MySQL 元数据 MySQL 处理重复数据

MySQL 序列使用

MySQL 序列是一组整数：1, 2, 3, ...，由于一张数据表只能有一个字段自增主键， 如果你想实现其他字段也实现自动增加，就可以使用MySQL序列来实现。

本章我们将介绍如何使用MySQL的序列。

使用 AUTO_INCREMENT

MySQL 中最简单使用序列的方法就是使用 MySQL AUTO_INCREMENT 来定义列。

实例

以下实例中创建了数据表 insect， insect 表中 id 无需指定值可实现自动增长。

```
mysql> CREATE TABLE insect
    -> (
    -> id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    -> PRIMARY KEY (id),
    -> name VARCHAR(30) NOT NULL, # type of insect
    -> date DATE NOT NULL, # date collected
    -> origin VARCHAR(30) NOT NULL # where collected
);
Query OK, 0 rows affected (0.02 sec)
mysql> INSERT INTO insect (id,name,date,origin) VALUES
    -> (NULL,'housefly','2001-09-10','kitchen'),
    -> (NULL,'millipede','2001-09-10','driveway'),
    -> (NULL,'grasshopper','2001-09-10','front yard');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
mysql> SELECT * FROM insect ORDER BY id;
+----+-------------+------------+------------+
| id | name        | date       | origin     |
+----+-------------+------------+------------+
|  1 | housefly    | 2001-09-10 | kitchen    |
|  2 | millipede   | 2001-09-10 | driveway   |
|  3 | grasshopper | 2001-09-10 | front yard |
+----+-------------+------------+------------+
3 rows in set (0.00 sec)
```

获取AUTO_INCREMENT值

在MySQL的客户端中你可以使用 SQL中的LAST_INSERT_ID( ) 函数来获取最后的插入表中的自增列的值。

在PHP或PERL脚本中也提供了相应的函数来获取最后的插入表中的自增列的值。

PERL实例

使用 mysql_insertid 属性来获取 AUTO_INCREMENT 的值。 实例如下：

```
$dbh->do ("INSERT INTO insect (name,date,origin)
VALUES('moth','2001-09-14','windowsill')");
my $seq = $dbh->{mysql_insertid};
```

PHP实例

PHP 通过 mysql_insert_id ()函数来获取执行的插入SQL语句中 AUTO_INCREMENT列的值。

```
mysql_query ("INSERT INTO insect (name,date,origin)
VALUES('moth','2001-09-14','windowsill')", $conn_id);
$seq = mysql_insert_id ($conn_id);
```

重置序列

如果你删除了数据表中的多条记录，并希望对剩下数据的AUTO_INCREMENT列进行重新排列，那么你可以通过删除自增的列，然后重新添加来实现。 不过该操作要非常小心，如果在删除的同时又有新记录添加，有可能会出现数据混乱。操作如下所示：

```
mysql> ALTER TABLE insect DROP id;
mysql> ALTER TABLE insect
    -> ADD id INT UNSIGNED NOT NULL AUTO_INCREMENT FIRST,
    -> ADD PRIMARY KEY (id);
```

设置序列的开始值

一般情况下序列的开始值为1，但如果你需要指定一个开始值100，那我们可以通过以下语句来实现：

```
mysql> CREATE TABLE insect
    -> (
    -> id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    -> PRIMARY KEY (id),
    -> name VARCHAR(30) NOT NULL,
    -> date DATE NOT NULL,
    -> origin VARCHAR(30) NOT NULL
)engine=innodb auto_increment=100 charset=utf8;
```

或者你也可以在表创建成功后，通过以下语句来实现：

```
mysql> ALTER TABLE t AUTO_INCREMENT = 100;
```

MySQL 元数据 MySQL 处理重复数据

1 篇笔记

写笔记

使用函数创建自增序列管理表(批量使用自增表,设置初始值,自增幅度)

第一步：创建Sequence管理表 sequence

```
DROP TABLE IF EXISTS sequence;
CREATE TABLE sequence (
name VARCHAR(50) NOT NULL,
current_value INT NOT NULL,
increment INT NOT NULL DEFAULT 1,
PRIMARY KEY (name)
) ENGINE=InnoDB;
```

第二步：创建取当前值的函数 currval

```
DROP FUNCTION IF EXISTS currval;
DELIMITER $
CREATE FUNCTION currval (seq_name VARCHAR(50))
RETURNS INTEGER
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN
DECLARE value INTEGER;
SET value = 0;
SELECT current_value INTO value
FROM sequence
WHERE name = seq_name;
RETURN value;
END
$
DELIMITER ;
```

第三步：创建取下一个值的函数 nextval

```
DROP FUNCTION IF EXISTS nextval;
DELIMITER $
CREATE FUNCTION nextval (seq_name VARCHAR(50))
RETURNS INTEGER
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
SQL SECURITY DEFINER
```

```
COMMENT ''
BEGIN
UPDATE sequence
SET current_value = current_value + increment
WHERE name = seq_name;
RETURN currval(seq_name);
END
$
DELIMITER;
```

第四步：创建更新当前值的函数 setval

```
DROP FUNCTION IF EXISTS setval;
DELIMITER $
CREATE FUNCTION setval (seq_name VARCHAR(50), value INTEGER)
RETURNS INTEGER
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN
UPDATE sequence
SET current_value = value
WHERE name = seq_name;
RETURN currval(seq_name);
END
$
DELIMITER ;
```

测试函数功能

当上述四步完成后，可以用以下数据设置需要创建的sequence名称以及设置初始值和获取当前值和下一个值。

```
INSERT INTO sequence VALUES ('TestSeq', 0, 1);
----添加一个sequence名称和初始值，以及自增幅度  添加一个名为TestSeq 的自增序列

SELECT SETVAL('TestSeq', 10);
---设置指定sequence的初始值     这里设置TestSeq 的初始值为10

SELECT CURRVAL('TestSeq');
--查询指定sequence的当前值   这里是获取TestSeq当前值

SELECT NEXTVAL('TestSeq');
--查询指定sequence的下一个值  这里是获取TestSeq下一个值
```

Narule 6个月前 (09-11)