

NORME INTERNATIONALE

**ISO/CEI
9075**

Deuxième édition
1989-04-01

Systèmes de traitement de l'information — Langage de bases de données SQL contenant l'amélioration sur l'intégrité

iTeh STANDARD PREVIEW

*Information processing systems — Database Language SQL with integrity
enhancement*
(standards.iteh.ai)

ISO/IEC 9075:1989

<https://standards.iteh.ai/catalog/standards/sist/7c7fa915-f99f-4ce9-8aa4-0e896d89ed84/iso-iec-9075-1989>



Numéro de référence
ISO/CEI 9075 : 1989 (F)

Avant-propos

L'ISO (Organisation internationale de normalisation) et la CEI (Commission électrotechnique internationale) forment ensemble un système consacré à la normalisation internationale considérée comme un tout. Les organismes nationaux membres de l'ISO ou de la CEI participent au développement de Normes internationales par l'intermédiaire des comités techniques créés par l'organisation concernée afin de s'occuper des différents domaines particuliers de l'activité technique. Les comités techniques de l'ISO et de la CEI collaborent dans des domaines d'intérêt commun. D'autres organisations internationales, gouvernementales ou non gouvernementales, en liaison avec l'ISO et la CEI participent également aux travaux.

Dans le domaine des technologies de l'information, l'ISO et la CEI ont créé un comité technique mixte, l'ISO/CEI JTC 1. Les projets de Normes internationales adoptés par le comité technique mixte sont soumis aux organismes nationaux pour approbation, avant leur acceptation comme Normes internationales. Les Normes internationales sont approuvées conformément aux procédures qui requièrent l'approbation de 75 % au moins des organismes nationaux votants.

ISO/IEC 9075:1989

La Norme internationale ISO/CEI 9075 a été élaborée par le comité technique mixte ISO/CEI JTC 1, *Technologies de l'information*.

Cette deuxième édition annule et remplace la première édition (ISO 9075 : 1987), qui comprend désormais l'additif 1.

L'attention des utilisateurs est attirée sur le fait que toutes les Normes internationales sont de temps en temps soumises à révision et que toute référence faite à une autre Norme internationale dans le présent document implique qu'il s'agit, sauf indication contraire, de la dernière édition.

Les annexes A, B, C, D, E et F de la présente Norme internationale sont données uniquement à titre d'information.

© ISO/CEI 1989

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

ISO/CEI Copyright Office • Case postale 56 • CH-1211 Genève 20 • Suisse

Version française tirée en 1990

Imprimé en Suisse

SOMMAIRE

	Page
1	Domaine et champ d'application 1
2	Références 3
3	Généralités 5
3.1	Organisation 5
3.2	Notation 5
3.3	Conventions 5
3.4	Conformité 6
4	Concepts 7
4.1	Ensembles 7
4.2	Types de données 7
4.3	Colonnes 8
4.4	Tables 8
4.5	Contraintes d'intégrité 8
4.6	Schémas 9
4.7	La base de données 9
4.8	Modules 9
4.9	Procédures 9
4.10	Paramètres 10
4.11	Langages de programmation normalisés 10
4.12	Curseurs 10
4.13	Instructions 11
4.14	Syntaxe d'incrustation 11
4.15	Privilèges 11
4.16	Transactions 12
5	Éléments communs 13
5.1	<caractère> 13
5.2	<littéral> 14
5.3	<jeton> 16
5.4	Noms 18
5.5	<type de donnée> 20
5.6	<spécification de valeur> et <spécification de cible> 22
5.7	<spécification de colonne> 24
5.8	<spécification de fonction sur ensemble> 25
5.9	<expression de valeur> 27
5.10	<prédicat> 29
5.11	<prédicat de comparaison> 30
5.12	<prédicat entre> 31
5.13	<prédicat dans> 32

5.14	<prédicat comme>	33
5.15	<prédicat nul>	35
5.16	<prédicat quantifié>	36
5.17	<prédicat existe>	37
5.18	<condition de recherche>	38
5.19	<expression de table>	40
5.20	<clause de>	41
5.21	<clause où>	43
5.22	<clause groupement par>	44
5.23	<clause ayant>	45
5.24	<sous-requête>	46
5.25	<spécification de requête>	48
6	Langage de définition de schéma	51
6.1	<schéma>	51
6.2	<définition de table>	52
6.3	<définition de colonne>	53
6.4	<clause par défaut>	54
6.5	<définition de contrainte de table>	55
6.6	<définition de contrainte d'unicité>	56
6.7	<définition de contrainte référentielle>	57
6.8	<définition de contrainte de contrôle>	59
6.9	<définition de vue>	60
6.10	<définition de privilège>	62
7	Langage de Module	65
7.1	<module>	65
7.2	<clause nom de module>	66
7.3	<procédure>	67
8	Langage de manipulation de données	71
8.1	<instruction fermeture>	71
8.2	<instruction exécution>	72
8.3	<déclaration de curseur>	73
8.4	<instruction suppression après positionnement>	76
8.5	<instruction suppression après recherches>	77
8.6	<instruction affectation>	78
8.7	<instruction insertion>	80
8.8	<instruction ouverture>	82
8.9	<instruction retour arrière>	83
8.10	<instruction sélection>	84
8.11	<instruction mise à jour après positionnement>	86
8.12	<instruction mise à jour après recherches>	88
9	Niveaux	91
	ANNEXE A — <incrustations SQL dans un programme hôte>	94
	ANNEXE B — <déclaration d'exception incrustée>	98
	ANNEXE C — <instruction SQL dans un programme COBOL>	100
	ANNEXE D — <instruction SQL dans un programme FORTRAN>	102
	ANNEXE E — <instruction SQL dans un programme Pascal>	104
	ANNEXE F — <instruction SQL dans un programme PL/I>	105
	Index	107

Systèmes de traitement de l'information — Langage de bases de données SQL contenant l'amélioration sur l'intégrité

1 Domaine et champ d'application

La présente norme définit la syntaxe et la sémantique de deux langages de base de données :

- a) un langage de définition de schéma (SQL-LDD), pour la déclaration des structures et des contraintes d'intégrité d'une base de données SQL ;
- b) un langage de définition de modules et un langage de manipulation de données (SQL-LMD), pour la déclaration des procédures de la base de données et des instructions exécutables d'un programme d'application base de données spécifique.

La présente norme définit les structures logiques de données et les opérations de base pour une base de données SQL. Elle fournit les fonctions nécessaires pour concevoir, accéder à, maintenir, contrôler et protéger la base de données.

La présente norme constitue un moyen de rendre portable les définitions de base de données et de programmes d'application entre mises en oeuvre conformes.

La présente norme spécifie deux niveaux et une caractéristique d'amélioration de l'intégrité. Le niveau 2 est le langage SQL complet sauf la caractéristique d'amélioration de l'intégrité. Le niveau 1 est le sous-ensemble du niveau 2 défini à l'article 9.

NOTE — *Il est prévu de développer des additifs à cette norme. Ces additifs porteront en particulier sur la gestion des transactions, la spécification de certaines règles définies par le réalisateur, les possibilités de manipulation des caractères et la prise en compte de jeux de caractères nationaux.*

La caractéristique d'amélioration de l'intégrité est un moyen de spécifier :

- 1) les contraintes référentielles entre les tables qui doivent être appliquées,
- 2) les contraintes de contrôle à appliquer aux lignes d'une table,
- 3) une valeur par défaut pour une colonne quand une ligne est insérée dans une table.

Les annexes de la présente norme définissent la syntaxe d'incrustation pour inclure des instructions du langage de manipulation de données SQL dans un autre programme d'application lui aussi normalisé. Une telle syntaxe d'incrustation est définie pour être une notation abrégée d'un programme d'application standardisé dans lequel les instructions SQL incrustées ont été remplacées par des «appels» explicites à des procédures base de données contenant les instructions SQL.

La présente norme s'applique aux réalisations existant dans un environnement pouvant inclure des langages de programmation d'application, des langages de requête utilisateur, des générateurs de rapport, des dictionnaires de données, des bibliothèques de programmes, des systèmes de communication distribués ainsi que des outils variés de conception de base de données, d'administration de données et d'optimisation de performances.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 9075:1989
<https://standards.iteh.ai/catalog/standards/sist/7c7fa915-f99f-4ce9-8aa4-0e896d89ed84/iso-iec-9075-1989>

2 Références

- ISO 1539, *Langages de programmation — FORTRAN*, (reprise par l'ISO de la norme ANSI X3.9).
- ISO 1989, *Langages de programmation — COBOL*, (reprise par l'ISO de la norme ANSI X3.23).
- ISO 6160, *Langages de programmation — PL/I*, (reprise par l'ISO de la norme ANSI X3.53).
- ISO 7185, *Langages de programmation — Pascal*, (reprise par l'ISO de la norme britannique 6192 pour le texte anglais).

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 9075:1989

<https://standards.iteh.ai/catalog/standards/sist/7c7fa915-f99f-4ce9-8aa4-0e896d89ed84/iso-iec-9075-1989>

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 9075:1989

<https://standards.iteh.ai/catalog/standards/sist/7c7fa915-f99f-4ce9-8aa4-0e896d89ed84/iso-iec-9075-1989>

3 Généralités

3.1 Organisation

Les articles 3.2, «Notation» et 3.3, «Conventions» définissent les notations et conventions utilisées dans la présente norme.

L'article 3.4, «Conformité» définit les critères de conformité.

L'article 4, «Concepts» définit les termes et présente les concepts utilisés dans la définition de SQL.

L'article 5, «Eléments communs» définit les éléments du langage apparaissant dans plusieurs parties du langage SQL.

L'article 6, «Langage de définition de schéma» définit les possibilités SQL de spécification de base de données.

L'article 7, «Langage de module» définit les modules et procédures SQL.

L'article 8, «Langage de manipulation de données» définit les instructions de manipulation de données SQL.

L'article 9, «Niveaux» définit les deux niveaux de SQL et la caractéristique d'amélioration de l'intégrité.

3.2 Notation

La notation syntaxique utilisée dans la présente norme est la BNF («Forme normale de Backus» ou «forme Backus-Naur»), avec les extensions suivantes :

- a) les crochets ([]) indiquent un élément optionnel ;
- b) les (...) indiquent un élément qui peut être répété une ou plusieurs fois ;
- c) les accolades { } regroupent des séquences d'éléments.

Dans la syntaxe BNF, un symbole de production <A> est déclaré «contenir» un symbole de production si apparaît quelque part dans l'expansion de <A>. Si <A> contient , alors est «contenu dans» <A>. Si <A> contient , alors <A> est le «contenant» <A>, symbole de production de .

3.3 Conventions

Les éléments syntaxiques de la présente norme sont spécifiés en termes de :

- a) *fonction* : une courte déclaration sur l'objet de l'élément ;
- b) *format* : une définition en BNF de la syntaxe de l'élément ;
- c) *règles syntaxiques* : les contraintes syntaxiques additionnelles non exprimées dans la BNF et auxquelles l'élément devra satisfaire ;
- d) *règles générales* : la spécification en séquence de l'effet à l'exécution de l'élément sur la base de données.

Dans les règles syntaxiques, le terme «doit» définit les conditions qui doivent être vérifiées, du point de vue de la syntaxe, par les éléments conformes du langage SQL. Le traitement du langage SQL non conforme au format ou aux règles syntaxiques est défini par le réalisateur.

Dans les règles générales, le terme «doit» définit les conditions testées au cours de l'exécution des instructions SQL. Si toutes les conditions sont vérifiées, alors l'instruction est exécutée avec succès et le paramètre SQLCODE est mis à un nombre positif défini. Si une au moins des conditions n'est pas vérifiée, alors l'instruction n'est pas exécutée avec succès et n'a pas d'effet sur la base de données ; le paramètre SQLCODE est mis à un nombre négatif défini par le réalisateur.

Une réalisation conforme n'est pas supposée exécuter la séquence exacte des actions définies par les règles générales, mais doit produire le même effet que cette séquence sur la base de données.

Le terme «effectivement» est utilisé dans les règles générales pour souligner les actions dont les effets peuvent être obtenus par d'autres moyens dans le cadre d'une réalisation.

Le terme «objet persistant» est utilisé pour caractériser des objets tels que <module>s et <schéma>s créés et détruits par des mécanismes définis par le réalisateur.

3.4 Conformité

La présente norme définit le langage SQL et les réalisations SQL conformes. Un langage SQL conforme doit se soumettre au format BNF ainsi qu'aux règles syntaxiques associées. Une réalisation SQL conforme doit exécuter le langage SQL conforme en respectant les règles générales.

Une réalisation déclarant la conformité SQL-LDD doit prendre en compte le SQL-LDD (<schéma>) au niveau 1 ou au niveau 2.

Une réalisation déclarant la conformité SQL-LMD doit prendre en compte soit au niveau 1, soit au niveau 2 :

- a) l'invocation directe des instructions du langage de manipulation de données SQL (<Instructions SQL>) ; et/ou
- b) le langage de définition de modules (<module>) ; et/ou
- c) une ou plusieurs :
 - 1) incrustation SQL dans COBOL (<incrustations SQL dans un programme COBOL>),
 - 2) incrustation SQL dans FORTRAN (<incrustations SQL dans un programme FORTRAN>),
 - 3) incrustation SQL dans Pascal (<incrustations SQL dans un programme Pascal>),
 - 4) incrustation SQL dans PL/I (<incrustations SQL dans un programme PL/I>).

Une réalisation déclarant la conformité SQL totale doit fournir une conformité SQL-LDD et SQL-LMD, soit au niveau 1, soit au niveau 2.

Une réalisation conforme peut fournir des facilités additionnelles ou des options non spécifiées par la présente norme. Une réalisation reste conforme même si elle prévoit des options de l'utilisateur pour exécuter un langage SQL non conforme ou si elle prévoit d'exécuter un langage SQL conforme d'une manière non conforme.

Une déclaration de conformité à la présente norme doit établir :

- a) quels types de conformité sont pris en compte parmi les suivants
 - 1) conformité totale à SQL, au niveau 1,
 - 2) conformité totale à SQL, au niveau 2,
 - 3) conformité au SQL-LDD, au niveau 1,
 - 4) conformité totale au SQL-LDD, au niveau 2,
 - 5) conformité au SQL-LMD, au niveau 1,
 - 6) conformité au SQL-LMD, au niveau 2 ;
- b) lesquelles des facilités suivantes sont mises en oeuvre :
 - 1) exécution directe des instructions du langage de manipulation de données SQL,
 - 2) modules langage (<module>),
 - 3) incrustations SQL dans COBOL (<incrustations SQL dans un programme COBOL>),
 - 4) incrustations SQL dans FORTRAN (<incrustations SQL dans un programme FORTRAN>),
 - 5) incrustations SQL dans Pascal (<incrustations SQL dans un programme Pascal>),
 - 6) incrustations SQL dans PL/I (<incrustations SQL dans un programme PL/I>) ;
- c) si la caractéristique d'amélioration de l'intégrité est mise en oeuvre.

La présente norme ne définit pas la méthode ou le moment de la liaison entre les programmes d'application et les composants du système de gestion de la base de données.

4 Concepts

4.1 Ensembles

Un ensemble est une collection non ordonnée d'objets distincts.

Un multi-ensemble est une collection non ordonnée d'objets non nécessairement distincts.

Une séquence est une collection ordonnée d'objets non nécessairement distincts.

La cardinalité d'une collection est le nombre d'objets de cette collection. A moins que l'inverse soit précisé, toute collection peut être vide.

4.2 Types de données

Un type de données est un ensemble de valeurs représentables. La représentation logique d'une valeur est un <littéral>. La représentation physique d'une valeur est laissée à la discrétion du réalisateur.

Une valeur est primitive, du fait qu'elle n'a aucune subdivision logique dans le cadre de cette norme. Une valeur est une valeur nulle ou une valeur non nulle.

Une valeur nulle est une valeur spéciale d'un type de donnée, définie par le réalisateur et est distincte de toutes les valeurs non nulles de ce type.

Une valeur non nulle est une chaîne de caractères ou un nombre. Une chaîne de caractères et un nombre ne sont pas des valeurs comparables.

4.2.1 Chaînes de caractères

Une chaîne de caractères est composée d'une séquence de caractères pris dans un ensemble de caractères défini par le réalisateur. Une chaîne de caractère a une longueur, qui est un entier positif définissant le nombre de caractères de la séquence.

Toutes les chaînes de caractères sont comparables. Une chaîne de caractères est identique à une autre si et seulement si elle est égale à cette chaîne de caractère en accord avec les règles de comparaison définies en 5.11, <prédicat de comparaison>.

4.2.2 Nombres

Un nombre est soit une valeur numérique exacte soit une valeur numérique approchée. Tous les nombres sont des valeurs comparables.

Une valeur numérique exacte a une précision et une graduation. La précision est un entier positif qui détermine le nombre de chiffres décimaux significatifs. La graduation est un entier non négatif. Une graduation de 0 indique que le nombre est un entier. Pour la graduation N, la valeur exacte est la valeur entière de la partie significative multipliée par 10 à la puissance N.

Une valeur approchée est constituée d'une mantisse et d'un exposant. La mantisse est une valeur numérique signée et l'exposant est un entier signé précisant la magnitude de la mantisse. Une valeur numérique approchée a une précision. La précision est l'entier positif qui définit le nombre de chiffre décimaux significatifs de la mantisse.

Quand une valeur numérique exacte est assignée à un élément de donnée ou à un paramètre représentant une valeur numérique exacte, une approximation de sa valeur préservant les premiers chiffres significatifs est représentée dans le type de donnée cible. La valeur est convertie pour avoir la précision et la graduation de la donnée cible.

Quand une valeur exacte ou approchée est assignée à un élément de donnée ou à un paramètre représentant une valeur approchée, une approximation de sa valeur est représentée dans le type de donnée cible. La valeur est convertie pour avoir la précision de la donnée cible.

4.3 Colonnes

Une colonne est un multi-ensemble de valeurs pouvant varier dans le temps. Toutes les valeurs d'une même colonne appartiennent au même type de donnée et sont des valeurs d'une même table. La valeur d'une colonne est le plus petit élément de donnée pouvant être choisi dans une table et le plus petit élément de donnée pouvant être mis à jour.

Une colonne a une description et une position ordinale dans une table. La description de la colonne inclut son type de donnée et une indication précisant si la colonne est contrainte à contenir uniquement des valeurs non nulles. La description d'une colonne chaîne de caractères définit son attribut de longueur. La description d'une colonne numérique approchée définit la précision de ses nombres. La description d'une colonne numérique exacte définit la précision et la graduation de ses nombres.

Une colonne nommée est une colonne d'une table nommée ou une colonne qui hérite de la description d'une colonne nommée. La description d'une colonne nommée inclut son nom.

4.4 Tables

Une table est un multi-ensemble de lignes. Une ligne est une séquence non vide de valeurs. Chaque ligne d'une même table a la même cardinalité et contient une valeur pour chaque colonne de cette table.

La nième valeur de chaque ligne d'une table est la valeur de la nième colonne de cette table. La ligne est le plus petit élément de donnée pouvant être ajouté à ou supprimé d'une table.

Le degré d'une table est le nombre de colonnes de cette table. A tout moment, le degré de la table est le même que la cardinalité de chacune de ses lignes et la cardinalité d'une table est la même que la cardinalité de chacune de ses colonnes.

Une table a une description. Cette description comprend une description de chacune de ses colonnes. Une table de base est une table nommée définie par une <définition de table>. La description d'une table de base comprend son nom.

Une table dérivée est une table dérivée directement ou indirectement d'une ou de plusieurs autres tables par évaluation d'une <expression de requête>. Les valeurs d'une table dérivée sont celles de la ou des tables d'origine lors de la dérivation.

Une table vue est une table dérivée nommée définie par une <définition de vue>. La description d'une table vue comprend son nom.

Une table est soit en mode mise à jour soit en mode lecture uniquement. Les opérations d'insertion, de mise à jour, de suppression sont permises pour les tables en mode mise à jour et non permises pour les tables en mode lecture uniquement.

Une table groupée est un ensemble de groupes obtenu par évaluation d'une <clause groupement par>. Un groupe est un multi-ensemble de lignes dans lequel toutes les valeurs des colonnes regroupées sont égales. Une table groupée peut être considérée comme une collection de tables. Les fonctions sur ensemble obtenue à partir d'une table peuvent agir sur les tables individuelles comprises dans la table groupée.

Une vue groupée est une table vue obtenue à partir d'une table sur ensemble groupée.

4.5 Contraintes d'intégrité

Les contraintes d'intégrité définissent les états valides de la base de données à l'aide de contraintes appliquées aux valeurs des tables de base.

Les contraintes d'intégrité sont effectivement testées à l'issue de l'exécution de chaque <instruction SQL>. Si la table de base associée à une contrainte d'intégrité ne satisfait pas à la contrainte d'intégrité, alors l'<instruction SQL> reste sans effet et le paramètre SQLCODE est mis à un nombre négatif défini par le réalisateur.

Une <définition de contrainte unique> implique que deux lignes d'une (ou de) colonne(s) spécifiée(s) d'une table n'aient pas la même valeur.

Une spécification NOT NULL implique qu'aucune des valeurs d'une colonne ne soit une valeur nulle.

Une <définition de contrainte référentielle> implique que, pour chaque ligne d'une table spécifiée, la «table de référence», les valeurs d'une (de) colonne(s) spécifiée(s) ont au moins une valeur nulle ou sont de même valeur qu'une (que des) colonnes spécifiées dans la ligne d'une autre table, dite «table référencée». La table de référence et la table référencée peuvent être les mêmes.

Une <définition de contrainte de contrôle> implique qu'une <condition de recherche> spécifiée ne donne un résultat faux dans aucune ligne d'une table.

4.6 Schémas

Un <schéma> est un objet persistant défini par le langage de définition de schéma. Il comprend une <clause d'autorisation de schéma>, toutes les <définition(s) de tables>, <définition(s) de vue>, et <définition(s) de privilège> connus du système pour un <identificateur d'autorisation> défini dans un environnement. Le concept d'environnement est défini par le réalisateur.

Les tables, vues et privilèges définis par un <schéma> sont considérés comme étant la «propriété de» ou «créés par» l'<identificateur d'autorisation> associé à ce <schéma>.

NOTE — Une réalisation peut fournir des facilités (telles que DROP TABLE, DROP VIEW, ALTER TABLE et REVOKE) permettant la définition des tables, des vues et des privilèges pour un <identificateur d'autorisation> donné, pouvant être créées, détruites et modifiées graduellement dans le temps. La présente norme s'intéresse uniquement aux <schémas> représentant les définitions connues par le système à un instant donné.

(standards.iteh.ai)

4.7 La base de données

La base de données est la collection de toutes les données définies par le <schéma> dans un environnement. Le concept d'environnement est défini par le réalisateur.

4.8 Modules

Un <module> est un objet persistant défini par le langage de module. Un <module> comprend, en option, un <nom de module>, une <clause de langage>, une <clause d'autorisation de module>, zéro ou plusieurs curseurs définis par <déclaration de curseur> et une ou plusieurs <procédure(s)>.

Un programme d'application est un segment de code exécutable comprenant éventuellement de multiple sous-programmes. Un seul <module> est associé à un programme d'application durant son exécution. Un programme d'application doit être associé à au moins un <module>. La façon dont cette association est définie, y compris la prise en compte possible d'instructions d'exécution définies par le réalisateur, est définie par le réalisateur.

4.9 Procédures

Une <procédure> est composée d'un <nom de procédure>, d'une séquence de <déclaration de paramètre(s)> et d'une simple <instruction SQL>.

Un programme d'application associé à un <module> peut faire référence à la ou les <procédure(s)> de ce <module> par une instruction d'«appel» spécifiant le <nom de procédure> de la <procédure>, et fournissant une séquence de valeurs de paramètres correspondant en nombre et en <type de donnée> à la <déclaration de paramètre(s)> de la <procédure>. L'appel d'une <procédure> entraîne l'exécution de <l'instruction SQL> qu'elle contient.

4.10 Paramètres

Un paramètre est déclaré dans une <procédure> par une <déclaration de paramètre>. La <déclaration de paramètre> spécifie le <type de donnée> de sa valeur. Un paramètre, soit suppose connue, soit fournit la valeur de l'argument correspondant dans l'appel de cette <procédure>.

4.10.1 Paramètre SQLCODE

Le paramètre SQLCODE est un paramètre entier spécial. Sa valeur correspond à un code statut qui indique soit qu'un appel de la <procédure> s'est effectué de manière satisfaisante, soit qu'une condition d'exception a été rencontrée durant l'exécution de cette <procédure>.

4.10.2 Paramètres indicateurs

Un paramètre indicateur est un paramètre entier qui est spécifié après un autre paramètre. Son premier usage est d'indiquer si la valeur que l'autre paramètre suppose ou fournit, est une valeur nulle.

4.11 Langages de programmation normalisés

La présente norme définit les actions de <procédure>(s) en <module>(s), lorsque ces <procédure>(s) sont appelées par des programmes se conformant à des langages de programmation normalisés spécifiques. Les termes «programme COBOL normalisé», «programme FORTRAN normalisé», «programme Pascal normalisé» et «programme PL/I normalisé» font référence à des programmes vérifiant respectivement les critères de conformité des normes citées à l'article 2.

4.12 Curseurs

Un curseur est spécifié par une <déclaration de curseur>.

Pour chaque <déclaration de curseur> dans un <module>, un curseur est effectivement créé quand une transaction (voir en 4.16 «Transactions»), faisant référence au <module>, est initialisée ; il est détruit quand cette transaction est terminée.

Un curseur est soit dans l'état ouvert, soit dans l'état fermé. L'état initial d'un curseur est l'état fermé. Un curseur passe à l'état ouvert par une <instruction ouverture> et retourne à l'état fermé par une <instruction fermeture>, une <instruction exécution>, ou une <instruction retour arrière>.

Un curseur dans l'état ouvert désigne une table, un ordre des lignes de cette table et une position relative dans cet ordre. Si la <déclaration de curseur> ne précise pas une <clause ordre par> ou si la <clause ordre par> ne peut pas spécifier complètement l'ordre des lignes, alors l'ordre de la ligne est défini partiellement ou totalement par le réalisateur.

A l'intérieur d'une transaction, où l'ordre des lignes n'est pas défini complètement ou partiellement par une <clause ordre par>, la position relative de deux lignes va être généralement identique chaque fois que le curseur est ouvert.

L'ordre peut changer d'une fois à l'autre si les valeurs de la base de donnée ou si les valeurs des paramètres référencés dans la <spécification de requête> de <déclaration de curseur> est différente.

Dans des transactions différentes, même quand les valeurs de la base de donnée et les valeurs des paramètres référencés dans la <spécification de requête> de <déclaration de curseur> sont les mêmes, l'ordre des lignes désigné par la même <spécification de curseur> et <instruction d'ouverture> peut être différent.

Un curseur dans l'état ouvert est mis soit avant une certaine ligne, soit sur une certaine ligne, soit après la dernière ligne. Si le curseur est sur une ligne, alors cette ligne est la ligne courante du curseur. Un curseur peut être avant la première ligne ou après la dernière ligne, même si la table est vide.

Une <instruction affectation> avance la position d'un curseur ouvert à la ligne suivante de l'ordre repéré par le curseur et récupère les valeurs des colonnes de cette ligne. Une <instruction mise à jour après positionnement> met à jour la ligne courante du curseur. Une <instruction suppression après positionnement> supprime la ligne courante du curseur.

Si un curseur est avant une ligne et qu'une nouvelle ligne est insérée à cette position, alors l'effet éventuel sur la position du curseur est défini par le réalisateur.

Si un curseur est sur une ligne ou avant une ligne et que cette ligne est supprimée, alors le curseur est positionné avant la ligne qui est immédiatement après la position de la ligne supprimée. Si une telle ligne est inexistante, le curseur est alors positionné après la dernière ligne.

Si une erreur apparaît durant l'exécution d'une <instruction SQL> identifiant un curseur ouvert, alors l'effet éventuel, tant sur la position du curseur que sur son état, est défini par le réalisateur.

Une table de travail est une table résultant de l'ouverture d'un curseur. Le réalisateur détermine si l'ouverture d'un curseur crée une table de travail de base, ou une vue de travail.

Chaque ligne d'une table vue de travail est dérivée seulement quand le curseur est mis sur cette ligne.

Une table de travail de base est créée quand le curseur est ouvert et détruite quand le curseur est fermé.

4.13 Instructions

Une <instruction SQL> définit une opération de base de données ou une opération de curseur. Une <instruction sélection> fournit des valeurs d'une table. Une <instruction insertion> insère des lignes dans une table. Une <instruction mise à jour après recherche> ou une <instruction mise à jour après positionnement> met à jour les valeurs de lignes d'une table. Une <instruction suppression après recherche> ou une <instruction suppression après positionnement> supprime des lignes d'une table.

4.14 Syntaxe d'incrustation

ISO/IEC 9075:1989

<https://standards.iteh.ai/catalog/standards/sist/7c7fa915-f99f-4ce9-8aa4-0c870d9c0b74/iso-iec-9075-1989>

Un <programme SQL incrusté dans un programme hôte> (<incrustation SQL dans un programme COBOL>, <incrustation SQL dans un programme FORTRAN>, <incrustation SQL dans un programme Pascal>, <incrustation SQL dans un programme PL/I>) est un programme d'application constitué d'instructions utilisant la syntaxe du langage de programmation et d'instructions utilisant la syntaxe SQL. La syntaxe du langage de programmation doit se conformer aux règles spécifiques du langage de programmation. La syntaxe SQL doit être constituée d'une ou de plusieurs <instruction(s) d'incrustation SQL> et, en option, d'une ou de plusieurs <déclarations de section SQL incrustée>. Ceci rend possible l'expression dans une forme hybride d'applications base de données dans lesquelles une ou des <instruction(s) SQL> sont incrustées directement dans un programme d'application. Un tel programme d'application hybride est défini comme équivalent à un programme d'application normalisé dans lequel la ou les <instruction(s) SQL> ont été remplacées par des procédures normalisées ou des appels de sous-programmes à une ou des <procédure(s) SQL dans un <module> SQL séparé.

4.15 Privilèges

Un privilège autorise l'exécution d'une catégorie donnée d'<action> sur une table spécifique, ou sur une vue, au moyen d'un <identificateur d'autorisation> spécifié. Les <action(s)> pouvant être spécifiées sont INSERT, DELETE, SELECT, UPDATE et REFERENCES.

Un <identificateur d'autorisation> est spécifié pour chaque <schéma> et chaque <module>.

L'<identificateur d'autorisation> défini pour un <schéma> doit être différent de l'<identificateur d'autorisation> de n'importe quel autre <schéma> dans le même environnement. L'<identificateur d'autorisation> d'un <schéma> est le «propriétaire» de toutes les tables et toutes les vues définies dans ce schéma.