文章
姚 鑫 · 九月 17, 2021 阅读大约需 9 分钟

第十八章 SQL命令 CREATE TABLE（五）

第十八章 SQL命令 CREATE TABLE（五）

定义外键

外键是引用另一个表的字段；存储在外键字段中的值是唯一标识另一个表中的记录的值。此引用的最简单形式如下例所示，其中外键显式引用Customers表中的主键字段CustID：

```
CREATE TABLE Orders (
   OrderID INT UNIQUE NOT NULL,
   OrderItem VARCHAR,
   OrderQuantity INT,
   CustomerNum INT,
   CONSTRAINT OrdersPK PRIMARY KEY (OrderID),
   CONSTRAINT CustomersFK FOREIGN KEY (CustomerNum) REFERENCES Customers (CustID)
   )
```

最常见的情况是，外键引用另一个表的主键字段。但是，外键可以引用RowID(ID)或标识列。在任何情况下，外键引用都必须存在于被引用的表中，并且必须定义为唯一的；被引用的字段不能包含重复值或NULL。

在外键定义中，可以指定：

- 字段名:FOREIGN KEY (CustomerNum) REFERENCES Customers (CustID)。
  外键字段(CustomerNum)和引用字段(CustID)可以有不同的名称(或相同的名称)，但必须具有相同的数据类型和字段约束。
- 以逗号分隔的字段名列表: FOREIGN KEY (CustomerNum,SalespersonNum) REFERENCES Customers (CustID,SalespID)。
  外键字段和引用字段的数量和列出的顺序必须相对应。
- 省略的字段名:FOREIGN KEY (CustomerNum)引用Customers。
- 显式的RowID字段:FOREIGN KEY (CustomerNum) REFERENCES Customers (%ID)。
  省略字段名的同义词。
  如果表的类定义包含SqlRowIdName，可以将此值指定为显式的RowID。

如果定义了一个外键并省略了引用的字段名，外键的默认值如下:

- 为指定的表定义的主键字段。
- 如果指定的表没有定义主键，则外键默认为为指定的表定义的IDENTITY列。
- 如果指定的表既没有定义的主键，也没有定义的标识列，则外键默认为RowID。仅当指定的表将RowID定义为PUBLIC时才会发生这种情况；指定的表定义可以通过指定%PUBLICROWID关键字或通过SqlRowIdPrivate=0(默认值)的相应类定义显式执行此操作。如果指定的表未将RowID定义为PUBLIC， IRIS会发出SQLCODE-315错误。在RowID上定义外键时必须省略引用的字段名；尝试将ID显式指定为引用的字段名会导致SQLCODE-316错误。

如果这些默认值都不适用，IRIS将发出SQLCODE-315错误。

在类定义中，可以指定包含基于父表IDKEY属性的字段的外键，如下例所示：

```
  ForeignKey Claim(CheckWriterPost.Hmo,Id,Claim) References SQLUser.Claim.Claim(DBMSK
eyIndex);
```

因为在子类的外键中定义的父字段必须是父类的IDKEY索引的一部分，所以此类型的外键唯一支持的引用操作是无操作。
- 如果外键引用了不存在的表， IRIS会发出SQLCODE-310错误，并在%msg中提供其他信息。
- 如果外键引用了不存在的字段， IRIS将发出SQLCODE-316错误，并在%msg中提供其他信息。
- 如果外键引用了非唯一字段， IRIS会发出SQLCODE-314错误，并在%msg中提供其他信息。

如果外键字段引用单个字段，则这两个字段必须具有相同的数据类型和字段数据约束。

在父/子关系中，没有定义的子项顺序。应用程序代码不得依赖于任何特定顺序。

可以定义引用以只读方式装载的数据库中的类的外键约束。要定义外键，用户必须对被引用的表或被引用的表的列具有REFERENCES特权。如果通过动态SQL或xDBC执行CREATE TABLE，则需要REFERENCES权限。

指称动作子句

如果一个表包含外键，对一个表的更改会对另一个表产生影响。为了保持数据的一致性，在定义外键时，还需要定义外键数据所来自的记录的更改对外键值的影响。

外键定义可能包含两个引用动作子句:

```
ON DELETE ref-action
```

或

```
ŝq;
ON UPDATE ref-action
```

```
`ON DELETE`????????????????????????????`ON DELETE`??????????????????

`ON UPDATE`??????????????????(??)??????????`ON UPDATE`??????????????????

SQL??????????

- `NO ACTION`
- `SET DEFAULT`
- `SET NULL`
- `CASCADE`

`NO ACTION`-????????????????????????????????????????????????????????(????????????????
???)????????????????????????????????????????`SQLCODE-400`??????????`Error#5600?Fe
ature not supported for sharded class Sample.MyShardT: Foreign Key ON UPDATE action o
f 'setnull'`?

`SET NULL`-????????????????????????????????????????????????????????????????????????
NULL????????????

`SET DEFAULT`-????????????????????????????????????????????????????????????????????
????????????????????????????`NULL`????????????????????????????
```

```
`CASCADE` -?????????????????????????????????????????????????????????????????????????

????????????????????????????????????????????????????????????????????????????????

????????????????????????????????-???????????????????ANSI?????????????????????????????(?
??`ON DELETE CASCADE`?`ON DELETE SET NULL`)? `SQL`???????????`DELETE`?`UPDATE`???????
?????????? `SQL`??????

????????`SQL`??????????????????`CREATE TABLE`???????????????`PhysNum`(?????`PhysNum
`)?????

```java
ClassMethod CreateTable6()
{
    d $SYSTEM.Security.Login("_SYSTEM","SYS")
    &sql(
        CREATE TABLE Patient
        (
            PatNum VARCHAR(16),
            Name VARCHAR(30),
            DOB DATE,
            Primary_Physician VARCHAR(16) DEFAULT 'A10001982321',
                CONSTRAINT Patient_PK PRIMARY KEY (PatNum),
                CONSTRAINT Patient_Physician_FK FOREIGN KEY
            Primary_Physician REFERENCES Physician (PhysNum)
                ON UPDATE CASCADE
                ON DELETE SET NULL
        )
    )
    WRITE !,"SQL code: ",SQLCODE
}
```

NOCHECK关键字

如果指定NOCHECK关键字， IRIS不检查外键引用完整性。这意味着INSERT或UPDATE操作可能会为外键字段指定一个与被引用表中的行不对应的值。NOCHECK关键字还阻止执行外键的引用操作子句。SQL查询处理器可以使用外键来优化表之间的联接。但是，如果将外键定义为NOCHECK，则SQL查询处理器不会将其视为已定义的外键。NOCHECK外键仍然作为外键报告给xDBC目录查询。

分片表和外键

外键支持分片和未分片表的任意组合，包括：键表分片、fkey表未分片；key表未分片、fkey表分片；同时支持key表和fkey表分片。被引用表中的键可以是碎片键，也可以是另一个键。外键可以是单个字段或多个字段。

NO ACTION是切片表支持的唯一引用操作。

隐式外键

最好显式定义所有外键。如果定义了显式外键， IRIS会报告此约束，而不定义隐式外键约束。

但是，可以将隐式外键投影到ODBC/JDBC和管理门户。所有字段引用都作为外键投影到ODBC/JDBC，如下所示：

这些隐式外键被报告为无操作的UPDATE和DELETE引用操作。此隐式引用外键不是真正的外键，因为没有强制执行引用操作。为引用报告的此外键的名称为 “ IMPLICITFKEYREFERENCE_” fieldname。将此引用报告为外键是为了与第三方工具实现互操作性。
```

定义分片键

提供将表定义为分片的选项是为了提高针对该表的查询性能，特别是对于包含大量记录的表。分片表只能在分片环境中使用；非分片表可以在分片或非分片环境中使用。并不是所有的表都适合进行分片。分片环境中的最佳性能通常是通过组合使用分片表(通常非常大的表)和非分片表来实现的。

如果当前命名空间配置为分片(分片主数据服务器上的主命名空间)，则可以为表指定分片键。如果没有为切分配置当前命名空间，则指定切片键的CREATE
TABLE失败，并返回SQLCODE-400致命错误，并显示%msg错误#9319：Current namespace %1 has no shards configured。

切片键定义应该紧跟在table-element-commist的右括号之后，但在WITH子句之前(如果指定)。为了向后兼容，支持将分片键定义指定为table-element-commist中的元素。在两个位置指定分片键定义会生成SQLCODE-327错误。

有三个选项可用于指定碎片键定义：

- SHARD：如果仅指定关键字Shard， IRIS使用表的RowID字段作为碎片键。对于几乎所有的切片表，这都是最有效的方法。如果表有定义的标识字段，但没有显式的分片键，它将使用该标识字段作为分片键。
- SHARD KEY(Fieldname)：可以使用此语法指定RowID以外的分片键。可以指定一个字段名称或逗号分隔的字段名称列表作为分片键。分片键字段的数据类型必须是数字或字符串数据类型。
- SHARD KEY (fieldname) COSHARD WITH (tablename)：可选的COSHARD WITH子句允许指定一个表，以便使用定义的切片表进行编码。此选项用于为查询中通常联接的大表启用共分联接。带有关键字的COSHARD子句和包含coshard表名的圆括号都是可选的。

定义的切片表必须具有显式指定的切片键(字段)。此分片键字段必须采用整数值；它应该与系统分配的协分片表的RowID值相匹配。例如，SHARD KEY (deptnum) COSHARD WITH department。COSHARD
WITH子句中指定的表必须是具有系统分配的切片键的切片表。

COSHARD
WITH子句在分割表的ShardKey索引中定义CoshardWith索引关键字。此CoshardWith索引关键字等于投影表的类。

可以通过查看Cosharding Comment选项来确定查询中指定的哪些已分片表是共分的。

必须使用CREATE TABLE或持久类定义将表定义为分割表。不能使用ALTER TABLE向现有表添加分片键。

如果表有定义的IDKEY，则必须将字段定义为分片键字段。既不能指定fieldname不是该字段的分片键(Fieldname)，也不能指定没有定义键字段的分片。尝试这样做会导致SQLCODE-400错误，并显示如下%msg：ERROR #5597: Sharded table's shard key (%1) must be the same as the idkey (%2) when the idkey is defined.

如果表中定义了标识字段，则可以将该字段定义为分片关键字字段，也可以在标识字段以外的一个或多个字段上定义分片关键字。

除非切片键是唯一键的子集，否则切片表上的唯一字段约束可能会对插入/更新性能产生重大负面影响。

涉及到需要原子性的复杂事务的表永远不应该被分片。

分片表在分片主数据服务器上的主命名空间中定义。该主命名空间还可以包括非分片表。分片对于SQL查询是透明的；不需要特殊的查询语法。查询不需要知道表是分片的还是非分片的。同一查询可以访问分割表和非分割表。查询可以包括分割表和非分割表之间的联接。

分片表定义限制

- 分片表不能包含ROWVERSION数据类型或SERIAL (%Library.Counter)数据类型字段。
- 分片表不能指定VERSIONPROPERTY类参数。

#SQL #Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E5%85%AB%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-create-table%EF%BC%88%E4%BA%94%EF%BC%89