# SQL Developer Test

Create a simple database schema, using SQLServer, which:

1. Contains the following information for Names:
   * Name details catering for Individuals and Organisations
   * ACN and ABN for organisations
   * Multiple Email Addresses
   * Multiple Phone Numbers
   * Multiple Address details (street, postal, registered office etc)
   * Marketing categories assigned to names
   * The only mandatory data is the person/organisation name
2. An “Application User” of the system will be assigned user role(s). Create a simple structure allowing “Application Users” to be stored and assigned one or more of the following roles:
   * Administrator
   * Client
   * Manager Responsible
   * Acting Manager
   * Secretary
3. The names data will be secured access to “application users” according to their assigned role(s). Create a data structure against the name so that the name can be secured according to any of the above roles. Note that a number of access roles can be assigned to a single name. An “Administrator” should have access to every name.
4. Things to consider when designing the schema:
   * Primary Keys
   * Foreign Keys/data
   * Cater for future enhancements/maintainability
5. Write a script to populate the data with 1million names along with random associated name data.
6. Write a script to create a number of different ‘users’ and assigned roles. Note that a number of different roles can be assigned to a single user.
7. Write a well-structured, performant, stored procedure to return name information. It should:
   * **Not** use dynamic SQL
   * **Not** use cursors
   * Display emails against a name as a comma separated string
   * Display phone numbers against a name as comma separated string
   * Display the suburb /city and State of the address(es) for the name
   * Order the results by last name
   * Use a parameter indicating the ‘application user’ executing the stored procedure. The data returned by the stored procedure needs to be filtered by the role(s) of the user. Write and use a sql function to apply this “security”.
   * Use an optional parameter indicating the sort sequence is ascending or descending
   * Use an optional parameter to filter the information by individual or organisation. The default is to display all names
   * Use a single optional parameter to filter the results. This value can be comma separated to enable multiple values for the filter which should be across any combination of:
     1. marketing categories assigned to the names
     2. a substring in the Name
     3. suburb in an address against the name
     4. state in an address against the name
   * Use optional parameters to return the results as pages of rows. The page size should be configurable
   * Return the total number of rows in the entire query resultset regardless of whether paging is applied or not
8. Deliver the scripts so that they can be run to automatically create the schema, populate the data and execute the stored procedure
9. Describe an approach and test plan to verify correctness of the stored procedure and schema functionality. Ideally scripts should be delivered to execute these tests