



## Guide Base

# *Appendice B* *Comparaison de HSQLDB et* *Firebird*

*Types de données et fonctions*

## Droits d'auteur

---

Ce document est protégé par Copyright © 2020 par l'Équipe de Documentation de LibreOffice. Les contributeurs sont nommés ci-dessous. Vous pouvez le distribuer et/ou le modifier sous les termes de la Licence Publique Générale GNU (<https://www.gnu.org/licenses/gpl.html>), version 3 ou ultérieure, ou de la Licence Creative Commons Attribution (<https://creativecommons.org/licenses/by/4.0/>), version 4.0 ou ultérieure.

Toutes les marques déposées citées dans ce guide appartiennent à leurs légitimes propriétaires.

## Contributeurs

### De cette édition

Pulkit Krishna

### Des éditions précédentes

Pulkit Krishna

Robert Großkopf

Jost Lange

Hazel Russman

Jean Hollis Weber

Jochen Schiffers

Randolph GAMO

## Traduction

Jean-Michel COSTE

## Relecteurs

Patrick Auclair

## Retour d'information

Veillez adresser tout commentaire ou suggestion concernant ce document à la liste de diffusion de l'Équipe de Documentation : [doc@fr.libreoffice.org](mailto:doc@fr.libreoffice.org)



### Note

Tout ce que vous envoyez à la liste de diffusion, y compris votre adresse mail et toute autre information personnelle incluse dans le message, est archivé publiquement et ne peut pas être effacé.

---

## Date de publication et version du logiciel

Publié en avril 2021. Basé sur LibreOffice 6.4.

## Table des matières

---

<b>Droits d'auteur.....</b>	<b>2</b>
Contributeurs.....	2
De cette édition.....	2
Des éditions précédentes.....	2
Traduction.....	2
Relecteurs.....	2
Retour d'information.....	2
Date de publication et version du logiciel.....	2
<b>Types de données et fonctions dans HSQLDB et Firebird.....</b>	<b>4</b>
<b>Fonctions intégrées et procédures stockées.....</b>	<b>4</b>
Numériques.....	5
Texte.....	9
Date/Heure.....	13
Connexion à la base de données.....	16
Système.....	17
Fonctions d'agrégat (en particulier avec GROUP BY).....	20
Variables (selon l'ordinateur).....	21
Opérateurs et déclarations.....	21
<b>Types de données pour l'éditeur de tables.....</b>	<b>22</b>
Entiers (Integer).....	22
Nombres à virgule flottante.....	23
Texte.....	23
Heure.....	23
Autres.....	24

## Types de données et fonctions dans HSQLDB et Firebird

---

Les tableaux de cette annexe sont tirés des manuels de HSQLDB et Firebird.

- <http://hsqldb.org/doc/guide/>
- <https://www.firebirdsql.org/en/documentation/>

Les informations relatives à HSQLDB interne sont les mêmes que dans l'annexe A de ce manuel.

La base de données interne supplémentaire Firebird est classée expérimentale.

Les tableaux fournissent d'abord une comparaison des fonctions, en particulier les fonctions qui sont populaires dans les forums, telles que :

- Ajouter un certain nombre de jours à une date (DATEADD)
- Valeurs de plusieurs lignes de données regroupées dans une seule ligne de données (LIST)

ne sont actuellement disponibles que dans la base de données externe Firebird, mais pas dans la version interne.

## Fonctions intégrées et procédures stockées

---

Les fonctions suivantes sont disponibles dans les bases de données intégrées. Malheureusement, une ou deux fonctions ne peuvent être utilisées que lorsque la commande **Exécuter directement l'instruction SQL** est choisie. Dans ce mode, les requêtes ne peuvent pas être modifiées.

Les fonctions qui fonctionnent avec l'interface utilisateur graphique sont marquées [Fonctionne dans l'IGU] (Interface Graphique Utilisateur). Les fonctions qui ne fonctionnent que dans les commandes SQL directes sont marquées [Ne fonctionne pas dans l'IGU].

## Numériques

Comme nous traitons ici de nombres à virgule flottante, assurez-vous de faire attention aux paramètres des champs dans les requêtes. La plupart du temps, l'affichage des décimales est restreint, ce qui peut entraîner un comportement inattendu dans certains cas. Par exemple, la colonne 1 peut afficher 0,00 mais contient en réalité 0,001 et la colonne 2, 1000. Si la colonne 3 est définie pour afficher Colonne 1 \* Colonne 2, elle affichera en fait 1.

<b>HSQLDB</b>		<b>Firebird</b>	
ABS(d)	Renvoie la valeur absolue d'un nombre. [Fonctionne dans l'IGU]	ABS(d)	
ACOS(d)	Renvoie l'arc cosinus. [Fonctionne dans l'IGU]	ACOS(d)	
ASIN(d)	Renvoie l'arc sinus. [Fonctionne dans l'IGU]	ASIN(d)	
ATAN(d)	Renvoie l'arc tangente. [Fonctionne dans l'IGU]	ATAN(d)	
ATAN2(a, b)	Renvoie l'arc tangente à l'aide de coordonnées, où a est la valeur de l'axe x, b la valeur de l'axe y. [Fonctionne dans l'IGU]	ATAN2(x, y)	
BITAND(a, b)	La forme binaire de a et la forme binaire de b doivent avoir 1 à la même position pour donner 1 dans le résultat. BITAND (3,5) donne 1 ; 0011 ET 0101 = 0001 [Fonctionne dans l'IGU]	BIN_AND(x, y [, z...])	
BITOR(a, b)	La forme binaire de a ou la forme binaire de b doit avoir 1 à la même position pour donner 1 dans le résultat. BITOR (3,5) donne 7 ; 0011 OU 0101 = 0111 [Fonctionne dans l'IGU]	BIN_OR(x, y [, z...])	
		BIN_SHL(n, exp)	n · 2 exp [Fonctionne dans l'IGU]

		BIN_SHR(n, exp)	n / 2 exp Le résultat est affiché sous la forme d'un entier arrondi. [Fonctionne dans l'IGU]
		BIN_XOR(x, y [, z...])	Soit la notation binaire de « a » ou la notation binaire de « b ». Doit avoir un « 1 » dans la même position pour donner « 1 » dans le résultat. BIN_XOR(3,5) returns 6 ; 0011 X OR 0101 = 011 0 [Fonctionne dans l'IGU]
CEILING(d)	Renvoie le plus petit nombre entier qui n'est pas inférieur à d. [Fonctionne dans l'IGU]	CEIL(d) CEILING(d)	
COS(d)	Renvoie le cosinus. [Fonctionne dans l'IGU]	COS(radians)	Les radians peuvent également être représentés à l'aide de l'angle (ici pour le cercle unité) : radians = (2 * PI () * angles / 360)
		COSH(d)	
COT(d)	Renvoie la cotangente. [Fonctionne dans l'IGU]	COT(d)	
DEGREES(d)	Convertit les radians en degrés. [Fonctionne dans l'IGU]		
EXP(d)	Renvoie e <sup>d</sup> (e : (2.718...)). [Fonctionne dans l'IGU]	EXP(d)	
FLOOR(d)	Renvoie le plus grand nombre entier qui n'est pas supérieur à d. [Fonctionne dans l'IGU]	FLOOR(d)	
LOG(d)	Renvoie le logarithme naturel en base e. [Fonctionne dans l'IGU]	LN(d)	
LOG10(d)	Renvoie le logarithme en base 10. [Fonctionne dans l'IGU]	LOG10(d)	

		LOG(base, d)	Renvoie à nouveau le logarithme dans n'importe quelle base.
MOD(a, b)	Renvoie le reste sous forme de nombre entier, dans la division de 2 nombres entiers. MOD (11,3) renvoie 2, car $3 * 3 + 2 = 11$ [Fonctionne dans l'IGU]	MOD(a, b)	
PI()	Renvoie $\pi$ (3,1415...). [Fonctionne dans l'IGU]	PI()	
POWER(a, b)	$a^b$ , POWER(2,3) = 8, car $2^3 = 8$ [Fonctionne dans l'IGU]	POWER(x, y)	
RADIANS(d)	Convertit les degrés en radians. [Fonctionne dans l'IGU]		
EDGE()	Renvoie un nombre aléatoire supérieur ou égal à 0,0 et inférieur à 1,0. [Fonctionne dans l'IGU]	EDGE()	
ROUND(a, b)	Arrondit a à b décimales. [Fonctionne dans l'IGU]	ROUND(d [, places])	Arrondit après le nombre de chiffres spécifié à partir du point décimal. ROUND (123.45, 1) renvoie 123.50 ROUND (123.45, -2) renvoie 100.00 [Fonctionne dans l'IGU]
ROUNDMAGIC(d)	Résout les problèmes d'arrondi résultant de l'utilisation de nombres à virgule flottante. 3.11-3.1-0.01 n'est pas exactement 0, mais est affiché comme 0 dans l'interface graphique. ROUNDMAGIC en fait une valeur zéro réelle. [Fonctionne dans l'IGU]		
SIGN(d)	Renvoie -1 si d est inférieur à 0, 0 si d est égal à 0 et 1 si d est supérieur à 0. [Fonctionne dans l'IGU]	SIGN(d)	

SIN(A)	Renvoie le sinus d'un angle en radians. [Fonctionne dans l'IGU]	SIN(radians)	
		Sinh(d)	
SQRT(d)	Renvoie la racine carrée. [Fonctionne dans l'IGU]	SQRT(d)	
TAN(A)	Renvoie la tangente d'un angle en radians. [Fonctionne dans l'IGU]	TAN(radians)	
		TANH(d)	
TRUNCATE (a, b)	Tronque a à b décimales. TRUNCATE(2.37456.2) = 2.37 [Fonctionne dans l'IGU]	TRUNC(d[, jobs])	Définit sur 0 après le nombre de chiffres spécifié à partir du point décimal. TRUNC (123.45, 1) renvoie 123. 4 0 ROUND (123.45, -2) renvoie 100.00 [Fonctionne dans l'IGU]



<b>Texte</b>			
<b>HSQLDB</b>		<b>Firebird</b>	
ASCII(s)	Renvoie le code ASCII de la première lettre de la chaîne. [Fonctionne dans l'IGU]	ASCII_VAL ("s")	Renvoie la valeur numérique qui correspond au caractère saisi. [Fonctionne dans l'IGU]
BIT_LENGTH (str)	Renvoie la longueur de la chaîne de texte str en bits. [Fonctionne dans l'IGU]	BIT_LENGTH ("s")	
CHAR(c)	Renvoie la lettre qui appartient au code ASCII. Il ne s'agit pas seulement de lettres, mais aussi de caractères de contrôle. CHAR (13) crée un saut de ligne dans une requête, qui est visible dans les champs multilignes d'un formulaire ou dans les rapports. [Fonctionne dans l'IGU]	ASCII_CHAR (n)	Renvoie la lettre qui appartient au code ASCII. Il ne s'agit pas seulement de lettres, mais aussi de caractères de contrôle. [Fonctionne dans l'IGU]
CHAR_LENGTH (str)	Renvoie la longueur du texte en nombre de lettres. [Fonctionne dans l'IGU]	CHAR_LENGTH ("s") CHARACTER_LENGTH ("s")	
		CHAR_TO_UUID	Converts a 36-character Universally Unique Identifier (UUID) to a 16-octet UUID (outputs unreadable characters). Convertit un identifiant unique universel (UUID) de 36 caractères en un UUID de 16 octets (génère des caractères illisibles).
CONCAT (STR1, STR2)	Joint str1 + str2. [Fonctionne dans l'IGU]		
DIFFERENCE (s1, s2)	Renvoie la différence de son entre s1 et s2. Seul un nombre entier est émis. 0 signifie qu'ils sonnent de la même manière. Par exemple, « sert » et « serre » donnent 0, « pair » et « faire » donnent 1, « mer » et « mère » donnent 0. [Fonctionne dans l'IGU]		

HEXTORAW (s1)	Traduit le code hexadécimal en d'autres caractères. [Fonctionne dans l'IGU]		
INSERT(s, start, len, s2)	Renvoie une chaîne de texte, avec une partie du texte remplacée. En commençant à d une longueur l est découpée dans le texte s et remplacée par le texte s2.  INSERT ("Consolation", 3, 4, firm) convertit Consolation en Confirmation, où la longueur du texte inséré peut être supérieure à celle du texte supprimé sans causer de problèmes. Donc INSERT (« Bundesbahn », 3, 5, s et B) donne « Bus und Bahn ». [Fonctionne dans l'IGU]	OVERLAY ("s" PLACING 's2' FROM pos [FOR length])	Si la position de départ est définie de sorte qu'elle soit supérieure ou égale au texte réel « s », alors « s2 » est directement ajouté à « s ».  OVERLAY "Consolation" PLACING "firm" FROM 3 FOR 4) transforme "Consolation" en "Confirmation", où la longueur du texte inséré peut être plus longue que celle du texte coupé. Donc  OVERLAY ("Bundesbahn" PLACING 's und B' FROM 3 FOR 5) donne 'Bus und Bahn'. [ne fonctionne pas dans l'interface graphique]
LCASE(s)	Convertit une chaîne en minuscules. [Fonctionne dans l'IGU]		
LEFT(s, count)	Renvoie le nombre de caractères spécifié par compte à partir du début de la chaîne s. [Fonctionne dans l'IGU]	LEFT("s", length)	
LENGTH(s)	Renvoie la longueur de la chaîne s en caractères. [Fonctionne dans l'IGU]		
LOCATE (search, s [, start])	Renvoie la première correspondance du terme recherché dans la chaîne s. La correspondance est donnée sous forme de nombre de décalage : (1 = gauche, 0 = introuvable). La définition d'un point de départ dans la chaîne de texte est facultative. [Fonctionne dans l'IGU]	POSITION ('s2' IN "s")  POSITION ('s2', "s" [, Début])	

POSITION (<string expression> IN <string expression>)	Si la première chaîne est contenue dans la seconde, la position de la première chaîne sera renvoyée, sinon 0 sera affiché.  Cela pourrait être utilisé à la place d'une option de recherche avec LIKE. [fonctionne dans l'interface graphique]	POSITION ('s2' IN "s")  POSITION ('s2', "s" [, Début])	
LTRIM(s)	Supprime les espaces de début et les caractères non imprimables au début d'une chaîne de texte. [Fonctionne dans l'IGU]		
OCTET_LENGTH (str)	Renvoie la longueur d'une chaîne de texte en octets. Cela correspond à deux fois la longueur de la chaîne. [Fonctionne dans l'IGU]	OCTET_LENGTH (str)	Renvoie le nombre réel de caractères, en tenant compte des espaces. Le nombre dépend du jeu de caractères. UTF-8 a besoin de deux octets pour les caractères spéciaux.
RAWTOHEX(s1)	Convertit en hexadécimaux, inverse deHEXTORAW(). [Fonctionne dans l'IGU]		
REPEAT(s, count)	Répète le nombre de fois de la chaîne de texte. [Fonctionne dans l'IGU]		
REPLACE(s, replace, s2)	Remplace toutes les occurrences de replace existantes dans la chaîne de texte s par la chaîne s2. [Fonctionne dans l'IGU]	REPLACE("s", 's2', replace)	REPLACE ("Confirmation", "firm", "test") convertit "confirmation" en "Contestation". Si « s2 » n'apparaît pas dans « s », rien ne sera remplacé. Si N2 apparaît dans « s2 » ou en remplacement, le résultat est NULL.
		LPAD("s", Longueur totale [, caractères])	LPAD ("Hello", 8, '+') = +++ Hello  Place tous les caractères devant une chaîne jusqu'à ce que la longueur totale soit atteinte. Si la longueur totale est inférieure à la longueur de la chaîne, la chaîne est coupée à droite. Si le troisième paramètre reste vide, des espaces sont placés devant lui.

		RPAD("s", Longueur totale [, caractères])	RPAD ("Hello", 8 , '+') = Hello +++ RPAD ("Bonjour", 8, '+') = Bonjour +++ Place tous les caractères derrière une chaîne jusqu'à ce que la longueur totale soit atteinte. Si la longueur totale est inférieure à la longueur de la chaîne, la chaîne est coupée à droite. Si le troisième paramètre reste vide, des espaces sont placés derrière lui.
		REVERSE("s")	linverse complètement la chaîne. Cela peut être utile, par exemple, si vous souhaitez trier par terminaisons de caractères (par exemple, les fins de domaine).
RIGHT(s, longueur)	Opposé de LEFT ; renvoie le dernier nombre de caractères à la fin d'une chaîne de texte. [Fonctionne dans l'IGU]	RIGHT("s", Longueur)	
RTRIM(s)	Supprime tous les espaces et les caractères non imprimables à la fin d'une chaîne de texte. [Fonctionne dans l'IGU]		
SOUNDEX(s)	Renvoie un code à 4 caractères, correspondant au son de s. Correspond à la fonction DIFFERENCE(). [Fonctionne dans l'IGU]		
SPACE(count)	Renvoie le nombre d'espaces. [Fonctionne dans l'IGU]		
SUBSTR(s, start [, len])	Abréviation de SUBSTRING. [Fonctionne dans l'IGU]		

SUBSTRING (s, début [, longueur])	Renvoie le texte s à partir de la position de début (1 = gauche). Si longueur est omis, la chaîne entière est renvoyée. [Fonctionne dans l'IGU]	SUBSTRING ("s" FROM début [FOR longueur])	Renvoie la partie d'une chaîne à partir de la position de départ spécifiée dans FROM, éventuellement dans la longueur spécifiée dans FOR. Si, par exemple, "Roberta" apparaît dans le champ "Nom", SUBSTRING ("Name" FROM 3 FOR 3) entraîne la sous-chaîne "bert". [Fonctionne dans l'IGU]
TRIM ( [{LEADING   TRAILING   BOTH}] FROM <expression chaîne>	Les caractères spéciaux et les espaces qui ne peuvent pas être imprimés sont supprimés. [Fonctionne dans l'IGU]	TRIM ( [{LEADING   TRAILING   BOTH}] FROM <expression chaîne>	BOTH   LEADING   TRAILING : la norme ici est BOTH pour les deux côtés de « s ». s2 : Le caractère à supprimer. Par défaut, ce sont des espaces (' '), mais peuvent également être d'autres caractères. TRIM (TRAILING '!' FROM 'Hallo !') Renvoie "Hallo"
UCASE(s)	Convertit une chaîne en majuscules. [Fonctionne dans l'IGU]		
LOWER(s)	Comme LCASE(s). [Fonctionne dans l'IGU]	LOWER("s")	
UPPER(s)	Comme UCASE(s). [Fonctionne dans l'IGU]	UPPER("s")	
		UUID_TO_CHAR ("s")	Converts a 16-octet UUID to a 36-character ASCII format.
<b>Date/Heure</b>			
<b>HSQldb</b>		<b>Firebird</b>	

		<p>DATEADD (n DAY TO date)</p> <p>DATEADD (DAY, n, date)</p>	<p>n est un entier et peut également être négatif pour la soustraction.</p> <p>ANNÉE   MOIS   SEMAINE   JOUR   HOUR   MINUTE   SECOND   MILLISECOND doit être utilisé comme terme pour l'intervalle de temps.</p> <p>Utilisez un champ de date / date, un champ d'heure / heure ou un horodatage comme terme de date.</p>
<p>DATEDIFF (string, datetime1, datetime2)</p>	<p>Différence de date entre deux dates ou heures.</p> <p>L'entrée dans la chaîne décide dans quelle unité la différence est affichée : "ms" = "milliseconde", "ss" = "seconde", "mi" = "minute", "hh" = "heure", "dd" = ' jour ', ' mm '=' mois ', ' aa '=' année '.</p> <p>La version longue et la version courte de la chaîne peuvent être utilisées. [Fonctionne dans l'IGU]</p>	<p>DATEDIFF (DAY FROM date TO date)</p> <p>DATEDIFF (DAY, date, date)</p>	<p>See DATEADD.</p>
<p>EXTRACT ({YEAR  MONTH   DAY  HOUR  MINUTE  SECOND} FROM &lt;date or time value&gt;)</p>	<p>Peut remplacer de nombreuses fonctions de date et d'heure. Renvoie l'année, le mois, le jour, etc. à partir d'une valeur de date ou d'heure.</p> <p>EXTRACT (DAY FROM "date") indique le jour du mois. [Fonctionne dans l'IGU]</p>	<p>EXTRACT ({YEAR  MONTH   WEEK   DAY   WEEKDAY   YEARDAY   HOUR   MINUTE   SECOND   MILLISECOND } FROM &lt;date temps&gt;)</p>	<p>WEEKDAY Dimanche = 0</p> <p>YEARDAY 1er Janvier = 0</p> <p>WEEK 1ère semaine : minimum. 4 jours</p>
<p>DAY(date)</p>	<p>Renvoie le jour du mois (1-31). [Fonctionne dans l'IGU]</p>		
<p>DAYNAME (date)</p>	<p>Renvoie le nom anglais du jour. [Fonctionne dans l'IGU]</p>		
<p>DAYOFMONTH (date)</p>	<p>Renvoie le jour du mois (1-31). Synonyme de DAY(). [Fonctionne dans l'IGU]</p>		

DAYOFWEEK (date)	Renvoie le jour de la semaine sous forme de nombre (1 représente dimanche). [Fonctionne dans l'IGU]		
DAYOFYEAR (date)	Renvoie le jour de l'année (1-366). [Fonctionne dans l'IGU]		
HOUR(time)	Renvoie l'heure (0-23). [Fonctionne dans l'IGU]		
MINUTE(time)	Renvoie les minutes (0-59). [Fonctionne dans l'IGU]		
MONTH(date)	Renvoie le mois (1-12). [Fonctionne dans l'IGU]		
MONTHNAME (date)	Renvoie le nom anglais du mois. [Fonctionne dans l'IGU]		
NOW()	Renvoie la date actuelle et l'heure actuelle ensemble sous forme d'horodatage. Vous pouvez également utiliser CURRENT_TIMESTAMP. [Fonctionne dans l'IGU]		
QUARTER (date)	Renvoie le trimestre de l'année (1-4). [Fonctionne dans l'IGU]		
SECOND(time)	Renvoie la partie des secondes du temps (0-59). [Fonctionne dans l'IGU]		
WEEK(date)	Renvoie la semaine de l'année (1-53). [Fonctionne dans l'IGU]		
YEAR(date)	Renvoie la partie année d'une entrée de date. [Fonctionne dans l'IGU]		

<b>Connexion à la base de données</b>			
<b>HSQLDB</b>		<b>Firebird</b>	
DATABASE ( )	Renvoie le nom de la base de données à laquelle appartient cette connexion. [Fonctionne dans l'IGU]		
		CURRENT_TRANSACTION	SELECT CURRENT_TRANSACTION FROM RDB \$ DATABASE renvoie l'identifiant unique de la transaction sous forme d'entier.
		CURRENT_CONNECTION	SELECT CURRENT_CONNECTION FROM RDB \$ DATABASE renvoie une valeur entière pour la connexion actuelle.
		CURRENT_ROLE	SELECT CURRENT_ROLE FROM RDB \$ DATABASE reflète le rôle de l'utilisateur actuel. Si aucun rôle n'est défini, le résultat est NONE.
		RDB \$ SET_CONTEXT ('<namespace>', '<variable name>', value   NULL)	Espace de noms : USER_SESSION   USER_TRANSACTION Le nom de la variable peut avoir un maximum de 80 caractères et la valeur peut avoir un maximum de 255 caractères.
CURRENT_USER	Fonction standard SQL, synonyme de USER(). Il convient de noter qu'il n'y a pas de parenthèses ici. [Fonctionne dans l'IGU]	CURRENT_USER	
USER ( )	Renvoie le nom d'utilisateur de cette connexion. Le nom d'utilisateur est important si la base de données doit être convertie en base de données externe. [SQL direct – ne fonctionne pas avec l'IGU]	USER	



IDENTITY()	<p>Renvoie la dernière valeur d'un champ de valeur automatique, qui a été créé dans la connexion actuelle. Ceci est utilisé dans le codage de macro pour transférer une clé primaire dans une table pour devenir une clé étrangère pour une autre table.</p> <p>[Fonctionne dans l'IGU]</p>	<p>GEN_ID (generator name, &lt;step&gt;)</p>	<p>Les valeurs automatiques sont créées avec un générateur. La taille du pas doit être donnée ici sous la forme 1. En principe, toute valeur entière est possible.</p> <p>new.rec_id = gen_id (gen_recnum, 1)</p>
------------	---	--	---

## Système

<b>HSQLDB</b>		<b>Firebird</b>	
<p>IFNULL (exp, valeur)</p>	<p>Si exp est NULL, la valeur est renvoyée, sinon exp. Au lieu de cela, COALESCE() peut également être utilisé comme une extension. Exp et value doivent avoir le même type de données.</p> <p>IFNULL est une fonction importante lorsque les champs sont liés les uns aux autres par facture ou CONCAT. Le contenu du résultat serait NULL, même si une seule valeur est NULL.</p> <p>« Nom de famille »    ','    "prénom" entraînerait un champ vide pour les personnes qui, par exemple, ne disposent pas de l'entrée pour "prénom", c'est-à-dire NULL."Nom de famille »    IFNULL (','    « First Name », ") afficherait simplement « Nom de famille » à la place.[Fonctionne dans l'IGU]</p>		
<p>CASE WHEN (exp, v1, v2)</p>	<p>Si exp est vrai, v1 est renvoyé, sinon v2. CASE WHEN peut également être utilisé à la place.</p> <p>CASEWHEN ("a"&gt; 10, 'objectif atteint', 'continuer à pratiquer') renvoie 'objectif atteint' si le contenu du champ "a" est supérieur à 10.</p> <p>CASE WHEN fonctionne mieux avec l'interface graphique</p> <p>[Fonctionne dans l'IGU]</p>	<p>IIF (&lt;condition&gt;, v1, v2)</p>	

<p><b>CONVERT</b> (term, type)</p>	<p>Convertit le terme en un autre type de données. CONVERT ("a", DECIMAL (5,2)) fait du champ "a" un champ à 5 chiffres, dont 2 décimales. Si le nombre est trop grand, une erreur est générée. [Fonctionne dans l'IGU]</p>			
<p><b>CAST</b> (terme AS type)</p>	<p>Synonyme pour CONVERT () [Fonctionne dans l'IGU]</p>	<p><b>CAST</b> (terme AS type)</p>	<p><b>From</b></p>	<p><b>To</b></p>
			Types numeriques	Types numeriques [VAR] CHAR BLOB
			[VAR] CHAR BLOB	[VAR] CHAR BLOB Types numeriques DATE TIME TIMESTAMP
			DATE TIME	[VAR] CHAR BLOB TIMESTAMP
			TIMESTAMP	[VAR] CHAR BLOB DATE TIME
<p><b>COALESCE</b> (expr1, expr2, expr3,...)</p>	<p>Si expr1 n'est pas NULL, expr1 est affiché, sinon expr2 est vérifié, puis expr3, etc. Toutes les expressions doivent avoir au moins un type de données similaire. Il s'agit de la représentation alternative des nombres entiers et des nombres à virgule flottante, mais pas également d'une valeur de date ou d'heure. [Fonctionne dans l'IGU]</p>	<p><b>COALESCE</b> (expr1, expr2 [, expr3...]</p>		

NULLIF (v1, v2)	Si v1 est égal à v2, null est renvoyé, sinon la valeur de v1 est renvoyée. Les données doivent être de type comparable. [Fonctionne dans l'IGU]	NULLIF (v1, v2)	
CASE v1 WHEN v2 THEN v3 [ELSE v4] END	Si v1 est égal à v2, v3 est renvoyé. Sinon, v4 est retourné ou NULL, s'il n'y a pas de condition ELSE. [SQL Direct – ne fonctionne pas dans l'IGU]	DECODE (test expression , expression , resultat [, expression2 , Earnings2 ...] [, default expression])	DECODE (UPPER ("sexe"), "M", "homme", "F", "femme", "inconnu")
CASE WHEN expr1 THEN v1 [WHEN expr2 THEN v2] [ELSE v4] END	Si expr1 est vrai, v1 est retourné [Facultativement, d'autres cas peuvent être spécifiés]. Sinon, la v4 est reproduite ou NULL si aucun ELSE n'est formulé.  CASE WHEN DAYOFWEEK ("date") = 1 THEN "Dimanche" WHEN DAYOFWEEK ("date") = 2 THEN "Lundi"... END pourra afficher le jour de la semaine via SQL, qui n'est autrement disponible qu'en anglais dans la fonction. [Fonctionne dans l'IGU]		DECODE (EXTRACT (WEEK DAY FROM "date"), 0 , "Dimanche", 1 , Lundi', etc.)
		GEN_UUID ( )	Renvoie un ID unique sous la forme d'un jeu de caractères de 16 octets.
		HASH (s)	Renvoie la valeur de hachage d'une chaîne arbitrairement longue. Les valeurs de hachage des mêmes chaînes de caractères doivent être identiques.
		MAXVALUE (expr [, expr...])	Renvoie la valeur maximale d'une liste de valeurs. Fonctionne avec des chaînes, des valeurs numériques, des valeurs de date ou d'heure.

		MINVALUE (expr [, expr...])	Renvoie la valeur minimale d'une liste de valeurs. Fonctionne avec des chaînes, des valeurs numériques, des valeurs de date ou d'heure.
<b>Fonctions d'agrégat (en particulier avec GROUP BY)</b>			
<b>HSQLDB</b>		<b>Firebird</b>	
		MAX (expr)	Valeur maximale d'un champ dans une table.
		MIN (expr)	Valeur minimale d'un champ dans une table.
		LIST ([ALL   DISTINCT] 's, [s2'])	Connecte les champs de plusieurs enregistrements de données à un champ avec le terme de connexion correspondant 's2 '. [Fonctionne dans l'IGU]

<b>Variables (selon l'ordinateur)</b>			
<b>HSQLDB</b>		<b>Firebird</b>	
CURRENT_TIME	Synonyme de CURTIME(), standard SQL. [Fonctionne dans l'IGU]	CURRENT_TIME	Temps en heures, minutes et secondes. CURRENT_TIME (3) spécifie également les millisecondes.
CURTIME ( )	Renvoie l'heure actuelle.[Fonctionne dans l'IGU]		
CURRENT_TIME STAMP	Synonyme de NOW (), standard SQL. [Fonctionne dans l'IGU]	CURRENT_TIMESTAMP [(accuracy )]	Spécification de l'heure avec date et millisecondes. La précision peut être définie avec [0   1   2   3]. La norme est de 3 décimales. SELECT CURRENT_TIMESTAMP (2) FROM RDB \$ DATABASE renvoie l'horodatage avec des dixièmes et des centièmes de seconde (2 décimales).
NOW ( )	Renvoie la date et l'heure actuelles sous forme d'horodatage. CURRENT_TIMESTAMP peut également être utilisé à la place. [Fonctionne dans l'IGU]	CAST ("NOW" AS DATE   TIME   TIMESTAMP) or DATE "NOW"	« NOW », écrit seul, est compris comme une chaîne. Avec la conversion appropriée, il devient une date, une heure ou un horodatage (chacun avec 1/1000 s). La forme courte ne fonctionne pas dans l'interface graphique.
CURRENT_DATE	Synonyme de CURDATE (), standard SQL. [Fonctionne dans l'IGU]	CURRENT_DATE	
CURDATE ( )	Returns the current date. [Fonctionne dans l'IGU]		
<b>Opérateurs et déclarations</b>			
<b>HSQLDB</b>		<b>Firebird</b>	

'str1'    'str2'    'str3' or 'str1' + 'str2' + 'str3'	Concatène str1 + str2 + str3 ; alternative plus simple à CONCAT. [Fonctionne dans l'IGU]	's1'    s 2 [  ' s3 '...].....	Concatène s1, s2, etc. en une nouvelle chaîne [Fonctionne dans l'IGU]
		ALL	
		ANY / SOME	
		IN ( )	
		IS [NOT] DISTINCT FROM	Le résultat est "yes" or "no".
		NEXT VALUE FOR nom sequence	Voir GEN_ID (), mais n'autorise aucune étape autre que 1.
		SOME	

## Types de données pour l'éditeur de tables

Entiers (Integer)					
Type	Option	HSQLDB	Firebird	Range	Storage Space
Tiny integer	TINYINT	TINYINT		$2^8 = 256$   - 128 à + 127	1 octet
Small integer	SMALLINT	SMALLINT	SMALLINT	$2^{16} = 65536$   - 32768 à + 32767	2 octets
integer	INTEGER	INTEGER   INT	INTEGER	$2^{32} = 4294967296$   - 2147483648 à + 2147483647	4 octets
BigInt	BIGINT	BIGINT	BIGINT	$2^{64}$ (- $2^{63}$ à + $2^{63}$ )	8 octets

<b>Nombres à virgule flottante</b>					
<i>Type</i>	<i>Option</i>	<i>HSQldb</i>	<i>Firebird</i>	<i>Scope</i>	<i>Memory requirements</i>
Decimal	DECIMAL	DECIMAL	DECIMAL (n, m)	Illimité, via GUI à 50 chiffres, décimales réglables et fixes, précision exacte	2, 4 or 8 octet
Number	NUMERIC	NUMERIC	NUMERIC (n, m)	Illimité, via GUI à 50 chiffres, décimales réglables et fixes, précision exacte	2, 4 or 8 octet
Float	FLOAT	(DOUBLE est utilisé à la place)	FLOAT	$3.4 * 10^{-38}$ to $3.4 * 10^{38}$ réglable, pas exact, 7 décimales maximum	4 octet
Real	REAL	REAL			
Double	DOUBLE	DOUBLE [PRECISION]   FLOAT	DOUBLE PRECISION	$1,7 * 10^{-308}$ to $1,7 * 10^{308}$ réglable, pas exact, 15 décimales maximum	8 octets
<b>Texte</b>					
<i>Type</i>	<i>Option</i>	<i>HSQldb</i>	<i>Firebird</i>	<i>Scope</i>	<i>Memory requirements</i>
Text	VARCHAR	VARCHAR	VARCHAR (n)	Ajustable	Variable Firebird : 1 à 32767 octets
Text	VARCHAR_IGNORECASE	VARCHAR_IGNORECASE		Réglable, affecte le tri, ignore les différences entre les majuscules et les minuscules	variable
Text (fixed)	CHAR	CHAR   CHARACTER		Réglable, le reste du texte réel est rempli d'espaces	fixed
Memo	LONGVARCHAR	LONGVARCHAR	BLOB (BLOB SUB_TYPE 1)		variable Firebird : <32 GB
<b>Heure</b>					

<i>Type</i>	<i>Option</i>	<i>HSQldb</i>	<i>Firebird</i>	<i>Scope</i>	<i>Memory requirements</i>
Date	DATE	DATE	DATE		4 octets
Time	TIME	TIME	TIME	Firebird : 0:00 to 23:59,9999	4 octets
Date/Time	TIME STAMP	TIMESTAMP   DATE TIME	TIME STAMP	Réglable (HSQldb : 0, 6-6 signifie avec des millisecondes)	8 octets

<b>Autres</b>					
<i>Type</i>	<i>Option</i>	<i>HSQldb</i>	<i>Firebird</i>	<i>Scope</i>	<i>Memory requirements</i>
Oui Non	BOOLEAN	BOOLEAN   BIT			
Champ binaire (fixed)	BINARY	BINARY		Comme integer	fixe
Champ binaire	VARBINARY	VARBINARY		Comme integer	variable
Image	LONGVARBINARY	LONGVARBINARY	BLOB SUB_TYPE 0 BLOB SUB_TYPE binary	Comme integer	variable, destiné aux images plus grandes Firebird : <32 GB
OTHER	OTHER	OTHER   OBJECT			