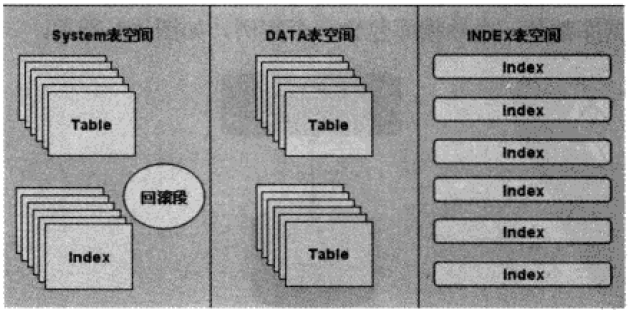
Oracle学习指南

数据库中每一行成为记录，每一列称为字段。

Oracle数据库系统中，用于存放数据库表，索引，回滚段等对象的磁盘逻辑空间叫表空间（tablespace）。



Oracle系统安装并创建Oracle实例后，系统会自动建立多个表空间。主要创建的表空间如下：

**SYSTEM表空间**：SYSTEM表空间用于存放内部表和数据字典的数据，如表名，列名，用户名等，不赞成将用户创建的表、索引等存放在SYSTEM表空间中。

**SYSAUX表空间**：用户存放Oracle系统内部的常用样例用户的对象，如存放CRM用户的表和索引等。SYSAUX表空间一般不存储用户的数据。

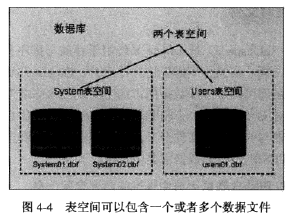
**撤销表空间**：用于存储撤销信息的表空间，所有的增删改操作，Oracle都会自动调用撤销表空间自动存储操作之前的数据，当所有的修改操作完成并提交（commit）后，Oracle系统可根据需要保留修改前的数据时间长短释放撤销表空间的部分空间。

**USER表空间**：USER表空间是Oracle建议用户使用的表空间，可以在这个表空间上创建各种兑点，如创建表、索引等，Oracle的基本样例用户SCOTT的对象就存放在USERS表空间中。

**数据文件**：用于保存用户应用程序和Oracle系统内部数据的文件，Oracle数据库由表空间组成，每个表空间可以包含一个或者多个数据文件。

表空间是一个数据库的逻辑区，每个表空间由一个或多个数据文件组成，一个数据文件只能属于一个表空间。

**可以在Oracle实例上创建多个表空间，每个表空间由一个或多个数据文件（物理文件）组成（即dbf文件）。**



数据文件可以存放下面两种类型的数据：

系统数据：管理用户数据和Oracle系统本身的数据：暴扣用户建立的表的名字、列的名字以及字段类型等等。这些数据存放在system01.dbf文件中。

用户数据：用户应用系统的数据。

**临时表空间：Oracle系统用于存放与排序有关的特殊表空间。**

**控制文件：**存储Orcale实例信息、数据文件和日志文件信息，desc v$controlfile;

日志文件分为重做日志文件（主体）和归档日志文件。

重做日志文件也叫做联机日志文件（Online Redo Log File），用来记录数据库所有发生过的变更信息以及Oracle内部行为而引起的数据库变化信息。在数据恢复时可以从该日志文件读出原来的交易数据。当用户发出commit命令时，数据库会将每笔变更记录到日志文件中，写入日志文件成功后才会完成信息传给用户，**所以日志文件系统上可以随时读出信息一恢复某些交易数据。对表或者整个表空间设定NOLOGGING属性时，所有表的DML操作都不会生成日志信息。**

**归档体制文件：**Oracle系统运行有两种模式，归档模式和非归档模式，非归档模式就是在系统运行期间，所产生的日志文件信息不断地记录到各个日志文件组中，当所有的日志文件组写满之后又重新从第一个日志组开始覆写日志信息内容，归档模式就是在即将要被覆盖前，将即将被覆盖的日志文件中的日志信息读出，并写道归档日志中，以便以后的恢复，默认情况下Oracle系统都不采用归档模式，这主要是因为归档模式带来一定的性能问题。

数据字典是存放整个数据库实例重要信息的一组表，这些数据字典多数都归于sys用户所有，数据字典由前缀和后缀组成，使用下划线连接，代表含义如下：

USER\_：记录用户的对象信息

ALL\_：记录用户的对象信息以及被授权访问的对象信息。

DBA\_：包含数据路实例的所有对象信息

V$\_：当前实例的动态视图

GV\_：分布式环境下所有实例的动态视图。

|  |  |
| --- | --- |
| 数据字典名称 | 说明 |
| DBA\_TABLES(=TABS) | 所有用户数的所有标信息 |
| DBA\_COLUMNS(=COLS) | 所有用户表的列信息 |
| DBA\_VIEWS | 所有用户的所有视图 |
| DBA\_SYNONYMS(=SYN) | 所有用户同义词信息 |
| DBA\_SEQUENCE(=SEQ) | 所有用户序列信息 |
| DBA\_CONSTRAINTS | 所有用户约束信息 |
| DBA\_INDEX(=IND) | 所有用户索引简要信息 |
| CAT | 当前用户可以访问的所有基表 |
| TAB | 当前用户创建的所有基表、视图、同义词等 |
| DICT | 构成数据字典的所有表信息 |

**在一条语句中可以用任何数量的列进行分组，在选择清单中的列必须是group by子句中有或者有一个函数使用它。**

**Group by还可以使用having字句选择特殊的组，having字句将组的一些属性与常数值比较，只显示满足having子句的查询结果。（注意一点如果查询语句中出现group by，having和select的相同点为，后面跟的必须是group by的一项或者非一项的函数操作，而且having必须放置在where之后）**

SQL语句主要有以下几种类型：

**SELECT语句：**在SQL所有语句中，SELECT语句功能和语法最为复杂，同时也最为灵活。

**数据查询（Data Manipulation Language DML）：**改变数据库数据，包括INSERT,UPDATE和DELETE三条语句。

**事务控制语言（Transaction Control Language TCL）：**用于维护数据的一致性，包括COMMIT,ROLLBACK和SAVEPOINT三条语句。其中COMMIT语句用于确认已经进行的数据库改变，ROLLBACK用于取消已经进行的数据可改变，而SAVEPOINT用于设置保存点，以取消部分数据库改变。

**数据定义语言（Data Definition Language）：**用于建立、修改和删除数据库对象，例如使用CREATE TABLE可以创建表，使用ALTER TABLE可以修改表，使用DROP TABLE可以删除某个表，**需要注意的是，DDL语句会自动提交事务。**

数据控制语言（Data Control Language）：用于执行权限授予和权限收回操作，包括GRANT和REVOKE两条命令，其是GRANT命令用于给某个用户赋予权限，而REVOKE用于回收用户的权限。**需要注意的是，DCL语句也会自动提交事务。**

Oracle Client软件。这是使用PL/SQL Developer的必须条件。从Oracle 10G开始,Oracle提供了一个较为轻量级的客户包，叫做Oracle Instant Client。将它安装好后，就不用再安装庞大的Oracle Client了。Oracle Client不光包括instant client，还包括很多工具。在PL/SQL中需要设置相应的环境变量：PATH-需要包含Instant Client的根目录（oci.dll所在位置）；TNS ADMIN-需要指出tnsnames.ora所在位置的根目录；NLS LANG-为客户设定语言，区域和特征。

SQL语句：

通配符：

%可以用来匹配任何长度的字符串。

\_可以

**SYSTEM和SYS用户的区别：**

**SYSTEM**是数据库内置的一个普通管理员，你手工创建的任何用户在被授予dba角色后都跟这个用户差不多。

**SYS**用数据库的超级用户，数据库内很多重要的东西（数据字典表、内置包、静态数据字典视图等）都属于这个用户，**SYS**用户必须以**SYSDBA**身份登录。

**SYSTEM**用户以**SYSDBA**身份登录时就是**SYS**，准确地说，任何用户以**SYSDBA**身份登录时都是**SYS**，这一点，登陆后执行show user可以验证。

**Oracle数据库三种登陆方式：**

**NORMAL** 普通用户，一般程序员使用 Normal 方式登陆。

**SYSOPER** 数据库操作员，主要包括 打开数据库服务器，关闭数据库服务器，备份数据库，恢复数据库，日志归档，会话限制

**SYSDBA** 数据库管理员，打开数据库服务器 关闭数据库服务器，备份数据库，恢复数据库，日志归档， 会话限制，管理功能，创建数据库

**集合操作，UNION UNION ALL INTERSECT MINUS**

**UNION为集合的并集，并且不删除重复项**

**UNION为集合的并集，并且删除重复项**

**INTERSECT为集合的交集  
 MINUS表示集合的差，只会显示第一个集合中存在，第二个集合中不存在的数据。**

Oracle数据库可以分为两部分。

1. 实例：实例是一个非固定的，基于内存在基本进程与内存结构，当服务器关闭后，实例也就不存在了。
2. 数据库：数据库只的是固定的，基于磁盘的数据文件、控制文件、日志文件、参数文件和归档日志文件，一般情况下，Oracle数据库都是一个数据库包含一个实例。

* **Case when的使用：**

1. 简单case语句，使用表达式确定返回值。

Case object\_id

When 1 then ‘book’

When 2 then ‘video’

Else ‘magazine’

End

1. 搜索case 语句，使用条件确认返回值。

Case

When object\_id =1 then ‘book’

When object\_id =2 then ‘video’

Else ‘magazine’

End

同义词和视图的区别：

视图和同义词分别都是数据库中的对象名称，它们都不对应实际的数据存储，都依赖其他对象的存在而存在。

视图：视图可以被看成是虚拟表或存储查询。除非是索引视图，否则视图的数据不会作为非重复对象存储在数据库中。一般是基于一个或多个数据表经数据查询语句构建而成；视图可以再被用于数据查询语句中。

同义词：是架构范围内的对象的另一名称。通过使用同义词，客户端应用程序可以使用由一部分组成的名称来引用基对象，而不必使用由两部分、三部分或四部分组成的名称。可以理解为是对数据库中一个对象的快捷方式或者在特定范围内的别名；不限于数据表，几乎所有的数据库对象都可以建立同义词。

**视图和同义词区别：**

1.视图可以对应一张或多张表，同义词只能对应一张表名称

2.视图可以设置其他约束条件

3.可以在同义词上建立视图：

例如，已经存在一个名为synonym\_city的同义词和一个名为table\_citys的表，可以创建视图：CREATE OR REPLACE NOFORCE VIEW view\_city AS SELECT \* FROM synonym\_city,table\_citys;

4.可以通过对同义词进行任何DML操作，对表数据进行插入、更新、删除，通过对视图只能进行更新和删除操作。

* **存储过程**

典型的存储过程由如下几个部分组成：

Declare：声明变量，游标，用户自定义类型和异常

Begin：程序开始标志，表示开始执行sql和pl/sql程序

Exception：异常处理部分，处理sql和pl/sql语句处理中的异常部分。

End：程序结束标志。

SQL优化：

使用not exists替代not in，当not in 后跟自查询，并且查询的结果集较多时，不宜使用not in，但是如果not in 后面的括号内是列表（可枚举的几个）或子查询所满足结果集很少时也是可以使用。

使用 exists替代in