TP Index – version 1

Objectifs et contexte du TP Index:

Savoir consulter le catalogue d'une base de données pour connaitre la taille d'une table et le domaine de ses attributs.

Savoir définir un index.

Comprendre l'utilisation des index pour évaluer des sélections.

Choix d'une méthode d'accès aux données : accès avec un ou plusieurs index, accès par lecture séquentielle.

Ces notions sont mises en application dans le contexte de la vente de produits où des clients passent des commandes. On s'intéresse aux requêtes couramment posées telles que celles pour connaitre les commandes d'un client, d'un produit, et toute autre analyse à laquelle vous pouvez penser.

La question générale guidant ce TP est : quels index peut-on définir pour accélérer les requêtes posées ? Pour cela : vous définissez votre base, vos index et montrez qu'ils sont bénéfiques pour les diverses requêtes que vous avez proposées.

Préparation

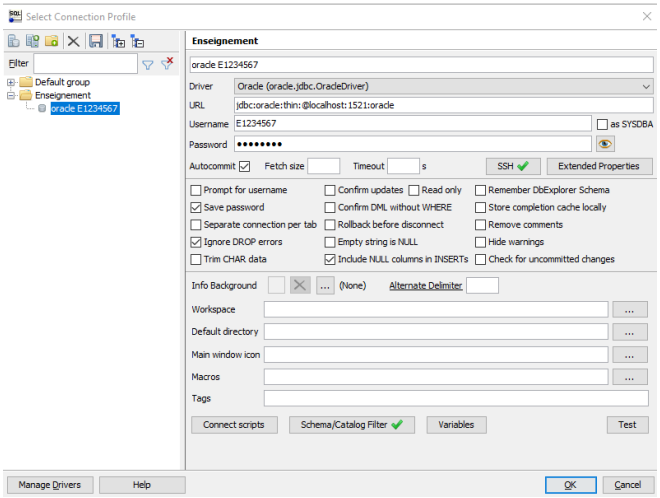
A faire **AVANT** la séance !

On utilise Oracle et l'interface SQLWorkbench.

1) Commencer par suivre les instructions Oracle avec SQLWorkbench pour installer les outils requis.

2) puis vérifier que les options suivantes sont bien cochées : ☒Autocommit, ☒Save password, ☒Ignore DROP errors.

3) Vérifier aussi que les case *Separate connection per tab* et *Remove comments* ne sont **PAS** cochées. Il est important de laisser décochée la case Remove comments afin de conserver les commentaires contenus dans une requête SQL pour qu'ils soient pris en compte par Oracle.



Rmq : si vous êtes sur une machine de la PPTI : ajuster le champ URL en conséquence.

3) Compléter la configuration telle que décrite ci-dessous.

Configuration nécessaire pour ce TP:

Accès au catalog

Ajouter un filtre pour explorer plus facilement les tables que vous allez créer et les tables de la base TPCH qu'on utilise dans ce TP :

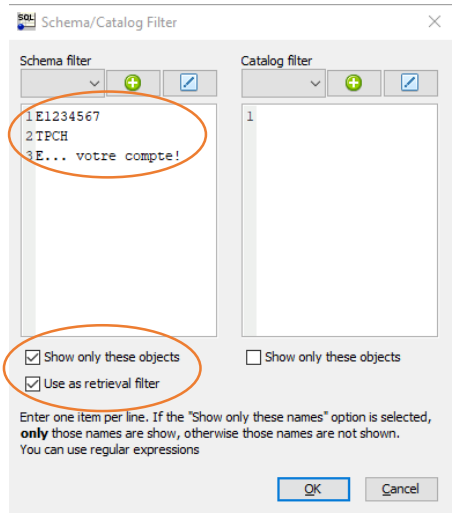
Dans File → Connect Window, cliquer sur le bouton « Schema/Catalog Filter », cela ouvre la fenêtre suivante, dans laquelle vous compléter la zone *Schema filter* avec deux nouvelles lignes :

1. TPCH
2. E*numéro,* avec *numéro* étant votre numéro d'étudiant, par exemple E1234567

Puis cocher les cases

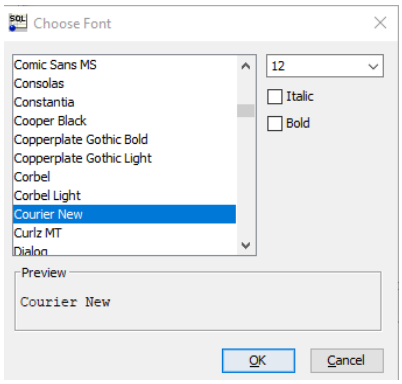
☒ Show only these objects

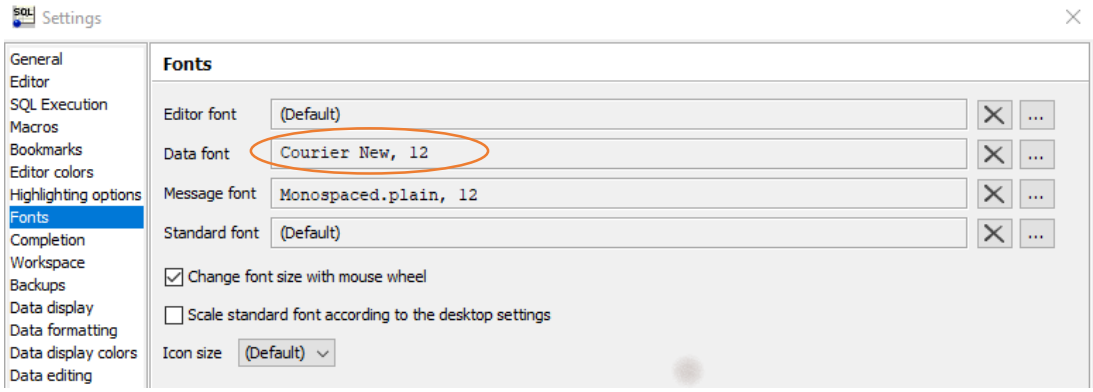
☒ Use as retrieval filter



Affichage d'un plan d'exécution

Pour afficher proprement les plans il faut utiliser la police de caractère "Courrier New". Pour cela, aller dans le menu Tools → Options → Fonts et modifier le champ *Data Font* : cliquer sur le bouton «…» puis, dans la fenêtre *Choose Font,* sélectionner **Courrier** ou **Courrier New,** en taille **12**

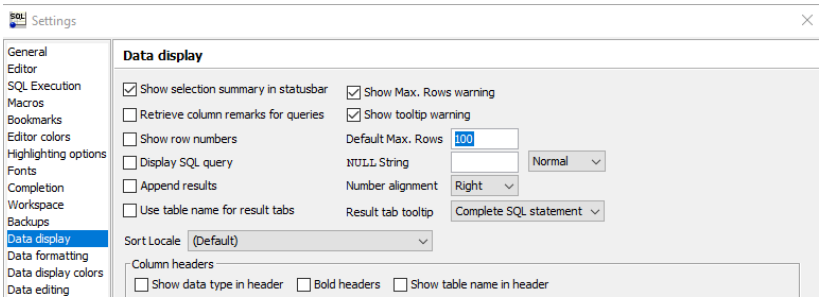




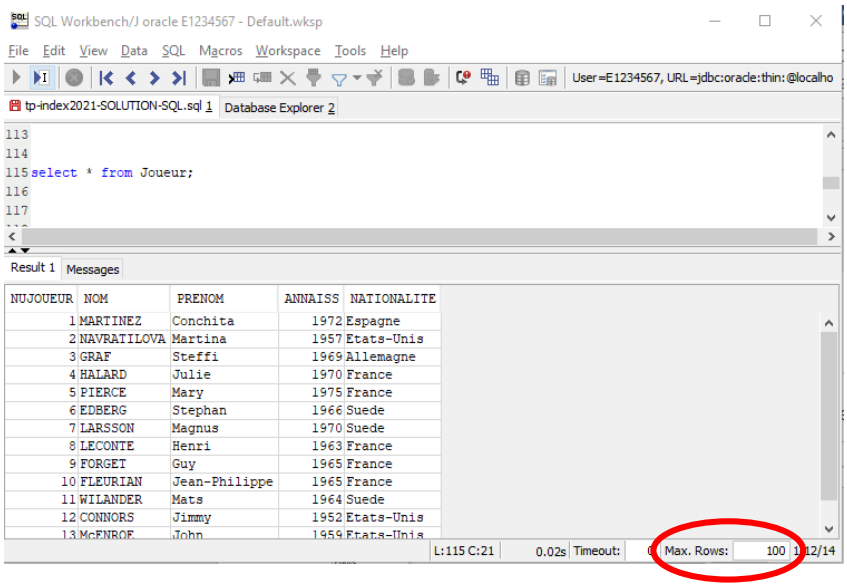
Taille du résultat d'une requête

Limiter la taille du résultat d'une requête à 100 tuples maxi.

Aller dans Tools → Options → Data Display. Puis fixer la valeur de Default Max Rows à 100.



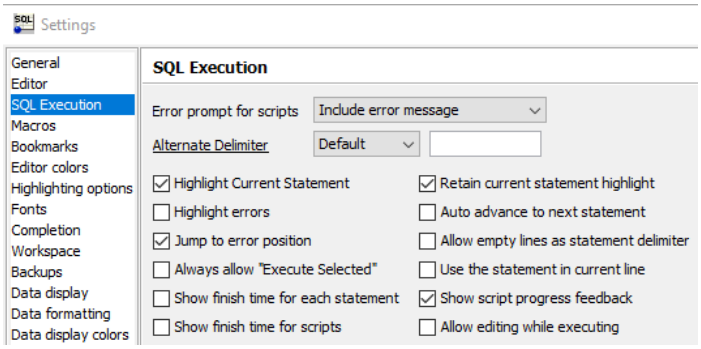
De plus, il faut <u>renseigner</u> le champ **Max. Rows** dans la barre inférieure de la fenêtre principale : saisir **100**



Mettre en évidence la requête en cours d'exécution

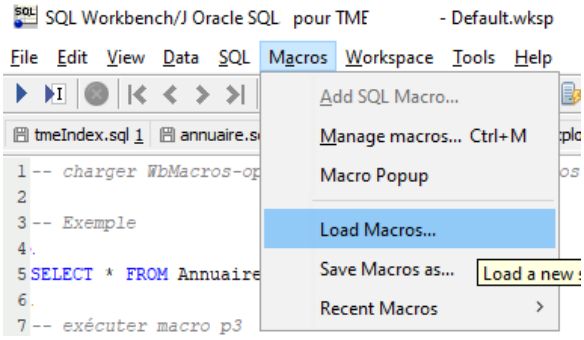
Aller dans Tools → Options → SQL Execution. Puis

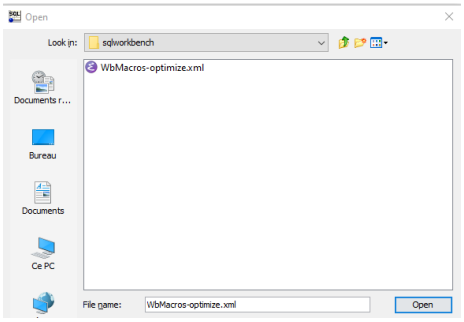
- cocher les cases : ☒Highlight current statement, ☒ Retain current statement, ☒ Jump to error position, ☒ show script progress feedback
- **Décocher** toutes les autres cases



Charger les macros :

Dans ce TP, nous étudions les plans d'exécution des requêtes. Avec SQL Workbench, il est possible d'afficher facilement le plan d'une requête à l'aide des touches F2 et F3 associées à des macros. Pour cela, il suffit de charger le fichier WbMacros-optimize.xml dans le menu Macros → Load Macros. Puis choisir le fichier WbMacros-optimize.xml qui se trouve dans le dossier du TP.



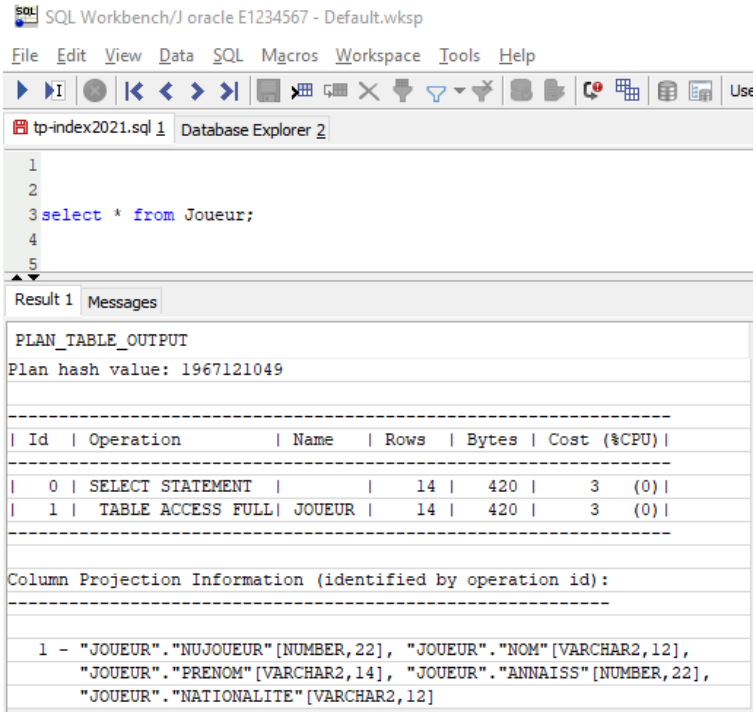


Vérifier le bon fonctionnement des macros :

Saisir une requête simple pour tester l'affichage de son plan, par exemple :

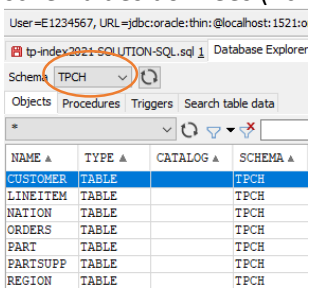
```
Select * from Joueur;
```

Positionner le curseur sur une ligne de la requête et presser la touche **F2**, cela doit afficher le plan suivant :



Données utilisées.

On considère une application de vente de produits. Les données sont issues du benchmark TPC-H. Voir le schéma des données (fichier TPCH_schema.pdf) ou voir le schéma TPCH dans le database Explorer.



Afficher la taille en nombre de pages et la cardinalité de chaque table :

```
select table_name, blocks, num_rows, avg_row_len, global_stats, user_stats
from all_tables
where owner = upper('tpch') and not(table_name like 'S%')
order by num_rows desc;
```

TABLE_NAME	BLOCKS	NUM_ROWS	AVG_ROW_LEN
LINEITEM	109037	6001215	125
ORDERS	24377	1500000	110
PARTSUPP	17237	800000	142
PART	3898	200000	132
CUSTOMER	3520	150000	159
NATION	5	25	102
REGION	5	5	97

Question 1 :

a) Quelle la plus grande table et quelle est sa cardinalité?
b) Sur combien de pages (ou blocs) cette table est-elle stockée?
c) Définir des synonymes vers les tables de la base TPCH

```
create or replace synonym Orders for tpch.Orders;
create or replace synonym Lineitem for tpch.Lineitem;
etc…
```

Question 2 : Définir la table Commande à partir de la table Orders

```
drop table Commande cascade constraints purge;
create table Commande (
        numCde          Number          not null,
        numClient       Number          not null,
        etat            Char(1)         not null,
        prixC            Number(15,2)   not null,
        dateC           Date            not null,
        priorite        Char(15)        not null,
        vendeur         Char(15)        not null,
        commentaire     Varchar2(100)
);
```

Insérer les commandes du 1er trimestre 1992 (pour les mois allant de 1 à 3)

```
insert into Commande (
  select
     o_orderkey as numCde,
     o_custkey  as numClient,
     o_orderstatus as etat,
     o_totalprice as prixC,
     o_orderdate as dateC,
     o_orderpriority as priorite,
```

```
        o_clerk as vendeur,
        o_comment as commentaire
    from Orders
    where extract(year from o_orderdate) = 1992
    and extract(month from o_orderdate) between 1 and 3
);
```

Question 3 : **table AchatProduit**

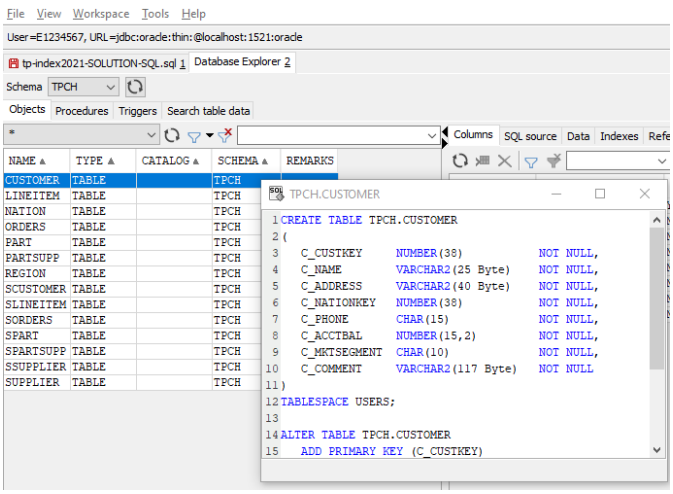
a) Définir la table AchatProduit à partir de la table LineItem

```
create table AchatProduit  (
        numCde              Number        not null,
        numAchat            Number        not null,
        dateAchat           Date          not null,
        prix                Number(15,2)  not null,
        quantite            Number(15,2)  not null,
        numProduit          Number        not null,
        numFournisseur      Number        not null,
        commentaire         varchar2(100)
);
```

b) Insérer les Achats correspondant aux commandes existant dans la table Commande. Indications : le numéro d'acaht est L_LINENUMBER, la date est L_COMMITDATE, le prix est L_EXTENDEDPRICE, le numéo du produit est L_PARTKEY. Ne pas oublier la jointure avec Commande.

Question 4 : Définir vos propres tables à partir des autres tables de cette base.

a) Définir **Client** (numClient, nom, numPays, segment, commentaire) à partir de Customer. Indications : s'inspirer de l'instruction de création de la table Customer. Dans Database Explorer, sélectionner la table Customer, puis clic droit, menu contextuel « create DDL script ». Le segment est C_MKTSEGMENT.



b) Insérer dans Client ceux dont le numéro figure dans la table Commande. Ne pas insérer plusieurs fois un même client qui a passé plusieurs commandes. Vérifier qu'il y a 42427 Clients.

c) Définir **Produit** (numProduit, nom, marque, type, taille, prixDetail) à partir de Part. Indications : la marque est P_BRAND.

d) Insérer les produits correspondant aux produits achetés dans AchatProduit. Ne pas insérer de produits en double, même s'ils correspondent à plusieurs achats. Vérifier qu'il y a 135 775 produits.

Question 5. Statistiques sur les tables

Créer une procédure *analyse(nomtable)* qui collecte des statistiques sur une table : cardinalité, taille et domaine de chaque attribut. Un domaine peut être décrit

- de manière succincte par ses bornes min et max
- ou de manière plus complète par un histogramme.

On remarque que les bornes min max sont un cas particulier d'histogramme de taille 1. Commencer par utiliser les options

*method_opt => 'for all columns size **1**'* pour collecter seulement les bornes min et max

estimate_percent => 100

Puis ultérieurement (après la séance), vous pourrez modifier la procédure *analyse* pour collecter des histogrammes.

```
create or replace procedure analyse(nomTable varchar2) as
  utilisateur varchar2(30);
begin
  select sys_context('USERENV', 'SESSION_USER') into utilisateur from dual;
-- avec histogramme :
--  dbms_stats.gather_table_stats(utilisateur, upper(nomTable));
-- SANS histogramme :
  dbms_stats.gather_table_stats(utilisateur,
                                upper(nomTable),
                                method_opt =>'for all columns size 1',
                                estimate_percent => 100);
end;
/
show error
```

Invoquer cette procédure sur les tables que vous avez créées.

```
exec analyse('Commande');
exec analyse('AchatProduit');
```

Cela affiche : *analyse executed successfully*

Les statistiques sont consultables en interrogeant les vues `user_tables` et `user_tab_columns` comme ceci :

Afficher pour chaque table sa cardinalité et son nombre de pages

```
select table_name,
       num_rows as card,
       blocks as pages,
       global_stats as analyseFaite
from user_tables;
```

Afficher, pour chaque attribut de chaque table, les bornes min et max de son domaine ainsi que le nombre de valeurs distinctes.

```
select table_name, column_name, data_type, sample_size,
       num_distinct as nb_distinct, -- cette valeur est une approximation
       utl_raw.cast_to_number(low_value) as borneInf,
       utl_raw.cast_to_number(high_value) AS borneSup
from user_tab_columns c
where data_type = 'NUMBER'
and table_name in ('ACHATPRODUIT', 'COMMANDE', 'CLIENT', 'PRODUIT')
--
UNION
--
select table_name, column_name, data_type, sample_size,
  num_distinct as nb_distinct, -- cette valeur est une approximation
```

```
  null,
  null
from user_tab_columns c
where data_type like '%CHAR%' or data_type = 'DATE'
and table_name in ('ACHATPRODUIT', 'COMMANDE', 'CLIENT', 'PRODUIT')
order by table_name, column_name;


select table_name, column_name, data_type, num_distinct, sample_size,
       avg_col_len as longueur_moyenne,
       utl_raw.cast_to_number(low_value) as borneInf,
       utl_raw.cast_to_number(high_value) AS borneSup,
       density
from user_tab_columns c
where data_type = 'NUMBER'
order by table_name, column_id;

select table_name, column_name, data_type, num_distinct,
       avg_col_len as longueur_moyenne,
       null as borneInf, null AS borneSup,
       density
from user_tab_columns c
where data_type = 'DATE' or data_type like '%CHAR%'
order by table_name, column_name;
```

Combien y a-t-il de dates et de prix différents pour les commandes ?

Combien y a-t-il de quantités distinctes pour les achats d'un produit ?

Index et requêtes

Question 6 Index simples

Créer un index sur chaque attribut d'une ligne de commande dans la table `AchatProduit`.

```
drop index I_Achat_quantite;
create index I_Achat_quantite on AchatProduit(quantite);

drop index I_Achat_prix;
create index I_Achat_prix on AchatProduit(prix);

drop index I_Achat_numCde;
create index I_Achat_numCde on AchatProduit(numCde);
```

etc…

Afficher les statistiques sur les index en interrogeant la vue `user_indexes`

```
SELECT index_name as nom,
       index_type as type_index,
       blevel as profondeur,
       distinct_keys as nb_valeurs_distinctes,
       num_rows as nb_rowids,
       leaf_blocks as nb_pages_de_rowids,
       uniqueness as unicite,
       clustering_factor as CF
FROM user_indexes;
```

Quel index a la plus grande profondeur ?

Quel index a le plus grand nombre de pages de rowid ?

Question 7. Requêtes avec index simple

Pour chaque requête afficher son plan (touche F2).

Préciser les méthodes d'accès utilisées parmi les méthodes suivantes :

INDEX **RANGE** SCAN, INDEX **UNIQUE** SCAN, INDEX **FULL** SCAN, INDEX **SKIP** SCAN

et quel prédicat est évalué par l'index ;

Préciser si l'opération TABLE ACCESS BY INDEX **ROWID** est nécessaire après avoir lu l'index et quel prédicat est évalué par filtrage (*filter*) pendant cette opération.

a) Quels sont les produits achetés à moins de 2000 euros ?

```
SELECT /*+ index(a I_achat_prix) */
     *
FROM AchatProduit a
WHERE prix < 2000;
```

b) Quels produits sont achetés en grande quantité (quantité supérieure à 40) ?

```
SELECT /*+ index(a I_achat_quantite) */
      *
FROM AchatProduit a
WHERE quantite > 40;
```

c) Lister toutes les valeurs des prix des achats

```
SELECT /*+ index(a I_achat_prix) */
     prix
FROM AchatProduit a;
```

d) Le numCde et numAchat des achats en quantité > 40 et dont le prix < 2000 : proposer une solution avec l'index sur la quantité puis une autre avec l'index sur le prix

…

e) Même requête mais en **combinant** l'utilisation des deux index sur le prix et la quantité

```
SELECT /*+ index_combine(a I_achat_prix I_achat_quantite) */
     numCde, numAchat
FROM AchatProduit a
WHERE prix < 2000 and quantite > 40;
```

f) Même requête mais en demandant l'utilisation **conjointe** de tous les index possibles.

```
SELECT /*+ index_join(a I_achat_prix I_achat_quantite) */

     numCde, numAchat
FROM AchatProduit a
WHERE prix < 2000 and quantite > 40;
```

Comparer les plans d'exécution pour deux situations différentes lorsque les index existants sont :

- I_achat_prix, I_achat_quantite, I_achat_numCde
- I_achat_prix, I_achat_quantite, I_achat_numCde **et I_achat_numAchat**

Dans quelle situation peut-on dire que les index couvrent la requête ?

Question 8. Requêtes avec index composé

Créer un index sur les attributs quantité et prix

```
drop index I_Achat_quantite_prix;
create index I_Achat_quantite_prix on AchatProduit(quantite, prix);
```

Expliquer l'utilisation d l'index pour les requêtes :

a) prix < 1000 and quantite =2
b) quantite = 2
c) prix < 1000
d) prix > 1000

Question 9 : Coût de l'opération INDEX RANGE SCAN

On rappelle qu'on connaît pour un index sa profondeur, son nombre total de rowids et le nombre de pages contenant ces rowids :

```
SELECT index_name as nom,
       index_type as type_index,
       blevel as profondeur,
       num_rows,
       leaf_blocks as nb_pages_de_rowids
FROM user_indexes;
```

Le coût de l'opération INDEX RANGE SCAN représente le nombre de lectures pour traverser l'index et sélectionner les rowids contenus dans les feuilles de l'index. Soit **n** le nombre de rowids sélectionnés dans l'index. La valeur de **n** est indiquée dans le plan de la requête : c'est le champ ROWS de l'operation INDEX RANGE SCAN. On a :

Cost_Index_Range_Scan(**n**) = (`blevel-1)` + **n** * `leaf_blocks` / `num_rows`

Autre formulation équivalente :

Cost_Index_Range_Scan(**predicat**) = (`blevel-1)` + `leaf_blocks` * SF(p)

avec SF(p) étant le facteur de sélectivité du prédicat évalué par l'index

Appliquer cette formule aux requêtes « simples » vues ci-dessus (question 7)

Question 10 : Coût de l'opération TABLE ACCESS BY ROWID

Soit la requête interrogeant les statistiques des indexes :

```
SELECT index_name as nom,
       num_rows,
       clustering_factor as CF
```

```
FROM user_indexes;
```

Le clustering_factor (CF) correspond au nombre de lectures à effectuer pour lire **num_rows** tuples qui ont la même valeur (ou qui sont dans la même plage de valeurs) pour l'attribut indexé. Sa valeur est comprise dans [blocks, num_rows] c'est-à-dire dans [nombre de pages de la table, cardinalité de la table].

CF permet d'estimer le nombre de lectures déclenchées par l'opération TABLE ACCESS BY ROWID. On a :

Cost_table_Access_By_Rowid(**n**) = Cost_Index_Range_Scan(**n**) + **n** * **CF** / **num_rows**

a) Pour quel index a-t-on CF très proche de num_rows ?

Pour quel index a-t-on CF très inférieur à num_rows ?

D'après vous, quelle particularité dans l'insertion des commandes et des achats a pour effet de diminuer le CF d'un index sur l'attribut numCde ?

b) Pour les requêtes proposées ci-dessus et dont le plan utilise un index, appliquez la formule de coût *Cost_table_Access_By_Rowid* et vérifiez que vous retrouvez approximativement le coût estimé par le SGBD.

Question 11 Cardinalité d'une requête

On veut déterminer la cardinalité d'une requête de type : BETWEEN v1 AND v2

Soit la table T et l'attribut ATT qui est de type Number. On dispose des informations suivantes :

```
select utl_raw.cast_to_number(low_value) as borneInf,
       utl_raw.cast_to_number(high_value) as borneSup
from user_tab_columns
where table_name = upper(T)
and column_name = upper(ATT)
```

et

```
select num_rows
from user_tables
where table_name = upper(T)
```

On suppose que v1 et v2 sont inclus dans [borneInf, borneSup]. On a la formule :

cardSelection(T, ATT, V1, V2) = (V2 – V1) / (**borneSup – borneInf**) * **num_rows**

a) Appliquer cette formule aux requêtes « simples » vues ci-dessus, en faisant attention de choisir des valeurs v1 et v2 dans l'intervalle [borneInf, borneSup].

b) Définir la **fonction** cardSelection(T, ATT, v1, v2) qui retourne la cardinalité d'une requête

Select * from T where ATT between V1 and V2

Indication: create or replace function cardSelection(…., …, …, …) return number as …

Question 12 : Requête un attribut unique ou clé primaire

Modifier la table Commande : définir la clé primaire étant l'attribut numCde

```
alter table Commande add constraint cle_commande primary key(numCde);
```

a) Est-ce que cela a déclenché la création de l'index CLE_COMMANDE ? Si oui, afficher les caractéristiques de cet index.

b) Expliquer la méthode INDEX UNIQUE SCAN utilisée dans la requête :

```
SELECT /*+ index(a cle_commande) */
     *
FROM Commande a
WHERE numCde = 200 ;
```

Question 13 : Requêtes avec tri : ORDER BY

Montrer que l'utilisation de l'index est utile pour les requêtes de type : ORDER BY

a) Quel est le coût de l'opération INDEX FULL SCAN dans la requête suivante ?

```
select prix
from AchatProduit
order by prix;
```

b) Quel est le coût de l'opération INDEX **FAST** FULL SCAN dans la requête suivante ?

```
select distinct prix
from AchatProduit
order by prix;
```

c) Pourquoi l'opération *fast full scan* provoque-t-elle moins de lectures que *full scan* ?

Question 14 : Requête avec agrégation min max

a) Expliquer la méthode d'accès INDEX FULL SCAN (MIN/MAX) utilisées dans la requête

```
select min(prix)
from AchatProduit;

select max(prix)
from AchatProduit;
```

b) Est-ce que cette méthode est utilisée pour une requête avec 2 agrégations et pourquoi ?

```
select min(prix), max(prix)
from AchatProduit;
```

Question 15 : Coût d'une opération TABLE ACCESS FULL

Le coût de la lecture séquentielle d'une table est la somme de deux termes :

coût_lecture_sequentielle(T) = **pages(T)** / multiRead + **card(T)** / cpu

avec multiRead = 3,77 et cpu = 58000

Le terme principal représente le nombre de lectures multi-pages, il dépend du nombre de pages.

Le terme minoritaire représente le coût CPU et dépend de la cardinalité.

On rappelle la requête donnant les valeurs de pages et card :

```
select blocks as pages, num_rows as card
from user_tables
```

Exemple, pour AchatProduit on a pages(AchatProduit) = 2119 et card(AchatProduit) = 226736

Le coût TABLE ACCESS FULL de AchatProduit est :

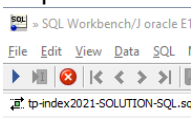
coût_lecture_sequentielle(AchatProduit) = ceil(2119/3,77) + ceil(226736/58000) = 567

Appliquer la formule aux autres tables de votre base.

Divers et questions fréquentes:

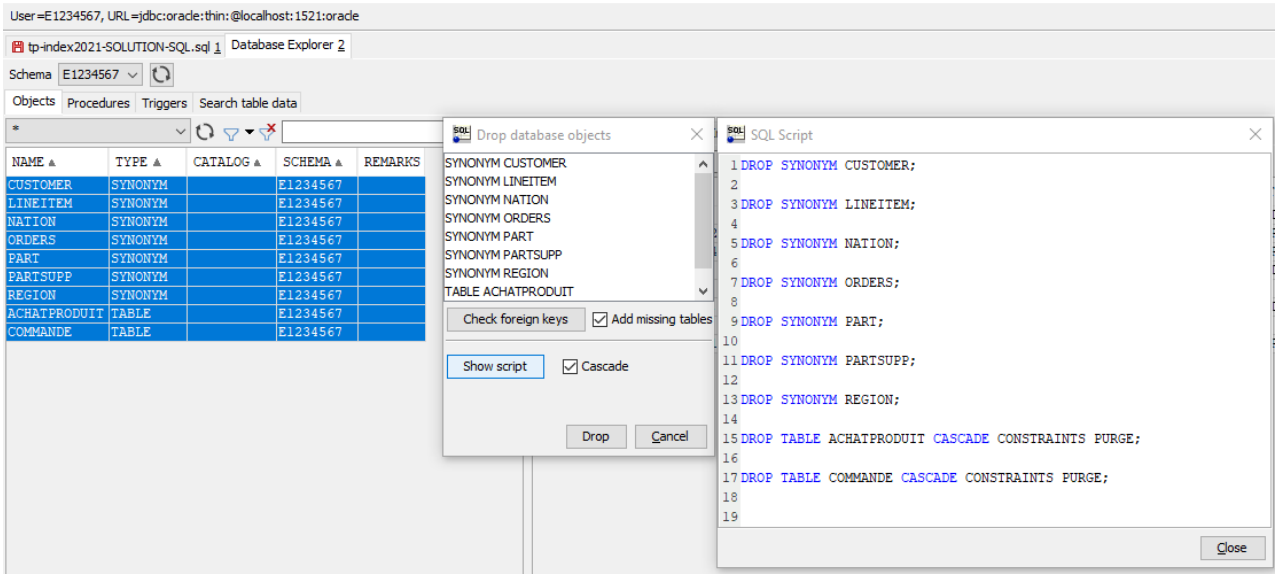
Comment stopper l'exécution d'une requête ?

➔ Cliquer sur le bouton rouge



Comment supprimer tout le contenu de mon compte oracle (table, type, synonyme, …) ?

➔ Dans Database Explorer, sélectionner tous vos objets (sauf les table qui sont en fait des « sous tables » ou nested tables car elles sont supprimées automatiquement quand on supprime la table qui les contient), puis passer la souris sur la zone sélectionnée et bouton droit menu contextuel **drop,** cocher cascade et bouton Drop.Le bouton « Show script » vous permet de générer les instructions drop pour les réutiliser automatiquement.



La touche F2 produit l'erreur "The macro uses current_statement no text is selected" ?

➔ Placer le curseur en début de requête avant d'appuyer sur F2.

*Les directive d'optimisation ne sont **jamais** prises en compte?*

→Vérifier que la case « Remove Comments » n'est **PAS** cochée dans la fenêtre « Connection Profile »