

```

-- corr_contraintes.sql
-- tourne le 23 janvier 2018
/* Contraintes SQL et non SQL.

Contraintes SQL (bonus en partie/examen si presentation comme suit) :

client :
  idc pk
  nom not null
  age not null check 16<=age<=120
  avoir not null check 0<=avoir<=2000

village :
  idv pk
  ville not null
  activite -- null possible
  idv fk client not null
  prix not null check 0<=prix<=2000
  capacite not null check 1<=capacite<=1000

sejour :
  ids pk
  idc fk client not null -- rappel : fk n'implique pas not null
  idv fk village not null
  jour not null check 1<=jour<=365
  (idc, jour) unique

creer sequences pour client, village, sejour
Contraintes non SQL :

1. le nombre de sejours pour un centre pour un jour ne peut pas
dépasser sa capacité :
  pour tout idv i de capacité n dans village,
  il y a au plus n lignes avec idv i pour chaque jour j dans sejour

2. l'avoir d'un client plus la somme des prix de ses sejours ne peut
exceder 2000 :
  pour tout idc i dans client : avoir + S <= 2000
  en effet, un client part de 2000 et achete des sejours, donc :
  avoir + somme des prix de ses sejours presents + somme des prix de
  ses sejours detruits = 2000 ;
  donc avoir + somme des prix de ses sejours presents <= 2000 ;
  comment obtient-on S : considerons toutes les lignes d'idc i dans
  sejour, et pour chacune, a partir de sa colonne idv dans sejour, son
  prix dans village identifie par idv ; on note S la somme de tous ces
  prix
*/

/* Rappel :
Null :
Peuvent être "null" les valeurs intuitivement non
indispensable au fonctionnement de l'application. De plus, par
convention dans le module les colonnes utilisées en paramètres ou a
l'intérieur d'un traitement ne peuvent être "null", mais il n'est pas
posé que celles en sortie d'un traitement ne le soient pas.
Rappel : fk et check n'impliquent pas not null.

Contraintes non SQL :
en general, il s'agit de compter un ensemble de lignes d'une table, ou de
comparer des valeurs dans deux tables, ou des variantes de ces situations
(ex : somme d'un ensemble de lignes). */

-- Ordres SQL et modeles d'ordre : completer avec nouveaux outils TD precedent
-- Ordres topes par le programmeur :

```

```

-- corr_independance.txt
-- 13 avril 2021

-- 6.3.1 : le probleme :

-- un programme correct :
create or replace procedure traitement2(
  l_idc client.idctype,
  la_ville village.ville%type,
  le_jour sejour.jour%type,
  l_idv out village.idv%type,
  l_ids out sejour.id%type,
  l_activite out village.activite%type)
is
  cursor c is
    select idv, prix, activite
      from village
     where ville = la_ville
     order by prix desc;
  le_prix village.prix%type;
begin
  open c;
  fetch c into l_idv, le_prix, l_activite;
  if c%found then
    l_ids := seq_sejour.nextval;
    insert into sejour
      values(l_ids, l_idc, l_idv, le_jour);
    update client
      set avoir = avoir - le_prix
      where idc = l_idc;
  else
    l_idv := -1;
    l_ids := -1;
    l_activite := 'neant';
  end if;
end;
/

-- une reorganisation SPI :
remplacer la table village
-- (1, Rio, kitesurf, 10, 25)
-- (2, Rio, sieste, 5, 250)
-- en :
-- activ(idv, prix, activite)
-- (1, 10, kitesurf)
-- (2, 5, sieste)
-- et :
-- lieu(idv, ville, capacite)
-- (1, Rio, 25)
-- (2, Rio, 250)
-- est une reorganisation ? oui
-- SPI ? oui

-- remarque : renommer village en village2 : est une reorganisation SPI

-- un appel du (meme) programme terminant en erreur apres la
reorganisation :
-- il termine en erreur ? oui : -942 : la table village n'existe pas

-- bref ce code est parfait logiquement mais "n'est pas robuste pour
l'independance"
declare

```

```

corr_confidentialite.txt
12 avril 2023

-- l'exercice 7.3 est un exercice d'annales. Voici la reponse complete
suffisante attendue. En examen il est totalement inutile d'ajouter des
commentaires.

-- comptes utilisateurs SGBD :
programmeur : p
employes : e1, ..., en
client : c

c'est p qui tape sur son compte tout ce qui suit

-- identification et suppression des droits existants :
q1= select grantee, table_name, privilege
  from user_tab_privs
  where table_name in ('CLIENT', 'VILLAGE', 'SEJOUR')
q2= select grantee, procedure_name
  from user_proc_privs
  where procedure_name in ('TRAITEMENT1', 'TRAITEMENT2', 'TRAITEMENT3')
retirer ces droits :
- pour chaque u dans q1 faire :
  revoke u.privilege on client, village, sejour from u.grantee
- pour chaque u dans q2 faire :
  revoke execute on traitement1, traitement2, traitement3 from u.grantee

-- programmation des actions :

grant insert village to e1, ..., en
grant select village e1...en
grant update(activite,capacite) village e1...en
grant select sejour e1...en

create procedure traitement3
grant execute traitement3 e1...en

create procedure traitement1
grant execute traitement1 c

create procedure traitement2
grant execute traitement2 c

create view village_sans_sejour as
  select idv,ville,activite,prix from village
  where idv not in (select idv from sejour)
grant select on village_sans_sejour to c

create procedure consulter_client(i)=select client where idc=i
create procedure consulter_sejour(i)=select sejour where idc=i
create procedure consulter_village(i)=
  select idv,ville,activite,prix,capacite from village,sejour
  where village.idv=sejour.idv and idc=i
grant execute consulter_client,consulter_sejour, consulter_village c

create procedure authentication(i,n)=select client where idc=i and nom=n
grant execute authentication c

iv village.idv%type;
l_ids sejour.id%type;
a village.activite%type;
begin
  traitement2(1, 'Chatelaillon', 361, iv, l_ids, a);
  dbms_output.put_line('idv ||iv||', ids '||l_ids||', activite '||a);
end;
/

-- 6.3.2 : corrigez l'appli pour que ce code soit en plus robuste aux
problemes
-- d'independance des niveaux

-- ???...

Une fois le problème existant il est trop tard.
Donc, il faut le gérer en amont, lors de la création de l'application,
comme suit.

A l'origine, écrire une appli "correcte".
1. créer les tables : village
2. créer une vue par table : vue_village = select * from village
3. coder les procédures PL/SQL SUR LES VUES :
create or replace procedure traitement2 ... is
  cursor c is
    select idv, prix, activite
      from vue_village
     where ville = la_ville
     order by prix desc;
  ...
begin
  open c;
  ...
end;

Au moment de la réorganisation :

4. bloquer la base
4bis. réorganiser :
remplacer la table village par activ(idv, prix, activite) et lieu(idv,
ville, capacite)

5. redéfinir la vue (pour chaque table) vue_village POUR QUE elle
contienne
toujours la meme chose :
create or replace vue_village as
  select activ.idv, ville, activite, prix, capacite
    from activ, lieu
   where activ.idv = lieu.idv
: vue_village contient exactement l'ancienne table village.

Sbis. débloquer la base

declare
  iv village.idv%type;
  l_ids sejour.id%type;
  a village.activite%type;
begin
  traitement2(1, 'Chatelaillon', 361, iv, l_ids, a);
  dbms_output.put_line('idv ||iv||', ids '||l_ids||', activite '||a);
end;
/

: cet appel donne exactement le même résultat que sur la base avant
réorganisation.

```

```

corr_mpphp.txt (10 mai 2018)
<!--
<!-- fichier traitement3.php
https://tp-ssh1.../~ewaller/.../traitement3.php?jour=50 -->

<html>
<body bgcolor=yellow>
<?php
    $texte = "delete sejour where jour<".$_REQUEST['jour'];
    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");
    $ordre = oci_parse($connection, $texte);
    oci_execute($ordre);
    echo "nombre de lignes detruites : ".oci_num_rows($ordre);
    oci_logoff($connection);
?>
</body>
</html>

<!--
<!-- fichier authentication.php
https://tp-ssh1.../~ewaller/.../authentification.php?idc=50&nom=Rita -->

<html>
<body bgcolor=yellow>
<?php
    $texte =
        "select *
         from client
         where idc = '$_REQUEST['idc']"
           and nom = '$_REQUEST['nom']";
    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");
    $ordre = oci_parse($connection, $texte);
    oci_execute($ordre);
    if (oci_fetch_into($ordre, $ligne))
        echo "bienvenue ".$ligne[0].", ".$ligne[1].", ".$ligne[2].", ".
    $ligne[3];
    else
        echo "desole, erreur identifiant/nom";
    oci_logoff($connection);
?>
</body>
</html>

<!--
<!-- fichier traitement3Appel.php
https://tp-ssh1.../~ewaller/.../traitement3Appel.php?jour=50 -->

<html>
<body bgcolor=lightblue>
<?php
    $texte="begin :1 := traitement3('$_REQUEST['jour'].'); end;";
    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");
    $ordre = oci_parse($connection, $texte);
    oci_bind_by_name($ordre, ':1', $nb_detruits);
    oci_execute($ordre);
    echo "nombre de lignes detruites : ".$nb_detruits;
    oci_logoff($connection);
?>
</body>
</html>

```

```

<!--
<!-- fichier traitement3outAppel.php
https://tp-ssh1.../~ewaller/.../traitement3outAppel.php?jour=50 -->

<html>
<body bgcolor=lightblue>
<?php
    $texte="begin traitement3_out('$_REQUEST['jour'].', :1); end;";
    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");
    $ordre = oci_parse($connection, $texte);
    oci_bind_by_name($ordre, ':1', $nb_detruits);
    oci_execute($ordre);
    echo "nombre de lignes detruites : ".$nb_detruits;
    oci_logoff($connection);
?>
</body>
</html>

<!--
<!-- fichier villages.php
https://tp-ssh1.../~ewaller/.../villages.php?idc=50 -->

<html>
<body bgcolor=lightgreen>
<?php
    $texte =
        "select village.idv, ville, activite, prix, capacite
         from village, sejour
         where sejour.idc = '$_REQUEST['idc']"
           and village.idv = sejour.idv";
    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");
    $ordre = oci_parse($connection, $texte);
    oci_execute($ordre);
    while (oci_fetch_into($ordre, $ligne))
        echo
            $ligne[0].", ".$ligne[1].", ".$ligne[2].", ".$ligne[3].",
            ".$ligne[4]
            .<br/>";
    oci_logoff($connection);
?>
</body>
</html>

<!--
<!-- fichier traitement1.php
https://tp-ssh1.../~ewaller/.../traitement1.php?nom=Marie&age=20 -->

<html>
<body bgcolor=red>
<?php
    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");

    $texte = "select seq_client.nextval from dual";
    $ordre = oci_parse($connection, $texte);
    oci_execute($ordre);
    oci_fetch_into($ordre, $ligne);
    $idc = $ligne[0];

    $texte =
        "insert into client(idc, nom, age) values("

```

**RAPPEL : ATTENTION: OCI: ANCIENNE SYNTAXE IN CORRECTE**

```

        . $idc.", " $_REQUEST['nom'].", " $_REQUEST['age'].");";
    $ordre = oci_parse($connection, $texte);
    oci_execute($ordre);

    echo "identifiant client cree : ".$idc;
    oci_logoff($connection);
?>
</body>
</html>

<!--
<!-- fichier traitement2Appel.php
https://tp-ssh1.../~ewaller/
.../traitement2Appel.php?idc=5&ville=Chatelaillon&jour=1 -->

<html>
<body bgcolor=redblue>
<?php
    $texte = "begin traitement2('$_REQUEST['idc'].', '$_
    $_REQUEST['ville'].', "
        . $_REQUEST['jour'].', :1, :2, :3) ;
end;";
    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");
    $ordre = oci_parse($connection, $texte);
    oci_bind_by_name($ordre, ':1', $idv);
    oci_bind_by_name($ordre, ':2', $ids);
    oci_bind_by_name($ordre, ':3', $activite, 2);
    // rem : 2 convient pour tango mais pas 1 ; PHP reserve
    // probablement 4
    // octets ; si chaîne ou entier (ex : 1000) depasse alors
    allouer plus
    oci_execute($ordre);
    oci_logoff($connection);
    echo "identifiant village : ".$idv;
    echo "identifiant sejour : ".$ids;
    echo "activite : ".$activite;
?>
</body>
</html>

<!--
<!-- fichier traitement2.php
https://tp-ssh1.../~ewaller/
.../traitement2.php?idc=5&ville=Chatelaillon&jour=1 -->

<html>
<body bgcolor=greenyellow>
<?php
    $idc = $_REQUEST['idc'];
    $ville = $_REQUEST['ville'];
    $jour = $_REQUEST['jour'];

    $l_idv = -1;
    $l_ids = -1;
    $l_activite = 'neant';

    $connection = oci_logon("c#ewaller_a", "ewaller_a", "dbinfo");

    $texte =
        "select idv, prix, activite
         from village
         where ville = '$_ville.'"

```

```

        order by prix desc";
    $ordre = oci_parse($connection, $texte);
    oci_execute($ordre);

    if (oci_fetch_into($ordre, $ligne)) {
        $idv = $ligne[0];
        $prix = $ligne[1];
        $activite = $ligne[2];

        $texte = "select seq_sejour.nextval from dual";
        $ordre = oci_parse($connection, $texte);
        oci_execute($ordre);
        oci_fetch_into($ordre, $ligne);
        $ids = $ligne[0];

        $texte =
            "insert into sejour values('".$ids.", " . $idc.", " . $idv.", " .
            $jour.");";
        $ordre = oci_parse($connection, $texte);
        oci_execute($ordre);

        $texte = "update client set avoir = avoir - ".$prix." where idc
        = ".$idc;
        $ordre = oci_parse($connection, $texte);
        oci_execute($ordre);
    }
    oci_logoff($connection);
    echo "identifiant village : ".$idv."<br/>";
    echo "identifiant sejour : ".$ids."<br/>";
    echo "activite : ".$activite."<br/>";
?>
</body>
</html>

<!--
<!-- tourne le 19 decembre 2023 -->
<!-- execution ordre BD : recapitulatif nouvelle version PHP 8.3 -->

<?php
    $prix = $_REQUEST['x'];
    $x = $_REQUEST['y'];
    $c = oci_connect("c#ewaller_a", "ewaller_a", "dbinfo");

    $texte =
        "select nom, prix "
        "from produit "
        "where prix <= ".$prix;
    $ordre = oci_parse($c, $texte);
    oci_execute($ordre);
    while($ligne = oci_fetch_row($ordre)) {
        echo $ligne[0].", ".$ligne[1]."<br/>";
        oci_close($c);

        $texte="begin p3('$_x.", :1, :2); end;";
        $ordre = oci_parse($c, $texte);
        oci_bind_by_name($ordre, ':1', $z);
        oci_bind_by_name($ordre, ':2', $w, 100);
        oci_execute($ordre);
        echo "valeurs renvoyées par p3 : ".$z.", ".$w."<br/>";

        oci_close($c);

```

**RAPPEL : ATTENTION : SYNTAXE OCI ANCIENNE CORRECTE**