

多表关系

一对多

多对多

一对一

一对多

案例：

员工表(emp) N

id	name	age
1	张无忌	20
2	杨逍	33
3	赵敏	18
4	常遇春	43

1 部门表(dept)

id	name
1	研发部
2	市场部
3	财务部
4	销售部

一个部门下可以对应多个员工，在数据库层面，在多的-方建立外键

多对多

id	name	no
1	熊绮丝	2000100101
2	谢逊	2000100102
3	殷天正	2000100103
4	韦一笑	2000100104

学生表(student)

id	name
1	Java
2	PHP
3	MySQL
4	Hadoop

N 课程表(course)

通常建立中间表，一个关联学生表的外键，一个关联课程表的外键

```

1 CREATE TABLE student2
2 (
3 id int auto_increment PRIMARY KEY COMMENT '主键ID',
4 name VARCHAR(10) COMMENT '姓名',
5 no VARCHAR(10) COMMENT '学号'
6 )COMMENT '学生表';
7 INSERT INTO student2 (name,no)VALUES('黛伊丝','2000100101'),('谢逊','200010
0102'),('韦一笑','2000100103');
8
9 CREATE TABLE course(
10 id INT auto_increment PRIMARY KEY COMMENT'主键ID',
11 name VARCHAR(10) COMMENT'课程名称'
12 )COMMENT'课程表';
13
14 INSERT INTO course (name) VALUES ('JAvA'),('PHP'),('MySQL'),('Hadoop');
15
16
17 CREATE TABLE student_table(
18 id INT auto_increment PRIMARY KEY,
19 studentid INT NOT NULL,
20 courseid INT NOT NULL,
21 CONSTRAINT fk_courseid FOREIGN KEY (courseid) REFERENCES course(id),
22 CONSTRAINT fk_studentid FOREIGN KEY (studentid) REFERENCES student2(id)
23 );
24
25
26 INSERT INTO student_table (studentid,courseid) VALUES (1,1),(1,2),(1,3),(2
,2),(2,3),(3,1);

```

创建中间表student_table将中间的studentid和courseid分别关联为student2中的id和course中的id外键。

一对一

- 一对一

- 案例: 用户与用户详情的关系
- 关系: 一对一关系, 多用于单表拆分, 将一张表的基础字段放在一张表中, 其他详情字段放在另一张表中, 以提升操作效率
- 实现: 在任意一方加入外键, 关联另外一方的主键, 并且设置外键为唯一的(UNIQUE)

id	name	age	gender	phone
1	黄渤	45	1	18800001111
2	冰冰	35	2	18800002222
3	码云	55	1	18800008888
4	李彦宏	50	1	18800009999

用户基本信息表(tb_user)

id	degree	major	primaryschool	middleschool	university	userid
1	本科	舞蹈	静安区第一小学	静安区第一中学	北京舞蹈学院	1
2	硕士	表演	朝阳区第一小学	朝阳区第一中学	北京电影学院	2
3	本科	英语	杭州市第一小学	杭州市第一中学	杭州师范大学	3
4	本科	应用数学	阳泉第一小学	阳泉区第一中学	清华大学	4

用户教育信息表(tb_user_edu)

创建一对一的表

SQL

```

1 CREATE TABLE tb_user(
2   id INT auto_increment PRIMARY KEY COMMENT '主键ID',
3   name VARCHAR(10) COMMENT '姓名',
4   age INT comment '年龄',
5   gender CHAR(1) COMMENT '1:男 2:女',
6   phone CHAR(11) COMMENT '手机号'
7 )COMMENT '用户基本信息表';
8
9 CREATE TABLE tb_user_edu(
10  id int auto_increment PRIMARY KEY COMMENT '主键ID',
11  degree VARCHAR(20) COMMENT '学历',
12  major VARCHAR(50) COMMENT '专业',
13  primaryschool VARCHAR(50) COMMENT '小学',
14  middleschool VARCHAR(50) COMMENT '中学',
15  university VARCHAR(50) COMMENT '大学',
16  user_id INT UNIQUE COMMENT '用户ID',
17  CONSTRAINT fk_user_id FOREIGN KEY(user_id) REFERENCES tb_user(id)
18 );

```