股票数据管理系统

设计一个股票数据管理系统需要兼顾数据采集、存储、处理、查询、分析及可视化等核心需求，同时需考虑系统的可扩展性、实时性和易用性。以下是一个完整的设计方案：

### **一、系统目标**

1. 实现股票数据的全生命周期管理（采集、清洗、存储、更新）。
2. 支持多维度数据查询（按股票代码、时间、指标等）。
3. 提供基础数据分析（技术指标计算、基本面分析）和可视化功能。
4. 满足不同用户（个人投资者、分析师）的个性化需求（如关注列表、预警）。

### **二、系统架构设计**

采用分层架构设计，确保模块解耦和可扩展性：

1. 用户层 → 前端层 → 后端服务层 → 数据处理层 → 数据存储层 → 数据源层 **用户层**
   1. 个人投资者：需查询行情、查看分析图表、设置预警。
   2. 分析师：需批量数据导出、自定义指标计算、多股票对比。
   3. 管理员：需管理数据源、用户权限、系统配置。
2. **前端层**
   1. **技术选型**：React（框架）+ ECharts/Highcharts（可视化）+ WebSocket（实时推送）。
   2. **核心功能**： 股票行情展示（实时/历史K线图、分时图）。
      1. 数据查询界面（多条件筛选：代码、时间范围、指标）。
      2. 分析工具（技术指标面板、基本面数据表格）。
      3. 个人中心（关注列表、预警设置、历史查询记录）。
3. **后端服务层**
   1. **技术选型**：SDjango（Python）+ Redis（缓存）+ JWT（认证）。
   2. **核心服务**： **用户服务**：注册、登录、权限管理（基于RBAC模型）。
      1. **数据查询服务**：接收前端请求，从数据库/缓存返回数据（支持分页、筛选）。
      2. **分析服务**：实时计算技术指标（如MACD、RSI、均线）、基本面指标（市盈率、市净率）。
      3. **预警服务**：监听行情数据，触发用户设置的阈值（如股价涨跌幅超5%），通过短信/APP推送。
4. **数据处理层**
   1. **数据采集**： 对接第三方数据源（如雅虎财经API、Tushare、聚宽、Wind接口，akshare，baoshare）。
      1. 定时任务（如每1分钟拉取实时行情，每日凌晨同步历史数据和财务数据）。
   2. **数据清洗**： 处理缺失值（用前值填充或标记）、异常值（如股价为负，触发告警）。
      1. 格式标准化（统一时间戳、数值精度，如保留2位小数）。
   3. **数据计算**： 预计算高频指标（如5/10/20日均线），减少实时查询压力。
5. **数据存储层**
   1. 采用混合存储方案，兼顾性能和成本：mongodb： 存储股票基础信息（代码、名称、行业、上市日期等）。
      1. 用户信息、关注列表、预警配置等结构化数据。
   2. **时序数据库（InfluxDB/TimescaleDB）**： 存储高频行情数据（时间戳、开盘价、收盘价、成交量等），优化时间序列查询性能。
   3. **缓存（Redis）**： 缓存热门股票实时行情（如近1小时数据）、用户频繁查询的指标结果。
6. **数据源层**
   1. 免费数据源：雅虎财经、新浪财经、Tushare（基础版）、akshare、baoshare。
   2. 付费数据源：Wind、聚宽（高级版）、同花顺API（适合专业场景）。只需要有界面不需要实现。
7. **数据更新：**
   1. 自动更新：采集脚本触发时同步更新原始数据 + 衍生指标（如采集日线后立即计算均线）。
   2. 手动更新：接收前端编辑请求（如用户修正错误数据），Python 脚本处理后更新数据库。
   3. 三、核心功能模块设计

#### 1. 数据采集与管理模块

* **自动采集**： 实时行情：每1分钟调用API拉取数据，通过WebSocket推送到前端。
  + 历史数据：每日凌晨同步前1日K线数据，每周同步周线/月线数据。
  + 财务数据：每季度同步上市公司财报（营收、净利润、资产负债等）。
* **手动管理**： 支持Excel/CSV批量导入历史数据（需校验格式）。
  + 管理员可手动修正异常数据（带操作日志，便于追溯）。

#### 2. 数据查询模块

* **基础查询**： 按股票代码/名称模糊搜索，返回基础信息+最新行情。
  + 按时间范围查询历史K线（支持日线、周线、月线切换）。
* **高级筛选**： 多条件组合（如“行业=半导体”+“市盈率<30”+“近30日涨幅>10%”）。
  + 指标筛选（如“MACD金叉”“RSI>70超买”）。

#### 3. 数据分析模块

* **技术分析**： 内置指标：均线（MA）、MACD、RSI、KDJ、布林带等，支持自定义周期（如5日/20日）。
  + 图表联动：K线图与指标图同步缩放、高亮显示关键节点（如金叉/死叉）。
* **基本面分析**： 财务指标计算：市盈率（PE）、市净率（PB）、净利润增长率、ROE等。
  + 对比分析：多只股票的财务指标横向对比（如“茅台vs五粮液近3年ROE”）。

#### 4. 个性化功能模块

* **关注列表**：用户可添加关注股票，首页展示聚合行情。
* **预警设置**：支持股价、涨跌幅、成交量等阈值预警（如“当贵州茅台股价>1800元时提醒”）。
* **数据导出**：支持将查询结果/分析图表导出为Excel、PDF或图片。

### **四、数据库设计（核心表）**

#### 1. 股票基础信息表（stock\_basic）

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **说明** |
| stock\_code | VARCHAR(10) | 股票代码（主键，如600036） |
| stock\_name | VARCHAR(20) | 股票名称（如招商银行） |
| industry | VARCHAR(50) | 所属行业（如银行业） |
| market | VARCHAR(10) | 市场（沪市/深市/港股/美股） |
| listing\_date | DATE | 上市日期 |

#### 2. 行情数据表（stock\_quote，时序库存储）

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **说明** |
| stock\_code | VARCHAR(10) | 股票代码（外键） |
| timestamp | DATETIME | 时间戳（精确到分钟） |
| open\_price | DECIMAL(10,2) | 开盘价 |
| close\_price | DECIMAL(10,2) | 收盘价 |
| high\_price | DECIMAL(10,2) | 最高价 |
| low\_price | DECIMAL(10,2) | 最低价 |
| volume | BIGINT | 成交量（股） |

#### 3. 财务数据表（stock\_finance）

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **说明** |
| id | INT | 自增主键 |
| stock\_code | VARCHAR(10) | 股票代码（外键） |
| report\_date | DATE | 财报日期（如2023Q3） |
| revenue | DECIMAL(20,2) | 营业收入（万元） |
| net\_profit | DECIMAL(20,2) | 净利润（万元） |
| pe | DECIMAL(8,2) | 市盈率 |
| pb | DECIMAL(8,2) | 市净率 |

#### 4. 用户关注表（user\_follow）

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **说明** |
| id | INT | 自增主键 |
| user\_id | INT | 用户ID（外键） |
| stock\_code | VARCHAR(10) | 股票代码（外键） |
| follow\_time | DATETIME | * 1. 关注时间 |

### **数据流程示例**

#### 自动采集与更新流程

graph LR

* A[Python定时任务] --> B[调用第三方API拉取数据]
* B --> C[Python清洗数据（去重/补缺失）]
* C --> D[计算衍生指标（如MA5）]
* D --> E[调用后端API写入InfluxDB/MySQL]
* E --> F[前端通过WebSocket实时刷新展示]

#### 2. 手动更新数据流程

graph LR

* A[前端页面编辑数据（如修正收盘价）] --> B[提交数据到后端API]
* B --> C[后端校验数据格式（Python逻辑）]
* C --> D[Python脚本更新数据库]
* D --> E[返回更新结果给前端]
* E --> F[前端刷新页面显示最新数据]

### **五、技术亮点与优化**

1. **实时性优化**： 行情数据采用WebSocket推送（减少轮询开销），Redis缓存热门股票数据（查询延迟<100ms）。
2. **查询性能**： 时序库按“股票代码+时间范围”分区，索引优化（如InfluxDB的tag索引）。
   1. 预计算高频指标（如均线），存储在时序库中，避免实时计算压力。
   2. 可扩展性： 数据源模块设计为插件化（通过配置文件新增API接口），支持动态切换数据源。
   3. 后端服务水平扩展（通过负载均衡应对高并发查询）。
3. **可靠性**： 数据采集失败时触发重试机制（最多3次），失败日志存入ELK系统，便于排查。
   1. 数据库定时备份（每日全量+增量备份），支持数据回滚。

### **六、应用场景**

* 个人投资者：通过关注列表实时跟踪持仓股票，用技术指标辅助决策。
* 分析师：批量导出多只股票的财务数据，进行行业对比分析，生成报告。
* 量化交易团队：对接系统API获取历史数据，用于策略回测。

该系统可根据实际需求逐步迭代（如增加AI预测模块、多语言支持等），核心是确保数据的准确性和查询分析的高效性。