

# Oracle 考试重点

有错的地方方概不负责。

---

## 题型及分数

- 填空 10\*2'
- SQL 3\*5'
- 简答 6\*5'
- PL/SQL 3\*10'

简答题：

### 1.数据库，表空间，数据文文件的关系

- 一个 **Oracle** 数据库是由一个或多个表空间的逻辑存储单位构成的，这些表空间共同来存储数据库的数据。
- **Oracle** 数据库的每个表空间由有一个或多个被称为数据文文件的物理理文件构成，这些文文件由 **Oracle** 所在的操作系统管理理。
- 数据库的数据实际存储在构成各个表空间的数据文文件中。

### 2.角角色色的定义，使用用的原因，查看角角色色信息的 SQL 语句句

- 角角色色是一组相关权限的集合，可将其授予 **Oracle** 数据库用用户或其他角角色色以简化权限的管理理。
- 可将其授予 **Oracle** 数据库用用户或其他角角色色以简化权限的管理理。
- `select * from user_role_privs`

### 3.启动数据库的几种模式

- 启动例程，但不装载数据库
- 启动例程并装载数据库，但保持数据库关闭状态
- 启动例程，装载并打开数据库

#### 4.关闭数据库的几种模式及方式

- 正常 shutdown normal

立即 shutdown immediate

- 事务 shutdown transactional
- 强制 shutdown abort

#### 5.控制文件包含哪些内容

- 数据库名称
- 数据库创建的时标
- 数据文件和联机重做日志文件的名称和位置
- 表空间的名字
- 当前日志的序号
- 最近检查点信息
- 恢复管理器 信息 ( RMAN )

#### 6.Oracle 的两种服务器 模式和适用条件

- 专用服务器 模式，适用于只有少量客户端连接系统或数据仓库系统，或者联机事务处理系统。
- 共享服务器 模式，适用于连接用户较多，事务都比较小，或者服务器 内存资源较少的情况下。

SQL:

##### 1.数据库链接的创建

connect 用户名/口令@主机字符串

例如：connect scott/tiger@orcl

##### 2.SQL 聚合操作

- Avg 平均值：select prd\_no, avg(qty) from saler group by prd\_no;

- Count 记录数 : select count(\*) from sales;
- Max 最大值 : select prd\_no, max(qty) from sales group by prd\_no;
- Min 最小值 : select prd\_no, min(qty) from sales group by prd\_no;

Sum 总和 : select prd\_no, sum(qty) from sales group by prd\_no; 3.嵌套 ( 子子查询 )

- eg.查询哪些雇员的工资高于他所在部门的平均工资 :

```
Conn scott/tiger@orcl
```

```
Select deptno, ename, sal from emp outer
```

```
Where sal > (select avg(sal) from emp where deptno = outer.deptno) Order
by deptno;
```

#### 4.修改 ( update? )

语法 : update table set column = value [ , column = value, ... ] [ where condition];

- eg.修改 part\_emp 表 , 将 7654 号雇员调整为与 7934 号雇员一个部门

```
Conn scott/tiger@orcl
```

```
Update part_emp set deptno =
```

```
(select deptno from part_emp where empno=7934) where empno = 7654; Commit;
```

PL/SQL:

#### 1.存储过程

例子 : create or replace procedure query\_emp

( p\_no in emp.empno%type, p\_name out emp.ename%type, p\_sal out emp.sal%type, p\_comm out emp.comm%type )

is

begin

n

```
select name,sal,comm
```

```
into p_name,p_sal,p_comm
```

```
from emp
```

```
where empno = p_no;
```

```
end query_emp;
```

#### 2.触发器

```
CREATE OR REPLACE TRIGGER tr_del_emp
BEFORE DELETE --指定触发时机为删除操作前触发
ON scott.emp
FOR EACH ROW --说明创建的是行级触发器
```

```

BEGIN
    --将修改前数据插入到日志记录表 del_emp ,以供监督使用。
    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;
DROP TABLE emp_his;
DROP TRIGGER del_emp;
/

```

### 3.游标

```

declare
    --类型定义
    cursor c_job
    is
    select empno,ename,job,sal
    from emp
    where job='MANAGER';
    --定义一个游标变量 v_cinfo c_emp%ROWTYPE , 该类型为游标 c_emp 中的一行数据类型
    c_row c_job%rowtype;
begin
    for c_row in c_job loop
        dbms_output.put_line(c_row.empno|| '-' ||c_row.ename|| '-' ||c_row.job|| '-'
||c_row.sal);
    end loop;
end;

```