

06_数据库多表关系&维护

2024全新MySQL企业开发版

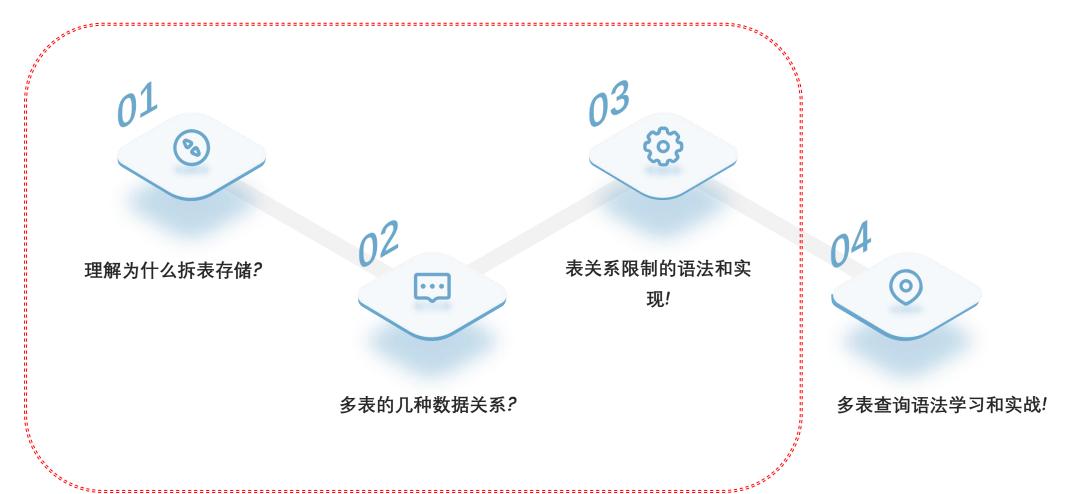




2 数据库多表关系维护







"捋顺多表查询学习的思路!!"



"为什么要拆表存储数据?"

学生表

sid	sname	sage
1	德华	18
2	二狗子	19
3	驴蛋蛋	20

学生和分数表

sid	sname	sage	score	course
1	德华	18	80	java
1	德华	18	90	mysql
2	二狗子	19	60	java
2	二狗子	19	70	mysql
3	驴蛋蛋	20	80	java
3	驴蛋蛋	20	100	mysql

学生表

sid	sname	sage
1	德华	18
2	二狗子	19
3	驴蛋蛋	20

分数表

cid	cnum	cid	sid
1	80	1	1
2	90	2	1
3	60	1	2
4	70	2	2
5	80	1	3
6	100	2	3

01

拆优表势

提高查询效率

分表存储不同数据类型可以根据数据特征和查询模式进行优化

防止数据冗余

将不同数据类型存储在不同的表中可以避免数据冗余。

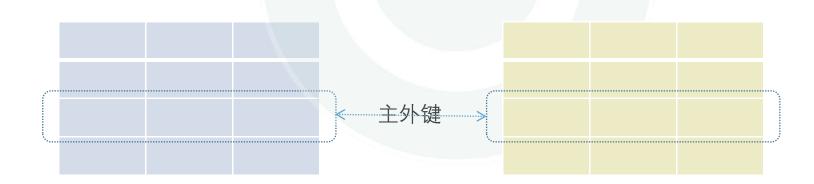
更方便进行数据

分表存储不同数据类型可以简化数据管理和维护工作



"数据库多表关系" 概念

- 1. 关系数据库中,数据都是按照类别存储到对应的表结构。
- 2. 表和表之间可以通过主外键建立表(数据)之间关系。
- 3. 拆表存储可以减少数据冗余,并提高数据查询和操作的效率。
- 4. 因为拆表存储的原因,我们不仅要掌握<mark>单表的查询语法</mark>还需要掌握<mark>多表关联查询语法</mark>。
- 5. 关系型数据库需要<mark>提前创建库表结构</mark>,所以我们还需要<mark>通过分析数据关系在建表添加合理约束</mark>。





多(两)表(数据)关系

1 一对一

两个表之间的每行数据都是唯一的对应关系! 例如,一个员工与其唯一的员工档案。

2 一对多

一个表关联另一个表多行数据,反方向只关联一行数据! 例如,一个作者与多个文章的关系。

3 多对多

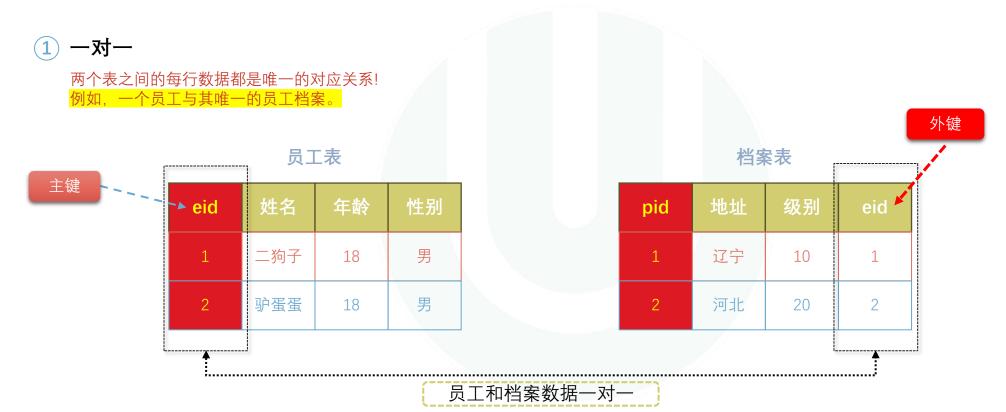
两个表中的记录都可以与对方表中的多个记录相关联! 例如,学生和课程之间的关系,一个学生可以选修多门课程,而一门课 程也可以由多个学生选修。

1

数据库多表关系



多表关系



特点

- 一条员工数据至多对应一条档案数据
- 一条档案数据至多对应一条员工数据

1

数据库多表关系



多表关系



- 一条作者数据至多对应多条文章数据
- 一条文章数据至多对应一条作者数据



多表关系

3 多对多



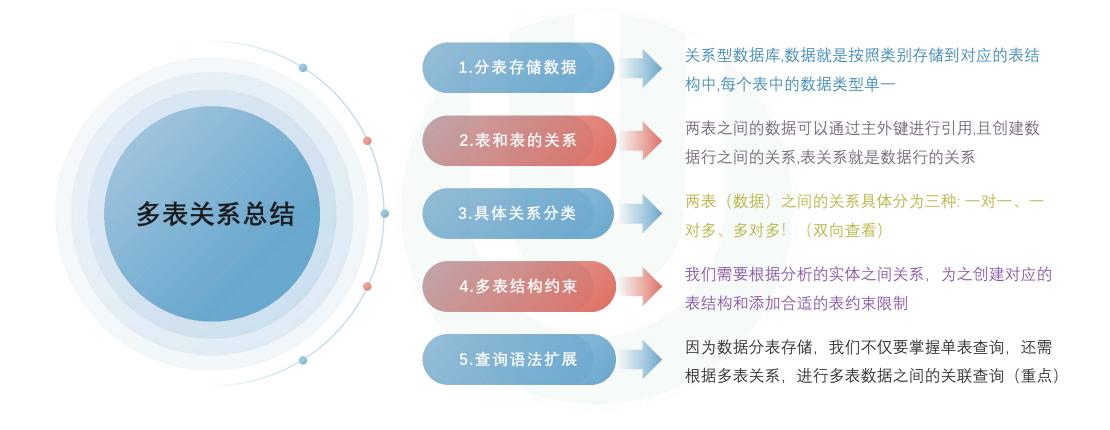
学生和课程数据多对多

特点

- 一条学生数据至多对应多条课程数据
- 一条课程数据至多对应多条学生数据

多对多关系需要创建中间表, 存储两个外键, 建立数据关联







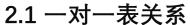


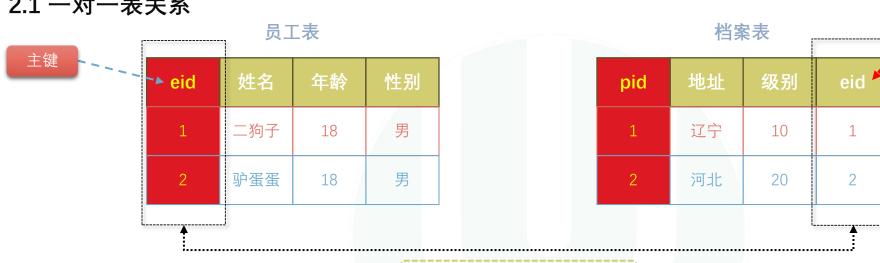
2 数据库多表关系维护





外键





员工和档案数据一对一

一对一特点: 双方的数据行都对应对方至多一条数据

一对一解决冗余:一对一并不能解决数据冗余问题。

一对一存在意义:一张表中有冷热数据,将表拆分成两张表存储,一张存热,一张存冷数据!

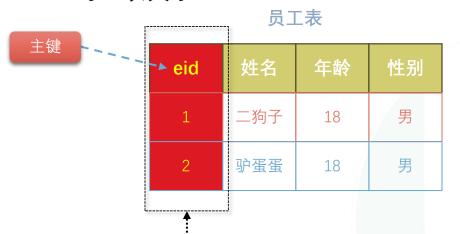
一对一关系维护:子表外键对应的数据不能重复(外键唯一)

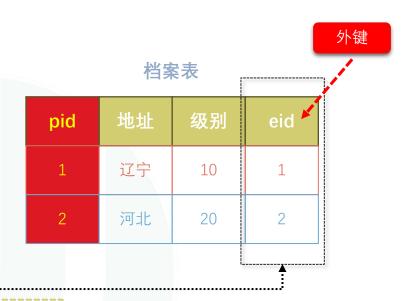
方案1: 子表的外键和主键融合, 两张表公用同一个主键

方案2: 子表的外键添加唯一约束



2.1 一对一表关系





员工和档案数据一对一

主表: 员工表

CREATE TABLE emp(e_id int primary key auto_increment, e_name varchar(20) NOT NULL, e_age int default 18, e_gender char default '男');

子表:档案表1(外键唯一)

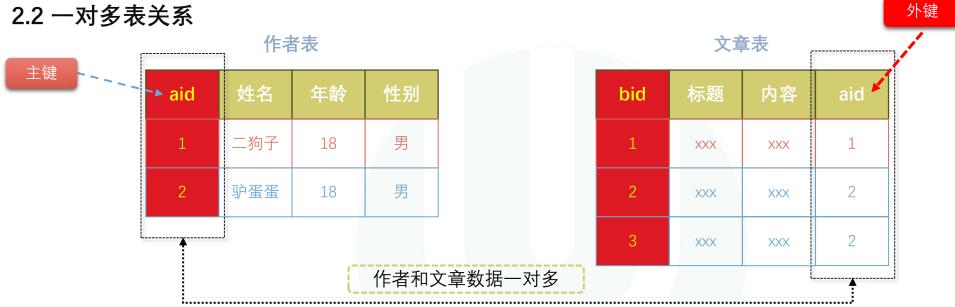
```
CREATE TABLE profile1(
    p_id int primary key auto_increment,
    p_address varchar(100) NOT NULL,
    p_level int default 10,
    e_id int unique, # 外键唯一
    CONSTRAINT s_p_1 FOREIGN
    KEY(e_id) references emp(e_id)
    );
```

子表:档案表2(共用主键)

```
create table profile2(
e_id_int primary key auto_increment,
p_address varchar(100) NOT NULL,
p_level_int_default 10,
CONSTRAINT s_p_2 FOREIGN KEY
(e_id) references emp(e_id)
);
```







一对多特点: 主表对应多条子表数据,子表对应主表至多一条数据

一对多解决冗余:一对多可以解决数据冗余问题。

一对多存在意义:一个实体对应多个子元素,通过分表,解决数据冗余问题和提高数据操作效率!

一对多关系维护:子表外键对应的数据可以重复(外键不唯一)

方案1: 正常创建主子表,添加外键约束即可



2.2 一对多表关系



主表: 作者表

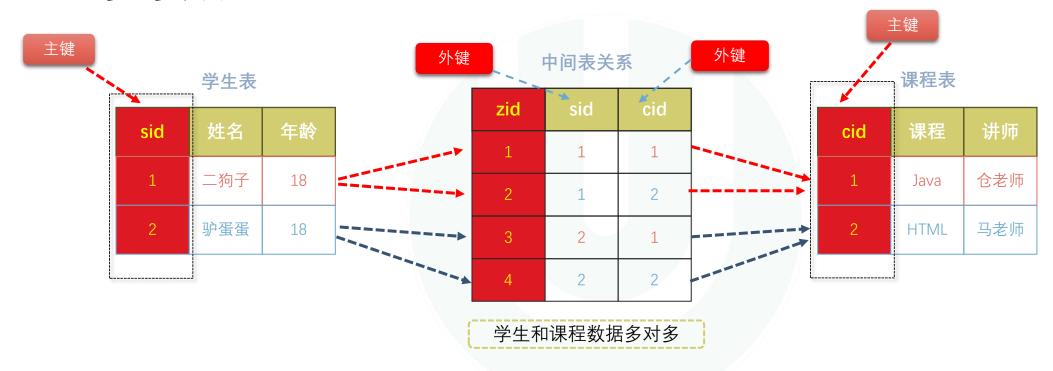
```
CREATE TABLE author(
  a_id int primary key auto_increment,
  a_name varchar(20) NOT NULL,
  a_age int default 18,
  a_gender char default '男'
```

子表: 文章表(外键)

```
CREATE TABLE blog(
  b_id int primary key auto_increment,
  b_title varchar(100) NOT NULL,
  b_content varchar(600) NOT NULL,
  a id int, #外键
  CONSTRAINT a_b_fk FOREIGN KEY(a_id)
references author(a_id)
```

U) 尚硅谷 www.atguigu.com

2.3 多对多表关系

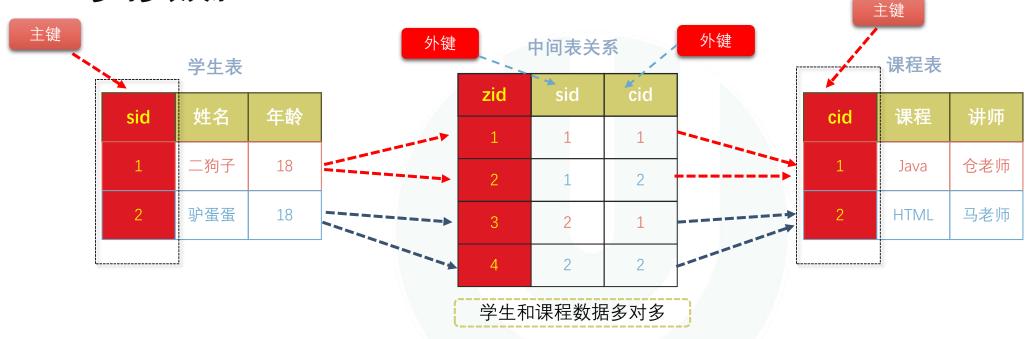


多对多特点: 双方的数据行都可以对应对方多条数据

多对多特殊情况:多对多需要创建中间表建立数据之间的关联。

多对多关系维护:中间表包含两个外键,主表数据之间间接关联

2.3 多对多表关系



主表: 学生表

CREATE TABLE student(s_id int primary key auto_increment, s_name varchar(20) NOT NULL,

s_age int default 18

中间表: student_course

CREATE TABLE student_course(

sc_id int primary key auto_increment,

s_id int,

c_id int,

FOREIGN KEY(s_id) references student(s_id),

FOREIGN KEY(c_id) references course(c_id)

CREATE TABLE course(

主表: 课程表

c_id int primary key auto_increment,

c_name varchar(10) NOT NULL,

c_teacher varchar(10)



感谢观看 THANK YOU