河南工業大學

课　程　设　计

课程设计名称：	数据分析与可视化综合实践
专　业　班　级：	数据科学与大数据技术 2101 班
学　生　姓　名：	韩林欣
学　　号：	211210100108
指　导　教　师：	黄孝楠
课程设计时间：	2023.12.18-2024.01.05

数据科学与大数据技术 专业课程设计任务书

<table>
<tr><td>学生姓名</td><td>韩林欣</td><td>专业班级</td><td>大数据 2101</td><td>学　　号</td><td>211210100108</td></tr>
<tr><td>题　　目</td><td colspan="5">地产图谱：多维数据解析中国房市——数仓分析</td></tr>
<tr><td>课题性质</td><td colspan="2">工程设计</td><td>课题来源</td><td colspan="2">自拟课题</td></tr>
<tr><td>指导教师</td><td colspan="2">黄孝楠</td><td>同组姓名</td><td colspan="2">李俊</td></tr>
<tr><td>主要内容</td><td colspan="5">一、数据
近十年的房地产年度开发企业经营情况和负债率。全国以及各省的商品房年度平均价格，销售额，销售面积，近十年年度城市竣工面积和施工面积。
二、基本功能
1. 数据存储：将 csv 文件通过 Linux 系统传入 HDFS 端存储。
2. 数仓分层：在 Hive 中处理数据并进行数据仓库分层。
3. 数据迁移：在 MySQL 中建立相同格式的表，使用 DataX 将 Hive 的表同步到 MySQL 中。
4. 后端开发：在后端编写 Servlet 建立后端数据访问接口。
5. 数据可视化：前端通过 Ajax 请求访问后端接口获取数据并用 ECharts 进行可视化。
6. 任务调度：使用 Azkaban 将业务进行调度。
7. 数据挖掘：利用数据挖掘技术分析全国房地产风向变化</td></tr>
<tr><td>任务要求</td><td colspan="5">1. 根据房地产数据进行分析，分析近十年来的全国及各省房地产风向
2. 对现有的数据进行清理提取等处理，并将处理后的数据通过 HDFS 建立 Hive 外部表并导入到 MySQL 数据库中，最后采用数据可视化工具对数据进行分析并得出数据分析图.
3. 使用 Linux、Servlet、Hadoop、Hive、MySQL，Azkaban 等技术，并运用 ECharts、IntelliJ IDEA、MobaXterm、VMWware,DataX 等操作工具.
4. 在设计过程中撰写规范的设计报告，在设计完成后成功通过答辩</td></tr>
<tr><td>参考文献</td><td colspan="5">[1]房地产大数据及其信息挖掘体系构建路径研究[J]. 隆林宁.产业与科技论坛,2023(04).
[2] “互联网+”背景下众筹模式在房地产开发项目中的应用[J]. 郭庆军;白思俊;张华波;朱海阔.项目管理技术,2016(01).
[3] 用互联网思维和手段创新房地产业发展[J]. 刘志峰.住宅产业,2015(11).
[4] 房地产行业“互联网+”模式研究与探讨[J]. 夏阳.中国房地产,2015(13).
[5] 大数据时代:房地产业的机遇与挑战[J]. 刘昱;张玉娟.河南商业高等专科学校学报,2013.</td></tr>
<tr><td>审查意见</td><td colspan="5">同意。
指导教师签字：黄孝楠
教研室主任签字：蒋建雨　　2023 年 12 月 4 日</td></tr>
</table>

目 录

1 需求分析

1.1 需求背景

随着中国经济的快速发展，房地产行业经历了巨大的变革。如今，这个行业不仅是国民经济的重要支柱，也是投资者、开发商、政府和普通民众关注的焦点。在这个庞大的市场中，数据的作用日益凸显。

然而，目前这些数据大多分散在各个机构、平台和部门中，缺乏一个统一、直观的展现方式。对于决策者来说，没有一个全局的视角，很难准确判断市场趋势和制定策略。对于普通民众来说，房地产市场的信息透明度不足，也增加了他们参与市场的难度。

地产图谱对近十年国家以及各省的房地产数据以及各个主要城市的施工竣工面积进行分析，为决策者提供我国房地产风向变化趋势。

1.2 数据分析需求

采集国家以及各个省份十年来的商品房年度平均价格，总均销售面积和总销售额，并采集出国家各个主要城市的近十年年度施工面积和竣工面积，并使用 Hive 对这些数据进行清洗，分层，汇总，最后达到数据挖掘和数据可视化的要求。

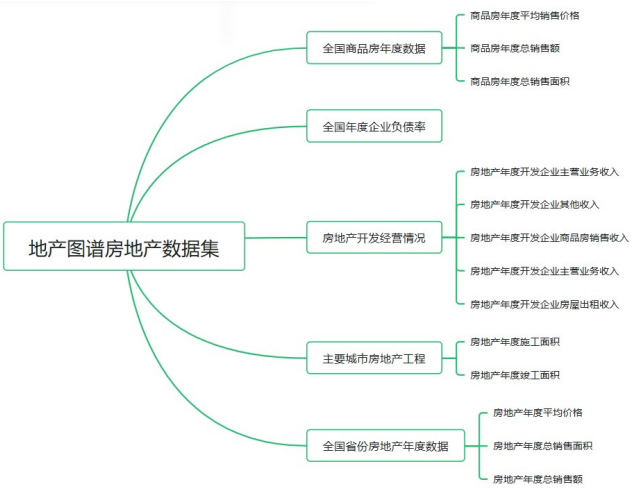


图 1-1 数据分析图

1.3 技术需求分析

1. 数据存储：将 csv 文件通过 Linux 系统传入 HDFS 端存储。

2. 数仓分层：在 Hive 中处理数据并进行数据仓库分层。

3. 数据迁移：在 MySQL 中建立相同格式的表，使用 DataX 将 Hive 的表同步到 MySQL 中。

4. 后端开发：在后端编写 Servlet 建立后端数据访问接口。

5. 数据可视化：前端通过 Ajax 请求访问后端接口获取数据并用 ECharts 进行可视化。

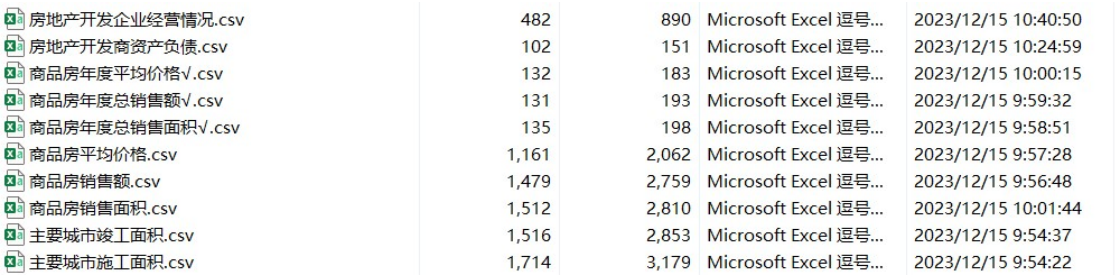
6. 任务调度：使用 Azkaban 将业务转变为工作流。

7. 数据挖掘：利用数据挖掘技术分析全国房地产风向变化。

2 概要设计

2.1 数据准备

在国家统计局采集国家国家房地产年度数据



房地产开发企业经营情况.csv	482	890	Microsoft Excel 逗号...	2023/12/15 10:40:50
房地产开发商资产负债.csv	102	151	Microsoft Excel 逗号...	2023/12/15 10:24:59
商品房年度平均价格√.csv	132	183	Microsoft Excel 逗号...	2023/12/15 10:00:15
商品房年度总销售额√.csv	131	193	Microsoft Excel 逗号...	2023/12/15 9:59:32
商品房年度总销售面积√.csv	135	198	Microsoft Excel 逗号...	2023/12/15 9:58:51
商品房平均价格.csv	1,161	2,062	Microsoft Excel 逗号...	2023/12/15 9:57:28
商品房销售额.csv	1,479	2,759	Microsoft Excel 逗号...	2023/12/15 9:56:48
商品房销售面积.csv	1,512	2,810	Microsoft Excel 逗号...	2023/12/15 10:01:44
主要城市竣工面积.csv	1,516	2,853	Microsoft Excel 逗号...	2023/12/15 9:54:37
主要城市施工面积.csv	1,714	3,179	Microsoft Excel 逗号...	2023/12/15 9:54:22

图 2-1 国家房地产年度数据

2.2 数据预处理

数据预处理使用 Python 的 pandas 库进行数据预处理，并重新导出为 csv 文件。

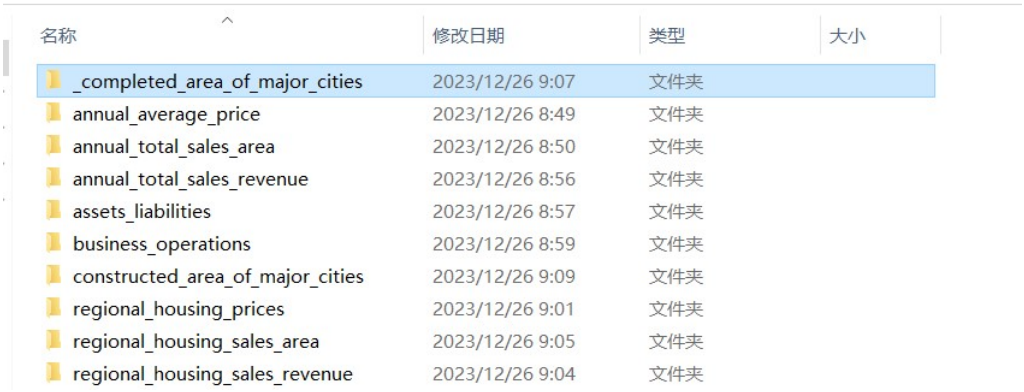


图 2-2 数据清洗结果

2.3 数据分层

数据在 Hive 中分层分为 ods 原始数据层，dwd 清洗数据层，dws 整合数据层和 ads 应用数据层

```
ads
azkaban
datax
default
dim
dwd
dws
movie
ods
```

图 2-3 数据分层

并通过以下方式进行维度表的设计

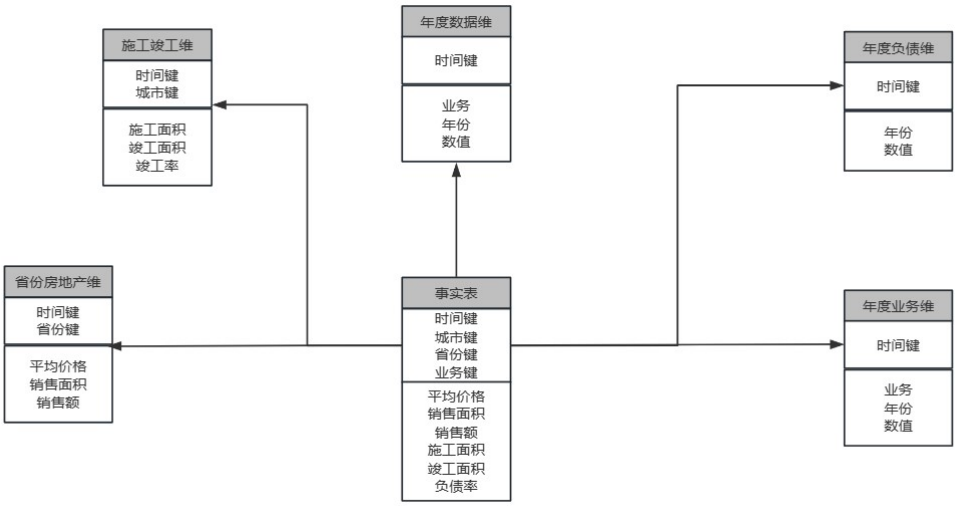


图 2-4 维度建模

2.4 数据迁移

数据迁移使用 DataX 完成，将 Hive 中的数据迁移到 MySQL 中。

图 2-5 数据迁移文件

2.5 任务调度

任务调度使用 Azkaban 完成，将 Hive 中的 SQL 文件转变为 Azkaban 的 job 进行处理。

图 2-6 任务调度文件

2.6 数据挖掘

本次数据挖掘的目的是建立房价预测模型，数据清洗去除异常值，重复值。

特征选择采取 Pearson 相关性系数和方差分析法，模型选择采用简单易懂，可解释性高的多元线性回归，模型训练使用 sklearn 机器学习库，模型预测所需的特征指标准备采用多项式拟合得到未来指标，然后代入进行预测。模型评估采用多个角度评估，如 R 方，F-statistic（F 统计量）等。

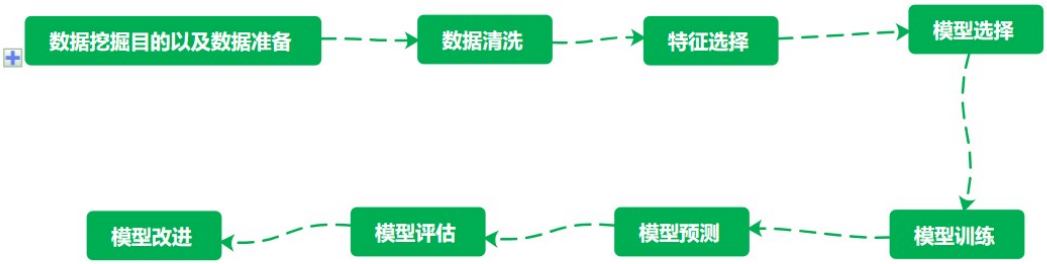


图 2-7 数据挖掘流程图

2.7 数据可视化

本次实践中可视化页面并没有像以前一样采用单一的可视化大屏。采用了模块化的设计，增加了交互性和可扩展性，为用户提供更详细的数据信息和好的使用体验。总体布局采用了 3 个主题域。通过使用 ECharts 完成数据可视化，通过 Ajax 获取后端接口的数据。具体可视化页面构造如下图所示。

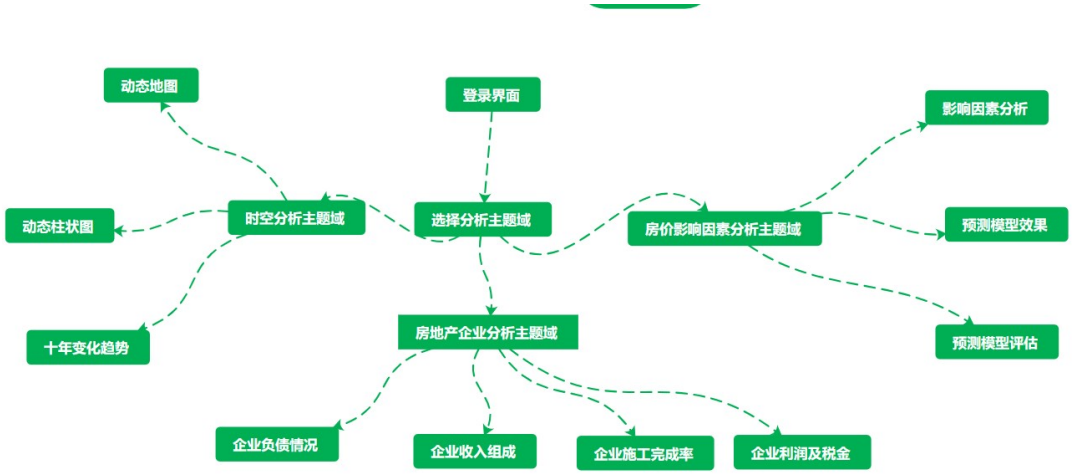


图 2-8 可视化页面模型图

3 开发工具和编程语言

3.1 开发工具

开发工具：

IDEA --代码编辑器，

VMware Workstation Pro --Linux 系统运行工具，

Visual Studio Code --代码编辑器，

DataX --数据迁移工具，

Linux --后端大数据基础操作系统，

Hadoop --提供 HDFS 和 Hive 运行环境，

Azkaban --任务调度。

3.2 编程语言

编程语言：

Java，JavaScript，Python。

4 详细设计及运行结果

4.1 数据来源

房地产数据从国家统计局官网收集得到，在国家统计局官网搜寻年度数据，并以省份和主要城市为过滤条件进行收集。

采集数据包括房地产开发企业经营情况，房地产开发商资产负债，商品房年度平均价格，商品房年度总销售额，商品房年度总销售面积，商品房平均价格，商品房销售额，商品房销售面积，主要城市竣工面积，主要城市施工面积。

图 4-1 国家统计局官网

4.2 数据分析

4.2.1 数据清洗

由于数据来源于国家统计局，相比于其他数据，国家统计局的数据完整度很高，且基本没有脏数据，不过存在一些省份出现数据情况为 0 或者为空的情况，所以对于这种情况，所需要的仅为通过数据清洗，将采集的数据中的空数据用 0 填充。

同时在清洗过程中，将省份数据进行优化，将其改为符合 ECharts 地图的类型，例如：内蒙古自治区改为内蒙古，并保存为新的数据文件。

```
import pandas as pd
df = pd.read_csv("商品房平均价格.csv")
print(df.head())
items = [[col,df[col].dtype,df[col].isnull().sum()] for col in df]
pd = pd.DataFrame(data=items,columns=["Attribute","dtype","null"])
print(pd)
# 清洗省市自治区数据，契合 echarts 的国家地图
def clean_city(city_name):
    if city_name.endswith("市") or city_name.endswith("省"):
        res = city_name[:-1]
    elif city_name=="内蒙古自治区":
        res = city_name[:3]
    else:
        res = city_name[:2]
    return res
df["地区"]=df["地区"].map(lambda x:clean_city(x))
print(df)
#用两列数据的结果填充某一列数据
df["2014 年增长率"]=None
print(df.head())
df["2014 年增长率"].fillna((df["2014 年"]-df["2013 年"])/df["2013 年"],inplace=True)
print(df.head())
```

4.2.2 数仓分层

首先是建立数据仓库的 ODS 层，ODS 层中存储的是收集到的原始数据，即将从国家统计局中采集到的是个 csv 文件保存到 HDFS 中。

ods_api_annual_total_sales_revenue_i_d

ods_api_assets_liabilities_i_d

ods_api_business_operations_i_d

ods_api_completed_area_of_major_cities_i_d

ods_api_constructed_area_of_major_cities_i_d

ods_api_regional_housing_prices_i_d

ods_api_regional_housing_sales_area_i_d

ods_api_regional_housing_sales_revenue_i_d

ods_api_annual_average_price_i_d

ods_api_annual_total_sales_area_i_d

且由于从国家统计局中采集到的数据以年份为列，而 Hive 中的列名不能以数字开头，所以在年份列名之前加上了一个 a 来符合标准。

表 4-1 ods_api_annual_average_price_i_d 表结构

business	string
a2022	double
a2021	double
a2020	double
a2019	double
a2018	double
a2017	double
a2016	double
a2015	double
a2014	double
a2013	double

而建表语句由于表结构类似，所以建表语句也非常类似，如下为例，表中的

首个字段为 business，另外的是个字段分别是 a+时间，其中 business 字段中的数据是各种的业务类型，而在时间字段下，存放着的是各种业务类型的详细数值数据，所以存放的数据类型为 double 型。

同时由于数据存储在 HDFS 中，在 Hive 中构建外部表，且从文件中读取数据，所以规定分隔符为，存储类型为文本类型，定位在 HDFS 的文件夹中。

```
create external table ods.ods_api_annual_average_price_i_d(
    business string,
    a2022 double,
    a2021 double,
    a2020 double,
    a2019 double,
    a2018 double,
    a2017 double,
    a2016 double,
    a2015 double,
    a2014 double,
    a2013 double)
row format delimited
fields terminated by ','
Stored as textfile
Location '/house/data/ods/annual_average_price';
```

其次是建立数据仓库的 DWD 层，DWD 层是 ODS 层的数据经过数据清洗后的样子，除去数据清洗外，表结构不发生变化，所以表结构与 ODS 层一样。

且由于从国家统计局中采集到的数据完整性和正确率都较高，基本没有什么脏数据，所以只需略加清洗即可。DWD 层中一共是 10 个表，分别是：

dwd_api_annual_average_price_i_d

dwd_api_annual_total_sales_revenue_i_d

dwd_api_assets_liabilities_i_d

dwd_api_business_operations_i_d

dwd_api_completed_area_of_major_cities_i_d

dwd_api_constructed_area_of_major_cities_i_d

dwd_api_regional_housing_prices_i_d

dwd_api_regional_housing_sales_area_i_d

dwd_api_regional_housing_sales_revenue_i_d

dwd_api_annual_total_sales_area_i_d

以上各表由于是将 ODS 层的数据清洗并重新存储，所以并未改变数据表的结构，表的结构仍然沿用 ODS 层的表结构，所以仍然是以 business 为义务，a+时间为时间数值 double 类型字段的表。

表 4-2　dwd_api_annual_average_price_i_d **表结构**

business	string
a2022	double
a2021	double
a2020	double
a2019	double
a2018	double
a2017	double
a2016	double
a2015	double
a2014	double
a2013	double

其次是建立数据仓库的 DWS 层，DWS 层是 DWD 层数据经过统一汇总成的宽表，相比于 DWD 层，这个表的数据更为统一，且更为方便调度和引用。

DWS 层经过观察分析，最后选择将 10 个表的内容合并为五个表，这五个表分别是表结构相似的三个表。

dws_api_assets_liabilities_i_d

dws_api_business_operations_i_d

dws_api_annual_data_i_d

表 4-3 dws_api_annual_data_i_d 表结构

business	string
a2022	double
a2021	double
a2020	double
a2019	double
a2018	double
a2017	double
a2016	double
a2015	double
a2014	double
a2013	double

以及表结构不同的两个表

dws_api_house_complete_rate_i_d

表结构经过行转列（explode）的统一汇总操作，将多个表的内容经过格式调整后转变为类似于以下表结构的类型。city 字段为主要城市字段，year 为年份字段，constructed 和 completed 分别是当年该城市的施工和竣工面积，最后的 rate 字段是由前面的施工面积和竣工面积得到的概念的房屋竣工率。

表 4-4 dws_api_house_complete_rate_i_d 表结构

city	string
year	string
constructed	double
completed	double
rate	double

建表语句时使用 explode 结构先将时间类型转变为 map 型，并以视图的方式存在，然后利用该数据进行转变，将多列数据变为少列数据，并在更改结束后使用 join 以城市和年份将不同表进行关联。

```
create table dws.dws_api_house_complete_rate_i_d as
select
t3.city as city,
```

```
    t3.year as year,
    t4.number as constructed,
    t4.number as completed,
    t4.number/t3.number as rate
    from
    (select city,year,number
    from
    (select
city,map('2022',a2022,'2021',a2021,'2020',a2020,'2019',a2019,'2018',a2018,'2017',a20
17,'2016',a2016,'2015',a2015,'2014',a2014,'2013',a2013) as value from
dwd.dwd_api_completed_area_of_major_cities_i_d) t2
    lateral view explode(value) temp_view2 as year,number) as t3
    join
    (select city,year,number
    from
    (select
city,map('2022',a2022,'2021',a2021,'2020',a2020,'2019',a2019,'2018',a2018,'2017',a20
17,'2016',a2016,'2015',a2015,'2014',a2014,'2013',a2013) as value from
dwd.dwd_api_constructed_area_of_major_cities_i_d) t2
    lateral view explode(value) temp_view2 as year,number) as t4
    on t3.city = t4.city and
    t3.year = t4.year;
```

dws_api_regional_house_i_d

表结构经过行转列（explode）的统一汇总操作，将多个表的内容经过格式调整后转变为类似于以下表结构的类型。city 字段为省份字段，year 为年份字段，price，area，revenue 分别为该城市当年的房屋平均价格，房屋总销售面积和房屋总销售额。

表 4-5 dws_api_regional_house_i_d 表结构

city	string
year	string
price	double
area	double
revenue	double

建表语句时使用 explode 结构先将时间类型转变为 map 型，并以视图的方式存在，然后利用该数据进行转变，将多列数据变为少列数据，并在更改结束后使用 join 以不同省份和年份将不同表进行关联。

```
create table dws.dws_api_regional_house_i_d as
select
t4.city as city
,t4.year as year
,t4.number as price
, t5.number as area
,t6.number as revenue
from
((select city,year,number
from
(select
city,map('2022',a2022,'2021',a2021,'2020',a2020,'2019',a2019,'2018',a2018,'2017',a2017,'2016',a2016,'2015',a2015,'2014',a2014,'2013',a2013) as value from dwd.dwd_api_regional_housing_prices_i_d) t1
lateral view explode(value) temp_view2 as year,number) as t4)
join
((select city,year,number
from
(select
city,map('2022',a2022,'2021',a2021,'2020',a2020,'2019',a2019,'2018',a2018,'2017',a2017,'2016',a2016,'2015',a2015,'2014',a2014,'2013',a2013) as value from dwd.dwd_api_regional_housing_sales_area_i_d) t2
lateral view explode(value) temp_view2 as year,number) as t5)
on t4.city = t5.city and t4.year = t5.year
join
((select city,year,number
from
(select
city,map('2022',a2022,'2021',a2021,'2020',a2020,'2019',a2019,'2018',a2018,'2017',a2017,'2016',a2016,'2015',a2015,'2014',a2014,'2013',a2013) as value from
```

```
dwd.dwd_api_regional_housing_sales_revenue_i_d) t3
    lateral view explode(value) temp_view2 as year,number) as t6)
    on t4.city = t6.city and t4.year = t5.year;
```

最后是 ADS 层的维度表，ADS 层一共分为五个维度，分别是

ads_api_annual_data_i_d，国家每年数据

表分为三个字段，分别是 business 业务，年份 year 和数值 number

表 4-6 ads_api_annual_data_i_d **表结构**

business	string
year	string
number	double

在编写过程中，使用 explode 将原本的表中的多列时间转变为一列时间列 year，并将原本的列作为数据填充进去。

```
create table ads.ads_api_annual_data_i_d as
select business,year,number
from
(select
business,map('2022',a2022,'2021',a2021,'2020',a2020,'2019',a2019,'2018',a2018,'2017',a2017,'2016',a2016,'2015',a2015,'2014',a2014,'2013',a2013) as value from dws.dws_api_annual_data_i_d) t2
    lateral view explode(value) temp_view2 as year,number;
```

ads_api_assets_liabilities_i_d，国家每年的企业负债率

表份三列，分别是 business 业务名字，year 时间和 number 当年的负债率数值。

表 4-7 ads_api_assets_liabilities_i_d **表结构**

business	string
year	string
number	double

在编写过程中，使用 explode 将原本的表中的多列时间转变为一列时间列 year，并将原本的列作为数据填充进去。由于该表内容独立，所以无需与其他表进行联

合。

```
create table ads.ads_api_assets_liabilities_i_d as
select business,year,number
from
(select
business,map('2022',a2022,'2021',a2021,'2020',a2020,'2019',a2019,'2018',a2018,'2017',a2017,'2016',a2016,'2015',a2015,'2014',a2014,'2013',a2013) as value from dws.dws_api_assets_liabilities_i_d) t2
lateral view explode(value) temp_view2 as year,number;
```

ads_api_business_operations_i_d，国家每年的企业营收

表结构是多个业务名的 business 列，时间 year 列和 number 列。number 列实际上是该业务当年的经济数值。

表 4-8 ads_api_business_operations_i_d 表结构

business	string
year	string
number	double

ads_house_complete_rate_i_d 国家各省施工竣工数和比例

表结构为和 DWS 层相似的五个列名，

表 4-9 ads_api_house_complete_rate_i_d 表结构

city	string
year	string
constructed	string
completed	string
rate	double

ads_regional_house_i_d 国家主要城市的销售价格，面积，和销售额。

city 字段为主要城市字段，year 为年份字段，constructed 和 completed 分别是当年该城市的施工和竣工面积，最后的 rate 字段是由前面的施工面积和竣工面积得到的概念的房屋竣工率。

表 4-10　ads_api_regional_house_i_d 表结构

city	string
year	string
price	double
area	double
revenue	double

4.2.3 ETL 数据迁移

数据迁移的作用是将Hive的维度表迁移到MySQL中，本部分使用的是DataX迁移工具。

在这部分中，首先要在 MySQL 中建立出和 Hive 的维度表相同的表结构。

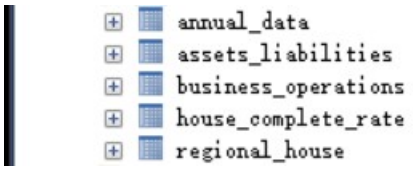


图 4-2 MySQL 中的表

其次，要根据每一个表写出一个专属的 json 文件使 DataX 可以识别和迁移，json 文件的 writer 和 reader 分别是 MySQL 的和 HDFS 类型的格式。并将 json 文件传入 Linux 虚拟机系统中进行保存并运行。

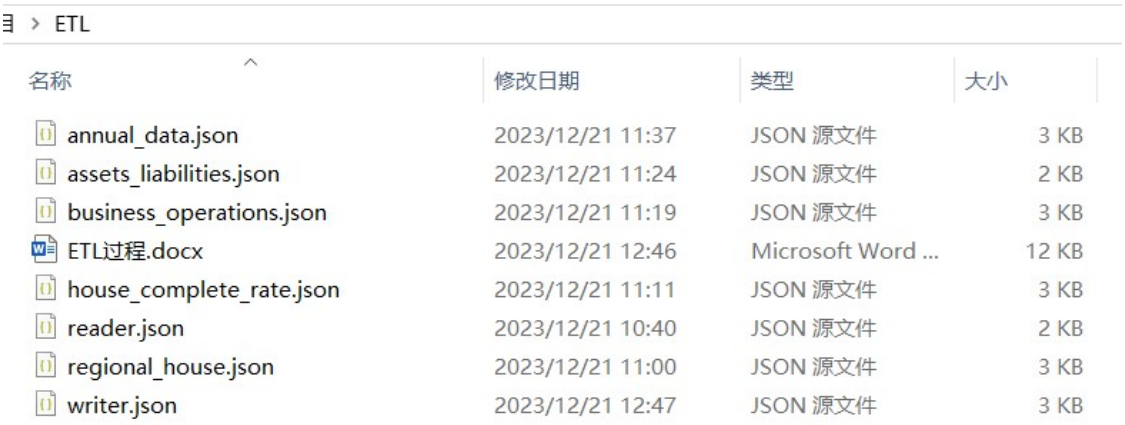


图 4-3 每个表的 json 文件

4.3 后端开发

在后端开发的过程中，采取了三层架构的方式，以 dao 层连接数据库，service 层作为中间层，并以 web 层作为开放端口的 Servlet，在 entity 层中为每个数据表创建实体，untils 层中包含着连接数据库所必须的包。

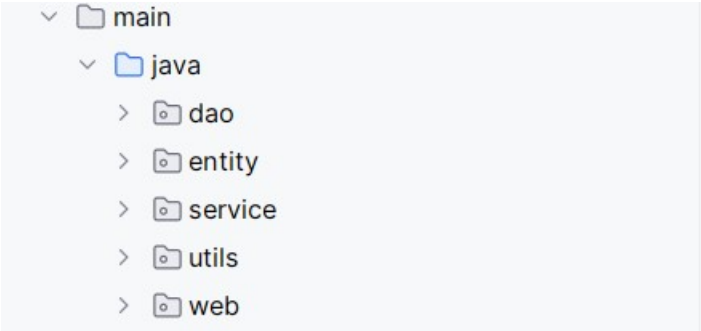


图 4-4 后端分层图

其中 dao 层和 service 层都以接口和实现类两部分组成，分别是 impl 包中的接口和外面的实现类。

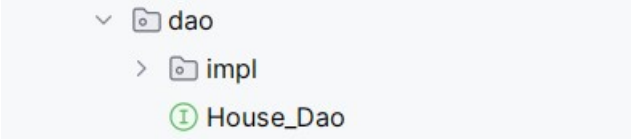


图 4-5 层的细分

在 web 层中，为每一个维度设置一个 Servlet 进行实现类的引用和端口和接口的开放，同时为了防止在访问过程中产生跨域问题，在 Servlet 中并设置过滤器文件。

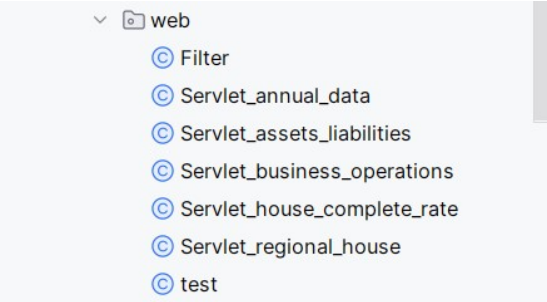


图 4-6 web 层视图

```
public void destroy() {
}
no usages
@Override
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws ServletException, IOE
    HttpServletRequest req = (HttpServletRequest) request;
    HttpServletResponse resp = (HttpServletResponse) response;
    resp.setHeader( s: "Access-Control-Allow-Origin",req.getHeader( s: "origin"));
    resp.setHeader( s: "Access-Control-Allow-Methods", s1: "*");
    resp.setHeader( s: "Access-Control-Allow-Headers", s1: "*");
    resp.setHeader( s: "Access-Control-Allow-Credentials", s1: "true");
    request.setCharacterEncoding("UTF-8");
    response.setCharacterEncoding("UTF-8");

    chain.doFilter(request,response);
}
```

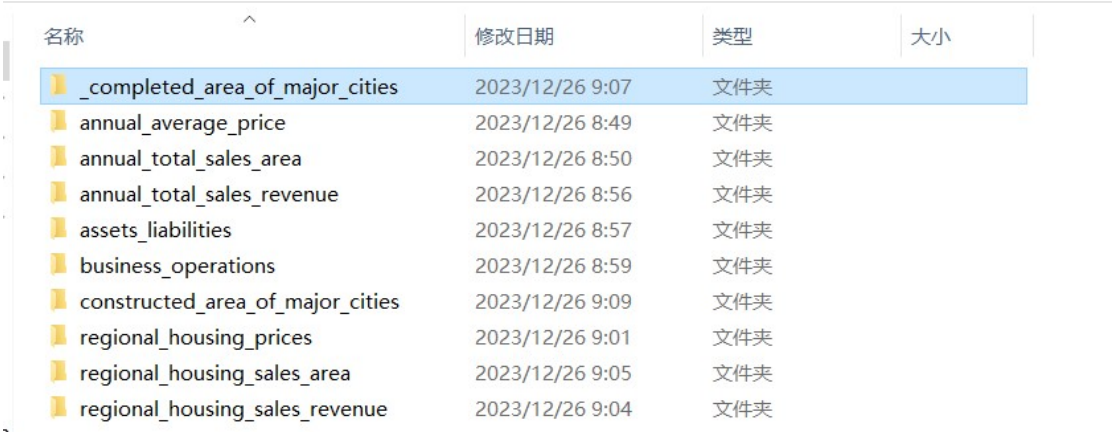
图 4-7 过滤器代码

4.4 任务调度

任务调度采用 Azkaban 完成，在任务调度的过程中使用 Azkaban 完成数据在 Hive 中的全过程，包括建表，数据存储，数据分层，数据整合，数据清洗，和数据变换。

由于在数仓分层的过程中将数仓分为了四层，所以在 Azkaban 中，将四层分为四个 job 分别运行。

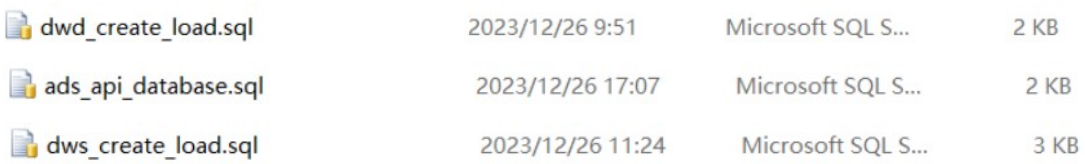
对于 ODS 层，为每一张表各做一张 SQL 文件



名称	修改日期	类型	大小
_completed_area_of_major_cities	2023/12/26 9:07	文件夹	
annual_average_price	2023/12/26 8:49	文件夹	
annual_total_sales_area	2023/12/26 8:50	文件夹	
annual_total_sales_revenue	2023/12/26 8:56	文件夹	
assets_liabilities	2023/12/26 8:57	文件夹	
business_operations	2023/12/26 8:59	文件夹	
constructed_area_of_major_cities	2023/12/26 9:09	文件夹	
regional_housing_prices	2023/12/26 9:01	文件夹	
regional_housing_sales_area	2023/12/26 9:05	文件夹	
regional_housing_sales_revenue	2023/12/26 9:04	文件夹	

图 4-8 Azkaban 实现 ODS 层

对于剩下的三层，每一层将所有 SQL 语句合并为一个 SQL 文件，最后剩下三个 SQL 文件。



dwd_create_load.sql	2023/12/26 9:51	Microsoft SQL S...	2 KB
ads_api_database.sql	2023/12/26 17:07	Microsoft SQL S...	2 KB
dws_create_load.sql	2023/12/26 11:24	Microsoft SQL S...	3 KB

图 4-9 Azkaban 实现剩下三层

然后根据 Azkaban 编写 flow 文件，组成压缩包后传入 Azkaban，分为四个 job

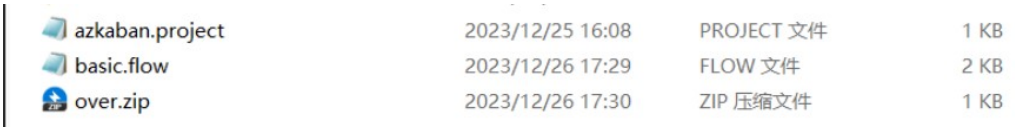


图 4-10 flow 文件和压缩包

四个 job 分别如下图所示，为防止内存不足导致 job 无法启动，所以将其设置为串行模式：

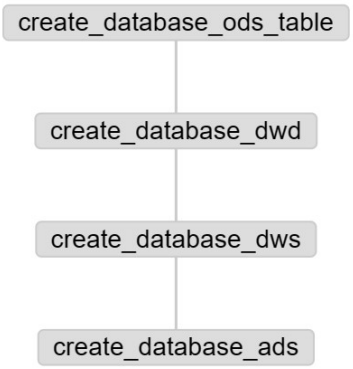


图 4-11 任务调度流程

当日志出现以下记录时则为运行成功结束。

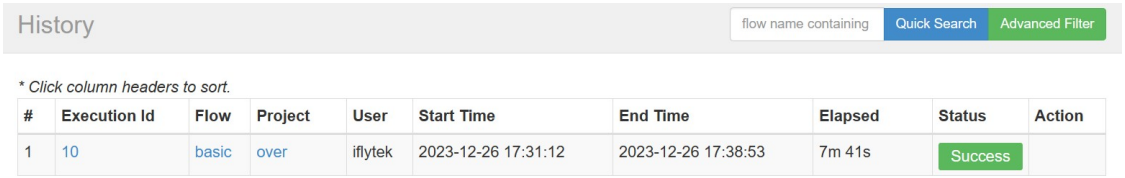


图 4-12 任务调度运行成功图示

5 调试分析

1. 问题：DataX 数据由 Hive 到 MySQL 中迁移时正常结束，但是写入为记录数量 0。

分析原因：在 Hive 的数据分层中，直接使用了不定长的字符串类型 String，而在 MySQL 中，使用了定长的类型 varchar，可能由于定长的设置，导致了字符串迁移过程中过长写入失败。

解决办法：将 MySQL 中响应的字符串类型字段的 varchar 长度增加。

2. 问题：tomcat 运行失败问题，无法创造实体类

分析原因：JavaWeb 配置环境出现问题，导致无法应用实体类来完成辅助工具类。

解决方法：重新配置环境并在 poem 文件中配置需要的 jar 包，并配置辅助工具类。

3. 问题：Azkaban 使用，出现权限问题，无法使用数据库和创造文件

分析原因：Azkaban 配置文件中配置的数据库使用用户为 Azkaban，不是 root 权限，导致一些功能无法使用。

解决方法：将 Azkaban 文件中的 MySQL 数据库的用户改为 root，并配置响应的账号和密码。

4. 问题：Azkaban 中分配调度工作无法运行

分析问题：内存配置不足，导致任务调度内存分配不够，不能启动任务，导致超时无法运行。

解决方法：给虚拟机增加内存，使其能够到达任务调度启动的最小要求。

5. 问题：Azkaban 任务调度启动后节点无法完成相应任务

分析问题：Azkaban 的任务会按照某种规则分配给节点，并让节点来进行完成，但是节点中可能并没有安装相应的软件，导致分配到该节点的任务无法启动运行。

解决方法：在 Linux 集群中对 master 和 slave 进行免密培训，使得 slave 节点可以直接访问到 master 节点，并在 job 的配置中，用到 slave 节点中没有的功能的时候使用 ssh 来用 master 节点进行功能的完成。

6 总结

优点：这次课设，熟悉了很多的 Hive 实用函数和视图的使用方法，对于一些 Hive 直接操作数据进行复杂操作也有了一些经验，同时了解了类似于 Azkaban 的任务调度的使用，了解到各个阶段语句的精简重要性。同时复习了 Spark，Python 数据分析，数据挖掘，DataX 数据迁移，Hive 内外部表的转换知识。了解了 Hive 中的行转列函数 explode，并在 explode 的学习过程中，拓展了解 lateral view 视图和 Hive 中的复杂架构 Array 和 map。

缺点评估：在本次的课程项目制作中，发现对于之前的 Spark 知识忘记很多，甚至环境搭配都需要花费很长时间，虽然最后采取了使用 Hive 进行分层和整合。后端接口方面也类似，一学期没有接触后，在做课设的时候类似从头学起。而这

次过程中的主要阶段 Hive，使用中发现，Hive 的熟练使用仅限于基础功能阶段，对于一些函数和视图方面，了解大有不足，且在做 Azkaban 的过程中，发现 Hive 语句过于零散，需要优化和整合，虽然最后整合成功，不过却发现了很大的语句规范问题。

7 参考文献

[1] 房地产大数据及其信息挖掘体系构建路径研究[J]. 隆林宁.产业与科技论坛,2023(04)
[2] 互联网+”背景下众筹模式在房地产开发项目中的应用[J]. 郭庆军;白思俊;张华波;朱海阔.项目管理技术,2016(01)
[3] 用互联网思维和手段创新房地产业发展[J]. 刘志峰.住宅产业,2015(11)
[4] 房地产行业“互联网+”模式研究与探讨[J]. 夏阳.中国房地产,2015(13)
[5] 大数据时代:房地产业的机遇与挑战[J]. 刘昱;张玉娟.河南商业高等专科学校学报,2013