

# 一、名词解释

## 1.关系数据模型（Relational Model）

关系模型是一种逻辑结构模型，用于表示数据在计算机系统中存储时所采用的逻辑结构和特征。

## 2.实例（Instance）

实例是一组用于管理数据库文件的内存结构，实例存在于内存中，（数据库存在于磁盘上）

## 3.数据字典（Data Dictionary）

数据库字典是Oracle存放有关数据库信息的地方，几乎所有的系统信息和对象信息都可以在数据库字典中进行查询。数据库字典是数据库系统的信息核心。它随数据库的建立而建立，随数据库的更新而更新。

## 4.事务（Transaction）

事务是若干操作的集合，集合中的所有操作作为一个整体，是数据处理的模型，是数据处理的基本单位，没有事务之外的操作。

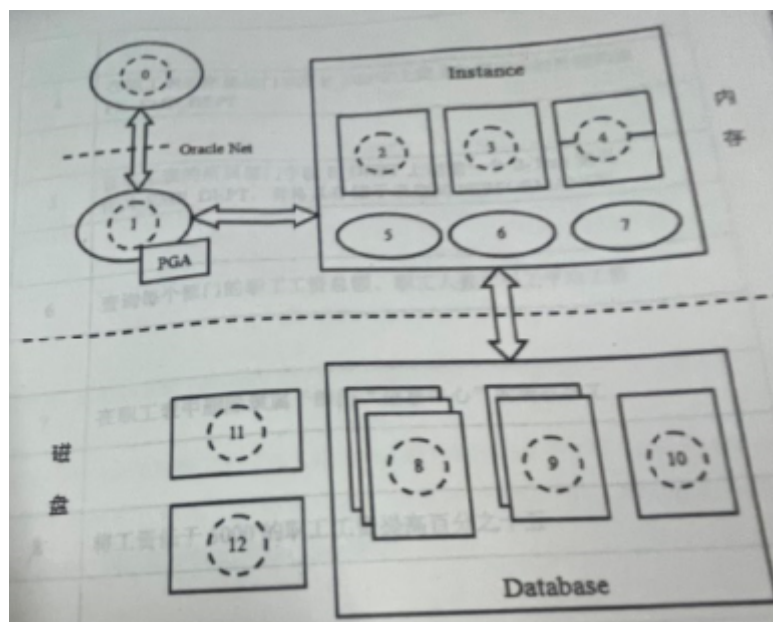
## 5.触发器（Tigger）

触发器在数据库中以独立的对象存储，触发器需要一个事件来启动运行。当某个事件发生时，触发器自动的隐式的运行。并且，触发器不能接受参数。

# 二、表空间类型及解释或内容说明

序号	Tablespace Type	Explanation or Description
1	系统表空间 (system和sysaux)	存放Oracle正常运行所需要的系统数据
2	undo表空间	存放数据库运行过程中产生的undo数据，从而实现数据的多版本读取以及数据库崩溃时的数据恢复
3	临时表空间	存放临时数据
4	用户表空间	存放用户数据
5		

# 三、数据库的体系结构图



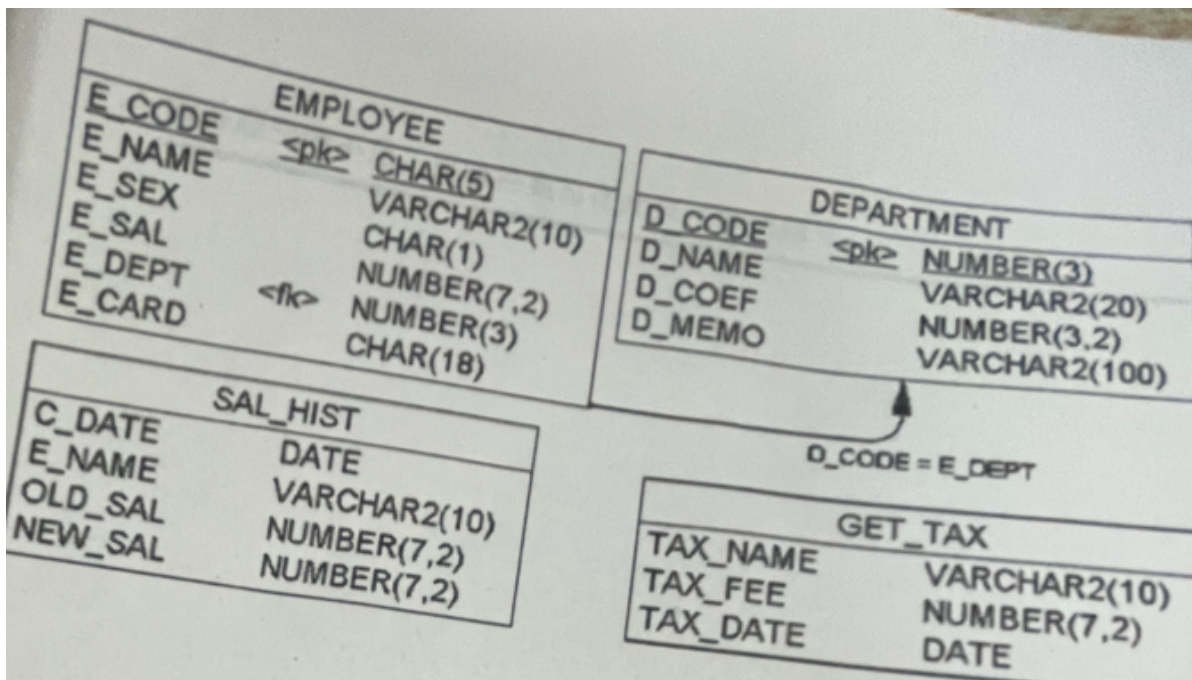
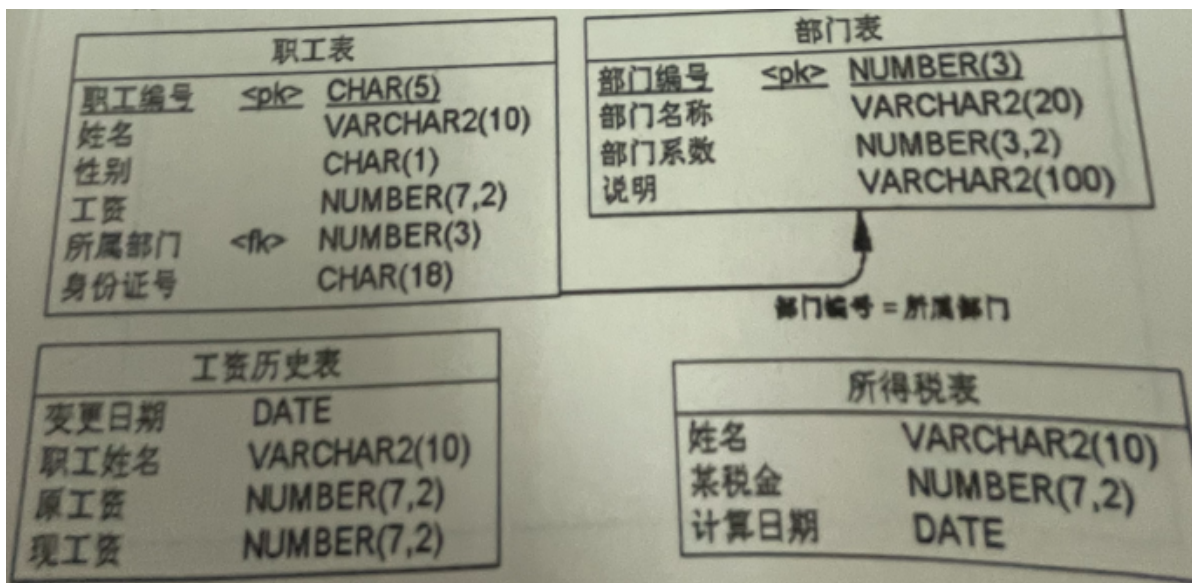
数字	英文名称 (中文说明)
1	Server process (处理用户连接提出的各种要求)
2	Database buffer cache (存放从数据文件读取的数据, 多次访问的数据保留于此, 下次访问不必从磁盘中重新读取)
3	Redo log buffer (由内存到内存写的速度快, 先将Redo log存在缓存中)
4	Shared pool (存储多种程序数据)
5	DMON (释放异常关闭所占用的内存、锁等资源; 检查调度器和服务器进程是否异常关闭)
6	SMON (程序发生故障重启时, 执行实例恢复)
7	DBWR (把数据缓存区内修改过的数据写入磁盘上的数据文件)
8	Data files (存放数据)
9	control files (存储数据库物理文件信息和RMAN备份信息等资源)
10	redo log files (所有对数据文件内容的改变都会记录到重做log中)
11	Parameter file (数据库启动时需要的重要参数)
12	Diagnostic files (数据库运行状态, 诊断错误)

#### 四、结合第五题给出的库表写出对应的SQL语句

序号	操作要求
1	创建职工所得税表GET_TAX
	<pre>create table get_tax(tax_name varchar2(10), tax_fee number(7,2), tax_date date)</pre>
2	为部门表DEPARTMENT添加一个位置字段LOC
	<pre>alter table department add loc char(10)</pre>
3	在职工表EMPLOYEE上定义主键约束PK_EMPLOYEE
	<pre>alter table employee add constraint pk_c primary key(code)</pre>
4	在职工表的所属部门字段E_DEPT上定义如图所示的外键约束FK_EMP_DEPT
	<pre>alter table employee add constraint fk_emp_dept foreign key(dept) references department(code)</pre>
5	在职工表的所属部门字段E_DEPT上创建一个B_Tree索引IDX_EMP_DEPT，并将其存储于表空间PERFORMANCE
	<pre>create index idx_emp_dept on department(dept) tablespace performance</pre>
6	查询每个部门的职工工资总额、职工人数和职工平均工资
	<pre>select count(sal) as 工资总额, count(name) as 职工人数, avg(sal) as 平均工资 from employee</pre>
7	在职工表中删除隶属于部门“信息中心”的所有职工
	<pre>delete from employee where d_name = '信息中心'</pre>
8	将工资低于5000的职工工资提高百分之十五
	<pre>update employee set sal = sal*1.15 where e_sal &lt; 5000</pre>
9	将步骤5、6、7、8的DML操作结果永久保存
	<pre>commit</pre>
10	以“工号、职工姓名、部门名称”的形式查询出所有职工信息
	<pre>select e.code as 工号, e.name as 职工姓名, e.dept as 部门名称 from employee e</pre>
11	查询工资低于单位平均工资的所有员工
	<pre>select * from employee where sal&lt;(select avg(sal) from employee)</pre>
12	查询那些工资高于所在部门平均工资的所有职工
	<pre>select * from employee e where sal&gt;(select avg(sal) from employee where e.dept = dept)</pre>
13	快速清空工资历史表SAL_HIST中的所有记录

序号	操作要求
	<code>truncate table sal_hist</code>
14	以管理员身份登陆系统，将SCOTT用户下的所得税表GET_TAX的查询权限授予给CCEC用户
	<code>sqlplus / as sysdba</code> <code>grant select on get_tax to ccec</code>
15	取消CCEC用户在工资历史表SAL_HIST上的数据增、删、改的操作权限
	<code>revoke insert,updat,delete on sal_hist from ccec</code>

## 五、编写对应的PL/SQL语句



1、编写一个用来根据职工工资收入计算职工税收额度的存储过程calculate\_tax求将计算结果存入职工“所得税”表，已知两个参数：职工缴税的起点income\_base，超过部分按比例tax\_rate缴纳

```
CREATE OR REPLACE PROCEDURE COMPUTE_TAX(income_base number, tax_rate number)
as
    curser c_emp is select name,sal from employee;
    v_emp c_emp%rowtype;
    income number(7,2);
begin
    open c_emp;
    fetch c_emp into v_emp;
    while c_emp%found loop
        income := v_emp.sal;
        if income > income_base then
            insert into get_tax values(
                v_emp.name,tax_rate*(income-income_base),sysdate);
        else
            insert into get_tax values(
                v_emp.name,0,sysdate);
        end if;
        fetch c_emp into v_emp;
    end loop;
    commit;
    close c_emp;
end;
```

2、在职工表上创建一个触发器salary\_rec，用来自动记录工资变更信息，表更前后的工资信息记入“工资历史表”

```
CREATE OR REPLACE TRIGGER SALART_REA
AFTER UPDATE OF SAL ON EMPLOYEE
FOR EACH ROW
begin
    insert into sal_hist values(sysdate, :new.name, :old.sal, :new.sal);
end;
```

声明：仅学习过程使用，其他一起用途与作者无关