

Tests sur les fonctionnalités de la base de données

Par

Olivier Dunn (duno5273)

Ahmed Hamissi (hama1461)

Maxime Malette (malm3308)

Thomas Paré (part5347)

Travail présenté à

Nadia Tahiri

Éléments de bases de données

Automne-2025 IFT 187

Université de Sherbrooke

Date de remise : 8 octobre 2025

Table des matières

Tests sur les fonctionnalités de la base de données.....	1
Table des matières	2
Contenu	3
Fonction 1 : Gestion.f_getLoyer(unLocataire).....	4
Description :	4
Exemples	4
Fonction 2 : Gestion.f_getStatutLogement(unLogement).....	5
Description	5
Exemples	5
Fonction 3 : Gestion.f_getSolde(unLocataire)	6
Description	6
Exemples	6
Fonction 4 : Gestion.f_initialiserLoyerAuSolde(NEW) à partir d'un trigger.	7
Description	7
Exemples	8
Fonction 5 : Gestion.f_bailActif(unBail)	9
Description	9
Exemple.....	10
Fonction 6 Gestion.f_getBail(unLocataire).....	10
Description	10
Exemples	10
Fonction 7 : Gestion.f_locataireBailActif(NEW).....	11
Description	11
Exemples	11
Fonction 8 Gestion.f_updateSoldePaiement(NEW).....	12
Description	12
Exemples	12
Fonction 9 : Gestion.f_genererProfitProprietaire(unePeriode)	13
Description	13
Exemple.....	14

Fonction 10 : Gestion.f_afficherDureeBail(unBail).....	14
Description	14
Exemples	15
Fonction 11 : Gestion.afficherPrenomLocataire(unLocataire).....	15
Description	15
Exemples	15
FONCTION 12 Gestion.afficherAdresse(unAdresse).....	16
Description	16
Exemples	16
Fonction 13 : Gestion.f_liste_des_baux_solde()	17
Description :	17
Exemples	18
PROCÉDURE 1) Gestion.p_soldeNouveauMois().....	19
Description	19
Exemples	19
Contrainte verif_date_valide_[table]	19
Description	19
Exemples	20

Contenu

Le document suivant contient l'ensemble des fonctions et des procédures qui sont intégrées dans la base de données de gestion. On y retrouve une description sommaire de chacune d'entre elles, des exemples d'utilisations dans des cas valides et critiques ainsi que les scripts permettant d'effectuer les tests.

IMPORTANT !!!

Chaque script de test (enfin la majorité) suppose que le script suivant a été inséré également.

```
--Script de base pour tous les test de fonctions

--Données de base pour les test.
--Adresse de test
INSERT INTO Gestion.Adresses (idAdresse, noCivique, rue, ville, codePostal) VALUES
(10, 680, 'Adresse test', 'TEST', 'TEST');
--Proprietaire de test
INSERT INTO Gestion.Proprietaires (idProprietaire, idAdresse, nbPortesPro, nbImmeublesPro) VALUES
(5, 11, 1, 1);
--Règles de tests
INSERT INTO Gestion.Regles (idRegle, fumeur, chat, chien, bbq) VALUES
(5, true, true, true, true);
--Inclusion de tests
INSERT INTO Gestion.Inclusions (idInclusion, meuble, chauffage, electricite, internet, eauChaude) VALUES
(5, true, true, true, true, true);
--Immeuble de test
INSERT INTO Gestion.Immeubles (idImmeuble, idAdresse, idProprietaire, idRegle, nbPortes, nbStationnement, description) VALUES
(5, 10, 5, 5, 5, 5, 'Immeuble TEST');
```

Fonction 1 : Gestion.f_getLoyer(unLocataire)

Description :

Cette fonction prend en paramètre l'identifiant d'un locataire et retourne le montant de son loyer. Si le locataire ne possède pas de bail actif, la valeur NULL est retournée, idem si le locataire n'existe pas.

Exemples

Voici le script pour tester cette fonction, à noter que chaque partie a été exécutée séparément.

```
BEGIN;
--Test 1 le locataire n'existe pas (Retourne NULL)
SELECT * FROM Gestion.f_getLoyer(97);

--Insertion Locataire test
INSERT INTO Gestion.Locataires(idLocataire, idLogement, langue, noTelephone, prenom, nom, courriel, modePaiement)
VALUES (97,19,'Français', '4371152227','Test','Normal','CourrielNormal','Comptant');

SELECT * FROM Gestion.f_getLoyer(97);
--Test 2 Le locataire n'a pas de bail (Retourne NULL)

-- Test 3 Retourne 1000.00
INSERT INTO Gestion.Baux(idBail,idLocataire, dateDebut, dateFin, loyer)
VALUES (97,97,'2025-12-04', '2026-12-03', 1000.00);

SELECT * FROM Gestion.f_getLoyer(97);
```

1) Test avec un locataire inexistant

	f_getloyer numeric
1	[null]

Le locataire 97 n'existant pas, cela retourne un loyer NULL.

2) Test avec un locataire sans bail

	f_getloyer numeric
1	[null]

3) Test avec un locataire et un bail valide

	f_getloyer numeric
1	1000.00

Fonction 2 : Gestion.f_getStatutLogement(unLogement)

Description

Cette fonction prend en paramètre l'identifiant d'un logement et retourne le statut de celui-ci.

Voici le script pour tester cette fonction, a noté que chaque partie a été exécuté séparément.

```
INSERT INTO Gestion.Logements (idLogement, idImmeuble, idInclusion, nbPieces, balcon, statut, noPorte, description)
VALUES --Test 1 Loué
      (100,5,5,4,False,'Loué',5, NULL),

      --Test 2 Vacant
      (101,5,5,4,False,'Vacant',5, NULL),

      --Test 3 En réno
      (102,5,5,4,False,'En réno',5, NULL);
-- Test 4 N'existe pas
SELECT * FROM Gestion.f_getStatutLogement(100);
SELECT * FROM Gestion.f_getStatutLogement(101);
SELECT * FROM Gestion.f_getStatutLogement(102);
SELECT * FROM Gestion.f_getStatutLogement(103);
```

Exemples

1) Le logement est loué

	f_getstatutlogement statutlogement
1	Loué

2) Le logement est Vacant

	f_getstatutlogement statutlogement
1	Vacant

3) Le logement est en réno

	f_getstatutlogement statutlogement
1	En réno

4) Le logement n'existe pas

```
ERROR: Le logement 103 n'existe pas
CONTEXT: PL/pgSQL function gestion.f_getstatutlogement(integer) line 11 at RAISE

SQL state: P0001
```

Fonction 3 : Gestion.f_getSolde(unLocataire)

Description

Cette fonction prend l'identifiant d'un locataire en paramètre et y retourne son solde actuelle.

Voici le script qui permet de tester la fonction.

```
INSERT INTO Gestion.Locataires(idLocataire, idLogement, langue, noTelephone, prenom, nom, courriel, modePaiement, solde)
VALUES --Test 1 pas de solde dans le insert
      (100,18,'Français', '4371152227','Test','Normal','CourrielNormal','Comptant', NULL),

      --Test 2 avec un solde en entré.
      (101,19,'Français', '4371152227','Test','Normal','CourrielNormal','Comptant', 100.00);

SELECT * FROM Gestion.f_getSolde(100);
SELECT * FROM Gestion.f_getSolde(101);

INSERT INTO Gestion.Baux(idBail,idLocataire, dateDebut, dateFin, loyer)
VALUES -- Test 3 le locataire à un loyer
      (106,100,'2025-12-04', '2026-12-03', 1000.00);
SELECT * FROM Gestion.f_getSolde(100);
```

Exemples

1) Test le locataire n'a pas de solde en entré ni de bail actif

	f_getsolde numeric
1	[null]

2) Test : le locataire a un solde en entré.

	f_getsolde numeric
1	100.00

3) Test : le locataire possède un bail avec un loyer

	f_getsolde numeric
1	1000.00

Fonction 4 : Gestion.f_initialiserLoyerAuSolde(NEW) à partir d'un trigger.

Description

Le trigger se déclenche AFTER un INSERT ou UPDATE pour chaque ligne de la table baux en exécutant la fonction Gestion.f_initialiserLoyerAuSolde(NEW). Cette fonction va initialiser le solde du locataire associé à l'identifiant du bail avec le montant du loyer. Si le locataire bénéficie de mois gratuits, son solde sera ajusté en conséquence.

Voici les scripts pour chacun des tests

```
-- Test 1 : locataire sans solde
--Création du logement de test:
INSERT INTO Gestion.Logements(idLogement, idImmeuble, idInclusion, nbPieces, balcon, statut, noPorte, description)
VALUES (21, 5, 5, 4, FALSE, 'Vacant', 20, 'Logement de test');
-- Création du locataire avec solde NULL
INSERT INTO Gestion.Locataires(idLocataire, idLogement, langue, noTelephone, prenom, nom, courriel, modePaiement, solde)
VALUES (201, 20, 'Français', '4371152227', 'Test', 'SansSolde', 'courriel@test', 'Comptant', NULL);

-- Vérification avant bail : doit retourner NULL
SELECT * FROM Gestion.f_getSolde(201);

-- Insertion d'un bail avec loyer
INSERT INTO Gestion.Baux(idBail, idLocataire, dateDebut, dateFin, loyer)
VALUES (201, 201, '2025-12-04', '2026-12-03', 1200.00);

-- Vérification après bail : doit retourner 1200.00
SELECT * FROM Gestion.f_getSolde(201);

1) -- Test 2 : locataire avec solde négatif (2 mois gratuits)
--Création du logement de test:
INSERT INTO Gestion.Logements(idLogement, idImmeuble, idInclusion, nbPieces, balcon, statut, noPorte, description)
VALUES (22, 5, 5, 4, FALSE, 'Vacant', 20, 'Logement de test');

-- Création du locataire avec solde = -2000.00 (si loyer = 1000.00)
INSERT INTO Gestion.Locataires(idLocataire, idLogement, langue, noTelephone, prenom, nom, courriel, modePaiement, solde)
VALUES (301, 22, 'Français', '4371152227', 'Test', 'Gratuit', 'courriel@test', 'Comptant', -2000.00);

-- Vérification avant bail : doit retourner -2000.00
SELECT * FROM Gestion.f_getSolde(301);

-- Insertion d'un bail avec loyer 1000.00
INSERT INTO Gestion.Baux(idBail, idLocataire, dateDebut, dateFin, loyer)
VALUES (301, 301, '2025-12-04', '2026-12-03', 1000.00);

-- Vérification après bail : doit retourner -1000.00 (le trigger ajoute le loyer au solde négatif)
SELECT * FROM Gestion.f_getSolde(301);

2)
```

```

-- Test 3 : Locataire avec solde positif
--Création du logement de test:
INSERT INTO Gestion.Logements(idLogement, idImmeuble, idInclusion, nbPieces, balcon, statut, noPorte, description)
VALUES (25, 5, 5, 4, FALSE, 'Vacant', 20, 'Logement de test');

-- Création du locataire avec solde = 500.00
INSERT INTO Gestion.Locataires(idLocataire, idLogement, langue, noTelephone, prenom, nom, courriel, modePaiement, solde)
VALUES (304, 25, 'Français', '4371152227', 'Test', 'Positif', 'courriel@test', 'Comptant', 500.00);

-- Vérification avant bail : doit retourner 500.00
SELECT * FROM Gestion.f_getSolde(304);

-- Insertion d'un bail avec loyer 1000.00
INSERT INTO Gestion.Baux(idBail, idLocataire, dateDebut, dateFin, loyer)
VALUES (304, 304, '2025-12-04', '2026-12-03', 1000.00);

-- Vérification après bail : doit toujours retourner 500.00
SELECT * FROM Gestion.f_getSolde(304);

```

3)

Exemples

1) Test locataire sans solde

	f_getsolde numeric
1	[null]

Le solde est null s'il n'y a pas de bail.

	f_getsolde numeric
1	1200.00

Lorsque le locataire est associé à un bail, le trigger s'enclenche et le solde est initialisé à ce montant

2) Test locataire avec mois gratuit. Solde négatif

	f_getsolde numeric
1	-2000.00

Le solde est créé à -2000.00 dans le locataire

	f_getsolde numeric
1	-1000.00

Après avoir créé le bail ayant un loyer de 1000.00, le solde du locataire augmente de 1000.00, car il avait 2 mois gratuits.

3) Test locataire avec solde positif

	f_getsolde numeric
1	500.00

Le solde est initialisé à 500.00.

	f_getsolde numeric
1	500.00

Après l'exécution de l'insertion d'un bail à 1000.00, le solde demeure à 500.00.

Fonction 5 : Gestion.f_bailActif(unBail)

Description

Cette fonction prend un identifiant de bail en entrée et retourne vraie si le bail est encore actif (en comparant la date de fin à la date actuelle).

Voici le script permettant de tester la fonction

```
-- Test 1 : bail expiré
-- Insertion d'un logement test
INSERT INTO Gestion.Logements(idLogement, idImmeuble, idInclusion, nbPieces, balcon, statut, noPorte, description)
VALUES (40,5,5,4,FALSE,'Vacant',30,NULL);
-- Insertion d'un locataire
INSERT INTO Gestion.Locataires(idLocataire, idLogement, langue, noTelephone, prenom, nom, courriel, modePaiement, solde)
VALUES (80,40,'Français','4371152227','Test','Locataire','locataire100@test.com','Comptant',NULL);
-- Insertion d'un bail avec date de fin passée
INSERT INTO Gestion.Baux(idBail,idLocataire,dateDebut,dateFin,loyer)
VALUES (400,80,'2024-01-01','2024-12-31',1000.00);

-- TEST doit retourner FALSE
SELECT Gestion.f_bailActif(400);
```

1)

```
--Test 2 Bail actif (retourne TRUE)
-- Insertion d'un logement test
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (41,5,5,4,FALSE,'Vacant',31,NULL);
-- Insertion d'un locataire
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (81,41,'Français','4371152227','Test','Actif','locataire101@test.com','Comptant',NULL);
-- Insertion d'un bail avec date de fin passée
INSERT INTO Gestion.Baux(idBail,idLocataire,dateDebut,dateFin,loyer)
VALUES (401,81,'2025-12-01','2026-12-01',1200.00);

SELECT Gestion.f_bailActif(401);
```

2)

Exemple

1) Test avec un bail expiré

	f_bailactif boolean
1	false

2) Test avec un bail actif

	f_bailactif boolean
1	true

Fonction 6 Gestion.f_getBail(unLocataire)

Description

Cette fonction prend un identifiant de locataire et retourne l'identifiant de son bail, s'il n'y a pas de bail associé, ça retourne NULL.

Voici les scripts permettant de tester cette fonction.

```
--Test 1 : locataire avec bail existant
--Insertion d'un logement de test
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (83,5,5,4,FALSE,'Vacant',40,NULL);
--Insertion d'un locataire de test
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (83,83,'Français','4371152227','Test','AvecBail','locataire200@test.com','Comptant',NULL);
--Insertion d'un bail de test
INSERT INTO Gestion.Baux(idBail,idLocataire,dateDebut,dateFin,loyer)
VALUES (500,83,'2025-12-01','2026-12-01',1000.00);
--Test doit retourner 500
1) SELECT Gestion.f_getBail(83);

--Test 2 : locataire sans bail
--Insertion d'un logement de test
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (84,5,5,4,FALSE,'Vacant',41,NULL);
--Insertion d'un locataire de test
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (501,84,'Français','4371152227','Test','SansBail','locataire201@test.com','Comptant',NULL);
--Test doit retourner NULL
2) SELECT Gestion.f_getBail(84);
```

Exemples

1) Test avec un bail existant

	f_getbail integer
1	500

2) Test avec un bail inexistant

	f_getbail integer
1	[null]

Fonction 7 : Gestion.f_locataireBailActif(NEW)

Description

Le Trigger de cette fonction est déclenché avant chaque insertion sur la table Gestion.PaiementsLocataire et ce, sur chaque ligne. La fonction vérifie qu'avant chaque paiement de locataire, le locataire en question possède un bail, s'il n'en possède pas, elle RAISE EXCEPTION.

Voici le script qui permet de faire les tests.

```
--Test 1 : locataire avec bail actif (le trigger ne bloque rien)
--Insertion d'un logement test
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (87,5,5,4,FALSE,'Vacant',50,NULL);
--Insertion d'un locataire test
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (87,87,'Français','4371152227','Test','Actif','locataire300@test.com','Comptant',NULL);
--Insertion d'un bail test
INSERT INTO Gestion.Baux(idBail,idLocataire,dateDebut,dateFin,loyer)
VALUES (87,87,'2025-12-01','2026-12-01',1000.00);
--Insertion d'un paiement test
INSERT INTO Gestion.PaiementsLocataire(idPaiementsLocataire,idLocataire,datePaiement,montant,periode)
VALUES (87,87,'2025-12-05',500.00,'Décembre');
1)

--Test 2 : locataire sans bail (le trigger bloque avec exception)
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (88,5,5,4,FALSE,'Vacant',51,NULL);

INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (88,88,'Français','4371152227','Test','SansBail','locataire301@test.com','Comptant',NULL);

-- Tentative de paiement sans bail actif → doit lever l'exception
INSERT INTO Gestion.PaiementsLocataire(idPaiementsLocataire,idLocataire,datePaiement,montant,periode)
VALUES (78,88,'2025-12-05',500.00,'Décembre');
2)
```

Exemples

1) Test valide

```
INSERT 0 1
```

```
Query returned successfully in 64 msec.
```

2) Test sans bail

```
ERROR: Le locataire 88 ne possède pas de bail actif
CONTEXT: PL/pgSQL function gestion.f_locatairebailactif() line 10 at RAISE

SQL state: P0001
```

Fonction 8 Gestion.f_updateSoldePaiement(NEW)

Description

Le trigger de cette fonction exécute celle-ci après chaque INSERT ou UPDATE sur la table Gestion.PaiementsLocataire sur chaque colonne. La fonction update le solde du locataire selon le montant du paiement que ce dernier a fait.

Voici les scripts qui permettent de tester cette fonction. À noter que ce trigger ne bloque rien, puisqu'il y a déjà un trigger qui vérifie que le locataire possède un bail.

Voici le script permettant de tester la fonction.

```
--Test update solde après paiement
--Insertion de logement test
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (90,5,5,4,FALSE,'Vacant',60,NULL);
--Insertion de locataire test
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (90,90,'Français','4371152227','Test','Paiement','locataire400@test.com','Comptant',NULL);
--Insertion d'un bail test
INSERT INTO Gestion.Baux(idBail,idLocataire,dateDebut,dateFin,loyer)
VALUES (90,90,'2025-12-01','2026-12-01',1000.00);
--vérification du solde qui doit retourner 1000.00
SELECT Gestion.f_getSolde(90);

--Insertion d'un paiement
INSERT INTO Gestion.PaiementsLocataire(idPaiementsLocataire,idLocataire,datePaiement,montant,periode)
VALUES (90,90,'2025-12-05',500.00,'Novembre');
--Test doit retourner 500.00
SELECT Gestion.f_getSolde(90);
```

Exemples

1) Test valide

	f_getsolde
	numeric
1	1000.00

Le solde est à 1000.00\$ car il s'agit du montant du loyer avant le paiement

	f_getsolde
	numeric
1	500.00

Après l'insertion d'un paiement de 500.00\$, le solde est modifié pour déduire le montant du paiement à son solde.

Fonction 9 : Gestion.f_genererProfitProprietaire(unePeriode)

Description

Cette fonction prend en paramètre la période d'un paiement, c'est-à-dire un mois de l'année. Elle retourne l'identifiant de chaque propriétaire avec le revenu de celui-ci pour le mois entré en paramètre. Il ne s'agit que des revenus engendrés par les paiements des locataires.

Voici le script qui permet de tester cette fonction.

```
--Test fonction generer profit
--Test mois de janvier retourne 0.00 pour chaque propriétaire
SELECT idProprietaire, CashFlow
FROM Gestion.f_genererProfitProprietaire('Janvier');
--Insertion du logement
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion, nbPieces,balcon,statut,noPorte,description)
VALUES (200,5,3,4,FALSE,'Vacant',200,NULL);
--Insertion d'un locataire qui fera 4 paiement
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (300,200,'Français','4371152227','Test','Janvier','locataire300@test.com','Comptant',4000.00);
-- Insertion de son bail test
INSERT INTO Gestion.Baux(idBail,idLocataire,dateDebut,dateFin,loyer)
VALUES (1000,300,'2025-01-01','2025-12-31',1000.00);
--Insertion de ces 4 paiements
INSERT INTO Gestion.PaiementsLocataire(idPaiementsLocataire,idLocataire,datePaiement,montant,periode)
VALUES (500,300,'2025-01-05',500.00,'Janvier'),
(501,300,'2025-01-10',600.00,'Janvier'),
(502,300,'2025-01-15',700.00,'Janvier'),
(503,300,'2025-01-20',800.00,'Janvier');
--Test le propriétaire 5 devrait avoir 2600.00$ de revenu.
SELECT idProprietaire, CashFlow
FROM Gestion.f_genererProfitProprietaire('Janvier');
```

Exemple

1) Test valide

	idproprietaire integer	cashflow numeric
1	1	0.00
2	2	0.00
3	3	0.00
4	4	0.00
5	5	0.00

En vérifiant avant l'entrée des paiements, on voit qu'il n'y a eu aucun cashflow.

	idproprietaire integer	cashflow numeric
1	1	0.00
2	2	0.00
3	3	0.00
4	4	0.00
5	5	2600.00

À la suite de l'insertion des 4 paiements totalisant 2600\$, le cashflow est bel et bien de 2600\$.

Fonction 10 : Gestion.f_afficherDureeBail(unBail)

Description

Cette fonction prend en paramètre un identifiant de bail et retourne la durée de ce bail sous la forme «dateDebut au dateFin».

Voici le script permettant de tester cette fonction.

```
--Le bail n'existe pas
SELECT* FROM Gestion.f_afficherDureeBail(905);
--Test afficher durée bail
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (70,5,5,4,FALSE,'Vacant',70,NULL);
--Insertion locataire de test
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (500,70,'Français','4371152227','Test','Durée','locataire500@test.com','Comptant',NULL);
--Insertion bail de test
INSERT INTO Gestion.Baux(idBail,idLocataire,dateDebut,dateFin,loyer)
VALUES (905,500,'1999-01-01','2042-12-31',1000.00);
--Test doit retourner 1999-01-01 au 2042-12-31
SELECT Gestion.f_afficherDureeBail(905);
```


Exemples

1) *Test avec un bail qui n'existe pas*

	f_afficherdureebail character varying
1	[null]

2) *Test avec un bail valide*

	f_afficherdureebail character varying
1	1999-01-01 au 2042-12-31

Fonction 11 : Gestion.afficherPrenomLocataire(unLocataire)

Description

Cette fonction prend en paramètre l'identifiant d'un locataire et retourne une chaîne de caractères de son prénom suivi de son nom.

Voici le script permettant de tester cette fonction.

```
--Test sur un locataire qui n'existe pas
SELECT Gestion.afficherPrenomLocataire(502);
--Insertion du logement pour le test
INSERT INTO Gestion.Logements(idLogement,idImmeuble,idInclusion,nbPieces,balcon,statut,noPorte,description)
VALUES (72,5,5,4,FALSE,'Vacant',70,NULL);
--Insertion locataire de test
INSERT INTO Gestion.Locataires(idLocataire,idLogement,langue,noTelephone,prenom,nom,courriel,modePaiement,solde)
VALUES (502,72,'Français','4371152227','JeSuisUnPrénom','EtMoiJeSuisUnNom','PrénomNom@test.com','Comptant',NULL);
--Doit retourner "JeSuisUnPrénom EtMoiJeSuisUnNom"
SELECT Gestion.afficherPrenomLocataire(502);
```

Exemples

1) *Test sur un locataire qui n'existe pas*

	afficherprenomlocataire character varying
1	[null]

2) Test sur un locataire qui existe

	afficherprenomlocataire character varying
1	JeSuisUnPrénom EtMoiJeSuisUnNom

FONCTION 12 Gestion.afficherAdresse(unAdresse)

Description

Cette fonction prend en paramètre l'identifiant d'une adresse et retourne celle-ci sous forme de chaîne de caractères (numéro civique rue, ville code postal).

Voici le script qui permet de tester cette fonction.

```
--Test sur une adresse qui n'existe pas
SELECT * FROM Gestion.afficherAdresse(999);

--Insertion d'une adresse pour le test
INSERT INTO Gestion.Adresses (idAdresse, noCivique, rue, ville, codePostal) VALUES
(999, 1234, 'Rue du test', 'La ville des tests', 'A1A1A1');
--Test sur l'adresse
SELECT * FROM Gestion.afficherAdresse(999);
```

Exemples

1) Test avec un adresse qui n'existe pas

	afficheradresse character varying
1	[null]

2) Test sur une adresse valide

	afficheradresse character varying
1	1234 Rue du test, La ville des tests A1A1A1

Fonction 13 : Gestion.f_liste_des_baux_solde()

Description :

Cette fonction retourne une table qui contient tous les propriétaires dont au moins un logement possède un locataire avec un bail actif. On y retrouve plusieurs informations utiles, comme l'adresse de l'immeuble, les soldes et les loyers actuels.

Voici le script permettant de tester cette fonction.

```
SELECT * FROM Gestion.f_liste_des_baux_solde();
INSERT INTO Gestion.Adresses (idAdresse, noCivique, rue, ville, codePostal) VALUES
(100, 101, 'boulevard de Portland', 'Sherbrooke', 'J1H 2K7'),
(200, 2500, 'boulevard de l'université', 'Sherbrooke', 'J1K 2R1'),
(300, 201, 'rue King O.', 'Sherbrooke', 'J1H 1P7'),
(400, 301, 'rue du Plaisir', 'Québec', 'G1R 2B5'),
(500, 401, 'rue du Soleil', 'Montréal', 'H2X 1Y7'),
(600, 1550, 'rue Wellington Sud', 'Sherbrooke', 'J1H 5C5'),
(700, 875, 'rue Galt Ouest', 'Sherbrooke', 'J1H 1Z8'),
(800, 3450, 'boulevard Saint-François', 'Sherbrooke', 'J1E 2B9'),
(900, 125, 'rue Alexandre', 'Sherbrooke', 'J1H 2N4'),
(1000, 680, 'avenue du Parc', 'Montréal', 'H2V 4E8');
--Insertion des propriétaires
INSERT INTO Gestion.Proprietaires (idProprietaire, idAdresse, nbPortesPro, nbImmeublesPro) VALUES
(500, 100, 2, 1),
(501, 200, 3, 1);
--Insertion du type de proprio
INSERT INTO Gestion.Compagnies (idProprietaire, noTelephone, courriel, nom) VALUES
(500, '4501234567', 'AdresseCompagnie@usherbrooke.com', 'TestCompagnie');
--Insertion de l'autre type de proprio
INSERT INTO Gestion.Particuliers (idProprietaire, noTelephone, courriel, nom, prenom) VALUES
(501, '9876543210', 'Particulier@gestionPro.com', 'JesuisUn', 'Particulier');
--Insertion des regles et inclusion
INSERT INTO Gestion.Regles VALUES
(600, true, true, true, true);
INSERT INTO Gestion.Inclusions VALUES
(700, true, true, true, true);
--Insertion des Immeubles
INSERT INTO Gestion.Immeubles VALUES
(100, 600, 500, 600, 2, 2, 'TEST'),
(200, 700, 500, 600, 2, 2, 'TEST'),
(300, 800, 501, 600, 2, 2, 'TEST'),
(400, 900, 501, 600, 2, 2, 'TEST'),
(500, 100, 501, 600, 2, 2, 'TEST');
```

```
--Insertion des logements
INSERT INTO Gestion.Logements VALUES
(1000,100,700,4,False,'Loué',1,NULL),
(2000,100,700,4,False,'Loué',2,NULL),
(3000,200,700,4,False,'Loué',1,NULL),
(4000,200,700,4,False,'Loué',2,NULL),
(5000,300,700,4,False,'Loué',1,NULL),
(6000,300,700,4,False,'Loué',2,NULL),
(7000,400,700,4,False,'Loué',1,NULL),
(8000,400,700,4,False,'Vacant',2,NULL),
(9000,500,700,4,False,'Loué',1,NULL),
(10000,500,700,4,False,'Vacant',2,NULL);

--Insertion des Locataires
INSERT INTO Gestion.Locataires VALUES
(1000,1000,'Français','1111111111','Locataire1','Test','Locataire1@Test.com','Interac',NULL),
(2000,2000,'Français','2222222222','Locataire2','Test','Locataire2@Test.com','Interac',NULL),
(3000,3000,'Français','3333333333','Locataire3','Test','Locataire3@Test.com','Interac',NULL),
(4000,4000,'Français','4444444444','Locataire4','Test','Locataire4@Test.com','Interac',NULL),
(5000,5000,'Français','5555555555','Locataire5','Test','Locataire5@Test.com','Interac',NULL),
(6000,6000,'Français','6666666666','Locataire6','Test','Locataire6@Test.com','Interac',NULL),
(7000,7000,'Français','7777777777','Locataire7','Test','Locataire7@Test.com','Interac',NULL),
(8000,9000,'Français','8888888888','Locataire8','Test','Locataire8@Test.com','Interac',NULL);

--Insertion des baux
INSERT INTO Gestion.Baux VALUES
(10000,1000,'2024-07-01','2025-06-30',100.00),
(20000,2000,'2025-07-01','2026-06-30',200.00),
(30000,3000,'2025-07-01','2026-06-30',300.00),
(40000,4000,'2025-07-01','2026-06-30',400.00),
(50000,5000,'2025-07-01','2026-06-30',500.00),
(60000,6000,'2025-07-01','2026-06-30',600.00),
(70000,7000,'2025-07-01','2026-06-30',700.00),
(80000,8000,'2025-07-01','2026-06-30',800.00);

--Insertion des paiements de locataires
INSERT INTO Gestion.PaiementsLocataire VALUES
(100,4000,'2025-12-01',400.00,'Novembre','Aucun retard'),
(200,2000,'2025-12-01',200.00,'Novembre','Aucun retard'),
(300,3000,'2025-12-01',300.00,'Novembre','Aucun retard');

--Test avec des données
SELECT * FROM Gestion.f_liste_des_baux_solde();
```

Exemples

1) Test avec des tables vides

idproprietaire	adresse	duree_du_bail	idbail	locataire	numero_tel	loyer	modepaiement	solde
integer	character varying	character varying	integer	character varying	character varying	numeric	typepaiement	numeric

2) Test avec des données

	idproprietaire	adresse	duree_du_bail	idbail	locataire	numero_tel	loyer	modepaiement	solde
	integer	character varying	character varying	integer	character varying	character varying	numeric	typepaiement	numeric
1	500	1550 rue Wellington Sud, Sherbrooke J1H 5C5	2025-07-01 au 2026-06-30	20000	Locataire2 Test	2222222222	200.00	Interac	0.00
2	500	875 rue Galt Ouest, Sherbrooke J1H 1Z8	2025-07-01 au 2026-06-30	30000	Locataire3 Test	3333333333	300.00	Interac	0.00
3	500	875 rue Galt Ouest, Sherbrooke J1H 1Z8	2025-07-01 au 2026-06-30	40000	Locataire4 Test	4444444444	400.00	Interac	0.00
4	501	3450 boulevard Saint-François, Sherbrooke J1E 2B9	2025-07-01 au 2026-06-30	50000	Locataire5 Test	5555555555	500.00	Interac	500.00
5	501	3450 boulevard Saint-François, Sherbrooke J1E 2B9	2025-07-01 au 2026-06-30	60000	Locataire6 Test	6666666666	600.00	Interac	600.00
6	501	125 rue Alexandre, Sherbrooke J1H 2N4	2025-07-01 au 2026-06-30	70000	Locataire7 Test	7777777777	700.00	Interac	700.00
7	501	101 boulevard de Portland, Sherbrooke J1H 2K7	2025-07-01 au 2026-06-30	80000	Locataire8 Test	8888888888	800.00	Interac	800.00

PROCÉDURE 1) Gestion.p_soldeNouveauMois()

Description

Cette procédure parcourt tous les baux actifs, et récupère le montant du loyer de chacun d'eux. Ensuite, la procédure ajoute le montant de chaque loyer au solde du locataire y correspondant. Le but est de simuler un changement de mois, donc tous les soldes doivent être payés pour le nouveau mois.

Le script qui permet de tester cette fonction est le même que la fonction 12 au niveau de l'insertion des données. Donc, voici le script qui permet de tester cette procédure en tenant pour acquis que l'insertion des données de la fonction 12 est faite.

```
--Avant
SELECT * FROM Gestion.f_liste_des_baux_solde();
--Appel de la procédure
CALL Gestion.p_soldeNouveauMois();
--Après
SELECT * FROM Gestion.f_liste_des_baux_solde();
```

Exemples

1) Test avant l'appel de la fonction

	idproprietaire integer	adresse character varying	duree_du_bail character varying	idbail integer	locataire character varying	numero_tel character varying	loyer numeric	modepaiement typepaiement	solde numeric
1	500	1550 rue Wellington Sud, Sherbrooke J1H 5C5	2025-07-01 au 2026-06-30	20000	Locataire2 Test	2222222222	200.00	Interac	0.00
2	500	875 rue Galt Ouest, Sherbrooke J1H 1Z8	2025-07-01 au 2026-06-30	30000	Locataire3 Test	3333333333	300.00	Interac	0.00
3	500	875 rue Galt Ouest, Sherbrooke J1H 1Z8	2025-07-01 au 2026-06-30	40000	Locataire4 Test	4444444444	400.00	Interac	0.00
4	501	3450 boulevard Saint-François, Sherbrooke J1E 2B9	2025-07-01 au 2026-06-30	50000	Locataire5 Test	5555555555	500.00	Interac	500.00
5	501	3450 boulevard Saint-François, Sherbrooke J1E 2B9	2025-07-01 au 2026-06-30	60000	Locataire6 Test	6666666666	600.00	Interac	600.00
6	501	125 rue Alexandre, Sherbrooke J1H 2N4	2025-07-01 au 2026-06-30	70000	Locataire7 Test	7777777777	700.00	Interac	700.00
7	501	101 boulevard de Portland, Sherbrooke J1H 2K7	2025-07-01 au 2026-06-30	80000	Locataire8 Test	8888888888	800.00	Interac	800.00

2) Test après l'appel de la fonction

	idproprietaire integer	adresse character varying	duree_du_bail character varying	idbail integer	locataire character varying	numero_tel character varying	loyer numeric	modepaiement typepaiement	solde numeric
1	500	1550 rue Wellington Sud, Sherbrooke J1H 5C5	2025-07-01 au 2026-06-30	20000	Locataire2 Test	2222222222	200.00	Interac	200.00
2	500	875 rue Galt Ouest, Sherbrooke J1H 1Z8	2025-07-01 au 2026-06-30	30000	Locataire3 Test	3333333333	300.00	Interac	300.00
3	500	875 rue Galt Ouest, Sherbrooke J1H 1Z8	2025-07-01 au 2026-06-30	40000	Locataire4 Test	4444444444	400.00	Interac	400.00
4	501	3450 boulevard Saint-François, Sherbrooke J1E 2B9	2025-07-01 au 2026-06-30	50000	Locataire5 Test	5555555555	500.00	Interac	1000.00
5	501	3450 boulevard Saint-François, Sherbrooke J1E 2B9	2025-07-01 au 2026-06-30	60000	Locataire6 Test	6666666666	600.00	Interac	1200.00
6	501	125 rue Alexandre, Sherbrooke J1H 2N4	2025-07-01 au 2026-06-30	70000	Locataire7 Test	7777777777	700.00	Interac	1400.00
7	501	101 boulevard de Portland, Sherbrooke J1H 2K7	2025-07-01 au 2026-06-30	80000	Locataire8 Test	8888888888	800.00	Interac	1600.00

On remarque qu'après l'appel de la procédure, chaque loyer a été ajouté au solde des locataires actifs.

Contrainte verif_date_valide_[table]

Description

Les contraintes CHECK qui se retrouvent dans le fichier de création de tables ont tous le même objectif. Soit d'empêcher une date de fin d'être avant la date de début.

Il y a 3 contraintes en tout et elles possèdent exactement le même code. Pour cette raison, on limitera les tests sur une seule d'entre elles.

Voici le script permettant de tester ces contraintes.

```
--CONTRAINTE

--TEST 1 insertion valide
INSERT INTO Gestion.Travaux (idTravail, dateDebut, dateFin, coutEstime, coutFinal, statut, description)
VALUES (200, '1900-01-01', '2000-01-01', '10.00', '10.00', 'Prévu', NULL);
--AUCUN BLOCAGE ATTENDU

--TEST 2 dateFin null INSERTION VALIDE
INSERT INTO Gestion.Travaux (idTravail, dateDebut, dateFin, coutEstime, coutFinal, statut, description)
VALUES (201, '1900-01-01', NULL, '10.00', '10.00', 'Prévu', NULL);
--AUCUN BLOCAGE ATTENDU

--TEST 3 INSERTION INVALIDE
INSERT INTO Gestion.Travaux (idTravail, dateDebut, dateFin, coutEstime, coutFinal, statut, description)
VALUES (202, '2000-01-01', '1900-01-01', '10.00', '10.00', 'Prévu', NULL);
--BLOCAGE ATTENDU
```

Exemples

1) Test insertion valide

```
INSERT 0 1
```

Query returned successfully in 88 msec.

2) Test avec dateFin NULL (valide)

```
INSERT 0 1
```

Query returned successfully in 99 msec.

3) Test avec dateFin > dateDebut (INVALIDE)

```
ERROR: new row for relation "travaux" violates check constraint "verif_date_valide_travaux"
Failing row contains (202, 2000-01-01, 1900-01-01, 10.00, 10.00, Prévu, null).
```

SQL state: 23514

Detail: Failing row contains (202, 2000-01-01, 1900-01-01, 10.00, 10.00, Prévu, null).