1) Phoenix的基本介绍:

注意:

hbase 仅支持对rowkey来设置索引, 而其他的列 无法设置的索引

hbase的 JavaAPI 较为繁琐的

由此如何解决呢? Phoenix

Phoenix是一款基于 hbase的 可以通过 SQL的方式对hbase中数据进行CURD的操作, 并且Phoenix提供了完备的ACID事务支持

同时Phoenix 可以对hbase表中 各个列 构建索引信息, 从而加快对hbase的数据读写操作

2) 安装Phoenix : 直接参考课件即可

3) Phoenix的基本使用操作

需求一: 在Phoenix中构建表: ORDER\_INFO

字段如下:

列名 说明

id 订单ID

status 订单状态

money 支付金额

pay\_way 支付方式ID

user\_id 用户ID

operation\_time 操作时间

category 商品分类

使用小写的方式:

create table if not exists ORDER\_INFO(

"id" varchar primary key,

"c1"."status" varchar,

"c1"."money" double, -- 执行后 , 为小写

"c1"."pay\_way" integer,

c1.user\_id integer, -- 执行后, 依然是大写

c1.operation\_time varchar,

c1.category varchar

);

create table if not exists ORDER\_DTL(

ID varchar primary key,

C1.STATUS varchar,

C1.MONEY double,

C1.PAY\_WAY integer,

C1.USER\_ID varchar,

C1.OPERATION\_TIME varchar,

C1.CATEGORY varchar

);

上下两种建表后的样子 是一模一样的

create table if not exists order\_dtl(

id varchar primary key,

c1.status varchar,

c1.money double,

c1.pay\_way integer,

c1.user\_id varchar,

c1.operation\_time varchar,

c1.category varchar

);

构建表的语法:

CREATE TABLE IF NOT EXISTS 表名 (

ROWKEY名称 数据类型 PRIMARY KEY

列蔟名.列名1 数据类型 NOT NULL,

列蔟名.列名2 数据类型 NOT NULL,

列蔟名.列名3 数据类型

);

注意:

1) Phoenix 默认 会将所有小写变为大写的操作

如果非要使用小写, 请在需要使用小写的字段上添加 双引号

如果在建表的时候, 使用了小写, 后续只要涉及到这个小写的字段, 都必须添加双引号

所以,在实际使用中, 建议都设置为大写

2) Phoenix 尽能执行一些简单 基础的 查询工作, 复杂工作无法胜任(比如子查询 join操作)

一般都是进行基础的单表操作

其他的操作:

!table : 查看 Phoenix所有的表

!desc 表名 : 查看某个表的结构信息

drop table if exists 表名 : 删除表的操作

插入数据:

upsert into 表名(列蔟列名, xxxx, ) VALUES(XXX, XXX, XXX)

查询的操作:

select \* from 表 where ....

需求2: 导入数据, 实现分页查询操作

UPSERT INTO "ORDER\_DTL" VALUES('000002','已提交',4070,1,'4944191','2020-04-25 12:09:16','手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000003','已完成',4350,1,'1625615','2020-04-25 12:09:37','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('000004','已提交',6370,3,'3919700','2020-04-25 12:09:39','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('000005','已付款',6370,3,'3919700','2020-04-25 12:09:44','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('000006','已提交',9380,1,'2993700','2020-04-25 12:09:41','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000007','已付款',9380,1,'2993700','2020-04-25 12:09:46','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000008','已完成',6400,2,'5037058','2020-04-25 12:10:13','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('000009','已付款',280,1,'3018827','2020-04-25 12:09:53','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('000010','已完成',5600,1,'6489579','2020-04-25 12:08:55','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('000011','已付款',5600,1,'6489579','2020-04-25 12:09:00','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('000012','已提交',8340,2,'2948003','2020-04-25 12:09:26','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('000013','已付款',8340,2,'2948003','2020-04-25 12:09:30','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('000014','已提交',7060,2,'2092774','2020-04-25 12:09:38','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('000015','已提交',640,3,'7152356','2020-04-25 12:09:49','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000016','已付款',9410,3,'7152356','2020-04-25 12:10:01','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000017','已提交',9390,3,'8237476','2020-04-25 12:10:08','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('000018','已提交',7490,2,'7813118','2020-04-25 12:09:05','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('000019','已付款',7490,2,'7813118','2020-04-25 12:09:06','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('000020','已付款',5360,2,'5301038','2020-04-25 12:08:50','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000021','已提交',5360,2,'5301038','2020-04-25 12:08:53','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000022','已取消',5360,2,'5301038','2020-04-25 12:08:58','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('000023','已付款',6490,0,'3141181','2020-04-25 12:09:22','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('000024','已付款',3820,1,'9054826','2020-04-25 12:10:04','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('000025','已提交',4650,2,'5837271','2020-04-25 12:08:52','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('000026','已付款',4650,2,'5837271','2020-04-25 12:08:57','机票;文娱;');

执行分页查询:

select \* from order\_dtl limit 每页展示多少条 offset 从第几条开始查询;

offset后的值 计算规则:

(当前页 -1) \* 每页展示条数

需求三: 在创建表的时候 指定预分区操作:

方式一: 手动构建

drop table ORDER\_DTL;

create table if not exists ORDER\_DTL(

"id" varchar primary key,

C1."status" varchar,

C1."money" float,

C1."pay\_way" integer,

C1."user\_id" varchar,

C1."operation\_time" varchar,

C1."category" varchar

)

COMPRESSION='GZ'

SPLIT ON ('3','5','7');

插入数据:

UPSERT INTO "ORDER\_DTL" VALUES('02602f66-adc7-40d4-8485-76b5632b5b53','已提交',4070,1,'4944191','2020-04-25 12:09:16','手机;');

UPSERT INTO "ORDER\_DTL" VALUES('0968a418-f2bc-49b4-b9a9-2157cf214cfd','已完成',4350,1,'1625615','2020-04-25 12:09:37','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('0e01edba-5e55-425e-837a-7efb91c56630','已提交',6370,3,'3919700','2020-04-25 12:09:39','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('0e01edba-5e55-425e-837a-7efb91c56630','已付款',6370,3,'3919700','2020-04-25 12:09:44','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('0f46d542-34cb-4ef4-b7fe-6dcfa5f14751','已提交',9380,1,'2993700','2020-04-25 12:09:41','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('0f46d542-34cb-4ef4-b7fe-6dcfa5f14751','已付款',9380,1,'2993700','2020-04-25 12:09:46','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('1fb7c50f-9e26-4aa8-a140-a03d0de78729','已完成',6400,2,'5037058','2020-04-25 12:10:13','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('23275016-996b-420c-8edc-3e3b41de1aee','已付款',280,1,'3018827','2020-04-25 12:09:53','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('2375a7cf-c206-4ac0-8de4-863e7ffae27b','已完成',5600,1,'6489579','2020-04-25 12:08:55','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('2375a7cf-c206-4ac0-8de4-863e7ffae27b','已付款',5600,1,'6489579','2020-04-25 12:09:00','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('269fe10c-740b-4fdb-ad25-7939094073de','已提交',8340,2,'2948003','2020-04-25 12:09:26','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('269fe10c-740b-4fdb-ad25-7939094073de','已付款',8340,2,'2948003','2020-04-25 12:09:30','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('2849fa34-6513-44d6-8f66-97bccb3a31a1','已提交',7060,2,'2092774','2020-04-25 12:09:38','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('28b7e793-6d14-455b-91b3-0bd8b23b610c','已提交',640,3,'7152356','2020-04-25 12:09:49','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('28b7e793-6d14-455b-91b3-0bd8b23b610c','已付款',9410,3,'7152356','2020-04-25 12:10:01','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('2909b28a-5085-4f1d-b01e-a34fbaf6ce37','已提交',9390,3,'8237476','2020-04-25 12:10:08','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('2a01dfe5-f5dc-4140-b31b-a6ee27a6e51e','已提交',7490,2,'7813118','2020-04-25 12:09:05','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('2a01dfe5-f5dc-4140-b31b-a6ee27a6e51e','已付款',7490,2,'7813118','2020-04-25 12:09:06','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('2b86ab90-3180-4940-b624-c936a1e7568d','已付款',5360,2,'5301038','2020-04-25 12:08:50','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('2b86ab90-3180-4940-b624-c936a1e7568d','已提交',5360,2,'5301038','2020-04-25 12:08:53','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('2b86ab90-3180-4940-b624-c936a1e7568d','已取消',5360,2,'5301038','2020-04-25 12:08:58','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('2e19fbe8-7970-4d62-8e8f-d364afc2dd41','已付款',6490,0,'3141181','2020-04-25 12:09:22','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('2fc28d36-dca0-49e8-bad0-42d0602bdb40','已付款',3820,1,'9054826','2020-04-25 12:10:04','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('31477850-8b15-4f1b-9ec3-939f7dc47241','已提交',4650,2,'5837271','2020-04-25 12:08:52','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('31477850-8b15-4f1b-9ec3-939f7dc47241','已付款',4650,2,'5837271','2020-04-25 12:08:57','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('39319322-2d80-41e7-a862-8b8858e63316','已提交',5000,1,'5686435','2020-04-25 12:08:51','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('39319322-2d80-41e7-a862-8b8858e63316','已完成',5000,1,'5686435','2020-04-25 12:08:56','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('3d2254bd-c25a-404f-8e42-2faa4929a629','已提交',5000,3,'1274270','2020-04-25 12:08:41','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('3d2254bd-c25a-404f-8e42-2faa4929a629','已付款',5000,3,'1274270','2020-04-25 12:08:42','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('3d2254bd-c25a-404f-8e42-2faa4929a629','已完成',5000,1,'1274270','2020-04-25 12:08:43','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('42f7fe21-55a3-416f-9535-baa222cc0098','已完成',3600,2,'2661641','2020-04-25 12:09:58','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('44231dbb-9e58-4f1a-8c83-be1aa814be83','已提交',3950,1,'3855371','2020-04-25 12:08:39','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('44231dbb-9e58-4f1a-8c83-be1aa814be83','已付款',3950,1,'3855371','2020-04-25 12:08:40','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('526e33d2-a095-4e19-b759-0017b13666ca','已完成',3280,0,'5553283','2020-04-25 12:09:01','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('5a6932f4-b4a4-4a1a-b082-2475d13f9240','已提交',50,2,'1764961','2020-04-25 12:10:07','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('5fc0093c-59a3-417b-a9ff-104b9789b530','已提交',6310,2,'1292805','2020-04-25 12:09:36','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('605c6dd8-123b-4088-a047-e9f377fcd866','已完成',8980,2,'6202324','2020-04-25 12:09:54','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('613cfd50-55c7-44d2-bb67-995f72c488ea','已完成',6830,3,'6977236','2020-04-25 12:10:06','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('62246ac1-3dcb-4f2c-8943-800c9216c29f','已提交',8610,1,'5264116','2020-04-25 12:09:14','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('62246ac1-3dcb-4f2c-8943-800c9216c29f','已付款',8610,1,'5264116','2020-04-25 12:09:18','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('625c7fef-de87-428a-b581-a63c71059b14','已提交',5970,0,'8051757','2020-04-25 12:09:07','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('625c7fef-de87-428a-b581-a63c71059b14','已付款',5970,0,'8051757','2020-04-25 12:09:19','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('6d43c490-58ab-4e23-b399-dda862e06481','已提交',4570,0,'5514248','2020-04-25 12:09:34','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('70fa0ae0-6c02-4cfa-91a9-6ad929fe6b1b','已付款',4100,1,'8598963','2020-04-25 12:09:08','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('7170ce71-1fc0-4b6e-a339-67f525536dcd','已完成',9740,1,'4816392','2020-04-25 12:09:51','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('7170ce71-1fc0-4b6e-a339-67f525536dcd','已提交',9740,1,'4816392','2020-04-25 12:10:03','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('71961b06-290b-457d-bbe0-86acb013b0e3','已付款',6550,3,'2393699','2020-04-25 12:08:47','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('71961b06-290b-457d-bbe0-86acb013b0e3','已付款',6550,3,'2393699','2020-04-25 12:08:48','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('71961b06-290b-457d-bbe0-86acb013b0e3','已完成',6550,3,'2393699','2020-04-25 12:08:49','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('72dc148e-ce64-432d-b99f-61c389cb82cd','已提交',4090,1,'2536942','2020-04-25 12:10:12','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('72dc148e-ce64-432d-b99f-61c389cb82cd','已付款',4090,1,'2536942','2020-04-25 12:10:14','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('7c0c1668-b783-413f-afc4-678a5a6d1033','已完成',3850,3,'6803936','2020-04-25 12:09:20','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('7fa02f7a-10df-4247-9935-94c8b7d4dbc0','已提交',1060,0,'6119810','2020-04-25 12:09:21','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('820c5e83-f2e0-42d4-b5f0-83802c75addc','已付款',9270,2,'5818454','2020-04-25 12:10:09','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('83ed55ec-a439-44e0-8fe0-acb7703fb691','已完成',8380,2,'6804703','2020-04-25 12:09:52','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('85287268-f139-4d59-8087-23fa6454de9d','已提交',9750,1,'4382852','2020-04-25 12:09:43','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('85287268-f139-4d59-8087-23fa6454de9d','已付款',9750,1,'4382852','2020-04-25 12:09:48','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('85287268-f139-4d59-8087-23fa6454de9d','已取消',9750,1,'4382852','2020-04-25 12:10:00','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('8d32669e-327a-4802-89f4-2e91303aee59','已提交',9390,1,'4182962','2020-04-25 12:09:57','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('8dadc2e4-63f1-490f-9182-793be64fed76','已付款',9350,1,'5937549','2020-04-25 12:09:02','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('94ad8ee0-8898-442c-8cb1-083a4b609616','已提交',4370,0,'4666456','2020-04-25 12:09:13','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('994cbb44-f0ee-45ff-a4f4-76c87bc2b972','已付款',3190,3,'3200759','2020-04-25 12:09:25','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('9bf92519-6eb3-449a-853b-0e19f6005887','已提交',1100,0,'3457528','2020-04-25 12:10:11','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('9ff3032c-8679-4247-9e6f-4caf2dc93aff','已提交',850,0,'8835231','2020-04-25 12:09:40','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('9ff3032c-8679-4247-9e6f-4caf2dc93aff','已付款',850,0,'8835231','2020-04-25 12:09:45','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('a467ba42-f91e-48a0-865e-1703aaa45e0e','已提交',8040,0,'8206022','2020-04-25 12:09:50','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('a467ba42-f91e-48a0-865e-1703aaa45e0e','已付款',8040,0,'8206022','2020-04-25 12:10:02','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('a5302f47-96d9-41b4-a14c-c7a508f59282','已付款',8570,2,'5319315','2020-04-25 12:08:44','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('a5b57bec-6235-45f4-bd7e-6deb5cd1e008','已提交',5700,3,'6486444','2020-04-25 12:09:27','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('a5b57bec-6235-45f4-bd7e-6deb5cd1e008','已付款',5700,3,'6486444','2020-04-25 12:09:31','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('ae5c3363-cf8f-48a9-9676-701a7b0a7ca5','已付款',7460,1,'2379296','2020-04-25 12:09:23','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('b1fb2399-7cf2-4af5-960a-a4d77f4803b8','已提交',2690,3,'6686018','2020-04-25 12:09:55','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('b21c7dbd-dabd-4610-94b9-d7039866a8eb','已提交',6310,2,'1552851','2020-04-25 12:09:15','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('b4bfd4b7-51f5-480e-9e23-8b1579e36248','已提交',4000,1,'3260372','2020-04-25 12:09:35','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('b63983cc-2b59-4992-84c6-9810526d0282','已提交',7370,3,'3107867','2020-04-25 12:08:45','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('b63983cc-2b59-4992-84c6-9810526d0282','已付款',7370,3,'3107867','2020-04-25 12:08:46','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('bf60b752-1ccc-43bf-9bc3-b2aeccacc0ed','已提交',720,2,'5034117','2020-04-25 12:09:03','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('c808addc-8b8b-4d89-99b1-db2ed52e61b4','已提交',3630,1,'6435854','2020-04-25 12:09:10','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('cc9dbd20-cf9f-4097-ae8b-4e73db1e4ba1','已付款',5000,0,'2007322','2020-04-25 12:08:38','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('ccceaf57-a5ab-44df-834a-e7b32c63efc1','已提交',2660,2,'7928516','2020-04-25 12:09:42','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('ccceaf57-a5ab-44df-834a-e7b32c63efc1','已付款',2660,2,'7928516','2020-04-25 12:09:47','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('ccceaf57-a5ab-44df-834a-e7b32c63efc1','已完成',2660,2,'7928516','2020-04-25 12:09:59','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('d7be5c39-e07c-40e8-bf09-4922fbc6335c','已付款',8750,2,'1250995','2020-04-25 12:09:09','食品;家用电器;');

UPSERT INTO "ORDER\_DTL" VALUES('dfe16df7-4a46-4b6f-9c6d-083ec215218e','已完成',410,0,'1923817','2020-04-25 12:09:56','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('e1241ad4-c9c1-4c17-93b9-ef2c26e7f2b2','已付款',6760,0,'2457464','2020-04-25 12:08:54','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('e1241ad4-c9c1-4c17-93b9-ef2c26e7f2b2','已提交',6760,0,'2457464','2020-04-25 12:08:59','数码;女装;');

UPSERT INTO "ORDER\_DTL" VALUES('e180a9f2-9f80-4b6d-99c8-452d6c037fc7','已付款',8120,2,'7645270','2020-04-25 12:09:28','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('e180a9f2-9f80-4b6d-99c8-452d6c037fc7','已完成',8120,2,'7645270','2020-04-25 12:09:32','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('e4418843-9ac0-47a7-bfd8-d61c4d296933','已付款',8170,2,'7695668','2020-04-25 12:09:11','家用电器;;电脑;');

UPSERT INTO "ORDER\_DTL" VALUES('e8b3bb37-1019-4492-93c7-305177271a71','已完成',2560,2,'4405460','2020-04-25 12:10:05','男装;男鞋;');

UPSERT INTO "ORDER\_DTL" VALUES('eb1a1a22-953a-42f1-b594-f5dfc8fb6262','已完成',2370,2,'8233485','2020-04-25 12:09:24','机票;文娱;');

UPSERT INTO "ORDER\_DTL" VALUES('ecfd18f5-45f2-4dcd-9c47-f2ad9b216bd0','已付款',8070,3,'6387107','2020-04-25 12:09:04','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('ecfd18f5-45f2-4dcd-9c47-f2ad9b216bd0','已完成',8070,3,'6387107','2020-04-25 12:09:17','酒店;旅游;');

UPSERT INTO "ORDER\_DTL" VALUES('f1226752-7be3-4702-a496-3ddba56f66ec','已付款',4410,3,'1981968','2020-04-25 12:10:10','维修;手机;');

UPSERT INTO "ORDER\_DTL" VALUES('f642b16b-eade-4169-9eeb-4d5f294ec594','已提交',4010,1,'6463215','2020-04-25 12:09:29','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('f642b16b-eade-4169-9eeb-4d5f294ec594','已付款',4010,1,'6463215','2020-04-25 12:09:33','男鞋;汽车;');

UPSERT INTO "ORDER\_DTL" VALUES('f8f3ca6f-2f5c-44fd-9755-1792de183845','已付款',5950,3,'4060214','2020-04-25 12:09:12','机票;文娱;');

方式二: hash 加盐

drop table if exists ORDER\_DTL;

create table if not exists ORDER\_DTL(

"id" varchar primary key,

C1."status" varchar,

C1."money" float,

C1."pay\_way" integer,

C1."user\_id" varchar,

C1."operation\_time" varchar,

C1."category" varchar

)

COMPRESSION='GZ', SALT\_BUCKETS=10;

注意事项:

采用Phoenix的hash加盐方式, 在hbase中对应表的rowkey前缀上会添加随机数据 从而在一定的程度上避免数据倾斜问题

如果使用Phoenix查询, 此时Phoenix会自动将前缀处理掉, 从而让无法感知到不到此操作, 提升体验度

4) Phoenix与HBase之间产生映射配置操作 :

当Phoenix构建好后, 如果想让其对hbase中原有的表进行操作, 需要在Phoenix中构建响应映射匹配关系

hbase中表 与 MySQL中表 之间还差一点什么内容呢? schema信息 (约束信息)

如何来构建呢?

-- 映射HBase中的表 (重要)

CREATE VIEW "my\_hbase\_table"

( k VARCHAR primary key, "v" UNSIGNED\_LONG) default\_column\_family='a';

-- 映射Phoenix中的表

CREATE VIEW my\_view ( new\_col SMALLINT )

AS SELECT \* FROM my\_table WHERE k = 100;

-- 映射到一个SQL查询

CREATE VIEW my\_view\_on\_view

AS SELECT \* FROM my\_view WHERE new\_col > 70;

映射 MOMO\_CHAT:MSG 表:

-- 创建MOMO\_CHAT:MSG视图

create view if not exists "MOMO\_CHAT". "MSG" (

"pk" varchar primary key, -- 指定ROWKEY映射到主键

"C1"."msg\_time" varchar,

"C1"."sender\_nickyname" varchar,

"C1"."sender\_account" varchar,

"C1"."sender\_sex" varchar,

"C1"."sender\_ip" varchar,

"C1"."sender\_os" varchar,

"C1"."sender\_phone\_type" varchar,

"C1"."sender\_network" varchar,

"C1"."sender\_gps" varchar,

"C1"."receiver\_nickyname" varchar,

"C1"."receiver\_ip" varchar,

"C1"."receiver\_account" varchar,

"C1"."receiver\_os" varchar,

"C1"."receiver\_phone\_type" varchar,

"C1"."receiver\_network" varchar,

"C1"."receiver\_gps" varchar,

"C1"."receiver\_sex" varchar,

"C1"."msg\_type" varchar,

"C1"."distance" varchar,

"C1"."message" varchar

);

需求: 根据 指定的发件人 和 收件人 以及聊天的时间, 查询对应数据 :

时间: 2020-12-15号

发件人: 13399763688

收件人: 18743137694

select "C1"."sender\_account" ,"C1"."receiver\_account","C1"."msg\_time", "C1"."message" from "MOMO\_CHAT". "MSG" where "C1"."msg\_time" between '2020-12-15 00:00:00' and '2020-12-15 23:59:59' and "C1"."sender\_account" = '13399763688' and "C1"."receiver\_account" = '18743137694' ;

5. Phoenix的索引相关使用 :

共计有四种: 全局索引 本地索引 覆盖索引 函数索引

全局索引:

全局索引适用于读多写少业务

因为: 全局索引在构建的时候, 需要单独构建一张索引表, 此表和目标表具有相同region的数量

当对目标表数据进行更新操作(增 删 改)的时候, 对应索引表也需要进行更新操作, 此时写入操作 就需要多干活, 效率自然就会降低

如何使用呢?

CREATE INDEX 索引名称 ON 表名 (列名1, 列名2, 列名3...)

注意事项:

SELECT查询中引用了其他非索引列，该索引是不会生效的

例如:

表A 中 有字段 B C D 三个字段 , 对C 进行构建索引的操作:

select C from A where C =''; 此时 C索引会生效的

如果:

select \* from A where C = ''and B ='' ; 此时 C索引无法生效

本地索引 :

本地索引适合写操作频繁，读相对少的业务

因为:

本地索引在建立索引数据的时候, 直接在目标表内部构建, 此时不需要在额外构建一张索引表, 这个时候, 写入的时候, 不需要写入多个表, 本质上

依然在写入一个表, 此时对写入没有太大的影响, 但是由于数据都在一块, 导致在内存中存储的数据量下降了,数据都存储在磁盘, 读取就会受到

一定的影响

注意事项:

只要在查询中 有本地索引的字段, 就会使用, 即使整个查询中 有非索引字段 也不受任何影响

如果目标表采用加盐的方式构建预分区表, 无法使用本地索引

如何使用呢?

CREATE local INDEX 索引名称 ON 表名 (列名1, 列名2, 列名3...)

覆盖索引:

当我们构建覆盖索引 后, 将原有的数据 和 索引的数据, 放置在一起, 形成一个单独的索引表, 这样在查询数据的时候, 不需要在索引表

查询到结果后 然后再根据结果到目标中查询数据了

如何构建?

CREATE INDEX my\_index ON my\_table (v1,v2) INCLUDE(v3)

函数索引:

针对某个函数来构建索引操作, 当查询的时候, 使用这个函数 那么就会使用到对应索引信息

如何构建?

-- 创建索引

CREATE INDEX UPPER\_NAME\_IDX ON EMP (UPPER(FIRST\_NAME||' '||LAST\_NAME))

-- 以下查询会走索引

SELECT EMP\_ID FROM EMP WHERE UPPER(FIRST\_NAME||' '||LAST\_NAME)='JOHN DOE'

比如说:

select b,c,d where b =''; 使用索引优化:

第一种:

CREATE INDEX 索引名称 ON A (b, c, d) --此种 可用使用, 但是对资源效率比较高, 因为需要对 c d 也需要单独索引

第二种: 推荐使用

CREATE INDEX 索引名称 ON A (b) INCLUDE(c,d) -- 相当于的对 b构建索引, 构建的时候, 将c d的数据也放置在 索引表中

对陌陌案例中 MOMO\_CHAT:MSG表 构建索引: 本地索引 + 函数索引

CREATE LOCAL INDEX LOCAL\_IDX\_MOMO\_MSG ON MOMO\_CHAT.MSG(substr("msg\_time", 0, 10), "sender\_account", "receiver\_account");

select "C1"."sender\_account" ,"C1"."receiver\_account","C1"."msg\_time", "C1"."message"

from "MOMO\_CHAT". "MSG"

where substring("C1"."msg\_time",0,10) = '2020-12-15' and "C1"."sender\_account" = '13399763688' and "C1"."receiver\_account" = '18743137694' ;

索引 一般不是在建表就指定好了, 除非已经预测到需要用到这些字段 如果不知道, 可以在后续执行查询的过程中 发现某些字段使用比较频繁

或者某些函数使用比较频繁, 可以尝试对这些字段或者函数构建索引了

6. hbase的读取数据的流程:

流程步骤:

1) 由客户端发起读取数据的请求, 先去连接zookeeper, 主要目的是用于获取hbase:meta表 所在的那个regionServer上

2) 连接meta对应的regionServer, 从meta表获取 对应目标表有那些region, 以及这些region被那些regionServer所管理

region1: node1

region2: node2

region3: node1

注意:

如果采用scan 此时一般要返回整个表每个region对应regionServer的地址

如果采用get 此时返回对应要读取rowkey所在region的regionServer的地址

3) 并发的连接对应regionServer, 开始进行读取数据操作, 首先先从memstore开始读取, 读取完成后接着到blockcache中读取

然后 再到 storeFile中读取, 最后再到大Hfile中读取数据

注意: 在此处读取的时候, 如果命令中存在过滤器 此时就会将过滤条件分发到各个region上, 进行过滤读取操作

4) 每个region可能都会有一定的数据返回, 将各个region读到数据都返回给客户端, 客户端对数据进行排序展示即可

hbase:meta表: 特殊的系统表, 此表只会有一个region 主要用于存储 hbase中各个表的元数据信息

元数据信息包含:

表对应有那些region, 每个region的范围是什么, 每个region被那些regionServer所管理 ....