

# Increase



- Code source du projet :  
<https://github.com/ApollineMallet/increase>
- Projet développé sous la méthode **SCRUM** !

## Présentation de la situation professionnelle

[e4\\_antoine\\_clavel\\_increase.pdf](#)

## Contexte

Une société de développement (Web, mobile, applications...) souhaite disposer en interne d'un outil permettant :

- de gérer la répartition de ses équipes sur les projets ⇒ affectations
- d'apprécier les tâches effectuées et le degré d'avancement sur chaque projet à chaque instant

L'application sera accessible en interne (Intranet destiné aux équipes de développement) mais aussi en externe, pour permettre au client commanditaire d'un projet de consulter l'avancement de celui-ci.

## Développement

### Outils utilisés

- Eclipse PHP
- Sublime Text 3
- GitHub
- MySQL Workbench pour créer la BDD et générer le script de création
- PhpMyAdmin

### Langages utilisés :

- PHP
- Ajax
- SQL
- HTML/CSS

### Equipe TripleA :

- Apolline Mallet
- Antoine Burnel
- Antoine Clavel

## Règles de gestion

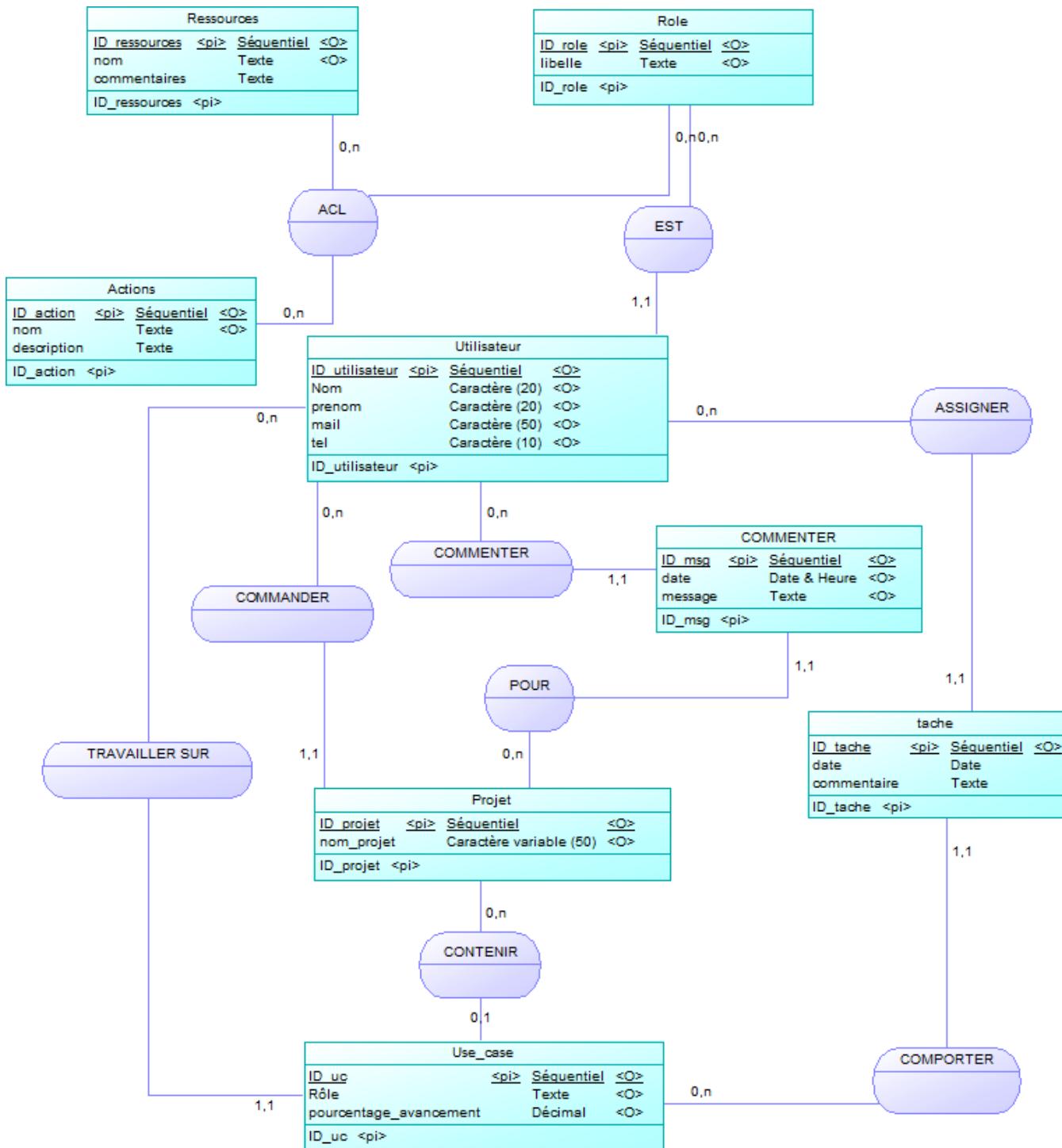
- Un utilisateur a un rôle (personnel ou client de l'entreprise).
- Un client peut commander un à plusieurs projets mais il ne peut pas travailler sur un projet.
- Un projet ne peut avoir qu'un seul client.
- Un développeur peut travailler sur plusieurs cas d'utilisation et sur plusieurs tâches.
- Un seul développeur peut travailler sur un cas d'utilisation et sur une tâche.
- Si un développeur est assigné à une tâche, il est assigné au cas d'utilisation de la tâche.
- Un projet peut contenir plusieurs cas d'utilisation.
- Un cas d'utilisation ne contient qu'un seul projet.
- Un cas d'utilisation peut comporter plusieurs tâches.
- Un utilisateur peut commenter des projets autant de fois qu'il le souhaite.
- Un rôle possède des ressources et des actions.

## Dictionnaire des données

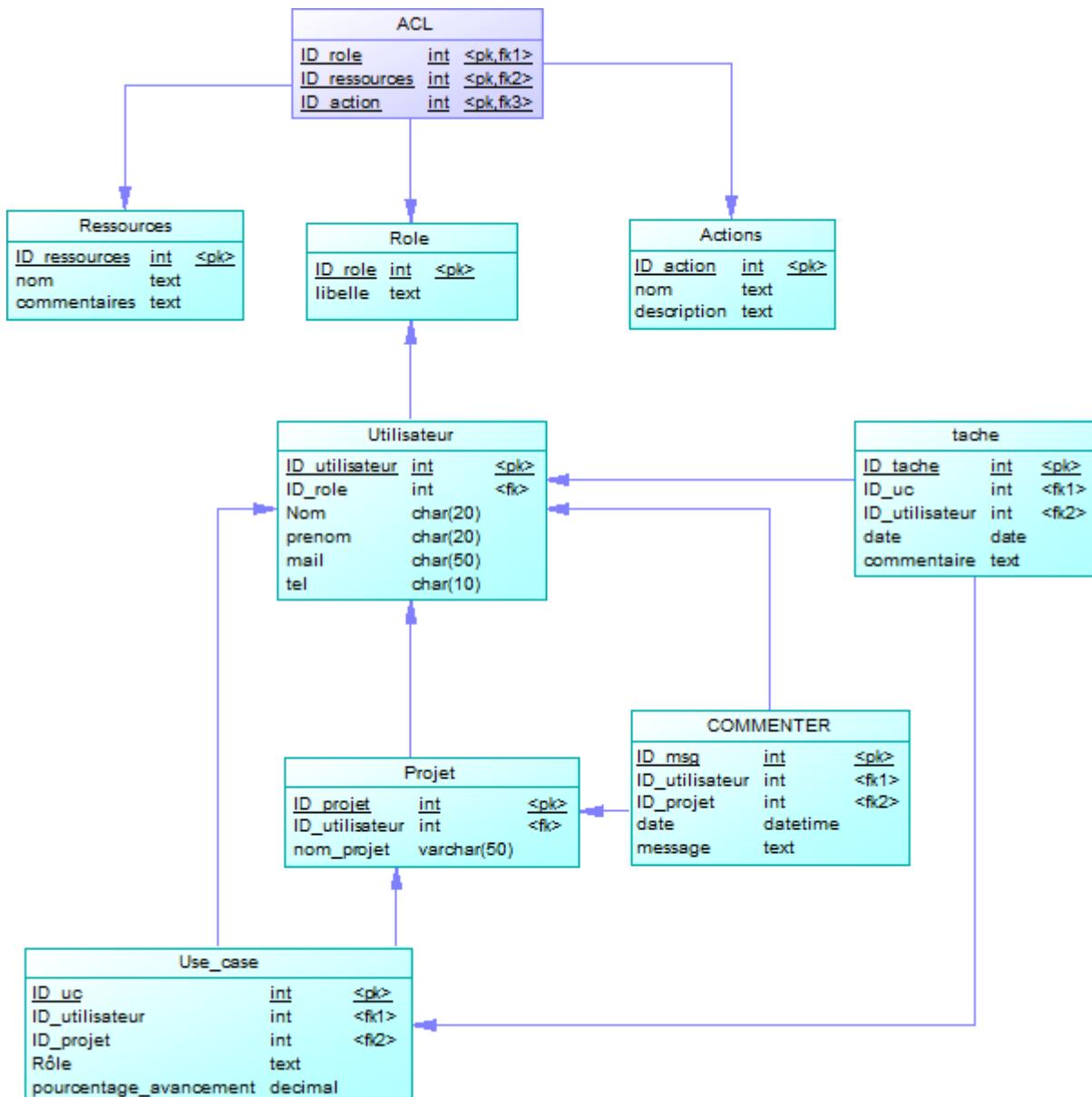
Nom	Table	Type de données	Longueur	Clef primaire	Clef étrangère
libelle	Role	text			
commentaire	tache	text			
commentaires	Ressources	text			
date	tache	date			
date	COMMENTER	datetime			
description	Actions	text			
ID_action	Actions	int		X	
ID_action	ACL	int		X	X
ID_msg	COMMENTER	int		X	
ID_projet	COMMENTER	int			X
ID_projet	Use_case	int			X
ID_projet	Projet	int		X	
ID_ressources	ACL	int		X	X
ID_ressources	Ressources	int		X	
ID_role	Role	int		X	
ID_role	Utilisateur	int			X
ID_role	ACL	int		X	X
ID_tache	tache	int		X	
ID_uc	tache	int			X
ID_uc	Use_case	int		X	
ID_utilisateur	tache	int			X
ID_utilisateur	Use_case	int			X
ID_utilisateur	COMMENTER	int			X
ID_utilisateur	Projet	int			X
ID_utilisateur	Utilisateur	int		X	
mail	Utilisateur	char(50)	50		
message	COMMENTER	text			
nom	Ressources	text			
nom	Actions	text			
Nom	Utilisateur	char(20)	20		
nom_projet	Projet	varchar(50)	50		
pourcentage_avancement	Use_case	decimal			
prenom	Utilisateur	char(20)	20		

Nom	Table	Type de données	Longueur	Clef primaire	Clef étrangère
Rôle	Use_case	text			
tel	Utilisateur	char(10)		10	

## Modèle Conceptuel de Données (MCD)



## Modèle Physique de Données (MPD)



## Modèle Logique de Données (MLD)

- **PROJET** (`numProjet`, `nom`)
- **UC** (`numUC`, `nomUC`, `avancement` / `partRealisation`, `#numProjet`, `#idUtilisateur`)
- **TACHE** (`idTache`, `libelleTache`, `dateTache`, `#numUC`)
- **UTILISATEURS** (`idUtilisateur`, `nom`, `prenom`)
- **ROLE** (`idRole`, `libelleRole`, `#idUtilisateur`)
- **ACL** (`#idRole`, `#idAction`, `#idRessources`)
- **ACTION** (`idAction`, `libelleAction`)
- **RESSOURCES** (`idRessource`, `libelleRessource`)
- **MESSAGE** (`numMessages`, `objet`, `contenu`, `date`, `#idUtilisateurs`, `#`)

## Script de création

```
/*=====
/* Nom de SGBD : MySQL 5.0
/* Date de création : 16/03/2015 17:17:53
/*=====

drop table if exists acl;
drop table if exists actions;
drop table if exists commenter;
drop table if exists projet;
drop table if exists ressources;
drop table if exists role;
drop table if exists tache;
drop table if exists useCase;
drop table if exists utilisateur;

/*=====
/* Table : acl
/*=====

create table acl (
    idRole          int not null,
    idRessources    int not null,
    idAction        int not null,
    primary key (idRole, idRessources, idAction)
);

/*=====
/* Table : actions
/*=====

create table actions (
    idAction        int not null auto_increment,
    nom             text not null,
    description     text,
    primary key (idAction)
);

/*=====
/* Table : commenter
/*=====

create table commenter (
    idMsg           int not null auto_increment,
    idUtilisateur   int not null,
```

```
    idProjet          int not null,
    date             datetime not null,
    message          text not null,
    primary key (idMsg)

);

/*=====
/* Table : projet
/*=====
create table projet (

    idProjet          int not null auto_increment,
    idUtilisateur     int not null,
    nomProjet         varchar(50) not null,
    primary key (idProjet)

);

/*=====
/* Table : ressources
/*=====
create table ressources (

    idRessources      int not null auto_increment,
    nom               text not null,
    commentaires      text,
    primary key (idRessources)

);

/*=====
/* Table : role
/*=====
create table role (

    idRole            int not null auto_increment,
    libelle           text not null,
    primary key (idRole)

);

/*=====
/* Table : tache
/*=====
create table tache (

    idTache           int not null auto_increment,
    idUc              int not null,
    idUtilisateur     int not null,
    date              date,
    commentaire       text,
    primary key (idTache)

);
```

```
=====
/* Table : useCase */
=====
create table useCase (

    idUc          int not null auto_increment,
    idUtilisateur int not null,
    idProjet      int,
    role          text not null,
    pourcentageAvancement decimal not null,
    primary key (idUc)

);

=====
/* Table : utilisateur */
=====
create table utilisateur (

    idUtilisateur   int not null auto_increment,
    idRole          int not null,
    nom             char(20) not null,
    prenom          char(20) not null,
    mail            char(50) not null,
    tel             char(10) not null,
    primary key (idUtilisateur)

);

alter table acl add constraint FK_ACL foreign key (idAction)
    references actions (idAction) on delete restrict on update restrict;

alter table acl add constraint FK_ACL foreign key (idRessources)
    references ressources (idRessources) on delete restrict on update restrict;

alter table acl add constraint FK_ACL foreign key (idRole)
    references role (idRole) on delete restrict on update restrict;

alter table commenter add constraint FK_COMMENTER foreign key (idUtilisateur)
    references utilisateur (idUtilisateur) on delete restrict on update restrict;

alter table commenter add constraint FK_POUR foreign key (idProjet)
    references projet (idProjet) on delete restrict on update restrict;

alter table projet add constraint FK_COMMANDER foreign key (idUtilisateur)
```

```
references utilisateur (idUtilisateur) on delete restrict on update restrict;
```

alter table tache add constraint FK\_ASSIGNEE foreign key (idUtilisateur)

```
references utilisateur (idUtilisateur) on delete restrict on update restrict;
```

alter table tache add constraint FK\_COMPORTER foreign key (idUc)

```
references useCase (idUc) on delete restrict on update restrict;
```

alter table useCase add constraint FK\_CONTENIR foreign key (idProjet)

```
references projet (idProjet) on delete restrict on update restrict;
```

alter table useCase add constraint FK\_TRAVAILLER\_SUR foreign key (idUtilisateur)

```
references utilisateur (idUtilisateur) on delete restrict on update restrict;
```

alter table utilisateur add constraint FK\_EST foreign key (idRole)

```
references role (idRole) on delete restrict on update restrict;
```

From:  
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**



Permanent link:  
[http://slamwiki2.kobject.net/etudiants/2015/antoine.clavel/ppe\\_increase](http://slamwiki2.kobject.net/etudiants/2015/antoine.clavel/ppe_increase)

Last update: **2019/08/31 14:21**