Sujet TD 3

GESTION D'UN PARC INFORMATIQUE

Un responsable de service informatique désire gérer un parc informatique à l'aide d'un SGBD Oracle. Le bâtiment est composé de trois étages. Chaque étage possède son réseau Ethernet. Ces réseaux traversent des salles équipées de postes de travail.

Un poste de travail est une machine sur laquelle sont installés certains logiciels. Quatre catégories de poste de travail sont recensées (station Unix, terminaux X, PC Windows et PC NT). La base de données devra aussi décrire les installations de logiciels.

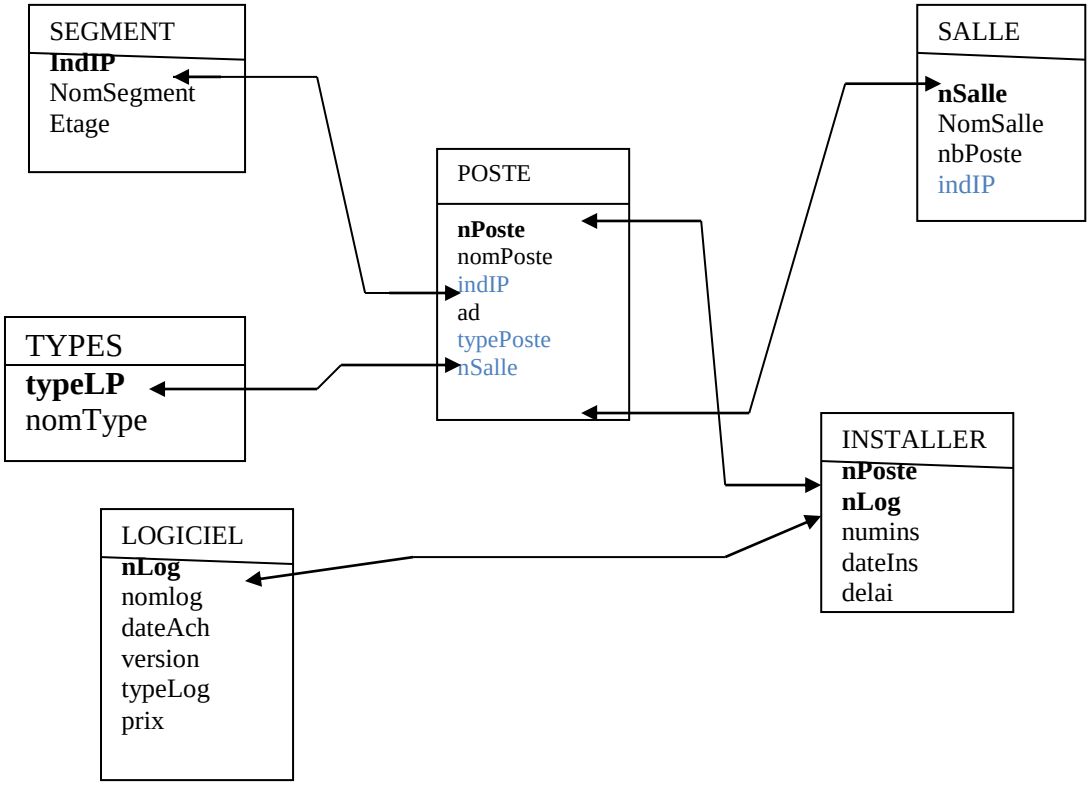
Ci-dessous le dictionnaire des données de la base :

Rubriques	Libellés	Types	Contraintes
IndIP	3 premiers groupes IP (130.120.80)	VARCHAR2(11)	
NomSegment	Nom du segment	VARCHAR2(30)	Non nuls
Etage	Etage du segment	NUMBER(2)	
NSalle	Numéro de salle	VARCHAR2(7)	
NomSalle	Nom de salle	VARCHAR2(30)	Non nuls
NbPoste	Nombre de poste dans la salle	NUMBER(2)	
NPoste	Code du poste de travail	VARCHAR2(7)	
NomPoste	Nom du poste de travail	VARCHAR2(20)	Non nuls
Ad	Dernier groupe de chiffres IP	VARCHAR2(3)	Domaine de valeurs de 0 à 255
TypePoste	Type de poste (Unix,TX,PCWS,PCNT)	VARCHAR2(9)	
DateIns	Date d'installation du logiciel sur le poste	DATE	
Nlog	Code du logiciel sur le poste	VARCHAR2(5)	
NomLog	Nom du logiciel	VARCHAR2(20)	
DateAch	Date d'achat du logiciel	DATE	
Version	Version du logiciel	VARCHAR2(7)	
TypeLog	Type du logicie l(Unix,TX,PCWS,PCNT)	VARCHAR2(9)	
Prix	Prix du logiciel	NUMBER(6,2)	> = 0
NumIns	Numéro séquentiel des installations	NUMBER(5)	
DateIns	Date d'installation du logiciel	DATE	= date du jour par défaut
Delai	Intervalle entre achat et installation	Interval day(5) to second (2)	
TypeLP	Types des logiciels et des postes	VARCHAR2(9)	
NomType	Noms des types (terminaux X, PC win…)	VARCHAR2(20)	

NOTA : ***Prévoir un moyen de vérification à chaque étape***.

1) - Après avoir créé les tables nécessaires au modèle « Parc Informatique », il faut insérer des données dans ces tables. Dans ce script, créer la séquence *sequenceINS* commençant à la valeur 1 ayant un incrément 1, de valeur maximale 10000 et sans cycle.

2) Ci-dessous le modèle des données




```
--**************************--
-- Suppression des tables   --
--**************************--

DROP TABLE Installer purge;
DROP TABLE Logiciel purge;
DROP TABLE Poste purge;
DROP TABLE Types purge;
DROP TABLE Salle purge;
DROP TABLE Segment purge;

--**************************--
-- Création des tables     --
--**************************--

CREATE TABLE Segment
(
indIP            VARCHAR2(11),
nomSegment   VARCHAR2(30)  NOT NULL,
etage            NUMBER(2)
);
ALTER TABLE Segment ADD CONSTRAINT pk_Segment PRIMARY KEY (indIP);

CREATE TABLE Salle
(
nSalle           VARCHAR2(7),
nomSalle       VARCHAR2(30)  NOT NULL,
nbPoste        NUMBER(2),
indIP            VARCHAR2(11)
);
ALTER TABLE Salle ADD CONSTRAINT pk_salle PRIMARY KEY (nSalle);
```

```
CREATE TABLE Poste
(
nPoste          VARCHAR2(7),
nomPoste        VARCHAR2(20) NOT NULL,
indIP           VARCHAR2(11),
ad                      VARCHAR2(3),
typePoste       VARCHAR2(9),
nSalle          VARCHAR2(7)
);
ALTER TABLE Poste ADD CONSTRAINT pk_Poste PRIMARY KEY (nPoste);
ALTER TABLE Poste ADD CONSTRAINT ck_ad CHECK (ad BETWEEN '0' AND '255'));

CREATE TABLE Logiciel
(
nLog    VARCHAR2(5),
nomLog  VARCHAR2(20) NOT NULL,
dateAch DATE,
version VARCHAR2(7),
typeLog VARCHAR2(9),
prix    NUMBER(6,2)
);
ALTER TABLE Logiciel ADD CONSTRAINT pk_Logiciel PRIMARY KEY (nLog);
ALTER TABLE Logiciel ADD CONSTRAINT ck_prix CHECK (prix >= 0);

CREATE TABLE Installer
(
nPoste  VARCHAR2(7),
nLog    VARCHAR2(5),
numIns          NUMBER(5),
dateIns DATE DEFAULT SYSDATE,
delai   INTERVAL DAY(5) TO SECOND(2)
);
ALTER TABLE Installer ADD CONSTRAINT pk_Installer PRIMARY KEY(nPoste,nLog);

CREATE TABLE Types
(
typeLP  VARCHAR2(9),
nomType VARCHAR2(20)
);

ALTER TABLE Types ADD CONSTRAINT pk_types PRIMARY KEY(typeLP);

--**********************************************--
-- Création de la séquence sequenceIns  --
--**********************************************--

CREATE SEQUENCE sequenceIns INCREMENT BY 1 START WITH 1 MAXVALUE 10000
NOCYCLE;

--*****************************--
-- Insertion de données     --
--*****************************--

INSERT INTO Segment VALUES ('130.120.80','Brin RDC',0);
INSERT INTO Segment VALUES ('130.120.81','Brin 1er étage',1);
INSERT INTO Segment VALUES ('130.120.82','Brin 2ème étage',2);
INSERT INTO Segment VALUES ('130.120.83','Brin 3ème étage',3);
INSERT INTO Segment VALUES ('130.120.84','Brin 4ème étage',4);

INSERT INTO Salle VALUES ('s01','Salle 1',3,'130.120.80');
INSERT INTO Salle VALUES ('s02','Salle 2',2,'130.120.80');
INSERT INTO Salle VALUES ('s03','Salle 3',2,'130.120.80');
INSERT INTO Salle VALUES ('s11','Salle 11',2,'130.120.81');
```

```
INSERT INTO Salle VALUES ('s12','Salle 12',1,'130.120.81');
INSERT INTO Salle VALUES ('s21','Salle 21',2,'130.120.82');
INSERT INTO Salle VALUES ('s22','Salle 22',0,'130.120.83');
INSERT INTO Salle VALUES ('s23','Salle 23',0,'130.120.83');
INSERT INTO Salle VALUES ('s23','Salle 23',4,'130.120.84');
INSERT INTO Salle VALUES ('s23','Salle 23',2,'130.120.84');
INSERT INTO Salle VALUES ('s23','Salle 23',3,'130.120.84');
INSERT INTO Salle VALUES ('s30','Salle 24',3'130.120.88');

INSERT INTO Poste VALUES ('p1','Poste 1','130.120.80','01','TX','s01');
INSERT INTO Poste VALUES ('p2','Poste 2','130.120.80','02','UNIX','s01');
INSERT INTO Poste VALUES ('p3','Poste 3','130.120.80','03','TX','s01');
INSERT INTO Poste VALUES ('p4','Poste 4','130.120.80','04','PCWS','s02');
INSERT INTO Poste VALUES ('p5','Poste 5','130.120.80','05','PCWS','s02');
INSERT INTO Poste VALUES ('p6','Poste 6','130.120.80','06','UNIX','s03');
INSERT INTO Poste VALUES ('p7','Poste 7','130.120.80','07','TX','s03');
INSERT INTO Poste VALUES ('p8','Poste 8','130.120.81','01','UNIX','s11');
INSERT INTO Poste VALUES ('p9','Poste 9','130.120.81','02','TX','s11');
INSERT INTO Poste VALUES ('p10','Poste 10','130.120.81','03','UNIX','s12');
INSERT INTO Poste VALUES ('p11','Poste 11','130.120.82','01','PCNT','s21');
INSERT INTO Poste VALUES ('p12','Poste 12','130.120.82','02','PCWS','s21');

INSERT INTO Logiciel VALUES ('log1','Oracle
7',to_date('13/05/1995','DD/MM/YYYY'),'6.2','UNIX',3000);
INSERT INTO Logiciel VALUES ('log2','Oracle
8',to_date('15/09/1999','DD/MM/YYYY'),'8i','UNIX',5600);
INSERT INTO Logiciel VALUES ('log3','SQL
Server',to_date('12/04/1998','DD/MM/YYYY'),'7','PCNT',3000);
INSERT INTO Logiciel VALUES ('log4','Front
Page',to_date('03/06/1997','DD/MM/YYYY'),'5','PCWS',500);
INSERT INTO Logiciel VALUES
('log5','WinDev',to_date('12/05/1997','DD/MM/YYYY'),'5','PCWS',750);
INSERT INTO Logiciel VALUES
('log6','SQL*Net',to_date('02/03/2001','DD/MM/YYYY'),'2.0','UNIX',500);
INSERT INTO Logiciel VALUES ('log7','I. I. S.',to_date('12/04/2002','DD/MM/YYYY'),'2','PCNT',900);
INSERT INTO Logiciel VALUES
('log8','DreamWeaver',to_date('21/09/2003','DD/MM/YYYY'),'2.0','BeOS',1400);

INSERT INTO Types VALUES ('TX', 'Terminal X-Window');
INSERT INTO Types VALUES ('UNIX','Système Unix');
INSERT INTO Types VALUES ('PCNT','PC Windows NT');
INSERT INTO Types VALUES ('PCWS','PC Windows');
INSERT INTO Types VALUES ('NC', 'Network Computer');

INSERT INTO Installer VALUES ('p2', 'log1',
sequenceIns.NEXTVAL,to_date('13/05/2001','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p2', 'log2',
sequenceIns.NEXTVAL,to_date('10/07/1999','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p4', 'log5',
sequenceIns.NEXTVAL,to_date('02/05/2001','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p6', 'log6',
sequenceIns.NEXTVAL,to_date('05/04/2005','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p6', 'log1',
sequenceIns.NEXTVAL,to_date('12/06/2004','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p8', 'log2',
sequenceIns.NEXTVAL,to_date('25/10/2003','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p8', 'log6',
sequenceIns.NEXTVAL,to_date('30/12/2010','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p11','log3',
sequenceIns.NEXTVAL,to_date('09/11/2008','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p12','log4',
sequenceIns.NEXTVAL,to_date('19/02/2009','DD/MM/YYYY'),NULL);
```

```
INSERT INTO Installer VALUES ('p11','log7',
sequenceIns.NEXTVAL,to_date('22/08/2000','DD/MM/YYYY'),NULL);
INSERT INTO Installer VALUES ('p7', 'log7',
sequenceIns.NEXTVAL,to_date('20/01/2002','DD/MM/YYYY'),NULL);

--***************--
-- Requêtes :  --
--***************--
```

1) Noms et adresses IP complète (numéro de segment avec ad) des postes de type TX
(l'opérateur de concaténation est la double barre; exemple : SELECT a||b FROM T ...).

2) Numéros et nom des logiciels d'une version et d'un type entré au clavier (essayez avec '5' et ''PCWS').

3) Affichage des postes (nom_p, adresse_ip, et n_salle) de type 'UNIX' et 'TX'.

4) Même question pour les postes appartenant à un type entré au clavier.

5) Pour chaque poste, le nombre de logiciels installés (en utilisant la table INSTALLER)
Affichage sous la forme :

```
-------------------------
Numéro poste    Nombre de logiciels
----------------      ----------------------
p1                              2
....
```

5) Dans chaque salle, le nombre de postes installés (à partir de la table POSTE)

6) Pour chaque logiciel, le nombre d'installations sur des postes différents.

7) Nombre de logiciels appartenant à un type entré au clavier avec l'affichage suivant (en prenant 'UNIX' par exemple) :

```
Nombre de logiciels pour UNIX
---------------------------------
                3
```

8) Date d''achat du logiciel le plus récent.

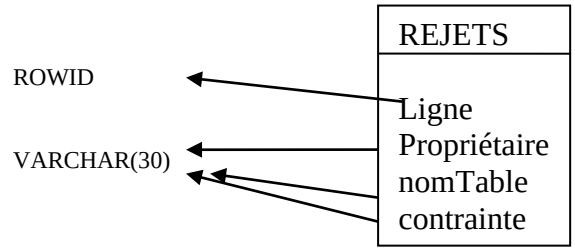
9) Nom du logiciel ayant la date d''achat la plus récente.

```
--***************************--
-- Contrôles et triggers : --
--***************************--
```

1) Mettre en place un contrôle afin de bloquer l''insertion et la mise à jour de la colonne AD de la table Poste
- CONSTRAINT ck_ad pour que le contenu de la colonne AD soit compris entre ('0' et '255');
Supprimer ce contrôle et remplacer le par un trigger qui fait la même chose

2) Mettre en place un contrôle afin de bloquer l''insertion et la mise à jour de la colonne PRIX de la table Logiciel
- CONSTRAINT ck_prix pour que le contenu de la colonne PRIX soit (>=0);
Supprimer ce contrôle et remplacer le par un trigger qui fait la même chose

3) – Traitements des rejets avec la structure suivante (ne pas mettre de clé primaire) :



Cette table permettra de retrouver les enregistrements qui ne vérifient pas des contraintes lors de leur création ou leur réactivation.
Ajouter les contraintes clés étrangères entre les tables Salle et Segment et entre Logiciel et Types (en gras dans le schéma suivant).
Utiliser la clause EXCEPTIONS INTO pour récupérer des informations sur les erreurs.

```
--************************************--
-- Suppression des contraintes   --
--************************************--
ALTER TABLE Poste DROP CONSTRAINT fk_Poste_indIP_Segment;
ALTER TABLE Poste DROP CONSTRAINT fk_Poste_nSalle_Salle;
ALTER TABLE Poste DROP CONSTRAINT fk_Poste_typePoste_Types;
ALTER TABLE Installer DROP CONSTRAINT fk_Installer_nPoste_Poste;
ALTER TABLE Installer DROP CONSTRAINT fk_Installer_nLog_Logiciel;
ALTER TABLE Logiciel DROP CONSTRAINT fk_Logiciel_typeLog_Types;
ALTER TABLE Salle DROP CONSTRAINT fk_Salle_indIP_Segment;

--**************************--
-- Ajout de contraintes                --
--**************************--
ALTER TABLE Poste ADD CONSTRAINT fk_Poste_indIP_Segment FOREIGN KEY(indIP)
REFERENCES Segment(indIP);

ALTER TABLE Poste ADD CONSTRAINT fk_Poste_nSalle_Salle FOREIGN KEY(nSalle)
REFERENCES Salle(nSalle);

ALTER TABLE Poste ADD CONSTRAINT fk_Poste_typePoste_Types FOREIGN KEY(typePoste)
REFERENCES Types(typeLP);

ALTER TABLE Installer ADD CONSTRAINT fk_Installer_nPoste_Poste FOREIGN KEY(nPoste)
REFERENCES Poste(nPoste);

ALTER TABLE Installer ADD CONSTRAINT fk_Installer_nLog_Logiciel FOREIGN KEY(nLog)
REFERENCES Logiciel(nLog);

--ALTER TABLE Logiciel ADD CONSTRAINT fk_Logiciel_typeLog_Types FOREIGN KEY(typeLog)
REFERENCES Types(typeLP);

--ALTER TABLE Salle ADD CONSTRAINT fk_Salle_indIP_Segment FOREIGN KEY(indIP)
REFERENCES Segment(indIP);

--*********************************************************************************--
-- Traitement des rejets lors de la création des contraints d'intégrité référentielles  --
--*********************************************************************************--
• On remarque que les deux contraintes d''intégrité fk_Logiciel_typeLog_Types et
fk_Salle_indIP_Segment
  posent problème lors de leurs créations.
```

• Afin de résoudre ce problème, on va procéder au traitement des enregistrement rejetés

• On va créer une table exceptions qui servira pour stocker les enregistrements qui non supportés par la contrainte

```
CREATE TABLE exceptions
(
row_id          rowid,
owner           VARCHAR2(30),
table_name      VARCHAR2(30),
constraint      VARCHAR2(30)
);

--ALTER TABLE Logiciel ADD CONSTRAINT fk_Logiciel_typeLog_Types FOREIGN KEY(typeLog)
REFERENCES Types(typeLP);

--ALTER TABLE Salle ADD CONSTRAINT fk_Salle_indIP_Segment FOREIGN KEY(indIP)
REFERENCES Segment(indIP);
```

• On va dirigera les enregistrements erronés ou non cohérents dans la table exceptions

```
ALTER TABLE Logiciel ADD CONSTRAINT fk_Logiciel_typeLog_Types FOREIGN KEY(typeLog)
REFERENCES Types(typeLP) EXCEPTIONS INTO exceptions;

ALTER TABLE Salle ADD CONSTRAINT fk_Salle_indIP_Segment FOREIGN KEY(indIP)
REFERENCES Segment(indIP) EXCEPTIONS INTO exceptions;

SELECT * from exceptions;

select logic.*
from logiciel logic, exceptions excp
where excp.row_id = logic.rowid

select sal.*
from salle sal, exceptions excp
where excp.row_id = sal.rowid

--*********************************************************************************************--
-- Les étapes pour résoudre les problèmes rencontrés lors de création d'une contrainte  : --
--*********************************************************************************************--
```

• On supprime les enregistrements de la table exceptions.

```
DELETE FROM exceptions;
```

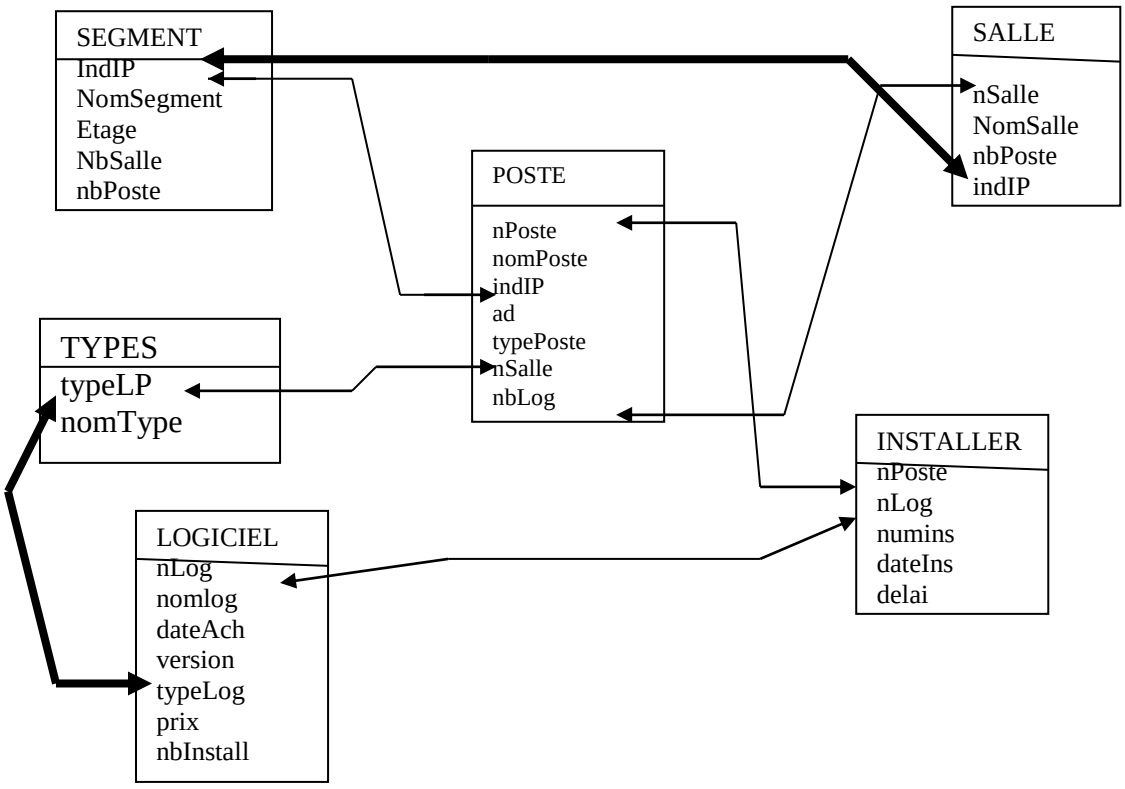
• Pour résoudre le problème de la contrainte fk_Salle_indIP_Segment
• On supprime les enregistrements de la table Salle qui ne respectent pas la contrainte.

```
DELETE FROM Salle WHERE indIP NOT IN (SELECT indIP FROM Segment);
ou
DELETE FROM Salle WHERE indIP = '130.120.88';
```

• Pour résoudre le problème de la contrainte fk_Logiciel_typeLog_Types
• On ajoute l''enregistrement manquant dans la table Types (‘BeOS’, ‘Système Be’)

```
INSERT INTO Types VALUES ('BeOS','Système Be');
```

• **En conclusion :**
L''ajout des deux contraintes d''intégrité référentielles ne renvoie plus d’erreur et la table "exceptions" reste vide.



--**********--
-- Affichage d'un plan d'exécution --
--**********--

SQL> SET AUTOTRACE ON

sql> SET AUTOTRACE TRACE EXPLAIN ON

```
SELECT s.nomSegment
FROM Segment s, Poste p
WHERE s.indIP = p.indIP
AND p.typePoste = 'TX'
GROUP BY s.nomSegment
HAVING COUNT(*)=3;
```

```
SQL> explain plan for
        SELECT s.nomSegment
        FROM Segment s, Poste p
        WHERE s.indIP = p.indIP
        AND p.typePoste = 'TX'
        GROUP BY s.nomSegment
        HAVING COUNT(*)=3;
```

Explicité. ou Explained.

```
SQL> SELECT * FROM TABLE (DBMS_XPLAN.DISPLAY);
```

```
SQL> EXPLAIN PLAN SET STATEMENT_ID = 'SQL_REQUETE_1' FOR
        SELECT s.nomSegment
        FROM Segment s, Poste p
        WHERE s.indIP = p.indIP
        AND p.typePoste = 'TX'
        GROUP BY s.nomSegment
        HAVING COUNT(*)=3;
```

```
SQL> SELECT * FROM TABLE (DBMS_XPLAN.DISPLAY);

/*****************************************************************************/
/* Create EXPLAIN_OUT procedure. User must pass STATEMENT_ID to */
/* to procedure, to uniquely identify statement.                */
/*****************************************************************************/
CREATE OR REPLACE PROCEDURE explain_out
  (statement_id IN VARCHAR2) AS
  -- Retrieve information from PLAN_TABLE into cursor EXPLAIN_ROWS.
  CURSOR explain_rows IS
    SELECT level, id, position, operation, options,
      object_name
    FROM plan_table
    WHERE statement_id = explain_out.statement_id
    CONNECT BY PRIOR id = parent_id
      AND statement_id = explain_out.statement_id
    START WITH id = 0
     ORDER BY id;
BEGIN
  -- Loop through information retrieved from PLAN_TABLE:
  FOR line IN explain_rows LOOP
    -- At start of output, include heading with estimated cost.
    IF line.id = 0 THEN
      DBMS_OUTPUT.PUT_LINE ('Plan for statement '
        || statement_id
        || ', estimated cost = ' || line.position);
    END IF;
    -- Output formatted information. LEVEL determines indention level.
    DBMS_OUTPUT.PUT_LINE (lpad(' ',2*(line.level-1)) ||
      line.operation || ' ' || line.options || ' ' ||
      line.object_name);
  END LOOP;
END;
/

--************************************--
-- Calcul de statistics des tables  --
--************************************--
execute dbms_stats.gather_table_stats(ownname => 'ORA_USER', tabname => 'INSTALLER',
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt => 'FOR ALL COLUMNS
SIZE AUTO');
execute dbms_stats.gather_table_stats(ownname => 'ORA_USER', tabname => 'LOGICIEL',
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt => 'FOR ALL COLUMNS
SIZE AUTO');
execute dbms_stats.gather_table_stats(ownname => 'ORA_USER', tabname => 'POSTE',
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt => 'FOR ALL COLUMNS
SIZE AUTO');
execute dbms_stats.gather_table_stats(ownname => 'ORA_USER', tabname => 'TYPES',
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt => 'FOR ALL COLUMNS
SIZE AUTO');
execute dbms_stats.gather_table_stats(ownname => 'ORA_USER', tabname => 'SALLE',
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt => 'FOR ALL COLUMNS
SIZE AUTO');
execute dbms_stats.gather_table_stats(ownname => 'ORA_USER', tabname => 'SEGMENT',
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt => 'FOR ALL COLUMNS
SIZE AUTO');

--************************************--
-- Calcul de statistics d'un indexe --
--************************************--
execute dbms_stats.gather_index_stats(ownname => 'ORA_USER', indname => 'nom_indexe',
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE);
```

```
--*****************************************************--
-- Le script de destruction des tables devient :  --
--*****************************************************--

--************************************--
-- Suppression des contraintes   --
--************************************--
--ALTER TABLE Poste DROP CONSTRAINT fk_Poste_indIP_Segment;
--ALTER TABLE Poste DROP CONSTRAINT fk_Poste_nSalle_Salle;
--ALTER TABLE Poste DROP CONSTRAINT fk_Poste_typePoste_Types;
--ALTER TABLE Installer DROP CONSTRAINT fk_Installer_nPoste_Poste;
--ALTER TABLE Installer DROP CONSTRAINT fk_Installer_nLog_Logiciel;
--ALTER TABLE Logiciel DROP CONSTRAINT fk_Logiciel_typeLog_Types;
--ALTER TABLE Salle DROP CONSTRAINT fk_Salle_indIP_Segment;

--DROP TABLE Installer purge;
--DROP TABLE Logiciel purge;
--DROP TABLE Poste purge;
--DROP TABLE Types purge;
--DROP TABLE Salle purge;
--DROP TABLE Segment purge;
```