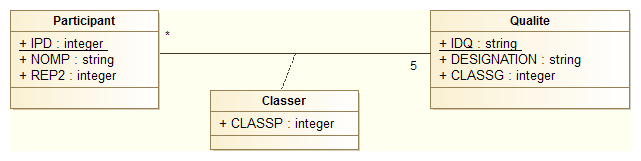
On se propose d'implanter le diagramme de classes suivant pour gérer le concours de la société YSLAV. La société YSLAV, numéro un sur le marché des poudres à laver avec sa lessive SAMOUSS, décide d’organiser un grand concours pour célébrer la sortie de son milliardième baril. Ce concours comporte deux questions figurant sur un bon de participation diffusé par différents médias. La société édite deux millions de bons de participation et compte recevoir en retour 1 million de réponses parmi lesquelles on déterminera 1000 gagnants.



Le schéma ci-dessus montre qu’un participant classe en fonction de ses préférences les **cinq** qualités qui lui sont proposées. Ces qualités représentent le ressenti du participant concernant cette lessive.

Le modèle relationnel correspondant au diagramme ci-dessus est le suivant :

* PART (IDP, NOMP, REP2)
* QUAL (IDQ, DESIGNATION, CLASSG)
* CLASS (#IDP, #IDQ, CLASSP)

Pour que la participation d’un joueur soit valide, il doit impérativement :

* Classer les cinq qualités par ordre de **préférence de 1 à 5** : 1 correspond à la qualité qu’il considère comme la plus représentative et 5 comme celle qui lui semble la moins pertinente. ***Il ne peut en aucun cas donner à deux qualités le même classement*** ;
* Mettre dans REP2 son estimation du nombre de participants ayant trouvé le classement correspondant au classement type calculé à partir des réponses de tous les participants.

CLASSG dans QUAL correspond au classement type. *Une fois le concours clos, le classement est calculé via une requête SQL et les résultats de celle-ci est saisi manuellement dans la relation QUAL.*

CLASSP dans CLASS correspond au classement fourni par chaque participant.

**Remarques** :

* IDP est un identifiant de type numérique. IDQ est un identifiant alphabétique codé **sur un** caractère ;
* CLASSG et CLASSP ont le même type ;
* Pour chaque réponse, vous veillerez à fournir une solution optimale. ***La seule syntaxe acceptée est celle d’Oracle***.

# Création des relations *(2,25 pts)*

1. Créez les relations PART, CLASS et QUAL sans positionner de contraintes ***(0,75 pt)***

create table PART (

IDP number,

NOMP varchar2(50),

REP2 number) ; ***(0,25 pt)***

create table QUAL (

IDQ char(1),

DESIGNATION varchar2(50),

CLASSG number(1)) ; ***(0,25 pt)***

create table CLASS (

IDP number,

IDQ char(1),

CLASSP number(1)) ; ***(0,25 pt)***

1. Mettez en œuvre les contraintes de clefs primaires et de clefs étrangères ***(1,50 pts)***

alter table PART

add constraint PK\_PART primary key (IDP) ; ***(0,25 pt)***

alter table QUAL

add constraint PK\_QUAL primary key (IDQ) ; ***(0,25 pt)***

alter table CLASS

add constraint PK\_QUAL primary key (IDP, IDQ) ; ***(0,50 pt)***

alter table CLASS

add constraint FK\_IDP\_QUAL foreign key (IDP) references PART ; ***(0,25 pt)***

alter table CLASS

add constraint FK\_IDQ\_QUAL foreign key (IDQ) references QUAL ; ***(0,25 pt)***

# Compréhension du schéma *(4,75 pts)*

1. Justifiez la cardinalité « 5 » dans le schéma UML.  Que signifie-t-elle par rapport au problème posé et de quelle information est-elle déduite ? ***(1,00 pt)***

Cette cardinalité est induite par le fait que chaque participant doit classer les cinq qualités pour que sa participation soit validée.

1. Quelle(s) contrainte(s) faut-il implanter pour **répondre au mieux au problème posé** ? indiquez celle(s) qui peu(ven)t être implantée(s) en SQL et celle(s) qui ne le peu(ven)t pas ? Dans le cas où la contrainte peut être implantée vous fournirez la requête correspondante et dans le cas où l’implantation est impossible vous expliquerez pourquoi ***(2,75 pts)***

La première contrainte qui devrait être mise en œuvre concerne la multiplicité « 5..5 ». Cette contrainte ne peut pas être mise en œuvre car il faudrait vérifier que chaque IDP est présent exactement cinq fois dans la table CLASS. ***(0,50 pt)***

La deuxième contrainte concerne l’obligation de fournir une réponse pour REP2. Il faut donc déclarer une contrainte not null sur REP2 ***(0,25 pt)***.

alter table PART

modify REP2 constraint NN\_REP2\_PART not null ; ***(0,50 pt)***

La troisième contrainte concerne le fait qu’un participant ne peut pas donner le même classement à deux qualités différentes. Il faut mettre une contrainte d’unicité sur le binôme (IDP, CLASSP) de la classe CLASS

alter table CLASS

add constraint UNIQ\_IDP\_CLASSP\_CLASS unique (IDP, CLASSP) ; ***(0,75 pt)***

La quatrième contrainte concerne les valeurs possibles pour classer les qualités. Les seules valeurs possibles sont 1, 2, 3, 4 et 5. ***(0,25 pt)***

alter table CLASS

add constraint CHK\_CLASSP\_CLASS check (CLASSP between 1 and 5) ; ***(0,50 pt)***

1. Proposez un exemple pertinent, couvrant les différents cas possibles, de cette base de données en fournissant quelques enregistrements pour les relations « PART » et « CLASS ». ***(1,00 pt)***

|  |  |  |  |
| --- | --- | --- | --- |
| **QUAL** | **IDQ** | **DESIGNATION** | **CLASSG** |
|  | A | SANS PHOSPHATE |  |
|  | B | LAVE PLUS BLANC |  |
|  | C | TOUS TEXTILES |  |
|  | D | CONSERVE LES COULEURS |  |
|  | E | TOUTES TEMPERATURES |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **PART** | **IDP** | **NOMP** | **REP2** |
|  | 1 | DUPONT Pierre | 10 |
|  | 2 | DURANT Louis | 6 |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **IDP** | **IDQ** | **CLASSP** |
|  | 1 | A | 1 |
|  | 1 | B | 3 |
|  | 1 | C | 5 |
|  | 1 | D | 4 |
|  | 1 | E | 2 |
|  | 2 | A | 1 |
|  | 2 | B | 3 |
|  | 2 | C | 5 |
|  | 2 | D | 2 |
|  | 2 | E | 4 |

# Donnez les requêtes suivantes dans le langage SQL *(8,50 pts)*

1. Mettre en place la contrainte qui stipule que pour la relation « PART » le nom du participant est toujours renseignée ***(1,00 pt)***

alter table PART

modify NOMP constraint NN\_NOMP\_PART not null ;

1. Pour chaque participant, son identifiant, son nom et la désignation des qualités triées par ordre croissant du classement ***(1,00 pt)***

select IDP, NOMP, DESIGNATION

from PART, QUAL, CLASS

where PART.IDP = CLASS.IDP

and CLASS.IDQ = QUAL.IDQ

order by IDP, CLASSP ;

1. Le résultat du concours s’obtient en faisant la somme des classements des différentes qualités et en ordonnant ces sommes par ordre croissant. Si pour deux qualités les sommes sont identiques, l’ordonnancement se fera par ordre alphabétique des identifiants. Donnez la requête permettant de calculer ce classement. ***(1,00 pt)***

select IDQ, sum(CLASSP)

from CLASS

group by IDQ

order by 2, IDQ

***Remarque*** : il ne sera pas tenu compte de la manière dont le classement est trié car la syntaxe n’a pas été formalisée en cours. Il en résulte que « order by sum(CLASSP) » sera considéré comme correct.

1. Donnez sous forme ***d’algèbre relationnelle***, le nom du participant, l’identifiant de chaque qualité et le classement qu’il a réalisé. (***1,50 pt***)

**// ou** 

1. Donnez l’identifiant des participants ayant classé les qualités de la lessive YSLAV dans le même ordre que le classement type. Pour ce faire vous considèrerez que l’attribut « CLASSG » de la relation « CLASS » est correctement renseigné. ***(2,00 pts)***

select IDP

from PART

where not exists ( select IDQ, CLASSG

from QUAL

minus

select IDQ, CLASSP

from CLASS

where PART.IDP = CLASS.IDP) ;

***ou***

select IDP

from PART

where not exists ( select IDQ, CLASSP

from CLASS

where PART.IDP = CLASS.IDP

minus

select IDQ, CLASSG

from QUAL) ;

1. Donnez la valeur moyenne de la réponse 2 ***(1,00 pt)***

select avg(REP2)

from PART;

1. Donnez le nombre de participant ayant classé la qualité d’identifiant « D » en première position ***(1,00 pt)***

select count(\*)

from CLASS

where CLASSP = 1

and IDQ = “D”  ;

# Implantez les index qui vous semblent pertinents *(3,00 pts)*

1. Quels sont les index positionnés automatiquement sur les quatre relations de l’application. ***Vous*** ***justifierez votre réponse***. ***(1,50 pts)***

Dans PART : IDP car c’est une clef primaire ; (0,25 pt)

Dans QUAL : IDQ car c’est une clef primaire ; (0,25 pt)

Dans CLASS : (IDP, IDQ) car c’est une clef primaire ; (0,50 pt)

(IDQ, CLASSP) car contrainte unique. (0,50 pt)

1. ***Implantez*** les index qui vous semblent pertinents en expliquant ***pourquoi ces index sont pertinents***. Vous ferez les hypothèses qui vous semblent nécessaires. ***(1,00 pt)***

Dans CLASS : IDQ car c’est une partie droite de clef primaire qui est aussi clef étrangère ;

Create index IDX\_CLASS\_IDQ on CLASS(IDQ) ; si IDP est proposé enlevé ***0,25 pt***

1. Après étude d’une statistique, il s’avère intéressant de positionner des index sur les attributs CLASSG et CLASSP. Parmi les différents types d’index présentées dans le cours, indiquez celui qui vous semble le plus pertinent dans ce cas. ***Vous justifierez votre choix***. En particulier, vous indiquerez quelles informations présentes dans le sujet ont guidé votre réflexion. ***(0,50 pt)***

Vu que ces attributs peuvent prendre uniquement les valeurs 1, 2, 3, 4 et 5 c’est-à-dire un petit nombre de valeur. Dans ce cas un index bitmap semble le plus adapté. Ce choix est guidé par le fait qu’il y a un nombre de qualités limité par hypothèse.

# Interprétation des plans d’exécution *(1,50 pts)*

1. En considérant les six formes de requêtes (minus, not exists, not in et demi-jointure, distinct et order by), recopiez et indiquez dans le tableau ci-dessous les opérations apparaissant dans le plan d’exécution si on les exécute sous **Oracle 10g** et qu’aucun index n’est positionné. ***(1,50 pts)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Mettre O (Oui) ou N (Non) dans CHAQUE case** | ***sort*** | ***hash***  ***(unique)*** | ***hash join*** |
| not exists | **N** | **N** | **O** |
| not in | **N** | **N** | **N** |
| demi-jointure | **N** | **O** | **O** |
| minus | **O** | **N** | **N** |
| Select distinct IDP from CLASS | **N** | **O** | **N** |
| Select NOMP from PART order by NOMP | **O** | **N** | **N** |