

MySQL第3天

一、列约束

4.默认值约束

在插入数据时候，不提供值的位置会使用默认值。

- 设置默认值

使用default关键字来设置默认值

- 应用默认值

要使用默认值的位置直接使用default

```
insert into 数据表名称 values(1, default,...)
```

给指定的列提供值，未出现的列会自动使用默认值

```
insert into 数据表名称(列1, 列2) values(值1, 值2);
```

5.检查约束

check: 用户可以根据指定的约束条件进行验证

```
create table student(  
    score tinyint check(score>=0 && score<=100)  
);
```

MySQL不支持检查约束，后期由JS来完成验证

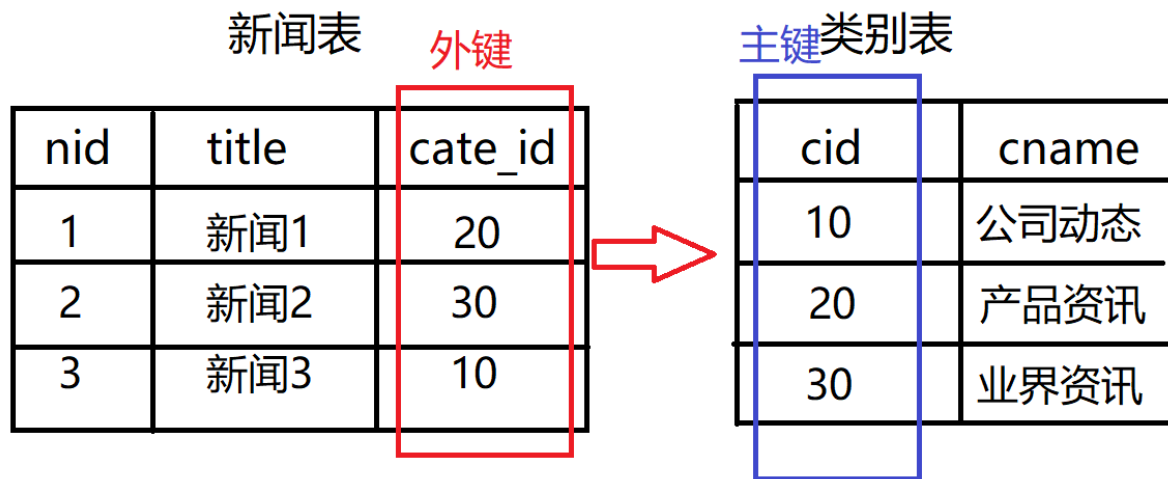
6.外键约束

声明了外键约束的列，插入的值必须在另一个表的主键列中出现过；目的是为了让两个表关联起来。

外键列的类型和对应的另一个表主键列的类型要保持一致。

外键约束允许插入null

```
foreign key(外键列) references 另一个表(主键列)
```



列约束练习:

编写脚本文件tedu.sql, 先删除再创建数据库tedu, 设置编码为UTF-8, 进入数据库, 创建保存部门数据的表dept, 包含的列有编号(主键约束、自增列)、部门名称(唯一约束), 插入以下数据

10 研发部 20 市场部 30 运营部 40 设计部

创建保存员工数据的表emp, 包含的列有编号(主键约束、自增列)、姓名(非空约束)、性别(默认值约束)、生日、工资、所属部门编号(外键约束); 插入若干条数据。

```
-- 设置客户端连接服务器端的编码
set names utf8;
-- 删除数据库, 如果存在
drop database if exists tedu;
-- 创建新的数据库, 设置存储字符的编码
create database tedu charset=utf8;
-- 进入数据库
use tedu;
-- 创建保存部门数据的表
create table dept(
    did int primary key auto_increment,
    dname varchar(16) unique
);
-- 插入数据
insert into dept values(10,'研发部');
insert into dept values(20,'市场部');
insert into dept values(30,'运营部');
insert into dept values(40,'设计部');
-- 创建保存员工数据的表
create table emp(
    eid int primary key auto_increment,
    ename varchar(16) not null,
    sex boolean default 0,    -- 1 - 男    0 - 女
    birthday date,
    salary decimal(8,2), -- 999999.99
    deptid int,
    foreign key(deptid) references dept(did)
);
-- 插入数据
insert into emp values(null,'yu',default,'1990-5-20',18900,30);
insert into emp(ename,birthday,salary,deptid) values('xin','1981-8-30',50000,20);
INSERT INTO emp VALUES(NULL,'Tom',1,'1990-5-5',6000,20);
```

```
INSERT INTO emp VALUES(NULL, 'Jerry', 0, '1991-8-20', 7000, 10);
INSERT INTO emp VALUES(NULL, 'David', 1, '1995-10-20', 3000, 30);
INSERT INTO emp VALUES(NULL, 'Maria', 0, '1992-3-20', 5000, 10);
INSERT INTO emp VALUES(NULL, 'Leo', 1, '1993-12-3', 8000, 20);
INSERT INTO emp VALUES(NULL, 'Black', 1, '1991-1-3', 4000, 10);
INSERT INTO emp VALUES(NULL, 'Peter', 1, '1990-12-3', 10000, 10);
INSERT INTO emp VALUES(NULL, 'Franc', 1, '1994-12-3', 6000, 30);
INSERT INTO emp VALUES(NULL, 'Tacy', 1, '1991-12-3', 9000, 10);
INSERT INTO emp VALUES(NULL, 'Lucy', 0, '1995-12-3', 10000, 20);
INSERT INTO emp VALUES(NULL, 'Jone', 1, '1993-12-3', 8000, 30);
INSERT INTO emp VALUES(NULL, 'Lily', 0, '1992-12-3', 12000, 10);
INSERT INTO emp VALUES(NULL, 'Lisa', 0, '1989-12-3', 8000, 10);
INSERT INTO emp VALUES(NULL, 'King', 1, '1988-12-3', 10000, 10);
INSERT INTO emp VALUES(NULL, 'Brown', 1, '1993-12-3', 22000, NULL);
```

二、简单查询

1. 查询特定的列

```
-- 示例：查询出所有员工的编号和姓名
select eid,ename from emp;
-- 练习：查询出所有员工的姓名、性别、生日、工资
select ename,sex,birthday,salary from emp;
```

2. 查询所有的列

```
select eid,ename,sex,birthday,salary,deptid from emp;
select * from emp;
```

3. 给列起别名

```
-- 示例：查询出所有员工的编号和姓名，使用别名
select eid as bianhao,ename as xingming from emp;
-- 练习：查询出所有员工的姓名、性别、生日、工资，使用一个字母作为别名
select ename as a,sex as b,birthday as c,salary as d from emp;
select ename a,sex b,birthday c,salary d from emp;
```

通过as关键字来设置别名，as关键字可以省略

4. 显示不同的记录

```
-- 示例：查询出所有的员工的性别有哪些(相同性别只显示1个)
select distinct sex from emp;
-- 练习：查询出员工都分布在哪些部门
select distinct deptid from emp;
```

distinct 不同的、有区别的

5.查询时执行计算

```
-- 示例：计算出2+3+4*8*2.45
select 2+3+4*8*2.45;
-- 练习：查询出所有员工的姓名及其年薪
select ename,salary*12 from emp;
-- 练习：假设每个员工的工资增长2000元，年终奖为20000元，查询出所有员工的姓名及其年薪，给列起别名。
select ename xingming,(salary+2000)*12+20000 nianxin from emp;
```

6.查询结果排序

```
-- 示例：查询出所有的部门，结果按照编号升序排列
select * from dept order by did asc;
-- 示例：查询出所有的部门，结果按照编号降序排列
select * from dept order by did desc;
-- 练习：查询出所有的员工，结果按照工资降序排列
select * from emp order by salary desc;
-- 练习：查询出所有的员工，结果按照性别升序排列
select * from emp order by sex asc;
-- 练习：查询出所有的员工，结果按照年龄从小到大排列(生日降序)
select * from emp order by birthday desc;
-- 练习：查询出所有的员工，结果按照姓名的升序排列
select * from emp order by ename asc;
-- 练习：查询出所有的员工，结果按照工资升序排列，如果工资相同按照姓名升序排列
select * from emp order by salary asc,ename asc;
-- 练习：查询出所有的员工，结果按照性别升序排列，如果性别相同按照年龄从小到大排列
select * from emp order by sex asc,birthday desc;
```

如果按照字符串排序，按照首个字符的编码来排列的。

不加排序规则，默认是按照升序排列

asc -> ascendant 升序的

desc -> describe 描述

desc -> descendant 降序的

7.条件查询

```
-- 示例：查询出编号为5的员工
select * from emp where eid=5;
-- 练习：查询出姓名为tom的员工
select * from emp where ename='tom';
-- 练习：查询出所有的女员工
select * from emp where sex=0;
-- 练习：查询出20号部门的员工
select * from emp where deptid=20;
-- 练习：查询出工资高于8000的员工
select * from emp where salary>8000;
-- 练习：查询出不在20号部门的员工
select * from emp where deptid!=20;
-- 练习：查询出没有明确部门的员工
select * from emp where deptid is null;
-- 练习：查询出有明确部门的员工
```

```

select * from emp where deptid is not null;
-- 练习: 查询出工资在6000~8000之间的员工有哪些
select * from emp where salary>=6000 && salary<=8000;
select * from emp where salary>=6000 and salary<=8000;
-- 练习: 查询出工资在6000以下, 或者10000以上的员工
select * from emp where salary<6000 || salary>10000;
select * from emp where salary<6000 or salary>10000;
-- 练习: 查询出20号部门或者30号部门的员工
select * from emp where deptid=20 || deptid=30;
select * from emp where deptid in(20,30);
-- 练习: 查询出不在20号部门, 并且不在30号部门的员工
select * from emp where deptid!=20 && deptid!=30;
select * from emp where deptid not in(20,30);

```

比较运算符: > < >= <= = !=

is null / is not null 找某一列的值为null, 或者不为null

&& (and) 并且

||(or) 或者

in() 满足等于其中的一个值

not() 满足不等于其中任意一个值

8.模糊条件查询

```

-- 示例: 查询出姓名中含有字母e的员工
select * from emp where ename like '%e%';
-- 练习: 查询出姓名中以e结尾的员工
select * from emp where ename like '%e';
-- 练习: 查询出姓名中倒数第2个字符是e的员工
select * from emp where ename like '%e_';

```

% 匹配的符号, 匹配任意个字符 >=0

_ 匹配任意一个字符 =1

注意事项: 以上两个匹配的符号必须结合like关键字使用

9.分页查询

查询的结果中有太多的数据, 一次显示不完可以做成分页显示。

需要两个已知条件: 当前的页码、每页的数据量

每页开始查询的值 = (当前的页码 - 1) * 每页的数据量

分页查询语法

```

select * from emp limit 开始查询的值, 每页的数据量;

```

假设每页显示5条数据, 查询出前4页每页的数据

```
-- 第1页
select * from emp limit 0,5;
-- 第2页
select * from emp limit 5,5;
-- 第3页
select * from emp limit 10,5;
-- 第4页
select * from emp limit 15,5;
```

注意事项:

1. limit后开始查询的值不能写运算，必须直接写结果
2. limit后开始查询的值和每页的数据量必须是数字，不能加引号

三、复杂查询

1.聚合查询/分组查询

聚合函数: count()/sum()/avg()/max()/min()

数量 总和 平均 最大 最小

```
-- 示例: 查询出所有员工的数量
select count(*) from emp; -- 17
-- 练习: 使用员工的编号列查询出员工数量
select count(eid) from emp; -- 17
-- 练习: 使用员工所属部门编号列查询出员工数量
select count(deptid) from emp; -- 16
-- 练习: 查询出所有女员工的工资总和
select sum(salary) from emp where sex=0;
-- 练习: 查询出10号部门的平均工资
select avg(salary) from emp where deptid=10;
-- 练习: 查询出年龄最大的男员工的生日
select min(birthday) from emp where sex=1;
-- 练习: 查询出20号部门的最高工资
select max(salary) from emp where deptid=20;

-- 示例: 查询出男女员工的数量、工资总和
select count(*),sum(salary),sex from emp group by sex;
```

分组查询中: 通常只是查询分组条件和聚合函数。

练习: 查询出各部门的平均工资, 最高工资, 最低工资。

练习: 查询出工资在6000以上的男员工, 工资最高的前3个人。

练习: 删除代码, 保留注释, 重新编写查询语句。