银行业务数据分析报告

数据仓库与数据挖掘第一次作业

2018-10-14

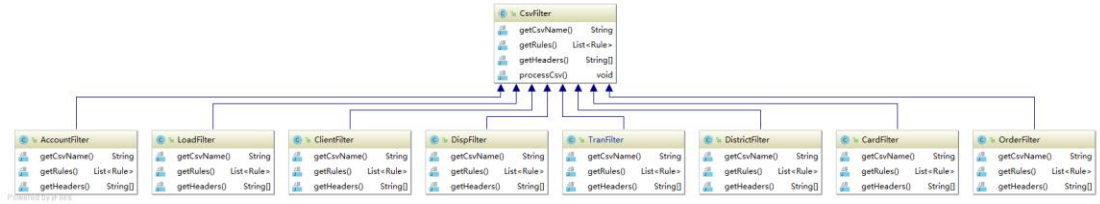
崔浩
2018214160

目录

一、数据清洗

第一步数据清洗的目的是分辨 CSV 文件中单个数据字段的问题，数据之间的**逻辑关系**问题在后续**数据导入时再处理**，在这一步**不考虑**。本实验通过 Java 程序进行初步的数据清洗，算法复杂度为 O(n)，在一次遍历后识别字段问题，尝试修复或丢弃数据。

1.1 Java 程序架构



CsvFilter 类定义了数据处理的模板方法，每个表只需要定义 Csv 文件的名字和处理每一个字段的规则就可以，核心的预处理方法为 processCsv()方法，方法的主要流程为：

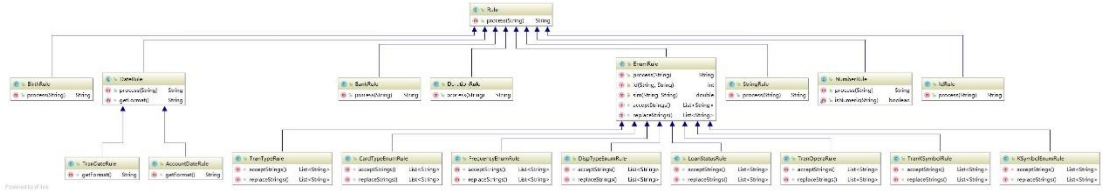
1）根据子类定义的 getCsvName()获取当前要处理的 csv 文件，创建读取和写出类。

2）根据子类定义的 getRules()获取当前 csv 文件每一个字段的处理规则

3）循环读取每一行数据，对每一段数据应用规则，如果返回空值则舍弃这一行数据

4）将处理好的数据写入新文件，将丢弃的数据写入丢弃日志，便于手动修改。

文件处理规则有 Rule 类定义，涵盖日期、数字、枚举等类型的数据，实现统一的 process 接口。有关 Rule 的层级结构如下图所示：



规则被分为生日/性别规则、日期规则、银行字段规则、还款期数规则、枚举字段规则、普通字符串规则、数字规则、ID 字段规则，部分规则会根据字段的具体的情况划分为更具体的规则，有关规则的定义，会在下文中提及。

1.2 清洗规则定义

1.2.1 枚举规则

本次实验在处理枚举类型的数据时，首先查找该字段与已定义字段是否相等，如果相等则输出翻译后的字段值，如果不相等则使用编辑距离算法计算该字段与已定义所有字段的文本相似度（Levenshtein Distance），选取最相似的字段输出。

关键调度算法为：

```
@Override
public String process(String item) {
    List<String> acceptStrings = acceptStrings();//获取子类提供的定义字段
    List<String> replaceStrings = replaceStrings();//获取子类提供的翻译字段
    for (int i = 0; i < acceptStrings.size(); i++) {//相同则直接返回翻译字段
        if (item.equals(acceptStrings.get(i))) return replaceStrings.get(i);
    }
    int maxIndex = 0;
    double maxValue = 0;
    for (int i = 0; i < acceptStrings.size(); i++) {//不相同则计算编辑距离返回相似字段
        double simValue = sim(acceptStrings.get(i).toUpperCase(), item.toUpperCase());
        if (simValue > maxValue) maxIndex = i;
    }
    return replaceStrings.get(maxIndex);
}
```

通过计算编辑距离对部分字段进行修复，将长度为 i 的字符串通过替换、删除和增加三种操作转换为长度为 j 的字符串的操作数成为这两个字符串的编辑距离，编辑距离越短活命这两个字符串越相似。编辑距离的计算公式为：

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

编辑距离的计算考虑三种情况：

1、可以在 k 个操作内将 s1[1…i]转换为 s2[1…j-1]，s2 加上一个字符就可以，距离为 k+1

2、可以在 k 个操作内将 s1[1…i-1]转换为 s2[1…j]，s1 移除最后字符就可以，距离为 k+1

3、可以在 k 个操作内将 s1[1…i-1]转换为 s2[1…j-1]，如果 s1[i]=s[j]，则距离为 k，否则需要一次替换操作，距离为 k+1

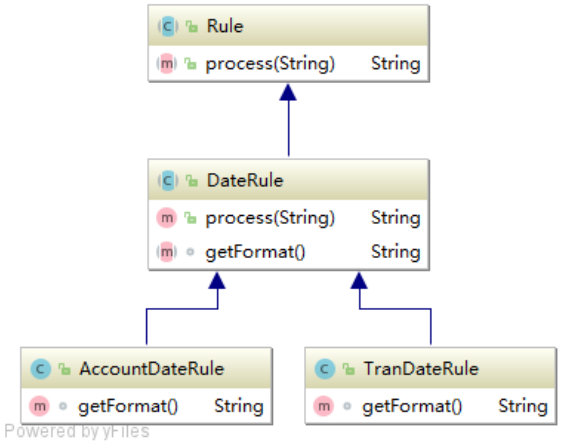
算法的关键代码为：

```
for (int i = 1; i <= n; i++) {
    ch1 = str1.charAt(i - 1);
    for (int j = 1; j <= m; j++) {
        ch2 = str2.charAt(j - 1);
        if (ch1 == ch2) {
            temp = 0;
        } else {
            temp = 1;
        }
        d[i][j] = Math.min(d[i - 1][j] + 1, Math.min(d[i][j - 1] + 1, d[i - 1][j - 1]+ temp));
    }
}
```

在修复字段之后，还需要对现有的语言进行翻译，翻译的字段表如下所示：

枚举类型	定义字段	翻译字段
CardType	JUNIOR	JUNIOR
	CLASSIC	CLASSIC
	GOLD	GOLD
DispType	OWNER	OWNER
	DISPONENT	DISPONENT
Frequency	POPLATEK MESICNE	ONCE A MONTH
	POPLATEK TYDNE	ONCE A WEEK
	POPLATEK PO OBRATU	AFTER DEAL
KSymbol	POJISTNE	INSURANCE
	SIPO	MANAGEMENT
	LEASING	RENTAL
	UVER	LOAN
TranKSymbol	POJISTNE	INSURANCE
	SLUZBY	BILL
	UROK	INTEREST_INCOME
	SANKC.UROK	INTEREST_PENALTY
	SIPO	HOUSEHOLD
	DUCHOD	OLD_AGE_PENSION
	UVER	LOAN
TranOperation	VYBER KARTOU	CREDIT_WITHDRAW
	VKLAD	CREDIT_CASH
	PREVOD Z UCTU	FROM_OTHER_BANK
	VYBER	CASH
	PREVOD NA UCET	TO_OTHER_BANK
TranType	PRIJEM	DEPOSIT
	VYDAJ	WITHDRAW

1.2.2 日期规则



日期规则定义了统一的输出的格式，输入格式则可以根据不同的 csv 文件字段具体定义，使用 DateFormat 尝试将 csv 字段中的字符串转换为 Date 类型，如果转换失败则丢弃这条字段。

日期规则的处理代码如下所示：

```
public abstract class DateRule extends Rule {

    @Override
    public String process(String item) {
        String dateFormat = getFormat();//由子类实现输入日期格式
        DateFormat format = new SimpleDateFormat(dateFormat, Locale.getDefault());
        DateFormat toFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
Locale.getDefault());
        try {
            Date date = format.parse(item);
            return toFormat.format(date);
        } catch (ParseException e) {
            System.err.println("Date format exception: date-"+item);
        }
        return null;
    }

    abstract String getFormat();
}
```

1.2.3 生日和性别

原 CSV 文件使用了特殊的生日表示法，若客户为女性，则月数表示的时候加 50，这一字段需要拆分为两个字段，为正常显示的生日日期和性别字段。

在本框架内数据预处理是按照每一个字段的规则来处理的，如何在处理一个字段的同时添加新的字段呢？本实验的处理方式是在处理完的字段后插入一个定义的分隔符，在分隔符后加入新的字段值，处理整条数据时程序会自动检测分隔符，将追加的字段加到所有字段后。字段的数量不作限制，不过需要在写 csv 表格的子类 Filter 时定义好追加字段的 header。

本实验处理 Birth 字段之后，除了恢复为正常的生日字段以外，还追加了性别、年龄和年龄类型字段，年龄类型的值为年龄/10 的值，不超过 6。

处理该字段的关键代码如下：

```
public class BirthRule extends Rule {
    @Override
    public String process(String item) {
        try {
            if (item.length() == 6) {
                int isMale = 1;
                int month = Integer.parseInt(item.substring(2,4));//获取月份值
                if (month > 50) {
```

```
                    month -= 50;
                    isMale = 0;//如果月份大于 50，说明性别是女性
                }
                Calendar calendar = Calendar.getInstance();
                calendar.set(Integer.parseInt("19"+item.substring(0,2)),month,Integer.p
arseInt(item.substring(4)));//设置正确的日期
                SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
                String date = format.format(calendar.getTime());
                int age = 2000-calendar.get(Calendar.YEAR);//计算年龄
                int ageType = age/10;
                if (ageType > 6) ageType = 6;
                return date + CsvFilter.SPLIT_TAG + isMale + CsvFilter.SPLIT_TAG + age
+ CsvFilter.SPLIT_TAG + ageType;//使用分隔符分隔追加的字段
            }
        } catch (Exception e) {//日期格式错误则返回 null，也就是丢弃这条数据
            return null;
        }
        return null;
    }
}
```

1.2.4 数字规则

对于有长度限制的数字，统一使用整数的转换方法 Integer.parseInt(item)，如果抛出格式错误异常则丢弃这条数据。对于没有长度限制的数字，则依次检查每一位是否为数字类型。

检查每一位数字的代码如下所示：

```
public static boolean isNumeric(String str){
    for (int i = str.length();--i>=0;){
        if (!Character.isDigit(str.charAt(i))){
            return false;
        }
    }
    return true;
}
```

1.2.5 其他规则

数据清洗还考虑了以下规则：

1. BankRule：银行名字由两位字母组成
2. DurationRule：还贷期数为 12 的倍数，将还贷期数离散化，替换为除以 12 的值

在第一步的数据清理过程中只考虑了每个字段的合法性，而没有考虑字段之间的逻辑和字段整体的合法性，因此需要在数据导入后使用 SQL 语句检查字段之间的合法性，包括主键唯一性、外键合法性、字段之间逻辑合法性等。

二、数据导入

2.1 导入后的数据清洗

在将 CSV 文件导入到数据库之后，需要检查主键的重复情况，外键的合法情况，字段间的逻辑情况，在检查这些情况并处理后才能开始修改表结构和外键。

2.1.1 主键唯一性检查

以 account 表为例，首先检查每个表的 ID 是否有重复：

```
select * from account where account_id in (select account_id from
account group by account_id having
count(account_id) > 1)
```

	account_id	district_id	frequency	date
1	576	55	ONCE A MONTH	1993-01-01 00:00:00
2	576	55	ONCE A MONTH	1993-01-01 00:00:00

随后删除并修复数据：

```
delete from account where account_id=576;
insert into account (account_id, district_id, frequency, date) values
('576','55','ONCE A MONTH','1993-01-01 00:00:00')
```

2.1.2 外键合法性检查

以 trans 表为例，首先检查无法对应的字段：

```
select * from trans where account_id not in (select account_id from
account)
```

	trans_id	account_id	date	type	operation	amount	balance	k_symbol	bank	account
1	999799	8030	1998-06-07	WITHDRAW	TO_OTHER_BANK	139	10258		KL	48610542
2	999804	8045	1998-06-09	WITHDRAW	TO_OTHER_BANK	139	80258		KL	48610547
3	999807	8093	1998-06-12	WITHDRAW	TO_OTHER_BANK	139	1042		KL	48610549
4	999810	9139	1998-06-13	WITHDRAW	TO_OTHER_BANK	139	25894		KL	48610557
5	999815	10162	1998-06-17	WITHDRAW	TO_OTHER_BANK	139	8965		KL	48610598
6	999817	10245	1998-06-17	WITHDRAW	TO_OTHER_BANK	139	6945		KL	48610604
7	999835	10725	1998-06-19	WITHDRAW	TO_OTHER_BANK	139	1245		KL	48610609
8	999839	11321	1998-06-21	WITHDRAW	TO_OTHER_BANK	139	90876		KL	48610632

然后删除这些字段：

```
delete from trans
where account_id not in (select account_id from account)
```

2.1.3 字段间逻辑检查

检查每期还款额度*还款期数是否等于还款总额：

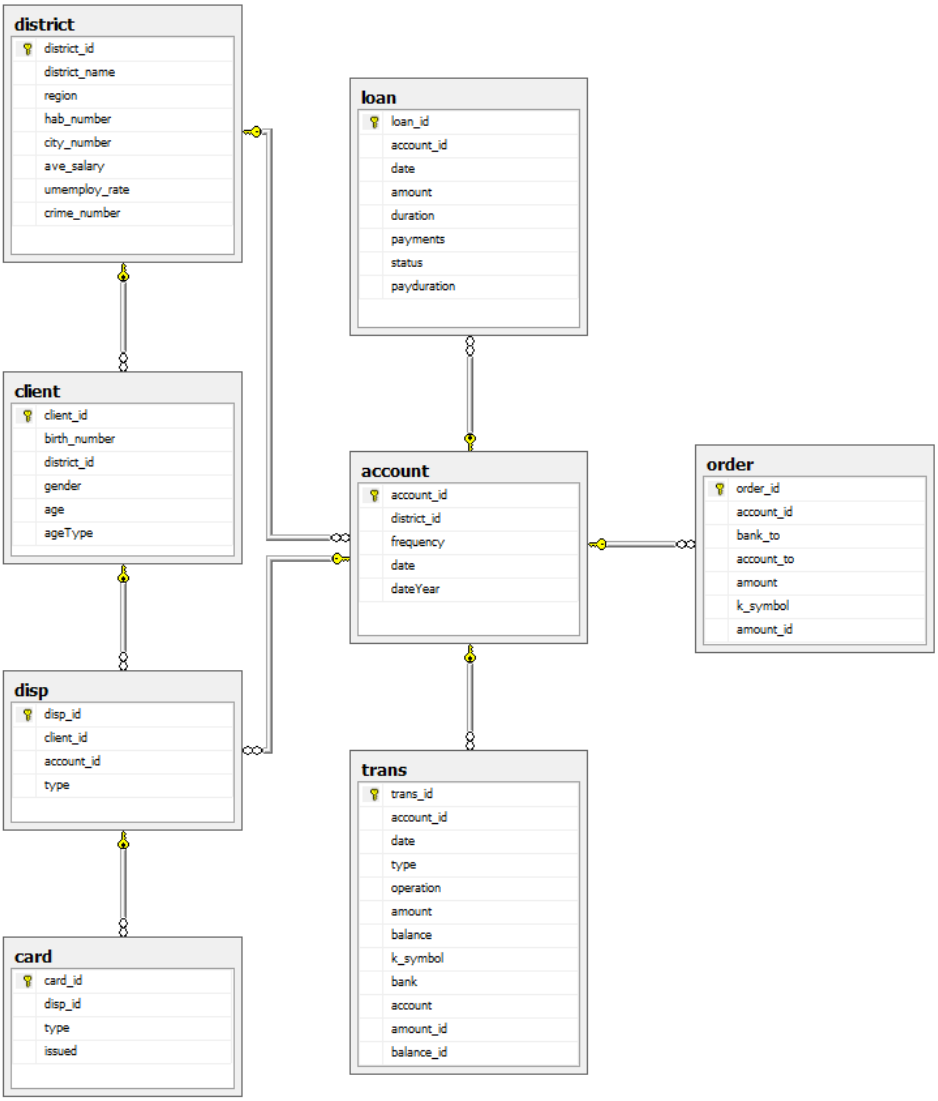
```
select * from loan where payments*duration*12 != amount;
```

	loan_id	account_id	date	amount	duration	payments	status	payduration
1	5495	2580	1996-08-02	89340	-5	1489	C	50
2	7130	10411	1996-12-11	123402	2	5142	A	17

```
update loan set duration=5 where loan_id=5495;
update loan set amount=duration*12*payments where loan_id=7130;
```

2.2 完善表格结构

在清洗数据之后需要设置表格主键、表格属性的数据类型、表的外键等，为后续的统计需要新增和计算部分新的字段，最终表的结构图如下所示:



三、维度表和事实表

建立年龄、信用卡类型、性别、操作类型、订单额平均值、订单类型、地区、交易类型、交易额平均值、账户余额、年份等维度，建立用户数、交易金额和贷款总额事实表。

3.1 维度表

3.1.1 年龄维度

列名	数据类型	含义
age_id	int	年龄维度 id
age_value	varchar(50)	年龄的范围解释，如'10~20'
age_group	varchar(50)	所属的群体，如老年、青年

分类的详细信息为：

age_id	age_value	age_group
0	0~9 岁	小孩
1	10~19 岁	少年
2	20~29 岁	青年
3	30~39 岁	壮年
4	40~49 岁	壮年
5	50~59 岁	壮年
6	60~岁	老年

3.1.2 信用卡类型维度

列名	数据类型	含义
card_id	int	维度 id
card_chinese	varchar(50)	信用卡类型中文翻译
card_value	varchar(50)	信用卡类型值：金卡、普通卡……

分类的详细信息为：

card_id	card_value	card_chinese
1	GOLD	金卡
2	CLASSIC	普通卡
3	JUNIOR	初级卡

3.1.3 性别维度

列名	数据类型	含义
gender_id	int	维度 id
gender_value	varchar(50)	性别值：男，女

3.1.4 操作类型维度

列名	数据类型	含义
operation_id	int	维度 id
operation_chinese	varchar(50)	操作类型中文翻译
operation_value	varchar(50)	操作类型值：信用卡取现……

分类的详细信息为：

operation_id	operation_value	operartion_chinese
1	CREDIT_WITHDRAW	信用卡取现
2	CREDIT_CASH	信用卡现金
3	FROM_OTHER_BANK	其他银行汇款
4	CASH	提取现金
5	TO_OTHER_BANK	汇款到其他银行

3.1.5 收款订单额维度

列名	数据类型	含义
order_amount_id	int	维度 id
amount_value	varchar(50)	订单额/1000 的数值，不超过 11
amount_group	varchar(50)	对订单额的描述，如很少、很多……

分类的详细信息为：

order_amount_id	amount_value	amount_group
-1	<=-500	取款
0	-499~499	较少
1	500~1499	较少
2	1500~2499	中等
3	2500~3499	中等
4	3500~4499	中等
5	4500~5499	较多
6	5500~6499	较多
7	6500~7499	较多
8	7500~8499	很多

9	8500~9499	很多
10	9500~10499	很多
11	>=15000	土豪

3.1.6 收款订单类型维度

列名	数据类型	含义
order_id	int	维度 id
orderk_chinese	varchar(50)	订单类型中文翻译
orderk_value	varchar(50)	订单类型值

分类的详细信息为:

orderk_id	orderk_value	orderk_chinese
1	INSURANCE	保险费支付
2	MANAGEMENT	物业管理费
3	RENTAL	租金缴纳
4	LOAN	偿还贷款
5		未知

3.1.7 地区维度

列名	数据类型	含义
region_id	int	地区维度 id，对应 district id
district_name	varchar(50)	地区名
region_name	varchar(50)	所属大区名

3.1.8 交易类型维度

列名	数据类型	含义
trank_id	int	维度 id
trank _chinese	varchar(50)	交易类型中文翻译
trank _value	varchar(50)	交易类型值

分类的详细信息为:

trank_id	trank_value	trank_chinese
1	INSURANCE	保险费支付
2	BILL	账单支付
3	INTEREST_INCOME	储蓄利息收入
4	INTEREST_PENALTY	处罚利息
5	HOUSEHOLD	物业管理费
6	OLD_AGE_PENSION	养老金

7	LOAN	还贷
8		未知

3.1.9 交易额维度

列名	数据类型	含义
trans_amount_id	int	维度 id
amount_value	varchar(50)	交易额/1000 的数值，不超过 11
amount_group	varchar(50)	对交易额的描述，如很少、很多……

分类的详细信息为：

trans_amount_id	amount_value	amount_group
-1	<=-500	取款
0	-499~499	较少
1	500~1499	较少
2	1500~2499	中等
3	2500~3499	中等
4	3500~4499	中等
5	4500~5499	较多
6	5500~6499	较多
7	6500~7499	较多
8	7500~8499	很多
9	8500~9499	很多
10	9500~10499	很多
11	>=15000	土豪

3.1.10 余额维度

列名	数据类型	含义
trans_balance_id	int	维度 id
balance_value	varchar(50)	余额/10000 的数值，不超过 11
balance_group	varchar(50)	对余额的描述，如很少、很多……

分类的详细信息为：

trans_balance_id	balance_value	balance_group
-1	<=-5000	负债
0	-4990~4990	较少
1	5000~14990	中等
2	15000~24990	中等
3	25000~34990	中等
4	35000~44990	中等
5	45000~54990	较多

6	55000~64990	较多
7	65000~74990	很多
8	75000~84990	很多
9	85000~94990	很多
10	95000~104990	很多
11	>=150000	土豪

3.1.11 年份额度

列名	数据类型	含义
year_id	int	年份 id
year_value	varchar(50)	年份值

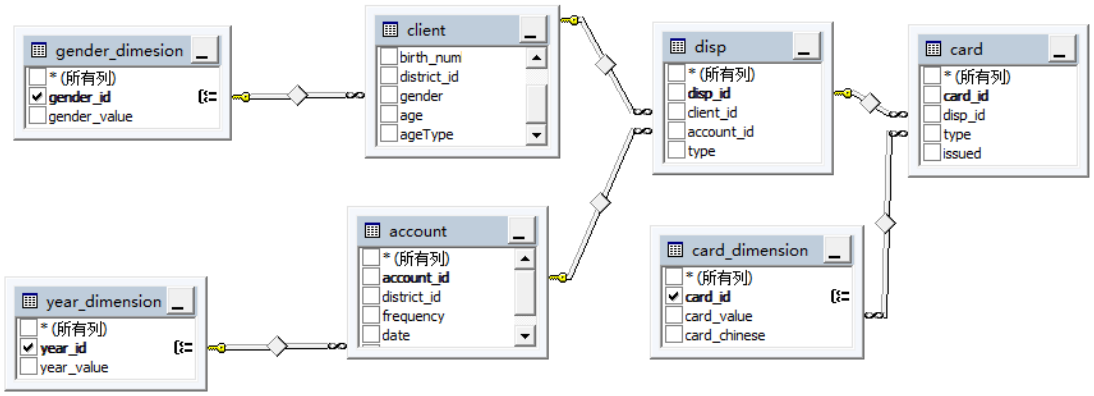
具体的分类为:

year_id	year_value
1990	1990
1991	1991
1992	1992
1993	1993
1994	1994
1995	1995
1996	1996
1997	1997
1998	1998
1999	1999
2000	2000

3.2 事实表

事实表是数据库内视图，视图与各个维度相关联，并添加一些统计量。

3.2.1 用户数事实表



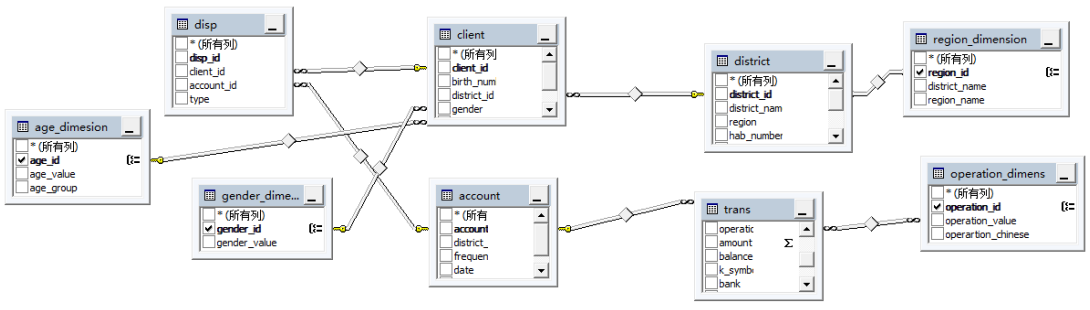
用户数事实表视图 SQL 语句为:

```
SELECT dbo.gender_dimesion.gender_id, dbo.card_dimension.card_id, dbo.year_dimension.year_id, COUNT(*) AS user_number
FROM dbo.disp INNER JOIN
    dbo.card ON dbo.disp.disp_id = dbo.card.disp_id INNER JOIN
    dbo.account ON dbo.disp.account_id = dbo.account.account_id INNER JOIN
    dbo.client ON dbo.disp.client_id = dbo.client.client_id INNER JOIN
    dbo.card_dimension ON dbo.card.type = dbo.card_dimension.card_value INNER JOIN
    dbo.gender_dimesion ON dbo.client.gender = dbo.gender_dimesion.gender_id INNER JOIN
    dbo.year_dimension ON dbo.account.dateYear = dbo.year_dimension.year_id
GROUP BY dbo.gender_dimesion.gender_id, dbo.card_dimension.card_id, dbo.year_dimension.year_id
```

各列的含义如下:

列	别名	表	分组依据
gender_id		gender_dimesion	分组依据
card_id		card_dimension	分组依据
year_id		year_dimension	分组依据
*	**user_number**		Count

3.2.2 操作金额事实表



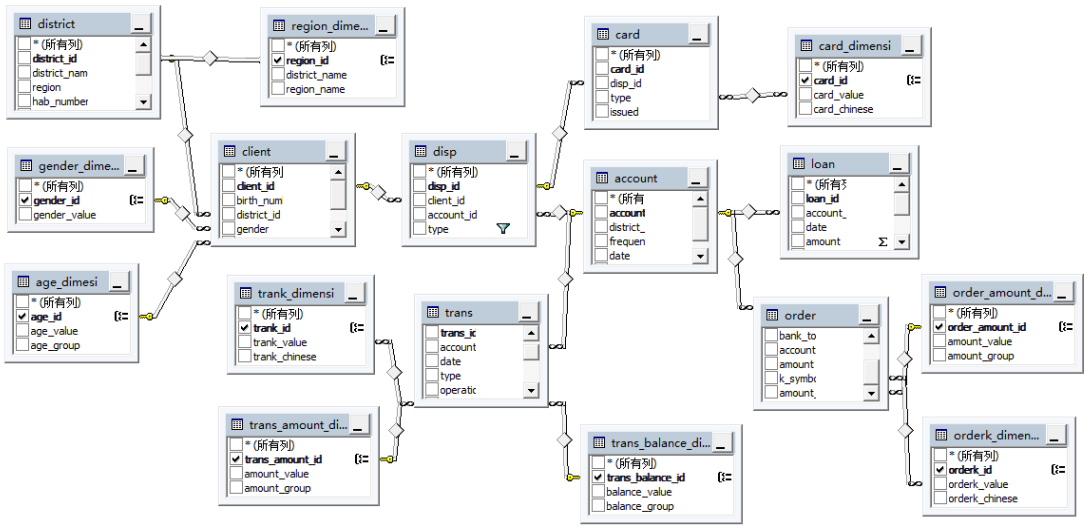
操作金额事实表 SQL 语句为：

```
SELECT dbo.age_dimesion.age_id, dbo.region_dimension.region_id, dbo.operation_dimension.operation_id,
dbo.gender_dimesion.gender_id, SUM(dbo.trans.amount) AS sum_value, SUM(dbo.trans.amount) / COUNT(*) AS avg_value
FROM dbo.district INNER JOIN
dbo.trans INNER JOIN
dbo.account ON dbo.trans.account_id = dbo.account.account_id INNER JOIN
dbo.disp ON dbo.account.account_id = dbo.disp.account_id INNER JOIN
dbo.client ON dbo.disp.client_id = dbo.client.client_id ON dbo.district.district_id = dbo.client.district_id INNER JOIN
dbo.age_dimesion ON dbo.client.ageType = dbo.age_dimesion.age_id INNER JOIN
dbo.gender_dimesion ON dbo.client.gender = dbo.gender_dimesion.gender_id INNER JOIN
dbo.operation_dimension ON dbo.trans.operation = dbo.operation_dimension.operation_value INNER JOIN
dbo.region_dimension ON dbo.district.district_id = dbo.region_dimension.region_id
GROUP BY dbo.age_dimesion.age_id, dbo.region_dimension.region_id, dbo.operation_dimension.operation_id, dbo.gender_dimesion.gender_id
```

各列的含义如下：

列	别名	表	分组依据
age_id		age_dimesion	分组依据
region_id		region_dimension	分组依据
operation_id		operation_dimension	分组依据
gender_id		gender_dimesion	分组依据
amount	**sum_value**	trans	Sum
SUM(dbo.trans.amount) / COUNT(*)	**avg_value**		表达式

3.2.3 贷款额事实表



贷款额事实表的 SQL 语句为：

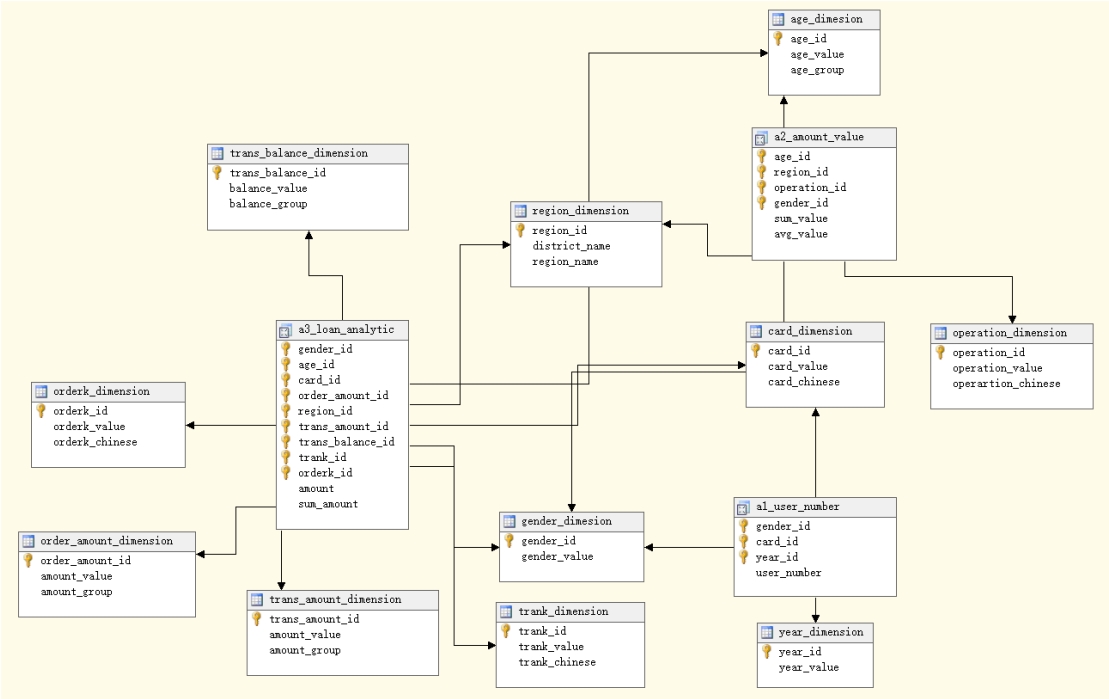
```
SELECT dbo.gender_dimesion.gender_id, dbo.age_dimesion.age_id, dbo.card_dimension.card_id, dbo.order_amount_dimension.order_amount_id,
dbo.region_dimension.region_id,dbo.trans_amount_dimension.trans_amount_id,dbo.trans_balance_dimension.trans_balance_id,
dbo.trank_dimension.trank_id, dbo.orderk_dimension.orderk_id, AVG(dbo.loan.amount)    AS amount,SUM(dbo.loan.amount) AS sum_amount
FROM            dbo.loan INNER JOIN
                         dbo.account ON dbo.loan.account_id = dbo.account.account_id INNER JOIN
                         dbo.disp ON dbo.account.account_id = dbo.disp.account_id INNER JOIN
                         dbo.client ON dbo.disp.client_id = dbo.client.client_id INNER JOIN
                         dbo.district ON dbo.client.district_id = dbo.district.district_id INNER JOIN
                         dbo.card ON dbo.disp.disp_id = dbo.card.disp_id INNER JOIN
                         dbo.trans ON dbo.account.account_id = dbo.trans.account_id INNER JOIN
                         dbo.[order] ON dbo.account.account_id = dbo.[order].account_id INNER JOIN
                         dbo.age_dimesion ON dbo.client.ageType = dbo.age_dimesion.age_id INNER JOIN
                         dbo.card_dimension ON dbo.card.type = dbo.card_dimension.card_value INNER JOIN
                         dbo.gender_dimesion ON dbo.client.gender = dbo.gender_dimesion.gender_id INNER JOIN
                         dbo.order_amount_dimension ON dbo.[order].amount_id= dbo.order_amount_dimension.order_amount_id INNER JOIN
                         dbo.region_dimension ON dbo.district.district_id = dbo.region_dimension.region_id INNER JOIN
                         dbo.trans_amount_dimension ON dbo.trans.amount_id = dbo.trans_amount_dimension.trans_amount_id INNER JOIN
                         dbo.trans_balance_dimension ON dbo.trans.balance_id = dbo.trans_balance_dimension.trans_balance_id INNER JOIN
                         dbo.orderk_dimension ON dbo.[order].k_symbol = dbo.orderk_dimension.orderk_value INNER JOIN
                         dbo.trank_dimension ON dbo.trans.k_symbol = dbo.trank_dimension.trank_value
WHERE       (dbo.disp.type <> 'DISPONENT')
GROUP       BY      dbo.gender_dimesion.gender_id,      dbo.age_dimesion.age_id,      dbo.card_dimension.card_id,
dbo.order_amount_dimension.order_amount_id, dbo.region_dimension.region_id,
```

各列的含义如下：

列	别名	表	分组依据
age_id		age_dimesion	分组依据
card_id		card_dimension	分组依据
order_amount_id		order_amount_dimension	分组依据
region_id		region_dimension	分组依据
trans_amount_id		trans_amount_dimension	分组依据
trans_balance_id		trans_balance_dimension	分组依据
trank_id		trank_dimension	分组依据
orderk_id		orderk_dimension	分组依据
amount	**amount**	loan	Avg
amount	**sum_amount**	loan	Sum

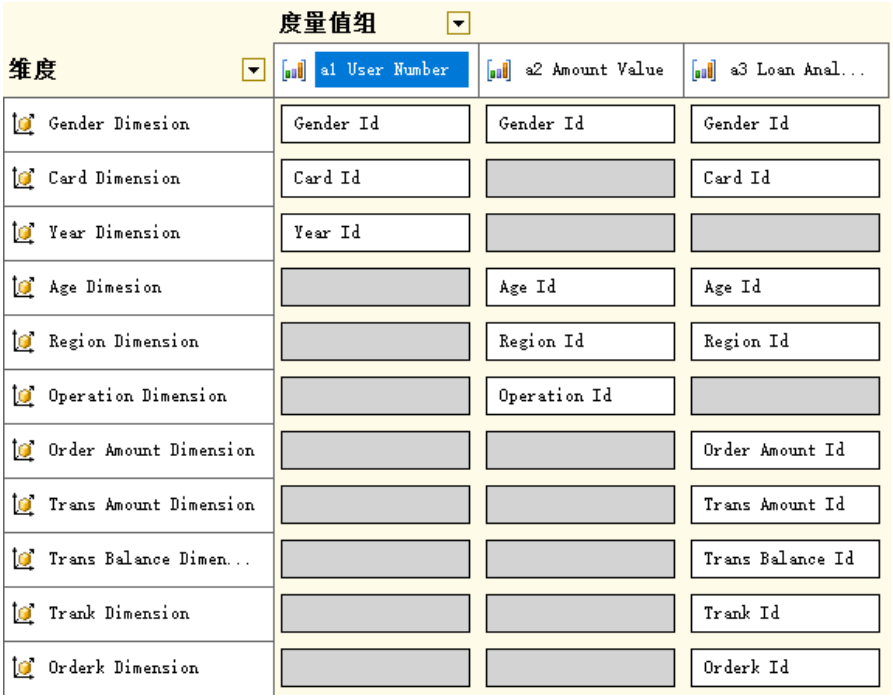
3.3 维度表与事实表的关系

维度表与事实表的关系如下图所示：



四、数据分析

对于不同的度量值，使用不同的维度来度量。不同度量值组与维度的关系如下图所示。



4.1 用户数量

分析不同性别，不同开通账户的时间段，不同信用卡类型的用户数量情况。

		Year Value					
		1993	1994	1995	1996	1997	总计
Gender Value	Card Chinese	User Number	User Number	User Number	User Number	User Number	User Number
⊟男	初级卡	16	6	22	19	12	75
	金卡	20	4	8	12	8	52
	普通卡	77	38	59	94	74	342
	汇总	113	48	89	125	94	469
⊟女	初级卡	23	10	7	19	12	71
	金卡	15	4	6	6	4	35
	普通卡	65	32	42	114	64	317
	汇总	103	46	55	139	80	423
总计		216	94	144	264	174	892

由以上统计可知，在 1993 年开通账户的女性中，持有金卡的用户数为 15 人。

		Card Chinese			
		初级卡	金卡	普通卡	总计
Gender Value	Year Value	User Number	User Number	User Number	User Number
⊟男	1993	16	20	77	113
	1994	6	4	38	48
	1995	22	8	59	89
	1996	19	12	94	125
	1997	12	8	74	94
	汇总	75	52	342	469
⊟女	1993	23	15	65	103
	1994	10	4	32	46
	1995	7	6	42	55
	1996	19	6	114	139
	1997	12	4	64	80
	汇总	71	35	317	423
总计		146	87	659	892

在 1996 年开通普通卡（CLASSIC）的女性人数最多，为 114 人，总体来说，1996 年开通账户的客户持有信用卡的人数最多，男性用户略多于女性用户，开通账户的客户中持有普通信用卡的人数最多。

4.2 交易金额

	Operartion Chinese							
	汇款到其他银行		其他银行汇款		提取现金		总计	
Gender Value	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value
男	136914023	1450239	34919000	439770	4268300	251899	176101323	2141908
女	130054636	1325862	36491492	424267	4541300	326318	171087428	2076447
总计	266968659	2776101	71410492	864037	8809600	578217	347188751	4218355

从统计结果可以看出，男性账户汇款到其他银行的操作数额更多，均值也更大，而女性收到其他银行的汇款更多，但收到汇款的均值不如男性客户多。女性在提取现金这个操作种类上总额和平均值都要超过男性客户。

		Operartion Chinese							
		汇款到其他银行		其他银行汇款		提取现金		总计	
Gender Value	Age Group	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value
男	老年	30445161	228078	34277034	410285	343500	22292	65065695	660655
	青年	24302150	285900			338900	34028	24641050	319928
	少年	582655	27521					582655	27521
	壮年	81584057	908740	641966	29485	3585900	195579	85811923	1133804
	汇总	136914023	1450239	34919000	439770	4268300	251899	176101323	2141908
女	老年	23686127	196018	36491492	424267	549000	24242	60726619	644527
	青年	24953185	256966			992200	81313	25945385	338279
	少年	583144	32962					583144	32962
	壮年	80832180	839916			3000100	220763	83832280	1060679
	汇总	130054636	1325862	36491492	424267	4541300	326318	171087428	2076447
总计		266968659	2776101	71410492	864037	8809600	578217	347188751	4218355

		Operartion Chinese							
		汇款到其他银行		其他银行汇款		提取现金		总计	
Age Group	Gender Value	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value
老年		54131288	424096	70768526	834552	892500	46534	125792314	1305182
青年		49255335	542866			1331100	115341	50586435	658207
少年		1165799	60483					1165799	60483
壮年		162416237	1748656	641966	29485	6586000	416342	169644203	2194483
总计		266968659	2776101	71410492	864037	8809600	578217	347188751	4218355

从年龄上开看，壮年和老年人交易总金额和平均值更多，壮年客户汇款到其他银行的数额和均值都远远超过老年人，而老年人收到其他银行汇款总额和均值都远远超过壮年客户，男性壮年客户汇出了更多的金额，女性老年客户收到了最多的其他银行汇款，男性壮年提取了更多的现金，但女性壮年客户提取现金的平均值更多。

			Operartion Chinese							
			汇款到其他银行		其他银行汇款		提取现金		总计	
Gender Value	Age Group	Region Name	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value	Sum Value	Avg Value
男	老年	central Bohemia	4254719	35246	4602705	65301			8857424	100547
		east Bohemia	3551834	31282	3841405	54872			7393239	86154
		north Bohemia	2602768	29250	3955617	53763			6558385	83013
		north Moravia	5844276	34438	6612812	59573			12457088	94011
		Prague	4028952	2839	4378762	5679	112600	5926	8520314	14444
		south Bohemia	2405003	24244	2283999	43694			4689002	67938
		south Moravia	4931097	42928	4968378	69721	174500	8309	10073975	120958
		west Bohemia	2826512	27851	3633356	57682	56400	8057	6516268	93590
		汇总	30445161	228078	34277034	410285	343500	22292	65065695	660655
	青年		24302150	285900			338900	34028	24641050	319928
	少年		582655	27521					582655	27521
	壮年	central Bohemia	9394829	149458			208800	15064	9603629	164522
		east Bohemia	10655028	124051			546800	50251	11201828	174302
		north Bohemia	9549053	122038	115587	4281	353800	17901	10018440	144220
		north Moravia	12703708	134209			666900	38817	13370608	173026
		Prague	10451899	11775	142128	5922	645100	24618	11239127	42315
		south Bohemia	5650080	85444	116232	4843	533700	17421	6300012	107708
		south Moravia	14754663	163288	166335	9817	630800	31507	15551798	204612
		west Bohemia	8424797	118477	101684	4622			8526481	123099
		汇总	81584057	908740	641966	29485	3585900	195579	85811923	1133804
	汇总		136914023	1450239	34919000	439770	4268300	251899	176101323	2141908
女	老年	central Bohemia	2738824	28065	4707048	64552			7445872	92617
		east Bohemia	2589302	29578	4761119	60533	41100	2283	7391521	92394
		north Bohemia	2246087	24924	3614518	52756			5860605	77680
		north Moravia	4199109	27809	6134554	61125	65800	2990	10399463	91924
		Prague	3957632	2765	5511478	5658			9469110	8423
		south Bohemia	1647537	21824	1897647	44035	417400	15459	3962584	81318
		south Moravia	3898608	35772	5975856	78082			9874464	113854
		west Bohemia	2409028	25281	3889272	57526	24700	3510	6323000	86317
		汇总	23686127	196018	36491492	424267	549000	24242	60726619	644527
	青年		24953185	256966			992200	81313	25945385	338279
	少年		583144	32962					583144	32962
	壮年	central Bohemia	8772740	122595			95200	9520	8867940	132115
		east Bohemia	8739637	124005			491400	45663	9231037	169668
		north Bohemia	9397642	112454					9397642	112454
		north Moravia	14269107	120977			477900	34059	14747007	155036
		Prague	10080746	11091			99500	3980	10180246	15071
		south Bohemia	6762819	80993			731100	45407	7493919	126400
		south Moravia	15109431	145761			871300	58807	15980731	204568
		west Bohemia	7700058	122040			233700	23327	7933758	145367
		汇总	80832180	839916			3000100	220763	83832280	1060679
	汇总		130054636	1325862	36491492	424267	4541300	326318	171087428	2076447
总计			266968659	2776101	71410492	864037	8809600	578217	347188751	4218355

加入地区的统计之后可以看到，South Moravia 地区在汇款和收款等各项指标上都高于其他地区，South Moravia 和 North Moravia 两个地区的交易额都很高，South Bohemia 则比

较少，来自 South Moravia 的女性壮年客户汇款到其他银行的总额更多，但来自 South Moravia 的男性壮年客户汇款到其他银行的平均值更大。汇款到其他银行和提取现金占交易额的大多数，而收到其他银行汇款的交易额比较少。

4.3 贷款总额

	Age Value					
	20~29岁	30~39岁	40~49岁	50~59岁	60~岁	总计
Gender Value	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount
男	297145752	148002984	129006108	96526428	56568600	727249872
女	140592936	135073404	64634988	103120080	27524472	470945880
总计	437738688	283076388	193641096	199646508	84093072	1198195752

从年龄和性别上看，20~29 岁男性贷款数额更多，男性和女性的贷款数额随着年龄增长都呈现下降趋势，50~59 岁的女性例外。

		Card Chinese			
		初级卡	金卡	普通卡	总计
Gender Value	Age Value	Sum Amount	Sum Amount	Sum Amount	Sum Amount
⊟男	20~29岁	194549844		102595908	297145752
	30~39岁		35445600	112557384	148002984
	40~49岁		8682912	120323196	129006108
	50~59岁		6239100	90287328	96526428
	60~岁		7013520	49555080	56568600
	汇总	194549844	57381132	475318896	727249872
⊟女	20~29岁	118743396	1362480	20487060	140592936
	30~39岁		282480	134790924	135073404
	40~49岁		3627720	61007268	64634988
	50~59岁		4934592	98185488	103120080
	60~岁			27524472	27524472
	汇总	118743396	10207272	341995212	470945880
总计		313293240	67588404	817314108	1198195752

从持卡人信用卡情况可以看出，持有初级卡（JUNIOR）的 20~29 岁男性客户贷款额非常高，远远超出其他群体，持有普通卡的 20~49 岁男性，持有普通卡的 30~39 岁女性和持有初级卡的 20~29 岁女性也有较高的贷款额度。

		Region Name								
		⊞ central Bohemia	⊞ east Bohemia	⊞ north Bohemia	⊞ north Moravia	⊞ Prague	⊞ south Bohemia	⊞ south Moravia	⊞ west Bohemia	总计
Gender Value	Age Value	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount
⊟男	20~29岁	128632476	33105216	965520	22594944	36887952		65868288	9091356	297145752
	30~39岁		58526904	31812960	1943520	30313920	2748312	9793272	12864096	148002984
	40~49岁	1792032	113436	20780784	4535712		15156288	62719896	23907960	129006108
	50~59岁	2170236	23562192	1316520	33508224		2373648	21979560	11616048	96526428
	60~岁	17104320			6008688	4493664	7013520	2809728	19138680	56568600
	汇总	149699064	115307748	54875784	68591088	71695536	27291768	163170744	76618140	727249872
⊟女	20~29岁	72746952	1056384	20946312	19776648	12128280	10070208	3868152		140592936
	30~39岁	294432	1501584	25937568	3554928	36237204	8512800	9153480	49881408	135073404
	40~49岁	16803216	1434696	3567024	13389948	20682240	2730600	6027264		64634988
	50~59岁	4631352	32296140	16577340	24612072	8504496	5765520	9473640	1259520	103120080
	60~岁					19408320	6310392		1805760	27524472
	汇总	94475952	36288804	67028244	61333596	96960540	33389520	28522536	52946688	470945880
总计		244175016	151596552	121904028	129924684	168656076	60681288	191693280	129564828	1198195752

从地区分布可以看出，来自 central Bohemia 和 South Moravia 地区的客户贷款数额更多，来自 central Bohemia 的 20~29 岁男性贷款数额巨大。

		Trank Chinese	Amount Group															
		⊟保险费支付				⊟还贷					⊟物业管理费						⊞未知	总计
		⊞较多	⊞较少	⊞中等	汇总	⊞很多	⊞较多	⊞较少	⊞中等	汇总	⊞很多	⊞较多	⊞较少	⊞土豪	⊞中等	汇总		
Gender Value	Age Value	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount
⊟男	20~29岁		7202112	3744000	10946112	5308272	44638596	567648	29399592	79914108	26585328	30318408	19778580	533568	46738428	123954312	82331220	297145752
	30~39岁		7771200	16235520	24006720	681072	16464504	200520	11199156	28545252		6442236			29846688	36288924	59162088	148002984
	40~49岁		4603860		4603860	5894112	5339280	6384420	17896320	35514132	16441836	12411552	16481340		9679440	55014168	33873948	129006108
	50~59岁		4157652	9498192	13655844	2057520	12507600	1564860	15733500	31863480	8955624		13906440		1332864	24194928	26812176	96526428
	60~岁		6509640		6509640		7498008		6959832	14457840	1443960	2685552	8351424	3004344	3489840	18975120	16626000	56568600
	汇总		30244464	29477712	59722176	13940976	86447988	8717448	81188400	190294812	53426748	51857748	58517784	3537912	91087260	258427452	218805432	727249872
⊟女	20~29岁		15377472		15377472	1471200	26339988	1081320	6256572	35149080	30820464	1286280	13762224	2699640	5741136	54309744	35756640	140592936
	30~39岁		14962860		14962860	8477472	20624832	1886208	10312596	41301108	10696320	9093216	13606344	3161184	9306108	45863172	32946264	135073404
	40~49岁	3077760	1841664		4919424		3456672	3040536	18673848	25171056		12060492	5059776		4336416	21456684	13087824	64634988
	50~59岁		1316520		1316520	218688	15908052	3461760	9417900	29006400	8553936	7781640	5380212	8519160	14986440	45221388	27575772	103120080
	60~岁		451440		451440		7290144	200640	1904472	9395256		4232160			12843696	17075856	601920	27524472
	汇总	3077760	33949956		37027716	10167360	73619688	9670464	46565388	140022900	50070720	34453788	37808556	14379984	47213796	183926844	109968420	470945880
总计		3077760	64194420	29477712	96749892	24108336	160067676	18387912	127753788	330317712	103497468	86311536	96326340	17917896	138301056	442354296	328773852	1198195752

		Orderk Chinese				Amount Group													
		保险费支付				偿还贷款					物业管理费						未知	总计	
		较多	较少	中等	汇总	很多	较多	较少	中等	汇总	很多	较多	较少	土豪	中等	汇总			
Gender Value	Age Value	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	
男	20~29岁		9198624	3556800	12755424	7760784	65372256	1371816	34448520	108953376	21153624	24693696	18942240	666960	33669612	99126132	76310820	297145752	
	30~39岁		6022680	16198440	22221120	681072	22255344	200520	18635016	41771952		4435896			28078848	32511744	51498168	148002984	
	40~49岁		3863448		3863448	7578144	10478352	9529272	22932828	50518596	13220352	7891680	16329960		7958952	45402944	29223120	129006108	
	50~59岁		5069776	7859712	12929488	4115040	13307880	2067132	18121776	37611828	6867768		13754040		1944576	22566384	23398728	96526428	
	60~岁		4892544		4892544		7498008		13781160	21279168	1402704	2509056	6379560	3004344	3489840	16785504	13611384	56568600	
	汇总		29067072	27614952	56682024	20135040	118911840	13168740	107919300	260134920	42644448	39530328	55405800	3671304	75138828	216390708	194042220	727249872	
女	20~29岁		10834128		10834128	1471200	38599392	1386000	13835772	55292364	29322708	1143360	9586752	2159712	4828176	47040708	27425736	140592936	
	30~39岁		15815628		15815628	8081856	29546868	2263272	12684504	52576500	8981640	7239648	13780296	2696304	8217900	40915788	25765488	135073404	
	40~49岁	2693040	1506816		4199856		6846540	4152420	17793516	28792476		12723240	4199856		3279204	20202300	11440356	64634988	
	50~59岁		1199496		1199496	328032	30060384	5346156	10684824	46419396	7854468	7036740	5077776	4259680	11395140	35623704	19877484	103120080	
	60~岁		451440		451440		9877920	451440	3068316	13397676		3068316			10155600	13223916	451440	27524472	
	汇总	2693040	29807508		32500548	9881088	114931104	13599288	58066932	196478412	46158816	31211304	32644680	9115596	37876020	157006416	84960504	470945880	
总计		2693040	58874580	27614952	89182572	30016128	233842944	26768028	165986232	456613332	88803264	70741632	88050480	12786900	113014848	373397124	279002724	1198195752	

结合交易数据表和订单数据表中的种类和数据，较多的贷款发生在支付较少或中等费用的保险金的 30~39 岁客户、支付中等费用物业管理费的 20~29 岁男性，支付较少费用物业管理费的 30~39 岁女性客户身上。

		Balance Group	Balance Value										
		较少	中等					较多			很多	土豪	总计
			5000~14990	15000~24990	25000~34990	35000~44990	汇总	45000~54990	55000~64990	汇总			
Gender Value	Age Value	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount	Sum Amount
男	20~29岁	743184	11865000	33436896	54061800	48084480	147448176	51319404	37209948	88529352	58806840	1618200	297145752
	30~39岁	3213900	6575160	15554400	20213244	19985856	62328660	34504908	14216604	48721512	33738912		148002984
	40~49岁	508728	7405644	15137400	25343112	16730268	64616424	24851928	16255596	41107524	22773432		129006108
	50~59岁	492480	8419404	12296760	18295116	15867252	54878532	12279936	12195060	24474996	15050100	1630320	96526428
	60~岁	1043928	4470432	3431928	4311048	13147656	25361064	4319952	8504040	12823992	16874568	465048	56568600
	汇总	6002220	38735640	79857384	122224320	113815512	354632856	127276128	88381248	215657376	147243852	3713568	727249872
女	20~29岁	20169288	8781408	9843384	13685232	18432876	50742900	17161116	19309320	36470436	33017568	192744	140592936
	30~39岁	371904	6843624	14250204	16968084	18993060	57054972	25421688	18812784	44234472	33412056		135073404
	40~49岁	307776	4079712	12129264	10972428	9371064	36552468	11784816	6705456	18490272	8920392	364080	64634988
	50~59岁	2283336	3673596	12411132	14119692	21536076	51740496	19208976	10721064	29930040	19166208		103120080
	60~岁		3496488	4256232	6960600	6100368	20813688	1881216	2788416	4669632	2041152		27524472
	汇总	23132304	26874828	52890216	62706036	74433444	216904524	75457812	58337040	133794852	96557376	556824	470945880
总计		29134524	65610468	132747600	184930356	188248956	571537380	202733940	146718288	349452228	243801228	4270392	1198195752

从账户余额上来看，账户余额在 25000~55000 的 20~29 岁男性客户贷款额更多，而余额低于 5000 的 20~29 岁女性相比来讲有更多的贷款数额。

总的来说，管理人员应该把宣传力度放到满足以下部分特征的人群中：

1、 男性，20~29 岁，持有初级或普通信用卡，需要支付中等的物业费用，账户余额在 25000~55000 左右。

2、 女性，20~39 岁，持有初级或普通信用卡，余额低于 5000，支付较少的物业费用。

3、 来自 Central Bohemia 和 South Moravia 地区的客户。