

数据库的组成部分

1. 表

基本表

- 本身独立存在的表
- SQL中一个关系就对应一个基本表
- 一个（或多个）基本表对应一个存储文件
- 一个表可以带若干索引

对表的操作有创建、查询、修改、删除

2. 索引

索引是根据指定的数据库表列建立起来的顺序。

建立索引的目的：它提供了快速访问数据的途径，并且可监督表的数据，使其索引所指向的列中的数据不重复。

关系数据库管理系统中常见索引：

- 顺序文件上的索引
- B+树索引
- 散列（hash）索引
- 位图索引

3. 视图

- 从一个或几个基本表导出的表
- 只存放视图的定义而不存放视图对应的数据（即只存储查询的表达式
- 视图是一个虚表，是从一个或几个基本表（或视图）导出的表
- 用户可以在视图上再定义视图
- 基表中的数据发生变化，从视图中查询出的数据也随之改变

4. 函数

4.1. 单行函数

4.1.1. 字符函数

对字符型数据进行常见的串操作，例如：

- 获取子串
- 串的拼接
- 串的匹配
- 等等

4.1.2. 数学函数

对 `number` 型数据进行科学计算

4.1.3. 日期函数

对 `date` 或 `timestamp` 类型的数据进行日期的获取以及计算等

- `add_months`，即对当前的日期进行月份的修改计算

4.1.4. 转换函数

实现基本类型之间的数据转换

4.2. 聚合函数

包括 `min()`、`max()`、`sum()`、`count()`、`avg()` 等运算，用于获取一个数据集的数据特征

4.3. 分析函数

- `row_number()...over(partition by 列1 order by 列2)`
 - 按照列1分组，按照列2排序

生成一系列连续序号，`order by` 列的值相同的序号也不同

- `rank()...over(partition by 列1 order by 列2)`，
 - 按照列1分组，按照列2排序

生成一系列可能不连续序号，`order by` 列的值相同的序号相同

- `dense_rank()...over(partition by 列1 order by 列2)`，
 - 按照列1分组，按照列2排序

5. 存储过程

存储过程是为完成特定的功能而汇集在一起的一组 SQL 程序语句，经编译后存储在数据库中的 SQL 程序。

特点

- 存储过程只在创建时进行编译，以后每次执行存储过程都不需再重新编译，可提高数据库执行速度。
- 当对数据库进行复杂操作时可将此复杂操作存储过程封装起来与数据库提供的事务处理结合一起使用。这些操作，如果用程序来完成，就变成了一条条的SQL 语句，可能要多次连接数据库。而换成存储，只需要连接一次数据库就可以了。
- 存储过程可以重复使用可减少数据库开发人员的工作量。
- 安全性高可设定只有某此用户才具有对指定存储过程的使用权。
- 更强的适应性：由于存储过程对数据库的访问是通过存储过程来进行的，因此数据库开发人员可以在不改动存储过程接口的情况下对数据库进行任何改动，而这些改动不会对应用程序造成影响。
- 分布式工作：应用程序和数据库的编码工作可以分别独立进行，而不会相互压制。
一般来说，存储过程的编写比基本SQL复杂
- 存储过程定义语法

```
create or replace procedure 过程名(参数列表) as | is
---变量的声明
begin
---语句块
exception
---异常处理
end 过程名;
---调用 begin
过程名(参数); end;
```

6. 触发器

触发器是一个用户定义的 SQL 事务命令的集合。当对一个表进行插入、更改、删除时，这组命令就会自动执行

7.用户（User）

即有权限访问数据库的人。