BDD_MYSQL

COMPTE RENDU

12/06/2020

Réalisé par :

- Salma Belassiri

Encadré par :

- Abdellatif Tijani

Introduction :

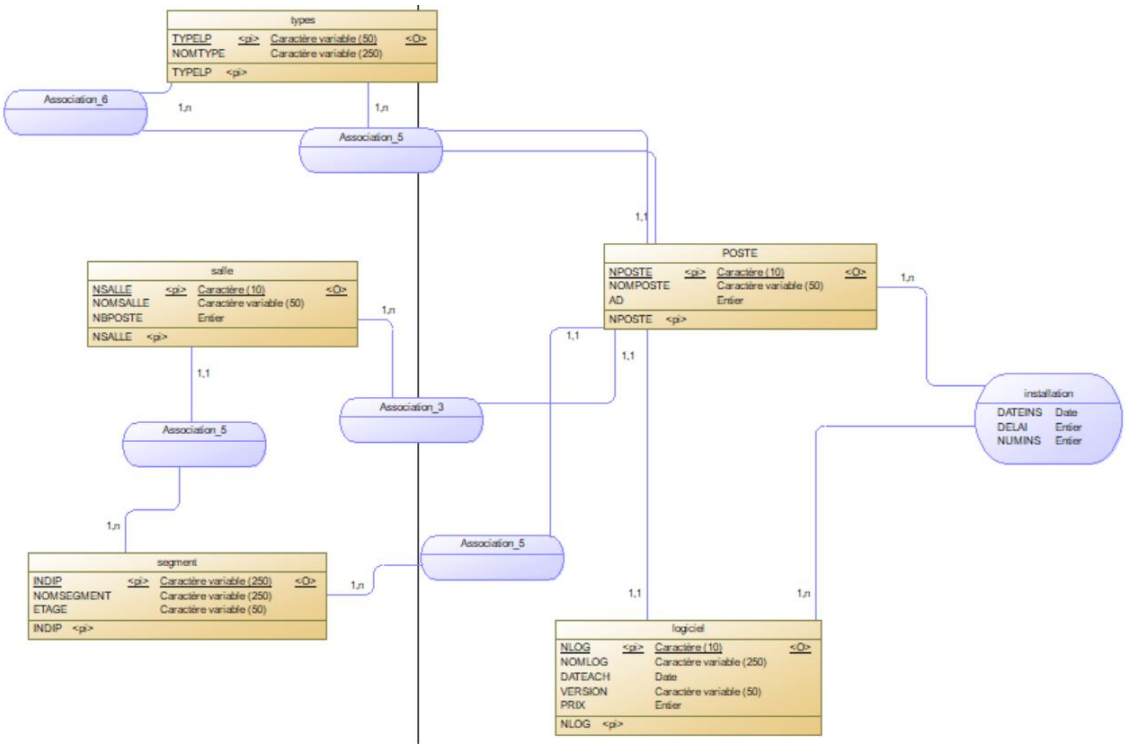
Dans ce rapport, je passerai en revue ce que j'ai appris aujourd'hui. C'était notre objectif
familiariser avec les outils de la gestion d'une base de données ainsi de la gérer en utilisant le langage SQL .
base de données est une suite de programmes qui manipule la structure de la base de données et dirige l'accès aux données qui y sont stockées. Une base de données est composée d'une collection de fichiers. Il est donc sûr de le dire
Un outil précieux que nous, étudiants YOUODODE, devons maîtriser si nous voulons
Un jour, des développeurs de logiciels chevronnés nous ont appelés..

Modélisation de la base de données

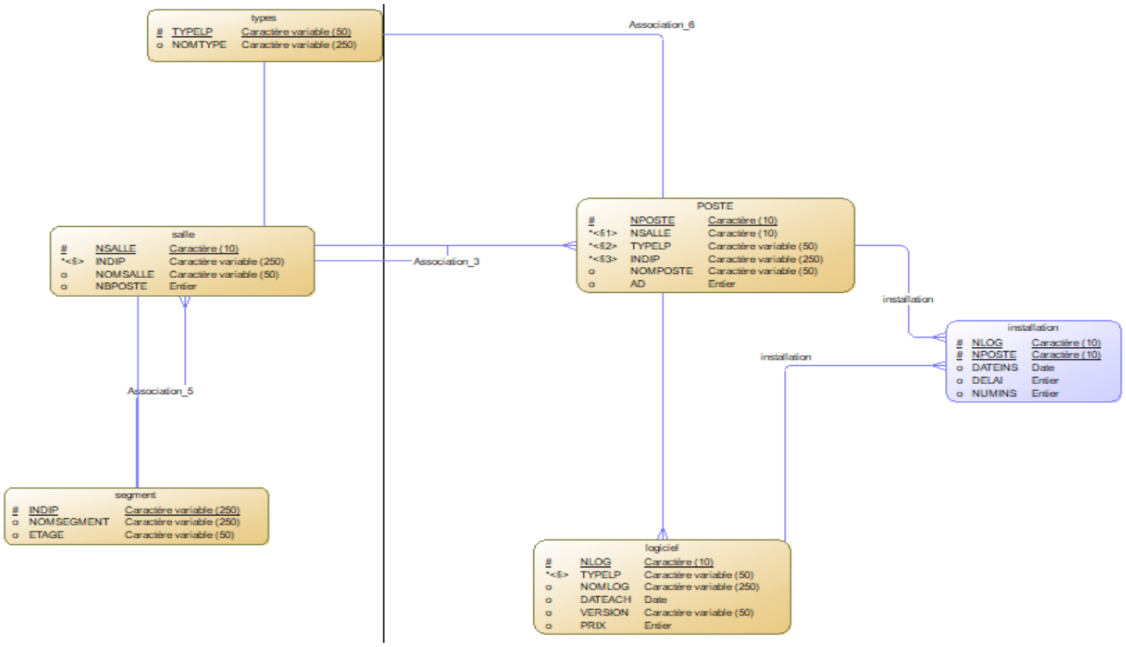
Dictionnaire de donnée :

Nom Symbolique	Description	TYPE	Commentaire	Contraintes ou règles
TYPELP	Diminutif nom OS	CHAR(4)		
NOMTYPE	NOM OS	CHAR()		
NPOSTE	NUMERO DE POSTE	VARCHAR()	sequentiel	AUTOMATIQUE & OBLIGATOIRE
NOMPOSTE	Nom du poste	VARCHAR()	sequentiel	unique
INDIP	Index IP	numeric	Format adresse IP	Souforme de ip
AD	adresse	integre		00A255
TYPEPOSTE	Diminutif nom OS	CHAR(4)		
NSALLE	Numero de salle	VARCHAR()		
NOMSEGMENT	Niveau des étages	VARCHAR()		Soi RDC ou 1er étage ou 2eme étage
NBPOSTE	Nombre de poste	Integre		
NLOG	Numero logiciel	VARCHAR()	sequentiel	AUTOMATIQUE & OBLIGATOIRE
NOMLOG	Nom logiciel	VARCHAR()	Nom de logiciel	unique
DATEACH	Date de sortie du logiciel	DATE		Format date
VERSION	NUM DE VERSION	VARCHAR()	Version des logiciels	
TYPELOG	Compatibilité de logiciel	CHAR()	Type des logiciel	
PRIX	Prix de logiciel	integre	price	Prix = >0
NUMINS	Nombre d'installation	integre	séquentiel	AUTOMATIQUE & OBLIGATOIRE
DATEINS	Date d'installation	date	Current date	Format date

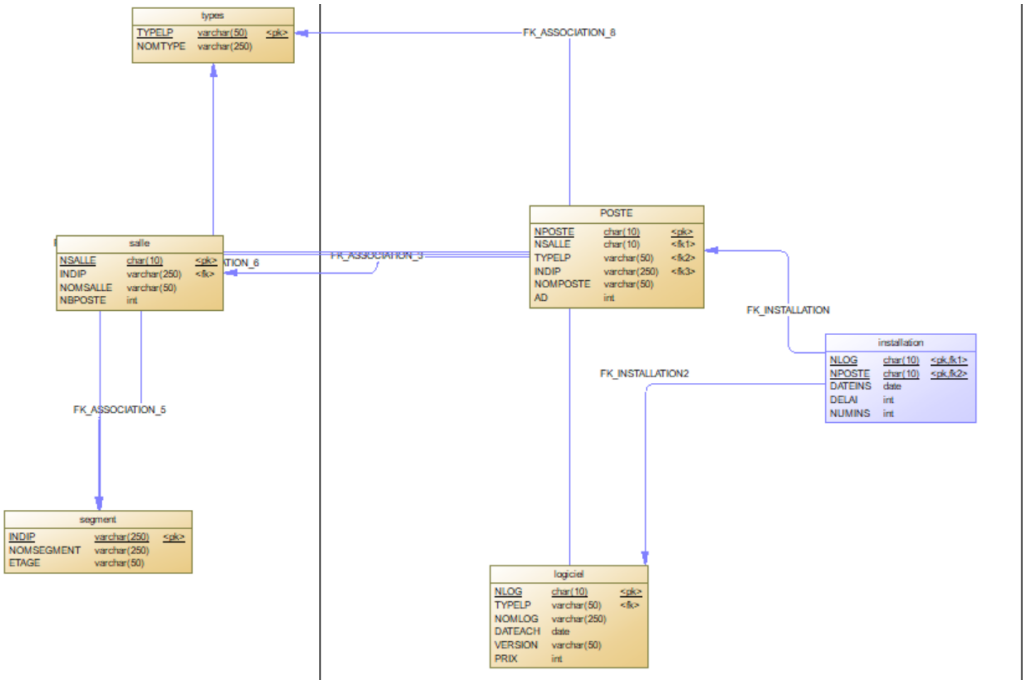
MCD


types
TYPELP <pi> Caractère variable (50) <O>
NOMTYPE Caractère variable (250)
TYPELP <pi>
Association_6
1,n
1,n
Association_5
1,1
POSTE
NPOSTE <pi> Caractère (10) <O>
NOMPOSTE Caractère variable (50)
AD Entier
NPOSTE <pi>
1,n
salle
NSALLE <pi> Caractère (10) <O>
NOMSALLE Caractère variable (50)
NBPOSTE Entier
NSALLE <pi>
1,n
1,1
1,1
1,1
Association_3
installation
DATEINS Date
DELAI Entier
NUMINS Entier
1,1
Association_5
1,n
Association_5
segment
INDIP <pi> Caractère variable (250) <O>
NOMSEGMENT Caractère variable (250)
ETAGE Caractère variable (50)
INDIP <pi>
1,n
1,1
1,n
logiciel
NLOG <pi> Caractère (10) <O>
NOMLOG Caractère variable (250)
DATEACH Date
VERSION Caractère variable (50)
PRIX Entier
NLOG <pi>

MLD


types
TYPELP Caractère variable (50)
o NOMTYPE Caractère variable (250)
Association_6
salle
NSALLE Caractère (10)
*<&> INDIP Caractère variable (250)
o NOMSALLE Caractère variable (50)
o NBPOSTE Entier
Association_3
POSTE
NPOSTE Caractère (10)
*<&1> NSALLE Caractère (10)
*<&2> TYPELP Caractère variable (50)
*<&3> INDIP Caractère variable (250)
o NOMPOSTE Caractère variable (50)
o AD Entier
installation
installation
installation
NLOG Caractère (10)
NPOSTE Caractère (10)
o DATEINS Date
o DELAI Entier
o NUMINS Entier
Association_5
segment
INDIP Caractère variable (250)
o NOMSEGMENT Caractère variable (250)
o ETAGE Caractère variable (50)
logiciel
NLOG Caractère (10)
*<&> TYPELP Caractère variable (50)
o NOMLOG Caractère variable (250)
o DATEACH Date
o VERSION Caractère variable (50)
o PRIX Entier

MPD



SQL

1) ***Création des tables :***

Table : typelp

```
create table typelp(
typelp varchar(9) primary key,
nomtype varchar(20))ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table : poste

```sql
CREATE TABLE poste (
  nposte varchar(7) NOT NULL,
  nom_poste varchar(20) NOT NULL,
  indip varchar(11) DEFAULT NULL,
  ad varchar(3) DEFAULT NULL,
  type_poste varchar(9) DEFAULT NULL,
  nsalle varchar(7) DEFAULT NULL,
  PRIMARY KEY (nposte)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table : segment

```sql
CREATE TABLE segment (
  indip varchar(11) NOT NULL,
  nom_segment varchar(20) DEFAULT NULL,
  etage tinyint(1) DEFAULT NULL,
  PRIMARY KEY (indip)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table : salle

```sql
CREATE TABLE salle (
  nsalle varchar(7) NOT NULL,
  nom_salle varchar(20) NOT NULL,
  nbposte tinyint(2) DEFAULT NULL,
  indip varchar(11) DEFAULT NULL,
  PRIMARY KEY (nsalle)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table : logiciel

```
CREATE TABLE logiciel (
  nlog varchar(5) NOT NULL,
  nomlog varchar(20) NOT NULL,
  dateach datetime DEFAULT NULL,
  version varchar(7) DEFAULT NULL,
  typelog varchar(9) DEFAULT NULL,
  prix decimal(6,2) DEFAULT NULL,
  PRIMARY KEY (nlog)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table : installation

```
CREATE TABLE installation (
  nposte varchar(7) DEFAULT NULL,
  nlog varchar(5) DEFAULT NULL,
  num_ins int(5) NOT NULL AUTO_INCREMENT,
  date_ins timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  delai decimal(8,2) DEFAULT NULL,
  PRIMARY KEY (num_ins)
) ENGINE=MyISAM AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;
```

2) ***INSERTION***

- Insertion table installation

```
55
56 • INSERT INTO installation VALUES ('p2','log1',1,'2003-05-15 00:00:00',NULL),
57   ('p2','log2',2,'2003-09-16 23:00:00',NULL),
58   ('p4','log5',3,NULL,NULL),
59   ('p6','log6',4,'2003-05-20 00:00:00',NULL),
60   ('p6','log1',5,'2003-05-20 00:00:00',NULL),
61   ('p8','log2',6,'2003-05-19 00:00:00',NULL),
62   ('p8','log6',7,'2003-05-20 00:00:00',NULL),
63   ('p11','log3',8,'2003-04-19 23:00:00',NULL),
64   ('p12','log4',9,'2003-04-19 23:00:00',NULL),
65   ('p11','log7',10,'2003-04-19 23:00:00',NULL),
66   ('p7','log7',11,'2002-03-31 23:00:00',NULL);
67 • select*from installation
68
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

nposte	nlog	num_ins	date_ins	delai
p2	log1	1	2003-05-15 00:00:00	NULL
p2	log2	2	2003-09-16 23:00:00	NULL
p4	log5	3	NULL	NULL
p6	log6	4	2003-05-20 00:00:00	NULL
p6	log1	5	2003-05-20 00:00:00	NULL

- Insertion table logiciel

```
68
69  INSERT INTO logiciel VALUES ('log1','Oracle 6','1995-05-13 00:00:00','6.2','UNIX',3000.00),
70  ('log2','Oracle 8','1999-09-15 00:00:00','8i','UNIX',5600.00),
71  ('log3','SQL Server','1998-04-12 00:00:00','7','PCNT',3000.00),
72  ('log4','Front Page','1997-06-03 00:00:00','5','PCWS',500.00),
73  ('log5','WinDev','1997-05-12 00:00:00','5','PCWS',750.00),
74  ('log6','SQL*Net',NULL,'2.0','UNIX',500.00),
75  ('log7','I. I. S.','2002-04-12 00:00:00','2','PCNT',900.00),
76  ('log8','DreamWeaver','2003-09-21 00:00:00','2.0','BeOS',1400.00);
77  select*from logiciel
78
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

nlog	nomlog	dateach	version	typelog	prix	nbinstall
log1	Oracle 6	1995-05-13 00:00:00	6.2	UNIX	3000.00	0
log2	Oracle 8	1999-09-15 00:00:00	8i	UNIX	5600.00	0
log3	SQL Server	1998-04-12 00:00:00	7	PCNT	3000.00	0
log4	Front Page	1997-06-03 00:00:00	5	PCWS	500.00	0
log5	WinDev	1997-05-12 00:00:00	5	PCWS	750.00	0

- Insertion table poste

```
79  INSERT INTO poste VALUES ('p1','Poste 1','130.120.80','01','TX','s01'),
80  ('p2','Poste 2','130.120.80','02','UNIX','s01'),
81  ('p3','Poste 3','130.120.80','03','TX','s01'),
82  ('p4','Poste 4','130.120.80','04','PCWS','s02'),
83  ('p5','Poste 5','130.120.80','05','PCWS','s02'),
84  ('p6','Poste 6','130.120.80','06','UNIX','s03'),
85  ('p7','Poste 7','130.120.80','07','TX','s03'),
86  ('p8','Poste 8','130.120.81','01','UNIX','s11'),
87  ('p9','Poste 9','130.120.81','02','TX','s11'),
88  ('p10','Poste 10','130.120.81','03','UNIX','s12'),
89  ('p11','Poste 11','130.120.82','01','PCNT','s21'),
90  ('p12','Poste 12','130.120.82','02','PCWS','s21');
91  select*from poste
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

nposte	nom_poste	indip	ad	type_poste	nsalle	nblog
p1	Poste 1	130.120.80	01	TX	s01	0
p2	Poste 2	130.120.80	02	UNIX	s01	0
p3	Poste 3	130.120.80	03	TX	s01	0
p4	Poste 4	130.120.80	04	PCWS	s02	0
p5	Poste 5	130.120.80	05	PCWS	s02	0

- Insertion table salle

```
INSERT INTO salle  VALUES ('s01','Salle 1',3,'130.120.80'),
('s02','Salle 2',2,'130.120.80'),
('s03','Salle 3',2,'130.120.80'),
('s11','Salle 11',2,'130.120.81'),
('s12','Salle 12',1,'130.120.81'),
('s21','Salle 21',2,'130.120.82'),
('s22','Salle 22',0,'130.120.83'),
('s23','Salle 23',0,'130.120.83');
select*from salle
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

nsalle	nom_salle	nbposte	indip
s01	Salle 1	3	130.120.80
s02	Salle 2	2	130.120.80
s03	Salle 3	2	130.120.80
s11	Salle 11	2	130.120.81
s12	Salle 12	1	130.120.81

- Insertion table segment

```
INSERT INTO segment VALUES ('130.120.80','Brin RDC',NULL),
('130.120.81','Brin 1er étage',NULL),
('130.120.82','Brin 2ème étage',NULL);
select*from segment

INSERT INTO typelp VALUES ('TX','Terminal X-Window'),
```

Result Grid | Filter Rows: | Edit: | Export/Import:

indip	nom_segment	etage	nbsalle	nbposte
130.120.80	Brin RDC	0	0	0
130.120.81	Brin 1erétage	1	0	0
130.120.82	Brin 2ème étage	2	0	0
NULL	NULL	NULL	NULL	NULL

- Insertion table typelp

```
110  INSERT INTO typelp VALUES ('TX','Terminal X-Window'),
111  ('UNIX','Système Unix'),
112  ('PCNT','PC Windows NT'),
113  ('PCWS','PC Windows'),
114  ('NC','Network Computer');
115  select*from typelp
116
```

Result Grid | Filter Rows: | Edit: | Export/Import:

typelp	nomtype
TX	Terminal X-Window
UNIX	Système Unix
PCNT	PC Windows NT
PCWS	PC Windows
NC	Network Computer
BeOS	System be

3) ***Modification des données :***

- Modification de nombre d'étage selon Etage = 0 pour indIP = '130.120.80',Etage = 1 pour indIP = '130.120.81',Etage = 2 pour indIP = '130.120.82',et afficher la table

```
116
117  update segment set etage=0 where  indip='130.120.80';
118  update segment set etage=1 where  indip='130.120.81';
119  update segment set etage=2 where  indip='130.120.82';
120  select*from segment
121
```

Result Grid | Filter Rows: | Edit: | Export/Import: | W

indip	nom_segment	etage	nbsalle	nbposte
130.120.80	Brin RDC	0	0	0
130.120.81	Brin 1erétage	1	0	0
130.120.82	Brin 2ème étage	2	0	0
NULL	NULL	NULL	NULL	NULL

- Modification de prix du type de logiciel 'PCNT' et afficher numlogiciel, typelogiciel, et prix.

```
121
122  update logiciel set prix=prix*0.9 where typelog='PNCT'
123  select nlog,typelog,prix from logiciel where typelog='PCNT';
124
```

Result Grid | Filter Rows: | Edit: | Export/Import:

nlog	typelog	prix
log3	PCNT	3000.00
log7	PCNT	900.00
NULL	NULL	NULL

4) Ajout des colonnes :

- Ajout les colonnes nbSalle et nbPoste de type int dans la table Segment

```
124
125 •   alter table segment
126     add(nbsalle int(2) default 0, nbposte int(2) default 0);
127 •   select *from segment
128
```

Result Grid | Filter Rows: | Edit: | Export/Import:

indip	nom_segment	etage	nbsalle	nbposte
130.120.80	Brin RDC	0	0	0
130.120.81	Brin 1erétage	1	0	0
130.120.82	Brin 2ème étage	2	0	0
NULL	NULL	NULL	NULL	NULL

- Ajout la colonne nbInstall de type int dans la table Logiciel

```
128
129 ⊠   alter table logiciel
130     add nbinstall int(2) default 0;
131 •   select *from logiciel
132
133
```

Result Grid | Filter Rows: | Edit: | Export/Import:

nlog	nomlog	dateach	version	typelog	prix	nbinstall
log1	Oracle 6	1995-05-13 00:00:00	6.2	UNIX	3000.00	0
log2	Oracle 8	1999-09-15 00:00:00	8i	UNIX	5600.00	0
log3	SQL Server	1998-04-12 00:00:00	7	PCNT	3000.00	0
log4	Front Page	1997-06-03 00:00:00	5	PCWS	500.00	0
log5	WinDev	1997-05-12 00:00:00	5	PCWS	750.00	0
log6	SQL *Net	NULL	2.0	UNIX	500.00	0

ogiciel 58 ×

- Ajout la colonne nbLog de type int dans la table POSTE

```
alter table poste
add nblog int(2) default 0;
select *from poste;
```

Result Grid | Filter Rows: | Edit: | Export/Imp

nposte	nom_poste	indip	ad	type_poste	nsalle	nblog
p1	Poste 1	130.120.80	01	TX	s01	0
p2	Poste 2	130.120.80	02	UNIX	s01	0
p3	Poste 3	130.120.80	03	TX	s01	0
p4	Poste 4	130.120.80	04	PCWS	s02	0
p5	Poste 5	130.120.80	05	PCWS	s02	0
p6	Poste 6	130.120.80	06	UNIX	s03	0

poste 59

5) *Modification des colonnes :*

- Modification de nomSalle VARCHAR (20) en nomSalle VARCHAR (30) dans la table Salle

```
alter table salle modify nom_salle varchar(30);
desc salle;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
nsalle	varchar(7)	NO	PRI	NULL	
nom_salle	varchar(30)	YES		NULL	
nbposte	tinyint(2)	YES		NULL	
indip	varchar(11)	YES	MUL	NULL	

- Modification de nomSegment varchar (20) en nomSegment varchar (30) dans la table segment

```
alter table segment modify nom_segment varchar(15);
desc salle;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
nsalle	varchar(7)	NO	PRI	NULL	
nom_salle	varchar(30)	YES		NULL	
nbposte	tinyint(2)	YES		NULL	
indip	varchar(11)	YES	MUL	NULL	

6) ***Ajout des contraintes : (clé étrangère)***

- Ajout une contrainte d'unicité pour les champs nPoste , nLog dans la table installer

```
ALTER TABLE  installation ADD CONSTRAINT UNIQUE(nposte,nlog);
```

- Ajoute les contraintes d'intégrité des clé étrangères pour réaliser la relation entre les tables

```
ALTER TABLE poste ADD CONSTRAINT fk_poste_indip foreign key(indip) references segment(indip);
ALTER TABLE poste ADD CONSTRAINT fk_poste_salle foreign key(nsalle) references salle(nsalle);
ALTER TABLE poste ADD CONSTRAINT fk_poste_type foreign key(type_poste) references typelp(typelp);
ALTER TABLE installation ADD CONSTRAINT fk_poste_inst FOREIGN KEY(nposte) REFERENCES poste(nposte);
ALTER TABLE installation ADD CONSTRAINT fk_logiciel_int FOREIGN KEY(nlog) REFERENCES logiciel(nlog);
ALTER TABLE logiciel ADD CONSTRAINT fk_log_type FOREIGN KEY(typelog) REFERENCES typelp(typelp)
```

✓ **Résoudre l'error :Logiciel => Types**

```
select nlog,typelog from logiciel where typelog NOT IN(SELECT typelp from typelp)
insert into typelp values('BeOS','System be');
```

- ✓ **Résoudre l'error :Salle=> segment**

```
alter table salle add constraint fk_segment foreign key(indip) references segment(indip);
delete from salle where indip not in(select indip from segment);
```

7) Création dynamique des tables :

- Création dynamique de table Softs qui va contenir les champs suivants : nomlog, version, prix

```
create table softs AS SELECT nomlog,version,prix from logiciel;
select *from softs;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nomsoft	version	prix
Oracle 6	6.2	3000.00
Oracle 8	8i	5600.00
SQL Server	7	3000.00
Front Page	5	500.00
WinDev	5	750.00

- Changement de nom du champ nomlog en nomSoft

```
alter table softs change nomlog nomsoft varchar(20);
desc softs;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
nomsoft	varchar(20)	YES		NULL	
version	varchar(7)	YES		NULL	
prix	decimal(6,2)	YES		NULL	

- Création dynamiquement de table PCSeuls qui contient les champs nPoste ,nomPoste,IndIP,ad , typePoste ,nSalle à condition que le typeposte = PCNT ou PCWS

```
create table pcseuls as select nposte,nom_poste,indip,ad,type_poste,nsalle from poste
where type_poste='PCNT' OR type_poste='PCWS';
select *from pcseuls ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nposte	nom_poste	indip	ad	type_poste	nsalle
p4	Poste 4	130.120.80	04	PCWS	s02
p5	Poste 5	130.120.80	05	PCWS	s02
p11	Poste 11	130.120.82	01	PCNT	s21
p12	Poste 12	130.120.82	02	PCWS	s21

- Changement de nPoste par np & nomposte par nom & indIp par seg & typeposte par TypeP & nsalle par lieu

```
ALTER TABLE pcseuls CHANGE nposte np VARCHAR(7);
ALTER TABLE pcseuls CHANGE nom_poste nomP VARCHAR(20);
ALTER TABLE pcseuls CHANGE indip seg VARCHAR(11);
ALTER TABLE pcseuls CHANGE type_poste typeP VARCHAR(9);
ALTER TABLE pcseuls CHANGE nsalle lieu VARCHAR(9);
desc pcseuls
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
np	varchar(7)	YES		NULL	
nomP	varchar(20)	YES		NULL	
seg	varchar(11)	YES		NULL	
ad	varchar(3)	YES		NULL	
typeP	varchar(9)	YES		NULL	
lieu	varchar(9)	YES		NULL	

8) Requête d'extraction et monotable :

- Extraction Type de poste nposte=P8

```
180
181   select nposte,type_poste from poste where nposte='P8';
182
```

Result Grid | Filter Rows: | Edit: | Export/Import:

nposte	type_poste
p8	UNIX
NULL	NULL

- Nom des logiciels UNIX

```
182
183   select nomlog from logiciel where typelog='UNIX';
184
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nomlog
Oracle 6
Oracle 8
SQL*Net

- Nom, adresse IP, numéro de salle des postes de type UNIX ou PCWS

```
184
185   select nom_poste,indip,ad,nsalle from poste where type_poste='UNIX' or type_poste='PCWS';
186
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nom_poste	indip	ad	nsalle
Poste 2	130.120.80	02	s01
Poste 4	130.120.80	04	s02
Poste 5	130.120.80	05	s02
Poste 6	130.120.80	06	s03
Poste 8	130.120.81	01	s11
Poste 10	130.120.81	03	s12
Poste 12	130.120.82	02	s21

- Même requête pour les postes du segment 130.120.80 triés Numéro de salle décroissant

```
186
187 •   select nom_poste,indip,ad,nsalle,type_poste from poste
188     where indip='130.120.80' order by nsalle desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nom_poste	indip	ad	nsalle	type_poste
Poste 6	130.120.80	06	s03	UNIX
Poste 7	130.120.80	07	s03	TX
Poste 4	130.120.80	04	s02	PCWS
Poste 5	130.120.80	05	s02	PCWS
Poste 1	130.120.80	01	s01	TX
Poste 2	130.120.80	02	s01	UNIX
Poste 3	130.120.80	03	s01	TX

```
189
190 •   select nlog from installation where nposte='P6';
191
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nlog
log1
log6

- Numéros des postes qui hébergent le logiciel log1

```
191
192 •   select nposte from installation where nlog='log1';
193
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nposte
p2
p6

- Nom et adresse IP complète (ex : 130.120.80.01) des postes de type TX

```
193
194 •   select nom_poste,CONCAT(indip,'.',ad) as adresse_ip from poste where type_poste='TX';
195
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nom_poste	adresse_ip
Poste 1	130.120.80.01
Poste 3	130.120.80.03
Poste 7	130.120.80.07
Poste 9	130.120.81.02

9) Groupements :

- Nombre de logiciels pour chaque poste

```
195
196 •   select nposte,count(nlog) as nbr_logiciel from installation group by nposte;
197
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nposte	nbr_logiciel
p11	2
p12	1
p2	2
p4	1
p6	2
p7	1
p8	2

- Nombre des postes pour chaque salle

```
197
198 •   SELECT nsalle, COUNT(nposte) FROM poste GROUP BY (nsalle) ORDER BY 2;
199
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nsalle	COUNT(nposte)
s12	1
s02	2
s03	2
s11	2
s21	2
s01	3

- Nombre de poste par logiciel

```
199
200 •   SELECT nlog, COUNT(nposte) FROM installation GROUP BY (nlog);
201
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nlog	COUNT(nposte)
log1	2
log2	2
log3	1
log4	1
log5	1
log6	2
log7	2

```
201
202 •   SELECT AVG(prix) FROM logiciel WHERE typelog = 'UNIX';
203
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVG(prix)
3033.333333

- Maximum de la date d'achat des logiciels

```
204
205 •   SELECT MAX(dateach) FROM logiciel;
```

Result Grid | Filter Rows: | Export: | Wr

MAX(dateach)
2003-09-21 00:00:00

- Nombre des postes qui ont au plus 2 logiciel installé

```
204
205 •   SELECT MAX(dateach) FROM logiciel;
```

Result Grid | Filter Rows: | Export: | Wr

MAX(dateach)
2003-09-21 00:00:00

10) Jointure procédurale :

- Affichage l'adresse IP des postes ou le logiciel log6 est installé

```
210
211 •   SELECT CONCAT(indip,'.',ad) FROM poste
212      WHERE nposte IN
213      (SELECT nposte FROM installation WHERE nlog = 'log6');
214
215
216
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	CONCAT(indip,'.',ad)
▶	130.120.80.06
	130.120.81.01

- Affichage de nom du segment qui contient l'indice de l'adresse IP ou le type de poste = « TX »

```
216 •   SELECT nom_segment FROM segment
217      WHERE indip IN (SELECT indip FROM Poste WHERE type_poste = 'TX'
218      GROUP BY indip HAVING COUNT(*)=3);
219
220
221
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	nom_segment
▶	Brin RDC

- Affichage de nom de la salle ou le logiciel oracle 6 est installé dans le poste concerné

```
SELECT nom_salle FROM salle WHERE nsalle IN
 (SELECT nsalle FROM poste WHERE nposte IN
 (SELECT nposte FROM installation WHERE nlog =
 (SELECT nlog FROM logiciel WHERE nomlog = 'Oracle 6')));
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nom_salle
Salle 1
Salle 3

- Nom du logiciel ayant la date d'achat la plus récente

```
SELECT nomlog FROM Logiciel WHERE dateach =
(SELECT MAX(dateach) FROM logiciel);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Con

nomlog
DreamWeaver

Conclusion :

Grâce à ce projet j'ai eu une journée très productive, j'ai beaucoup appris. Le breif était simple mais il me permet de mieux comprendre les commandes de SQL.