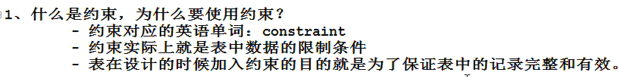
# 约束

## 什么是约束



## 分类



# 外键约束

## demo

有一个部门的表，还有一个员工表，

create database day16;

use day16;

create table dept(

did int primary key auto\_increment,

dname varchar(30)

);

create table emp(

eid int primary key auto\_increment,

ename varchar(20),

salaly double,

dno int

);

insert into dept values(null,'研发部');

insert into dept values(null,'销售部');

insert into dept values(null,'人事部');

insert into dept values(null,'扯淡部');

insert into dept values(null,'牛宝宝部');

insert into emp values(null,'班长',10000,1);

insert into emp values(null,'美美',10000,2);

insert into emp values(null,'小凤',10000,3);

insert into emp values(null,'如花',10000,2);

insert into emp values(null,'芙蓉',10000,1);

insert into emp values(null,'东东',800,null);

insert into emp values(null,'波波',1000,null);

把研发部删除？

\* 研发部下有人员？该操作不合理。

\* 引入外键约束？

\* 作用：保证数据的完整性。

## 定义外键约束

可以直接在create语句中定义外键

foreign key 当前表名 (字段名) references 目标表名 (目标表的主键)

### 添加约束

创建完语句后，可以直接使用修改语句定义

alter table 表名 add foreign key 当前表名 (字段名) references 目标表名 (目标表的主键);

alter table emp add foreign key emp(dno) references dept(did);

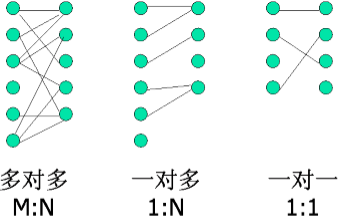
### 删除约束



### 修改约束

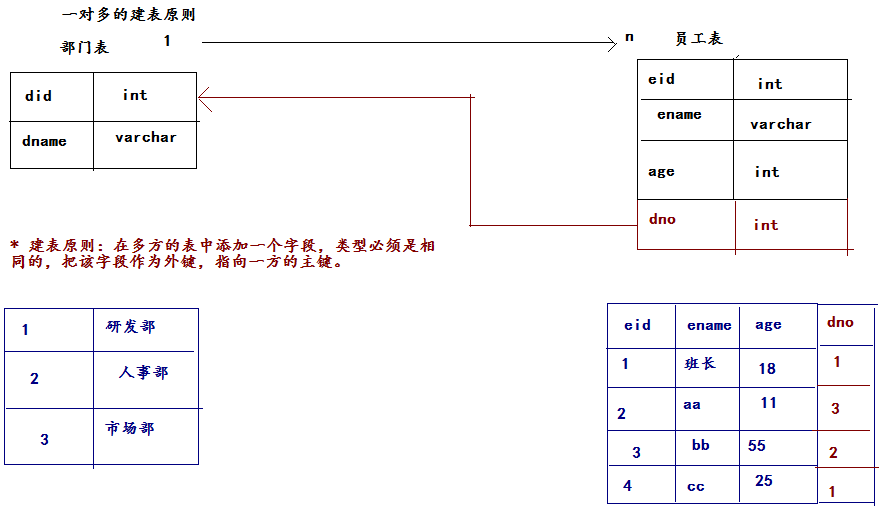
# 数据库的设计

## 概述



## 1:n

生活中一个部门下有多个员工，一个员工属于一个部门。



注意：

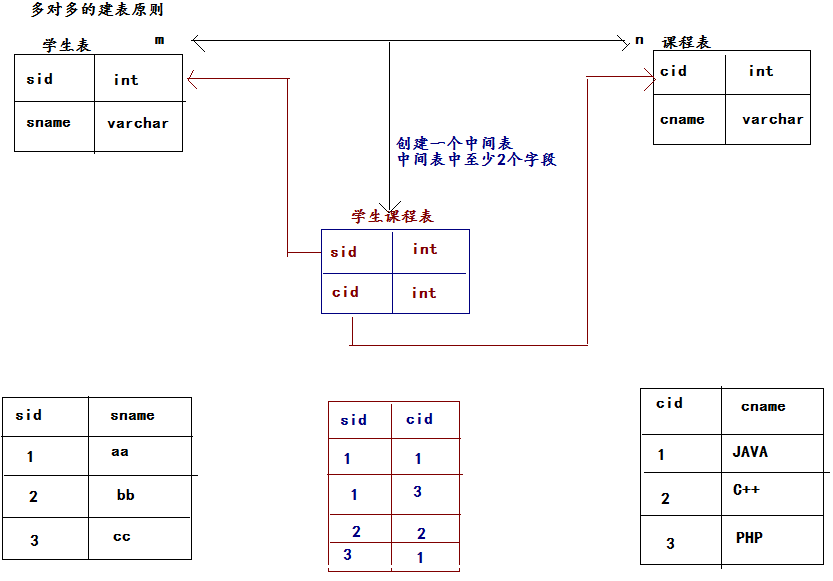
在多方需要添加一个字段，并且和一放主键的类型必须是相同的。

把该字段作为外键指向一方的主键。

一方部门 ， 多方员工

## n:n

学生可以选择多门课程，课程又可以被多名学生选择。



建表原则：

拆开两个一对多的关系，中间创建一个中间表，至少有两个字段。作为外键指向两个多对多关系表的主键。

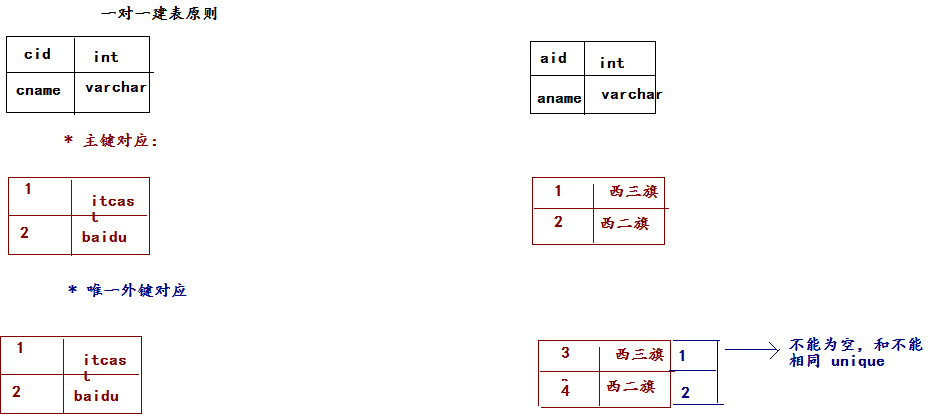
联系的属性及两个实体的主标识形成关系表。

关系表的主键为两个实体主标识的组合

## 1:1

公司，地址，一个公司对应的是一个地址。 一张表包含公司名称、公司地址。

根据公司的业务需求，会把公司这张表拆开，形成一对一。



建表原则

第一种是主键对应

将两个表的主键进行关联

第二种是唯一外键对应

在任何一方加一个外键，但是需要设置成唯一（unique），指向另一方的主键

