项目数据分析文档

- 实训背景

  随着互联网技术的发展，网络信息呈暴涨式增长，丰富多彩的网络生活给人们带来更多体验的同时，也给高校图书馆带来更大的压力。如何让高校图书馆保持活跃，给广大师生提供优质的服务，是一个比较重要的研究课题，如何在数以十万甚至百万的书籍中给用户推荐最适合的图书是其中的关键。

  HELLOWORLD2022接收到了大三的第一个订单——为一家书店制作一份推荐进货单。根据现在网络图书网站所拥有的图书数据进行图书推荐分析，了解大众喜好的同时根据不同读者的喜好，推荐给读者喜欢的图书类型、作者、出版社以及一个经济实惠的参考价格。最终交付给书店老板一张推荐进书单。

- 实训的目标

- 掌握数据仓库的概念并能设计数据仓库的库表结构。
- 熟练HDFS的shell命令的使用。
- 掌握Hive和Mysql的安装及配置。
- 掌握Hive数据仓库的内置数据类型。
- 掌握Hive的DDL语句和DML语句。
- 掌握如何建立事实表和维度表的方法。
- 掌握Sqoop的安装配置和导入导出命令。

- 实训的环境

- 项目开发环境:Pycharm2022，或者IDEA2022
- Linux操作系统: Centos7
- 分析框架: Hive3.1.3或Spark Sql
- 大数据平台: Hadoop3.3.2或2.9 Hive 3.1.3 Sqoop1.4.7开发语言:hql
- 分析框架: Hive3.1.3或Spark Sql、Spark Mlib

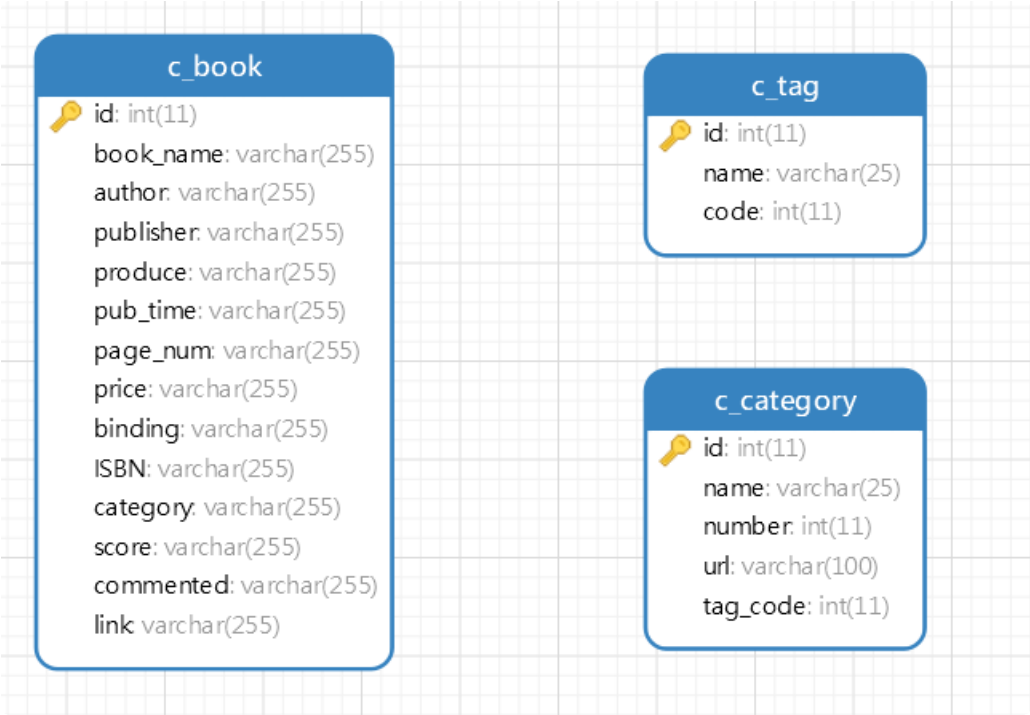
- 实训的过程

- 实训的过程包括:

  设计数据仓库的库表结构、加载数据仓库需要的数据、首页界面的图书类别排名的数据分析、图书的各个类别里分数排行的类别得分排行、不同作者的图书得分排行分析、同一本书不同出版社的得分和价格区别分析、图书类别的词云分析、导出分析结果到Mysql数据库。

- 实训任务一:设计数据仓库的库表结构

举例:原型设计中招聘大数据首页界面功能点所需要的库表结构设计如下，学生可参考这个功能的库表设计出其他功能点所需要的库表结构。



- 实训任务二:加载数据仓库的事实表数据

在slave1启动hive metastore服务：

hive --service metastore

- 实训任务三：首页图书可视化大屏展示

1、展示图书总本数的SQL：

创建一个采集图书总数的统计表book_num：

bookNum　　Scala　　复制代码

```
val BookNum = spark.sql("select count(*) as `图书总数` from bigdata.book")
```

图书总数
14683

2、展示图书大类统计的SQL：

创建一个图书大类统计的统计表book_category：

bookCategory Scala 复制代码

```scala
val BookCategory = spark.sql(
"""
|SELECT
|t.`name` as `类别名称`,
|count(*) as `图书数量`
|FROM
|bigdata.tag t
|JOIN bigdata.category c ON t.`code`=c.tag_code
|JOIN bigdata.book b ON c.`name`=b.category
|GROUP BY t.`name`
|""".stripMargin
)
```

3、展示得分等宽分布的SQL（运用了SparkMlib中的等宽分布函数Bucketizer）：

创建一个得分分布的统计表book_score：

bookScore Scala 复制代码

```scala
val df1 = spark.sql("select score from bigdata.book order by score")
val splits = Array(Double.NegativeInfinity, 2.0, 4.0, 6.0, 8.0, 10.0, Double.PositiveInfinity)
new Bucketizer()
.setInputCol("score")
.setOutputCol("scorepartithon")
.setSplits(splits)
.transform(df1)
.createTempView("scorebuck")
val bookScore = spark.sql(
"""
|SELECT
|scorepartithon ,
|concat("[", min(score), "-", max(score), "]") as `partitionsplit`,
|COUNT(*) as `num`
|FROM
|scorebuck
|GROUP BY scorepartithon
|ORDER BY scorepartithon
|""".stripMargin
)
```

```
================3.得分分布统计================
+--------------+--------------+----+
|scorepartithon|partitionsplit| num|
+--------------+--------------+----+
|           0.0|     [0.0-0.0]|1294|
|           1.0|     [2.6-3.5]|   2|
|           2.0|     [4.0-5.9]| 109|
|           3.0|     [6.0-7.9]|4029|
|           4.0|     [8.0-9.9]|9239|
|           5.0|   [10.0-10.0]|  10|
+--------------+--------------+----+
```

4、展示图书价格等频分布的统计表book_price的SQL（运用了SparkMlib中的等频分布函数QuantileDiscretizer）：

bookPrice Scala 复制代码

```scala
val df = spark.sql("select price from bigdata.book where price > 1")
val discretizer = new QuantileDiscretizer()
.setInputCol("price")
.setOutputCol("partition")
.setNumBuckets(6).fit(df).transform(df)
discretizer.createTempView("qd")
val bookPrice = spark.sql(
"""
|select
|partition,
|concat("[", min(price), "-", max(price), "]") as `partitionsplit`,
|count(1) as `num`
|from
|qd
|group by partition
|order by partition
|""".stripMargin)
```

```
================4.图书价格分布统计================
+---------+--------------+----+
|partition|partitionsplit| num|
+---------+--------------+----+
|      0.0|  [1.04-20.99]|2290|
|      1.0|  [21.0-29.99]|2280|
|      2.0|   [30.0-41.7]|2251|
|      3.0|   [42.0-57.3]|2344|
|      4.0|   [58.0-88.0]|2422|
|      5.0|[89.0-13670.0]|2345|
+---------+--------------+----+
```

5、展示图书出版社图书数量top10的统计表book_publisher的SQL:

bookPublisher Scala 复制代码

```scala
val bookPublisher = spark.sql(
"""
|select
|publisher,
|ROUND(AVG(score),1) as `avgscore`,
|count(1) as `num`
|from
|bigdata.book
|group by publisher
|order by count(1) desc limit 10
|""".stripMargin)
```

publisher	avgscore	num
人民文学出版社	8.2	854
上海译文出版社	8.2	689
北京联合出版公司	7.7	453
新星出版社	7.8	367
译林出版社	8.4	351
南海出版公司	8.1	313
广西师范大学出版社	7.7	310
中信出版社	8.1	309
江苏凤凰文艺出版社	7.7	282
生活·读书·新知三联书店	8.3	279

6、展示图书发售时间顺序的统计表book_time的SQL:

bookTime Scala 复制代码

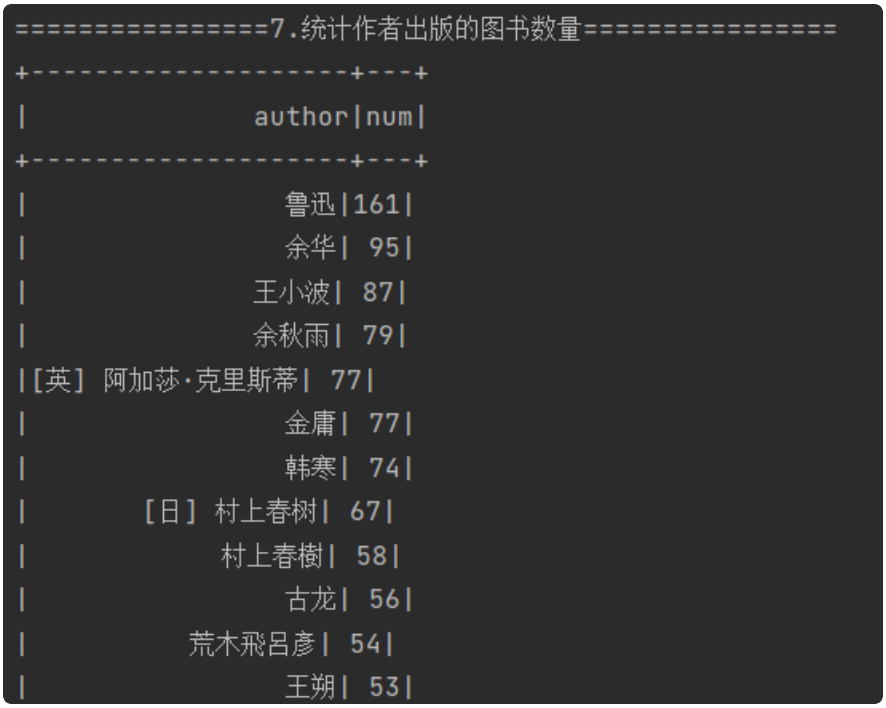
```scala
val bookTime = spark.sql(
"""
|select
|pub_time,
|count(1) as `num`
|from
|bigdata.book
|where pub_time != '其他'
|group by pub_time
|order by pub_time desc
|""".stripMargin)
```

```
================6.图书出版时间统计================
+--------+---+
|pub_time|num|
+--------+---+
|  2022-9|107|
|  2022-8|239|
|  2022-7|300|
|  2022-6|265|
|  2022-5|206|
|  2022-4|162|
|  2022-3|148|
|  2022-2| 82|
| 2022-12|  1|
| 2022-10| 29|
|  2022-1|129|
|  2021-9| 92|
|  2021-8|107|
|  2021-7| 87|
|  2021-6| 71|
```

7、展示作者出版的图书数量的统计表book_author的SQL:

bookAuthor Scala 复制代码

```scala
val bookAuthor = spark.sql(
"""
|select
|author,
|count(1) as `num`
|from
|bigdata.book
|where author != '其他'
|group by author
|order by num desc limit 10
|""".stripMargin)
```



8、展示评论数量多的top15的统计表book_comment的SQL:

bookComment Scala 复制代码

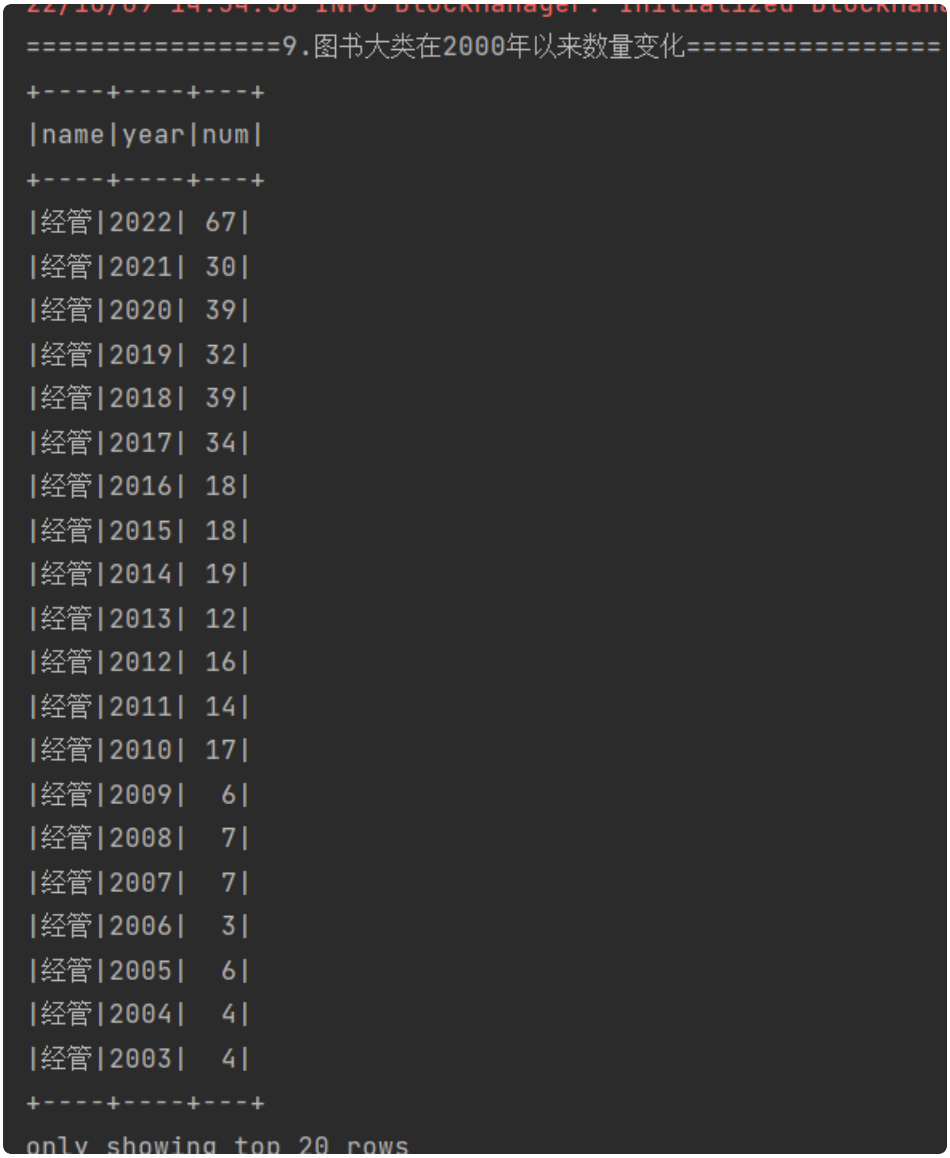
```scala
val bookComment = spark.sql(
"""
|select
|book_name,
|commented
|from
|bigdata.book
|ORDER BY commented desc limit 15
|""".stripMargin)
```

```
================8.最热图书的top15================
+--------------------+---------+
|           book_name|commented|
+--------------------+---------+
|           追风筝的人|   752525|
|           解忧杂货店|   726255|
|                活着|   716460|
|              小王子|   701191|
|              白夜行|   497365|
|       嫌疑人X的献身|   489689|
|                三体|   446737|
|                围城|   426057|
|              白夜行|   407002|
|            百年孤独|   383237|
|              红楼梦|   380856|
|           挪威的森林|   340927|
|    房思琪的初恋乐园|   324633|
|                看见|   320965|
|平凡的世界（全三部）|   301872|
+--------------------+---------+
```

9、展示图书大类在2000年以来数量变化的统计表category_time的SQL:

categoryTime　　Scala　　复制代码

```
val categoryTime = spark.sql(
"""
|select
|t.`name`,
|`year`,
|COUNT(1) as `num`
|from
|bigdata.book b
|JOIN bigdata.category c ON b.category=c.`name`
|JOIN bigdata.tag t on c.tag_code = t.`code`
|WHERE pub_time != '其他' and year BETWEEN 2000 AND 2022
|GROUP BY t.`name`,`year`
|ORDER BY t.`name` DESC , `year` desc
|""".stripMargin)
```

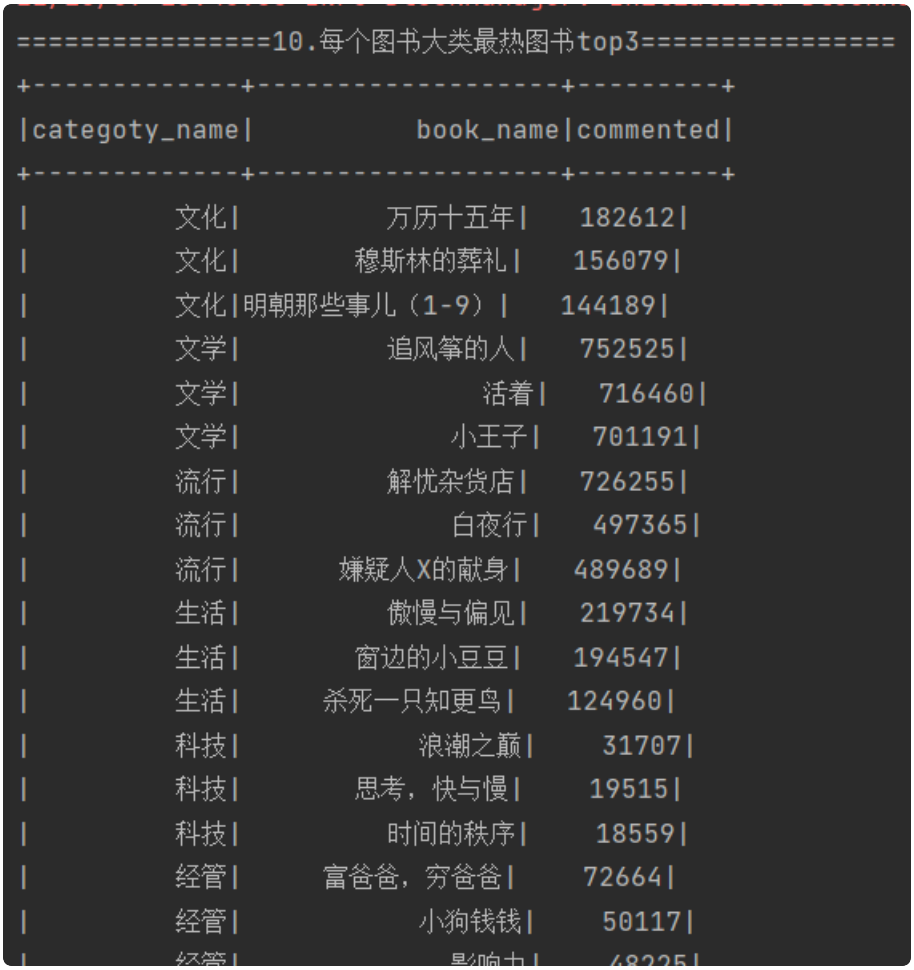


10、展示每个图书大类最热图书top3的统计表category_top3的SQL:

categoryTop3 Scala 复制代码

```scala
val tempTable = spark.sql(
"""
|SELECT
|t.`name` as `categoty_name`,
|b.book_name,
|b.commented
|FROM
|bigdata.book b
|JOIN bigdata.category c ON b.category=c.`name`
|JOIN bigdata.tag t on c.tag_code = t.`code`
|GROUP BY book_name,t.`name`,commented
|HAVING COUNT(t.`name`) < 3
|ORDER BY t.`name`,commented DESC
|""".stripMargin)
tempTable.createTempView("book_temp")

val categoryTop3 = spark.sql(
"""
|select
|a.categoty_name,
|a.book_name,
|a.commented
|from book_temp as a
|left join book_temp b on a.categoty_name = b.categoty_name and a.commented < b.commented
|group by a.categoty_name,a.book_name,a.commented
|having count(b.categoty_name) < 3
|order by a.categoty_name asc,a.commented desc
|""".stripMargin)
```



```
================10.每个图书大类最热图书top3================
+-------------+--------------------+---------+
|categoty_name|           book_name|commented|
+-------------+--------------------+---------+
|         文化|          万历十五年|   182612|
|         文化|        穆斯林的葬礼|   156079|
|         文化|明朝那些事儿（1-9）|   144189|
|         文学|          追风筝的人|   752525|
|         文学|                活着|   716460|
|         文学|              小王子|   701191|
|         流行|          解忧杂货店|   726255|
|         流行|              白夜行|   497365|
|         流行|      嫌疑人X的献身|   489689|
|         生活|          傲慢与偏见|   219734|
|         生活|        窗边的小豆豆|   194547|
|         生活|      杀死一只知更鸟|   124960|
|         科技|            浪潮之巅|    31707|
|         科技|        思考，快与慢|    19515|
|         科技|          时间的秩序|    18559|
|         经管|      富爸爸，穷爸爸|    72664|
|         经管|            小狗钱钱|    50117|
|         经管|              影响力|    48225|
```

11、展示最热门的作者排行top10的统计表author_top10的SQL:

authorTop10 Scala 复制代码

```scala
val authorTop10 = spark.sql(
"""
|SELECT
|author,
|sum(commented) as `commenteds`
|FROM
|bigdata.book
|WHERE author != '其他'
|GROUP BY author
|ORDER BY sum(commented) DESC limit 10
|""".stripMargin)
```

```
=================11.最热门的作者排行top10================
+--------------------------+----------+
|                    author|commenteds|
+--------------------------+----------+
|              [日] 东野圭吾|   1945203|
|                        余华|   1447532|
|              [日] 村上春树|   1370741|
|                      刘慈欣|   1335178|
|[哥伦比亚] 加西亚·马尔克斯|    860820|
|                        三毛|    858186|
|                        金庸|    766822|
|                      王小波|    724032|
|                      张爱玲|    579536|
|                      钱锺书|    570090|
+--------------------------+----------+
```

12、展示作者书的包装统计表author_blinding的SQL:

authorTop10 Scala 复制代码

```scala
val bookauthorTemp = readMysql("book_author",spark).createTempView("book_author")
val authorBlinding = spark.sql(
"""
|SELECT
| author,
| binding,
| COUNT( 1 ) as `num`
|FROM
| bigdata.book
|WHERE
| author IN (SELECT author FROM book_author) and binding in ('平装','精装','其他')
|GROUP BY
| author,
| binding
|ORDER BY
| author DESC
|""".stripMargin)
```

author	binding	num
古龙	平装	52
鲁迅	平装	121
[英] 阿加莎·	平装	65
王小波	精装	40
鲁迅	精装	18
古龙	其他	3
[英] 阿加莎·	精装	9
余秋雨	精装	24
[英] 阿加莎·	其他	3
[日] 村上春·	精装	23
余秋雨	平装	38
[日] 村上春·	平装	43
余秋雨	其他	10
[日] 村上春·	其他	1
鲁迅	其他	12
韩寒	平装	56
余华	精装	32
王小波	平装	41
韩寒	其他	1
韩寒	精装	17
余华	平装	50
王小波	其他	6
金庸	平装	66

13、展示图书评论分布统计表book_commented的SQL:

bookCommented Scala 复制代码

```scala
val bookCommented = spark.sql(
"""
|select
|partition,
|concat("[", min(commented), "-", max(commented), "]") as `partitionsplit`,
|count(1) as `num`
|from
|pricedis
|group by partition
|order by partition
|""".stripMargin)
```

	partition	partitionsplit	num
▶	3	[639-2414]	2787
	4	[2418-752525]	2795
	0	[0-53]	2753
	2	[213-638]	2793
	1	[54-212]	2804

14、展示最终推荐书单统计表book_list的SQL:

bookList Scala 复制代码

```scala
val finaltemp = spark.sql(
"""
|select distinct
|book_name,
|author,
|publisher,
|isbn,
|CASE
| WHEN score>=2.6 AND score<=3.5 THEN 1
| WHEN score>=4.0 AND score<=5.9 THEN 2
| WHEN score>=6.0 AND score<=7.9 THEN 3
| WHEN score>=8.0 AND score<=9.9 THEN 4
| ELSE 5
|END AS `score_weight`,
| CASE
| WHEN price>=1.04 AND price<=23.99 THEN 1
| WHEN price>=24 AND price<35 THEN 2
| WHEN price>=35 AND price<49 THEN 3
| WHEN price>=49 AND price<79 THEN 4
| ELSE 5
|END AS `price_weight`,
| CASE
|   WHEN commented>=0 AND commented<=53 THEN 1
|   WHEN commented>=54 AND commented<=212 THEN 2
|   WHEN commented>=213 AND commented<=638 THEN 3
|   WHEN commented>=639 AND commented<=2414 THEN 4
|   ELSE 5
| END AS `commented_weight`
|from
|bigdata.book
|where score != 0 and price != 0 and commented != 0
|""".stripMargin)
```

partition	partitionsplit	num
3	[639-2414]	2787
4	[2418-752525]	2795
0	[0-53]	2753
2	[213-638]	2793
1	[54-212]	2804

实训项目分析步骤配置