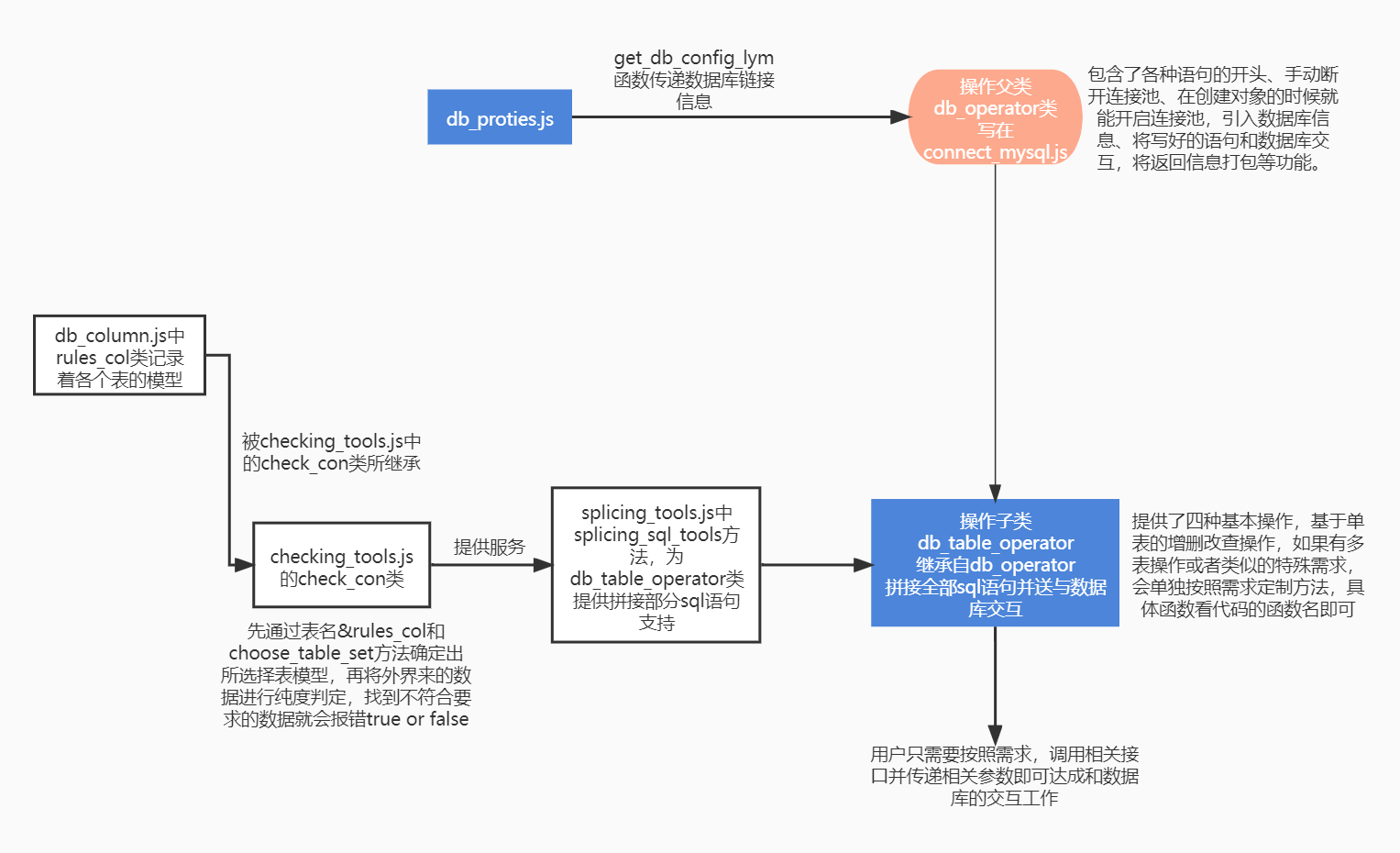
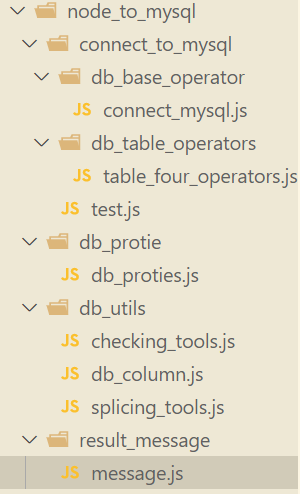
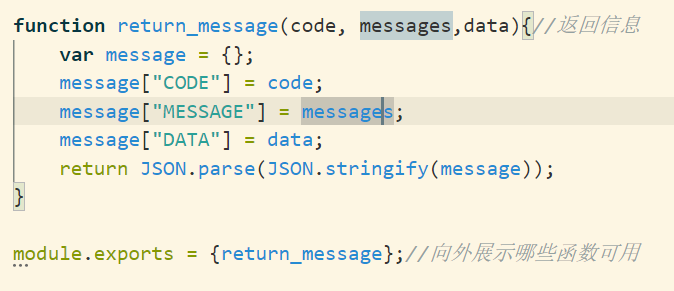
1、整体设计：



文件结构

2、返回信息类



3、dynamic\_select\_one\_table\_sql(select\_con,where\_con,table\_name)

为向用户提供的查询接口

参数：

<1>拼接select部分语句所用的信息，格式为：

**var** l = [ 'id', 'password'];

其内部为想构造的结果集样式，名称一定要和表的属性名一致。

<2>拼接where部分语句所用的信息，格式为：

**var** item = {

    user\_name : {

        operator:'=',

        value:'lwc'

    },

    id : {

        operator:'=',

        value:'16'

    }

};

不可以有任何一处为undefined、operator和value咬拼对，外层key要求和上面一个参数一致。

<3> 表名，用于选择模型来校验数据。

4、dynamic\_insert\_one\_table\_sql(insert\_con,table\_name,primary\_key,attribute\_num)

<1>构建insert部分sql时候所用数据，属性名(key)不可空，value不能是undefined

格式：

**var** items = {

    interface\_id : '10',

    rules\_id : '8'

}

<2>表名，用于选择模型来校验数据

<3>主键列表，用于进行重复插入判断，加入的是主键列表！！！！！！！！！！不要加其他属性进去！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！

<4>表中的属性个数，用来判断给的数据是否包含了全部数据(根据需求约定，插入时数据必须全都得有)

5、

dynamic\_update\_one\_table\_sql(update\_con,where\_con,table\_name)

<1>拼接set部分语句用的数据，属性值(key)不能写错，value不可空，操作符为=(不变的)

格式：

**var** items = {

    id : '16',

}

<2>拼接where时候语句，格式见上。

<3>表名，用于选择模型来校验数据

6、

dynamic\_delete\_one\_table\_sql(where\_con,table\_name)

<1>拼接where时候语句，格式见上。

<2>表名，用于选择模型来校验数据

值得注意的是，这是逻辑删除，内部使用update语句.

7、调用举例

target.dynamic\_select\_one\_table\_sql(l,item,"user").then(res **=>** {*//查询*

    console.log(res);

    target.close\_pool();

})

target.dynamic\_insert\_one\_table\_sql(items,"interface\_rules",['interface\_id','rules\_id'],2).then(res **=>** {

    console.log(res);

    target.close\_pool();

})

target.dynamic\_update\_one\_table\_sql(items,item,"user").then(res **=>** {*//更新*

    console.log(res);

    target.close\_pool();

})

target.dynamic\_delete\_one\_table\_sql(item,"user").then(res **=>** {*//删除*

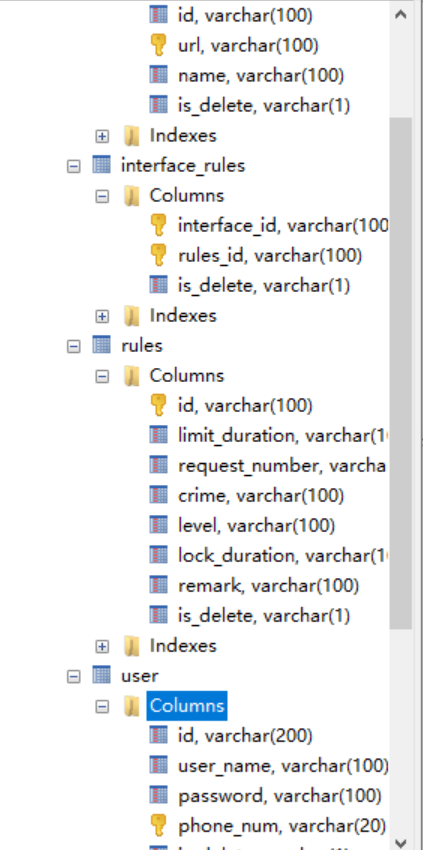
    console.log(res);

    target.close\_pool();

})

res 即为结果；一个对象有一个连接池，该对象不用了之后，要手动关闭连接池。

8、数据库设计说明：



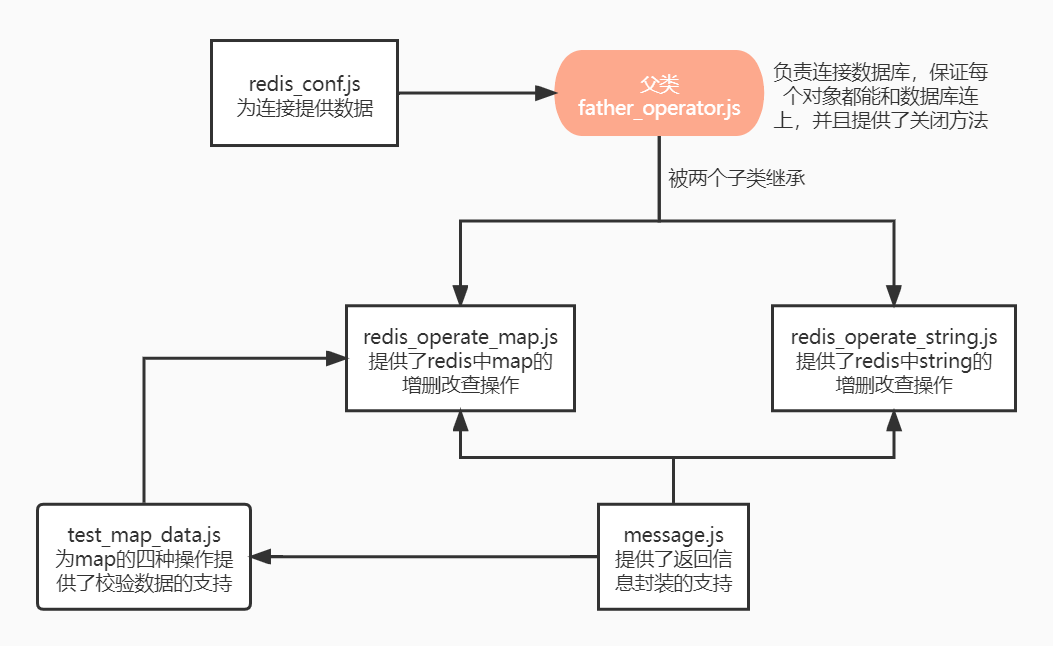
interface表主键 url

interface\_rules 主键 interface\_id+rules\_id

rules表主键 id

user 表主键 phone\_num

Redis库操作设计：



1. 文件结构：



1. redis\_operate\_map.js接口说明：

<1>insert\_into\_redis(table\_name,data,limit\_time)

向redis插入数据，参数：

第一个：表名(String)，作为key

第二个：数据(map<phone,Applycode>)

第三个：有效时间(秒)

功能：插入数据，更新数据(只要键值一样，重复插入就是更新)

校验：所有部分数据不能是undefined

<2>select\_from\_redis(table\_name)

向redis查询数据，参数：

第一个：表名(作为key值在数据库里查询)

功能：查询数据

校验：所有部分数据不能是undefined

正确返回：一串JSON数据

<3> delete\_from\_redis(table\_name,field\_list)

在redis里面删除键值对，参数：

第一个，表名

第二个：key对应的值域列表，要写全，针对本项目是两个：phone,Applycode

功能：删除数据

校验：所有部分数据不能是undefined+对field\_list是否正确以及是否写全的校验。

1. redis\_operate\_string.js接口说明：

<1> insert\_into\_redis(table\_name,data,limit\_time)

向redis插入数据，参数：

第一个：表名(String)，作为key

第二个：数据(String)，作为value

第三个：有效时间(秒)

功能：插入数据，更新数据(只要键值一样，重复插入就是更新)

校验：所有部分数据不能是undefined

<2> select\_from\_redis(table\_name)

向redis查询数据，参数：

第一个：表名(作为key值在数据库里查询)

功能：查询数据

校验：所有部分数据不能是undefined

正确返回：一串JSON数据

<3> delete\_from\_redis(table\_name)

在redis里面删除键值对，参数：

第一个，表名

功能：删除数据

校验：所有部分数据不能是undefined