

# Hive视图与索引

- What?Why?How?





# Hive Lateral View

- hive Lateral View
  - Lateral View用于和UDTF函数 (explode、split) 结合来使用。
  - 首先通过UDTF函数拆分成多行，再将多行结果组合成一个支持别名的虚拟表。
  - 主要解决在select使用UDTF做查询过程中，查询只能包含单个UDTF，不能包含其他字段、以及多个UDTF的问题
  - 语法：
    - LATERAL VIEW udtf(expression) tableAlias AS columnAlias (' columnAlias)





# Hive Lateral View

▪ 例:

- 统计人员表中共有多少种爱好、多少个城市?

col_name	data_type	comment
id	int	
name	string	
age	int	
likes	array<string>	
address	map<string,string>	
country	string	
city	string	

  

# Partition Information		
# col_name	data_type	comment
country	string	
city	string	

```
select count(distinct(myCol1)), count(distinct(myCol2)) from psn2
```

```
LATERAL VIEW explode(likes) myTable1 AS myCol1
```

```
LATERAL VIEW explode(address) myTable2 AS myCol2, myCol3;
```





# Hive 视图

- hive View视图
  - 和关系型数据库中的普通视图一样，hive也支持视图
  - 特点：
    - 不支持物化视图
    - 只能查询，不能做加载数据操作
    - 视图的创建，只是保存一份元数据，查询视图时才执行对应的子查询
    - view定义中若包含了ORDER BY/LIMIT语句，当查询视图时也进行ORDER BY/LIMIT语句操作，view当中定义的优先级更高
    - view支持迭代视图





# Hive 视图

- View语法

- 创建视图:

```
CREATE VIEW [IF NOT EXISTS] [db_name.]view_name  
    [(column_name [COMMENT column_comment], ...) ]  
    [COMMENT view_comment]  
    [TBLPROPERTIES (property_name = property_value, ...)]  
    AS SELECT ... ;
```

- 查询视图:

```
select cols from view;
```

- 删除视图:

```
DROP VIEW [IF EXISTS] [db_name.]view_name;
```





# Hive 索引

- Hive 索引

- 目的：优化查询以及检索性能
- 创建索引：

```
create index t1_index on table psn2(name)
```

```
as 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler' with deferred rebuild
```

```
in table t1_index_table;
```

- *as*：指定索引器；
- *in table*：指定索引表，若不指定默认生成在default\_\_psn2\_t1\_index\_\_表中

```
create index t1_index on table psn2(name)
```

```
as 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler' with deferred rebuild;
```





# Hive 索引

## - 查询索引

- `show index on psn2;`

## - 重建索引 (建立索引之后必须重建索引才能生效)

- `ALTER INDEX t1_index ON psn2 REBUILD;`

## - 删除索引

- `DROP INDEX IF EXISTS t1_index ON psn2;`

