Neuedu

MySQL数据库开发技术

—— 视图和索引

V1.0

本章内容

节	知识点	掌握程度	难易程度
视图概述	什么是视图	理解	
	为什么使用视图	理解	
	视图的分类	了解	
创建视图	创建视图	掌握	
	创建复杂视图	掌握	
从视图中检索数据	从视图中检索数据	掌握	
修改视图	修改视图	掌握	
视图上执行DML操作	视图上执行DML操作的规则	了解	难
	拒绝DML操作	理解	
删除视图	删除视图	掌握	
索引概述	索引简介	理解	
	索引分类	了解	
创建索引	创建表的时候创建索引	掌握	
	在已经存在的表上创建索引	掌握	
	使用ALTER TABLE语句来创建索引	了解	
删除索引	删除索引	掌握	
索引的设计原则	索引的设计原则	理解	难

视图概述

- 什么是视图
  - 视图是逻辑上来自一个或多个表的数据集合。

EMP 表

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7782	CLARK	MANAGER	7839	09-JUN-81	1500	300	10
7934	MILLER	CLERK	7782	23-JAN-82	1300		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
							20
							20
							20
							30
						1400	30
						300	30
						0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30

EMPVU10 视图

EMPNO	ENAME	JOB
7839	KING	PRESIDENT
7782	CLARK	MANAGER
7934	MILLER	CLERK

视图概述

Neuedu

- 为什么使用视图
  - 限制其它用户对数据库表的访问, 因为视图可以有选择性的显示数据库表的一部分;
  - 容易实现复杂的查询;
  - 对于相同的数据可以产生不同的视图;

视图概述

- 视图分类
  - 视图分为简单视图和复杂视图，最基本差别在DML操作上

特征	简单视图	复杂视图
基表数量	一个	一个或多个
包含函数	没有	有
包含数据组	没有	有
通过视图实现DML操作	可以	不一定

创建视图

- 创建视图语法；

```
CREATE [OR REPLACE] [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
VIEW view_name [(column_list)]
AS select_statement
 [WITH [CASCADED | LOCAL] CHECK OPTION]
```

- OR REPLACE：如果所创建的视图已经存在，该选项表示修改原视图的定义；
- view_name ：视图的名称；
- column_list ：列名，列名的数量必须和视图所对应查询语句的列数量相等；
- select_statement ：一条完整的SELECT语句；
- WITH CHECK OPTION：一个约束条件，通过视图所插入或修改的数据行必须满足视图所定义的查询；
- ALGORITHM子句是可选的，它表示使用何种算法来处理视图。此外，它并不属于标准SQL的一部分，而是MySQL对标准SQL进行的功能扩展。ALGORITHM可以设置三个值：MERGE、TEMPTABLE或UNDEFINED。如果没有ALGORITHM子句，则默认值为UNDEFINED(未定义的)。

创建视图

- 对于MERGE，会将引用视图的语句的文本与视图定义合并起来，使得视图定义的某一部分取代语句的对应部分。
- 对于TEMPTABLE，视图的结果将被置于临时表中，然后使用它执行语句。
- 对于UNDEFINED，MySQL将选择所要使用的算法。如果可能，它倾向于MERGE而不是TEMPTABLE，这是因为MERGE通常更有效，而且如果使用了临时表，视图是不可更新的。
- [WITH [CASCADED | LOCAL] CHECK OPTION]是可选的。该选项中的CASCADED为默认值，LOCAL CHECK OPTION用于在可更新视图中防止插入或更新行，此选项一般不使用。

创建视图

- 例：创建一个视图v_emp10，通过该视图只能查看10号部门的员工编号，员工姓名，职位。

```
SQL> CREATE VIEW     empvu10
  2  AS SELECT       empno, ename, job
  3  FROM            emp
  4  WHERE           deptno = 10;
View created.
```

- 在用SQL*Plus中，可以使用DESC命令显示视图的结构。

```
SQL> DESC empvu10
```

练习1

- 1. 创建一个视图，通过该视图可以查询到工资在2000-5000内并且姓名中包含有A的员工编号，姓名，工资。
- 2. 通过上述创建的视图查询数据

创建视图

Neuedu

- 创建视图时，在子查询中使用列的别名

```
SQL> CREATE VIEW     salvu30
  2  AS SELECT       empno EMPLOYEE_NUMBER, ename NAME,
  3                  sal SALARY
  4  FROM            emp
  5  WHERE           deptno = 30;
View created.
```

创建复杂视图

Neuedu

- 例：创建一个视图，通过该视图可以查看每个部门的名称，最低工资，最高工资，平均工资

```
SQL> CREATE VIEW    dept_sum_vu
  2                 (name, minsal, maxsal, avgsal)
  3  AS SELECT      d.dname, MIN(e.sal), MAX(e.sal),
  4                 AVG(e.sal)
  5  FROM           emp e, dept d
  6  WHERE          e.deptno = d.deptno
  7  GROUP BY       d.dname;
View created.
```

从视图检索数据

- 从视图中检索数据，同从表中检索数据一样，只不过是只能看到视图所定义的那些列。

```
SQL>   SELECT *
  2    FROM    salvu30;
```

```
EMPLOYEE_NUMBER NAME           SALARY
--------------- ---------- ---------
           7698 BLAKE           2850
           7654 MARTIN          1250
           7499 ALLEN           1600
           7844 TURNER          1500
           7900 JAMES            950
           7521 WARD            1250

6 rows selected.
```

练习2

- 1.创建一个视图，通过该视图可以查询到工作在NEW YORK和CHICAGO的员工编号，姓名，部门编号，入职日期。
- 2.创建一个视图，通过该视图可以查询到每个部门的部门名称及最低工资。
- 3.通过如上视图，查询每个部门工资最低的员工姓名及部门名称

修改视图

- 用 CREATE OR REPLACE VIEW子句修改视图empvu10，为每个列添加别名。

```
SQL> CREATE OR REPLACE VIEW empvu10
  2          (employee_number, employee_name, job_title)
  3   AS SELECT       empno, ename, job
  4   FROM            emp
  5   WHERE           deptno = 10;
View created.
```

- CREATE VIEW子句中别名的顺序必须和内部查询中的列的顺序一一对应。

视图上执行DML操作

- 在简单视图上可以执行 DML 操作；
- 您可以通过视图**删除**基表中数据，只要视图中不出现以下情况：
  - Group 函数；
  - GROUP BY 子句；
  - DISTINCT 关键字；
- 您可以通过视图**修改**基表中数据，只要视图中不出现以下情况：
  - GROUP函数、GROUP BY子句，DISTINCT关键字；
  - 使用表达式定义的列；

视图上执行DML操作

- 您可以通过视图向基表**插入**数据，只要视图中不出现以下情况：
  - GROUP函数、GROUP BY子句，DISTINCT关键字；
  - 使用表达式定义的列；
  - 基表中未在视图中选择的其它列定义为非空并且没有默认值；

视图上执行DML操作

- 如果要确保在视图上执行的DML操作仅限于一定的范围，便可使用WITH CHECK OPTION子句；

```
SQL> CREATE OR REPLACE VIEW empvu20
  2   AS SELECT       *
  3   FROM            emp
  4   WHERE           deptno = 20
  5   WITH CHECK OPTION;
View created.
```

- 在视图中任何修改部门编号的操作都会失败，因为这违反了 WITH CHECK OPTION约束。

删除视图

- 删除视图并不会删除数据，因为视图是基于数据库中的基表的虚表。

```
DROP VIEW view;
```

```
SQL> DROP VIEW empvu10;
View dropped.
```

索引概述

- 索引简介
  - 索引是一种特殊的数据库结构，可以用来快速查询数据库表中的特定记录。索引是提高数据库性能的重要方式。MySQL中，所有的数据类型都可以被索引。

索引概述

Neuedu

- 索引简介
  - 索引由数据库表中一列或多列组合而成，其作用是提高对表中数据的查询速度。
  - 索引是创建在表上的，是对数据库表中一列或多列的值进行排序的一种结构。
  - 索引可以提高查询的速度。通过索引，查询数据时可以不必读完记录的所有信息，而只是查询索引列。

索引概述

Neuedu

- 索引简介
  - 索引的优点是可以提高检索数据的速度，这是创建索引的最主要的原因；对于有依赖关系的子表和父表之间的联合查询时，可以提高查询速度；使用分组和排序子句进行数据查询时，同样可以显著节省查询中分组和排序的时间。
  - 索引的缺点是创建和维护索引需要耗费时间，耗费时间的数量随着数据量的增加而增加；索引需要占用物理空间，每一个索引要占一定的物理空间；增加、删除和修改数据时，要动态的维护索引，造成数据的维护速度降低了。

索引概述

- 索引分类
  - 普通索引
  - 惟一性索引
  - 全文索引
  - 单列索引
  - 多列索引
  - 空间索引

创建索引

- 创建索引是指在某个表的一列或多列上建立一个索引，以便提高对表的访问速度。创建索引有三种方式，这三种方式分别是：
  - 创建表的时候创建索引
  - 在已经存在的表上创建索引
  - 使用ALTER TABLE语句来创建索引

创建表的时候创建索引

- 创建表的时候可以直接创建索引，这种方式最简单、方便。其基本形式如下：
- CREATE TABLE 表名（属性名 数据类型 [完整性约束条件],
  属性名 数据类型 [完整性约束条件],
  ...
  属性名 数据类型
  [UNIQUE | FULLTEXT | SPATIAL] INDEX | KEY
  [别名](属性名1 [(长度)] [ASC | DESC])
  );

创建表的时候创建索引

- 普通索引

```
Create table index1(
Id int,
Name varchar(20),
Sex boolean,
Index(id)
);
```

创建表的时候创建索引

Neuedu

- 创建唯一性索引

```
Create table index2(
Id int unique,
Name varchar(20),
Unique index index2_id(id asc)
);
```

创建表的时候创建索引

Neuedu

- 创建全文索引只能创建在char,varchar或text类型的字段上，而且，现在只有MyISAM存储引擎支持全文索引。

```
Create table index3(
Id int,
Info varchar(20),
Fulltext index index3_info(info)
)engine=myisam;
```

创建表的时候创建索引

- 创建单列索引

```
Create table index4(
Id int,
Subject varchar(30),
Index index4_st(subject(10))
);
```

创建表的时候创建索引

- 创建多列索引

```
Create table index5(
Id int,
Name varchar(20),
Sex char(4),
Index index5_ns(name,sex)
);
```

- 使用多列索引时一定要特别注意，只有使用了索引中的第一个字段时才会触发索引。如果没有使用索引中的第一个字段，那么这个多列索引就不会起作用。

创建表的时候创建索引

Neuedu

- 创建空间索引

```
Create table index6(
Id int,
Space geometry not null,
Spatial index index6_sp(space)
)engine=myisam;
```

- 创建空间索引时，表的存储引擎必须是myisam类型，而且索引字段必须有非空约束。空间数据类型包括geometry,point,linestring和polygon类型等。平时很少用到。

在已经存在的表上创建索引

- 在已经存在的表上，可以直接为表上的一个或几个字段创建索引。基本形式如下：
- CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX 索引名
  ON 表名 (属性名 [(长度)] [ASC | DESC]);
  - 创建普通索引
  - 创建惟一性索引
  - 创建全文索引
  - 创建单列索引
  - 创建多列索引
  - 创建空间索引

用ALTER TABLE语句来创建索引

- 在已经存在的表上，可以通过ALTER TABLE语句直接为表上的一个或几个字段创建索引。基本形式如下：
- ALTER TABLE 表名 ADD [UNIQUE | FULLTEXT | SPATIAL] INDEX
  索引名（属性名 [(长度)] [ASC | DESC]）；
- 其中的参数与上面的两种方式的参数是一样的
  - 创建普通索引
  - 创建惟一性索引
  - 创建全文索引
  - 创建单列索引
  - 创建多列索引
  - 创建空间索引

删除索引

- 删除索引是指将表中已经存在的索引删除掉。一些不再使用的索引会降低表的更新速度，影响数据库的性能。对于这样的索引，应该将其删除。
- 对应已经存在的索引，可以通过DROP语句来删除索引。基本形式如下：
- DROP INDEX 索引名 ON 表名；

索引的设计原则

- 为了使索引的使用效率更高，在创建索引的时候必须考虑在哪些字段上创建索引和创建什么类型的索引。索引的设计原则如下：
  - 选择惟一性索引
  - 为经常需要排序、分组和联合操作的字段建立索引
  - 为常作为查询条件的字段建立索引
  - 限制索引的数目
  - 尽量使用数据量少的索引
  - 尽量使用前缀来索引
  - 删除不再使用或者很少使用的索引

小结

- 本章介绍了MySQL的视图和索引
  - 掌握如何创建简单视图
  - 掌握如何创建复杂视图
  - 理解带约束视图的含义
  - 掌握如何从视图中检索数据
  - 掌握索引的创建
  - 了解设计索引的基本原则

课后作业

- 1. 创建视图v_emp_20，包含20号部门的员工编号，姓名，年薪列(年薪=12*(工资+奖金）；
- 2. 从视图v_emp_20中查询年薪大于1万元员工的信息；
- 3. 请为工资大于2000的员工创建视图，要求显示员工的部门信息，职位信息，工作地点；
- 4. 针对以上视图执行insert,update,delete,语句能否成功，为什么？