CONSTRAINT

Версии сервера

0.9	1.0	1.5.3	1.5.4	1.5.5	2.0	2.0.3	2.0.4	2.1	2.5	3.0
Да	Да	Да	Да	Да	Да	Да	Да	Да	Да	?

Доступно в

ISQL

Формат

```
  .....
  ,[CONSTRAINT < constraint_name > ] {
     PRIMARY KEY ( < domain1 >  [,< domainN >] ) [USING [ASC|DESC] INDEX <
index_name > ]|
     FOREIGN KEY ( < domain1 >  [,< domainN >] )
         REFERENCES < table_name >( < domain1 >  [,< domainN >] )
         [ ON UPDATE < rule > ]
         [ ON DELETE < rule > ] [USING INDEX < index_name> ] |
     UNIQUE ( < domain1 > [,< domainN >] ) [USING [ASC|DESC] INDEX <
index_name > ] |
     CHECK <check_condition>
  }
  ...
```

Параметр	Значение
constraint_name	Имя создаваемого ограничения ссылочной целостности. При выполнении оператора может быть опущено, тогда сервер создаст имя автоматически в формате INTEG_NNNN. Соответственно, рекомендуется всегда указывать имя constraint. Внимание: если при создании ограничения ссылочной целостности Вы не указываете имя ограничения, то зарезервированное слово CONSTRAINT нужно пропускать тоже.
domain1	Имя домена (или имена доменов, перечисленные через запятую), по которому накладывается ограничение на таблицу.

Параметр	Значение
rule	Правило автоматической установки новых значений записей в таблице на которую накладывается ограничение ссылочной целостности, при изменении значения записи в таблице, на которую ссылаются записи. Подробно описание видов правил см. ниже.
check_condition	Логическое условие проверки значения, по которому накладывается ограничение на таблицу. Подробнее, см. CHECK
index_name	Явно задаваемое имя индекса. Если не указано, то SQL-сервер создаст индекс с системным именем.

Начиная с Firebird 1.5 имена индексов, автоматически создаваемых для PRIMARY, FOREIGN и UNIQUE constraint, при явном указании имени CONSTRAINT принимают это же имя. Например, если PRIMARY KEY создан как constraint PK_TABLE primary key, то индекс по этому constraint будет иметь имя PK_TABLE.

Описание

Объявление ограничения ссылочной целостности

Зарезервированное слово CONSTRAINT языка SQL сервера Firebird служит для работы с ограничениями ссылочной целостности создаваемых в базе данных таблиц. Ограничения ссылочной целостности бывают двух уровней: ограничения, накладываемые на отдельный столбец, и ограничения, накладываемые на всю таблицу. Одноименный оператор CONSTRAINT служит для работы с ограничениями ссылочной целостности, накладываемых на таблицу. Для наложения ограничений ссылочной целостности на столбец см. CREATE DOMAIN.

Оператор ограничения ссылочной целостности CONSTRAINT не употребляется самостоятельно, а выполняется в рамках операторов CREATE TABLE и ALTER TABLE, например:

```
CREATE TABLE MY_TABLE(
  ID INTEGER NOT NULL,
  SOME_ID INTEGER NOT NULL,
  ......
  CONSTRAINT PK_MY_TABLE PRIMARY KEY (ID)
);
```

или в рамках оператора ALTER TABLE

```
ALTER TABLE MY_TABLE
  ADD CONSTRAINT FK_MY_TABLE_SOME_ID FOREIGN KEY (SOME_ID)
        REFERENCES SOME_TABLE(ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE;
```

Виды ограничений ссылочной целостности

1.Первичный ключ

Первичный ключ - домен или несколько доменов таблицы, однозначно характеризующие запись согласно третьей нормальной форме Бойесса-Кода. Объявляется при помощи служебных слов PRIMARY KEY(< domain1 > [,< domainN >]), где < domainN > - имя домена или имена доменов, перечисленные через запятую. Первичный ключ также может быть задан при создании домена таблицы. Например:

```
CREATE TABLE MY_TABLE(
  ID INTEGER NOT NULL PRIMARY KEY,
  ....
);
```

или (рекомендуемый вариант, с указанием имени constraint)

```
CREATE TABLE MY_TABLE(
  ID INTEGER NOT NULL,
  ....
  CONSTRAINT PK_MY_TABLE PRIMARY KEY (ID),
);
```

При создании первичного ключа, необходимо еще также накладывать дополнительное ограничение ссылочной целостности на домены, входящие в первичный ключ NOT NULL. Подробнее, см. CREATE DOMAIN

У таблицы может быть только один первичный ключ.

Первичный ключ также можно создать командой ALTER TABLE

2.Внешний ключ

Внешний ключ служит для связки родительской и дочерней таблиц в базе данных, когда одной записи в родительской таблице соответствует множество записей в дочерней таблице. На этом условии могут быть построены отношения сущностей в базе данных «один-к-одному» и «один-ко-многим». Внешний ключ строится по столбцам в дочерней таблице, значения которых ссылаются на значения записей в родительской таблице.

Формат объявления оператора:

```
FOREIGN KEY (  < domain1 >  [,< domainN >] )
  REFERENCES < TABLE_NAME > [  < Rdomain1 >  [,< RdomainN >] ]
  [ON DELETE < rule > ] [ON UPDATE < rule > ];

где
```

```
< rule > = {NO ACTION | CASCADE | SET DEFAULT | SET NULL };
```

Параметр	Значение
< domain1 > [,< domainN >]	определяет столбцы дочерней таблицы, по которым строится внешний ключ.
< table_name >	Определяет таблицу, в которой описан первичный ключ (или столбец с атрибутом UNIQUE, см. ниже). На этот ключ (столбец) должен ссылаться внешний ключ дочерней таблицы для обеспечения ссылочной целостности.
< Rdomain1 > [,< RdomainN >]	необязателен при ссылке на первичный ключ родительской таблицы. При ссылке на ограничение ссылочной целостности с атрибутом UNIQUE (см. ниже) этот список требуется привести.

Правило < rule >, применяемое в необязательных атрибутах ON UPDATE и ON DELETE определяет поведение сервера для изменения записей в дочерних таблицах при изменении или удалении соответственно записей в родительской таблице. Ниже определены следующие правила.

Правило	Поведение сервера
NO ACTON	запрет изменения/удаления записи в родительской таблице при наличии записей в дочерней подчиненной таблице. Если правило ON UPDATE или ON DELETE не задано явно (пропущено при объявлении оператора), то действует по умолчанию правило NO ACTION.
CASCADE	для оператора DELETE: при удалении записей в родительской таблице приведет к удалению всех записей в подчиненной таблице, зависящих по внешнему ключу от записей в родительской таблице. Для оператора UPDATE при изменении полей связи в родительской таблице приведет к автоматическому обновлению записей в дочерней таблице на новое значение записей в родительской таблице.
SET DEFAULT	в столбец (столбцы) внешнего ключа у записей дочерней таблицы заносятся значения столбца по-умолчанию, указанное при определении столбца (параметр DEFAULT, см. CREATE DOMAIN); если это значение отсутствует, возбуждается исключение.
SET NULL	в столбец (столбцы) внешнего ключа дочерней таблицы заносятся значения NULL

Пример:

(рекомендуемый вариант, с указанием имени constraint)

```
CREATE TABLE MY_TABLE(
  ...,
  CID INTEGER NOT NULL,
  ...,
  CONSTRAINT FK_MY_TABLE_TABLE2 FOREIGN KEY (CID) REFERENCES TABLE2(ID))
```

Внешний ключ также можно создать командой ALTER TABLE. Автоматические генераторы скриптов обычно создают описание Внешних ключей именно через ALTER TABLE, и располагают их после создания всех таблиц, т.к. это позволяет избежать проблем, когда Внешний ключ ссылается на таблицу, которая еще не создана скриптом.

3.Уникальный ключ

Уникальный ключ - дополнительная возможность ограничения значений записей таблицы с целью профилактики занесения в нее двух и более записей, имеющих одинаковое значение указанного столбца (столбцов). В отличие от PRIMARY KEY количество уникальных ключей в таблице неограничено (точнее, ограничено максимальной комбинаторной суммой вариантов комбинации имен доменов, входящих в таблицу). Уникальный ключ также называется «альтернативным», и чаще всего предназначен не для однозначной идентификации столбца, как Первичный ключ, а для указания столбца, который позволяет осуществить дополнительную идентификацию строки. Например - номер паспорта, код ИНН, номер пенсионной страховки, и т.д.

Указание NOT NULL для UNIQUE, начиная с Firebird 1.5, не является обязательным. При этом уникальный индекс, который будет контролировать данное ограничение UNIQUE, может содержать NULL. Такой индекс невозможно создать командой CREATE INDEX.

Уникальный ключ может быть объявлен как и при описании домена:

```
CREATE TABLE MY_TABLE(
  ......
  FIELD1 INTEGER NOT NULL UNIQUE,
  .....
);
```

так и при описании таблицы (рекомендуемый вариант, с указанием имени constraint)

```
CREATE TABLE MY_TABLE(
  ....
  FIELD1 INTEGER NOT NULL,
  FIELD2 INTEGER NOT NULL,
  ....
  CONSTRAINT UNQ_MY_TABLE_1 UNIQUE(FIELD1, FIELD2),
  ....
);
```

При создании уникального ключа для версий Firebird ниже 1.5, столбец или домен должен содержать дополнительное ограничение ссылочной целостности NOT NULL. Подробнее см. CREATE DOMAIN

4.Проверка значений

Ограничение CHECK позволяет задать проверочное выражение, которое будет вычисляться при каждой вставке или обновлении данных. В случае, если выражение будет вычислено в true (истина), операция DML будет разрешена. И напротив, если выражение будет вычислено в false (ложь), операция будет прервана с выдачей соответствующего исключения о нарушении ограничения CHECK.

В случае, если выражение CHECK вычисляется в NULL, конечный результат проверки зависит

от версии сервера:
До FB2.0 NULL вычислялось в false, и тем самым, ограничение считалось нарушенным.
Начиная с FB2.0 поведение сервера было скорректировано в соответствие стандарту: итоговое значение NULL в контексте CHECK эквивалентно true, и тем самым, ограничение считается выполненным.

```
CREATE TABLE DRINK_VODKA(
  MAN_NAME VARCHAR(50) NOT NULL,
  MAN_AGE INTEGER NOT NULL,
  VODKA_VOL NUMERIC(12,3) NOT NULL,
  CONSTRAINT CHK1_DRINK_VODKA CHECK (MAN_AGE>=21),
  CONSTRAINT CHK2_DRINK_VODKA CHECK (VODKA_VOL BETWEEN 0.05 AND 0.5)
);
```

Ограничение CHECK также можно создать командой ALTER TABLE.

См. также

CREATE TABLE, ALTER TABLE

Источник

2008-05-02