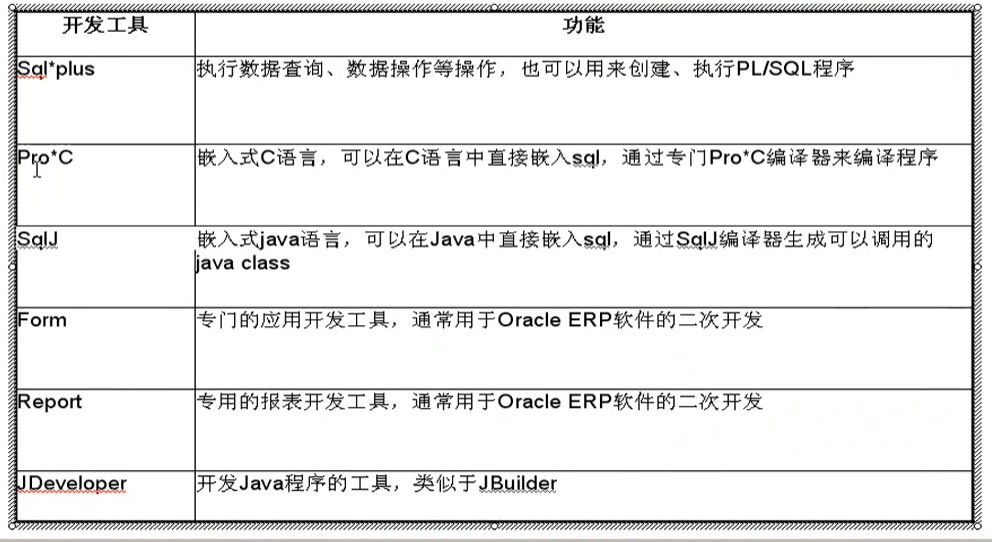
# 概述

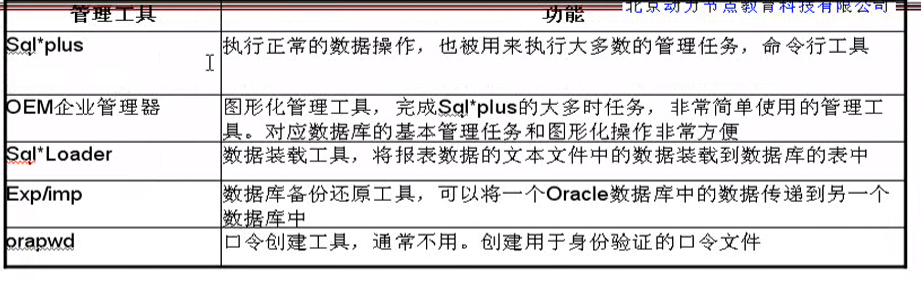
## 开发工具

Oracle数据库主要开发工具：

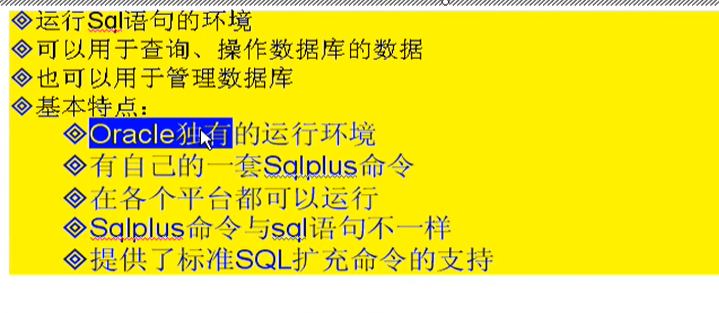


## 管理工具

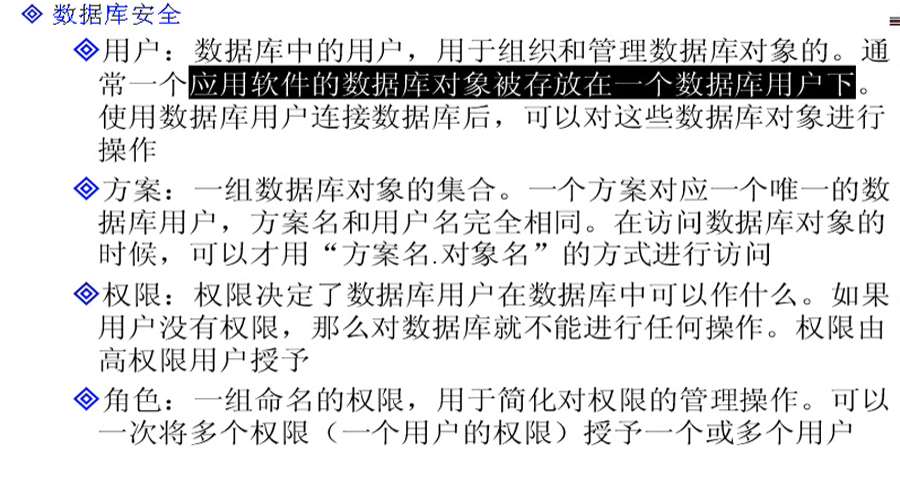
Oracle数据库主要管理工具：



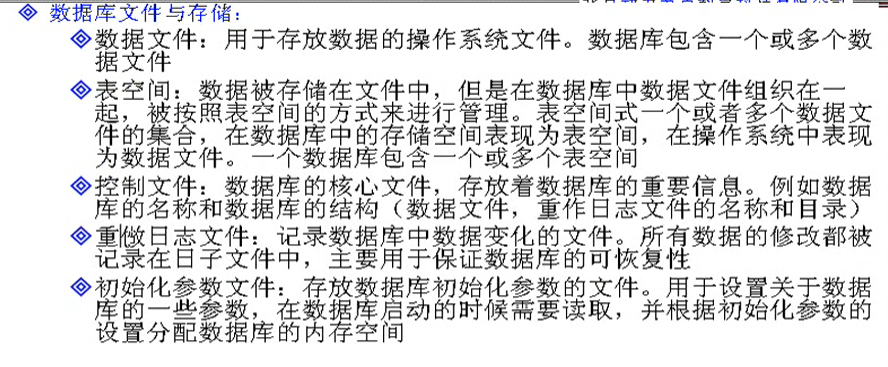
## sqlplus



## 数据库安全



## 数据库文件与存储



# 函数

# DDL语句

**sequence**

**概述**

在oracle中sequence就是序号，是用来在多用户环境下产生唯一整数的数据库对象。每次取的时候它会自动增加，sequence与表没有关系。

序列产生器顺序生成数字，它可用于自动生成主键值，并能协调多行或者多表的主键操作。没有sequence，顺序的值只能靠编程来生成，先找出最近产生的值然后自增长。

这种方法要求一个事务级别的锁，这将导致在多用户并发操作的环境下，必须有人等待下一个主键值的产生，而且此方法很容易产生主键冲突的错误。

**功能**

Sequence的功能就是产生分布式系统单调变化的序列，为了分布式系统也能像单机系统一样产生可靠的全局单调序列，因此选择系统中控制全局的网元承担数据库表的创建、维护和消耗功能，但是不维护sequence元数据。

其功能主要体现在：

1. 作为代理主键，唯一识别；
2. 用于记录数据库最新动作的语句，只要语句有动作（增删改），sequence号就会随着更新，所以我们可以根据sequence号来select出更新的语句。

**操作**

**创建sequence**

**获取sequence值**

**修改sequence**

**删除sequence**

# DML语句

# DQL语句

**hint**

**概述**

为什么要引入hint？

Hint是oracle数据库中很有特色的一个功能，是很多DBA优化中经常使用的一个手段。那么为什么oracle会考虑引入优化器呢？基于代价的优化器是很聪明的，在绝大多数情况下它会选择正确的优化器，减轻DBA的负担。

但有时它聪明反被聪明误，选择了很差的执行计划，使得某个语句的执行变得异常慢。此时就需要DBA进行人为地干预，告诉优化器使用指定的存储路径或连接类型生成执行计划，从而使语句高效地进行。Hint就是oracle提供的一种机制，用来告诉优化器按照告诉它的方式执行计划。

Hint与注释

提示是oracle为了不解决和其他数据库引擎之间对SQL语句的兼容性而提供的一种拓展功能。

Oracle决定把提示作为一种特殊的注释来添加，它的特殊性表现在提示必须紧跟着DELETE、INSERT、UPDATE或MERGE关键字。

换句话说，提示不能像普通注释那样在SQL语句中随处添加。且在注释分隔符之后的第一个字符必须是加号。

**功能**

Hint提供的功能非常多，可以很灵活地调整语句的执行过程。通过hint，我们可以调整：

优化器类型、优化器优化目标、数据读取方式（访问路径）、查询转换类型、表间关联的顺序、表间关联的类型、并行特性、其他特性

**使用**

对于hint的使用注意一点：不要过分依赖hint

当遇到SQL执行计划不好的情况，应优先考虑统计信息等问题，而不是直接添加hint了事。如果统计信息无误，应该考虑物理结构是否合理，即没有合适的索引。只有在最后仍然不能SQL按优化的执行计划执行时，才考虑hint。

毕竟使用hint，需要应用系统修改代码，hint只能解决一条SQL的问题，并且由于数据分布的变化或其他原因（如索引更名）等，会导致SQL再次出现性能问题。

**特点**

Hint存在一些弊端：

1. Hint是比较“暴力”的一种解决方式，不是很优雅。需要开发人员手动修改代码。
2. Hint不会去适应新的变化。比如数据结构、数据规模发生了重大变化，但使用hint的语句是不会感知变化并产生更优的执行计划。
3. Hint随着数据库版本的变化，可能会有一些差异，设置废弃的情况。此时，语句本身是无感知的，必须人工测试并修正。

# 事务

# 视图

**同义词**

**概述**

Oracle的同义词（synonyms）从字面上街就是别名的意思，和视图的功能类似，就是一种映射关系。它可以节省大量的数据库空间，对不同用户的操作同一张表没有多少差别，它拓展了数据库的使用范围，能够在不同的数据库用户之间实现无缝交互。

同义词是数据库对象的一个别名，经常用于简化对象访问和提高对象访问的安全性。在使用同义词时，oracle数据库将它翻译成对应方案对象的名字。与视图类似，同义词并不占用实际存储空间，只有在数据字典中保存了同义词的定义。在oracle数据库中的大部分数据库对象，如表、视图、序列、函数、存储过程、同义词等等，数据库管理员都可以根据实际情况为它们定义同义词。

**分类**

Oracle同义词分为两种类型，分别是oracle共用同义词与oracle私有同义词。普通用户创建的同义词一般都是私有同义词，公有同义词一般由DBA创建，普通用户如果希望创建同义词，则需要CREATE PUBLIC SYNONYM 这个系统权限。

1. oracle公有同义词：由一个特殊的用户组public所拥有。顾名思义，数据库中所有的用户都可以使用公用同义词。公用同义词往往用来标示一些比较普通的数据库对象，这些对象往往大家都需要引用。
2. oracle私有同义词：它跟公有同义词相对应，由创建它的用户所有。当然，这个同义词的创建者，可以通过授权控制其他用户是否有权使用属于自己的私有同义词。

**创建**

如果要创建远程数据库上的一张表的同义词，需要先创建一个Database Link（数据库连接）来拓展访问，然后再使用如下语句创建数据库同义词：

create synonym table\_name for table\_name@DB\_Link;

必须给在user用户中给当前用户（user1）授权:

grant select/delete/update on user1

创建同义词SQL语句：

create synonym table\_name for user1.table\_name;

**删除**

删除同义词的SQL语句：

drop public synonym table\_name;

**查看**

查看同义词的SQL语句：

select \* from dba\_synonyms;

**特点**

同义词具有以下的优点：

1. 多用户协同开发中，可以屏蔽对象的名字及其持有者。如果没有同义词，当操作其他用户的表时，必须通过user\_name.object\_name的形式，采用了oracle同义词之后就可以屏蔽掉user\_name。

当然，这里需要注意的是：public同义词只是为数据库对象定义了一个公共的别名，其他用户能够通过这个别名访问这个数据库对象，还要看是否已经为这个用户授权。

1. 为用户简化SQL语句。上面的一条其实就是一种简化SQL的体现，同时如果自己建的表的名字很长，可以为这个表创建一个oracle同义词来简化SQL开发。
2. 为分布式数据库的远程对象提供位置透明性。

**使用**

# 游标

# 异常处理

# 存储过程

# 触发器

# 设计范式