

学习目标

* 能够掌握ER图的概念

- * 项目计划--需求调研--需求分析--概要设计--数据库设计--详细设计--编码--测试--上线

- * E:Entity R: RelationShip : 概要设计

- * 矩形--实体,椭圆形--属性, 菱形--关系

* 能够使用PowerDesigner画ER图

- * CDM

* 能够掌握Oracle的约束

- * primary key,not null,unique,check,foreign key

* 能够使用PowerDesigner设计的数据库表与表之间的关系

- * PDM

- * 1对多, 多对多

* 能够使用PowerDesigner根据已经设计好的表关系生成DDL语句

- * 自定生成DDL

* 能够掌握数据库设计范式

- * 1NF:原子性

- * 2NF : 主键, 不能产生部分依赖, 要完成依赖 (多对多关系, 尽量不要使用复合主键)

- * 3NF : 不能产生传递性依赖 (一对多的关系)

* 回顾

* Oracle 的锁

- * 按级别: S锁, X锁

- * 用户和系统

- * 自动锁

- * DML锁

- * TX锁: 事务锁, 行锁, 排他锁

- * TM锁

- * S锁：多个事务可以获得S锁，假如只有一个事务获得S锁，它是可以更新，但是假如多个事务

- 获得S锁，就只能查询

- * create index

- * lock table stu in share mode

- * X锁：只有一个事务可以获得排它锁

- * DDL , lock table stu in exclusive mode

- * RX:行级排他锁

- * DML语句 , lock table stu in row exclusive mode

- * RS：行级共享锁

- * lock table stu in row share mode

- * SRX：共享行级排他锁

- * lock table stu in share row exclusive mode

- * DDL锁

- * 系统锁

- * 显示锁

- * 死锁

- * 悲观锁和乐观锁

- * Oracle常见运算符

- * ||

- * 集合运算符

- * union

- * union all

- * intersect

- * minus

- * Oracle常见的函数

- * to_char

- * to_date
- * to_number
- * rank() over(order by age)--> 1 2 3 3 5 6 7
- * dense_rank() over(order by age) -->1 2 3 3 4 5 6
- * row_number over(order by age) --> 1 2 3 4 5 6 7
- * partition by: ank() over(partition by name order by age)
- * nvl,nvl2,decode

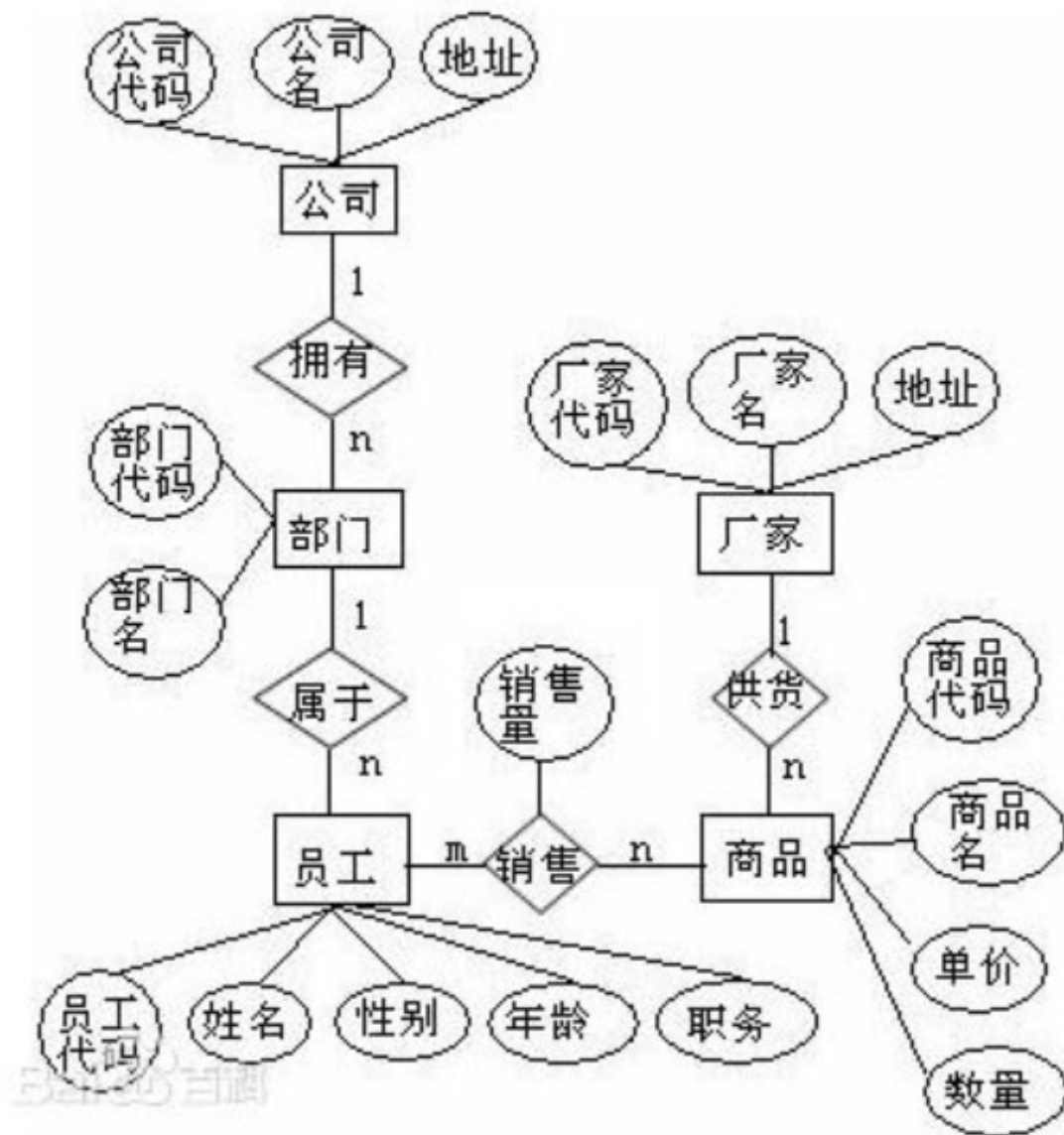
* 能够掌握ER图的概念

```

1 * ER图概念
2   E-R图也称实体-联系图(Entity Relationship Diagram),
3   提供了表示实体类型、属性和联系的方法,用来描述现实世界的概念模型。
4   它是描述现实世界关系概念模型的有效方法。是表示概念关系模型的一种方式。
5 * E-R图表示方法
6   * 用“矩形框”表示实体型,矩形框内写明实体名称;
7   * 用“椭圆图框”或圆角矩形表示实体的属性,并用“实心线段”将其与相应关系的“实体型”连接起来;
8   * 用“菱形框”表示实体型之间的联系成因,在菱形框内写明联系名,
9     并用“实心线段”分别与有关实体型连接起来,同时在“实心线段”旁标上联系的类型 (1:1,1:n或
10
11 * CDM图示在概要设计阶段创建
12 * 了解开发过程: 项目计划--需求调研--需求分析--概要设计--数据库设计--详细设计--编码--

```

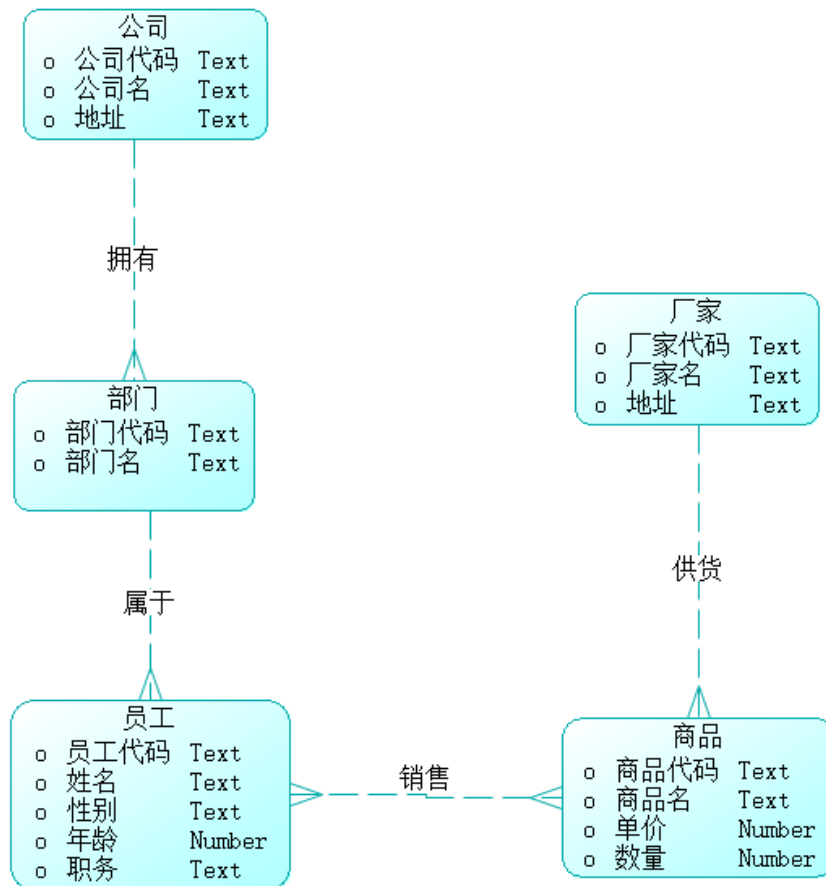
* ER图 (来自百度)



* 能够使用PowerDesigner画ER图

* 参考[08-PowerDesign画ER](#)

* 画ER成果图



* 能够掌握Oracle的约束

* 概念：对表中的数据进行限定，保证数据的正确性、有效性和完整性。

* 约束分类

* 主键约束：primary key

* 非空约束：not null

* 唯一约束：unique

* 外键约束：foreign key

* check约束

- 1 * 主键约束: primary key
- 2 * 非空且唯一
- 3 * 一张表只能有一个字段为主键
- 4 * 主键就是表中记录的唯一标识
- 5 * 测试

```

6  * 在创建表时, 添加主键约束
7  create table emp1(
8      id int primary key, -- 给id添加主键约束
9      name varchar(20)
10 );
11 insert into emp1(id,name) values(1,'xiaohei');
12 insert into emp1(id,name) values(1,'xiaohei');
13 重复插入:
14 ORA-00001: unique constraint (SCOTT.SYS_C0011212) violated (违反)
15 * 删除主键
16 ALTER TABLE emp1 DROP PRIMARY KEY;
17 select dbms_metadata.get_ddl('TABLE','EMP1') from dual;
18 * 创建完表后, 添加主键
19 ALTER TABLE emp1 MODIFY id INT PRIMARY KEY;
20 select dbms_metadata.get_ddl('TABLE','EMP1') from dual;
21
22 * 非空约束: not null, 值不能为null
23 * 创建表时添加约束
24 CREATE TABLE emp2(
25     id int primary key,
26     NAME VARCHAR(20) NOT NULL -- name为非空
27 );
28 * 唯一约束: unique, 值不能重复
29     * 创建表时, 添加唯一约束
30     CREATE TABLE emp3(
31         id int primary key,
32         phone_number VARCHAR(20) UNIQUE -- 添加了唯一约束
33     );
34
35 * 外键约束: foreign key, 让表于表产生关系, 从而保证数据的正确性。
36     * 在创建表时, 可以添加外键
37     * 语法:
38     create table 表名(
39         ....
40         外键列
41         constraint 外键名称 foreign key (外键列名称) references 主表名称(主表
42         );
43     * 删除外键
44     ALTER TABLE 表名 DROP CONSTRAINT 外键名称;
45     * 创建表之后, 添加外键

```

```

46 ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY (外键字段名称) REFERENC
47 * 级联删除
48 语法: ALTER TABLE 表名 ADD CONSTRAINT 外键名称
49 FOREIGN KEY (外键字段名称) REFERENCES 主表名称(主表列名称) on delete casc
50
51 * 测试
52 * 创建公司表, 创建部门表
53 create table company(
54     id int primary key not null,
55     code varchar2(18) not null,
56     name varchar2(50) not null,
57     loc varchar2(50))
58 create table deptm(
59     id int primary key not null,
60     deptno int not null,
61     name varchar2(50) not null,
62     cid int,
63     constraint company_deptm_fk foreign key(cid) references company(id))
64 * 查看主外键
65 select dbms_metadata.get_ddl('TABLE','COMPANY') from dual;
66 select dbms_metadata.get_ddl('TABLE','DEPTM') from dual;
67 * 删除外键
68 alter table deptm drop constraint company_deptm_fk;
69 * 创建表添加外键
70 alter table deptm add constraint company_deptm_fk foreign key(cid) refere
71 * 删除表
72     * 先删除有外键的表, 再删除主表
73     drop table deptm;
74     drop table company;
75     * 假如先删除主表会报异常: unique/primary keys in table referenced by fore
76 * 级联删除
77 * 插入数据
78 insert into company values(1,'91220105MA14YRT65G','亮哥教育','广州');
79 insert into deptm values(1,1001,'教学部',1);
80 insert into deptm values(2,1002,'行政部',1);
81 insert into deptm values(3,1003,'渠道部',1);
82 * 查询
83 select * from company;
84 select * from deptm;
85 * 删除

```

```

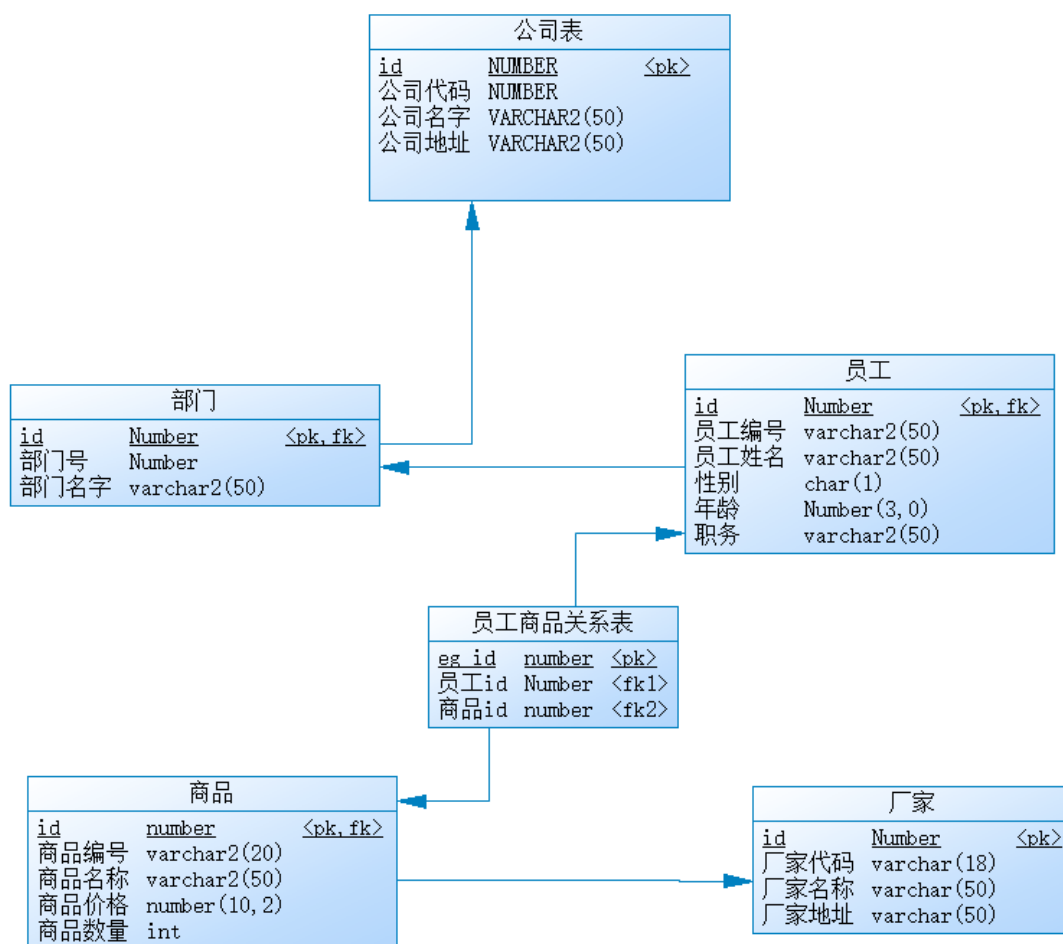
86      * 先删除外键表的所有数据，再删除主表的数据
87      delete from deptm;
88      delete from company where id=1;
89      * 假如直接删除主表的数据
90      会报异常: ORA-02292:integrity constraint(SCOTT.COMPANY_DEPTM_FK)violated
91      解决方案:
92      * 级联删除(on delete cascade)
93          * alter table deptm drop constraint company_deptm_fk;
94          * alter table deptm add constraint company_deptm_fk
95            foreign key(cid) references company(id) on delete cascade;
96          * delete from company where id=1;
97          * 结果: 删除公司，公司所有部门也被删除
98      * 级联删除(on delete set null)
99          * alter table deptm drop constraint company_deptm_fk;
100         * alter table deptm add constraint company_deptm_fk
101           foreign key(cid) references company(id) on delete set null;
102         * delete from company where id=1;
103         * 结果: 删除公司，部门表把外键设置为null
104 * 温馨提醒: 外键约束是个有争议性的约束，它一方面能够维护数据库的数据一致性，数据的完整
105   防止错误的垃圾数据入库； 另外一方面它会增加表插入、更新等SQL性能的额外开销
106   不少系统里面通过业务逻辑控制来取消外键约束
107
108 * check 约束
109     * 创建表
110     CREATE TABLE employee
111     (
112         emp_no CHAR(8) PRIMARY KEY NOT NULL,    --工号，主键，非空
113         emp_name VARCHAR2(30) NOT NULL, --姓名,非空
114         emp_id VARCHAR2(18), --身份证号，代表18位整数
115         emp_address varchar2(30),
116         emp_age number(3,0),
117         check(length(emp_id)=18 and emp_age>22 and emp_age<35)
118     );
119 insert into emp1(emp_no,emp_name,emp_id,emp_address,emp_age)
120 values('lg006','张大','441521199909092111','4401',25);
121 假如插入不合法数据汇报: ORA-02290:check constraint(SCOTT.SYS_C0011318)violated
122

```


* 能够使用PowerDesigner设计数据库表与表之间的关系

* 表与表之间的关系：一对一（不常用），一对多，多对一，多对多

* 物理模型（PDM,Physical Data Model）



* 能够使用PowerDesigner根据已经设计好的表关系生成DDL语句

* 步骤：DataBase-->Generate DataBase

* 生成的DDL语句

```
1 /*=====*/
2 /* DBMS name:      ORACLE Version 11g          */
3 /* Created on:     2019/9/23 14:17:51          */
4 /*=====*/
```

```
5
6
7 alter table department
8     drop constraint FK_DEPARTME_REFERENCE_COMPANY;
9
10 alter table eg
11     drop constraint FK_EG_REFERENCE_EMPLOYEE;
12
13 alter table eg
14     drop constraint FK_EG_REFERENCE_GOODS;
15
16 alter table employee
17     drop constraint FK_EMPLOYEE_REFERENCE_DEPARTME;
18
19 alter table goods
20     drop constraint FK_GOODS_REFERENCE_FACTORY;
21
22 drop table company cascade constraints;
23
24 drop table department cascade constraints;
25
26 drop table eg cascade constraints;
27
28 drop table employee cascade constraints;
29
30 drop table factory cascade constraints;
31
32 drop table goods cascade constraints;
33
34 /*=====*/
35 /* Table: company */
36 /*=====*/
37 create table company
38 (
39     id            NUMBER            not null,
40     code          NUMBER,
41     name          VARCHAR2(50),
42     loc           VARCHAR2(50),
43     constraint PK_COMPANY primary key (id)
44 );
```

```

45
46 /*=====*/
47 /* Table: department */
48 /*=====*/
49 create table department
50 (
51     id                Number                not null,
52     deptno            Number,
53     name              varchar2(50),
54     constraint PK_DEPARTMENT primary key (id)
55 );
56
57 /*=====*/
58 /* Table: eg */
59 /*=====*/
60 create table eg
61 (
62     eg_id             number                not null,
63     eid              Number,
64     gid              number,
65     constraint PK_EG primary key (eg_id)
66 );
67
68 /*=====*/
69 /* Table: employee */
70 /*=====*/
71 create table employee
72 (
73     id                Number                not null,
74     empNo            varchar2(50),
75     name             varchar2(50),
76     sex              char(1),
77     age              Number(3,0),
78     job              varchar2(50),
79     constraint PK_EMPLOYEE primary key (id)
80 );
81
82 /*=====*/
83 /* Table: factory */
84 /*=====*/

```

```

85 create table factory
86 (
87     id                Number                not null,
88     code              varchar(18),
89     name              varchar(50),
90     loc               varchar(50),
91     constraint PK_FACTORY primary key (id)
92 );
93
94 /*=====*/
95 /* Table: goods */
96 /*=====*/
97 create table goods
98 (
99     id                number                not null,
100    goodsNo           varchar2(20),
101    name              varchar2(50),
102    price             number(10,2),
103    num               int,
104    constraint PK_GOODS primary key (id)
105 );
106
107 alter table department
108     add constraint FK_DEPARTME_REFERENCE_COMPANY foreign key (id)
109         references company (id);
110
111 alter table eg
112     add constraint FK_EG_REFERENCE_EMPLOYEE foreign key (eid)
113         references employee (id);
114
115 alter table eg
116     add constraint FK_EG_REFERENCE_GOODS foreign key (gid)
117         references goods (id);
118
119 alter table employee
120     add constraint FK_EMPLOYEE_REFERENCE_DEPARTME foreign key (id)
121         references department (id);
122
123 alter table goods
124     add constraint FK_GOODS_REFERENCE_FACTORY foreign key (id)

```

```
125     references factory (id);
126
127 温馨提醒:
128  * PowerDesigner生成SQL脚本时, 对象带有双引号的问题解决
129  * 修改PowerDesigner
```

* PowerDesigner生成SQL脚本时, 对象带有双引号的问题解决

* 参考[13-PowerDesigner生成SQL脚本时, 对象带有双引号的问题解决](#)

* 能够掌握数据库设计范式

* 数据库设计范式的概述

```
1  * 设计关系数据库时, 遵从不同的规范要求, 设计出合理的关系型数据库,
2  这些不同的规范要求被称为不同的范式, 各种范式呈递次规范, 越高的范式数据库冗余越小。
3  * 目前关系数据库有六种范式 (Normal form):
4  * 第一范式 (1NF)
5  * 第二范式 (2NF)
6  * 第三范式 (3NF)
7  * 巴斯-科德范式 (BCNF)
8  * 第四范式 (4NF)
9  * 第五范式 (5NF, 又称完美范式)
10 * 一般来说, 数据库只需满足第三范式 (3NF) 就行了。
11
```

* 第一范式 (1NF)

* 每个字段是原子性不能再分。、

* 不符合第一范式示例

员工编号	员工姓名	联系方式
10001	小黑	xiaohei@163.com ;15915813550
10002	小白	xiaobai@163.com ;15915813551
10001	小黑	xiaohei@163.com ;15915813550

* 存在问题

* 联系方式可以再分，不是原子性

* 符合第一范式的示例

员工编号(pk)	员工姓名	邮箱	手机号码
10001	小黑	xiaohei@163.com	15915813550
10002	小白	xiaobai@163.com	15915813551
10003	小明	xiaoming@163.com	15915813552

* 关于原子性，应该根据业务而定，如联系方式，为开发的便利性，可能会采用一个字段。（反范式）

* 第二范式

* 第二范式是建立在第一范式基础之上，要求有主键，数据库中不能出现重复的记录，要求数据库中所有非主键字段完全依赖主键，不能产生部分依赖；

* 严格意义上来说：尽量不要使用复合主键。

* 不符合第二范式示例

员工编号	员工姓名	商品编号	商品名称
10001	小黑	1001	西游记
10002	小白	1002	红楼梦
10003	小明	1001	西游记
10001	小黑	1002	红楼梦

* 调整一下（联合主键）

员工编号（PK）	商品编号（PK）	员工姓名	商品名称
10001	1001	小黑	西游记
10002	1002	小白	红楼梦
10003	1001	小明	西游记
10001	1002	小黑	红楼梦

* 存在问题

以上虽然确定了主键，但此表会出现大量冗余，主要涉及到的冗余字段为‘员工姓名’，‘商品名称’，

出现原因在于：员工姓名部分依赖了主键的一个字段员工编号，而没有依赖商品编号，而商品名称

部分依赖了主键的一个字段商品编号。

* 解决方案（多对多：弄个中间表，三张表）

员工编号(pk)	员工姓名
10001	小黑
10002	小白
10003	小明

商品编号 (PK)	商品名称
1001	西游记
1002	红楼梦

员工商品关系	员工编号	商品编号
1	10001	1001
2	10002	1002
3	10003	1001
4	10001	1002

* 第三范式

* 第三范式建立在建立在第二范式基础之上，要求非主键字段不能产生传递依赖于主键字段。

* 不符合第三范式示例

员工编号(pk)	员工姓名	部门编号	部门名称
10001	小黑	10	教学部
10002	小白	11	渠道部
10003	小明	12	行政部
10004	小红	11	渠道部

* 存在问题

* 部门名称字段存在冗余，因为部门名称没有直接依赖于主键，部门名称依赖于部门编号，

部门编号依赖于员工编号，那这就是传递依赖。

* 解决方案（一对多关系：部门表独立出来，部门表主键作为员工表的外键）

部门编号	部门名称
10	教学部
11	渠道部
12	行政部

员工编号(pk)	员工姓名	部门编号
10001	小黑	10
10002	小白	11
10003	小明	12
10004	小红	11