

oracle上课资料:

第一天:

### 第一节课: 数据库基本概念学习

#### 1、数据 书

其实就是表示信息的一种特殊符号的集合

例如: 图片、txt、音频、视频等等

引入: 互联网的发展交互的其实就是数据

#### 2、数据库: 图书馆

把用来存储数据的东西称之为数据库。

#### 3、数据库管理系统 应用程序 图书馆管理系统

将数据按照指定的方式存储到数据库中的系统称之为数据库管理系统

可以高效率的取出存储好的数据 (重点)

#### 4、数据库系统

数据库管理员

数据库

数据库管理系统

数据库应用系统

用户

### 第二节课: 常见数据库及数据库存储结构

#### 1、数据库存储数据结构

网状结构

层次结构

二维表结构:

二维表结构可以清晰的将数据与数据之间的关系表述清楚

注意:

单纯的存储数据很简单, 但是存储好的数据又要将数据之间的关系描述清楚

就会比较困难, 所以采用二维表的结构存储数据会比较合理

#### 2、数据库管理系统

oracle: 大型数据库管理系统

mysql: 中小型

DB2: 中小型

Acess: 小型

Sql Server: 大型数据库

.....

#### 3、oracle的历史介绍 参照PPT

### 第三节课: oracle的安装及oracle目录结构介绍及卸载

#### 1、oracle的安装:

参照百度搜索 “oracle11g安装图解”

全局数据库名: 默认是orcl, 是用来唯一标示数据库的名字

SID: 在程序中唯一标示数据的名字

#### 2、oracle的服务: (需要开启的服务)

oracleServiceorcl: oracle启动服务

OracleOraDb11g\_home1TNSListener: oracle监听器服务

OracleDBConsoleorcl: 控制台服务

#### 3、oracle账户:

sys: 超级管理员

system: 管理员

scott: 普通用户 默认密码: tiger

#### 4、oracle修改密码:

在cmd命令行中使用命令: sqlplus / as sysdba

然后使用命令：

```
alter user 用户名 identified by 新密码;
```

在cmd命令行中使用命令：sqlplus /nolog

然后使用命令：conn as sysdba

然后输入一个已知的用户名和密码

然后使用命令：

```
alter user 用户名 identified by 新密码;
```

第四节课：oracle目录及卸载

1、oracle的目录介绍：

oradata：数据库存储文件的目录

db\_home：

network >admin:配置网络服务和监听器服务

jdk:oracle自带jdk

deinstall:卸载命令

jdbc:与java交互的jar包

2、oracle的卸载：

使用oracle自带的程序卸载

删除app目录

删除注册表

删除环境变量

第四节课：oracle的监听器及服务器配置

问题1：

我的电脑装了oralce，同桌怎么查看数据？

客户端：

针对普通用户使用的软件，主要是用来发送用户的请求。

服务器：

接受用户的请求，并处理用户的请求。

问题2：张三和李四都装了oracle，王五装了客户端，王五要访问张三，怎么访问？

ip

tcp

数据库全局名

端口号

问题3：

从服务器的角度观察，假如1000个请求过来，怎么区分那些是请求数据库的

监听器

第三方工具：plsql（不是客户端）

配置本地网络服务:(客户端)

第一种使用暴力方式直接操作：

修改：C:\app\Administrator\product\11.2.0\dbhome\_1\NETWORK\ADMIN\tnsnames.ora

ORCL = >网络服务名

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP) (HOST = 服务器ip) (PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcl)

)

)

第二种方式:

使用图形界面: NCA

配置监听器: (服务器端)

常见问题:

配置监听器, 计算机名必须是英文的

第一天的主要任务是安装数据库和配置数据库的网络服务, 第二天正式进行sql语句的学习

第二天:

第一节课: sql语言介绍(参照PPT)及基本查询sql学习

1、数据库表的介绍

emp表: 员工表

dept表: 部门表

salgrady: 薪资水平表

Balance:

2、基本的查询语句:

知识点:

select \* from 表名 查询表中所有的数据

select 字段名1, 字段名2, ..from 表名 查询指定字段的所有值, 不同字段之间使

用逗号隔开

使用算术表达式: select 字段名(运算符), 字段名(运算符)...from 表名

案例:

select \* from emp--查询所有员工信息

select ename, job, sal from emp--查询员工姓名, 工作, 薪资

查询所有员工一年的工资, 以及每位员工年终3000后的所有薪资(不包括津贴)

select ename, job, sal\*12, sal\*12+3000 from emp 使用运算符

第二节课:

1、使用别名

as可以省略不写, 如果有特殊字符获取空格使用双引号添加别名

select ename 姓名, job 工作, sal\*12 年资, sal\*12+3000 年薪 from emp

直接在字

段后面添加别名

select ename "姓名", job 工作, sal\*12 ' 年资', sal\*12+3000 年薪 from emp

使用双引

号添加别名, 不能使用单引号

select ename as 姓名, job "工作", sal\*12 年资, sal\*12+3000 年薪 from emp

使用as关键字添加

别名

select ename "姓 名", job 工作, sal\*12 年资, sal\*12+3000 年薪 from emp

如果别名中有空格

或者其他特殊字符, 使用双引号

2、使用链接符:

select ename, job, sal from emp;

select ename||job, sal from emp; 使用||符号进行字符串链接

select ename||' 的工作是 '||job||' 并且月工资是 '||sal from emp; 字符串链接,

普通字符使用单引号

3、去除重复 distinct

select \* from emp;

select job from emp--发现工作是重复的, 而工作种类是需要去除重复的

问题: 查看员工工作种类

select distinct job from emp 使用distinct关键字去除重复值

问题：怎么多字段去除重复

`select distinct job,sal from emp;` 多字段去除重复，去除的是重复的一整条信息

#### 4、排序

知识点：

`select 字段名, 字段名...from 表名 order by 字段名`

`select *from 表名 order by 字段名`

讲解：

—查询员工姓名，工作，薪资

`select ename, job, sal from emp`

—查询按照薪资排序的员工姓名，工作，薪资

`select ename, job, sal from emp order by sal;` 使用order by排序

`select ename, job, sal from emp order by sal desc;`

使用desc关键从大到小排序

`select ename, job, sal from emp order by sal asc;`

使用asc关键从小到大排序，

默认是从小到大排序

`select * from emp order by sal,ename`

多字段作为排序条

件的时候，优先按照第一个字段排序，然后依次按照其他字段排序。

`select ename, job, sal 工资 from emp order by 工资`

使用别名

`select ename, job, sal from emp order by sal*12`

使用算术表达式

#### 第三节课：简单的where子句

使用where条件进行结果筛选

语法：`select * from 表名 where 条件`；注意：条件中字段值区分大小写，字段名不区分大小写，字

段值使用单引号括起来

1、在where中使用算术表达式 `= , < , > , >= , <= , <>`

2、使用order by 对筛选结果进行排序，order by 出现在where后面

查看工资等于1250的员工信息

`select * from emp where sal='1250'`—筛选条件是个数字也可以使用单引号

查看工作等于CLERK的员工信息

`select * from emp where job='CLERK'`—在筛选条件中字段值如果是字符需要加上单引号

`select * from emp where job='clerk'`—在sql语句中字段的值是区分大小写的

`select * from emp where JOB='CLERK'`—在sql中字段是不区分大小写的

查看工资大于1250的员工姓名和工作

`select ename, job ,sal from emp where sal>'1250' order by sal`—使用order by 对筛

选结果进行排序，order by 出现在where后面

查看工资大于等于2000的员工信息

`select * from emp where sal>=2000;`

查看工资小于等于2000的员工信息；

`select * from emp where sal<=2000`

查看工资不等于1500的员工信息

`select * from emp where sal<>1500 order by sal`

查看入职日期在81年后的员工信息

`select * from emp where hiredate>'1981年12月31号'`

`select * from emp where hiredate>'1981/12/31'`

`select * from emp where hiredate>'31-12月-1981'`—使用日期的默认格式查询符合要求的数据，日-月-年

#### 第四节课：where子句使用关键字

使用where子句进行结果的筛选

知识点: where查询条件中使用关键字

- 1、and 用于多条件的与筛选: select \* from 表名 where 条件 and 条件 and 条件....
- 2、or 用于多条件的或筛选: select \* from 表名 where 条件 or 条件 or 条件....
- 3、in 用于多条件的或筛选: select \* from 表名 where 字段名 in(值, 值, 值....)
- 4、like用于模糊查询: select \* from 表名 where 字段名 like '值%' 包含
- 5、is null 和is not null 用来判断字段是否为空 select \* from 表名 where 字段名

is null

讲解: 在where子句中使用关键字

--查询工资在2000-3000之间的员工信息

select \* from emp where sal>=2000 and sal<=3000--使用and关键字进行“与”的多条件筛

选;

select \* from emp where sal between 2000 and 3000;--使用between and 关键字进行筛

选;

--查询工作为SALESMAN, ANALYST, MANAGER的员工信息

select \* from emp where job='SALESMAN' or job='ANALYST' or job='MANAGER' --使用or

关键字进行“或”的多条件筛选

select \* from emp where job in('SALESMAN','ANALYST','MANAGER');--使用in关键字进

行“或”的多条件筛选

select \* from emp where job='ANALYST'

--查询姓名中包含s的, 以s开头的, 以s结尾的, 第二个字符为A的。

select \* from emp where ename like '%S%';--使用like关键字, 姓名中包含S的,%代表任

意多个字符

select \* from emp where ename like 'S%';--使用like关键字, 以S开头的

select \* from emp where ename like '%S';--以S结尾的

select \* from emp where ename like '\_A%' --使用“\_”指定位置包含指定字符的信

息, “\_”代表任意一个字符

-----查询名字中包含下划线的用户信息

select \* from emp where ename like '%A\_%' escape 'A';--使用escape关键字将普通字符

设置成为转译字符。

--查询有津贴的员工信息

select \* from emp where comm is not null;

select \* from emp where comm is null;--查询没有津贴的员工信息

## 第五节课: 函数学习

--查询工作为SALESMAN, MANAGER并且工资大于2500的员工信息

--1、使用小括号提升where筛选条件的执行优先级别

--2、and的优先级别高于or

select \* from emp

select \* from emp where job='SALESMAN' or job='MANAGER' and sal>2500

select \* from emp where (job='SALESMAN' or job='MANAGER') and sal>2500

使用函数 单行函数 多行函数 转换函数 其他函数

--单行函数学习: 不改变原始数据, 只改变结果

--1、字符函数

select \* from emp;

select INITCAP(ename) from emp;--initcap函数将首字母大写

select lower(ename) from emp;--lower 字母小写

select replace(ename,'S','M') from emp;--replace 替换

```
--2、数值函数--Math
---伪表 dual
select * from dual
select abs(-3) 绝对值, ceil(3.1415926)向上取整, floor(3.1415926)向下取整, power(2, 3)
幂, round(3.4)四舍五入 from dual

--3、日期函数
select months_between('13-12月-2016', '13-10月-2016') from dual--months_between两个日期之
间的月份数
```

#### 第六节课：多行函数学习

多行函数：

max: max(字段名) 返回此字段的最大值

min: min(字段名) 返回此字段的最小值

avg: avg(字段名) 返回平均值

sum: sum(字段名) 返回字段的和

count: count

--count(\*), 用来查询表中有多少条记录

--count(字段名), 用来查询某个字段有值的个数

--count(distinct 字段名), 可以先去除重复再计数。

注意:

--多行函数不能和普通字段直接出现在查询语句中, 除非group by

--多行函数和单行函数不能直接出现在查询语句中, 除非group by

查看员工的最高工资

```
select max(sal), ename from emp--多行函数不能和普通字段直接出现在查询语句中, 除非
```

group by

```
select max(sal), lower(ename) from emp--多行函数和单行函数不能直接出现在查询语句
```

中, 除非group by

```
select ename from emp
```

查看员工的最低工资

```
select min(sal) from emp
```

查看员工的平均工资

```
select avg(sal) from emp
```

查看所有的员工工资之和

```
select sum(sal) from emp
```

查询公司有多少员工

```
select * from emp
```

```
select count(*) from emp--使用count(*)来查看一张表中有多少条记录
```

查询有津贴的员工人数

```
select count(comm) from emp--使用count(字段名), 查询该字段有值的记录数
```

```
select count(ename) from emp
```

查询公司有多少工作种类

```
select count(job) from emp
```

```
select count(distinct job) from emp
```

=====

第三天:

#### 第一节课：转换函数学习

转换函数: 在转换的时候改变的是数据的类型, 数据内容不会改变, 可以指定格式。

1、to\_number: 将数字字符转换为数字类型的数值, to\_number(数字字符)

2、to\_char: 将数字转换为字符类型, 将日期转换为字符类型 to\_char(数字/日期)

3、to\_date: 将字符类型的日期转换为日期类型: to\_date(char)

number-->char 转换的时候使用的是默认格式,

```

select to_char(123) from dual--将数字转换为字符类型，使用to_char(number)
select sal, to_char(sal) from emp
number-->char 使用指定的格式
to_char(number,'格式'), 格式:

```

\$代表美元符号，9代码数字占位。例如，L999,999,999表示三位一组的显示方式L代表人民币符号，0代表保留数字个数，不足使用0补充。

```

select to_char(sal),to_char(sal,'L999,999,999') from emp
select to_char(sal),to_char(sal,'L0000.00') from emp
char-->number to_number(数字字符)
select to_number('123') from dual

```

char-->date, 转换的字符必须是日期格式的字符串，默认格式 dd-mm-yyyy

注意：因为字符串有很多，所以在字符转换为日期的时候，需要指定格式，因为日期是具备一定格式的字符组合。

字符转换为日期的时候，指定的格式为字符的日期顺序，无需指定间隔符。

```

yyyy: 表示年 mm:表示月 dd表示日
select to_date('05-12月-2016') from dual;
select to_date('2016-05-12','yyyy-mm-dd') from dual--使用指定的格式

```

将指定的日期字符串转换为日期

```

select to_date('12-05-2016','mm/dd/yyyy') from dual
date-->char

```

注意：因为日期本身就具备一定的格式在不是指定格式的情况下会默认使用dd-mm-yyyy格式显示数据

指定的格式会作为日期转换为字符串类型的显示格式存在。例如：

```

yyyy-mm-dd'      '2016-12-05'
yyyy/mm/dd'      '2016/12/05'
yyyy"年"mm"月"dd"日"'      2016年12月05日

```

select hiredate,to\_char(hiredate) from emp--日期转换为字符的时候，不指定格式使用默认格式：dd-mm-yyyy

select hiredate,to\_char(hiredate,'yyyy-mm-dd') from emp;--使用指定格式将日期转换为字符串类型

select hiredate,to\_char(hiredate,'yyyy/mm/dd') from emp;--使用指定格式将日期转换为字符串类型

select hiredate,to\_char(hiredate,'yyyy"年"mm"月"dd"日"') from emp;--使用指定格式将日期转换为字符串类型

查询入职日期在81年10月20日后的员工信息

第一种：自动转型

```
select * from emp where hiredate>'20-10月-1981'
```

第二种：将日期转换为字符串

```
select * from emp where to_char(hiredate,'yyyy-mm-dd')>'1981-10-
```

20'

第三种：

```
select * from emp where hiredate>to_date('1981-10-
```

20','yyyy/mm/dd')

第二节课：其他函数：

单行函数：lower

多行函数:min max sum avg count

转换函数:to\_number to\_char to\_date

其他函数：

- 1、nvl():nvl(字段名, 执行)——相当于java中的if条件判断
- 2、nvl2():nvl2(字段名, 值, 值)——相当于java中的If(){}else{}判断
- 3、decode():decode(字段名, 条件1, 执行内容1, 条件2, 执行内容2, 条件3, 执行内容3,

默认执行内容)

相当于java中if(){}else if(){}else if(){}...else{}

查询所有员工的月薪及姓名和工作

```
select * from emp
select ename, sal 基本工资, comm 绩效, sal+comm 月薪 from emp
select ename, sal 基本工资, comm 绩效, sal+nvl(comm, 0)月薪 from emp
```

查询所有员工的月薪及姓名和工作

```
select ename, sal 基本工资, comm 绩效, nvl2(comm, sal+comm, sal)月薪 from emp
```

显示员工的职称

```
select ename, job, decode(job, 'MANAGER', '经理', 'SALESMAN', '销售人员', '普通员工')
```

from emp

### 第三节：使用group by分组

在多行函数中不能直接使用普通字段，除非group by

在多行函数中不能直接使用单行函数，除非group by

group by学习：

——1、使用group by进行数据分组 select 多行函数, 分组字段 from 表名 group

by 分组字段

——2、多字段进行分组的时候，按照字段顺序进行分组，第一条分组完成后，继

续使用其他条件依次分组。

——3、group by依然可以和order by 联合使用

——4、可以和单行函数联合进行分组，注意使用了单行函数那么在查询语句中必须

也要使用

查询最高工资和员工数

```
select max(sal), count(*) from emp
```

查询不同部门的最高工资

```
select * from emp order by deptno
select deptno, max(sal) from emp group by deptno——使用group进行分组查询，分组的字
```

段可以出现在查询中，其他字段依然不可以

查询不同工作岗位的员工数

```
select * from emp for update
select lower(job), count(*) from emp group by lower(job)——使用单行函数进行分组
```

查询不同部门的不同工作岗位的人数

```
select deptno, job , count(*) from emp group by deptno, job——使用多字段组合进行分组
select deptno, job , count(*) from emp group by deptno, job order by deptno
```

查询不同部门的不同工作岗位的并且人数大于1的信息

```
select count(*) from emp where count(*)>3 group by deptno
select deptno, job , count(*) from emp where count(*)>1 group by deptno, job order
```

by deptno

查询部门号大于10的不同部门的不同工作岗位的人数

```
select deptno, job , count(*) from emp where deptno>10 group by deptno, job order
```

by deptno

使用having进行分组后筛选

having学习：

——1、使用group by分组后在进行数据筛选的时候，where中不能出现多行函数，所以使用

新的关键字having进行条件筛选

——2、where条件筛选的执行顺序：from——>where——>group ——>select



--3、having条件筛选的执行顺序: from-->group by -->having-->select

--4、where的执行效率比having要高, 能使用where的情况下尽量不要使用having

查询不同部门的不同工作岗位的并且人数大于1的信息

使用where语句进行筛选

where条件语句sql执行顺序: from-->where-->group -->select

```
select count(*) from emp where count(*)>1 group by deptno, job
```

使用having语句进行筛选

having条件语句的执行顺序: from-->group by -->having-->select

```
select deptno, count(*) from emp group by deptno having count(*)>5
```

```
select deptno, job ,count(*) from emp group by deptno, job having deptno>10
```

```
order by deptno
```

#### 第四节课: 插入数据学习及数据的备份

单表查询语句 (select)

1、插入数据(insert)

1、语法规范 insert into 表名(字段1, 字段2, 字段3, ....)values('值1','值2','值3'.....)

2、主键: 用来唯一标识一条数据的字段通常设置主键, 主键是唯一不可以重复的

3、如果插入的数据是全字段数据, 字段可以省略不写。部分字段, 必须加上字段说明和字段值, 但是主键不能为空

4、事务的提交: 如果一个事件是由多个动作组成, 只要有一个动作没有执行成功则自动将数据回滚到原始状态, 此技术称之为事务

保证数据的安全和完整

事物的提交:

使用第三放插件的提交按钮

使用commit语句

```
select * from dept for update
```

在北京新建了一个名为LOL学院的新部门, 请插入

```
insert into dept(deptno,dname,loc)values('50','lol学院','北京');
```

主键是唯一不可以重复的

```
insert into dept(deptno,dname,loc)values('50','教学部','北京');
```

如果插入的数据是全字段数据, 字段可以省略不写。部分字段, 必须加上字段说明和字段值, 但是主

键不能为空

```
insert into dept values('80','教学部','北京');
```

```
insert into dept values('90','教学部','北京');
```

```
insert into dept values(100,'教学部','北京');
```

```
insert into dept values('110','教学部','北京');
```

2、创建数据的备份

(1) create table 表名 as 查询语句, 创建的是和查询结果一样的表, 查询结果是什么就会备份一

个相同的表

(2) insert into 表名 查询语句, 注意: 查询出来的结果在结构上必须和插入数据的表相同, 字段

个数必须相同

(3) 注意: 备份表只有字段和数据相同, 并不会备份约束。

1、备份完整的数据和表

```
select * from dept;
```

```
create table tdept as select * from dept;--备份dept表和数据, 只能备份数据
```

和字段

```
select * from tdept
```

2、备份完整表

```
create table tdept1 as select * from dept where 1>2--备份表, 不备份数据。
```

3、备份部分数据和表.create table 表名 as 查询语句, 创建的是和查询结果一样的表, 查询结果是什么就会备份一个相同的表

```
create table tdept2 as select dname,loc from dept
select *from tdept2
```

4、给备份表添加数据 insert into 表名 查询语句, 注意: 查询出来的结果在结构上必须和插入数据的表相同, 字段个数必须相同

```
select * from tdept1
insert into tdept1 select dname,loc from dept where deptno>40
select *from tdept2
```

```
insert into tdept2 select dname,loc from dept where deptno>40
```

第五节课: 数据的更新和删除:

更新数据:update

语法结构: update 表名 set 字段名1='字段值1', 字段名2='字段值2', ...where 条件

将部门70的名字改为教学部2

```
update dept set dname='教学部2' where deptno='70'
```

```
update dept set dname='教学部2', loc='上海' where deptno='70'
```

```
select * from dept
```

删除数据: delete

1)删除语句: delete 表名 where 条件

—删除部门标号为70的数据

```
delete dept where deptno='70'——删除指定数据
```

```
select * from dept
```

delete tdept——清空表数据

truncate table tdept——清空表数据建议使用truncate关键字, 但是此关键字不能回滚数据

第四天:

第一节: sql92学习

—查询员工姓名, 工作, 薪资, 部门名称

sql的联合查询(多表查询)

—1、sql92标准

——笛卡尔积: 一件事情的完成需要很多步骤, 而不同的步骤有很多种方式, 完成这件事情的所有方

式称之为笛卡尔积

```
select * from emp——14
```

```
select * from dept——7
```

```
select * from emp,dept order by ename——7*14=98
```

等值链接, 链接条件. 等值链接的时候字段的名称可以不相同, 但是字段的值要相同。

—查询员工姓名, 工作, 薪资, 部门名称

```
select * from emp,dept where emp.deptno=dept.deptno——使用等值链接进行结果筛选
```

```
select ename, job, sal, dname from emp,dept where emp.deptno=dept.deptno;——使用等值
```

链接查询指定字段数据

```
select ename, job, sal, emp.deptno, dname from emp,dept where
```

emp.deptno=dept.deptno;——多表查询的时候, 查看相同字段的值, 必须声明所在表

```
select emp. ename, emp. job, emp. sal, emp. deptno, dept. dname from emp,dept where
```

emp.deptno=dept.deptno;——在查询指定字段值的时候, 加上表名提高查询效率

```
select e. ename, e. job, e. sal, e. deptno, d. dname from emp e,dept d where
```

e. deptno=d. deptno;——多表查询中使用别名进行表的区分

```
select e. ename, e. job, e. sal, e. deptno, d. dname from emp e,dept d where
```

e. deptno=d. deptno order by deptno;——还可以使用order by 排序

非等值链接

--查询员工姓名, 工作, 工资, 工资等级

```
select * from salgrade
```

```
select * from emp, salgrade order by ename
```

```
select * from emp, salgrade where sal > losal and sal < hisal
```

```
select e.ename, e.job, e.sal, s.grade from emp e, salgrade s where e.sal > s.losal and
```

```
e.sal < s.hisal;
```

自连接: 使用频率不是很高, 因为自连接的条件要求不同信息共存在一张里, 其实就两张相同的表的等值链接。

--查询员工姓名, 工作, 薪资, 及上级领导姓名

```
select * from emp for update
```

```
create table temp as select * from emp
```

```
select * from emp e, temp t where e.mgr=t.empno
```

## 第二节课: sql99

sql99学习:

交叉链接(没啥用)

```
select * from emp cross join dept
```

自然链接:natural join

(1)、自然链接会自动使用多表中所有相同字段(不但值相同, 名字也要相同)进行筛选

前提: 多表一定要有同名同值的字段

注意: 自然连接会自动使用所有的相同字段进行结果筛选

选

(2)、使用using关键字可以指定字段进行链接查询, 但是必须是同名字

段 inner join

(3)、使用on关键字可以直接在其后书写链接条件, 没有限制

inner join

查询员工及员工所在的部门信息

```
select * from dept natural join emp
```

```
select * from dept d, emp e where d.deptno=e.deptno
```

问题1: 假如在链接查询的时候不想使用所有相同字段进行筛选怎么办?

----使用using关键字

---查询员工及员工所在的部门信息

```
select * from dept join emp using(deptno)
```

问题2: 假如在链接查询中没有同名字段, 但是有同值字段怎么筛选?

使用on关键字

查询员工及员工所在的部门信息

```
select * from dept inner join emp on
```

```
dept.deptno=emp.deptno
```

查询部门及部门的城市信息

```
select * from dept inner join city on dept.loc=city.cid
```

外链接

(1)、左外链接 left outer join

(2)、右外链接 right outer join

(3)、全外链接 full outer join

左外链接

查询没有部门的员工信息及员工和部门信息

```
select * from emp e, dept d where d.deptno(+) = e.deptno--sql92
```

```
select * ---sql99
```

```
from emp e
```

```
left join dept
```

```

using(deptno)
--右外链接
select * from emp e,dept d where d.deptno=e.deptno(+)--sql92

select * ---sql99
from emp e
right outer join dept d
on e.deptno=d.deptno
--全外链接
select *
from emp e
full outer join dept d
on e.deptno=d.deptno

```

### 第三节课：自连接及三表联合查询

查询员工姓名，职位，薪资，上级领导姓名

```

select e.ename,e.job,e.sal,m.ename from emp e,emp m where e.mgr=m.empno--sql92
sql99实现:

```

```

select e.ename,e.job,e.sal,m.ename
from emp e
inner join emp m
on e.mgr=m.empno

```

---

sql99三表联合查询

查询员工姓名，工作，薪资，部门名称，城市名称

sql92实现:

```

select e.ename,e.job,e.sal,d.dname,c.cname
from emp e,dept d,city c
where e.deptno=d.deptno and d.loc=c.cid

```

sql99实现:

```

select e.ename,e.job,e.sal,d.dname,c.cname
from emp e
inner join dept d
on e.deptno=d.deptno
inner join city c
on d.loc=c.cid

```

查询工资高于2000的员工姓名，工作，薪资，部门名称，城市名称

sql92实现:

```

select e.ename,e.job,e.sal,d.dname,c.cname
from emp e,dept d,city c
where e.deptno=d.deptno and d.loc=c.cid and sal>2000

```

sql99实现:

```

select e.ename,e.job,e.sal,d.dname,c.cname --查询内容
from emp e --查询表
inner join dept d --链接表
on e.deptno=d.deptno --链接条件
inner join city c --链接表
on d.loc=c.cid --链接条件
where e.sal>2000 --筛选条件

```

#### 第四节课：单行子查询

子查询（单行子查询、多行子查询）

单行子查询

select 内容 from 表名 where 子查询语句

1、什么时候使用单行子查询？

--在不能直接获取有效信息的时候，考虑使用子查询

2、单行子查询的注意事项？

---子查询的结果必须只有一个值

---可以直接使用算术链接符

---子查询出现在where中，一般出现在条件语句的右边

问题：查询所有比雇员“CLARK”工资高的员工信息

解读1：查询雇员CLARK的工资

select sal from emp where ename='CLARK'

解读2：查询工资大于2450的员工信息

select \* from emp where sal>'2450'

使用子查询

select \* from emp where sal>(select sal from emp where ename='CLARK')

select \* from emp where (select sal from emp where ename='CLARK')<sal

查询工资高于平均工资的员工的名字和工资

select \* from emp where sal>(select avg(sal) from emp) order by sal

查询和scott属于同一部门且工资比他低的员工资料

select \* from emp where deptno=(select deptno from emp where ename='SCOTT') and

sal<(select sal from emp where ename='SCOTT')

查询工资最高的员工资料

select \* from emp where sal=(select max(sal) from emp)

查询职务和scott相同，雇佣时间早的员工信息

select \* from emp where job=(select job from emp where ename='SCOTT') and

hiredate<(select hiredate from emp where ename='SCOTT')

查询工资比scott高或者雇佣时间早的员工编号和名字

select empno,ename from emp where sal>(select sal from emp where ename='SCOTT') or

hiredate<(select hiredate from emp where ename='SCOTT')

查询工资高于任意一个CLERK的所有员工信息

select \* from emp where sal>(select min(sal) from emp where job='CLERK') and job<>'CLERK'

-----

#### 第五节课：多行子查询

多行子查询学习：其实就使用使用关键字：any all in

注意：子查询返回多个值建议使用多行子查询，返回的单个值使用单行子查询

select \* from emp where sal>any (select sal from emp where job='CLERK') and job<>'CLERK'

查询工资高于所有SALESMAN的员工信息

select \* from emp where sal>(select max(sal) from emp where job='SALESMAN')--单行子查询的

写法

select \* from emp where sal>all (select sal from emp where job='SALESMAN');

查询部门20中同部门10的雇员工作一样的雇员信息

select \* from emp where deptno='20' and (job='MANAGER' or job='PRESIDENT' or job='CLERK')

select \* from emp where deptno='20' and job in ('MANAGER', 'PRESIDENT', 'CLERK')

select \* from emp where deptno='20' and job in (select job from emp where deptno='10')

select \* from emp where deptno='20' and job= any (select job from emp where deptno='10')

select job from emp where deptno='10'

## 第六节课：用户创建

### 创建用户

当前用户介绍：权限级别的分配

-----system: 系统账户

-----sys: 超级管理员

-----scott: 普通用户

1、 创建自定义用户: create user 用户名 identified by 密码;

普通用户不具备创建用户的权限，需要使用system账户进行创建

直接创建好的用户不能登录，还需要使用system进行权限的分配（角色：

一系列权限的集合）

```
create user zyp identified by 123456;
```

给创建的用户赋予角色。 grant 角色名, 角色名...to 用户名;

```
grant connect to zyp;--赋予链接库权限
```

```
grant resource to zyp;--赋予操作资源权限
```

```
grant dba to zyp;--赋予DBA角色
```

```
grant dba to zyp
```

删除用户权限

```
revoke dba from zyp
```

```
revoke connect from zyp
```

2、 删除用户

```
drop user zyp
```

=====

第五天:

## 第一节课：创建表和字段讲解

### 一创建表学习

1、 创建表的基本语句: create table 表名 ( 字段名 类型, 字段名 类型, 字段名 类

型,...)

创建学生表: 学号, 姓名, 性别, 年龄, qq号, 邮箱。

```
create table student(
```

```
    snum number(10),--指定学号的数字长度为10
```

```
    sname varchar2(100),--指定姓名的存储上限为100个字节
```

```
    sex char(4),
```

```
    age number,
```

```
    qq number,
```

```
    sal number(6,2),--指定工资的整数位为6和小数位为2
```

```
    mail varchar2(50)
```

```
)
```

2、 字段类型

number类型: 数字类型, 用来声明存储数字的字段。 number(指定数字的整数长度, 指定数字的小数位长度)

varchar2类型: 字符类型, 用来声明存储字符的字段。会根据存储的数据自动增长大小, varchar2(存储上限)

char类型: 字符类型, 用来声明存储字符的字符。会开辟指定大小的内存来存储数据。

区别1: char的存储效率高于varchar2

区别2: char是开辟指定大小的内存空间, varchar2是根据数据大小来开辟空间的大小

date类型: 存储日期类型

## 第二节课：修改表学习:

### 修改表

在命令窗口查看表结构 desc 表名

添加新字段 alter table 表名 add 字段名 类型

alter table student add phone number(11)

修改字段类型 alter table 表名 modify 字段名 类型

alter table student modify sname varchar2(200)

删除字段 alter table 表名 drop column 字段名

alter table student drop column phone

修改表名 rename 表名 to 新的表名

rename student to stu

rename stu to student

删除表 drop table 表名

drop table student

第三节课：约束学习(根据实际情况切成两节课)

select \* from student for update

添加数据

问题1:学号重复了，数据还可以插入成功

insert into student(snum, sna, sex, age, qq, sal, mail) values('123', '张三', '男', 18, 12345678, 88.88, '12345678@qq.com')

insert into student values(123, '李四', '男', 18, 7890122, 88.99, '7890122@qq.com')

问题2:姓名可以为空。

insert into student values(456, '柳岩', '女', 18, 666999, 99.66, '666999@qq.com')

insert into student (snum, sex, age, qq, sal, mail) values(789, '女', 18, 888444, 99.66, '888444@qq.com')

问题3: 性别不但可以为空，还可以为其他不知道的字符

insert into student (snum, sna, age, qq, sal, mail) values(108, '景甜', 18, 000999, 99.66, '000999@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values(102, '景甜', 'a', 18, 000999, 99.66, '000999@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values(102, '景甜', '女', 18, 000999, 99.66, '000999@qq.com')

问题4: 年龄可以超过200

insert into student (snum, sna, sex, age, qq, sal, mail) values(103, '唐嫣', '女', 23, 78900, 99.66, '78900@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values(103, '唐嫣', '女', 230, 78900, 99.66, '78900@qq.com')

问题5: qq号一致

insert into student (snum, sna, sex, age, qq, sal, mail) values(104, '关晓彤', '女', 19, 111000, 99.66, '111000@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values(105, '袁华', '男', 22, 111000, 99.66, '111000@qq.com')

约束学习:

create table student(

snum number(10), --primary key, --使用主键约束

sna varchar2(100), --not null,

sex char(4), --default '男' check(sex='男' or sex='女') not null,

age number check(age>0 and age<120),

qq number, --unique,

sal number(6,2),

mail varchar2(50)

后添加主键

```
-- constraints pk_student_snum primary key(snum)
-- constraints ck_student_sna check(sna is not null)
-- constraints ck_student_sex check(sex = '男' or sex='女')
--constraints uk_student_qq unique(qq)
)
alter table student add constraints pk_student_snum primary key(snum);--在创建表
```

```
alter table student drop constraints pk_student_snum--删除主键
```

```
alter table student modify sna varchar2(100) not null;--添加非空约束
```

```
alter table student modify sna varchar2(100) null;--修改字段为空
```

添加检查约束

```
alter table student add constraints ck_student_sex check(sex='男' or sex='女')--
```

```
alter table student drop constraints ck_student_sex --删除检查约束
```

```
alter table student add constraints uk_student_qq unique(qq)
```

```
alter table student drop constraints uk_student_qq
```

```
drop table student
```

问题1:学号重复了,数据还可以插入成功

使用主键约束:学号是唯一标识一条数据的,所以必须唯一且不能为空

--- (1)、在确定为主键的字段后添加 primary key关键字

---(2)、在创建表的后面使用: constraints pk\_表名\_字段名 primary

key(字段名)

--- (3)、在创建表后使用 alter table 表名 add constraints pk\_表

名\_字段名 primary key(字段名);

--删除主键: alter table 表名 drop constraints pk\_表名\_字段名

问题2:姓名可以为空。

使用非空约束

--- (1)、创建表的时候在字段后面添加not null

---(2)、在创建表字段后使用 constraints ck\_表名\_字段名 check(字

段名 is not null) 了解

--- (3)、在创建表后使用alter table 表名 modify 字段名 类型 not

null;

---(4)、修改字段可以存储空值: alter table 表名 modify 字段名

类型 null;

问题3: 性别不但可以为空,还可以为其他不知道的字符

使用检查约束

---(1)、创建表的时候在字段后使用 default 值 check(条件),

-----但是会允许空值的出现,并且默认值只有在字段不声明的情况

下生效

--- (2)、在创建表所有字段后使用: constraints ck\_表名\_字段名

check(条件)

---(3)、在创建表后使用: alter table 表名 add constraints ck\_表

名\_字段名 check(条件)

问题4: 年龄可以超过200

--使用检查约束条件

问题5: qq号一致

使用唯一约束

--- (1)、在字段后直接使用unique关键字



段名)

-- (2)、在所有字段后使用: constraints uk\_表名\_字段名 unique(字段名)

-- (3)、alter table 表名 add constraints uk\_表名\_字段名

--删除唯一约束: alter table 表名 drop constraints uk\_表名\_字段名

#### 第四节课: 外键约束学习:

##### 创建学生表

```
create table stu(
    snum number(10) primary key,
    sname varchar2(100) not null,
    sex char(4) check(sex='男' or sex='女'),
    cinfo number(10)-- references clazz(cid)--外键
    -- constraints fk_stu_cinfo foreign key(cinfo)

references clazz(cid)
)
alter table stu add constraints fk_stu_cinfo foreign key(cinfo)
references clazz(cid) on delete cascade--添加外键
alter table stu add constraints fk_stu_cinfo foreign key(cinfo)
references clazz(cid) on delete set null--添加外键
```

```
alter table stu drop constraints fk_stu_cinfo --删除外键
drop table stu
```

##### 创建班级表

```
create table clazz(
    cid number(10) primary key,
    cname varchar2(100) not null,
    cdesc varchar2(500)

)
```

##### 创建班级表数据

```
insert into clazz values(1,'软件01班','最厉害的班级');
insert into clazz values(2,'软件02班','挂科最多的班级');
insert into clazz values(3,'508班','大牛和最帅的班级');
select * from clazz
delete from clazz where cid=1
```

##### 创建学生数据

```
insert into stu values(1002,'张三','男',1);
insert into stu values(1003,'李四','女',2);
insert into stu values(1004,'王五','男',3);
insert into stu values(1005,'赵四','男',4);
select * from stu
```

问题1: 在插入学生信息的时候, 出现了不存在的班级信息?

##### 使用外键约束

-- (1)、在字段后使用 references 参照表表名(参照字段)

-- (2)、在所有字段后使用 constraints fk\_表名\_字段名

foreign key(字段名) references 参照表名(参照字段名)

-- (3)、在创建表后使用alter table 表名 add

constraints fk\_表名\_字段名 foreign key(字段名) references 参照表名(参照字段名)

--删除外键 alter table 表名 drop constraints fk\_表名\_

字段名

```
select table_name from all_all_tables where owner='SCOTT'
```

问题2: 在删除父表数据的时候需要先删除子表数据?

解决方式1: 先解除主外键关联, 然后删除数据, 再然后添加主外

解决方式2: 在创建外键的时候使用级联操作。

--在创建外键时 on delete cascade

--在创建外键时 on delete set null

怎么选取外键?

一般将主表的主键作为子表的外键

外键的值能为 not null? 不建议在外键后使用非空约束

第五节课: 序列

--序列

```
select cid from clazz order by cid
```

```
insert into clazz values(4,'402班级','老师最污的班');
```

```
insert into clazz values(10,'402班级','老师最污的班');
```

--问题1: 如果数据量比较大, 插入新的数据的时候id怎么选取才能避免重复?

使用序列

1、创建序列 create sequence 序列名

特点: 默认没有初始值的, nextval的值默认从1开始, 默认每次

特点: 可以使用 序列名.nextval作为主键使用

```
create sequence cnum;
```

```
select cnum.currval from dual--获取当前序列值
```

```
select cnum.nextval from dual--序列自增后返回当前值
```

```
truncate table clazz
```

```
select * from clazz
```

```
insert into clazz values(cnum.nextval,'402班级','老师最污的班');
```

自增+1

--将序列作为主键

问题2: 如果当前表中已有很多数据, 但是接下来需要使用序列进行主键自增?

创建序列时指定序列初始值及步长

```
create sequence cc
```

```
start with 90--设定序列初始值
```

```
increment by 5--设定步长
```

```
insert into clazz values(cc.nextval,'402班级','老师最污的班');
```

```
select max(cid) from clazz
```

```
--
```

```
select cc.nextval from dual
```

```
select cc.currval from dual
```

```
select * from clazz
```

删除序列

```
drop sequence cc
```

-----完整的序列格式-----

```
create sequence aa--序列名
```

```
start with 20 --设置序列初始值
```

```
increment by 10 --设置步长
```

```
maxvalue 10000--设置最大值
```

```
cache 10 -----设置缓存
```

第六节课: 索引

## 索引学习

```
select * from emp order by sal
select * from emp where sal=3000
```

问题：当表中数据量比较大的时候，使用条件查询就会出现效率的问题

使用索引进行解决

1、原理：类似于java中数组的折半查找，在数据库中数据使用B树形结构进行数据

的存储，

--这样可以对外提供快速的数据查找方式。

2、创建索引 create index 表名\_index\_字段名 on 表名(字段名)

特点：显式的创建，隐式的执行. 在数据中会给主键默认创建索引。

```
create index emp_index_sal on emp(sal)--创建查询索引
```

```
select * from emp where sal=3000
```

```
select * from emp order by sal
```

```
create index emp_index_sal_desc on emp(sal desc);--创建单排序索引
```

```
select * from emp order by sal, job
```

```
create index emp_index_sal_desc2 on emp(sal desc, job asc);--创建
```

多条件排序索引

3、删除索引 drop index 索引名

```
drop index emp_index_sal_desc
```

=====

oracle上课资料：

第一天：

第一节课：数据库基本概念学习

1、数据 书

其实就是表示信息的一种特殊符号的集合

例如：图片、txt、音频、视频等等

引入：互联网的发展交互的其实就是数据

2、数据库： 图书馆

把用来存储数据的东西称之为数据库。

3、数据库管理系统 应用程序 图书馆管理系统

将数据按照指定的方式存储到数据库中的系统称之为数据库管理系统

可以高效率的取出存储好的数据（重点）

4、数据库系统

数据库管理员

数据库

数据库管理系统

数据库应用系统

用户

第二节课：常见数据库及数据库存储结构

1、数据库存储数据结构

网状结构

层次结构

二维表结构：

二维表结构可以清晰的将数据与数据之间的关系表述清楚

注意：

单纯的存储数据很简单，但是存储好的数据又要将数据之间的关系描述清楚

就会比较困难，所以采用二维表的结构存储数据会比较合理

## 2、数据库管理系统

oracle:大型数据库管理系统

mysql: 中小型

DB2: 中小型

Acess: 小型

Sql Server: 大型数据库

.....

## 3、oralce的历史介绍 参照PPT

### 第三节课: oralce的安装及oracle目录结构介绍及卸载

#### 1、oracle的安装:

参照百度搜索“oracle11g安装图解”

全局数据库名:默认是orcl, 是用来唯一标示数据库的名字

SID: 在程序中唯一标示数据的名字

#### 2、oracle的服务: (需要开启的服务)

orcaleServiceorcl:orcale启动服务

OracleOraDb11g\_home1TNSListener:oracle监听器服务

OracleDBConsoleorcl:控制台服务

#### 3、oracle账户:

sys:超级管理员

system: 管理员

scott: 普通用户 默认密码: tiger

#### 4、oracle修改密码:

在cmd命令行中使用命令: sqlplus / as sysdba

然后使用命令:

alter user 用户名 identified by 新密码;

在cmd命令行中使用命令: sqlplus /nolog

然后使用命令: conn as sysdba

然后输入一个已知的用户名和密码

然后使用命令:

alter user 用户名 identified by 新密码;

### 第四节课: oracle目录及卸载

#### 1、oracle的目录介绍:

oradata: 数据库存储文件的目录

db\_home:

network >admin:配置网络服务和监听器服务

jdk:oracle自带jdk

deinstall:卸载命令

jdbc:与java交互的jar包

#### 2、oracle的卸载:

使用oracle自带的程序卸载

删除app目录

删除注册表

删除环境变量

### 第四节课: oracle的监听器及服务器配置

#### 问题1:

我的电脑装了oralce, 同桌怎么查看数据?

客户端:

针对普通用户使用的软件，主要是用来发送用户的请求。

服务器：

接受用户的请求，并处理用户的请求。

问题2：张三和李四都装了oracle，王五装了客户端，王五要访问张三，怎么访问？

ip

tcp

数据库全局名

端口号

问题3：

从服务器的角度观察，假如1000个请求过来，怎么区分那些是请求数据库的

监听器

第三方工具：plsql（不是客户端）

配置本地网络服务:(客户端)

第一种使用暴力方式直接操作：

修改：C:\app\Administrator\product\11.2.0\dbhome\_1\NETWORK\ADMIN\tnsnames.ora

ORCL = >网络服务名

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 服务器ip)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcl)

)

)

第二种方式：

使用图形界面：NCA

配置监听器：（服务器端）

常见问题：

配置监听器，计算机名必须是英文的

第一天的主要任务是安装数据库和配置数据库的网络服务，第二天正式进行sql语句的学习

第二天：

第一节课：sql语言介绍（参照PPT）及基本查询sql学习

1、数据库表的介绍

emp表：员工表

dept表：部门表

salgrady：薪资水平表

Balance：

2、基本的查询语句：

知识点：

select \* from 表名 查询表中所有的数据

select 字段名1, 字段名2, ..from 表名 查询指定字段的所有值，不同字段之间使

用逗号隔开

使用算术表达式：select 字段名(运算符)， 字段名(运算符)...from 表名

案例：

select \* from emp---查询所有员工信息

select ename, job, sal from emp---查询员工姓名，工作，薪资

查询所有员工一年的工资，以及每位员工年终3000后的所有薪资（不包括津贴）

select ename, job, sal\*12, sal\*12+3000 from emp 使用运算符

## 第二节课:

### 1、使用别名

as可以省略不写, 如果有特殊字符获取空格使用双引号添加别名

select ename 姓名, job 工作, sal\*12 年资, sal\*12+3000 年薪 from emp 直接在字  
段后面添加别名  
select ename "姓名", job 工作, sal\*12 ' 年资', sal\*12+3000 年薪 from emp 使用双引  
号添加别名, 不能使用单引号  
select ename as 姓名, job "工作", sal\*12 年资, sal\*12+3000 年薪 from emp 使用as关键字添加  
别名  
select ename "姓 名", job 工作, sal\*12 年资, sal\*12+3000 年薪 from emp 如果别名中有空格  
或者其他特殊字符, 使用双引号

### 2、使用链接符:

select ename, job, sal from emp;  
select ename||job, sal from emp; 使用||符号进行字符链接  
select ename||' 的工作是 '||job||' 并且月工资是 '||sal from emp; 字符链接,  
普通字符使用单引号

### 3、去除重复 distinct

select \* from emp;  
select job from emp--发现工作是重复的, 而工作种类是需要去除重复的

问题: 查看员工工作种类

select distinct job from emp 使用distinct关键字去除重复值

问题: 怎么多字段去除重复

select distinct job, sal from emp; 多字段去除重复, 去除的是重复的一整条信息

### 4、排序

知识点:

select 字段名, 字段名...from 表名 order by 字段名  
select \*from 表名 order by 字段名

讲解:

--查询员工姓名, 工作, 薪资

select ename, job, sal from emp

--查询按照薪资排序的员工姓名, 工作, 薪资

select ename, job, sal from emp order by sal; 使用order by排序

select ename, job, sal from emp order by sal desc; 使用desc关键从大到小排序

select ename, job, sal from emp order by sal asc; 使用asc关键从小到大排序,

默认是从小到大排序

select \* from emp order by sal, ename 多字段作为排序条

件的时候, 优先按照第一个字段排序, 然后依次按照其他字段排序。

select ename, job, sal 工资 from emp order by 工资 使用别名

select ename, job, sal from emp order by sal\*12 使用算术表达式

## 第三节课: 简单的where子句

使用where条件进行结果筛选

语法: select \* from 表名 where 条件; 注意: 条件中字段值区分大小写, 字段名不区分大小写, 字  
段值使用单引号括起来

1、在where中使用算术表达式 = , < , > , >= , <= , <>

2、使用order by 对筛选结果进行排序, order by 出现在where后面

查看工资等于1250的员工信息

```
select * from emp where sal='1250' --筛选条件是个数字也可以使用单引号
```

查看工作等于CLERK的员工信息

```
select * from emp where job='CLERK' --在筛选条件中字段值如果是字符需要加上单引号
```

```
select * from emp where job='clerk' --在sql语句中字段的值是区分大小写的
```

```
select * from emp where JOB='CLERK' --在sql中字段是不区分大小写的
```

查看工资大于1250的员工姓名和工作

```
select ename, job ,sal from emp where sal>'1250' order by sal --使用order by 对筛
```

选结果进行排序, order by 出现在where后面

查看工资大于等于2000的员工信息

```
select * from emp where sal>=2000;
```

查看工资小于等于2000的员工信息;

```
select * from emp where sal<=2000
```

查看工资不等于1500的员工信息

```
select * from emp where sal<>1500 order by sal
```

查看入职日期在81年后的员工信息

```
select * from emp where hiredate>'1981年12月31号'
```

```
select * from emp where hiredate>'1981/12/31'
```

```
select * from emp where hiredate>'31-12月-1981' --使用日期的默认格式查询符合要求的
```

数据, 日-月-年

#### 第四节课: where子句使用关键字

使用where子句进行结果的筛选

知识点: where查询条件中使用关键字

1、and 用于多条件的与筛选: select \* from 表名 where 条件 and 条件 and 条件....

2、or 用于多条件的或筛选: select \* from 表名 where 条件 or 条件 or 条件....

3、in 用于多条件的或筛选: select \* from 表名 where 字段名 in(值, 值, 值....)

4、like用于模糊查询: select \* from 表名 where 字段名 like '%值%' 包含

5、is null 和is not null 用来判断字段是否为空 select \* from 表名 where 字段名

is null

讲解: 在where子句中使用关键字

--查询工资在2000-3000之间的员工信息

```
select * from emp where sal>=2000 and sal<=3000 --使用and关键字进行"与"的多条件筛
```

选;

```
select * from emp where sal between 2000 and 3000; --使用between and 关键字进行筛
```

选;

--查询工作为SALESMAN, ANALYST, MANAGER的员工信息

```
select * from emp where job='SALESMAN' or job='ANALYST' or job='MANAGER' --使用or
```

关键字进行"或"的多条件筛选

```
select * from emp where job in('SALESMAN','ANALYST','MANAGER'); --使用in关键字进
```

行"或"的多条件筛选

```
select * from emp where job='ANALYST'
```

--查询姓名中包含s的, 以s开头的, 以s结尾的, 第二个字符为A的。

```
select * from emp where ename like '%S%'; --使用like关键字, 姓名中包含S的,%代表任
```

意多个字符

```
select * from emp where ename like 'S%'; --使用like关键字, 以S开头的
```

```
select * from emp where ename like '%S'; --以S结尾的
```

```
select * from emp where ename like '_A%' --使用"_"指定位置包含指定字符的信
```

息, "\_"代表任意一个字符

设置成为转译字符。

```
-----查询名字中包含下划线的用户信息
select * from emp where ename like '%A_%' escape 'A';--使用escape关键字将普通字符

--查询有津贴的员工信息
select * from emp where comm is not null;
select * from emp where comm is null;--查询没有津贴的员工信息
```

#### 第五节课：函数学习

```
--查询工作为SALESMAN，MANAGER并且工资大于2500的员工信息
--1、使用小括号提升where筛选条件的执行优先级别
--2、and的优先级别高于or
select * from emp
select * from emp where job='SALESMAN' or job='MANAGER' and sal>2500
select * from emp where (job='SALESMAN' or job='MANAGER') and sal>2500
```

使用函数 单行函数 多行函数 转换函数 其他函数

--单行函数学习：不改变原始数据，只改变结果

--1、字符函数

```
select * from emp;
select INITCAP(ename) from emp;--initcap函数将首字母大写
select lower(ename)from emp;--lower 字母小写
select replace(ename,'S','M') from emp;--replace 替换
```

--2、数值函数--Math

---伪表 dual

```
select * from dual
select abs(-3) 绝对值, ceil(3.1415926)向上取整, floor(3.1415926)向下取整, power(2,3)
```

幂, round(3.4)四舍五入 from dual

--3、日期函数

```
select months_between('13-12月-2016','13-10月-2016') from dual--months_between两个日期之
```

间的月份数

#### 第六节课：多行函数学习

多行函数：

max: max(字段名) 返回此字段的最大值

min: min(字段名) 返回此字段的最小值

avg: avg(字段名) 返回平均值

sum: sum(字段名) 返回字段的和

count: count

--count(\*), 用来查询表中有多少条记录

--count(字段名), 用来查询某个字段有值的个数

--count(distinct 字段名), 可以先去除重复再计数。

注意:

--多行函数不能和普通字段直接出现在查询语句中, 除非group by

--多行函数和单行函数不能直接出现在查询语句中, 除非group by

查看员工的最高工资

```
select max(sal), ename from emp--多行函数不能和普通字段直接出现在查询语句中, 除非
```

group by

```
select max(sal), lower(ename) from emp--多行函数和单行函数不能直接出现在查询语句
```

中, 除非group by

```
select ename from emp
```



查看员工的最低工资

```
select min(sal) from emp
```

查看员工的平均工资

```
select avg(sal) from emp
```

查看所有的员工工资之和

```
select sum(sal) from emp
```

查询公司有多少员工

```
select * from emp
```

```
select count(*) from emp--使用count(*)来查看一张表中有多少条记录
```

查询有津贴的员工人数

```
select count(comm) from emp--使用count(字段名),查询该字段有值的记录数
```

```
select count(ename) from emp
```

查询公司有多少工作种类

```
select count(job) from emp
```

```
select count(distinct job) from emp
```

第三天:

第一节课: 转换函数学习

转换函数:在转换的时候改变的是数据的类型,数据内容不会改变,可以指定格式。

1、to\_number: 将数字字符转换为数字类型的数值, to\_number(数字字符)

2、to\_char: 将数字转换为字符类型,将日期转换为字符类型 to\_char(数字/日期)

3、to\_date: 将字符类型的日期转换为日期类型: to\_date(char)

number-->char 转换的时候使用的是默认格式,

```
select to_char(123) from dual--将数字转换为字符类型,使用to_char(number)
```

```
select sal, to_char(sal) from emp
```

number-->char 使用指定的格式

to\_char(number,'格式'), 格式:

\$代表美元符号,9代码数字占位。例如,L999,999,999表示三位一组的显示

方式L代表人民币符号,0代表保留数字个数,不足使用0补充。

```
select to_char(sal),to_char(sal,'L999,999,999') from emp
```

```
select to_char(sal),to_char(sal,'L0000.00') from emp
```

char-->number to\_number(数字字符)

```
select to_number('123') from dual
```

char-->date,转换的字符必须是日期格式的字符串,默认格式 dd-mm-yyyy

注意:因为字符串有很多,所以在字符转换为日期的时候,需要指定格

式,因为日期是具备一定格式的字符组合。

字符转换为日期的时候,指定的格式为字符的日期顺序,无需指

定间隔符。

yyyy: 表示年 mm:表示月 dd表示日

```
select to_date('05-12月-2016') from dual;
```

```
select to_date('2016-05-12','yyyy-mm-dd') from dual--使用指定的格
```

式将指定的日期字符串转换为日期

```
select to_date('12-05-2016','mm/dd/yyyy') from dual
```

date-->char

注意:因为日期本身就具备一定的格式在不是指定格式的情况下会默认使

用dd-mm-yyyy格式显示数据

指定的格式会作为日期转换为字符串类型的显示格式存在。例

如:

候，不指定格式使用默认格式：dd-mm-yyyy

定格式将日期转换为字符串类型

定格式将日期转换为字符串类型

emp;--使用指定格式将日期转换为字符串类型

20'

20', 'yyyy/mm/dd')

```
yyyy-mm-dd'      '2016-12-05'
yyyy/mm/dd'      '2016/12/05'
yyyy"年"mm"月"dd"日"      2016年12月05日
select hiredate,to_char(hiredate) from emp--日期转换为字符的时
```

```
select hiredate,to_char(hiredate,'yyyy-mm-dd') from emp;--使用指
```

```
select hiredate,to_char(hiredate,'yyyy/mm/dd') from emp;--使用指
```

```
select hiredate,to_char(hiredate,'yyyy"年"mm"月"dd"日"') from
```

查询入职日期在81年10月20日后的员工信息

第一种：自动转型

```
select * from emp where hiredate>'20-10月-1981'
```

第二种：将日期转换为字符串

```
select * from emp where to_char(hiredate,'yyyy-mm-dd')>'1981-10-
```

第三种：

```
select * from emp where hiredate>to_date('1981-10-
```

## 第二节课：其他函数：

单行函数：lower

多行函数:min max sum avg count

转换函数:to\_number to\_char to\_date

其他函数：

1、nvl():nvl(字段名，执行)--相当于java中的if条件判断

2、nvl2():nvl2(字段名，值，值)--相当于java中的If() {} else {} 判断

3、decode():decode(字段名，条件1，执行内容1，条件2，执行内容2，条件3，执行内容3，

默认执行内容)

相当于java中if() {} else if() {} else if() {}...else {}

查询所有员工的月薪及姓名和工作

```
select * from emp
```

```
select ename,sal 基本工资,comm 绩效,sal+comm 月薪 from emp
```

```
select ename,sal 基本工资,comm 绩效,sal+nvl(comm,0)月薪 from emp
```

查询所有员工的月薪及姓名和工作

```
select ename,sal 基本工资,comm 绩效,nvl2(comm,sal+comm,sal)月薪 from emp
```

显示员工的职称

```
select ename,job,decode(job,'MANAGER','经理','SALESMAN','销售人员','普通员工')
```

from emp

## 第三节课：使用group by分组

在多行函数中不能直接使用普通字段，除非group by

在多行函数中不能直接使用单行函数，除非group by

group by学习：

---1、使用group by进行数据分组 select 多行函数，分组字段 from 表名 group

by 分组字段

---2、多字段进行分组的时候，按照字段顺序进行分组，第一条件分组完成后，继

续使用其他条件依次分组。

---3、group by依然可以和order by 联合使用

也要使用

——4、可以和单行函数联合进行分组，注意使用了单行函数那么在查询语句中必须

查询最高工资和员工数

```
select max(sal),count(*) from emp
```

查询不同部门的最高工资

```
select * from emp order by deptno
```

```
select deptno,max(sal) from emp group by deptno--使用group进行分组查询，分组的字
```

段可以出现在查询中，其他字段依然不可以

查询不同工作岗位的员工数

```
select * from emp for update
```

```
select lower(job),count(*) from emp group by lower(job)--使用单行函数进行分组
```

查询不同部门的不同工作岗位的人数

```
select deptno, job ,count(*) from emp group by deptno, job--使用多字段组合进行分组
```

```
select deptno, job ,count(*) from emp group by deptno, job order by deptno
```

查询不同部门的不同工作岗位的并且人数大于1的信息

```
select count(*) from emp where count(*)>3 group by deptno
```

```
select deptno, job ,count(*) from emp where count(*)>1 group by deptno, job order
```

by deptno

查询部门号大于10的不同部门的不同工作岗位的人数

```
select deptno, job ,count(*) from emp where deptno>10 group by deptno, job order
```

by deptno

使用having进行分组后筛选

having学习:

——1、使用group by分组后在进行数据筛选的时候，where中不能出现多行函数，所以使用

新的关键字having进行条件筛选

——2、where条件筛选的执行顺序: from-->where--->group -->select

——3、having条件筛选的执行顺序: from-->group by -->having-->select

——4、where的执行效率比having要高，能使用where的情况下尽量不要使用having

查询不同部门的不同工作岗位的并且人数大于1的信息

使用where语句进行筛选

where条件语句sql执行顺序: from-->where--->group -->select

```
select count(*) from emp where count(*)>1 group by deptno, job
```

使用having语句进行筛选

having条件语句的执行顺序:from-->group by -->having-->select

```
select deptno, count(*) from emp group by deptno having count(*)>5
```

```
select deptno, job ,count(*) from emp group by deptno, job having deptno>10
```

order by deptno

#### 第四节课: 插入数据学习及数据的备份

单表查询语句 (select)

1、插入数据(insert)

1、语法规范 insert into 表名(字段1, 字段2, 字段3, ....)values('值1','值2','值3'.....)

2、主键: 用来唯一标识一条数据的字段通常设置主键，主键是唯一不可以重复的

3、如果插入的数据是全字段数据，字段可以省略不写。部分字段，必须加上字段说明和字段值，但是主键不能为空

4、事务的提交: 如果一个事件是由多个动作组成，只要有一个动作没有执行成功则自动将数据回滚到原始状态，此们技术称之为事务

保证数据的安全和完整

事物的提交:

使用第三放插件的提交按钮

使用commit语句

```
select * from dept for update
```

在北京新建了一个名为LOL学院的新部门，请插入

```
insert into dept(deptno,dname,loc)values('50','lol学院','北京');
```

主键是唯一不可以重复的

```
insert into dept(deptno,dname,loc)values('50','教学部','北京');
```

如果插入的数据是全字段数据，字段可以省略不写。部分字段，必须加上字段说明和字段值，但是主

键不能为空

```
insert into dept values('80','教学部','北京');
```

```
insert into dept values('90','教学部','北京');
```

```
insert into dept values(100,'教学部','北京');
```

```
insert into dept values('110','教学部','北京');
```

## 2、创建数据的备份

(1) create table 表名 as 查询语句，创建的是和查询结果一样的表，查询结果是什么就会备份一

个相同的表

(2) insert into 表名 查询语句，注意：查询出来的结果在结构上必须和插入数据的表相同，字段

个数必须相同

(3) 注意：备份表只有字段和数据相同，并不会备份约束。

### 1、备份完整的数据和表

```
select * from dept;
```

```
create table tdept as select * from dept;--备份dept表和数据，只能备份数据
```

和字段

```
select * from tdept
```

### 2、备份完整表

```
create table tdept1 as select * from dept where 1>2--备份表，不备份数据。
```

### 3、备份部分数据和表.create table 表名 as 查询语句，创建的是和查询结果一样的表，查

询结果是什么就会备份一个相同的表

```
create table tdept2 as select dname,loc from dept
```

```
select *from tdept2
```

### 4、给备份表添加数据 insert into 表名 查询语句，注意：查询出来的结果在结构上必须和

插入数据的表相同，字段个数必须相同

```
select * from tdept1
```

```
insert into tdept1 select dname,loc from dept where deptno>40
```

```
select *from tdept2
```

```
insert into tdept2 select dname,loc from dept where deptno>40
```

## 第五节课：数据的更新和删除：

### 更新数据:update

语法结构：update 表名 set 字段名1='字段值1', 字段名2='字段值2', ... where 条件

将部门70的名字改为教学部2

```
update dept set dname='教学部2' where deptno='70'
```

```
update dept set dname='教学部2',loc='上海' where deptno='70'
```

```
select * from dept
```

### 删除数据: delete

1)删除语句: delete 表名 where 条件

--删除部门标号为70的数据

```
delete dept where deptno='70' ---删除指定数据
```

```
select * from dept
```

delete tdept---清空表数据

truncate table tdept---清空表数据建议使用truncate关键字，但是此关键字不能回滚数据

第四天:

第一节课: sql92学习

--查询员工姓名, 工作, 薪资, 部门名称

sql的联合查询(多表查询)

--1、sql92标准

---笛卡尔积: 一件事情的完成需要很多步骤, 而不同的步骤有很多方式, 完成这件事情的所有方

式称之为笛卡尔积

```
select * from emp--14
```

```
select * from dept--7
```

```
select * from emp,dept order by ename--7*14=98
```

等值链接, 链接条件。等值链接的时候字段的名字可以不相同, 但是字段的值要相同。

--查询员工姓名, 工作, 薪资, 部门名称

```
select * from emp,dept where emp.deptno=dept.deptno--使用等值链接进行结果筛选
```

```
select ename, job, sal, dname from emp,dept where emp.deptno=dept.deptno;--使用等值
```

链接查询指定字段数据

```
select ename, job, sal, emp.deptno, dname from emp,dept where
```

emp.deptno=dept.deptno;--多表查询的时候, 查看相同字段的值, 必须声明所在表

```
select emp.ename, emp.job, emp.sal, emp.deptno, dept.dname from emp,dept where
```

emp.deptno=dept.deptno;--在查询指定字段值的时候, 加上表名提高查询效率

```
select e.ename, e.job, e.sal, e.deptno, d.dname from emp e,dept d where
```

e.deptno=d.deptno;--多表查询中使用别名进行表的区分

```
select e.ename, e.job, e.sal, e.deptno, d.dname from emp e,dept d where
```

e.deptno=d.deptno order by deptno;--还可以使用order by 排序

非等值链接

---查询员工姓名, 工作, 工资, 工资等级

```
select * from salgrade
```

```
select * from emp,salgrade order by ename
```

```
select * from emp,salgrade where sal>losal and sal<hisal
```

```
select e.ename, e.job, e.sal, s.grade from emp e,salgrade s where e.sal>s.losal and
```

e.sal<s.hisal;

自连接: 使用频率不是很高, 因为自连接的条件要求不同信息共存在一张里, 其实就两张相同的表的等

值链接。

--查询员工姓名, 工作, 薪资, 及上级领导姓名

```
select * from emp for update
```

```
create table temp as select * from emp
```

```
select * from emp e,temp t where e.mgr=t.empno
```

第二节课: sql99

sql99学习:

交叉链接(没啥用)

```
select * from emp cross join dept
```

自然链接:natural join

(1)、自然链接会自动使用多表中所有相同字段(不但值相同, 名字也

要相同)进行筛选

前提: 多表一定要有同名同值的字段

注意: 自然连接会自动使用所有的相同字段进行结果筛

选

(2)、使用using关键字可以指定字段进行链接查询, 但是必须是同名字

段 inner join

(3)、使用on关键字可以直接在其后书写链接条件，没有限制

inner join

查询员工及员工所在的部门信息

```
select * from dept natural join emp
```

```
select * from dept d,emp e where d.deptno=e.deptno
```

问题1: 假如在链接查询的时候不想使用所有相同字段进行筛选怎么办?

----使用using关键字

---查询员工及员工所在的部门信息

```
select * from dept join emp using(deptno)
```

问题2: 假如在链接查询中没有同名字段，但是有同值字段怎么筛选?

使用on关键字

查询员工及员工所在的部门信息

```
select * from dept inner join emp on
```

dept.deptno=emp.deptno

查询部门及部门的城市信息

```
select * from dept inner join city on dept.loc=city.cid
```

外链接

(1)、左外链接 left outer join

(2)、右外链接 right outer join

(3)、全外链接 full outer join

左外链接

查询没有部门的员工信息及员工和部门信息

```
select * from emp e,dept d where d.deptno(+) =e.deptno--sql92
```

```
select * ---sql99
```

```
from emp e
```

```
left join dept
```

```
using(deptno)
```

--右外链接

```
select * from emp e,dept d where d.deptno=e.deptno(+)--sql92
```

```
select * ---sql99
```

```
from emp e
```

```
right outer join dept d
```

```
on e.deptno=d.deptno
```

--全外链接

```
select *
```

```
from emp e
```

```
full outer join dept d
```

```
on e.deptno=d.deptno
```

### 第三节课：自连接及三表联合查询

查询员工姓名，职位，薪资，上级领导姓名

```
select e.ename,e.job,e.sal,m.ename from emp e,emp m where e.mgr=m.empno--sql92
```

sql99实现:

```
select e.ename,e.job,e.sal,m.ename
```

```
from emp e
```

```
inner join emp m
```

```
on e.mgr=m.empno
```

---

sql99三表联合查询

查询员工姓名，工作，薪资，部门名称，城市名称

sql192实现：

```
select e.ename,e.job,e.sal,d.dname,c.cname
from emp e,dept d,city c
where e.deptno=d.deptno and d.loc=c.cid
```

sql199实现：

```
select e.ename,e.job,e.sal,d.dname,c.cname
from emp e
inner join dept d
on e.deptno=d.deptno
inner join city c
on d.loc=c.cid
```

查询工资高于2000的员工姓名，工作，薪资，部门名称，城市名称

sql192实现：

```
select e.ename,e.job,e.sal,d.dname,c.cname
from emp e,dept d,city c
where e.deptno=d.deptno and d.loc=c.cid and sal>2000
```

sql199实现：

```
select e.ename,e.job,e.sal,d.dname,c.cname --查询内容
from emp e --查询表
inner join dept d --链接表
on e.deptno=d.deptno --链接条件
inner join city c --链接表
on d.loc=c.cid --链接条件
where e.sal>2000 --筛选条件
```

#### 第四节：单行子查询

子查询（单行子查询、多行子查询）

单行子查询

```
select 内容 from 表名 where 子查询语句
```

1、什么时候使用单行子查询？

---在不能直接获取有效信息的时候，考虑使用子查询

2、单行子查询的注意事项？

---子查询的结果必须只有一个值

---可以直接使用算术链接符

---子查询出现在where中，一般出现在条件语句的右边

问题：查询所有比雇员“CLARK”工资高的员工信息

解读1：查询雇员CLARK的工资

```
select sal from emp where ename='CLARK'
```

解读2：查询工资大于2450的员工信息

```
select * from emp where sal>'2450'
```

使用子查询

```
select * from emp where sal>(select sal from emp where ename='CLARK')
```

```
select * from emp where (select sal from emp where ename='CLARK')<sal
```

查询工资高于平均工资的员工的名字和工资

```
select * from emp where sal>(select avg(sal) from emp) order by sal
```

查询和scott属于同一部门且工资比他低的员工资料

```
select * from emp where deptno=(select deptno from emp where ename='SCOTT') and
```

```
sal<(select sal from emp where ename='SCOTT')
```

查询工资最高的员工资料

```

select * from emp where sal=(select max(sal) from emp)
查询职务和scott相同，雇佣时间早的员工信息
select * from emp where job=(select job from emp where ename='SCOTT') and
hiredate<(select hiredate from emp where ename='SCOTT')
查询工资比scott高或者雇佣时间早的员工编号和名字
select empno,ename from emp where sal>(select sal from emp where ename='SCOTT') or
hiredate<(select hiredate from emp where ename='SCOTT')

查询工资高于任意一个CLERK的所有员工信息
select * from emp where sal>(select min(sal) from emp where job='CLERK') and job<>'CLERK'

```

---

## 第五节课：多行子查询

多行子查询学习：其实就使用使用关键字：any all in

注意：子查询返回多个值建议使用多行子查询，返回的单个值使用单行子查询

```

select * from emp where sal> any (select sal from emp where job='CLERK') and job<>'CLERK'
查询工资高于所有SALESMAN的员工信息
select * from emp where sal>(select max(sal) from emp where job='SALESMAN')--单行子查询的写法

select * from emp where sal>all (select sal from emp where job='SALESMAN' ;
查询部门20中同部门10的雇员工作一样的雇员信息
select * from emp where deptno='20' and (job='MANAGER' or job='PRESIDENT' or job='CLERK')
select * from emp where deptno='20' and job in ('MANAGER', 'PRESIDENT', 'CLERK')
select * from emp where deptno='20' and job in (select job from emp where deptno='10')
select * from emp where deptno='20' and job= any (select job from emp where deptno='10')
select job from emp where deptno='10'

```

## 第六节课：用户创建

创建用户

当前用户介绍：权限级别的分配

```

-----system: 系统账户
-----sys: 超级管理员
-----scott: 普通用户

```

- 1、 创建自定义用户：create user 用户名 identified by 密码；  
普通用户不具备创建用户的权限，需要使用system账户进行创建  
直接创建好的用户不能登录，还需要使用system进行权限的分配（角色：一系列权限的集合）

```

create user zyp identified by 123456;
给创建的用户赋予角色。      grant 角色名, 角色名... to 用户名;
grant connect to zyp;--赋予链接库权限
grant resource to zyp;--赋予操作资源权限
grant dba to zyp;--赋予DBA角色
grant dba to zyp

删除用户权限
revoke dba from zyp
revoke connect from zyp

2、 删除用户
drop user zyp

```

---

## 第五天：

### 第一节课：创建表和字段讲解



## 一创建表学习

1、 创建表的基本语句: create table 表名 (字段名 类型, 字段名 类型, 字段名 类型,...)

创建学生表: 学号, 姓名, 性别, 年龄, qq号, 邮箱。

```
create table student(  
    snum number(10), ---指定学号的数字长度为10  
    sname varchar2(100), ---指定姓名的存储上限为100个字节  
    sex char(4),  
    age number,  
    qq number,  
    sal number(6, 2), ---指定工资的整数位为6和小数位为2  
    mail varchar2(50)  
)
```

## 2、字段类型

number类型: 数字类型, 用来声明存储数字的字段。 number(指定数字的整数长度, 指定数字的小数位长度)

varchar2类型: 字符类型, 用来声明存储字符的字段。会根据存储的数据自动增长大小, varchar2(存储上限)

char类型: 字符类型, 用来声明存储字符的字符。会开辟指定大小的内存来存储数据。

区别1: char的存储效率高于varchar2

区别2: char是开辟指定大小的内存空间, varchar2是根据数据大小来开辟空间的大小

date类型: 存储日期类型

## 第二节课: 修改表学习:

修改表

在命令窗口查看表结构 desc 表名

添加新字段 alter table 表名 add 字段名 类型

```
alter table student add phone number(11)
```

修改字段类型 alter table 表名 modify 字段名 类型

```
alter table student modify sname varchar2(200)
```

删除字段 alter table 表名 drop column 字段名

```
alter table student drop column phone
```

修改表名 rename 表名 to 新的表名

```
rename student to stu
```

```
rename stu to student
```

删除表 drop table 表名

```
drop table student
```

## 第三节课: 约束学习(根据情况切分成两节课)

select \* from student for update

添加数据

问题1: 学号重复了, 数据还可以插入成功

```
insert into student(snum, sna, sex, age, qq, sal, mail) values('123', '张三', '男', 18, 12345678, 88.88, '12345678@qq.com')
```

```
insert into student values(123, '李四', '男', 18, 7890122, 88.99, '7890122@qq.com')
```

问题2: 姓名可以为空。

```
insert into student values(456, '柳岩', '女', 18, 666999, 99.66, '666999@qq.com')
```

```
insert into student (snum, sex, age, qq, sal, mail)  
values(789, '女', 18, 888444, 99.66, '888444@qq.com')
```

问题3: 性别不但可以为空, 还可以为其他不知道的字符

```

insert into student (snum, sna, age, qq, sal, mail) values (108, '景甜', 18, 000999, 99.66, '000999@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values (102, '景甜', 'a', 18, 000999, 99.66, '000999@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values (102, '景甜', '女', 18, 000999, 99.66, '000999@qq.com')

```

问题4: 年龄可以超过200

```

insert into student (snum, sna, sex, age, qq, sal, mail) values (103, '唐嫣', '女', 23, 78900, 99.66, '78900@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values (103, '唐嫣', '女', 230, 78900, 99.66, '78900@qq.com')

```

问题5: qq号一致

```

insert into student (snum, sna, sex, age, qq, sal, mail) values (104, '关晓彤', '女', 19, 111000, 99.66, '111000@qq.com')

insert into student (snum, sna, sex, age, qq, sal, mail) values (105, '袁华', '男', 22, 111000, 99.66, '111000@qq.com')

```

约束学习:

```

create table student(
    snum number(10), --primary key, --使用主键约束
    sna varchar2(100), --not null,
    sex char(4), --default '男' check(sex='男' or sex='女') not null,
    age number check(age>0 and age<120),
    qq number, --unique,
    sal number(6,2),
    mail varchar2(50)
    -- constraints pk_student_snum primary key(snum)
    -- constraints ck_student_sna check(sna is not null)
    -- constraints ck_student_sex check(sex='男' or sex='女')
    --constraints uk_student_qq unique(qq)
)

alter table student add constraints pk_student_snum primary key(snum); --在创建表后添加主键

alter table student drop constraints pk_student_snum --删除主键

alter table student modify sna varchar2(100) not null; --添加非空约束
alter table student modify sna varchar2(100) null; --修改字段为空

alter table student add constraints ck_student_sex check(sex='男' or sex='女') --添加检查约束

alter table student drop constraints ck_student_sex --删除检查约束

alter table student add constraints uk_student_qq unique(qq)
alter table student drop constraints uk_student_qq

```

drop table student

问题1: 学号重复了, 数据还可以插入成功

使用主键约束: 学号是唯一标识一条数据的, 所以必须唯一且不能为空

--- (1)、在确定为主键的字段后添加 primary key关键字

--- (2)、在创建表的后面使用: constraints pk\_表名\_字段名 primary

key(字段名)

```

名_字段名 primary key(字段名);

-- (3)、在创建表后使用 alter table 表名 add constraints pk_表
--删除主键: alter table 表名 drop constraints pk_表名_字段名
问题2:姓名可以为空。
使用非空约束
-- (1)、创建表的时候在字段后面添加not null
---(2)、在创建表字段后使用 constraints ck_表名_字段名 check(字
段名 is not null) 了解
null;
类型 null;

问题3: 性别不但可以为空, 还可以为其他不知道的字符
使用检查约束
--(1)、创建表的时候在字段后使用 default 值 check(条件),
-----但是会允许空值的出现, 并且默认值只有在字段不声明的情况
下生效
-- (2)、在创建表所有字段后使用: constraints ck_表名_字段名
check(条件)
--(3)、在创建表后使用: alter table 表名 add constraints ck_表
名_字段名 check(条件)

问题4: 年龄可以超过200
--使用检查约束条件
问题5: qq号一致
使用唯一约束
-- (1)、在字段后直接使用unique关键字
-- (2)、在所有字段后使用: constraints uk_表名_字段名 unique(字
段名)
-- (3)、 alter table 表名 add constraints uk_表名_字段名
unique(字段名)
--删除唯一约束: alter table 表名 drop constraints uk_表名_字段名
第四节课: 外键约束学习:
创建学生表
create table stu(
    snum number(10) primary key,
    sname varchar2(100) not null,
    sex char(4) check(sex='男' or sex='女'),
    cinfo number(10)-- references clazz(cid)--外键
    -- constraints fk_stu_cinfo foreign key(cinfo)
references clazz(cid)
)
alter table stu add constraints fk_stu_cinfo foreign key(cinfo)
references clazz(cid) on delete cascade--添加外键
alter table stu add constraints fk_stu_cinfo foreign key(cinfo)
references clazz(cid) on delete set null--添加外键

alter table stu drop constraints fk_stu_cinfo --删除外键
drop table stu
创建班级表
create table clazz(
    cid number(10) primary key,

```

```
cname varchar2(100) not null,
cdesc varchar2(500)
```

```
)
```

创建班级表数据

```
insert into clazz values(1,'软件01班','最厉害的班级');
insert into clazz values(2,'软件02班','挂科最多的班级');
insert into clazz values(3,'508班','大牛和最帅的班级');
select * from clazz
delete from clazz where cid=1
```

创建学生数据

```
insert into stu values(1002,'张三','男',1);
insert into stu values(1003,'李四','女',2);
insert into stu values(1004,'王五','男',3);
insert into stu values(1005,'赵四','男',4);
select * from stu
```

问题1: 在插入学生信息的时候, 出现了不存在的班级信息?

使用外键约束

- (1)、在字段后使用 references 参照表名(参照字段)
- (2)、在所有字段后使用 constraints fk\_表名\_字段名

foreign key(字段名) references 参照表名(参照字段名)

- (3)、在创建表后使用alter table 表名 add

constraints fk\_表名\_字段名 foreign key(字段名) references 参照表名(参照字段名)

- 删除外键 alter table 表名 drop constraints fk\_表名\_

字段名

```
select table_name from all_all_tables where owner='SCOTT'
```

问题2: 在删除父表数据的时候需要先删除子表数据?

解决方式1: 先解除主外键关联, 然后删除数据, 再然后添加主外

键关联

解决方式2: 在创建外键的时候使用级联操作。

- 在创建外键时 on delete cascade
- 在创建外键时 on delete set null

怎么选取外键?

一般将主表的主键作为子表的外键

外键的值能为 not null? 不建议在外键后使用非空约束

第五节课: 序列

--序列

```
select cid from clazz order by cid
insert into clazz values(4,'402班级','老师最污的班');
```

```
insert into clazz values(10,'402班级','老师最污的班');
```

问题1: 如果数据量比较大, 插入新的数据的时候id怎么选取才能避免重复?

使用序列

1、创建序列 create sequence 序列名

特点: 默认没有初始值的, nextval的值默认从1开始, 默认每次

自增+1

特点: 可以使用 序列名.nextval坐为主键使用

```
create sequence cnum;
select cnum.currval from dual--获取当前序列值
select cnum.nextval from dual--序列自增后返回当前值
```

-将序列作为主键

```
truncate table clazz
select * from clazz
insert into clazz values(cnum.nextval,'402班级','老师最污的班');
```

问题2: 如果当前表中已有很多数据, 但是接下来需要使用序列进行主键自增?

创建序列时指定序列初始值及步长

```
create sequence cc
start with 90--设定序列初始值
increment by 5--设定步长
insert into clazz values(cc.nextval,'402班级','老师最污的班');
```

```
select max(cid) from clazz
--
select cc.nextval from dual
select cc.currval from dual
select * from clazz
```

删除序列

```
drop sequence cc
```

-----完整的序列格式-----

```
create sequence aa--序列名
start with 20 --设置序列初始值
increment by 10 --设置步长
maxvalue 10000--设置最大值
cache 10 -----设置缓存
```

## 第六节课: 索引

### 索引学习

```
select * from emp order by sal
select * from emp where sal=3000
```

问题: 当表中数据量比较大的时候, 使用条件查询就会出现效率的问题

使用索引进行解决

1、原理: 类似于java中数组的折半查找, 在数据库中数据使用B树形结构进行数据

的存储,

--这样可以对外提供快速的数据查找方式。

2、创建索引 create index 表名\_index\_字段名 on 表名(字段名)

特点: 显示的创建, 隐式的执行. 在数据中会给主键默认创建索引。

```
create index emp_index_sal on emp(sal)--创建查询索引
select * from emp where sal=3000
select * from emp order by sal
create index emp_index_sal_desc on emp(sal desc);--创建单排序索引
select * from emp order by sal, job
create index emp_index_sal_desc2 on emp(sal desc, job asc);--创建
```

多条件排序索引

3、删除索引 drop index 索引名

```
drop index emp_index_sal_desc
```

=====