

Modélisation de données UCAC-ICAM-UdeS

Automne 2021

Enseignants

Lorince S. Tawamba Kouakam (lorince.tawamba@ucac-icam.com)

Christina Khnaisser (christina.khnaisser@usherbrooke.ca)

Luc Lavoie (luc.lavoie@usherbrooke.ca)

Modèle des effectifs cliniques autonomes

MECA

(énoncé de problème, servant de point de départ à l'analyse, version 010a)

Simplification d'un système réel.

Plusieurs éléments critiques ne sont pas pris en compte.

Cette description ne doit pas être la base d'un système utilisé en pratique.

La République pataphysique s'est dotée d'un réseau de santé publique dans lequel interviennent des personnes appartenant à différents corps de métiers cliniques. Chaque année, les responsables de la santé publique doivent planifier les affectations des effectifs cliniques autonomes (ECA) pour les mois, voire les années, à venir pour les établissements faisant partie du réseau. Les ECA (des médecins, des sages-femmes, des infirmières spécialisées, etc.) sont affectés à différentes unités organisationnelles (UO) pour effectuer un type d'activité durant une période donnée. Les UO sont organisées hiérarchiquement (région, établissement, département, service, etc.). Cette structure devra être reflétée par le modèle afin de pouvoir faire des analyses en fonction du niveau.

Les gestionnaires du réseau doivent prévoir les activités de leurs ECA pour les différentes UO relevant de leur administration. Une prévision d'activité définit la quantité d'un type d'activité qui doit être réalisée par un ECA au sein d'une UO pendant une période donnée. Ces prévisions sont produites pour des périodes définies selon un calendrier prescrit (généralement à une date suffisamment antérieure à la période visée pour permettre une prise en compte effective). De plus, les prévisions peuvent être modifiées en raison d'erreurs de saisie ou pour s'adapter à certaines circonstances (ex. médecins malades, augmentation du nombre de patients); le temps entre le moment de la prévision et sa portée est très variable (parfois la date de la prévision coïncide avec, ou même suit de quelques jours la date du début de la portée de la prévision). La correction d'une prévision a pour effet de placer une activité hors prévision relativement aux prévisions en vigueur au moment de la consignment, mais conforme aux prévisions courantes.

Ces prévisions sont ensuite utilisées pour construire une vue complète des trajectoires des ECA dans le temps au niveau de plusieurs établissements. Le principal défi consiste à maintenir à tout moment un nombre adéquat d'ECA pour chaque activité dans chaque UO.

Une analyse préliminaire dans le cadre de l'étude faisabilité a conduit à l'élaboration du schéma présenté ci-après. On remarque l'omission de la définition des types et des contraintes. Cette lacune doit être corrigée lors de la programmation en SQL d'une première version du schéma de la base de données.

Description des entités principales et dictionnaire de données

La description des entités principales de leurs associations, de leur contraintes et du dictionnaire de données permettant d'en définir les attributs est directement disponible dans le l'esquisse du schéma relationnel fourni en annexe.

Précisions sur la notation utilisée pour le schéma relationnel

Le schéma est présenté sous la forme d'un diagramme relationnel. Les rectangles représentent les relations (tables en SQL) et leurs attributs (colonnes en SQL). Les flèches représentent les contraintes référentielles, orientées de la relation référente (penne de la flèche →) vers la relation référée (pointe de la flèche →). Les icônes [👉] signalent les attributs faisant partie de la clé (candidate) primaire. D'autres clés candidates peuvent exister.

Il existe d'autres notations relationnelles qui permettent d'exprimer l'existence de plusieurs clés au sein d'une relation.

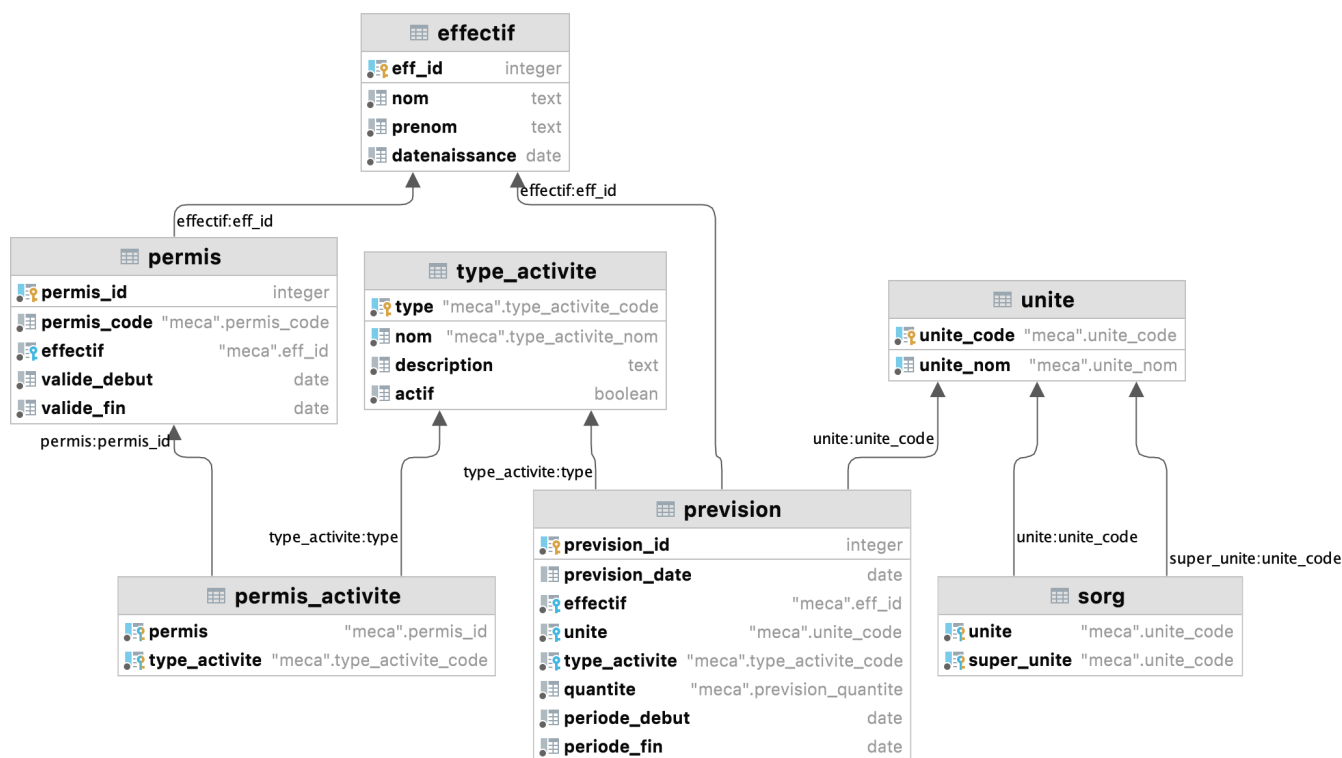


Figure 1 — Diagramme relationnel du modèle préliminaire.

Annexe A – Esquisse du modèle logique de données

```

create domain Unite_Code
    VARCHAR(8)
    constraint unite_code_dom check (value similar to '[A-Z]{5}[0-9]{3}');
comment on domain Unite_Code is $$
Code unique d'une unité organisationnelle.
$$;
--
create domain Unite_Nom
    VARCHAR(240);
comment on domain Unite_Nom is $$
Nom d'une unité organisationnelle.
$$;
--
create table Unite
(
    unite_code Unite_Code not null,
    unite_nom Unite_Nom not null,
    constraint unite_cc0 primary key (unite_code),
    constraint unite_cc1 unique (unite_nom)
);
comment on table Unite is $$
Une unité organisationnelle est identifiée par "id" et nommée par "nom".
$$;
--
-- ===== Structure organisationnelle
--
  
```

```

create table S0rg
(
    unite Unite_Code not null,
    super_unite Unite_Code not null,
    constraint sorg_cc0 primary key (unite, super_unite),
    constraint sorg_cr0 foreign key (unite) references unite(unite_code),
    constraint sorg_cr1 foreign key (super_unite) references unite(unite_code)
);
comment on table S0rg is $$
Une unité "unite" dépend administrativement de l'unité "super_unite".
La structure doit strictement respecter une topologie hiérarchique
(un graphe connexe sans cycle).
$$;
--
-- ===== Effectif clinique autonome
--
create domain Eff_ID
    INTEGER
    constraint eff_id_dom check (value > 0);
comment on domain Unite_nom is $$
Identifiant artificiel d'un effectif médical.
$$;
--
create table Effectif
(
    eff_id INTEGER not null generated always as identity (minvalue 1),
    nom text not null,
    prenom text not null,
    dateNaissance Date not null,
    constraint effectif_cc0 primary key (eff_id)
);
comment on table Effectif is $$
Un effectif est identifié par "id" et nommé par "nom" et "prenom" est né le
"dateNaissance".
$$;
--
-- ===== Type d'activité
--
create domain Type_activite_Code
    VARCHAR(7)
    constraint type_activite_code_dom check (value similar to '[a-zA-Z][a-zA-Z0-9]{2,7}');
comment on domain Type_activite_Code is $$
Code unique d'un type d'activité.
$$;
--
create domain Type_activite_Nom
    VARCHAR(40);
comment on domain Type_activite_Nom is $$
Nom de référence d'un type d'activité.
$$;
--
create table Type_activite
(
    type Type_activite_Code not null,
    nom Type_activite_Nom not null,
    description TEXT not null,

```

```

    actif BOOLEAN not null,
    constraint Type_activite_cc0 primary key (type),
    constraint Type_activite_cc1 unique (nom)
);
comment on table Type_activite is $$
Un type d'activités.
$$;
--
-- ===== Permis
--
create domain Permis_ID
    INTEGER;
comment on domain Permis_ID is $$
Identifiant artificiel unique d'un permis d'exercice.
$$;
--
create domain Permis_CODE
    VARCHAR(7)
    constraint permis_code_dom check (value similar to '[a-zA-Z]{4}[0-9]{3}');
comment on domain Permis_CODE is $$
Code unique d'un permis d'exercice.
$$;
--
create table Permis
(
    permis_id INTEGER not null generated always as identity (minvalue 1),
    permis_code Permis_CODE not null,
    effectif Eff_ID not null,
    valide_debut DATE not null,
    valide_fin DATE not null,
    constraint permis_cc0 primary key (permis_id),
    constraint permis_cr0 foreign key (effectif) references effectif(eff_id)
);
comment on table Permis is $$
Le permis d'exercice d'un effectif pour une durée de validité spécifique.
Une effectif peut détenir plus d'un permis, mais leurs périodes de validité ne peuvent se
chevaucher.
$$;
--
create table Permis_activite
(
    permis Permis_ID not null,
    type_activite Type_activite_CODE not null,
    constraint permis_activite_c00 primary key (permis, type_activite),
    constraint permis_activite_cr0 foreign key (permis) references Permis(permis_id),
    constraint permis_activite_cr1 foreign key (type_activite) references Type_activite(type)
);
comment on table Permis_activite is $$
Le permis permet d'exercer pour un ou plusieurs types d'activités.
$$;
--
-- ===== Prévision d'activité
--
create domain Prevision_ID
    INTEGER
    constraint Prevision_ID_dom check (value > 0);

```

```

comment on domain Prevision_ID is $$
Identifiant unique artificiel d'une prévision.
$$;
--
create domain Prevision_quantite
    NUMERIC(12,2)
    constraint Prevision_quantite_dom check (value > 0);
comment on domain Prevision_quantite is $$
La quantité de travail d'une prévision est exprimée en heures-personnes
(qui est plutôt une mesure de l'effort, mais les us et coutumes du réseau sont ainsi).
$$;
--
create table Prevision
(
    prevision_id INTEGER not null generated always as identity (minvalue 1),
    prevision_date DATE,
    effectif Eff_ID not null,
    unite Unite_Code not null,
    type_activite Type_activite_CODE not null,
    quantite Prevision_quantite not null,
    periode_debut DATE not null,
    periode_fin DATE not null,
    constraint prevision_cc0 primary key (prevision_id),
    constraint prevision_cr0 foreign key (effectif) references Effectif(eff_id),
    constraint prevision_cr1 foreign key (unite) references Unite(unite_code),
    constraint prevision_cr2 foreign key (type_activite) references Type_activite(type)
);
comment on table Prevision is $$
Une prévision d'activité estime la quantité de travail d'un type d'activités à être
réalisées par un effectif médical au sein d'une unité organisationnelle pendant une période
donnée.
Une prévision ne peut être saisie que si l'effectif possède un permis valide permettant
d'accomplir le type d'activité durant toute la période prévue.
$$;

```