

Build Control

Documentation technique

Informations générales

Base de données

La solution *BuildControl* s'appuie sur une base de données SQL hébergée sur Azure, la technologie cloud de Microsoft. De ce fait, son hébergement est entièrement modulable.

La version du SGBD SQL utilisée est et sera toujours la dernière grâce au service Azure (actuellement SQL Server 2019 15.0).

La base de données utilisée nommée « BuildControl » peut donc être accédée via un gestionnaire de base de données tel quel S.S.M.S. (SQL Server Management Studio) ou autre outils tier.

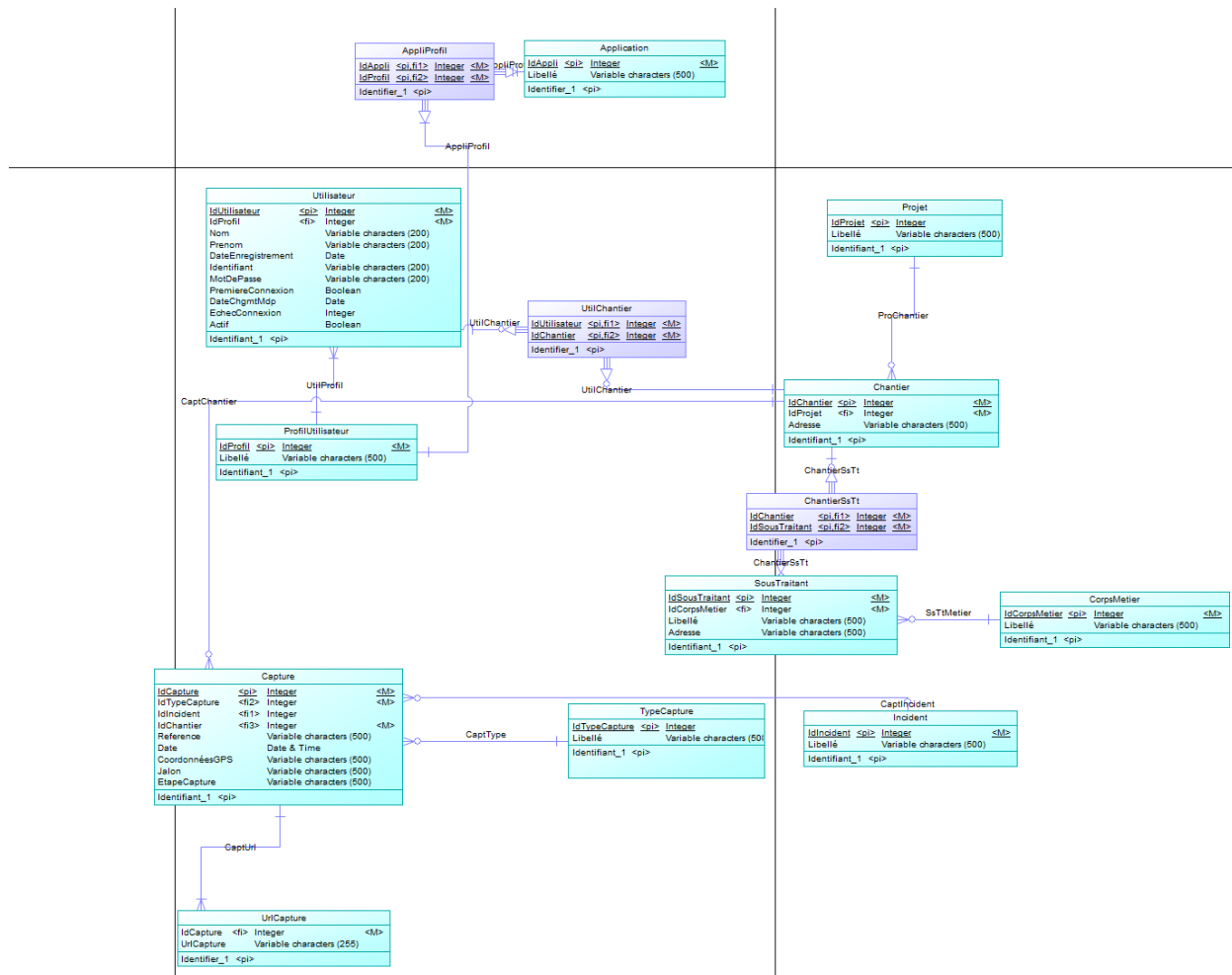
Tablespaces

L'espace de stockage des tables utilisé est celui par défaut : « dbo. ». Il sera possible de subdiviser ce dernier en plusieurs sous-espaces pour correspondre à la gestion des accès des différents profils de l'application.

Objets de la base de données

Tables et index, avec indication de schéma

Le modèle logique de données utilisé est le suivant :



On y retrouve différentes tables :

- Application (identifiant : *IdApplication*) : cette table référence les différents types d'applications présents dans la solution. Cela permettra notamment d'effectuer plus facilement un filtre sur les utilisateurs selon leur profil.
- AppliProfil (identifiant : *IdApplication, IdProfil*) : table de jointure entre une application et un profil. Un profil peut utiliser de 0 à n applications.
- Profil : (identifiant : *IdProfil*) : cette table référence tous les profils d'utilisateur. Elle a pour but de lier un utilisateur à un profil et de donc de pouvoir gérer ses droits et accès via le profil.
- Utilisateur (identifiant : *IdUtilisateur*) : table de stockage des utilisateurs ainsi que de leurs informations. Des champs sont également présents afin d'offrir une gestion du compte et des identifiants. Un utilisateur est lié à un type de profil.
- UtilChantier (identifiant : *IdUtilisateur, IdChantier*) : table de jointure entre un utilisateur et le chantier auquel il est affecté. Un utilisateur peut être affecté à 0 ou n chantiers.
- Chantier (identifiant : *IdChantier*) : table contenant les informations des chantiers présents dans la plateforme. Chaque chantier est lié à un projet.
- Projet (identifiant : *IdProjet*) : table listant l'ensemble des projets ajoutés à la solution.
- ChantierSSTT (identifiant : *IdChantier, IdSousTraitant*) : table reliant un chantier à ses sous-traitants. Un sous-traitant peut intervenir dans 1 à n chantiers et un chantier peut comprendre 0 à n sous-traitants.
- SousTraitant (identifiant : *IdSousTraitant*) : table contenant les informations des sous-traitants intervenant au sein d'un chantier.
- CorpsMetierSSTT (identifiant : *IdSoustraitant, IdCorpsMetier*) : table reliant un sous-traitant à son corps de métier. Un sous-traitant possède 1 à n corps de métier et un corps de métier peut être affecté à 1 à n sous-traitants.
- CorpsMetier (identifiant : *IdCorpsMetier*) : table référençant l'ensemble des corps métier utilisés au sein de l'application.
- Capture (identifiant : *IdCapture*) : table contenant les informations des captures prises dans un chantier. Une capture est reliée à un type de capture, à 0 ou 1 incident ainsi qu'à un chantier. Elle possède également une ou plusieurs URLs redirigeant vers l'image de la capture.
- URLCapture (identifiant : *IdURLCapture*) : table contenant l'url d'une capture.
- TypeCapture (identifiant : *Id*) : table définissant le type de capture (exemple : drone ou matterport). Cette table permettra le filtre des captures selon leur type.
- Incident (identifiant : *IdIncident*) : table référençant les incidents pouvant intervenir dans le cadre d'un chantier. Il sera possible de filtrer les captures selon le type d'incident.

Vues

Les vues créées au sein de la base de données BuildControl correspondent aux demandes du client concernant l'utilisation de la base de données. Ces vues sont créées à partir de requêtes statiques afin de simplifier leur utilisation.

- View_Demande : retourne la référence et la date des captures validant un filtre daté.

- View_InformationDemande : retourne l'ensemble des informations partant d'une capture. Le chantier lié, les sous-traitants, leur corps de métier ainsi que les urls de la capture. La capture est sélectionnée à partir de sa référence (ici #00234).
- View_CountProjet_ByDate : retourne le nombre de projet ayant été effectué à partir d'une date (ici à partir du 1^{er} décembre 2019).
- View_SousTraitant_Projet : retourne les sous-traitants intervenus dans le cadre d'un projet ainsi que leur corps de métier.

Procédures, fonctions, triggers

Les procédures et triggers créés au sein de la base de données correspondent aux demandes du client concernant l'automatisation.

- getProjects : cette procédure prend un paramètre une date de début et de fin et retourne les projets présents dans cet intervalle ainsi que les sous-traitants.
- getIncidents : cette procédure retourne les informations liées à un incident étant intervenu dans le cadre d'un chantier. Elle prend en paramètre la référence de capture afin de déterminer un potentiel incident.
- checkIncidents : ce trigger permet de vérifier l'unicité d'une capture lors de sa création. Elle s'appuie sur les informations d'une capture voulant être créée et vérifie si cette capture existe en base de données. Dans le cas d'une duplication, une erreur est retournée par le trigger.

Accès aux données

L'objectif final est de ne pas stocker dans la base de données les informations relatives à la gestion de l'authentification mais de déléguer cette gestion à un outil tier tel que KeyCloak. Un tel outil permettrait la gestion des utilisateurs, de leur identifiant ainsi que les droits pouvant être appliqués à des groupes.

Seules les informations relatives à l'utilisateur seraient stockées dans la base de données qui aurait pour seul rôle la gestion de cas métier.

Un rôle global à tous les employés présents sur le terrain donne l'accès aux requêtes de lecture.