1、根据案例数据库中数据库表、视图、序列的设计，在数据库HUMAN\_RESOURCE的HUMAN模式下创建各种对象。

创建如下表：

（1）REGIONS表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| region\_id | NUMBER |  | PRIMARY KEY | 区域编号 |
| region\_name | VARCHAR2 | 25 | NOT NULL | 区域名称 |



（2）COUNTRIES表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| country\_id | CHAR | 2 | PRIMAR KEY | 国家编号 |
| country\_name | VARCHAR2 | 40 | NOT NULL | 国家名称 |
| region\_id | NUMBER |  | FOREIGN KEY | 所属区域编号 |



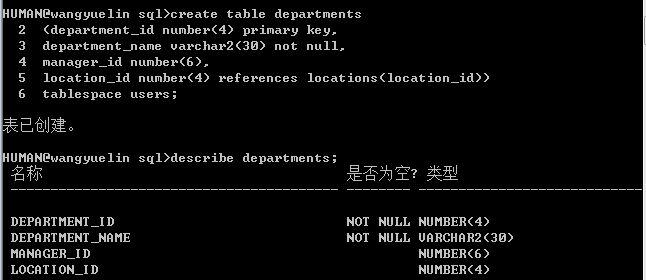
（3）LOCATIONS表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| location\_id | NUMBER | 4 | PRIMARY KEY | 位置编号 |
| street\_address | VARCHAR2 | 40 |  | 街道 |
| postal\_code | VARCHAR2 | 12 |  | 邮政编码 |
| city | VARCHAR2 | 30 | NOT NULL | 城市名称 |
| state\_province | VARCHAR2 | 25 |  | 州或省名称 |
| country\_id | CHAR | 2 | FOREIGN KEY | 所属国家编号 |



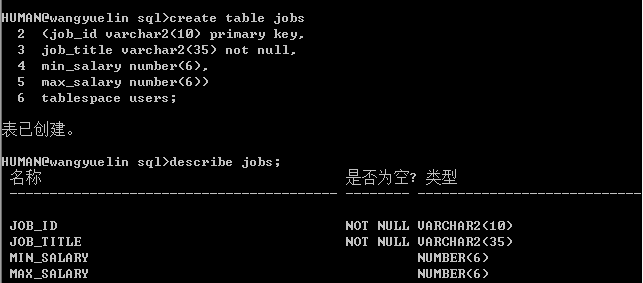
（4）DEPARTMENTS表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| department\_id | NUMBER | 4 | PRIMARY KEY | 部门编号 |
| department\_name | VARCHAR2 | 30 | NOT NULL | 部门名称 |
| manager\_id | NUMBER | 6 |  | 部门管理者编号 |
| location\_id | NUMBER | 4 | FOREIGN KEY | 部门所属位置编号 |



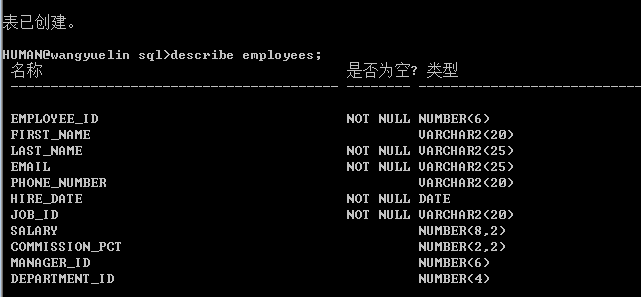
（5）JOBS表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| job\_id | VARCHAR2 | 10 | PRIMARY KEY | 职位编号 |
| job\_title | VARCHAR2 | 35 | NOT NULL | 职位名称 |
| min\_salary | NUMBER | 6 |  | 职位最低工资 |
| max\_salary | NUMBER | 6 |  | 职位最高工资 |



（6）EMPLOYEES表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| employee\_id | NUMBER | 6 | PRIMARY KEY |  |
| first\_name | VARCHAR2 | 20 |  |  |
| last\_name | VARCHAR2 | 25 | NOT NULL |  |
| email | VARCHAR2 | 25 | NOT NULL  UNIQUE |  |
| phone\_number | VARCHAR2 | 20 |  |  |
| hire\_date | DATE |  | NOT NULL |  |
| job\_id | VARCHAR2 | 10 | NOT NULL  FOREIGN KEY |  |
| salary | NUMBER | 8,2 | 大于0 |  |
| commission\_pct | NUMBER | 2,2 |  |  |
| manager\_id | NUMBER | 6 | FOREIGN KEY |  |
| department\_id | NUMBER | 4 | FOREIGN KEY |  |



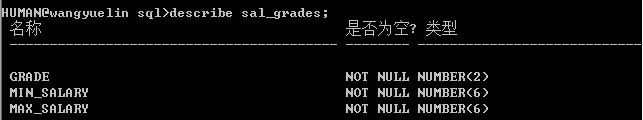
（7）JOB\_HISTORY表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| employee\_id | NUMBER | 6 | PRIMARY KEY |  |
| start\_date | DATE |  | PRIMARY KEY |  |
| end\_date | DATE |  | NOT NULL  end\_date>start\_date |  |
| job\_id | VARCHAR2 | 10 | NOT NULL |  |
| department\_id | NUMBER | 4 | FOREIGN KEY |  |



（8）SAL\_GRADES表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| grade | NUMBER | 2 | PRIMARY KEY | 工资等级 |
| min\_salary | NUMBER | 6 | NOT NULL | 等级最低工资 |
| max\_salary | NUMBER | 6 | NOT NULL | 等级最高工资 |



（9）USERS表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| user\_id | NUMBER | 2 | PRIMARY KEY | 管理员编号 |
| user\_name | CHAR | 20 |  | 管理员名 |
| password | VARCHAR2 | 20 | NOT NULL | 密码 |

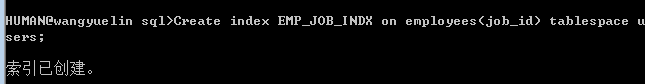


创建如下索引：

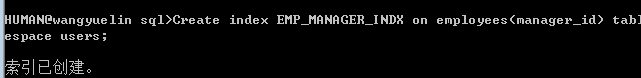
（1）在EMPLOYEES表的DEPARTMENT\_ID列上创建名为“EMP\_DEPARTMENT\_INDX”的平衡树索引。



（2）在EMPLOYEES表的JOB\_ID列上创建名为“EMP\_JOB\_INDX”的平衡树索引。

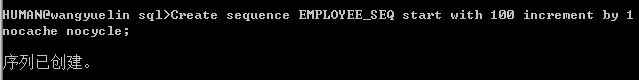


（3）在EMPLOYEES表的MANAGER\_ID列上创建名为“EMP\_MANAGER\_INDX”



创建如下序列：

（1）创建一个名为“EMPLOYEES\_SEQ”的序列，用于产生员工编号，起始值为100，步长为1，不缓存，不循环。



（2）创建一个名为“DEPARTMENT\_SET”的序列，用于产生部门编号，起始值为10，步长为10，最大值为9990，不缓存，不循环。

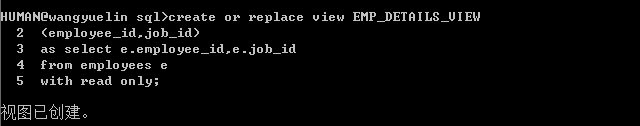


（3）创建一个名为“LOCATIONS\_SEQ”的序列，用于产生位置编号，起始值为1000，步长为100，最大值为9990，不缓存，不循环。

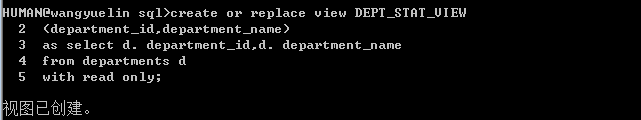


创建如下视图：

（1）创建一个名为“EMP\_DETAILS\_VIEW”的视图，用于嗣信息综合查询，包括员工编号、员工名、工资、资金、职位编号、职位名称、部门编号、部门名称、部门所在地信息、国家信息、区域信息等。



（2）创建一个名为“DEPT\_STAT\_VIEW”的视图，包含部门号、部门人数、部门平均工资、部门最高工资、部门最低工资以及部门工资总和。

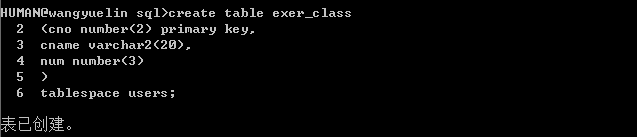


2.按下列表结构利用SQL语句完成如下工作

（1）创建exer\_class,exer\_student两个表。

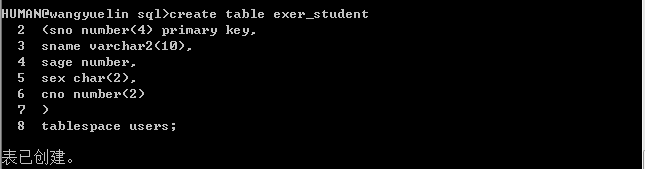
exer\_class

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 备注 |
| CNO | NUMBER(2) | 主键 | 班号 |
| CNAME | VARCHAR2(20) |  | 班名 |
| NUM | NUMBER(3) |  | 人数 |

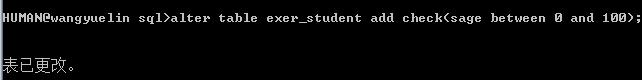


exer\_student

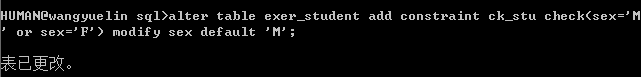
|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 备注 |
| SNO | NUMBER(4) | 主键 | 学号 |
| SNAME | VARCHAR2(10) | 唯一 | 班名 |
| SAGE | NUMBER |  | 年龄 |
| SEX | CHAR(2) |  | 性别 |
| CNO | NUMBER(2) |  | 班级号 |



（2）为exer\_student表的SAGE列添加一个检查约束，保证该列取值在0~100之间。



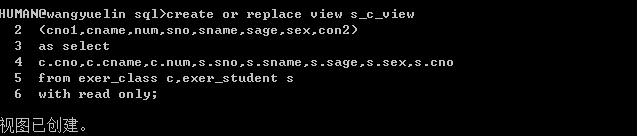
（3）为exer\_student表的SEX列添加一个检查约束，保证该列取值为“M”或“F”，且默认值为“M”。



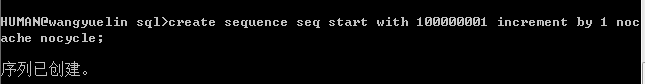
（4）在exer\_class表的CNAME列上创建一个唯一性索引。



（5）创建一个视图，包含学生及其班级信息。

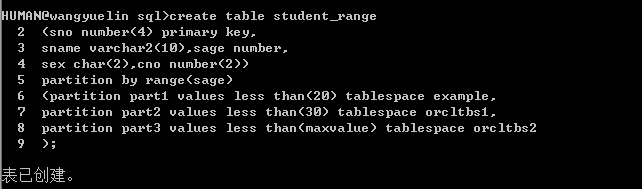


（6）创建一个序列，起始值为100000001，作为学生的学号。

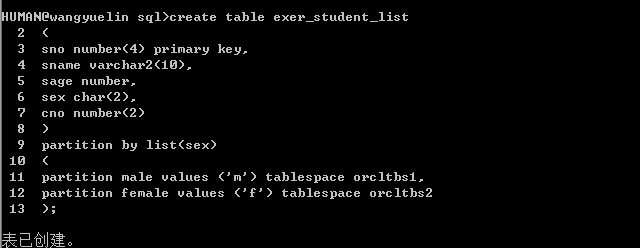


（7）创建一个exer\_student\_range表（列、类型与exer\_student表的列、类型相同），按学生年龄分为3个区，低于20岁的学生信息放入part1区，存储在EXAMPLE表空间中；20~30岁的学生信息放在part2区，存放在ORCLTBS1表空间中；其他数据放在part3区，存储在ORCLTBS2表空间中。





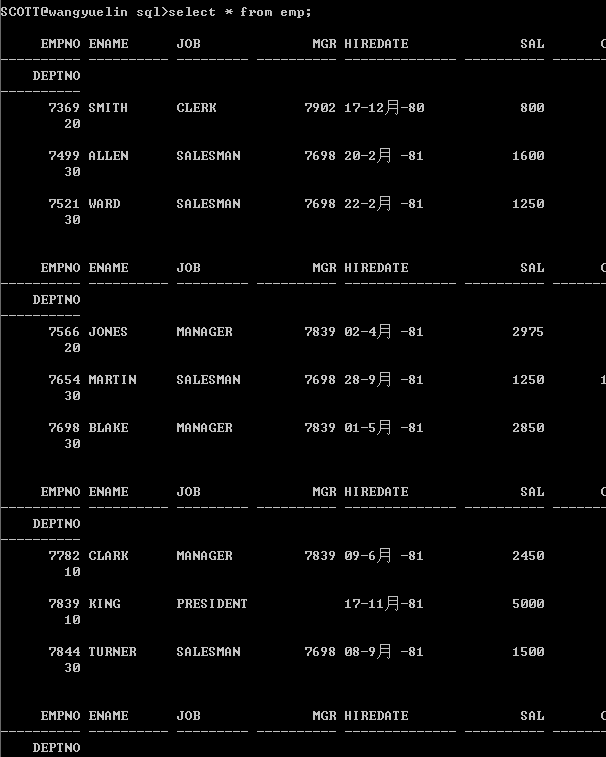
（8）创建一个exer\_student\_list表（列、类型与exer\_student表的列、类型相同），按学生性别分为两个区。

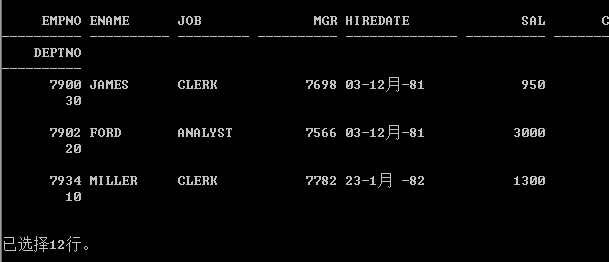


利用scott用户在scott模式下完成如下sql操作：

（1）描述scott.emp表的结构，查询emp表中所有数据

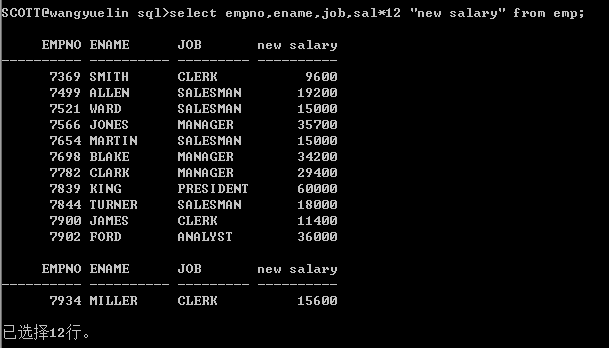




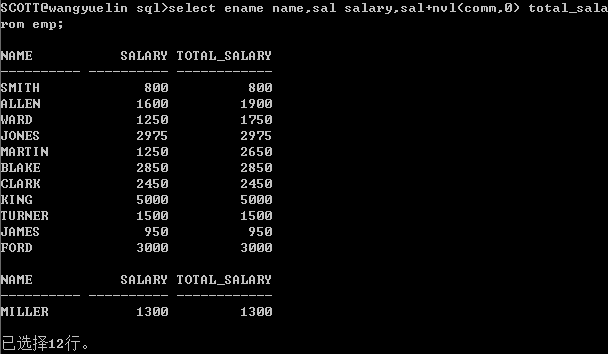


（2）找出下面语句中的错误并修正。

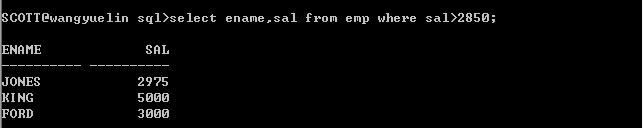
select empno,ename,job,sal×12 New Salary from emp;



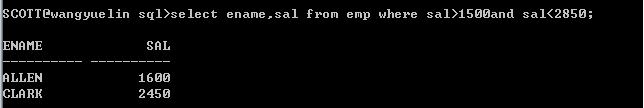
（3）查询scott.emp表中，所有员工的姓名、工资、工资与奖金之和，列头显示为如下格式：NAME SALARY Total\_Salary.



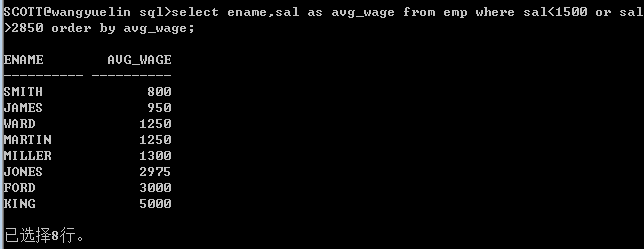
（4）查询员工的姓名和工资，要求工资大于$2850.



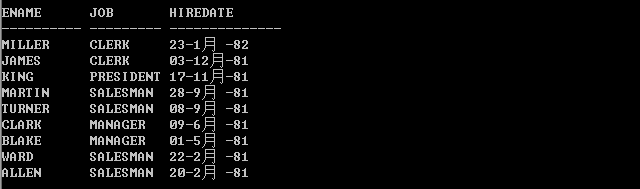
（5）查询员工的姓名和工资，要求工资在$1500和$2850之间.



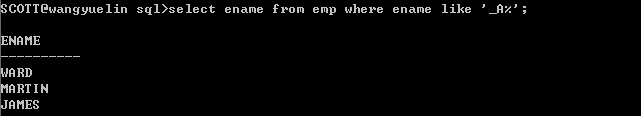
（6）查询员工的姓名和工资，要求工资不在$1500和$2850之间，按工资升序排序.



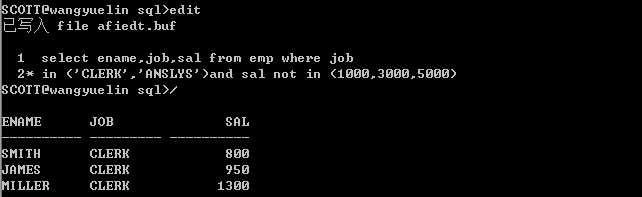
（7）查询出部门号为10和30的所有员工的姓名、工作、入职时间，按照入职时间的降序排序。



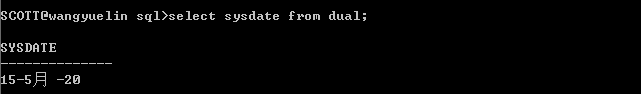
（8）查询出姓名中第二个字母为：’A’的所有员工姓名。



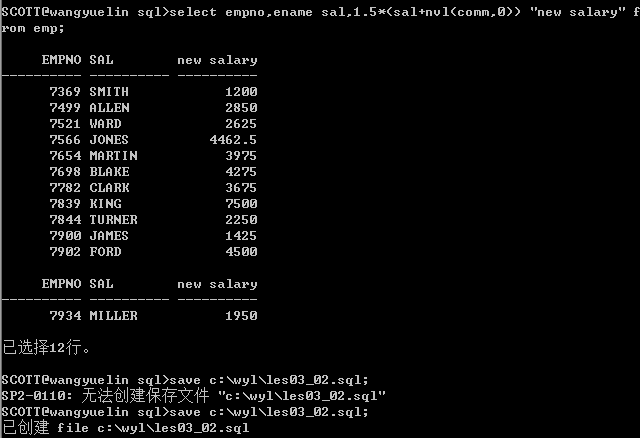
（9）查询员工的姓名、工作、工资，要求工作是Clerk或Analyst,并且工资不等于$1000、$3000、$5000.



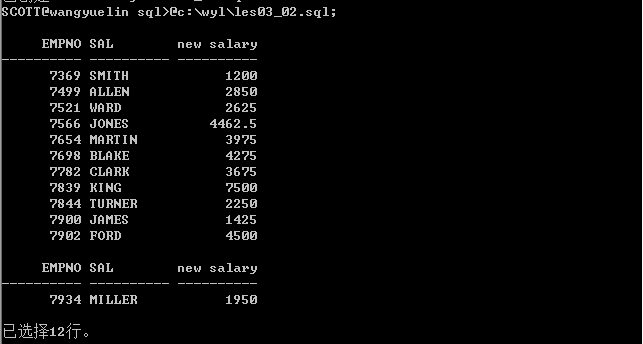
（10）写一个查询，选择出当前的日期。



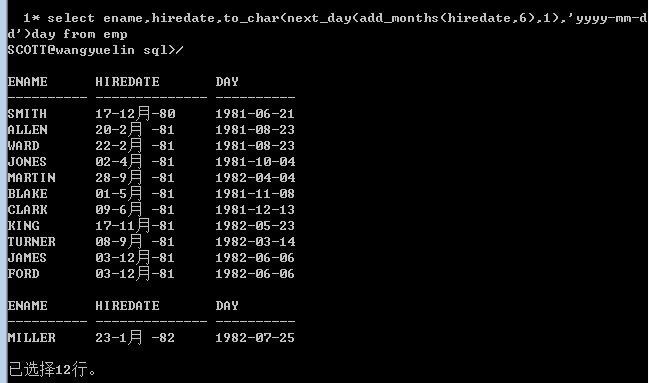
（11）查询出雇员表（emp）表中的员工编号、姓名、工资、新的薪水（是原先薪水的1.5倍），并将该列标记为New Salary。将该脚本保存到ｃ:/les03\_02.sql中。



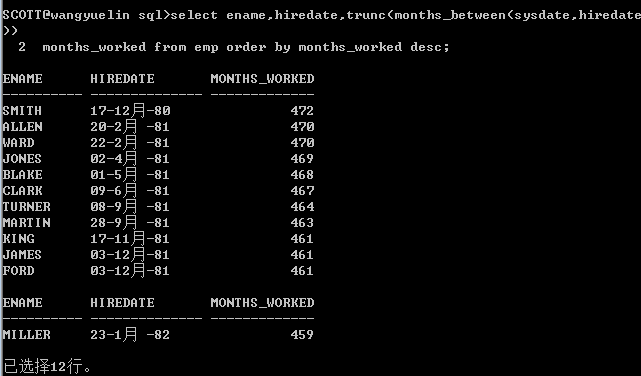
（12）运行les03\_02.sql中的查询。



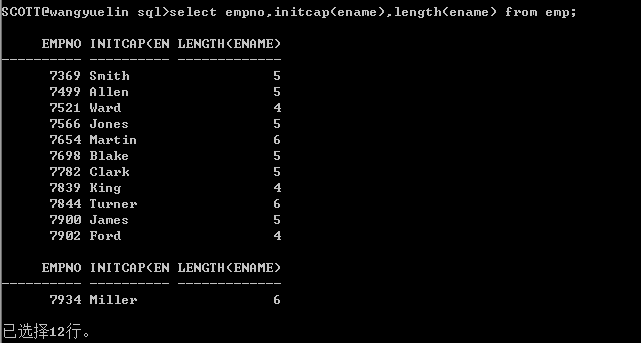
（13）查询出雇员表中所有员工的姓名、入职时间和工资调整日期（工作6个月之后的第一个星期一）。其格式如下：“1981-09-23”。



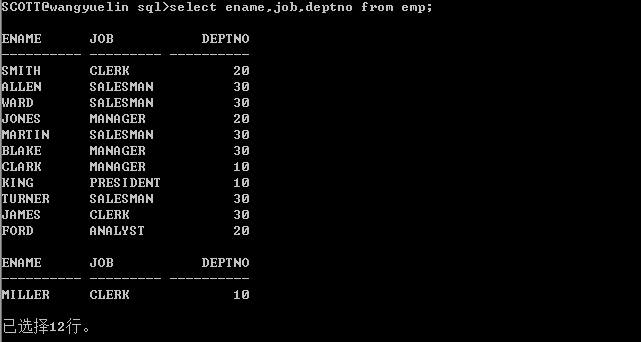
（14）查出每个员工的名字（ename）、雇佣日期到现在的日期间隔的月数（要求四舍五入为整数），该列命名为MONTHS\_WORKED，并以该列的降序排序。



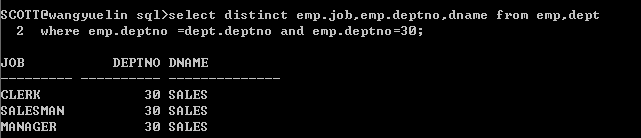
（15）查询员工的姓名，以及姓名的长度。要求其姓名首字母大写。



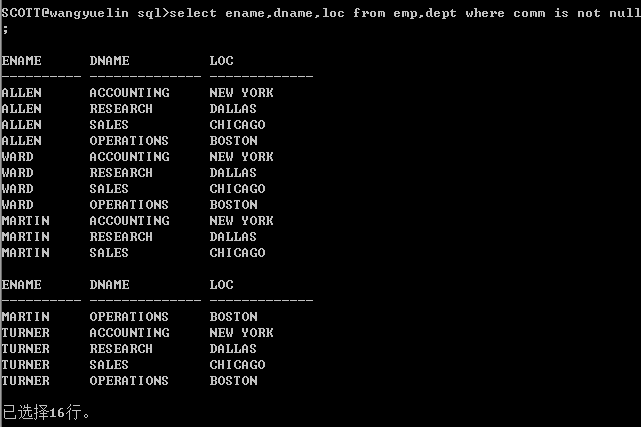
（16）查询员工的姓名，部门号，部门名称。



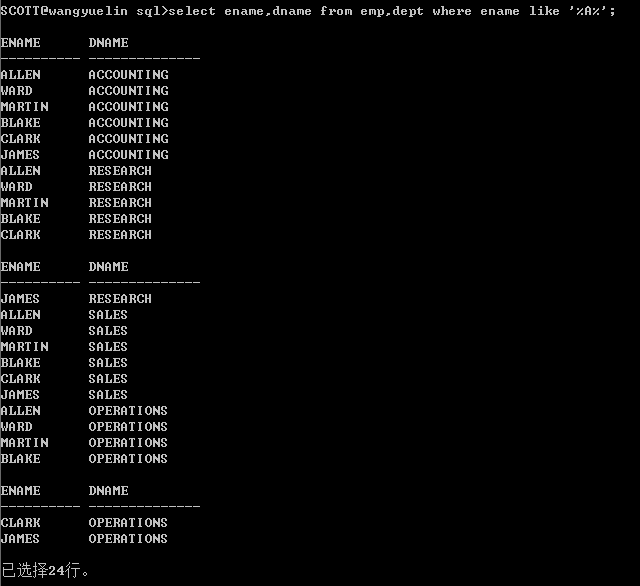
（17）查询30号部门的所有工作的列表，以及部门号



（18）查询出奖金（comm）非空的员工姓名、部门名称以及部门所处的位置。



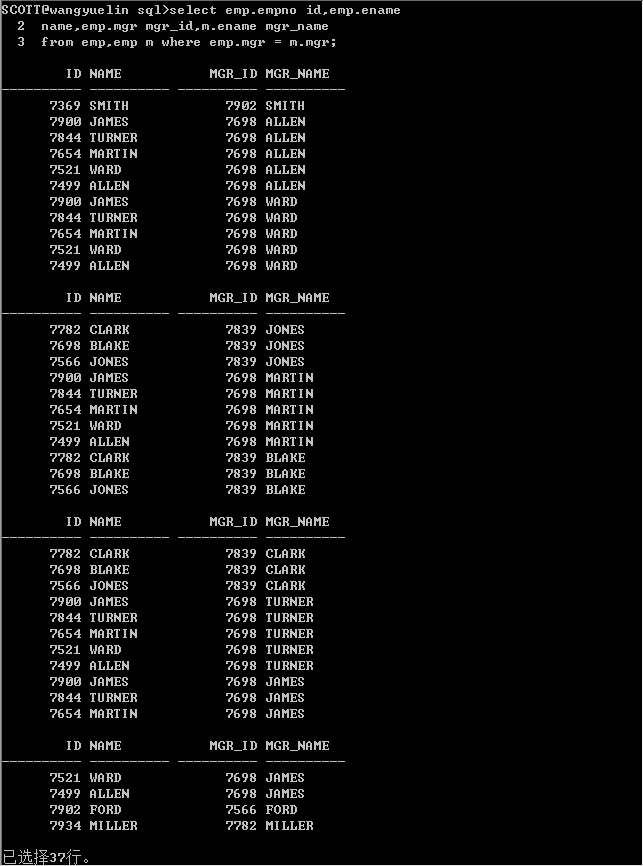
（19）查询姓名中包含“A”字母的所有员工姓名及其所在部门的名称。



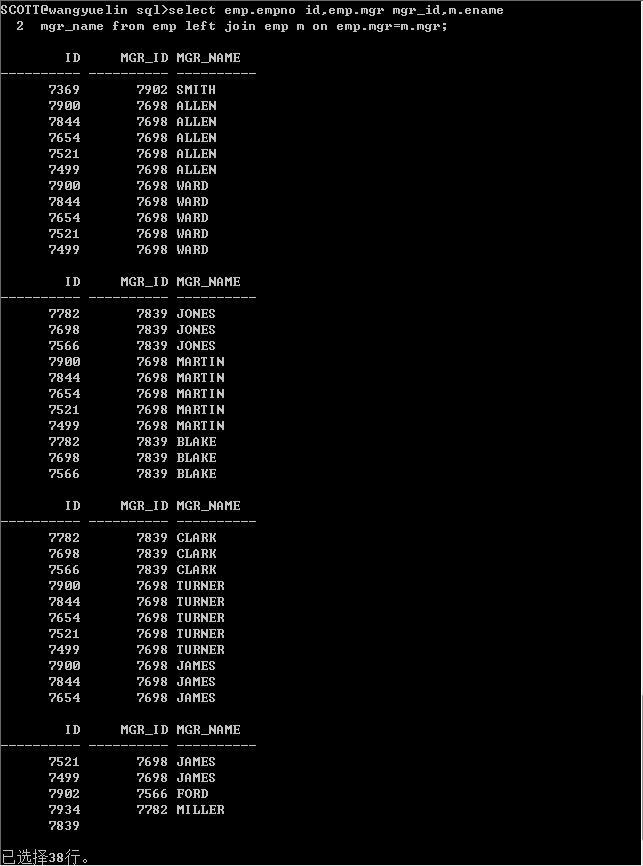
（20）查询工作在DALLS的所有员工的姓名、工作、部门号、部门名称。



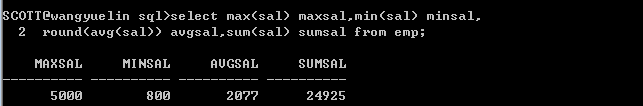
（21）查出每个雇员的编号、姓名、其经理的编号和姓名。各个列分别命名为：ID,NAME,MGR\_ID,MGR\_NAME。



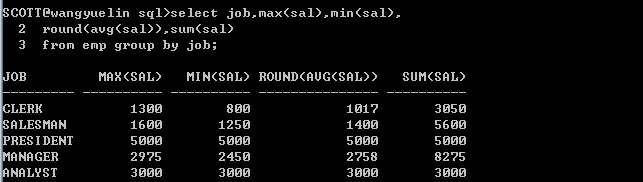
（22）修改第6题的查询，使得没有经理的员工的信息也可以显示出来。



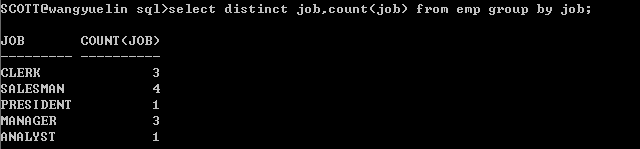
（23）查出所有员工的最高、最低、平均的工资，以及工资的总和，列名分别是maxsal,minsal,avgsal,sumsal。这些数值都用四舍五入进行处理。



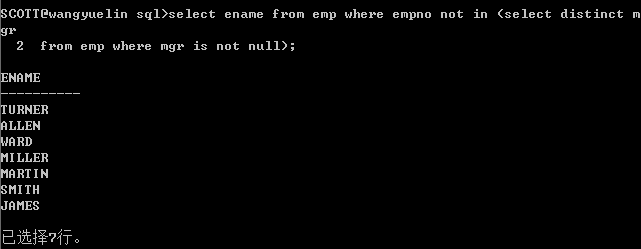
（24）按照工作类型分类，查出每种工作的名称和基员工的最高、最低、平均的工资、工资的总和，这些数值都用四舍五入进行处理。



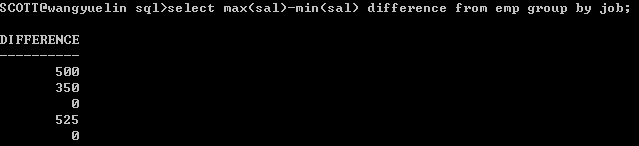
（25）查出每种工作的名称和从事该工作的人数。



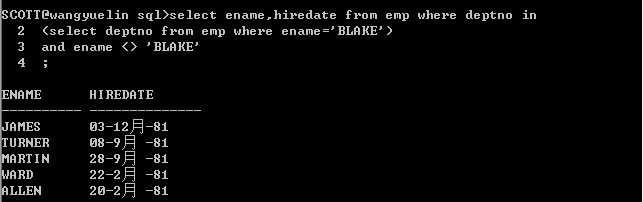
（26）查出不当领导的职工姓名。



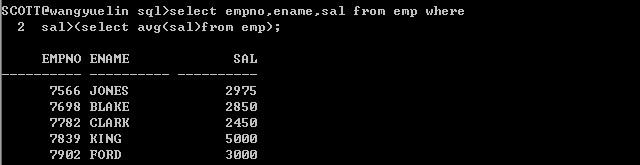
（27）按照工作分类，查工作及最高最低工资的差额，该列命名为difference。



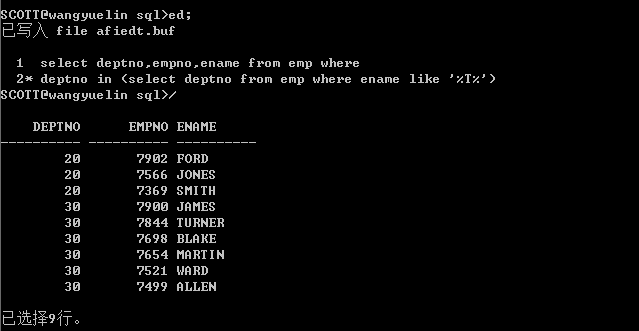
（28）查询BLAKE所在部门的所有员工的姓名、入职时间。



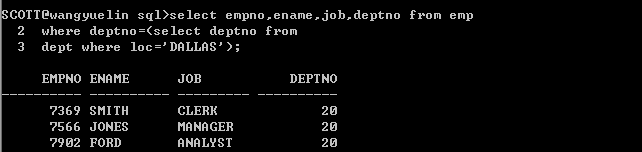
（29）查询工资高于平均工资的所有员工的编号和姓名，其结果按照工资的降序排列。



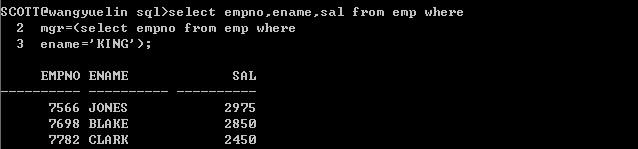
（30）查询姓名中包含“T”字母的员工所在的部门号，以及这些部门所包含的所有员工的编号和姓名。



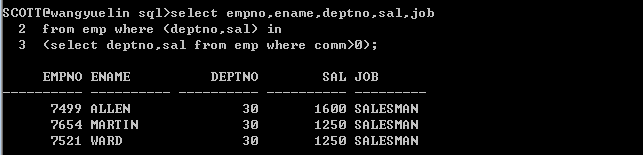
（31）查出所有工作在Dallas的员工的姓名、部门编号、工作。



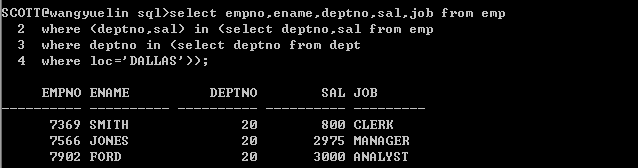
（32）查出由King直接管辖的所有员工的姓名和工资。



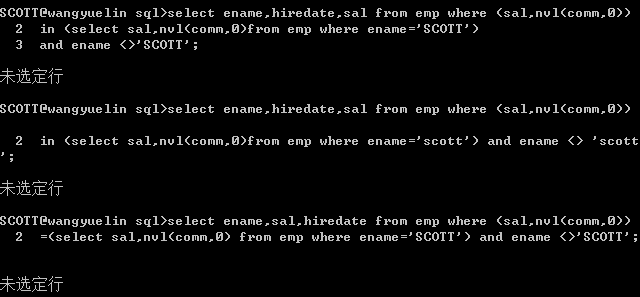
（33）写出与任何一个奖金非空的员工的部门号、薪水相匹配的员工的姓名、部门号。



（34）查询出与在Dallas的任何一个员工的薪水和奖金相匹配的所有员工的姓名、部门名称和工资。



（35）查询出所有工资和佣金与Scott相同的雇员的姓名、雇佣日期和工资。注意：结果中不包括Scott本人。

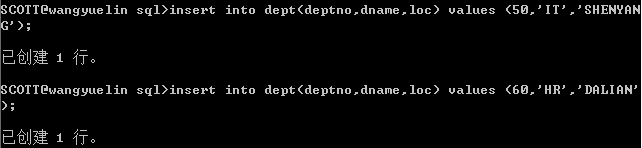


这里试了三种方法均是未选定，百度一下说是没有符合条件的数据...实在不清楚该怎么做了

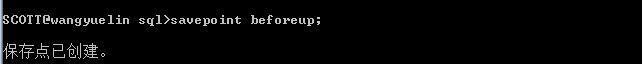
（36）在dept表中插入两行数据

50,’IT’,’SHENYANG’;

60,’HR’,’DALIAN’;



（37）设置保存点beforeup



（38）更新dept表，将60号部门位置改为BEIJING



（39）设置保存点afterup



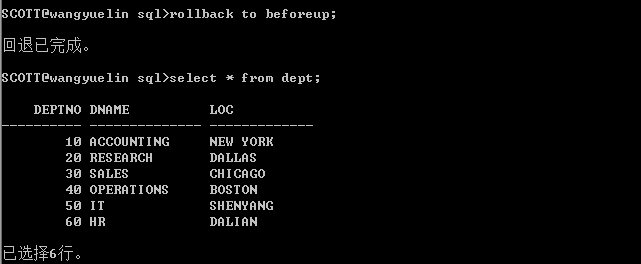
（40）使用delete删除dept表中部门号为50、60两行数据



（41）回滚到保存点afterup并查看结果



（42）回滚到保存点before并查看结果



（43）回滚整个事务

