Assignment Guidance and Front Sheet

This front sheet for assignments is designed to contain the brief, the submission instructions, and the actual student submission for any WMG assignment. As a result the sheet is completed by several people over time, and is therefore split up into sections explaining who completes what information and when. Yellow highlighted text indicates examples or further explanation of what is requested, and the highlight and instructions should be removed as you populate 'your' section.

This sheet is only to be used for components of assessment worth more than 3 CATS (e.g. for a 15 credit module, weighted more than 20%; or for a 10 credit module, weighted more than 30%).

To be completed by the student(s) prior to final submission:

Your actual submission should be written at the end of this cover sheet file, or attached with the cover sheet at the front if drafted in a separate file, program or application.

Student ID or IDs for group work	**e.g. 1234567**

To be completed (highlighted parts only) by the programme administration after approval and prior to issuing of the assessment; to be consulted by the student(s) so that you know how and when to submit:

Date set	20/02/2023
Submission date (excluding extensions)	15/05/23
Submission guidance	Submission details are provided in the assignment sheet (A single ZIP file to be submitted to Tabula)
Marks return date (excluding extensions)	12/06/23
Late submission policy	If work is submitted late, penalties will be applied at the rate of **5 marks per University working day** after the due date, up to a **maximum of 10 working days** late. After this period the mark for the work will be reduced to 0 (which is the maximum penalty). "Late" means **after the submission deadline time as well as the date** – work submitted after the given time even on the same day is counted as 1 day late. For **Postgraduate** students only, who started their **current course before 1 August 2019**, the daily penalty is **3 marks** rather than 5.

Resubmission policy	If you fail this assignment or module, please be aware that the University allows students to remedy such failure (within certain limits). Decisions to authorise such resubmissions are made by Exam Boards. Normally these will be issued at specific times of the year, depending on your programme of study. More information can be found from your programme office if you are concerned.

To be completed by the module owner/tutor prior to approval and issuing of the assessment; to be consulted by the student(s) so that you understand the assignment brief, its context within the module, and any specific criteria and advice from the tutor:

Module title & code	WM243 Information Management
Module owner	Anita Khadka
Module tutor	Same as above
Assessment type	Report / database/test scripts
Weighting of mark	50%

Assessment brief
Details of the assessment brief are provided in each case study and a separate note section.

Word count	The word count for the PDF report must not exceed 2000 words (Excluding cover sheet, tables, diagrams and references).
Module learning outcomes (numbered)	• Critically identify entities, attributes and their relationships • Interact with database(s) of information through suitable programming queries • Critically evaluate the cyber consequences that flow from management of information in a given scenario. • Critically identify and evaluate database security and take useful measures
Learning outcomes assessed in this assessment (numbered)	As above
Marking guidelines	Marking guidelines are provided in the Assessment Brief.
Academic guidance resources	Academic guidance will be provided throughout the module.

Students are expected to work through the cases below.

Note that the purpose of this assignment is not data modelling and normalisation but **Database security**, therefore Tables are fictitious and may not be following the normal forms.

CASE 1

Consider a national supermarket chain as an example. The chain has several stores all over the nation and employees work in various levels and stores. You are assigned the job to design their database such that necessary privileges will be granted to employees based on their job title and the store they are based in. The database has several tables in the database holding information about its product, inventory, customers, and employees. Particularly, let's take the ***EMPLOYEE*** table that contains the attributes as shown in below:

Employee
emp_id
name
address
date_of_birth
sort_code
bank_account_number
salary

Table 1: Employee details

Details of every employee in the company are entered in this table and contain sensitive data. Many employees in the company need to access data in this table however, the access to data depends on the role of the subject. For simplicity, only the following roles need access to the table:

- Store Manager
- HR Manager
- Admin
- Finance Manager
- Area Manager

These employees can have access to following information in the Employee table. How will you enforce this?

job_title	name	address	date_of_birth	sort_code	bank_account_number	salary
Store_manager	y	y	n	n	n	n
HR_manager	y	y	y	n	n	y
Admin	y	n	n	n	n	n
Finance_manager	y	y	y	y	y	y
Area_manager	y	y	n	n	n	n

Table 2: Privileges (**y** for grant access and **n** for do not grant access)

In addition, there should be a restriction based on which store the managers are based. For example, the store manager, Human Resource (HR) manager, and Finance manager can only access information of employees in their own store. The area manager can access information within their area. To add these restrictions there is another table **EmployeeJobDetails** shown below.

EmployeeJobDetails
emp_id
area_id
store_id
department_id
job_role

Table 3: Employee link

The implementation and associated writeup of carries 25% mark.

CASE 2

SQL injection is often used to exploit a database, typically accessing sensitive data, modifying a database or executing administrative operations on a database. In preparation for an SQL injection exploitation, attackers often use SQL injection to discover information about the database. The supermarket chain has an internal website where the employees can login using their credentials. The login page uses the Login table to authenticate the user. This is described in Table 4. Investigate how SQL injection can be used to discover information about the database. In particular, do the following:

Login
emp_id
username
password

Table 4: Login Table

- Login into the website without using the correct username and password.
- Show how SQL injection can be used to insert a new employee into the Employee table.
- What are the reasons why the SQL injection attack to insert a new user might fail?
- If you know an employee's email address, show how SQL injection can be used to change the email address to your preferred email address.
- Assuming you are successful at changing the employee's email address, how can you then get access to the employee's password?
- List all the methods and precautions to secure the system against SQL injection attacks

The app for the website is provided with the assignment. The installation is the same as the SQL Injection lab which took place on 23rd Jan 2023.

Case 2 carries 25% of the marks for attacks + 15% for defensive strategies

CASE 3

Explain the importance of auditing in information security with examples (e.g., cases when auditing may be used to detect malicious activity in the database)

Imagine a scenario where a manager is recorded as being in their office late one night (past 10 pm). Subsequently at the time when they were in their office the audit trail records several unsuccessful attempts to access database objects using a password of a member of clerical staff to objects to which the manager had no rights of access.

- What are the threats in the above scenario? Explain the nature of the threats. If you feel you need more information, explain what you need to know.
- Write a script to set up a basic auditing procedure for your database tables. This script represents auditing process which will allow a DBA to track all changes (e.g., insert, update, delete) made to a specific table. This process will also track the user who performed those actions on the table.

Case 3 carries 25% mark.

NOTES

The word limit of the report is 2000 words excluding tables, diagrams, appendix, and cover page. The answers should provide **Test scripts (SQL)** with explanations in detail. You should demonstrate to write a clear and concisely explained report. You are also required to demonstrate the thorough testing of the correct operations of your database security solution and submit the test scripts and database dump.

Overall presentation and argument of thought/ideas, applicability to real life scenarios etc: 10% mark.