

Devoir n°1 (correction) : Merise 2 + SQL

Thème : extensions Merise 2 et requêtes SQL

Le sujet est constitué de deux exercices. Les deux exercices sont indépendants. Vous indiquerez les réponses aux questions sur votre ou vos copies.

Exercice 1 : modèle conceptuel 9 points

Contexte :

L'agence orléanaise de la société de services informatiques PRASORIASTE dispose de deux plateaux de développement constitués d'ingénieurs étude et développement. Le premier plateau, orienté « télécommunication », s'occupe en outre de projets d'envergures à destination d'opérateurs de télécommunication et d'organismes financiers. Le second plateau s'occupe des autres projets, souvent d'envergure également. Les ingénieurs travaillant sur ces deux plateaux, très largement issus de cursus Bac+5 de l'Université d'Orléans et de l'Ecole Polytech' de Tours, sont regroupés sous forme d'équipes projet.

Afin d'améliorer le suivi de ses projets, l'agence orléanaise de PRASORIASTE souhaite, à titre expérimental, mettre en place un outil de gestion de projet développé sur mesure. Pour ce faire, elle a confié le développement de la 1^{ère} version du logiciel à des stagiaires dans le cadre de leur stage de fin d'études. En tant que stagiaire, vous avez à charge la modélisation des données du logiciel à concevoir. Ce modèle de données devra prendre en compte les attentes de PRASORIASTE en matière de gestion de projet.

Fonctionnement attendu :

Pour l'agence orléanaise, l'objectif général du logiciel est double :

- permettre à PRASORIASTE d'établir des plannings hebdomadaire (par semaine) : par salarié, par projet, etc. Ces plannings ont vocation à lui permettre d'adapter ses effectifs à la charge de travail ;
- permettre à PRASORIASTE de suivre le coût de ses projets, la rentabilité de ces derniers et l'efficacité de ses ingénieurs.

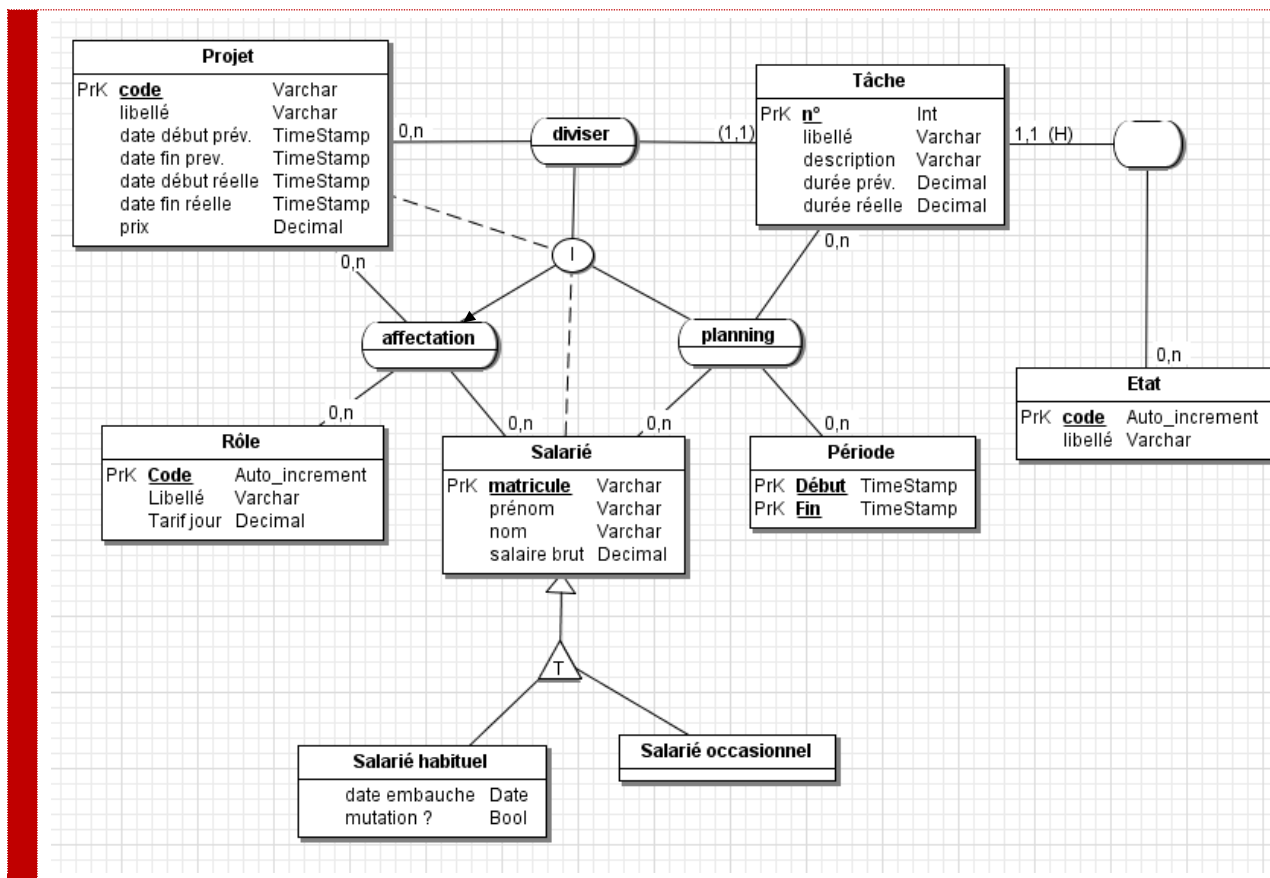
Le logiciel stockera les salariés ingénieurs étude et développement intervenant de manière habituelle sur ses plateaux de développement. En pratique, l'appartenance à un plateau de développement n'a aucune importance. Lorsque la charge de travail devient trop importante, PRASORIASTE fait appel à des ingénieurs en provenance d'autres agences. Ces salariés occasionnels seront également stockés en base de données. Il conviendra de pouvoir identifier les salariés qui sont habituels de ceux qui ne le sont pas. On notera qu'un salarié occasionnel finit parfois par devenir un salarié habituel en cas de mutation. Et il convient de savoir quand la mutation a eu lieu.

PRASORIASTE, selon le cas, souhaite pouvoir évaluer l'amélioration de l'efficacité de ses salariés à compter de leur date d'embauche ou à compter de leur mutation. Quoiqu'il en soit, tous les salariés sont identifiés par un matricule, ont un prénom, un nom et un salaire brut.

Le logiciel permettra de suivre des projets, ayant entre autres un code, un libellé, une date de début et une date fin prévisionnelles, une date de début et une date fin effectives, un prix de vente. Un projet est divisé en tâches ayant une référence faisant elle-même référence au projet (exemple : projet « ALPHABRAVO », la tâche de référence « #10 » s'écrit « ALPHABRAVO#10 »). Chaque tâche a par ailleurs un libellé, une description, une durée prévisionnelle et une durée effective exprimée en jours homme. Une tâche a finalement une donnée état (exemple : « En attente », « En cours », « Terminée ») pouvant varier dans le temps.

Chaque salarié affecté au projet a un ou plusieurs rôles sur ce projet (exemple : « chef de projet », « développeur », « testeur », etc.). D'ailleurs, plusieurs salariés peuvent avoir le même rôle sur ce projet. Et à chaque rôle correspond un tarif journalier, exprimé en €. Les salariés affectés au projet peuvent ensuite être affectés à une tâche pour une période donnée.

Question : proposer un MCD permettant de répondre au besoin décrit ci-avant. Vous justifierez succinctement vos choix (extensions Merise 2 et cardinalités) sur votre copie.



Justifications :

- o **Contrainte d'inclusion** entre les associations « planning », « affectation » et « diviser » : elle permet de modéliser la règle de gestion qui consiste à faire de sorte qu'un salarié ne puisse être affecté à une tâche d'un projet que si ce salarié est affecté à ce projet ;

- **Identifiant relatif (1,1)** entre tâche et projet : une tâche est relative à un projet. Autrement dit, elle n'a de sens que pour un projet donné. Elle en est dépendante. Si l'on supprime le projet, les tâches du projet n'ont plus lieu d'être conservées. D'où la nécessité de recourir à un identifiant relatif ;
- Une **tâche a un seul état** à un instant donné, d'où la cardinalité 1,1. Cependant, l'état varie au travers du temps. L'on peut dès lors souhaiter conserver l'historique des états que la tâche a pris au travers du temps. On modélise cette **historisation grâce au symbole (H)** ;
- **Spécialisation** quant aux salariés : on distingue deux types de salariés ayant quelques propriétés distinctes, ce qui justifie le recours à une spécialisation. On utilise un **héritage de type « T »** (totalité) car un salarié est nécessairement de l'un de ces deux types. Toutefois, un salarié occasionnel peut devenir salarié habituel. Les deux types ne s'exclut donc pas ;
- **Association ternaire** « affectation » : elle permet tout bonnement de respecter le fait qu'un salarié peut participer à plusieurs projet, qu'il peut avoir plusieurs rôles sur un même projet et que plusieurs salariés peuvent avoir le même rôle sur un même projet ;
- **Association ternaire** « planning » : une tâche peut être commune à plusieurs salariés et un salarié peut avoir plusieurs tâches figurant dans son planning. Toutefois, il est convenu que la tâche est attribuée au salarié pour une période donnée. Ceci ajoute une **dimension temporelle** à cette relation. En notation Merise, cela se traduit par la nécessité d'ajouter une troisième entité, représentant la dimension temporelle. Finalement, obtient une ternaire.

Barème :

- Bonnes entités : **1 point** ;
N.B. : entités cohérentes
- Bonnes propriétés : **1 point** ;
N.B. : présence des informations nécessaires au logiciel, clefs primaires soulignées
- Bonnes cardinalités : **1 point** ;
- Spécialisation : **1,5 point**
Dont : spécialisation présente (0,5 point) + type de spécialisation (1 point si « T » ; 0,5 si « XT » ; 0 sinon)
- Ternaire « affectation » : **1 point** ;
- Ternaire « planning » : **1 point**
Alternative : représentation approximative au moyen de deux propriétés portées (0,5 point)
- Contrainte d'inclusion : **1 point**
- Identifiant relatif : **1 point**
- Historisation : **0,5 point**

Commentaire : toute solution cohérente et justifiée était acceptée.

Exercice 2 : modèle logique et requêtes 11 points

Sujet :

A l'aide du MCD fourni en annexe, répondez aux questions suivantes.

Questions :

1. Rédiger le schéma relationnel correspondant à toutes les entités et associations, sauf :
 - associations : « réservation », « montable sur » ;
 - entités : « CLIENT », « DEPOT » et « MATERIEL ».

Article(Code, Libelle, PrixUnitaire, Type)

Clef primaire : Code

ArticleRemplacement(#Original, #Remplacant)

Clef primaire : Original, Remplacant

Clefs étrangères :

- Original en référence à l'attribut Code de la relation Article

- Remplacant en référence à l'attribut Code de la relation Article

PieceDatachee(#CodeArticle, Fonction, Fabriquant)

Clef primaire : CodeArticle

Clef étrangère : CodeArticle en référence à Article(Code)

Consommable(#CodeArticle, Categorie, Conditionnement)

Clef primaire : CodeArticle

Clef étrangère : CodeArticle en référence à Article(Code)

Stock(#SigleDepot, #CodeArticle, QteDisponible)

Clef primaire : SigleDepot, CodeArticle

Clefs étrangères :

- SigleDepot en référence à Depot(Sigle)

- CodeArticle en référence à Article(Code)

Commande(Num, Creation, Livraison, #NumClient, #NumFacturation, #SigleDepot)

Clef primaire : Num

Clefs étrangères :

- NumClient en référence à Client(Num)

- NumFacturation en référence à Client(Num)

- SigleDepot en référence à Depot(Sigle)

LigneCommande(#NumCommande, Num, #CodeArticle, QteCommandee)

Clef primaire : NumCommande, Num

Clefs étrangères :

- NumCommande en référence à Commande(Num)

- CodeArticle en référence à Article(Code)

Barème :

- Bonnes clefs primaires : **1,5 point** (-0,5 point par clef erronée) ;
- Bonnes clefs étrangères : **2,5 points** (-0,5 point par clef manquante ou erronée).

2. Quelles sont les contraintes d'associations ? Et quelle est la signification de chacune d'entre elles ?

Il y a deux contraintes d'associations :

- Une **contrainte d'exclusion** entre les associations « passé par » et « facturé à autre » : cette contrainte signifie que, si la commande passée par un **client X** est facturée à un **client Y**, on doit impérativement avoir **X ≠ Y** ;
- Une **contrainte d'inclusion** entre les associations « réservation » et « stocké dans » : cette contrainte signifie qu'il ne doit être possible de **réserver un article que s'il figure dans le stock** d'un dépôt.

Barème : **2 points**, un par contrainte d'association (dénomination + signification).

3. Etablir la requête SQL permettant de consulter la liste de tous les articles (code et libellé) avec son stock global (qu'importe le dépôt).

SELECT Code, Libelle, SUM(QteDisponible)

FROM Article **INNER JOIN** Stock **ON** Article.Code = Stock.CodeArticle
GROUP BY Code, Libelle ;

Barème : 3 points, un point par ligne juste.

4. Etablir la requête SQL permettant de connaître par année les articles (code et libellé) commandés plus de 100 fois.

SELECT Article.Code, Article.Libelle, YEAR(Commande.Creation)
FROM Article **INNER JOIN** LigneCommande **ON** Article.Code = LigneCommande.CodeArticle
INNER JOIN Commande **ON** LigneCommande.NumCommande = Commande.Num
GROUP BY Article.Code, Article.Libelle, YEAR(Commande.Creation)
HAVING SUM(QteCommandee) > 100

Barème : 2 points. Toute réponse cohérente est acceptée.

Annexe - modèle de donnée « gestion de commandes »

