (in 200 words or less) how the changes you have made maintain relational integrity. |
| Filename (SQL only) | D1.sql or D1.txt |
| SQL Solution |  |
| Explanation |  |

|  |  |
| --- | --- |
| Question D.2 | |
| Task | BestTechLtd's current authorization system tracks roles granted to employees through their RoleGranting relation but lacks the ability to record when roles are revoked.  The company now wishes to enhance the existing schema to retain a complete history of both role assignments and revocations. This means when a role is granted to an employee, it will continue to be recorded, but additionally, when a role is taken away from an employee, a record of this removal should also be preserved in the database.  In 200 words or less, describe how you would extend the current implementation to meet these new business requirements while maintaining relational integrity constraints. Your solution should build upon rather than replace the existing functionality.  SQL scripts are not required but you may include them if they help illustrate your proposed changes. |
| Filename | **No SQL submission for this question** |
| Solution |  |