Le Langage SQL
Création de tables

Le Langage de Définition de Données (DDL)

Introduction

- **Le langage de définition de données permet de créer la structure des tables qui composent votre schéma de données.**

- **Un schéma de données est issu d'un modèle conceptuel de données ou de tout autre modèle ou langage (UML par exemple) permettant la modélisation des données d'une application.**

La création de tables

- **La syntaxe générale**

```
CREATE TABLE nom_table
(
nom_colonne { type } [ DEFAULT valeur_default ][ contrainte.. ],
…… ,
CONSTRAINT nom_contrainte {   UNIQUE |
                    PRIMARY KEY ( liste_colonne )|
                    FOREIGN KEY liste colonne
                          REFERENCES nom_table (liste_colonne)
);
```

La création de tables

Quelques exemples

```
CREATE TABLE CLIENT
(
   NOM      CHAR(32),
   PRENOM   VARCHAR(32) );

CREATE TABLE VOITURE
(
   ID              INTEGER       NOT NULL PRIMARY KEY,
   MARQUE          CHAR(32)      NOT NULL,
   MODELE          VARCHAR(16),
   IMMATRICULATION CHAR(10)      NOT NULL UNIQUE,
   COULEUR         VARCHAR(15)
   constraint C_color
     check (COULEUR IN('BLANC', 'NOIR', 'ROUGE', 'VERT'))
);
```

Le type des colonnes

- il existe plusieurs types pour définir les colonnes d'une table :
  - Les types numériques
  - le type auto_incrémenter
  - Les types de dates
  - Les types chaînes de caractères
  - Le type booléen

Le type des colonnes

- Les types numériques
  - Les entiers

smallint	2 octets	-32768 à +32767
integer	4 octets	-2147483648 à +2147483647
bigint	8 octets	-9223372036854775808 à 9223372036854775807
decimal	variable	pas de limite
numeric	variable	pas de limite
real	4 octets	précision de 6 décimales
double precision	8 octets	précision de 15 décimales
serial	4 octets	1 à 2147483647
bigserial	8 octets	1 à 9223372036854775807

Le type des colonnes

- **Les types numériques**
  - **Les réels**

DECIMAL (length,decimals) [UNSIGNED] [ZEROFILL]
NUMERIC (length,decimals) [UNSIGNED] [ZEROFILL]

Salaire decimal(5,2)

Standard SQL = > -999.99 et 999.99

Le type des colonnes

- **Les types numériques**
  - **Le type auto_incrémenter (SERIAL ou BIGSERIAL)**

```
CREATE TABLE personne
(    id    SERIAL NOT NULL PRIMARY KEY ,
     name VARCHAR(60) NOT NULL
);

INSERT INTO personne VALUES (default, 'Antonio Paz');
INSERT INTO personne VALUES (default, 'Lilliana Angelovska');
INSERT INTO personne VALUES (default, 'André Dopund');
INSERT INTO personne VALUES (default, 'René Dunp');
```

id	name
1	Antonio Paz
2	Lilliana Angelovska
3	André Dopund
4	René Dunp

Le type des colonnes

Les types numériques

Le type auto_incrémenter (SERIAL ou BIGSERIAL)

Quelques fonctions sur les séquences

Un objet séquence est créé dans le schéma de la base de données il est nommé par défaut : NOMTABLE_NOMDUCHAMP_SEQ

Dans notre exemple : *Personne_id_seq*

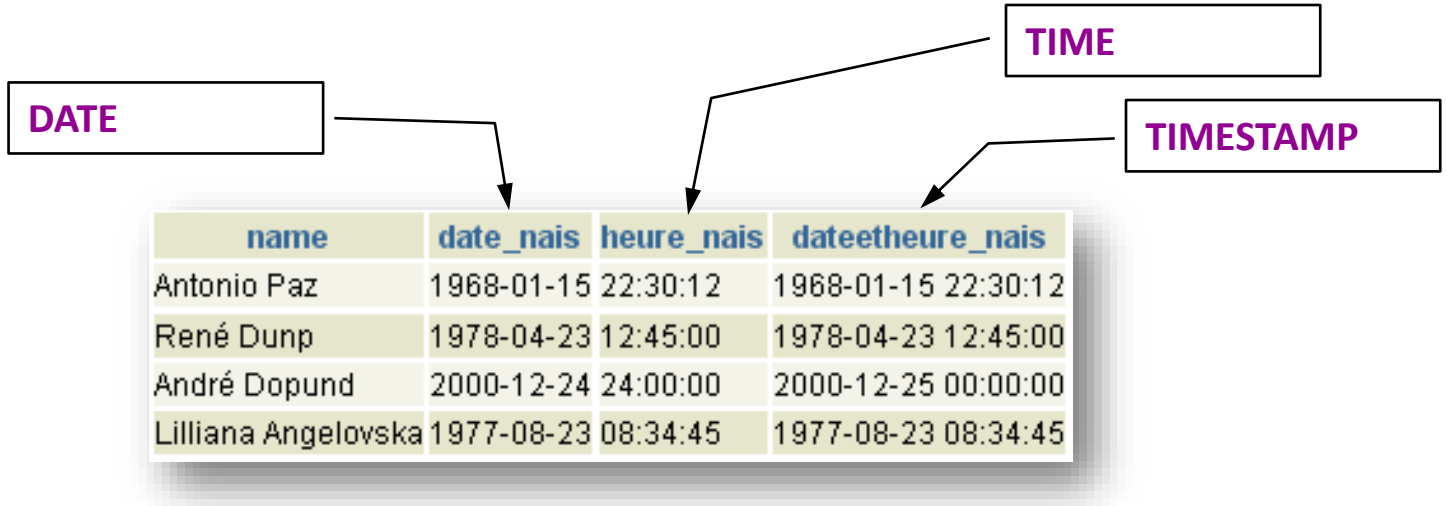
Fonction	Type de retour	Description
nextval('NomSeq')	bigint	Incrémente la valeur du champ auto incrémenté de la séquence spécifiée
currval('NomSeq')	bigint	Valeur de retour obtenue le plus récemment avec nextval pour la séquence spécifiée
select setval('NomSeq', bigint)	bigint	Initialise la valeur courante de la séquence

```
INSERT INTO personne VALUES (nextval('personne_id_seq'), 'didier');
SELECT currval('personne_id_seq'); => retournera 4
```

Le type des colonnes

- **Les types date et heure**

Nom	Description
DATE	dates seulement
INTERVAL	intervalle de temps
TIMESTAMP	date et heure
TIME	heures seulement



Le type des colonnes

Les types chaînes de caractères

Nom	longueur
character(*n*) ou char(*n*)	longueur fixe, comblé avec des espaces
varying(n) ou varchar(n)	Longueur variable avec limite
text	longueur variable illimitée

Valeur	CHAR(4)	Espace requis	VARCHAR(4)	Espace requis
''	' '	4 octets	''	1 octet
'ab'	'ab '	4 octets	'ab'	3 octets

taille maximale possible pour une chaîne de caractères est de l'ordre 1 Go.

Le type des colonnes

- **Les types boolean**

  Le type booléen ne peut avoir que deux états:
  TRUE (vrai) et FALSE (faux)

  Les valeurs littérales valides pour l'état vrai sont :
  TRUE | 't' | 'true' | 'y' | 'yes' | ' 1'

  Pour l'état faux, les valeurs suivantes peuvent être utilisées:
  FALSE | ' f' | ' false' | 'n' | 'no' | 'o'

Il est recommandé d'utiliser TRUE et FALSE (qui sont compatibles avec la norme SQL).

Il existe d'autres types sous postgres, pour en savoir plus :
http://docs.postgresqlfr.org/9.1/datatype.html

Les contraintes

Introduction

name	date_nais	heure_nais	dateetheure_nais
Antonio Paz	1968-01-15	22:30:12	1968-01-15 22:30:12
René Dunp	1978-04-23	12:45:00	1978-04-23 12:45:00
André Dopund	2000-12-24	24:00:00	2000-12-25 00:00:00
Lilliana Angelovska	1977-08-23	08:34:45	1977-08-23 08:34:45

Contrainte de colonne :
Ne concerne que la colonne spécifiée

Contrainte de ligne ou de table :
concerne plusieurs colonnes de la table

Les contraintes

- **Les contraintes de colonnes**

  - **PRIMARY KEY**
  - **NULL / NOT NULL**
  - **DEFAULT valeur**
  - **UNIQUE**
  - **CHECK**

Les contraintes

- **Les contraintes de colonnes**
  - **Clé primaire (PRIMARY KEY) obligatoire et unique**

```
CREATE TABLE PERSONNE
(
  ID            INTEGER     PRIMARY KEY,
  NOM           VARCHAR(32),
  PRENOM        VARCHAR(32) );

  INSERT INTO PERSONNE VALUES (NULL, 'Duchemin', 'Paul');
```

Erreur SQL :

ERREUR: une valeur NULL viole la contrainte NOT NULL de la colonne « id »

```
  INSERT INTO PERSONNE VALUES (1, 'Duchemin', 'Paul');
  INSERT INTO PERSONNE VALUES (1, 'Duchemol', 'Pierre');
```

Erreur SQL :

ERREUR: la valeur d'une clé dupliquée rompt la contrainte unique « personne_pkey »

Les contraintes

- Les contraintes de colonnes
  - Valeur obligatoire NOT NULL - NULL

```
CREATE TABLE PERSONNE
(
  ID                INTEGER       PRIMARY KEY,
  NOM               VARCHAR(32)   NOT NULL,
  PRENOM            VARCHAR(32)   NULL,
  DATE_NAISSANCE    DATE
);

  INSERT INTO PERSONNE VALUES (10, null, null, now() );
  ou
  INSERT INTO PERSONNE(prenom,date_naissance) VALUES (null, now());
```

Erreur SQL :

ERREUR: une valeur NULL viole la contrainte NOT NULL de la colonne « id »

Les contraintes

- Les contraintes de colonnes
  - Valeur par défaut (DEFAULT)

```
CREATE TABLE PERSONNE
(
    ID              INTEGER,
    NOM             VARCHAR(32),
    PRENOM          VARCHAR(32),
    SEXE            CHAR(1)    DEFAULT 'M',
    DATE_NAISSANCE  TIMESTAMP  DEFAULT CURRENT_TIMESTAMP
);

   INSERT INTO PERSONNE  VALUES(10, 'Duchemin', 'Paul',default, default);
```

id	nom	prenom	sexe	date_naissance
10	Duchemin	Paul	M	2010-05-06 11:23:54.937

Les contraintes

- **Les contraintes de colonnes**
  - **Unicité (UNIQUE) (Si non renseigné = NULL)**

```
CREATE TABLE PERSONNE
(
   NOM          VARCHAR(32),
   PRENOM       VARCHAR(32),
   TELEPHONE    CHAR(14)  UNIQUE
);
```

```
INSERT INTO PERSONNE VALUES('DUPONT','MARCEL', '01 44 21 57 18');
INSERT INTO PERSONNE VALUES ('DUVAL', 'ANDRÉ',NULL);
INSERT INTO PERSONNE VALUES ('DURAND','JEAN', '06 11 86 46 69');
INSERT INTO PERSONNE VALUES ('DUGLAND','ALFRED','06 11 86 46 69');
```

Erreur SQL :

ERREUR: la valeur d'une clé dupliquée rompt la contrainte unique « personne_telephone_key »

Les contraintes

- **Les contraintes de colonnes**
  - **La verification : CHECK**

```
CREATE TABLE PERSONNE
(
   NOM      VARCHAR(32),
   PRENOM   VARCHAR(32),
   AGE      INTEGER  constraint c_age CHECK (AGE between 0 and 125),
   TEL      VARCHAR(15) constraint c_tel CHECK (TEL like '__-__-__-__-__')
);
```

```
INSERT INTO PERSONNE  VALUES('DUPONT', 'MARCEL', 52,'22-22-22-22-22'); --OK
INSERT INTO PERSONNE  VALUES('DUVAL',  'ANDRÉ', 126,'22-22-22-22-22');-- erreur
INSERT INTO PERSONNE  VALUES('DURAND', 'JEAN', -25,'22-22-22-22-22'); -- erreur
INSERT INTO PERSONNE  VALUES('DURAND', 'JEAN', 25,'22-22-22-22');     -- erreur
```

Les contraintes

- **Les contraintes de colonnes**
  - **La verification : CHECK**

Autre exemple pour forcer le format d'un N° de telephone

```
CREATE TABLE PERSONNE
(
   NOM       VARCHAR(32),
   PRENOM    VARCHAR(32),
   AGE       INTEGER  constraint c_age CHECK (AGE between 0 and 125),
   TEL       VARCHAR(15)
               constraint c_tel
                  CHECK (TEL  SIMILAR TO '[0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]')
);
INSERT INTO PERSONNE  VALUES('DURAND', 'JEAN', 52,'aa-22-22-22-22');   -- erreur
INSERT INTO PERSONNE  VALUES('DURAND', 'JEAN', 25,'22-22-22-22');      -- erreur
```

Les contraintes

- **Les contraintes de table**

  Les contraintes de table mettent en jeux plusieurs colonnes de la table

  - **PRIMARY KEY**
  - **UNIQUE**
  - **FOREIGN KEY**
  - **CHECK**

Les contraintes

- **Les contraintes de table**
  - **Les clés primaires (PRIMARY KEY)**

Exemple de clé composée de 2 colonnes

```
CREATE TABLE PERSONNE
(
 NOM       VARCHAR(32),
 PRENOM    VARCHAR(32),
 TELEPHONE CHAR(14),
 CONSTRAINT PK_PRS PRIMARY KEY (NOM, PRENOM)
);
```

Les contraintes

- **Les contraintes de table**
  - **Unicité(UNIQUE)**

```
CREATE TABLE  PERSONNE
(
 ID        INTEGER,
 NOM       VARCHAR(32),
 PRENOM    VARCHAR(32),
 CONSTRAINT UNI_N_P UNIQUE (NOM, PRENOM)
);
```

Les contraintes

- **Les contraintes de table**
  - La vérification **CHECK**

```
CREATE TABLE FACTURE
(
ID        INTEGER,
DATE      DATE,
MT_MIN    DECIMAL(16,2),
MONTANT   DECIMAL(16,2),
Constraint C_ValMt    CHECK (MONTANT > MT_MIN)
);

INSERT INTO FACTURE values (10, '2007-06-04',10, 100); -- ok
INSERT INTO FACTURE values (10, '2007-06-04',10, 8); --erreur
```

Erreur SQL :

ERROR: new row for relation "facture" violates check constraint "c_valmt"

Les contraintes

- **Les contraintes de table**
  - **Les clés étrangéres (FOREIGN KEY)**

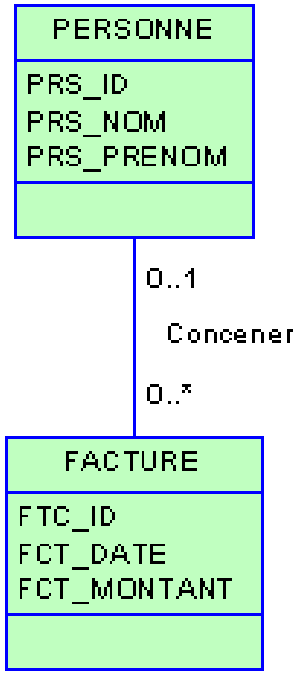
```
CONSTRAINT NOM_CONTRAINTE FOREIGN KEY (LISTE_COLONNE) REFERENCES
                          NOM_TABLE_REF (LISTE_COLONNE_REF);
```

Les contraintes

- **Les contraintes de table**
  - **Les clés étrangères (FOREIGN KEY)**

```
CREATE TABLE PERSONNE
(PRS_ID                 INTEGER  PRIMARY KEY,
  PRS_NOM               VARCHAR(32),
  PRS_PRENOM            VARCHAR(32)
);

CREATE TABLE FACTURE
( FTC_ID                INTEGER PRIMARY KEY,
  PRS_ID                INTEGER ,
  FCT_DATE              DATE,
  FCT_MONTANT           DECIMAL(16,2),
  Constraint FK_facture FOREIGN KEY (PRS_ID)
                        REFERENCES PERSONNE (PRS_ID)
);
```



Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**

Le mode de gestion de l'intégrité consiste à se poser la question de ce que la machine doit faire dans le cas ou l'on tente de briser une intégrité référentielle.

Clé étrangère => clé primaire

```
CONSTRAINT nom_contrainte FOREIGN KEY (liste_colonne_table)
              REFERENCES table_réf(liste_colonne_référencées)
[ON DELETE {CASCADE | SET NULL | SET DEFAULT | NO ACTION | RESTRICT}]
[ON UPDATE {CASCADE | SET NULL | SET DEFAULT | NO ACTION | RESTRICT}]
```

Les contraintes

- Les contraintes de table
  - La gestion de l'intégrité référentielle
    - Mode de gestion de l'intégrité

NO ACTION / RESTRICT
Les mises à jour ou suppression de la clé primaire sont interdites si cette dernière est référencée par une clé étrangère

CASCADE
Les mise à jour ou suppression de la clé primaire entraînent les mise à jour ou suppression de toutes les clés étrangères qui la référencent.

SET NULL
Les mises à jour ou suppression de la clé primaire entraînent toutes les clés étrangères qui la référencent à ne pas être définies (valeur NULL).

SET DEFAULT
Les mise à jour ou suppression de la clé primaire provoque l'affectation de leur valeur par défaut à toutes les clés étrangères qui la référencent.

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE NO ACTION / ON UPDATE NO ACTION**

    La contrainte sera vérifiée et une erreur générée si les valeurs violent la conrainte, c'est le comportement pas défaut

```
CREATE TABLE FACTURE
(
FTC_ID          INTEGER primary key,
PRS_ID          INTEGER ,
FCT_DATE        DATE,
FCT_MONTANT     DECIMAL(16,2),
Constraint FK_facture FOREIGN KEY (PRS_ID) REFERENCES PERSONNE (PRS_ID)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
);
```

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE NO ACTION / ON UPDATE NO ACTION**
    - **Exemple**

facture

ftc_id	prs_id	fct_date	fct_montant
100	1	2009-03-01	2300.00
200	1	2009-02-23	1890.00
300	2	2009-01-15	590.00
400	2	2009-02-28	890.00

personne

prs_id	prs_nom	prs_prenom
1	Carmignac	laurent
2	Dupond	Jacques

```
-- tentative de suppression d'une personne ayant une facture
Par exemple si l'on veut supprimer la personne d'id 1 (prs_id= 1) on aura
l'erreur suivante
```

Erreur SQL :

ERREUR: UPDATE ou DELETE sur la table « personne » viole la contrainte de clé étrangère « facture_prs_id_fkey » de la table « facture »
DETAIL: La clé (prs_id)=(1) est toujours référencée à partir de la table « facture ».

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE CASCADE / ON UPDATE CASCADE**

    La contrainte NE SERA PAS vérifiée et le SGDB supprimera ou modifiera toutes les valeurs de la clé étrangère qui sont liées à la valeur de la clé primaire supprimée ou modifiée

```
CREATE TABLE FACTURE
(
FTC_ID          INTEGER,
PRS_ID          INTEGER ,
FCT_DATE        DATE,
FCT_MONTANT     DECIMAL(16,2),
Constraint FK_facture FOREIGN KEY (PRS_ID) REFERENCES PERSONNE (PRS_ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE CASCADE / ON UPDATE CASCADE**
    - **Exemple**

personne

prs_id	prs_nom	prs_prenom
1	Carmignac	laurent
2	Dupond	Jacques

facture

ftc_id	prs_id	fct_date	fct_montant
100	1	2009-03-01	2300.00
200	1	2009-02-23	1890.00
300	2	2009-01-15	590.00
400	2	2009-02-28	890.00

```
-- Modification de la valeur de la clé d'une personne ayant une facture
Si l'on modifie l'id de la personne 1 (prs_id=1) avec la valeur 10 on aura :
```

personne

prs_id	prs_nom	prs_prenom
10	Carmignac	laurent
2	Dupond	Jacques

facture

ftc_id	prs_id	fct_date	fct_montant
100	10	2009-03-01	2300.00
200	10	2009-02-23	1890.00
300	2	2009-01-15	590.00
400	2	2009-02-28	890.00

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE CASCADE / ON UPDATE CASCADE**
    - **Exemple 2**

personne

prs_id	prs_nom	prs_prenom
10	Carmignac	laurent
2	Dupond	Jacques

facture

ftc_id	prs_id	fct_date	fct_montant
100	10	2009-03-01	2300.00
200	10	2009-02-23	1890.00
300	2	2009-01-15	590.00
400	2	2009-02-28	890.00

```
-- suppression d'une personne ayant une facture
Par exemple si l'on veut supprimer la personne d'id 10 (prs_id=10) on aura la
suppression en cascade des factures qui lui sont associées
```

personne

prs_id	prs_nom	prs_prenom
2	Dupond	Jacques

facture

ftc_id	prs_id	fct_date	fct_montant
300	2	2009-01-15	590.00
400	2	2009-02-28	890.00

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE SET NULL / ON UPDATE SET NULL**

    La contrainte NE SERA PAS vérifiée et le SGDB modifiera toutes les valeurs de la clé étrangère qui sont liées à la valeur de la clé primaire supprimée ou modifiée en plaçant la valeur NULL dans la clé étrangére

```
CREATE TABLE FACTURE
(
FTC_ID          INTEGER,
PRS_ID          INTEGER ,
FCT_DATE        DATE,
FCT_MONTANT     DECIMAL(16,2),
Constraint FK_facture FOREIGN KEY (PRS_ID) REFERENCES PERSONNE (PRS_ID)
        ON DELETE SET NULL
        ON UPDATE SET NULL
);
```

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE SET NULL / ON UPDATE SET NULL**
    - **Exemple**

personne

prs_id	prs_nom	prs_prenom
1	Carmignac	laurent
2	Dupond	Jacques

facture

ftc_id	prs_id	fct_date	fct_montant
100	1	2009-03-01	2300.00
200	1	2009-02-23	1890.00
300	2	2009-01-15	590.00
400	2	2009-02-28	890.00

-- Modification de la valeur de la clé d'une personne ayant une facture

Si l'on modifie l'id de la personne 1 (prs_id=1) avec la valeur 10 on aura :

personne

prs_id	prs_nom	prs_prenom
2	Dupond	Jacques
10	Carmignac	Laurent

facture

ftc_id	prs_id	fct_date	fct_montant
300	2	2009-01-15	590.00
400	2	2009-02-28	890.00
100	NULL	2009-02-23	1890.00
200	NULL	2009-03-01	2300.00

La suppression d'une personne aura le même effet sur la table facture

Les contraintes

- **Les contraintes de table**
  - **La gestion de l'intégrité référentielle**
  - **ON DELETE SET DEFAULT/ ON UPDATE SET DEFAULT**

La contrainte NE SERA PAS vérifiée et le SGDB modifiera toutes les valeurs de la clé étrangère qui sont liées à la valeur de la clé primaire supprimée ou modifiée en plaçant la valeur NULL dans la clé étrangère

```
CREATE TABLE FACTURE
(
FTC_ID          INTEGER,
PRS_ID          INTEGER DEFAULT 0,
FCT_DATE        DATE,
FCT_MONTANT     DECIMAL(16,2),
Constraint FK_facture FOREIGN KEY (PRS_ID) REFERENCES PERSONNE (PRS_ID)
        ON DELETE SET DEFAULT
        ON UPDATE SET DEFAULT
);
```

Les contraintes

Les contraintes de table

La gestion de l'intégrité référentielle

ON DELETE SET DEFAULT/ ON UPDATE SET DEFAULT

- **Exemple**

personne

prs_id	prs_nom	prs_prenom
0	pas de client	*NULL*
1	Carmignac	Laurent
2	Dupond	Jacques

facture

ftc_id	prs_id	fct_date	fct_montant
100	1	2009-12-12	2345.00
200	1	2009-10-23	1890.00
300	2	2009-03-12	590.00
400	2	2009-09-05	890.00

-- Modification de la valeur de la clé d'une personne ayant une facture

Si l'on modifie l'id de la personne 2 (prs_id=2) avec la valeur 10 on aura :

personne

prs_id	prs_nom	prs_prenom
0	pas de client	*NULL*
1	Carmignac	Laurent
10	Dupond	Jacques

facture

ftc_id	prs_id	fct_date	fct_montant
100	1	2009-12-12	2345.00
200	1	2009-10-23	1890.00
300	0	2009-03-12	590.00
400	0	2009-09-05	890.00

La suppression d'une personne aura le même effet sur la table facture

FIN