**Task description**

**Database design**

Design relational database, which will describe following domain model:

Blog posts with localized content. The set of blog post attributes is following: author, date, rating, localized content (which inclu1des title and post body).

User profiles. Includes the set of such attributes as first name, last name, e-mail, and phone.

User groups. This entity is used for grouping users into groups, which can has assigned roles, which should be applied for each user of this group. In addition, groups support inheritance, i.e. can has parent group and set of the child ones. So that any child group inherits permissions of all its parents.

User credentials. This entity is necessary for storing user credentials information such as login name, password hash.

User roles. Includes role name and set of role permissions.

Role permissions. This entity should be related to user role and has following set of attributes: permissions type and the flag, which indicated whether this permission is allowed. Let`s consider that there can be following set of permission types: 1 – read blog posts, 2 - manage blog posts, 3 – manage users.

\* Note: all auxiliary not domain tables should be designed as well, if necessary.

**Programmability**

Complete each of tasks below in order to learn the most typical scenarios of advanced database querying and demonstrate this skill on the practice.

* 1 - Create query, which returns localized content of blog post by blog post ID. There should be set of values for each language registered in system even if there no corresponded content records found.
* 2 - Create query, which returns localized content of blog posts by paging info: skip, take parameters.
* 3 - Create stored procedure, which calculates password hash and adds credentials for user.
* 4 - Create stored procedure, which validates credentials for user by input password.
* 5 - Create trigger, which will delete all child user group in case when the parent one is deleted (cascade rules will not be acceptable due to possible circular references)
* 6 - reate function, which returns set of parent group IDs for target group (use recursive queries).
* 7 - Create function, which returns set of child group IDs for target group (use recursive queries).
* 8 - Create query, which returns all assigned groups for user (including parent groups, which may not be assigned to user directly)
* 9 - Create view, which shows full permissions matrix for target user. Consider that:
* All permissions mapped with roles
* All user permissions should be taken from assigned roles
* User can has assigned roles directly
* User can be assigned to groups which in its turn can has assigned roles
* User groups inherits roles of parent ones
* 10 - Create query, which returns all users which has target permission
* 11 - Create stored procedure, which will process bulk import of users. Use table UDT ‘users’ as input parameter. New users should be added, existing – updated (use Merge statement)