

Oracle实战

数据库基础和Oracle概述

1. 数据库 - 数据的集散地，长久保存并高效的检索数据
2. 关系型数据库 - 基于关系代数和集合论，用二维表组织数据
 - 行：一条记录
 - 列：一个字段 - 主键：唯一标识一条记录
 - 外键：其他表的主键
3. 数据完整性
 - 实体完整性
 - 参照完整性
 - 域完整性
4. 实体关系图（E-R图）
5. 规范化（范式理论）
6. SQL - 结构化查询语言，关系型数据库编程语言
 - DDL：数据定义语言（create、drop、alter）
 - DML：数据操纵语言（insert、delete、update）
 - DQL：数据查询语言（select）
 - DCL：数据控制语言（grant、revoke、commit、rollback、savepoint）

Oracle的体系结构

1. Oracle实例：一个非固定的、基于内存的基本进程。Oracle实例由SGA（系统全局区）和后台进程组成，一个Oracle服务器上可以配置多个实例
2. Oracle数据库：由数据文件、控制文件、日志文件、参数文件和归档文件构成的整体，一个数据库包含一个实例，数据库通过实例来管理。
3. 表空间（Tablespace）：用于存放数据表、索引、回滚段等对象的逻辑空间，一个表空间可以包含一个或多个数据文件，一个数据文件只能属于一个表空间。表、索引等占用表空间的对象称为Segment，Segment又是由很多的Extent构成的，而Extent又是由多个Block构成的。

- SYSTEM表空间：存放Oracle系统内部表和数据字典
- SYSAUX表空间：存放样例用户对象
- UNDO表空间：存放增删改之前的数据
- USERS表空间：建议用户使用的表空间
- TEMP表空间：存放临时数据

```
select file_name, tablespace_name, bytes from dba_data_files;
```

```
select file_name, tablespace_name from dba_temp_files;
```

4. 控制文件：存储Oracle实例信、数据文件和日志文件信息的二进制文件，在Oracle实例创建时必须访问控制文件。

```
select name, status from v$controlfile;
```

5. 日志文件：分为重做日志文件和归档日志文件，出于数据安全方面的考虑，每个Oracle实例用一个日志线程来记录数据库的变化，当用户发出commit命令时，数据库将每笔交易都记录到日志文件中，从日志文件可以取出用于恢复数据的信息。

```
select group#, status, member from v$logfile;
```

```
select dbid, name, log_mode from v$database;
```

```
archive log list;
startup mount;
alter database archivelog;
```

6. 参数文件 / 密码文件 / 跟踪文件 / 警告日志

7. SGA：主要由三个部分构成，分别是数据缓冲区、日志缓冲区、共享池。

8. 后台进程

- DBWR、LGWR、SMON、PMON、ARCH、LCK、.....

```
select name, description from v$bgprocess;
```

9. PGA / UGA / CGA

10. 数据字典

- 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_: 数据库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
- 常用数据字典:
 - dba_tables / tabs: 所有用户的表信息
 - dba_tab_columns / cols: 所有用户的列信息
 - dba_views: 所有用户的视图信息
 - dba_synonyms / syn: 所有用户的同义词信息
 - dba_sequences / seq: 所有用户的序列信息
 - dba_constraints: 所有表的约束信息
 - dba_indexes / ind: 所有索引的简要信息
 - dba_index_columns: 所有索引的列信息
 - dba_triggers: 所有用户的触发器信息
 - dba_source: 所有用户的存储过程信息
 - dba_objects: 所有用户对象的基本信息（表、索引、视图、序列等）

SQL

1. 函数的使用

- 字符函数: concat / length / lengthb / lower / upper / substr / ltrim / rtrim / trim / translate / replace / lpad / rpad / initcap
- 数字函数: abs / ceil / floor / round / trunc
- 日期函数: add_months / last_day / months_between / sysdate
- 转换函数: to_date / to_number / to_char
- 其他函数: nvl / decode

2. 集合操作: union / intersect / minus / in / any / all

3. 常用数据类型

- varchar2(n): n<=4000

- char(n): n<=2000
- number(p,s): 整数和浮点数, p+s<=38
- binary_float: 浮点型, 比number效率高
- binary_double: 双精度浮点型, 比number效率高
- date: 日期时间
- timestamp: 时间戳
- blob: 二进制大对象
- clob: 字符大对象
- bfile: 外部二进制文件

```
select type_name from dba_types where owner is null;
```

4. 索引: 索引是关系型数据库中存放表中每条记录位置的对象, 相当于一本书的目录, 可以用来加速查询, 但是更新和删除数据会更新索引因此对更新和删除操作性能有一定影响, 同时会使用更多的存储空间。索引要建在被查询得最多的列上才能起到更好的作用。

```
create unique index idx_goods_id on TbGoods (goodsid);
create index idx_goods_name on TbGoods (gname);
alter index idx_goods_id rebuild storage (initial 1M next 512k);
alter index idx_goods_id rebuild reverse;
drop index idx_goods_name coalesce;
```

5. 视图: 视图是查询结果的存储, 相当于一张虚拟的表, 可以简化查询所使用的SQL语句, 此外还可以起到安全和保密的作用 (将访问限制在指定的列上)。

```
create or replace view view_name as select ...
    [with check option]
    [with read only]
drop view view_name;
```

6. 序列: 通过序列可以自动生成一个的整数, 一般用于自增主键。

```
create sequence seq_name start with a increment by b minvalue c maxvalue d cache e cycle;
select seq_name.nextval from dual;
select seq_name.currval from dual;
```


7. 同义词：数据库对象的别名。

```
create public synonym emp for scott.emp;  
drop synonym emp;
```

8. SQL-99: cross join / natural join / using

9. SQL练习

- 查询薪资最高的员工姓名和工资
- 查询员工的姓名和年薪 ((月薪+补助)*12)
- 查询入职超过30年的员工
- 查询有员工的部门的编号和人数
- 查询所有部门的名称和人数
- 查询薪资最高的员工(Boss除外)的姓名和工资
- 查询薪水超过平均薪水的员工的姓名和工资
- 查询薪水超过其所在部门平均薪水的员工的姓名、部门编号和工资
- 查询部门中薪水最高的人姓名、工资和所在部门名称
- 查询主管的姓名和职位
- 求平均薪水最高的部门的名称和平均工资
- 查询薪资排名前5的员工姓名和工资
- 查询薪水排名6-10名的员工姓名和工资
- 求薪水最低的部门经理所在部门的名称

10. 触发器

- 例子1: 删除员工时转移员工记录

```
CREATE TABLE emp_his AS SELECT * FROM EMP WHERE 1=2;
```

```
CREATE OR REPLACE TRIGGER tr_del_emp  
BEFORE DELETE ON scott.emp FOR EACH ROW  
BEGIN  
    INSERT INTO emp_his  
    (deptno, empno, ename, job, mgr, sal, comm, hiredate)  
    VALUES (:old.deptno, :old.empno, :old.ename, :old.job
```

```
, :old.mgr, :old.sal, :old.comm, :old.hiredate );  
END;
```

```
DELETE emp WHERE empno=7788;
```

- 例子2: 限定只能在指定时间修改部门信息

```
CREATE OR REPLACE TRIGGER tr_dept_time  
BEFORE INSERT OR DELETE OR UPDATE ON departments  
BEGIN  
    IF (TO_CHAR(sysdate, 'DAY') IN ('星期六', '星期日'))  
        OR (TO_CHAR(sysdate, 'HH24:MI') NOT BETWEEN '08:30' AND  
D '18:00') THEN  
        RAISE_APPLICATION_ERROR(-20001, '不是上班时间, 不能修  
改departments表');  
    END IF;  
END;
```

PL/SQL

1. 概述: 过程化语言与SQL的结合, 它扩展了SQL, 提供了更强大的功能、更好的性能、更高的安全性。
2. 基本语法
 - 变量和常量
 - 变量属性: %type、%rowtype
 - 表达式和运算符
 - 算术运算符
 - 关系运算符
 - 逻辑运算符
 - 其他运算符: :=、||
 - 流程控制
 - if-then-elsif-then-else
 - case-when
 - loop
 - for-loop

- while-loop
- if-goto

3. 过程和函数

4. 错误处理

- 预定义异常
 - DUP_VAL_INDEX
 - NO_DATA_FOUND
 - TOO_MANY_ROWS
 - ZERO_DIVIDE
 - VALUE_ERROR
 - CASE_NOT_FOUND
- 抛出异常 raise

5. 包

- 包定义
- 包体

6. 集合

7. 游标

- 普通游标
- 可更新游标
- 带参数游标
- 游标引用

8. 新特性

Oracle中的SQL优化

常用技巧

1. 不要用*代替列名
2. 用truncate代替delete
3. 在确保完整性的情况下尽早commit
4. 减少表查询的次数
5. 用exists / not exists 替换in / not in / distinct
6. 有效利用共享游标
7. 以合理的方式使用函数

表连接的方式

1. 驱动表的选择
2. where条件的顺序

使用索引

1. 索引列的选择
2. 复合索引的运用
3. 避免大表的全表扫描
4. 监控索引是否被使用

```
alter index idx_name monitoring usage;
```

执行计划管理和SQL重演

闪回、导入/导出、备份和恢复

1. 回收站
2. 闪回操作
3. 导出
4. 导入
5. 复制表数据
6. 恢复表数据

```
show parameter pfile;
```

数据挖掘
