**BCDE214 Database Administration Assignment 2– Portfolio**

**Appendix A – View, Trigger, Stored Procedure, Backup, Security, and Indexing**

**Due Date:** Friday 28 October; 11:59 p.m.

**Total marks Available:** 50

**Design Policies** and **Write scripts** to perform the following tasks

**Stored Procedures**

1. People will make appointments to be vaccinated through some sort of interface. Multiple vaccinations are required. The interface may let them make multiple bookings at once, and for these bookings to be at different places. It is not your job to create the interface, but you are expected to create stored procedures that the interface may use. You have inherited some stored procedures, but you have to create some procedures which will bulk load vaccinators and appointments.

**Backup**

The database will be high usage and should always be available.

1. Produce a document recommending a backup policy. This should at least cover:

* Full Backups
* Partial/Differential Backups
* Transaction Log Backups
* Replication

Your policy should describe when the backups should occur, and where and how long the backups should be stored.

1. Write well documented script(s) implementing the backup policy. You may wish to produce these by reverse engineering, but you will need to document them to show full understanding.

**Users and Security**

The database will not be directly accessed by the public, but only through stored procedures. The developer (who does not really care about security) has written some stored procedures. Be aware of the potential for SQL injections.

1. Produce a document recommending a security setting. This should describe what Users and Roles the database needs, and the rights these Users and Roles will require. You will probably enhance this policy when Views are introduced to your system.
2. Write well documented script(s) implementing your security policy. You may wish to produce these by reverse engineering, but you will need to document them to show full understanding.

**Indexing**

The database is currently only set up for data entry. There have been no views created for users to access.

1. Write documented scripts (Views) that creates indexes to:

* Assist with searching for who has vaccinated people
* Who has been vaccinated
* Where vaccinations have occurred

1. Comment on what these indexes may do to database performance and propose an alternative solution to indexes.

**Views**

Write scripts to create the following views:

* A view that shows each vaccination place, with the total number of vaccinations that could happen each day, and the number of vaccinations actually happening or booked each day.
* A view showing all people that have missed a booking, and do not have a later booking. We will want to contact these people, so include names and contact details.
* A view that can be used to show people broken down by age in decades as at the date of their first vaccination. Only show people that have been vaccinated.
* A view showing the number of sessions and the number of vaccines all vaccinators have done. Include those vaccinators that have yet to vaccinate anybody.

**Trigger**

Write scripts to create the following trigger:

* When someone cancels an appointment, the table for Old Appointments is populated, with the reason being “Changed appointment.”

**Marking Schedule**

Stored Procedures 8 marks

createVaccinator 2 marks

bulkLoadVaccinators 2 marks

createAppointment 2 marks

bulkLoadAppointments 2 marks

Backup 5 marks

Policy 2.5 marks

Implementation 2.5 marks

Users and Security 8 marks

Users 2 marks

Roles 2 marks

Rights 2 marks

Implementation 2 marks

Indexing 4 marks

Indexes 2 marks

Implementation 2 marks

View 12 marks

**Four** views, each worth **three** marks.

0: No attempt, or very little attempt made

1: A syntactically correct attempt, that does not completely fulfill the requirement

2: A view that fulfills the requirement

3: A view that fulfills the requirement, **and** test script(s) provided that shows that the view is working for data that changes affecting the view, and data changes that does not affect the view.

Trigger 3 marks

**One** trigger case provided, worth **three** marks.

0: No attempt, or very little attempt made

1: A syntactically correct attempt, that does not completely fulfill the requirement

3: A trigger that fulfills the requirement with test