

Département Informatique

BD50

Rapport de projet

Création d’un web agenda

Groupe 14



**Enseignant** : M. Christian FISCHER

FALL Cheikh Alioune

GUEYE Arona

TETOU Mayaba

**Semestre P18**

15/04/2018

# Historique du document

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Auteur** | **Modifications du document** |
| **Étape 1** | 22/03/2018 | Tout le groupe | Création du MEA |
| 05/04/2018 | Tout le groupe | Premières corrections apportées au MEA |
| 08/04/2018 | Tout le groupe | Structuration du dossier de projet et rédaction du rapport |
| 12/04/2018 | Tout le groupe | Dernières corrections apportées au MEA |
| 13/04/2018 | Tout le groupe | Création du dictionnaire de données |
|  |  |  |  |
| **Étape 2** | 03/05/2018 | Tout le groupe | Génération du MLR et optimisation |
| 11/05/2018 | Tout le groupe | Premières corrections apportées à l’optimisation |
| 13/05/2018 | Tout le groupe | Finalisation de l’optimisation et rédaction du rapport |
| 14/05/2018 | Tout le groupe | Création des triggers de gestion de la redondance et génération du script de création de la base de données |
|  |  |  |  |

Table des matières

[Historique du document 1](#_Toc514192412)

[Introduction 3](#_Toc514192413)

[I. Description du projet 4](#_Toc514192414)

[1.1 Extensions possibles 4](#_Toc514192415)

[II. Modélisation conceptuelle 5](#_Toc514192416)

[2.1 Dictionnaire de données 5](#_Toc514192417)

[2.2 Modèle entité-association (MEA) 6](#_Toc514192418)

[2.3 Explication des cardinalités et associations 7](#_Toc514192419)

[III. Modélisation logique 9](#_Toc514192420)

[3.1 Modèle logique des données relationnelles 9](#_Toc514192421)

[3.2 Justification du niveau de normalisation 10](#_Toc514192422)

[3.3 Modèle logique des données relationnelles optimisé 11](#_Toc514192423)

[3.3.1 Fusion de tables 11](#_Toc514192424)

[3.3.2 Redondance 11](#_Toc514192425)

# Introduction

BD50 est une UV qui porte la conception des bases de données relationnelles. Elle vise à fournir aux étudiants des compétences sur les méthodes et outils de conception pour la mise en œuvre de base données notamment grâce au logiciel de modélisation Win’Design et au système de gestion de base de données ORACLE.

Dans cette optique, il nous est demandé de concevoir et déployer une application de création d’un agenda en ligne (Web agenda). Notre choix s'est porté sur ce sujet car, pour nous, il présente un double intérêt. En effet, en dehors de nous permettre de mettre en pratique les connaissances acquises dans cette UV, il nous permet aussi de savoir comment fonctionnent certains outils populaires de gestion d’événements comme Doodle ou Google Calendar. Ce projet va également nous permettre de développer des compétences qui nous seront peut-être utiles dans notre future carrière car il est envisageable qu’une entreprise veuille développer son propre système de gestion de l’agenda de son personnel.

Dans notre étude, nous ne nous sommes pas focalisés sur le fonctionnement d'une entreprise en particulier et n’avons pas non plus calqué notre application sur celles qui existent actuellement. Nous avons voulu développer un modèle générique, qui puisse être utilisé par la plupart des personnes désirant gérer facilement leur agenda.

# Description du projet

Le projet de création d’un web Agenda a pour objectif majeur de permettre aux utilisateurs de l’application de pouvoir gérer avec grande facilité leur agenda. Cela inclut la gestion des événements auxquels ils doivent participer (et leur périodicité), leurs indisponibilités.

Les principales fonctionnalités que l’application devra offrir aux utilisateurs sont :

* Création d’un agenda personnel avec la gestion des indisponibilités, des événements et leur périodicité
* Partage de l’agenda c’est-à-dire la possibilité d’inviter des personnes (qu’elles aient un compte ou non dans l’application) à des événements en leur proposant une série de dates

Pour qu’un utilisateur puisse avoir accès à toutes ces fonctionnalités, il faut qu’il dispose d’un compte client qui devra être validé par un administrateur. Cela requiert l’existence d’une partie administration pour la validation des comptes mais aussi la définition des événements communs à tous.

## Extensions possibles

Pour ce projet, nous avons eu plusieurs idées d’extensions. En effet, il est possible de prendre en compte la gestion des tâches d’un utilisateur et permettre leur partage avec d’autres utilisateurs ou groupe d’utilisateurs qu’il aura préalablement définis. Il serait intéressant également de donner à tout utilisateur la possibilité d’attribuer des tâches à d’autres utilisateurs et suivre leur avancement.

# Modélisation conceptuelle

## Dictionnaire de données

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom Conceptuel** | **Nom Logique (ou Alias)** | **Type (E,Ca,Co)** | **Nature + Longueur** | **Type Windesign** | **Identifiant** |
| Année agenda | ANNEE\_AGENDA | E | N4 | N |  |
| Code postal occurrence | CP\_OCCURENCE | E | AN8 | AV |  |
| Commune occurrence | COMMUNE\_OCCURENCE | E | AN50 | AV |  |
| Complément adresse occurrence | COMP\_ADR\_OCCURENCE | E | AN60 | AV |  |
| Date création compte | DATE\_CREA\_COMPTE | E | DH12 | DH |  |
| Date début indisponibilité | DATE\_DEB\_INDISPO | E | DH12 | DH |  |
| Date début événement | DATE\_DEB\_EVENEMENT | E | DH12 | DH |  |
| Date début indisponibilité | DATE\_DEB\_INDISPO | E | DH12 | DH |  |
| Date début périodicité | DATE\_DEB\_PERIODICITE | E | D8 | D |  |
| Date fin événement | DATE\_FIN\_EVENEMENT | E | DH12 | DH |  |
| Date fin indisponibilité | DATE\_FIN\_INDISPO | E | DH12 | DH |  |
| Date fin indisponibilité | DATE\_FIN\_INDISPO | E | DH12 | DH |  |
| Date fin périodicité | DATE\_FIN\_PERIODICITE | E | D8 | D |  |
| Date naissance participant | DATE\_NAISS\_PARTICIPANT | E | D8 | D |  |
| Date occurrence | DATE\_OCCURENCE | E | D8 | D |  |
| Etat événement | ETAT\_EVENEMENT | E | BOOL1 | BOOL |  |
| Extension occurrence | EXT\_OCCURENCE | E | AN3 | AV |  |
| Heure début occurrence | HEURE\_DEB\_OCCURENCE | E | H4 | H |  |
| Heure fin occurrence | HEURE\_FIN\_OCCURENCE | E | H4 | H |  |
| Libellé événement | LIBELLE\_EVENEMENT | E | AN100 | AV |  |
| Lieu événement | LIEU\_EVENEMENT | Co | AN215 | AV |  |
| Login utilisateur | LOGIN | E | AN20 | AV |  |
| Mail participant | MAIL\_PARTICIPANT | E | AN100 | AV |  |
| Mot de passe utilisateur | MDP | E | AN40 | AV |  |
| Nom agenda | NOM\_AGENDA | E | AN50 | AV |  |
| Nom groupe | NOM\_GROUPE | E | AN50 | AV |  |
| Nom participant | NOM\_PARTICIPANT | E | AN50 | AV |  |
| Nom rue occurrence | NOM\_RUE\_OCCURENCE | E | AN50 | AV |  |
| Nombre répétitions | NB\_REPETITION | E | N4 | N |  |
| Numéro agenda | NUM\_AGENDA | E | N4 | N | OUI |
| Numéro disponibilité | NUM\_DISPO | E | N4 | N | OUI |
| Numéro événement | NUM\_EVENEMENT | E | N4 | N | OUI |
| Numéro groupe | NUM\_GROUPE | E | N4 | N | OUI |
| Numéro indisponibilité | NUM\_INDISPO | E | N4 | N | OUI |
| Numéro occurrence | NUM\_OCCURENCE | E | N4 | N | OUI |
| Numéro participant | NUM\_PARTICIPANT | E | N4 | N | OUI |
| Numéro périodicité | NUM\_PERIODICITE | E | N4 | N | OUI |
| Numéro rue occurrence | NUM\_RUE\_OCCURENCE | E | N4 | N |  |
| Pays occurrence | PAYS\_OCCURENCE | E | AN40 | AV |  |
| Prénom participant | PRENOM\_PARTICIPANT | E | AN50 | AV |  |
| Privé | PRIVE | E | BOOL1 | BOOL |  |
| Raison indisponibilité | RAISON\_INDISPO | E | AN50 | AV |  |
| Répétitif | REPETITIF | E | BOOL1 | BOOL |  |
| Sexe participant | SEXE\_PARTICIPANT | E | AN1 | AV |  |
| Statut compte | STATUT\_COMPTE | E | BOOL1 | BOOL |  |
| Téléphone participant | TEL\_PARTICIPANT | E | AN15 | AV |  |
| Type événement | TYPE\_EVENEMENT | E | AN50 | AV |  |
| Type participant | TYPE\_PARTICIPANT | E | AN1 | AV |  |
| Type périodicité | TYPE\_PERIODICITE | E | AN40 | AV |  |
| Type utilisateur | TYPE\_UTILISATEUR | E | AN6 | AV |  |

## Modèle entité-association (MEA)

Le MEA que nous avons créé pour ce projet est le suivant :

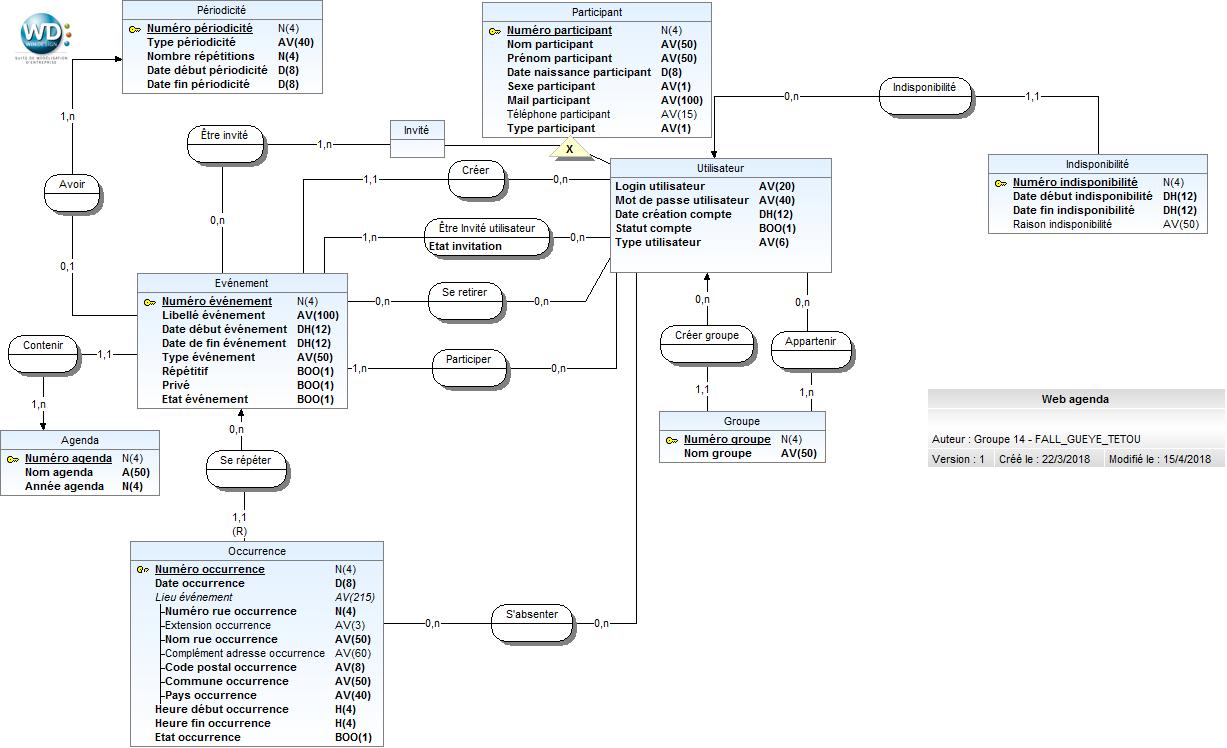


Figure : Modèle Entité-Association

Lorsqu’un participant à un événement est un invité (non utilisateur de l’application), il faut qu’il ait un numéro de participant. Il faudrait également connaître son nom, prénom, mail et éventuellement son numéro de téléphone. Pour factoriser tous ces attributs qui se retrouvent également dans l’entité utilisateur, nous avons créé un super-type Participant et fait hériter les entités invité et utilisateur de ce dernier.

## Explication des cardinalités et associations

**Avoir (1,n) – (0,1) :** un événement peut ne pas avoir de périodicité (un événement ponctuel par exemple) mais a au plus une périodicité. Par contre, plusieurs événements différents peuvent avoir la même périodicité.

**Contenir (1,n) – (1,1) :** un agenda peut contenir un ou plusieurs événements. En revanche, un événement appartient à un et un seul agenda.

**Être invité (1,n) – (0,n) :** une personne qui n’a pas de compte dans l’application peut être invitée à aucun ou plusieurs événement(s). Un événement particulier peut avoir comme participant plusieurs personnes n’étant pas utilisateurs de l’application (invité).

**Créer (0,n) – (1,1) :** un événement est créé par un seul utilisateur. Ce même utilisateur peut n’avoir auparavant créé aucun événement et peut dans le futur en créer plusieurs autres.

**Être invité utilisateur** **(1,n) – (0,n) :** à un événement est invité au moins un utilisateur. Cela permet d’éviter de créer un événement sans participant. Plusieurs utilisateurs peuvent être invités à un événement. Un utilisateur peut n’être invité à aucun événement ou être invité à plusieurs. Cette association est porteuse de la donnée **Etat invitation**. Cette propriété permet de savoir si l’invitation a été acceptée, refusée ou si elle en attente.

**Se retirer (0,n) – (0,n) :** un utilisateur peut se retirer d’aucun ou de plusieurs événements (après avoir accepté l’invitation). A un événement, plusieurs utilisateurs peuvent se retirer en même temps.

**Participer (1,n) – (0,n) :** à un événement participe un ou plusieurs utilisateurs. Un utilisateur participe à aucun ou plusieurs événement(s).

**Se répéter (0,n) – (1,1) :** Un événement peut avoir aucune ou plusieurs occurrence(s). Une même occurrence appartient à un seul événement.

**S’absenter (0,n) – (0,n) :** Un utilisateur peut s’absenter à plusieurs occurrences ou à aucune d’entre elles. A aucune occurrence peut s’absenter aucun ou plusieurs utilisateurs.

**Créer groupe (0,n) – (1,1) :** Un utilisateur peut créer plusieurs groupes. Un groupe n’est créé que par un seul utilisateur.

**Appartenir (1,n) – (0,n) :** Un groupe contient au moins un utilisateur (celui qui l’a créé). Il peut en contenir plusieurs. Un utilisateur peut appartenir à plusieurs groupes en même temps. Il peut également n’appartenir à aucun groupe.

**Indisponibilité (0,n) – (1,1) : Un** utilisateur peut être tout le temps disponible (aucune indisponibilité) comme il peut avoir plusieurs indisponibilités. Une indisponibilité appartient à un et un seul utilisateur.

**Remarques :**

* L’entité événement est reliée à l’entité occurrence grâce à un identifiant relatif. En effet, il est impossible d’avoir une occurrence d’événement s’il n’y a pas d’événement. La dépendance entre événement et occurrence est une dépendance très forte.
* Il n’y pas d’associations « participer », « inviter » entre utilisateur et occurrence car nous considérons que lorsqu’un utilisateur est invité à un événement, il est automatiquement invité à toutes les occurrences de cet événement. Il peut cependant s’absenter à une ou plusieurs occurrence(s) particulière(s) de l’événement ; d’où l’association « s’absenter » entre utilisateur et occurrence.
* Lors de la création d’un événement, le lieu n’est pas requis. C’est lors de la création des occurrences pour cet événement qu’il va falloir déterminer le lieu de chaque occurrence. Nous avons choisi cette modélisation car nous jugeons que toutes les occurrences ne se déroulent pas forcément au même endroit. Il serait donc inutile de mettre un lieu à la création de l’événement s’il faudra après le modifier pour des occurrences données.

# Modélisation logique

## Modèle logique des données relationnelles

Le MLR généré pour Oracle 10G/11G, à partir du modèle entité-association, est le suivant :

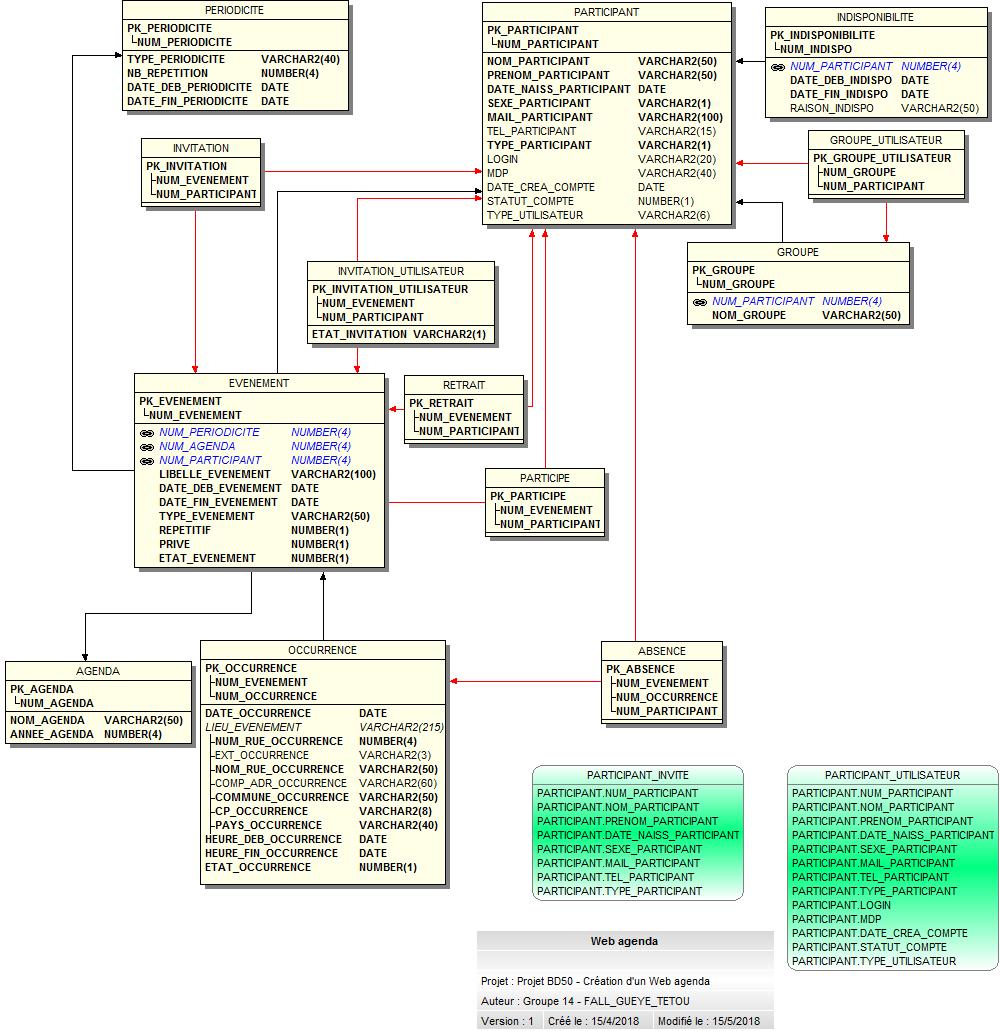


Figure : Modèle logique des données relationnelles

## Justification du niveau de normalisation

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1FN | 2FN | 3FN |
| ABSENCE | X | X | X |
| AGENDA | X | X | X |
| EVENEMENT | X | X | X |
| GROUPE | X | X | X |
| GROUPE\_UTILISATEUR | X | X | X |
| INDISPONIBILITE | X | X | X |
| INVITATION | X | X | X |
| INVITATION\_UTILISATEUR | X | X | X |
| PARTICIPANT | X | X | X |
| PARTICIPE | X | X | X |
| PERIODICITE | X | X | X |
| RETRAIT | X | X | X |
| OCCURRENCE | X | X | X |

**1FN**

Toutes les tables sont en 1FN parce qu’aucun de leurs attributs n’est multivalué.

**2FN**

Toutes les tables sont en 2FN car chacun de leurs attributs non clés dépend de la totalité de leurs clés primaires.

**3FN**

Toutes les tables sont 3FN parce qu’aucun de leurs attributs non clés ne dépend d’un autre.

## Modèle logique des données relationnelles optimisé

Afin d’obtenir une base de données présentant des performances les plus optimales possibles en terme de temps de réponse, nous avons procédé à l’optimisation du modèle logique des données relationnelles ci-dessus. Ainsi, nous avons mis en œuvre plusieurs techniques d’optimisation que nous exposerons dans les parties suivantes.

### Fusion de tables

En analysant le MLR non optimisé, il apparaît l’existence de tables ayant la même clé primaire : INVITATION\_UTILISATEUR, RETRAIT et PARTICIPE. Plutôt que d’interroger trois tables différentes pour avoir des informations (participation, retrait, invitations reçues) sur les participants concernant certains événements, il serait plus aisé de n’en interroger qu’une seule. Ainsi, nous avons fusionné ces tables ayant la même clé primaire.

Des propriétés discriminantes sont ajoutées à la table obtenue, PARTICIPATION, afin de distinguer les participations des annulations (retraits).

### Redondance

Pour une gestion plus facile des événements, nous avons dupliqué des informations se trouvant dans les tables AGENDA et PERIODICITE dans la table EVENEMENT. Nous avons fait de même pour OCCURRENCE (duplication des informations essentielles de la table EVENEMENT dans OCCURRENCE). Ceci introduit une redondance d’informations car les tables d’où proviennent ces informations dupliquées ne sont pas supprimées. Ainsi, nous pouvons garantir l’intégrité de la base de données grâce à des triggers de gestion de la redondance.

Cette redondance simplifie les requêtes en évitant les jointures qui auraient été nécessaires pour avoir certaines informations sur les événements (nom de l’agenda auquel appartient un événement par exemple).

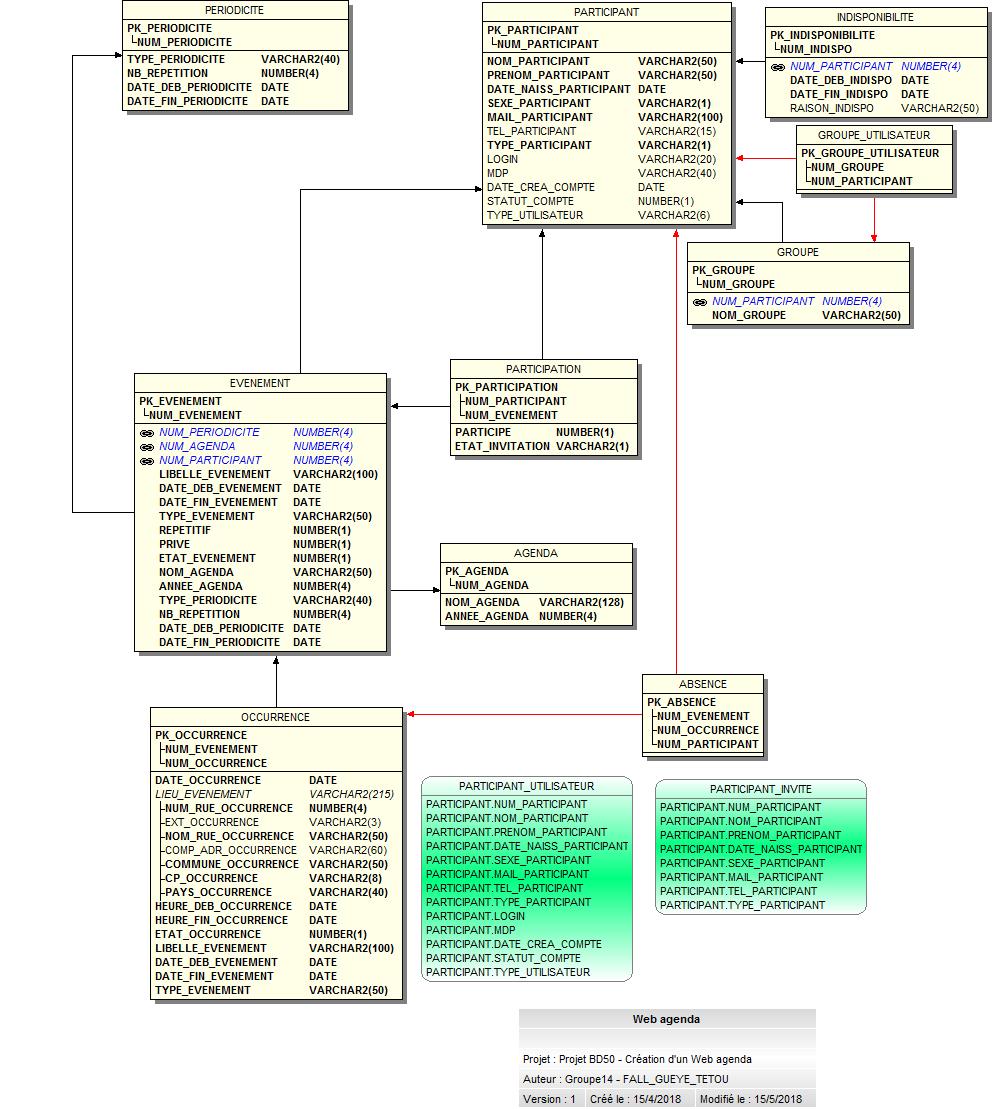


Figure : Modèle logique des données relationnelles optimisé

**Remarque :**

Au niveau physique, nous avons défini certaines tables en index (Index Organized Table). Cela permet d’optimiser le modèle physique puisque les données sont incluses dans les index et il n’y a donc pas besoin de faire une deuxième lecture dans la table associée ; la table étant l’index.

Pour tirer pleinement profit des IOT, seules les tables qui ne sont accédées que par la clé primaire et qui ne font pas fréquemment l’objet de mise à jour ont été organisées en index.

### 