

005-多表查询分析方法及复杂查询

可改写的文档：

W Oracle CRUD (第五天) 学习笔记.docx
2016/03/28 15:16, 310.55KB

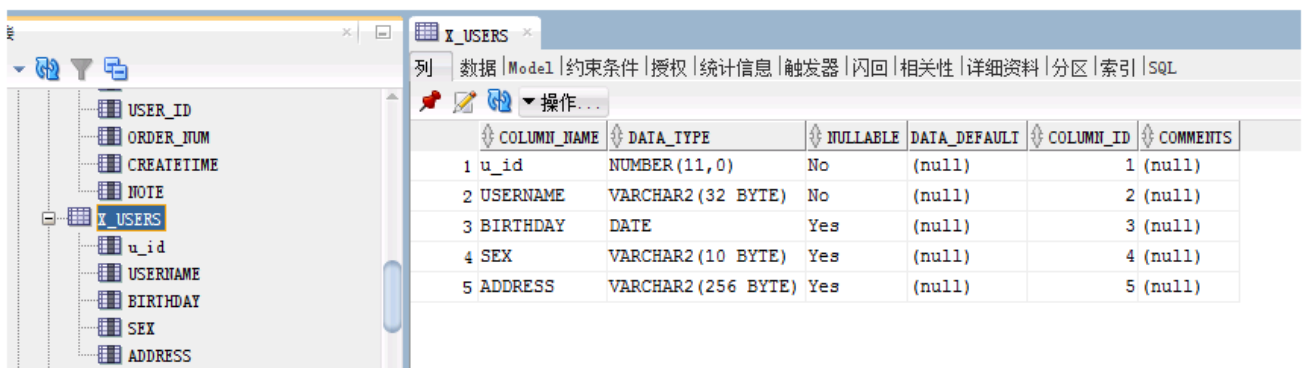
一、***项目中的多表查询***

总体原则：

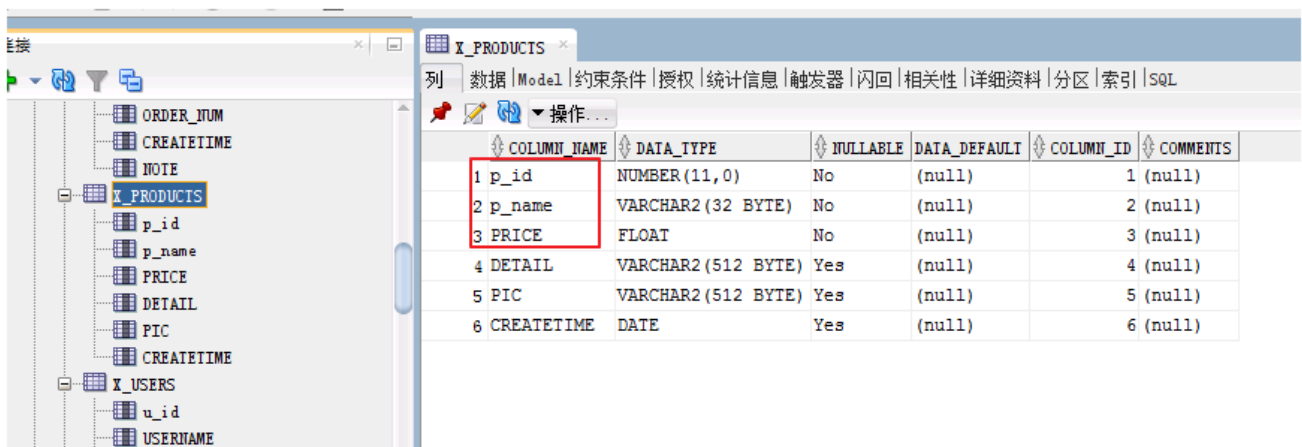
1. 找到与功能相关的表
2. 把主键、非空字段（表中的重要信息一般都是非空的）进行设计
3. 分析有外键关联关系的表之间的ORM（对象关系映射）映射关系：**1:1 1: n m:n**
4. 建立基础数据表【多表查询的结果】
5. 根据需求完成复杂查询【在基础数据表上进行操作】

【网上商城项目---订单功能】

1. 找出相关表：**x_users**（用户表）, **x_products**（商品表）, **x_orders**（订单表）, **x_order_detail**（订单细节表）
2. 列出表的核心字段：



COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 u_id	NUMBER(11,0)	No	(null)	1	(null)
2 USERNAME	VARCHAR2(32 BYTE)	No	(null)	2	(null)
3 BIRTHDAY	DATE	Yes	(null)	3	(null)
4 SEX	VARCHAR2(10 BYTE)	Yes	(null)	4	(null)
5 ADDRESS	VARCHAR2(256 BYTE)	Yes	(null)	5	(null)



COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 p_id	NUMBER(11,0)	No	(null)	1	(null)
2 p_name	VARCHAR2(32 BYTE)	No	(null)	2	(null)
3 PRICE	FLOAT	No	(null)	3	(null)
4 DETAIL	VARCHAR2(512 BYTE)	Yes	(null)	4	(null)
5 PIC	VARCHAR2(512 BYTE)	Yes	(null)	5	(null)
6 CREATETIME	DATE	Yes	(null)	6	(null)

连接	X_ORDERS
	列 数据 Model 约束条件 授权 统计信息 触发器 闪回 相关性 详细资料 分区 索引 SQL
	操作...

连接

+

USERS

X_ORDER_DETAIL

d_id

o_id

p_id

p_num

+

X_ORDERS

+

X_PRODUCTS

+

X_USERS

X_ORDER_DETAIL

列

数据 | Model | 约束条件 | 授权 | 统计信息 | 触发器 | 闪回 | 相关性 | 详细资料 | 分区 | 索引 | SQL

📌

✎

🔗

▼ 操作...

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 d_id	NUMBER(11,0)	No	(null)	1 (null)	
2 o_id	NUMBER(11,0)	Yes	(null)	2 (null)	
3 p_id	NUMBER(11,0)	Yes	(null)	3 (null)	
4 p_num	NUMBER(11,0)	Yes	(null)	4 (null)	

1. 根据核心字段，把有外键关联关系的表之间的ORM关系明确

The screenshot shows the '我的订单' (My Orders) page on JD.com. Red arrows originate from the '全部订单' (All Orders) link in the left sidebar and point to specific order entries in the main list. These arrows illustrate the ORM relationships between the 'X_ORDER_DETAIL' table and the 'X_ORDERS' table. The arrows point to the '订单号' (Order Number) field in each order entry, which corresponds to the 'o_id' column in the 'X_ORDER_DETAIL' table. The order entries shown are:

- Order 11732028835: 天王星 (Telesonic) 挂钟14寸时尚创意客厅田园钟米静音简约时钟 Q3644-...
- Order 11233549559: 辽宁移动手机话费充值100元 快充
- Order 11005935438: 小欢园玩具专营店

2015-09-06 23:12:58 订单号: 10032485441

黄山易酷营养保健专营...



修正辅酶Q10维生素E软胶囊 增强免疫力 1瓶(60粒)
找搭配

x2

返修/退换货



总额 ¥316.00
在线支付

已完成
订单详情

2015-08-13 12:24:41 订单号: 9885657513

您订单中的商品在不同库房或属不同商家,故拆分为以下订单分开配送,给您带来的不便敬请谅解...

收货人: 于春芳 订单金额: ¥697.40

支付方式: 在线支付

订单状态: 已拆分

查看拆分详情>

2015-08-13 12:24:41 订单号: 9885686050

京东



美的(Midea) 电磁炉匀火科技纤薄触屏弧形滑控RH2121 (赠304不锈钢汤
找搭配

x1

返修/退换货



总额 ¥618.40
在线支付

已完成
订单详情



立客(LIKE) LK-S1509 全钢电水壶 自动断电电水壶 1.5L
找搭配

x1

返修/退换货

2015-08-13 12:24:41 订单号: 9885686054

京东



纽曼(Newman) V5 移动/联通2G 老人手机 金色
找搭配

x1

返修/退换货



总额 ¥79.00
在线支付

已完成
订单详情

立即购买

2015-06-18 22:13:25 订单号: 9552208171

京东



格兰仕(Galanz) 微波炉/光波炉 G80F20CN2L-B8 (RO) 20L
找搭配

x1

返修/退换货



总额 ¥467.00
在线支付

已完成
订单详情

立即购买

2015-06-07 14:55:36 订单号: 9466881899

京东



希捷(Seagate) Backup Plus睿品 (升级版) 1T 2.5英寸 USB3.0移动硬盘

x1

返修/退换货



总额 ¥425.00
在线支付

已完成
订单详情

立即购买

2015-08-13 14:05:51 您的订单已经打印完毕

2015-08-13 12:26:31 您的订单预计8月14日送达您手中

2015-08-13 12:26:30 您的订单已经进入京东2号库准备出库

2015-08-13 12:24:41 您提交了订单,请等待系统确认

订单明细

收货人信息

收货人:
地址:

手机号码:



配送信息

配送方式: 普通快递
运费: ¥0.00
送货日期: 2015-08-14

付款信息

付款方式: 在线支付
商品总额: ¥618.90
应支付金额: ¥618.40
运费金额: ¥0.00
优惠券: ¥0.00
返现: ¥0.00
礼品卡: ¥0.00
京豆: ¥0.50
订单优惠: ¥0.50

发票信息

发票类型: 普通发票
发票抬头: 个人
发票内容: 明细

商品

商品编号

京东价

商品数量

京豆

操作



美的(Midea) RH2121 高科技匀火至尊超薄触屏弧形滑控电磁炉 (赠食品级304不锈钢汤锅)

1074271

¥599.00

1

0

申请返修/退换货



立客(LIKE) LK-S1509 全钢电水壶 自动断电电水壶 1.5L

1470508

¥19.90

1

0

申请返修/退换货

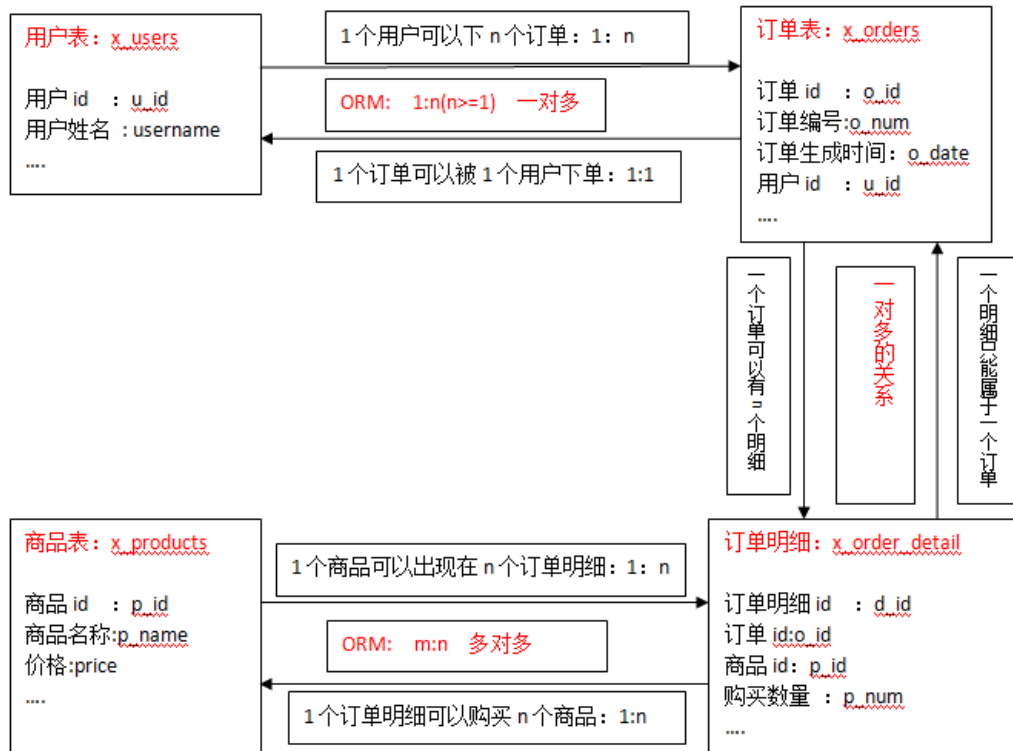
立即购买

产品列表

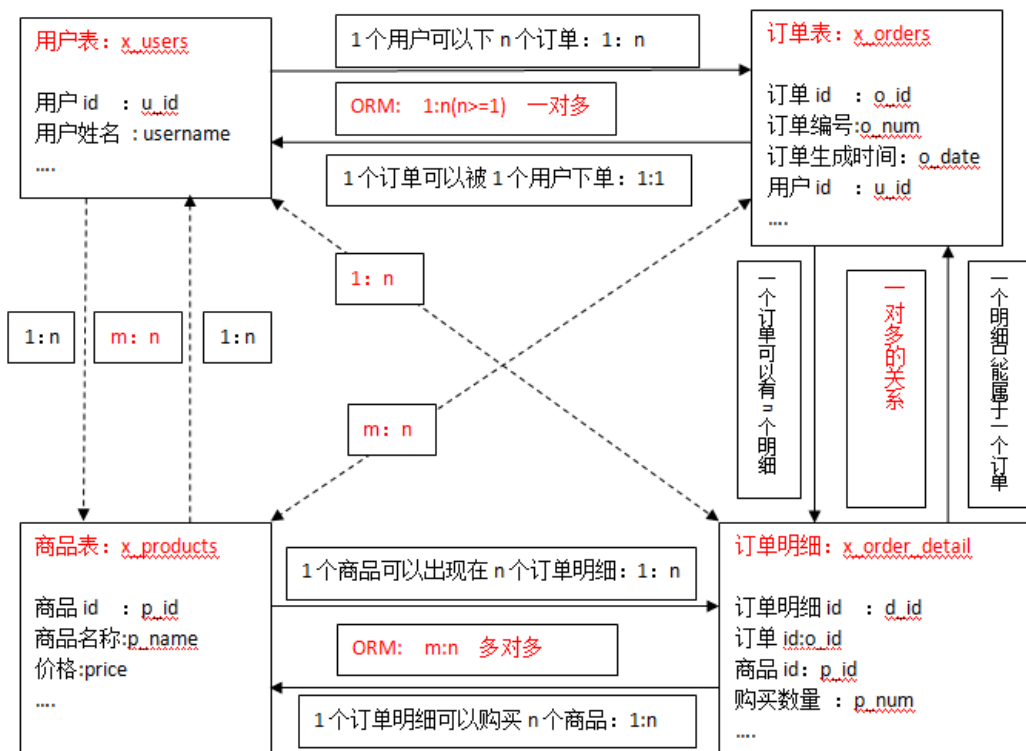
商品总额: ¥618.90

返 现: -¥0.00

京 豆: -¥0.50



1. 根据上图分析没有外键关联关系的表之间ORM映射



例如: **x_users**表和**x_products**表之间的关系

5-根据图表分析的结果, 可以写出任何复杂的查询语句, 按步骤如下:

a)实现4张有外键关联表的多表查询(注意: 必须规避笛卡尔乘积), 得到【基础数据总表】

注意: 在Oracle中, 默认的表名和字段名均为大写, 如果通过sqldeveloper修改了列名为小写, 则引用该列时, 需要加上双引号, 例如: **X_USER.USERNAME** 和

X_USER."username" 在oracle中均是有效的字段命名。

```
--users and orders
SELECT x_users.*,
x_orders.*
FROM x_users,
x_orders
WHERE x_users."u_id"=x_orders."user_id";
--users and order_detail
SELECT x_users.*,
x_orders.*,
x_order_detail.*
FROM x_users,
x_orders,
x_order_detail
WHERE x_users."u_id"=x_orders."user_id"
AND x_orders."o_id" =x_order_detail."o_id";
--users and products
SELECT x_users.*,
x_orders.*,
x_order_detail.*,
x_products.*
FROM x_users,
x_orders,
x_order_detail,
x_products
WHERE x_users."u_id" =x_orders."user_id"
AND x_orders."o_id" =x_order_detail."o_id"
AND x_order_detail."p_id"=x_products."p_id";
```

b)对【基础数据总表】进行复杂查询

```

--查询用户admin曾经购买过什么商品？
SELECT DISTINCT x_products."p_name" AS admin_buy_products
FROM x_users,
x_orders,
x_order_detail,
x_products
WHERE x_users."u_id" =x_orders."user_id"
AND x_orders."o_id" =x_order_detail."o_id"
AND x_order_detail."p_id"=x_products."p_id"
AND x_users.USERNAME = 'admin';
--请统计admin用户购买过的商品种类？
SELECT COUNT(DISTINCT x_products."p_name") AS admin_buy_products
FROM x_users,
x_orders,
x_order_detail,
x_products
WHERE x_users."u_id" =x_orders."user_id"
AND x_orders."o_id" =x_order_detail."o_id"
AND x_order_detail."p_id"=x_products."p_id"
AND x_users.USERNAME = 'admin';
--请统计用户们在我们的电子商务网站上各自总共消费了多少钱？
SELECT username,
SUM(totalMoney)
FROM
(SELECT x_users.USERNAME,
x_products.PRICE*x_order_detail."p_num" AS totalMoney
FROM x_users,
x_orders,
x_order_detail,
x_products
WHERE x_users."u_id" =x_orders."user_id"
AND x_orders."o_id" =x_order_detail."o_id"
AND x_order_detail."p_id"=x_products."p_id"
)
GROUP BY username;

```

二、***关于连接查询***

1-内连接

```

SELECT x_users.*,
x_orders.*
FROM x_users,
x_orders
WHERE x_users."u_id"=x_orders."user_id";
--INNER JOIN
SELECT x_users.*,
x_orders.*
FROM x_users
INNER JOIN x_orders
ON x_users."u_id"=x_orders."user_id";

```

	u_id	USERNAME	BIRTHDAY	SEX	ADDRESS	o_id	user_id	o_num	o_date	NOTE
1	1001	admin	09-3月 -16	male	dalian	2001	1001	JD_0001	01-3月 -16	baozhuang
2	1001	admin	09-3月 -16	male	dalian	2002	1001	JD_0002	08-3月 -16	gift for girl
3	1002	user	01-3月 -16	female	shanghai	2003	1002	JD_0003	09-2月 -16	fruit
4	1002	user	01-3月 -16	female	shanghai	2004	1002	JD_0004	29-2月 -16	cloths

2-交叉连接（笛卡尔乘积）

```
--CROSS JOIN
SELECT x_users.*,
x_orders.*
FROM x_users
CROSS JOIN x_orders;
--等价于笛卡尔乘积
SELECT * FROM x_users,x_orders;
```

	u_id	USERNAME	BIRTHDAY	SEX	ADDRESS	o_id	user_id	o_num	o_date	NOTE
1	1001	admin	09-3月 -16	male	dalian	2001	1001	JD_0001	01-3月 -16	baozhuang
2	1001	admin	09-3月 -16	male	dalian	2002	1001	JD_0002	08-3月 -16	gift for girl
3	1001	admin	09-3月 -16	male	dalian	2003	1002	JD_0003	09-2月 -16	fruit
4	1001	admin	09-3月 -16	male	dalian	2004	1002	JD_0004	29-2月 -16	cloths
5	1002	user	01-3月 -16	female	shanghai	2001	1001	JD_0001	01-3月 -16	baozhuang
6	1002	user	01-3月 -16	female	shanghai	2002	1001	JD_0002	08-3月 -16	gift for girl
7	1002	user	01-3月 -16	female	shanghai	2003	1002	JD_0003	09-2月 -16	fruit
8	1002	user	01-3月 -16	female	shanghai	2004	1002	JD_0004	29-2月 -16	cloths

3-外连接

左外连接：

```
--LEFT OUTER JOIN...ON...
SELECT *
FROM x_users users,
x_orders orders
WHERE users."u_id"=orders."user_id"(+);
```

```
SELECT *
FROM x_users users
LEFT OUTER JOIN x_orders orders
ON users."u_id"=orders."user_id";
```

为什么使用左外连接？

--外连接只发送一次sql请求，子查询需要至少发送两次请求，所以，外连接能够提高sql执行效率

--例如：请查询admin用户的订单信息？

--原来使用的是子查询的方式

```
SELECT x_orders.*
FROM x_orders
WHERE "user_id"=
(SELECT "u_id" FROM x_users WHERE x_users.USERNAME='admin'
);
```

--请使用左外连接的方式修改上面的sql语句

```
SELECT *
FROM x_users users,
x_orders orders
WHERE users."u_id"=orders."user_id" (+) and users.USERNAME='admin';
```

--但凡是子查询，我们都可以使用外连接的方式进行实现，从而提高程序效率

右外连接

右表：x_products（产品）左表：x_order_detail（订单详情）

--RIGHT OUTER JOIN...ON

SELECT *

```

SELECT ^
FROM x_products products,
x_order_detail detail
WHERE detail."p_id"(+)=products."p_id";
SELECT products.*,
detail.*
FROM x_order_detail detail
RIGHT OUTER JOIN x_products products
ON detail."p_id"=products."p_id";

```

演示问题：

--查询2001号订单中orange的购买数量？

--子查询实现

```

SELECT "p_num"
FROM x_order_detail
WHERE "o_id"=2001
AND "p_id" =
(SELECT "p_id" FROM x_products WHERE "p_name"='orange'
);

```

--右外连接实现

```

SELECT detail."p_num"
FROM x_products products,
x_order_detail detail
WHERE detail."p_id"(+)=products."p_id" and detail."o_id"=2001 and products."p_
name"='orange';
--right outer join on的改写方法

```

查询结果 x

SQL | 提取的所有行: 5, 用时 0.009 秒

	u_id	USERNAME	BIRTHDAY	SEX	ADDRESS	o_id	user_id	o_num	o_date	NOTE
1	1001	admin	09-3月 -16	male	dalian	2001	1001	JD_0001	01-3月 -16	baozhuang
2	1001	admin	09-3月 -16	male	dalian	2002	1001	JD_0002	08-3月 -16	gift for girl
3	1002	user	01-3月 -16	female	shanghai	2003	1002	JD_0003	09-2月 -16	fruit
4	1002	user	01-3月 -16	female	shanghai	2004	1002	JD_0004	29-2月 -16	cloths
5	1003	tianyu	24-3月 -16	female	beijing	(null)	(null)	(null)	(null)	(null)

```

SELECT detail."p_num"
FROM x_products products,
x_order_detail detail
WHERE detail."p_id" (+)=products."p_id" and detail."o_id"=2001 and products."p_name"='orange';

```

查询结果 x

SQL | 提取的所有行: 1, 用时 0.005 秒

p_num
54

```

SELECT detail."p_num"
FROM x_order_detail detail
RIGHT OUTER JOIN x_products products
ON (detail."p_id"=products."p_id" and detail."o_id"=2001 and products."p_name"='orange');

```

查询结果 x

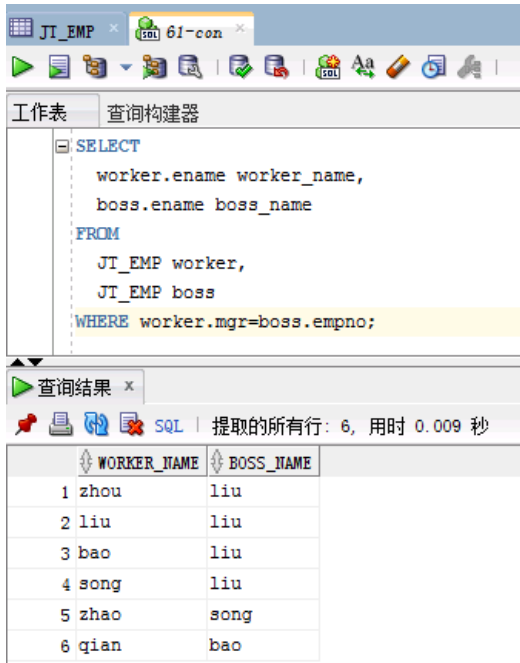
SQL | 提取的所有行: 5, 用时 0.004 秒

p_num
54
(null)
(null)
(null)
(null)

4-自连接

请查询：【员工姓名 上级姓名】

```
SELECT
worker.ename worker_name,
boss.ename boss_name
FROM
JT_EMP worker,
JT_EMP boss
WHERE worker.mgr=boss.empno;
```



工作表 查询构建器

```
SELECT
worker.ename worker_name,
boss.ename boss_name
FROM
JT_EMP worker,
JT_EMP boss
WHERE worker.mgr=boss.empno;
```

查询结果 x

提取的所有行: 6, 用时 0.009 秒

	WORKER_NAME	BOSS_NAME
1	zhou	liu
2	liu	liu
3	bao	liu
4	song	liu
5	zhao	song
6	qian	bao

5-不等连接

不等连接：> < !=或者 (<>)，左表中的每一条记录与右表中的记录一条条比较

--求所有人员的工资等级

```
SELECT emp.*,
salgrade.*
FROM JT_EMP emp
INNER JOIN JT_SALGRADE salgrade
ON emp.sal BETWEEN salgrade.minsal AND salgrade.maxsal;
```

```
SELECT emp.*,
salgrade.*
FROM JT_EMP emp
INNER JOIN JT_SALGRADE salgrade
ON emp.sal >salgrade.minsal
AND emp.sal<salgrade.maxsal;
```