

关卡 2

数据库设计与优化



华为技术有限公司

数据库设计与优化

作业说明：此关卡作业共 100 分，提交作业命名规则“关卡二_姓名”

Part1 《鲲鹏创新实践课：鲲鹏应用数据分析与管理实战 —— 数据库设计与优化》理论作业

Q1:

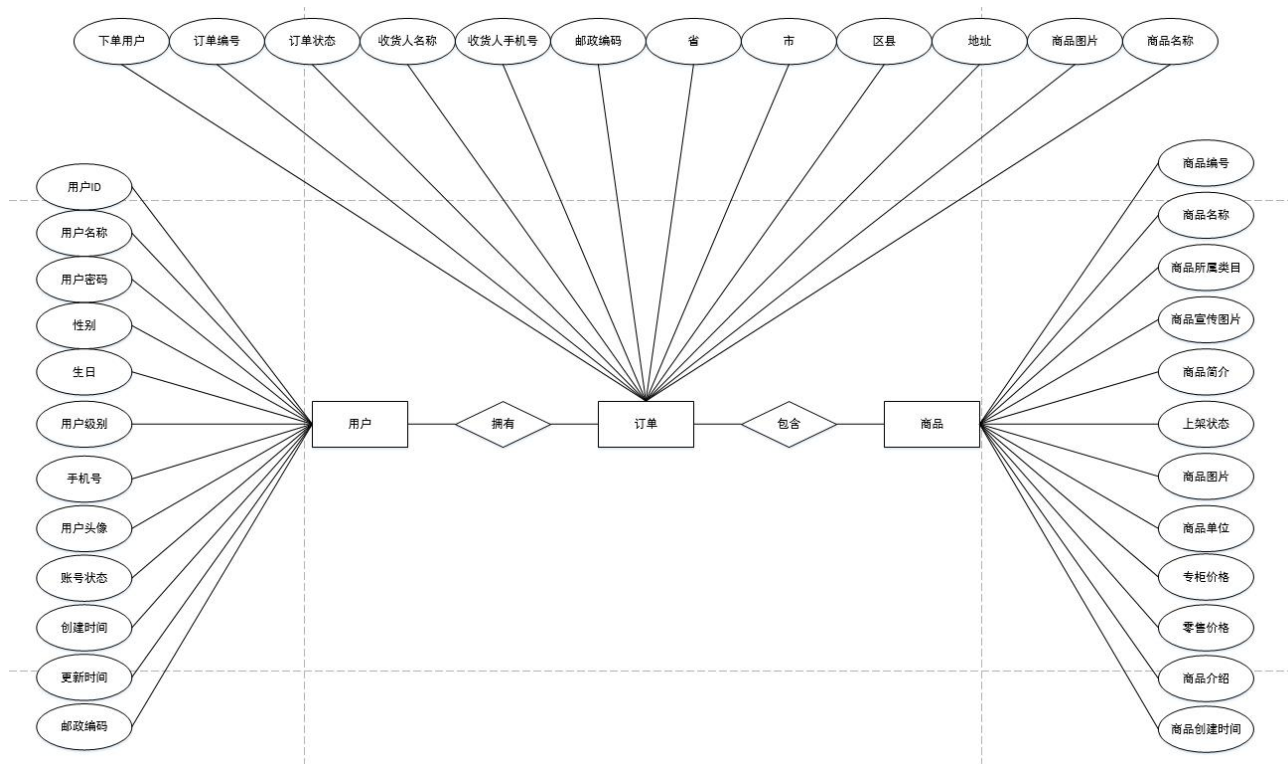
思考：在电商平台中，订单交易流程的数据库设计需要收集和分析哪些信息为后续做准备？（5 分）

业务流情况如完成一笔订单完整的交易流程、商品上架的流程、退款流程等，以及各类指标如新增访问人数、注册人数、订单数量、买家评价数、好评率等。

Q2:

概念设计和概念模型 E-R 图（10 分）

将画好的 E-R 图，复制粘贴到下方：



Q3:

数据库表设计（15 分）

步骤一：从用户下单购买商品的订单中提取属性列（3 分）

可提取的属性列表：

订单编号、订单状态、订单用户、用户留言、收货信息（收货人、手机号、地址）、商品信息（商品名称、商品编号、货品规格、货品价格、货品数量、货品图片）、费用信息（实际费用、商品总价、快递费用、优惠减免、积分减免）、支付信息（支付渠道、支付时间）、快递信息

（快递公司、快递单号、发货时间）、退款信息（退款金额、退款类型、退款备注、退款时间）、收货信息（确认收货时间）。

步骤二：对内容进行三范式处理（4 分，各 1 分）

订单表：

订单编号	订单状态	订单用户
PK		

地址表：

收货人	省	市	区县	详细地址
PK				

商品表：

商品名称	商品编号	货品规格	货品价格	货品数量	货品图片
PK					

用户表：

订单用户	用户留言
PK	

步骤三：反范式及物理模型设计，输出最终表设计结果（8 分，各 2 分）

基于步骤 2，设计 4 张表的字段名称、字段类型、约束，并以表格形式呈现。

订单表：

字段名称	字段类型	约束
订单编号	CHAR	主键约束、非空约束
订单状态	INT	非空约束
订单用户	VARCHAR	非空约束

地址表：

字段名称	字段类型	约束
收货人	VARCHAR	主键约束、非空约束
省	VARCHAR	非空约束
市	VARCHAR	非空约束
区县	VARCHAR	非空约束
详细地址	VARCHAR	非空约束

商品表：

字段名称	字段类型	约束
商品名称	VARCHAR	主键约束、非空约束
商品编号	CHAR	非空约束
货品规格	VARCHAR	非空约束
货品价格	DOUBLE	非空约束
货品数量	INT	非空约束
货品图片	VARCHAR	

用户表：

字段名称	字段类型	约束
订单用户	VARCHAR	主键约束、非空约束
用户留言	VARCHAR	

Part2 《关卡 2-1：基于 Litemall 商城的华为云数据库开发》作业

任务一：完成新商品类目建立（7 分）

作业提交如下：

1. 新增商品一级类目（3 分，每空 1 分）：

```
INSERT INTO `litemall_category` VALUES (2006000,'其它','其它产品',0,'https://res.vmallres.com/pimages//product/6944047805713/428_428_D99DAD5967499AE56B2B0001AF623B8C625AF5A3B2ADADA4mp.png','L1',6,'2020-03-01 00:00:00','2020-03-01 00:00:00',0);
```

2. 新增商品二级类目（3 分，每空 1 分）

```
INSERT INTO `litemall_category` VALUES (2006100,'牙刷','智能声波牙刷',2006000,'https://res.vmallres.com/pimages//product/6944047805720/800_800_3FE6A054A258D2B5109D0F89AD95F795F379F399087614E9mp.png','https://res.vmallres.com/pimages//product/6944047805720/800_800_3FE6A054A258D2B5109D0F89AD95F795F379F399087614E9mp.png','L2',1,'2020-03-01 00:00:00','2020-03-01 00:00:00',0);
```

3. 新添加商品类目截图（1 分）



任务二：完成新商品的上架（11 分）

1. 完成如下语法并运行（6 分，每空 1 分）

```
INSERT INTO `litemall_goods` VALUES (2006101,'2006101','HUAWEI 智能声波牙刷',2006100,0,['https://res.vmallres.com/pimages/detailImg/2020/03/17/005EA094AF3AACEAEBA2ACFDCA3648B7A1B255CBE1A1974D.jpg'],'优漾 珊瑚粉（支持 HUAWEI HiLink）支持 HUAWEI HiLink',1,23,'https://res.vmallres.com/pimages//product/6944047805713/428 428 D99DAD5967499AE56B2B0001AF623B8C625AF5A3B2ADADA4mp.png',1,1,'件',169,169,'<p><img src=\"https://res.vmallres.com/pimages/detailImg/2020/03/17/007C4FD3D63338A9E5A3DF002DAC1778F8C689DCDC039D0C.jpg\" _src=\"https://res.vmallres.com/pimages/detailImg/2020/03/17/007C4FD3D63338A9E5A3DF002DAC1778F8C689DCDC039D0C.jpg\" style=\"\"/></p><p><img src=\"https://res.vmallres.com/pimages/detailImg/2020/03/17/8592F23F305BFC6D002ED06ED031DD776A3BF929D5D9DB12.jpg\" _src=\"https://res.vmallres.com/pimages/detailImg/2020/03/17/8592F23F305BFC6D002ED06ED031DD776A3BF929D5D9DB12.jpg\" style=\"\"/></p>',2020-04-01 00:00:00,2020-04-01 00:00:00,0);
```

```
INSERT INTO `litemall_goods_attribute` VALUES (1019,2006101,'产品尺寸','28*28.8*191mm（不含刷头高度）',2020-04-01 00:00:00,2020-04-01 00:00:00,0);
```

```
INSERT INTO `litemall_goods_product` VALUES (330,2006101,['标准'],169.00,20,'https://res.vmallres.com/pimages//product/6944047805713/428 428 D99DAD5967499AE56B2B0001AF623B8C625AF5A3B2ADADA4mp.png',2020-04-01 00:00:00,2020-04-01 00:00:00,0);
```

2. 新增商品手机应用端查看截图：（1 分）

牙刷

牙刷


智能声波牙刷



HUAWEI 智能声波牙刷
优漾 珊瑚粉 (支持HUAWEI HiLink) 支持HUAWEI HiLink

¥169 ~~¥169~~

没有更多了



Lebooo

力博得智能声波牙刷·优漾
智能洁净亮白 健康长效护龈

华为HiLink 智生活 | 38000次/分钟高频震动 智能清洁
护龈按摩 有效性除牙菌斑 | 5h快充 90天长续航

HUAWEI HiLink 生态产品

¥169.00 ~~¥169.00~~
HUAWEI 智能声波牙刷
优漾 珊瑚粉 (支持HUAWEI HiLink) 支持HUAWEI HiLink

规格	请选择 >
属性	>
运费	满88免邮费

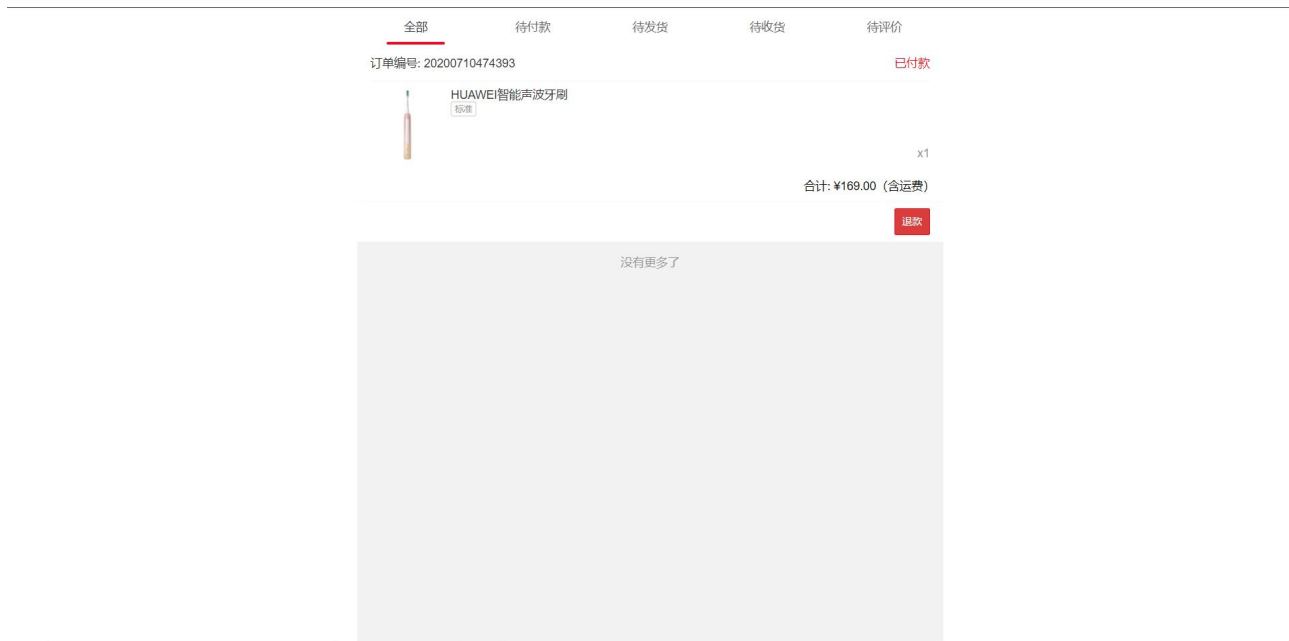
商品详情



加入购物车

立即购买

3. 用自己账号完成一次新商品下单，并附上订单截图（可在订单中查看）：（1 分）



4. 完成如下语句，并附上商品价格修改后截图（3 分，各 1 分）


```
UPDATE litemall_goods SET counter_price=159, retail_price=159 WHERE id =2006101;  
UPDATE litemall_goods_product SET price=159 WHERE goods_id=2006101;
```

截图：

牙刷

牙刷


智能声波牙刷



HUAWEI 智能声波牙刷
优漾 珊瑚粉 (支持HUAWEI HiLink) 支持HUAWEI HiLink
¥159 ~~¥169~~

没有更多了

Lebooo



力博得智能声波牙刷·优漾
智能洁净亮白 健康长效护龈
华为HiLink 智生活 | 38000次/分钟高频震动 智能清洁
护龈按摩 有效祛除牙菌斑 | 5h快充 90天续航

HUAWEI HiLink 生态产品

¥159.00 ~~¥169.00~~
HUAWEI 智能声波牙刷
优漾 珊瑚粉 (支持HUAWEI HiLink) 支持HUAWEI HiLink

规格 请选择 >

属性 >

运费 满88免邮费

商品详情

加入购物车

立即购买

任务三：数据库优化（4 分）

1. 优化前，执行计划结果截图（1 分）

```
1 SELECT * FROM litemall_order WHERE order_sn = '20280438326717';
```

SQL执行记录 消息 结果集1 × 执行计划2 × 添加模式

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	litemall_order	ALL					59830	10.0	Using where

2. 完成优化代码（2 分，每空各 1 分）

```
ALTER TABLE litemall_order ADD INDEX index_ordersn (order_sn);
```

3. 优化后，执行计划结果截图（1 分）

1

SELECT * FROM litemall_order WHERE order_sn = '20200430326717';

SQL执行记录

消息

结果集1 X

执行计划2 X

结果集3 X

结果集4 X

结果集5 X

结果集6 X

结果集7 X

追加模式

执行时间	SQL语句	消耗时间	执行结果
2020-07-10 00:32:57	SELECT * FROM litemall_order WHERE order_sn = '20200430326717'	5ms	执行成功

1 SELECT * FROM litemall_order WHERE order_sn = '20200430326717';

SQL执行记录

消息

执行计划1 X

追加模式

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	litemall_order	ref	index_ordersn	index_ordersn	254	const	1	100.0	

课堂挑战：

挑战一：查询某一用户在某段时间内的购买记录（9 分）

1. 查询语句及结果截图（3 分）

查询语句举例：SELECT XXX FROM XXX WHERE XXX

此处填入语句：

-- 删除 order_sn 索引

ALTER TABLE litemall_order DROP INDEX index_ordersn;

-- 优化前查询

SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371';

-- 给 user_id 建立索引并查询

ALTER TABLE litemall_order ADD INDEX index_userid (user_id);

SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371';

-- 给 order_sn 建立索引并查询

ALTER TABLE litemall_order DROP INDEX index_userid;

ALTER TABLE litemall_order ADD INDEX index_ordersn (order_sn);

SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371';

-- 给 add_time 建立索引并查询

ALTER TABLE litemall_order DROP INDEX index_ordersn;

ALTER TABLE litemall_order ADD INDEX index_addtime (add_time);

SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371';

```
1 SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371';
```

SQL执行记录 消息 结果集2 X 执行计划3 X 结果集4 X 结果集5 X 结果集6 X 结果集7 X

以下是SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371'的执行结果

① 主键列未全部包含在查询结果中，因此无法进行编辑、导出SQL操作

	user_id	order_sn	add_time
1	371	20200405237256	2020-04-05 16:23:17

优化前执行用时 26ms，分别给 user_id, order_sn, add_time 添加索引优化后用时为 4ms, 27ms, 27ms，因此最优方案为给 user_id 添加索引，此时扫描行数 rows 最小，效率最高。

2. 优化前，执行计划结果截图（3 分）

```
1 SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371';
```

SQL执行记录 消息 执行计划3 X 执行计划8 X

	id	select_type	table	partitions	type	possible_keys	key	key_len
1	1	SIMPLE	litemall_order		ALL			
ref	rows	filtered	Extra					
	59830	1.11	Using where					

3. 优化后，执行计划结果截图（3 分）

-- 给 user_id 建立索引

```
1 SELECT user_id, order_sn, add_time FROM litemall_order WHERE add_time>'2020-03-31' and add_time<'2020-04-15' and user_id='371';
```

SQL执行记录 消息 结果集2 X 执行计划3 X 结果集4 X 结果集5 X 结果集6 X 结果集7 X 执行计划8 X

	id	select_type	table	partitions	type	possible_keys	key	key_len
1	1	SIMPLE	litemall_order		ref	index_userid	index_userid	4
ref	rows	filtered	Extra					
const	4	11.11	Using where					

-- 给 order_sn 建立索引

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref
1	1	SIMPLE	litemall_order		ALL				
rows	filtered	Extra							
59830	1.11	Using where							

-- 给 add_time 建立索引

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref
1	1	SIMPLE	litemall_order		range	index_addtime	index_addtime	6	
rows	filtered	Extra							
7568	10.0	Using index condition;...							

挑战二：查询某个用户在平台上所买的所有的商品名称（9 分）

1. 查询语句及结果截图（3 分）

查询语句举例：SELECT XXX FROM XXX WHERE XXX

此处填入语句：

-- 优化前

SELECT consignee, goods_name FROM litemall_order, litemall_order_goods WHERE consignee = '杭高峰';

-- 对 consignee 添加索引优化后

ALTER TABLE litemall_order ADD INDEX index_consignee (consignee);

SELECT consignee, goods_name FROM litemall_order, litemall_order_goods WHERE consignee = '杭高峰';

```
1 SELECT consignee, goods_name FROM litemall_order, litemall_order_goods WHERE consignee = '杭高峰';
```

SQL执行记录 消息 结果集1 X
追加模式

以下是SELECT consignee, goods_name FROM litemall_order, litemall_order_goods WHERE ...的执行结果集

① 查询结果集中包含了多个表，不能进行编辑、导出SQL操作

复制行
复制列
列设置

	consignee	goods_name
1	杭高峰	HUAWEI P30
2	杭高峰	HUAWEI P30
3	杭高峰	HUAWEI P30
4	杭高峰	HUAWEI P30
5	杭高峰	HUAWEI nova 6 SE
6	杭高峰	HUAWEI nova 6 SE
7	杭高峰	HUAWEI nova 6 SE

因为我们要查找 consignee 为杭高峰的 goods_name，所以对 consignee 添加索引优化后，查询时间从 22ms 变成了 4ms。

2. 优化前，执行计划结果截图（3 分）

	id	select_type	table	partitions	type
1	1	SIMPLE	litemall_order		ALL
2	1	SIMPLE	litemall_order_goods		ALL

possible_keys	key	key_len	ref	rows	filtered
				59830	10.0
				62793	100.0

Extra

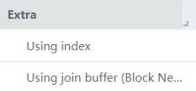
Using where

Using join buffer (Block Nes...

3. 优化后，执行计划结果截图（3 分）

	id	select_type	table	partitions	type
1	1	SIMPLE	litemall_order		ref
2	1	SIMPLE	litemall_order_goods		ALL

possible_keys	key	key_len	ref	rows	filtered
index_consignee	index_consignee	254	const	4	100.0
				62793	100.0



挑战三：在 litemall 上新增一个类目并上架一个商品（14 分，每一问各 2 分）

1. 新建一级类目代码：（2 分）

```
INSERT INTO `litemall_category` VALUES (2007000,'视频','——华为影视有限公司——',0,'https://tse4-mm.cn.bing.net/th/id/OIP.NsSj1k5roiagGzJkzr8WVAHaHa?pid=Api&rs=1','http://a2.att.hudong.com/60/30/01300542749023141378303950564.jpg','L1',7,'2020-07-12 00:00:00','2020-07-12 00:00:00',0);
```

2. 新建一级类目代码截图：（2 分）



3. 新建二级类目代码：（2 分）

```
INSERT INTO `litemall_category` VALUES (2007100,'鲲鹏','','TaiShan',2007000,'https://img.huahuo.com/hulianwang/2020-01-07/e292b12a7036af3030dac9c22
```

929a0c4.png','http://s.laoyaoba.com/jwImg/news/2019/09/04/15675864839403.png','L2',1,'2020-07-12 00:00:00','2020-07-12 00:00:00',0);

4. 新建二级类目截图：（2 分）



5. 新增商品代码（涉及三张表）：（2 分）

```
INSERT INTO `litemall_goods` VALUES (2007101,'2007101','HUAWEI TaiShan 服务器',2007100,0,['https://gd1.alicdn.com/imgextra/i1/2890642103/O1CN01Xljzo01RPDWwMKp1v_!!2890642103.jpg'],,"新一代数据中心服务器，基于华为鲲鹏处理器，适合为大数据、分布式存储和数据库等应用高效加速',1,100,'https://tse1-mm.cn.bing.net/th/id/OIP.IMTlj4AbG9399BZwhDiHAAHaEn?pid=Api&rs=1',",1,1,'件',16000,16000,'<p><img src=\"https://img.alicdn.com/imgextra/i4/2983470940/O1CN01RhrxcO1IoYuL9RUQd_!!2983470940.jpg\" _src=\"\" style=\"\"/></p><p><img src=\"https://img.alicdn.com/imgextra/i1/2983470940/O1CN014d510z1IoYuL92JYE_!!2983470940.jpg\" _src=\"\" style=\"\"/></p>','2020-07-12 00:00:00','2020-07-12 00:00:00',0);
```

```
INSERT INTO `litemall_goods_attribute` VALUES (1020,2007101,'硬盘容量',  
'1000GB','2020-07-12 00:00:00','2020-07-12 00:00:00',0);
```

```
INSERT INTO `litemall_goods_product` VALUES (331,2007101,['2U 机架',  
'\'],16000.00,120,'https://gd3.alicdn.com/imgextra/i2/853706716/O1CN01FqtSGQ1zTyVMwO1gi_!!  
853706716.jpg','2020-07-12 00:00:00','2020-07-12 00:00:00',0);
```

6. 新增商品截图：（2 分）

鲲鹏

鲲鹏

TaiShan



HUAWEI TaiShan服务器

新一代数据中心服务器，基于华为鲲鹏处理器，适合为大数据、分布式存储和数据库等应...

¥16000. ~~¥16000~~

没有更多了



¥16000.00 ~~¥16000.00~~

HUAWEI TaiShan服务器

新一代数据中心服务器，基于华为鲲鹏处理器，适合为大数据、分布式存储和数据库等应用高效加速

规格 请选择 >

属性 >

运费 满88免邮费

商品详情



加入购物车

立即购买

7. 完成新增商品订单截图：（2 分）

全部

待付款

待发货

待收货

待评价

订单编号: 20200712143349

已付款



HUAWEI TaiShan服务器



2U机架

x1

合计: ¥16000.00 (含运费)

退款

没有更多了

Part3. 《Litmall 数据库优化》

任务: 使用 HASH 分区对订单表进行分区 (4 分)

完成如下 SQL 语句填空: (3 分, 每空 1 分)

```
CREATE TABLE `litmall_order` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `user_id` int(11) NOT NULL COMMENT '用户表的用户 ID',  
  `order_sn` varchar(63) NOT NULL COMMENT '订单编号',  
  `order_status` smallint(6) NOT NULL COMMENT '订单状态',  
  `aftersale_status` smallint(6) DEFAULT '0' COMMENT '售后状态, 0 是可申请, 1 是用户已申请,  
  2 是管理员审核通过, 3 是管理员退款成功, 4 是管理员审核拒绝, 5 是用户已取消',
```

```
`consignee` varchar(63) NOT NULL COMMENT '收货人名称',
`mobile` varchar(63) NOT NULL COMMENT '收货人手机号',
`address` varchar(127) NOT NULL COMMENT '收货具体地址',
`message` varchar(512) NOT NULL DEFAULT '' COMMENT '用户订单留言',
`goods_price` decimal(10,2) NOT NULL COMMENT '商品总费用',
`freight_price` decimal(10,2) NOT NULL COMMENT '配送费用',
`coupon_price` decimal(10,2) NOT NULL COMMENT '优惠券减免',
`integral_price` decimal(10,2) NOT NULL COMMENT '用户积分减免',
`groupon_price` decimal(10,2) NOT NULL COMMENT '团购优惠价减免',
`order_price` decimal(10,2) NOT NULL COMMENT '订单费用, = goods_price + freight_price - coupon_price',
`actual_price` decimal(10,2) NOT NULL COMMENT '实付费用, = order_price - integral_price',
`pay_id` varchar(63) DEFAULT NULL COMMENT '微信支付编号',
`pay_time` datetime DEFAULT NULL COMMENT '微信支付时间',
`ship_sn` varchar(63) DEFAULT NULL COMMENT '发货编号',
`ship_channel` varchar(63) DEFAULT NULL COMMENT '发货快递公司',
`ship_time` datetime DEFAULT NULL COMMENT '发货开始时间',
`refund_amount` decimal(10,2) DEFAULT NULL COMMENT '实际退款金额, (有可能退款金额小于实际支付金额)',
`refund_type` varchar(63) DEFAULT NULL COMMENT '退款方式',
`refund_content` varchar(127) DEFAULT NULL COMMENT '退款备注',
`refund_time` datetime DEFAULT NULL COMMENT '退款时间',
`confirm_time` datetime DEFAULT NULL COMMENT '用户确认收货时间',
`comments` smallint(6) DEFAULT '0' COMMENT '待评价订单商品数量',
`end_time` datetime DEFAULT NULL COMMENT '订单关闭时间',
`add_time` datetime DEFAULT NULL COMMENT '创建时间',
`update_time` datetime DEFAULT NULL COMMENT '更新时间',
`deleted` tinyint(1) DEFAULT '0' COMMENT '逻辑删除',
PRIMARY KEY (`id`, `user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=100002 DEFAULT CHARSET=utf8mb4 COMMENT='订单表'
partition by hash(user_id)
```

```
partitions 5;
```

分区后，查看执行计划截图：（1 分）

```
1 SELECT * FROM litemall_order where add_time BETWEEN '2020-04-01' AND '2020-05-01' and
2 user_id = 371
```

SQL执行记录 消息 执行计划1 X 结果集2 X 执行计划3 X

	id	select_type	table	partitions	type
1	1	SIMPLE	litemall_order	p1	ALL
possible_keys	key	key_len	ref	rows	filtered
				12069	1.11

Extra

Using where

挑战：完成基于 RANGE 分区的订单表分区（12 分）

分区表创建语句代码：（8 分）

```
CREATE TABLE `litemall_order` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL COMMENT '用户表的用户 ID',
  `order_sn` varchar(63) NOT NULL COMMENT '订单编号',
  `order_status` smallint(6) NOT NULL COMMENT '订单状态',
  `aftersale_status` smallint(6) DEFAULT '0' COMMENT '售后状态，0 是可申请，1 是用户已申请，2 是管理员审核通过，3 是管理员退款成功，4 是管理员审核拒绝，5 是用户已取消',
  `consignee` varchar(63) NOT NULL COMMENT '收货人名称',
  `mobile` varchar(63) NOT NULL COMMENT '收货人手机号',
  `address` varchar(127) NOT NULL COMMENT '收货具体地址',
  `message` varchar(512) NOT NULL DEFAULT '' COMMENT '用户订单留言',
  `goods_price` decimal(10,2) NOT NULL COMMENT '商品总费用',
  `freight_price` decimal(10,2) NOT NULL COMMENT '配送费用',
  `coupon_price` decimal(10,2) NOT NULL COMMENT '优惠券减免',
  `integral_price` decimal(10,2) NOT NULL COMMENT '用户积分减免',
  `groupon_price` decimal(10,2) NOT NULL COMMENT '团购优惠价减免',
  `order_price` decimal(10,2) NOT NULL COMMENT '订单费用， = goods_price + freight_price - coupon_price',
  `actual_price` decimal(10,2) NOT NULL COMMENT '实付费用， = order_price - integral_price',
  `pay_id` varchar(63) DEFAULT NULL COMMENT '微信支付编号',
  `pay_time` datetime DEFAULT NULL COMMENT '微信支付时间',
  `ship_sn` varchar(63) DEFAULT NULL COMMENT '发货编号',
  `ship_channel` varchar(63) DEFAULT NULL COMMENT '发货快递公司',
  `ship_time` datetime DEFAULT NULL COMMENT '发货开始时间',
```

```

`refund_amount` decimal(10,2) DEFAULT NULL COMMENT '实际退款金额，（有可能退款金
额小于实际支付金额）',
`refund_type` varchar(63) DEFAULT NULL COMMENT '退款方式',
`refund_content` varchar(127) DEFAULT NULL COMMENT '退款备注',
`refund_time` datetime DEFAULT NULL COMMENT '退款时间',
`confirm_time` datetime DEFAULT NULL COMMENT '用户确认收货时间',
`comments` smallint(6) DEFAULT '0' COMMENT '待评价订单商品数量',
`end_time` datetime DEFAULT NULL COMMENT '订单关闭时间',
`add_time` datetime DEFAULT NULL COMMENT '创建时间',
`update_time` datetime DEFAULT NULL COMMENT '更新时间',
`deleted` tinyint(1) DEFAULT '0' COMMENT '逻辑删除',
PRIMARY KEY (`id`, `user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=100002 DEFAULT CHARSET=utf8mb4 COMMENT='
订单表'
PARTITION BY RANGE (user_id) (
    PARTITION p1 VALUES LESS THAN (3001),
    PARTITION p2 VALUES LESS THAN (6001),
    PARTITION p3 VALUES LESS THAN (9001),
    PARTITION p4 VALUES LESS THAN (12001),
    PARTITION p5 VALUES LESS THAN MAXVALUE
);

```

未分区之前查询语句的消耗时间为 62ms，用 HASH 方法分区之后查询时间为 16ms，用 RANGE 方法分区之后查询时间为 17ms，HASH 和 RANGE 分区后的查询速度都有明显提升。

HASH 分区和 RANGE 分区的区别：HASH 分区需要借助 hash 函数，首先将分区进行编号，然后通过 hash 函数来计算确定分区内存储的数据；RANGE 分区就是将数据表内的记录按照某个属性的取值范围进行分区。

截图：分区后，查询语句 SQL 执行记录截图（2 分）

执行时间	SQL语句	消耗时间	执行结果
2020-07-12 16:23:30	SELECT * FROM litmall_order where add_time BETWEEN '2020-04-01' AND '2020-05...	17ms	执行成功

截图：分区后，查询语句查看执行计划截图（2 分）

	id	select_type	table	partitions	type
1	1	SIMPLE	litmall_order	p1	ALL

possible_keys	key	key_len	ref	rows	filtered
				11849	1.11

Extra

Using where