



UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING, FACULTY ENGINEERING

SESSION 2021/2022 SEMESTER 2

GROUP PROJECT: PART 2 (MONGO DB)

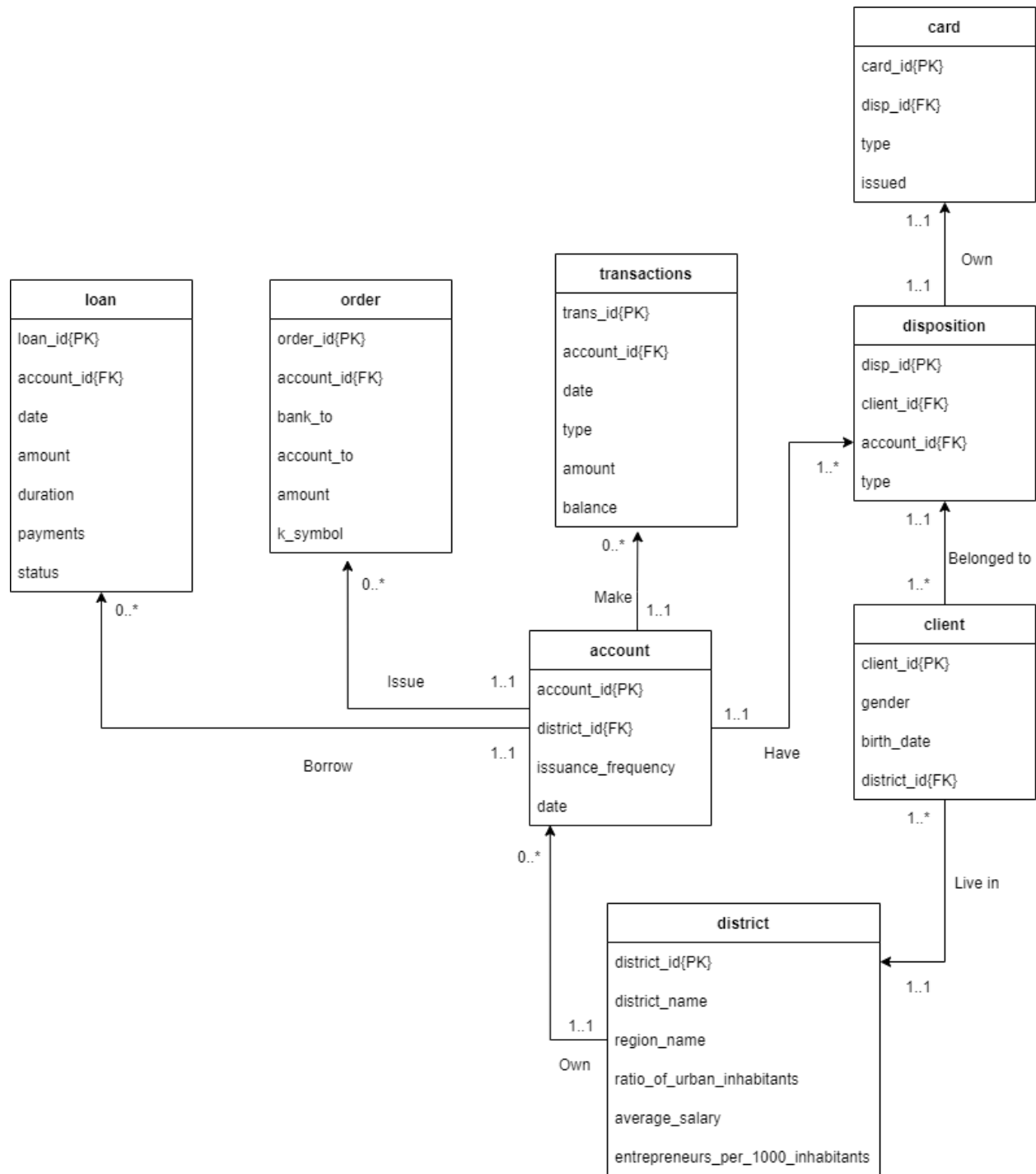
SCSP5023/MCSD1123

BIG DATA MANAGEMENT

NAME	MATRIC NO
ALYA NASUHA BINTI MOHAMMAD NASIRUDDIN	B19EC3001
CHONG XIAN JUN	MCS211047
TAN FEI ZHI	B19EC0041

QUESTION 1

In Project 1, a RDBMS with the following schema as shown in the ER diagram below is used.



As we can see from the ER diagram above, there is a total of 8 tables. For Project 2, we are converting the tables into documents in the embedded form to fit into document type NoSQL database structure.

The foundational logic of NoSQL database building is inherently distinct from that of a SQL database. For relational databases built by SQL, one of the signature functions that it uses is the `JOIN` function, which allows atomic tables to be threaded together and form complex databases. While it is incredibly powerful, it does not facilitate horizontal scaling as the complex relations impede data sharding and replication required. With its flexible schema, NoSQL database structure tackles this issue with the use of embedded documents to minimize the amount of relations between collections. Thus, in this project, we try to integrate and embed as much information in a single document as possible.

However, embedding all tables within a single document is definitely impractical. As there are an enormous amounts of combinations that we can adopt to build our NoSQL database, the rule of thumb is to design the database based on our goal. Therefore, the database is reverse engineered by referring to the desired output of our queries. As a result, 3 collections are derived from the 8 tables, i.e. **accountInfo.json**, **accountStatement.json** and **clientInfo.json**. They are derived from the tables from the original tables with aggregate queries, e.g. \$lookup and \$unwind to join tables in a single document, \$project to reshape the document schema and \$out to export aggregate results as new collections. Below are the examples for the three collections derived:

accountInfo.json (account + district + loan + disposition)

```
_id: ObjectId('62cd5c2b57a549773de37427')
account_id: 1
issuance_frequency: "monthly issuance"
date: "1995-03-24"
district: Object
  district_id: 18
  district_name: "Pisek"
  region_name: "south Bohemia"
  average_salary: "8968"
client_id: Array
  0: 1
```

The embedded document is derived with aggregation pipeline as shown below:

Pipeline  \$lookup > \$unwind > \$lookup > \$unwind > \$lookup > \$project > \$out

```
db.account.aggregate([
  $lookup: {
    from: 'district',
    localField: 'district_id',
    foreignField: 'district_id',
    as: 'district'
  }, {
    $unwind: {
      path: '$district',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $lookup: {
      from: 'loan',
      localField: 'account_id',
      foreignField: 'account_id',
      as: 'loan'
    }
  }, {
    $unwind: {
      path: '$loan',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $lookup: {
      from: 'disposition',
      localField: 'account_id',
      foreignField: 'account_id',
      as: 'disposition'
    }
  }, {
    $project: {
      _id: 1,
      account_id: 1,
      client_id: '$disposition.client_id',
      issuance_frequency: 1,
      date: 1,
    }
  }
])
```

```

loan: 1,
district: {
  district_id: 1,
  district_name: '$district.district_name',
  region_name: '$district.region_name',
  average_salary: '$district.average_salary'
}
}, {
  $out: 'accountInfo'
}]])

```


accountStatement.json (account + transactions + order)

```

{
  "_id": ObjectId("62cdbc2b57a549773de37427"),
  "account_id": 1,
  "issuance_frequency": "monthly issuance",
  "bank_order_to": Array(
    0: {
      "_id": ObjectId("62cddb8c57a549773de3b8d8"),
      "order_id": 29401,
      "account_id": 1,
      "bank_to": "YZ",
      "account_to": 87144583,
      "amount": 2452.0,
      "k_symbol": "SIPO"
    }
  ),
  "transactions": Array(
    0: {
      "_id": ObjectId("62cddbba57a549773de3d221"),
      "trans_id": 1,
      "account_id": 1,
      "date": "1995-03-24",
      "type": "credit"
    }
  )
}

```

The embedded document is derived with aggregation pipeline as shown below:

Pipeline  \$project \$lookup \$lookup \$out

```

db.account.aggregate([
  $project: {
    account_id: 1,
    issuance_frequency: 1
  }, {

```

```
$lookup: {
  from: 'order',
  localField: 'account_id',
  foreignField: 'account_id',
  as: 'bank_order_to'
}, {
  $lookup: {
    from: 'transactions',
    localField: 'account_id',
    foreignField: 'account_id',
    as: 'transactions'
  }
}, {
  $out: 'accountStatements'
}]])
```

clientInfo.json (client + district + account + disposition + card + loan)

```
_id: ObjectId('62cdbc7d57a549773de3893b')
client_id: 1
gender: "F"
birth_date: "1970-12-13"
district: Object
  district_id: 18
  district_name: "Pisek"
  region_name: "south Bohemia"
  district_avg_salary: "8968"
account: Object
  account_id: 1
  issuance_freq: "monthly issuance"
  open_date: "1995-03-24"
  account_type: "owner"
```

The embedded document is derived with aggregation pipeline as shown below:

```
$lookup > $unwind > $lookup > $unwind > $lookup > $unwind > $lookup > $unwind > $lookup > $unwind >
$project > $out
```

```
db.client.aggregate([
  $lookup: {
    from: 'district',
```

```

    localField: 'district_id',
    foreignField: 'district_id',
    as: 'district'
  }, {
    $unwind: {
      path: '$district',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $lookup: {
      from: 'disposition',
      localField: 'client_id',
      foreignField: 'client_id',
      as: 'disposition'
    }
  }, {
    $unwind: {
      path: '$disposition',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $lookup: {
      from: 'account',
      localField: 'disposition.account_id',
      foreignField: 'account_id',
      as: 'accounts'
    }
  }, {
    $unwind: {
      path: '$accounts',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $lookup: {
      from: 'card',
      localField: 'disposition.disp_id',
      foreignField: 'disp_id',
      as: 'cards'
    }
  }, {
    $unwind: {
      path: '$cards',

```

```

    preserveNullAndEmptyArrays: true
  }
}, {
  $lookup: {
    from: 'loan',
    localField: 'accounts.account_id',
    foreignField: 'account_id',
    as: 'loan'
  }
}, {
  $unwind: {
    path: '$loan',
    preserveNullAndEmptyArrays: true
  }
}, {
  $project: {
    client_id: 1,
    gender: 1,
    birth_date: 1,
    district: {
      district_id: '$district_id',
      district_name: '$district.district_name',
      region_name: '$district.region_name',
      district_avg_salary: '$district.average_salary'
    },
    account: {
      account_id: '$accounts.account_id',
      issuance_freq: '$accounts.issuance_frequency',
      open_date: '$accounts.date',
      account_type: '$disposition.type'
    },
    loan: {
      loan_account: '$loan.account_id',
      loan_date: '$loan.date',
      amount: '$loan.amount',
      duration: '$loan.duration',
      payments: '$loan.payments',
      status: '$loan.status'
    },
    cards: {
      card_id: '$cards.card_id',
      card_type: '$cards.type',
      issued_date: '$cards.issued'
    }
  }
}

```



```
    }  
  }  
}, {  
  $out: 'clientInfo'  
}]
```

```
db.clientInfo_01.updateMany({loan: {}}, {$unset: {loan:1}})  
db.clientInfo_01.updateMany({cards: {}}, {$unset: {cards:1}})
```

QUESTION 2

Knowing the Clients of Bank A

1) The distribution of male and female clients

```
db.clientInfo.aggregate([{\n  $group: {\n    _id: '$gender',\n    count: {\n      $sum: 1\n    }\n  }\n}])
```

Output:

_id	count
M	2724
F	2645

2) The distribution of clients by age

Pipeline  \$addFields > \$addFields > \$group > \$project > \$group > +1

```
db.clientInfo.aggregate([{\n  $addFields: {\n    birth_date: {\n      $dateFromString: {\n        dateString: '$birth_date',\n        format: '%Y-%m-%d'\n      }\n    }\n  }\n}])
```

```

}, {
  $addFields: {
    age: {
      $dateDiff: {
        startDate: '$birth_date',
        endDate: ISODate('1999-01-01T00:00:00.000Z'),
        unit: 'year'
      }
    }
  }
}, {
  $group: {
    _id: '$age',
    count: {
      $sum: 1
    }
  }
}, {
  $project: {
    name: 1,
    client_id: 1,
    count: 1,
    age_range: {
      $switch: {
        branches: [
          {
            'case': {
              $lte: [
                '$_id',
                20
              ]
            },
            then: 'age <= 20'
          },
          {
            'case': {
              $and: [
                {
                  $gt: [
                    '$_id',
                    20
                  ]
                }
              ]
            },
            then: 'age > 20'
          }
        ]
      }
    }
  }
}

```

```
{
  $lte: [
    '$_id',
    25
  ]
}
],
},
then: '20 < age <= 25'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          25
        ]
      },
      {
        $lte: [
          '$_id',
          30
        ]
      }
    ]
  },
  then: '25 < age <= 30'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          30
        ]
      },
      {
        $lte: [
          '$_id',
          35
        ]
      }
    ]
  },
  then: '30 < age <= 35'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          35
        ]
      },
      {
        $lte: [
          '$_id',
          40
        ]
      }
    ]
  },
  then: '35 < age <= 40'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          40
        ]
      },
      {
        $lte: [
          '$_id',
          45
        ]
      }
    ]
  },
  then: '40 < age <= 45'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          45
        ]
      },
      {
        $lte: [
          '$_id',
          50
        ]
      }
    ]
  },
  then: '45 < age <= 50'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          50
        ]
      },
      {
        $lte: [
          '$_id',
          55
        ]
      }
    ]
  },
  then: '50 < age <= 55'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          55
        ]
      },
      {
        $lte: [
          '$_id',
          60
        ]
      }
    ]
  },
  then: '55 < age <= 60'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          60
        ]
      },
      {
        $lte: [
          '$_id',
          65
        ]
      }
    ]
  },
  then: '60 < age <= 65'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          65
        ]
      },
      {
        $lte: [
          '$_id',
          70
        ]
      }
    ]
  },
  then: '65 < age <= 70'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          70
        ]
      },
      {
        $lte: [
          '$_id',
          75
        ]
      }
    ]
  },
  then: '70 < age <= 75'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          75
        ]
      },
      {
        $lte: [
          '$_id',
          80
        ]
      }
    ]
  },
  then: '75 < age <= 80'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          80
        ]
      },
      {
        $lte: [
          '$_id',
          85
        ]
      }
    ]
  },
  then: '80 < age <= 85'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          85
        ]
      },
      {
        $lte: [
          '$_id',
          90
        ]
      }
    ]
  },
  then: '85 < age <= 90'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          90
        ]
      },
      {
        $lte: [
          '$_id',
          95
        ]
      }
    ]
  },
  then: '90 < age <= 95'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          95
        ]
      },
      {
        $lte: [
          '$_id',
          100
        ]
      }
    ]
  },
  then: '95 < age <= 100'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          100
        ]
      },
      {
        $lte: [
          '$_id',
          105
        ]
      }
    ]
  },
  then: '100 < age <= 105'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          105
        ]
      },
      {
        $lte: [
          '$_id',
          110
        ]
      }
    ]
  },
  then: '105 < age <= 110'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          110
        ]
      },
      {
        $lte: [
          '$_id',
          115
        ]
      }
    ]
  },
  then: '110 < age <= 115'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          115
        ]
      },
      {
        $lte: [
          '$_id',
          120
        ]
      }
    ]
  },
  then: '115 < age <= 120'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          120
        ]
      },
      {
        $lte: [
          '$_id',
          125
        ]
      }
    ]
  },
  then: '120 < age <= 125'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          125
        ]
      },
      {
        $lte: [
          '$_id',
          130
        ]
      }
    ]
  },
  then: '125 < age <= 130'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          130
        ]
      },
      {
        $lte: [
          '$_id',
          135
        ]
      }
    ]
  },
  then: '130 < age <= 135'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          135
        ]
      },
      {
        $lte: [
          '$_id',
          140
        ]
      }
    ]
  },
  then: '135 < age <= 140'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          140
        ]
      },
      {
        $lte: [
          '$_id',
          145
        ]
      }
    ]
  },
  then: '140 < age <= 145'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          145
        ]
      },
      {
        $lte: [
          '$_id',
          150
        ]
      }
    ]
  },
  then: '145 < age <= 150'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          150
        ]
      },
      {
        $lte: [
          '$_id',
          155
        ]
      }
    ]
  },
  then: '150 < age <= 155'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          155
        ]
      },
      {
        $lte: [
          '$_id',
          160
        ]
      }
    ]
  },
  then: '155 < age <= 160'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          160
        ]
      },
      {
        $lte: [
          '$_id',
          165
        ]
      }
    ]
  },
  then: '160 < age <= 165'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          165
        ]
      },
      {
        $lte: [
          '$_id',
          170
        ]
      }
    ]
  },
  then: '165 < age <= 170'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          170
        ]
      },
      {
        $lte: [
          '$_id',
          175
        ]
      }
    ]
  },
  then: '170 < age <= 175'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          175
        ]
      },
      {
        $lte: [
          '$_id',
          180
        ]
      }
    ]
  },
  then: '175 < age <= 180'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          180
        ]
      },
      {
        $lte: [
          '$_id',
          185
        ]
      }
    ]
  },
  then: '180 < age <= 185'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          185
        ]
      },
      {
        $lte: [
          '$_id',
          190
        ]
      }
    ]
  },
  then: '185 < age <= 190'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          190
        ]
      },
      {
        $lte: [
          '$_id',
          195
        ]
      }
    ]
  },
  then: '190 < age <= 195'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          195
        ]
      },
      {
        $lte: [
          '$_id',
          200
        ]
      }
    ]
  },
  then: '195 < age <= 200'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          200
        ]
      },
      {
        $lte: [
          '$_id',
          205
        ]
      }
    ]
  },
  then: '200 < age <= 205'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          205
        ]
      },
      {
        $lte: [
          '$_id',
          210
        ]
      }
    ]
  },
  then: '205 < age <= 210'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          210
        ]
      },
      {
        $lte: [
          '$_id',
          215
        ]
      }
    ]
  },
  then: '210 < age <= 215'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          215
        ]
      },
      {
        $lte: [
          '$_id',
          220
        ]
      }
    ]
  },
  then: '215 < age <= 220'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          220
        ]
      },
      {
        $lte: [
          '$_id',
          225
        ]
      }
    ]
  },
  then: '220 < age <= 225'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          225
        ]
      },
      {
        $lte: [
          '$_id',
          230
        ]
      }
    ]
  },
  then: '225 < age <= 230'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          230
        ]
      },
      {
        $lte: [
          '$_id',
          235
        ]
      }
    ]
  },
  then: '230 < age <= 235'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          
```

```

    }
  ]
},
then: '30 < age <= 35'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          35
        ]
      },
      {
        $lte: [
          '$_id',
          40
        ]
      }
    ]
  },
  then: '35 < age <= 40'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          40
        ]
      },
      {
        $lte: [
          '$_id',
          45
        ]
      }
    ]
  },
  then: '40 < age <= 45'
},

```

```

{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          45
        ]
      },
      {
        $lte: [
          '$_id',
          50
        ]
      }
    ]
  },
  then: '45 < age <= 50'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$_id',
          50
        ]
      },
      {
        $lte: [
          '$_id',
          55
        ]
      }
    ]
  },
  then: '50 < age <= 55'
},
{
  'case': {
    $and: [
      {
        $gt: [

```

```

        '$_id',
        55
    ],
    },
    {
        $lte: [
            '$_id',
            60
        ]
    }
]
},
then: '55 < age <= 60'
},
{
    'case': {
        $gt: [
            '$_id',
            60
        ]
    },
    then: 'age > 60'
}
],
'default': 'No age found'
}
}
}, {
    $group: {
        _id: '$age_range',
        count: {
            $sum: '$count'
        }
    }
}, {
    $sort: {
        _id: 1
    }
}
}]

```

Output:

_id	count
20 < age <= 25	528
25 < age <= 30	480
30 < age <= 35	482
35 < age <= 40	502
40 < age <= 45	450
45 < age <= 50	490
50 < age <= 55	492
55 < age <= 60	521
age <= 20	274
age > 60	1150

3) The distribution of clients by the district in descending orderPipeline 

\$group

\$sort

\$project

```

db.accountInfo.aggregate([
  $group: {
    _id: {
      district_id: '$district.district_id',
      district_name: '$district.district_name',
      region_name: '$district.region_name'
    },
    number_of_account: {
      $sum: 1
    }
  }, {
    $sort: {

```



```

    number_of_account: -1
  }
}, {
  $project: {
    _id: 1,
    number_of_account: 1,
    accounts_percentage: {
      $multiply: [
        {
          $divide: [
            '$number_of_account',
            4500
          ]
        },
        100
      ]
    }
  }
}
}]]))

```

Output:

_id.district_id	_id.district_name	_id.region_name	number_of_account	accounts_percentage
1	Hl.m. Praha	Prague	554	12.31111111
70	Karvina	north Moravia	152	3.377777778
74	Ostrava - mesto	north Moravia	135	3
54	Brno - mesto	south Moravia	128	2.844444444
64	Zlin	south Moravia	92	2.044444444
72	Olomouc	north Moravia	88	1.955555556
68	Frydek - Mistek	north Moravia	83	1.844444444
5	Kolin	central	65	1.444444444

		Bohemia		
46	Nachod	east Bohemia	59	1.3111111111
59	Kromeriz	south Moravia	57	1.2666666667
...

[refer to table 3.csv]

Demographic profiles of loan clients

4) idProfiling loan borrowers by age range and gender

- a) To know which age group and gender has the highest total loan amount

```
$addFields > $addFields > $project > $project > $group > $sort > $project
```

```
db.clientInfo.aggregate([
  $addFields: {
    birth_date: {
      $dateFromString: {
        dateString: '$birth_date',
        format: '%Y-%m-%d'
      }
    }
  },
  $addFields: {
    age: {
      $dateDiff: {
        startDate: '$birth_date',
        endDate: ISODate('1999-01-01T00:00:00.000Z'),
        unit: 'year'
      }
    }
  }
])
```

```

    }
  }, {
    $project: {
      _id: 1,
      client_id: 1,
      gender: 1,
      age: 1,
      account_id: '$account.account_id',
      loan_amount: '$loan.amount'
    }
  }, {
    $project: {
      _id: 1,
      client_id: 1,
      gender: 1,
      age: 1,
      account_id: 1,
      loan_amount: 1,
      age_range: {
        $switch: {
          branches: [
            {
              'case': {
                $lte: [
                  '$age',
                  20
                ]
              },
              then: 'age <= 20'
            },
            {
              'case': {
                $and: [
                  {
                    $gt: [
                      '$age',
                      20
                    ]
                  }
                ]
              },
              then: 'age > 20'
            }
          ]
        }
      }
    }
  }
}

```

```
{
  $lte: [
    '$age',
    25
  ]
}
],
},
then: '20 < age <= 25'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          25
        ]
      },
      {
        $lte: [
          '$age',
          30
        ]
      }
    ]
  },
  then: '25 < age <= 30'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          30
        ]
      },
      {

```

```

        $lte: [
          '$age',
          35
        ]
      }
    ]
  },
  then: '30 < age <= 35'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          35
        ]
      },
      {
        $lte: [
          '$age',
          40
        ]
      }
    ]
  },
  then: '35 < age <= 40'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          40
        ]
      },
      {
        $lte: [

```

```
        '$age',
        45
      ]
    }
  ]
},
then: '40 < age <= 45'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          45
        ]
      },
      {
        $lte: [
          '$age',
          50
        ]
      }
    ]
  },
  then: '45 < age <= 50'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          50
        ]
      },
      {
        $lte: [
          '$age',

```

```

        55
      ]
    }
  ]
},
then: '50 < age <= 55'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          55
        ]},
      {
        $lte: [
          '$age',
          60
        ]
      }
    ]
  },
  then: '55 < age <= 60'
},
{
  'case': {
    $gt: [
      '$age',
      60
    ]
  },
  then: 'age > 60'
}
],
'default': 'No age found'
}
}
}

```

```

}, {
  $group: {
    _id: {
      gender: '$gender',
      age: '$age_range',
      loan_account: '$loan.loan_account',
      amount: '$loan.amount'
    },
    total_loan_amount: {
      $sum: '$loan_amount'
    },
    mean_loan_amount: {
      $avg: '$loan_amount'
    }
  }
}, {
  $sort: {
    total_loan_amount: -1
  }
}, {
  $project: {
    _id: 0,
    gender: '$_id.gender',
    age: '$_id.age',
    total_loan_amount: 1,
    mean_loan_amount: 1
  }
}]})

```

Output:

age	gender	total_loan_amount	mean_loan_amount
30 < age <= 35	F	8780664	162604.8889
30 < age <= 35	M	8677236	170141.8824
55 < age <= 60	F	8602392	150919.1579

45 < age <= 50	M	8320512	169806.3673
50 < age <= 55	F	7695720	150896.4706
45 < age <= 50	F	6994884	155441.8667
25 < age <= 30	F	6795264	154437.8182
40 < age <= 45	F	6790416	144476.9362
20 < age <= 25	F	6523812	167277.2308
35 < age <= 40	F	6427896	146088.5455
20 < age <= 25	M	6381588	177266.3333
40 < age <= 45	M	6351228	132317.25
25 < age <= 30	M	6213744	138083.2
35 < age <= 40	M	6174660	143596.7442
50 < age <= 55	M	5707536	129716.7273
age > 60	M	5015712	135559.7838
55 < age <= 60	M	4904100	122602.5
age <= 20	F	4017192	154507.3846
age <= 20	M	3593448	211379.2941
age > 60	F	1571868	157186.8

b) To know which age group has the highest total loan amount

```
$addFields > $addFields > $project > $project > $group > $sort > $project
```

```
db.clientInfo.aggregate([
  $addFields: {
    birth_date: {
      $dateFromString: {
        dateString: '$birth_date',
        format: '%Y-%m-%d'
      }
    }
  }
])
```

```

    }
  }, {
    $addFields: {
      age: {
        $dateDiff: {
          startDate: '$birth_date',
          endDate: ISODate('1999-01-01T00:00:00.000Z'),
          unit: 'year'
        }
      }
    }
  }, {
    $project: {
      _id: 1,
      client_id: 1,
      age: 1,
      account_id: '$account.account_id',
      loan_amount: '$loan.amount'
    }
  }, {
    $project: {
      _id: 1,
      client_id: 1,
      age: 1,
      account_id: 1,
      loan_amount: 1,
      age_range: {
        $switch: {
          branches: [
            {
              'case': {
                $lte: [
                  '$age',
                  20
                ]
              },
              then: 'age <= 20'
            }
          ]
        }
      }
    }
  }, {
    {

```

```
'case': {
  $and: [
    {
      $gt: [
        '$age',
        20
      ]
    },
    {
      $lte: [
        '$age',
        25
      ]
    }
  ]
},
then: '20 < age <= 25'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          25
        ]
      },
      {
        $lte: [
          '$age',
          30
        ]
      }
    ]
  },
  then: '25 < age <= 30'
},
{
  'case': {
```

```
$and: [  
  {  
    $gt: [  
      '$age',  
      30  
    ]  
  },  
  {  
    $lte: [  
      '$age',  
      35  
    ]  
  }  
]  
,  
then: '30 < age <= 35'  
,  
{  
  'case': {  
    $and: [  
      {  
        $gt: [  
          '$age',  
          35  
        ]  
      },  
      {  
        $lte: [  
          '$age',  
          40  
        ]  
      }  
    ]  
  }  
],  
then: '35 < age <= 40'  
,  
{  
  'case': {  
    $and: [  

```

```
{
  $gt: [
    '$age',
    40
  ]
},
{
  $lte: [
    '$age',
    45
  ]
}
]
},
then: '40 < age <= 45'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          45
        ]
      },
      {
        $lte: [
          '$age',
          50
        ]
      }
    ]
  },
  then: '45 < age <= 50'
},
{
  'case': {
    $and: [
      {
```

```
    $gt: [
      '$age',
      50
    ]
  },
  {
    $lte: [
      '$age',
      55
    ]
  }
]
},
then: '50 < age <= 55'
},
{
  'case': {
    $and: [
      {
        $gt: [
          '$age',
          55
        ]
      },
      {
        $lte: [
          '$age',
          60
        ]
      }
    ]
  }
},
then: '55 < age <= 60'
},
{
  'case': {
    $gt: [
      '$age',
      60
    ]
  }
}
```

```

        ]
      },
      then: 'age > 60'
    }
  ],
  'default': 'No age found'
}
}
}, {
  $group: {
    _id: {
      age: '$age_range',
      loan_account: '$loan.loan_account',
      amount: '$loan.amount'
    },
    total_loan_amount: {
      $sum: '$loan_amount'
    },
    mean_loan_amount: {
      $avg: '$loan_amount'
    }
  }
}, {
  $sort: {
    total_loan_amount: -1
  }
}, {
  $project: {
    _id: 0,
    age: '$_id.age',
    total_loan_amount: 1,
    mean_loan_amount: 1
  }
}]]))

```

Output:

age	total_loan_amount	mean_loan_amount
30 < age <= 35	17457900	166265.7143
45 < age <= 50	15315396	162929.7447
55 < age <= 60	13506492	139242.1856
50 < age <= 55	13403256	141086.9053
40 < age <= 45	13141644	138333.0947
25 < age <= 30	13009008	146168.6292
20 < age <= 25	12905400	172072
35 < age <= 40	12602556	144856.9655
age <= 20	7610640	176991.6279
age > 60	6587580	140161.2766

c) To know which gender has the highest total loan amount



```

db.clientInfo.aggregate([
  $project: {
    _id: 1,
    client_id: 1,
    gender: 1,
    account_id: '$account.account_id',
    loan_amount: '$loan.amount'
  }, {
    $group: {
      _id: {
        gender: '$gender',
        loan_account: '$loan.loan_account',
        amount: '$loan.amount'
      },

```



```

    total_loan_amount: {
      $sum: '$loan_amount'
    },
    mean_loan_amount: {
      $avg: '$loan_amount'
    }
  }, {
    $sort: {
      total_loan_amount: -1
    }
  }, {
    $project: {
      _id: 0,
      gender: '$_id.gender',
      total_loan_amount: 1,
      mean_loan_amount: 1
    }
  }
}]

```

Output:

gender	total_loan_amount	mean_loan_amount
F	64200108	153957.0935
M	61339764	149609.1805

5. The amount of loans for different districts

- a) The amount of loans for the districts with the highest total amount of loan

Pipeline ▾

\$group > \$sort > \$project > \$out

```

db.accountInfo.aggregate([
  $group: {
    _id: {

```

```

    district_id: '$district.district_id'
  },
  total_loan_amount: {
    $sum: '$loan.amount'
  },
  mean_loan_amount: {
    $avg: '$loan.amount'
  }
}, {
  $sort: {
    mean_loan_amount: -1
  }
}, {
  $project: {
    district_id: '$_id.district_id',
    total_loan_amount: 1,
    mean_loan_amount: 1
  }
}, {
  $out: '5a'
}]})

```

Output:

district_id	mean_loan_amount	total_loan_amount
46	294,730.00	1,768,380.00
14	251,365.50	2,010,924.00
3	243,466.00	1,460,796.00
69	242,304.00	1,938,432.00
6	232,886.67	2,095,980.00

(ref full table)

a i) Find districts where total loan > 150000.

```

db.5a.find(
  {total_loan_amount: {$gt: 150000}}
)

```

district_id	mean_loan_amount	total_loan_amount
46	294730	1768380
14	251365.5	2010924
69	242304	1938432
6	232886.6667	2095980
5	190116	1901160
64	177221.6471	3012768
50	175007	2100084
54	168725	4049400
19	168335.1429	2356692
62	168204	1682040
47	164836.8	1648368
72	163399.7143	2287596
74	163011	3260220
52	158082.8571	2213160
1	153957.2857	12932412
68	128359.5	2053752
70	127492.5	3059820
38	118465.8462	1540056

b) The amount of loans for the 5 districts with the lowest total amount of loan

Pipeline
\$group
\$sort
\$project
\$limit
\$out

```

db.accountInfo.aggregate([
  $group: {
    _id: {
      district_id: '$district.district_id'
    },
    total_loan_amount: {
      $sum: '$loan.amount'
    }
  }
])

```

```

    },
    mean_loan_amount: {
      $avg: '$loan.amount'
    }
  }
}, {
  $sort: {
    mean_loan_amount: 1
  }
}, {
  $project: {
    district_id: '$_id.district_id',
    total_loan_amount: 1,
    mean_loan_amount: 1
  }
}, {
  $limit: 5
}, {
  $out: '5b'
}]})

```

Output:

district_id	mean_loan_amount	total_loan_amount
30	74262	148524
32	77114	462684
77	81830	490980
15	83475.4	584328
25	90292	270876

6) Do people from places with higher or lower salaries tend to borrow more?

Pipeline  \$addFields \$bucket \$project \$out

```

db.accountInfo.aggregate([
  $addFields: {
    'district.average_salary': {
      $toInt: '$district.average_salary'
    }
  }, {
    $bucket: {
      groupBy: '$district.average_salary',
      boundaries: [7000, 8000, 9000, 10000, 11000, 12000, 13000],
      'default': 'Others',
      output: {
        mean_loan: {
          $avg: '$loan.amount'
        },
        mean_salary: {
          $avg: '$district.average_salary'
        }
      }
    }, {
      $project: {
        _id: 0,
        salary_group_lower_bound: '$_id',
        mean_loan: {
          $round: [
            '$mean_loan',
            1
          ]
        },
        mean_salary: {
          $round: [
            '$mean_salary',
            1
          ]
        }
      }, {
        $out: '6'
      }
    ]
  })

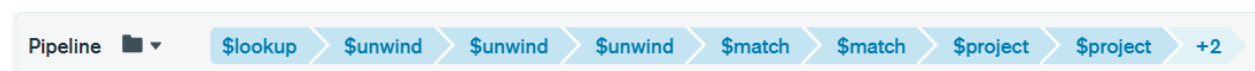
```

Output:

salary_group_lower_bound	mean_loan	mean_salary
8000	155459	8587.2
9000	140600.7	9525.4
10000	155327.8	10388
11000	123504	11277
12000	153957.3	12541

Credit Cards

7) What is the client_id, account_id and credit transactions for clients who own the goldtype card?



```
db.clientInfo.aggregate([
  {
    $lookup: {
      from: 'accountStatement',
      localField: 'account.account_id',
      foreignField: 'account_id',
      as: 'transactions'
    }
  }, {
    $unwind: {
      path: '$transactions',
      includeArrayIndex: 'string',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $unwind: {
      path: '$account.account_id'
    }
  }, {
    $unwind: {
```

```

        path: '$transactions.transactions'
      }
    }, {
      $match: {
        'transactions.transactions.type': 'credit'
      }
    }, {
      $match: {
        'cards.card_type': 'gold'
      }
    }, {
      $project: {
        _id: 1,
        client_id: 1,
        card_type: '$cards.card_type',
        account_id: '$account.account_id',
        transaction_amount: '$transactions.transactions.amount'
      }
    }, {
      $group: {
        _id: {
          client_id: '$client_id',
          account_id: '$account_id',
          card_type: '$card_type'
        },
        total_transaction_amount: {
          $sum: '$transaction_amount'
        }
      }
    }, {
      $project: {
        _id: 0,
        client_id: '$_id.client_id',
        card_type: '$_id.card_type',
        account_id: '$_id.account_id',
        total_transaction_amount: 1
      }
    }, {
      $sort: {

```

```

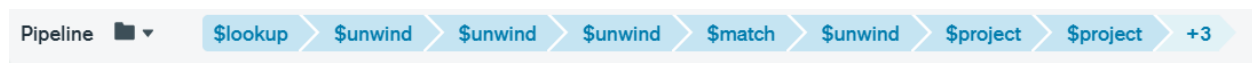
    client_id: 1
  }
}]

```

Output:

client_id	account_id	card_type	total_transaction_amount
9	7	gold	623,775
41	33	gold	1,652,341
79	68	gold	1,857,968
326	270	gold	1,215,611
548	456	gold	3,683,681
562	468	gold	821,280

8) Detect potential gold-card holders



```

db.clientInfo.aggregate([
  $lookup: {
    from: 'accountStatement',
    localField: 'account.account_id',
    foreignField: 'account_id',
    as: 'transactions'
  }, {
    $unwind: {
      path: '$transactions',
      includeArrayIndex: 'string',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $unwind: {
      path: '$account.account_id'
    }
  }
])

```



```

    }
  }, {
    $unwind: {
      path: '$transactions.transactions'
    }
  }, {
    $match: {
      'transactions.transactions.type': 'credit'
    }
  }, {
    $unwind: {
      path: '$cards.card_type'
    }
  }, {
    $project: {
      _id: 1,
      disp_id: 1,
      client_id: 1,
      card_type: '$cards.card_type',
      account_id: '$account.account_id',
      transaction_amount: '$transactions.transactions.amount'
    }
  }, {
    $group: {
      _id: {
        account_id: '$account_id',
        card_type: '$card_type'
      },
      total_transaction_amount: {
        $sum: '$transaction_amount'
      }
    }
  }, {
    $project: {
      _id: 0,
      account_id: '$_id.account_id',
      card_type: '$_id.card_type',
      total_transaction_amount: 1
    }
  }
}

```

```

}, {
  $sort: {
    account_id: 1
  }
}]

```

Output:

account_id	card_type	total_transaction_amount
7	gold	623,775
14	classic	408,448
33	gold	1,652,341
34	classic	2,181,988
43	junior	1,436,880
48	classic	628,885
51	junior	702,325
65	classic	701,415
66	classic	2,973,820
68	gold	1,857,968
...

[refer to table 8.csv]

9) Client profiling for clients who use credit cards

a) By Gender

Pipeline  \$lookup > \$unwind > \$unwind > \$unwind > \$match > \$unwind > \$unwind > \$project > +1

```

db.clientInfo.aggregate([
  $lookup: {
    from: 'accountStatement',
    localField: 'account.account_id',
    foreignField: 'account_id',
    as: 'transactions'
  }
])

```

```

    }
  }, {
    $unwind: {
      path: '$transactions',
      includeArrayIndex: 'string',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $unwind: {
      path: '$transactions.transactions'
    }
  }, {
    $match: {
      'transactions.transactions.type': 'credit'
    }
  }, {
    $unwind: {
      path: '$cards.card_type'
    }
  }, {
    $project: {
      _id: 1,
      gender: 1,
      client_id: 1,
      card_type: '$cards.card_type',
      account_id: '$account.account_id',
      transaction_amount: '$transactions.transactions.amount'
    }
  }, {
    $group: {
      _id: {
        gender: '$gender',
        card_type: '$card_type'
      },
      total_transaction_amount: {
        $sum: '$transaction_amount'
      }
    }
  }, {

```

```
$project: {
  _id: 0,
  gender: '$_id.gender',
  card_type: '$_id.card_type',
  total_transaction_amount: 1
}
}]
```

Output:

gender	card_type	total_transaction_amount
F	gold	3,894,859
M	gold	5,959,797
F	junior	7,058,889
M	junior	1,060,704
M	classic	43,707,008
F	classic	47,875,861

b) By District

Pipeline  \$lookup > \$unwind > \$unwind > \$unwind > \$unwind > \$match > \$unwind > \$project > +2

```
db.clientInfo.aggregate([
  [
    {
      $lookup: {
        from: 'accountStatement',
        localField: 'account.account_id',
        foreignField: 'account_id',
        as: 'transactions'
      }
    }
  ]
])
```

```

    }
  }, {
    $unwind: {
      path: '$transactions',
      includeArrayIndex: 'string',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $unwind: {
      path: '$district.district_id'
    }
  }, {
    $unwind: {
      path: '$account.account_id'
    }
  }, {
    $unwind: {
      path: '$transactions.transactions'
    }
  }, {
    $match: {
      'transactions.transactions.type': 'credit'
    }
  }, {
    $unwind: {
      path: '$cards.card_type'
    }
  }, {
    $project: {
      _id: 1,
      client_id: 1,
      district_id: '$district.district_id',
      card_type: '$cards.card_type',
      account_id: '$account.account_id',
      transaction_amount: '$transactions.transactions.amount'
    }
  }, {
    $group: {
      _id: {

```

```

    district_id: '$district_id',
    card_type: '$card_type'
  },
  total_transaction: {
    $sum: '$transaction_amount'
  }
}
}, {
  $project: {
    _id: 0,
    district_id: '$_id.district_id',
    card_type: '$_id.card_type',
    total_transaction: 1
  }
}, {
  $sort: {
    district_id: 1
  }
}
}]

```

Output:

district_id	card_type	total_transaction_amount
1	classic	11,578,695
27	classic	6,029,138
70	classic	5,687,121
74	classic	5,172,302
49	gold	4,504,961
47	classic	3,938,938
60	classic	3,798,656
46	classic	3,600,395
37	classic	3,594,848
39	junior	318,899
49	gold	4,504,961
16	junior	832,355

41	classic	3,560,933
...

[refer to table 9.csv]

Account

10) Detect inactive accounts by querying account_id where the last transfer was more than 1 year ago

`$project > $unwind > $addFields > $addFields > $match > $project`

```
db.accountStatements.aggregate([
  $project: {
    id: 1,
    account_id: 1,
    transactions: {
      $slice: [
        '$transactions.date',
        -1
      ]
    }
  }, {
    $unwind: {
      path: '$transactions',
      preserveNullAndEmptyArrays: true
    }
  }, {
    $addFields: {
      last_transaction_date: {
        $dateFromString: {
          dateString: '$transactions',
          format: '%Y-%m-%d'
        }
      }
    }
  }
])
```

```

}, {
  $addFields: {
    DayDiff: {
      $dateDiff: {
        startDate: '$last_transaction_date',
        endDate: ISODate('1999-01-01T00:00:00.000Z'),
        unit: 'day'
      }
    }
  }
}, {
  $match: {
    DayDiff: {
      $gt: 365
    }
  }
}, {
  $project: {
    _id: 0,
    account_id: 1,
    last_transaction_date: 1
  }
}]})

```

Output:

account_id	last_transaction_date
6	1996-01-20T00:00:00.000Z
55	1996-07-18T00:00:00.000Z
63	1997-11-11T00:00:00.000Z
83	1997-06-24T00:00:00.000Z
112	1997-12-16T00:00:00.000Z
123	1997-09-18T00:00:00.000Z
146	1995-12-21T00:00:00.000Z
148	1996-07-05T00:00:00.000Z

151	1997-06-17T00:00:00.000Z
157	1997-12-23T00:00:00.000Z
162	1997-01-06T00:00:00.000Z
163	1997-12-02T00:00:00.000Z
187	1997-01-06T00:00:00.000Z
220	1996-07-22T00:00:00.000Z
279	1995-10-31T00:00:00.000Z
288	1996-11-07T00:00:00.000Z
305	1997-11-12T00:00:00.000Z
306	1997-12-14T00:00:00.000Z
313	1997-07-31T00:00:00.000Z
317	1996-12-19T00:00:00.000Z
324	1997-11-16T00:00:00.000Z
353	1997-05-21T00:00:00.000Z
359	1997-02-28T00:00:00.000Z
368	1996-12-17T00:00:00.000Z
...	...

[refer to table 10.csv]

QUESTION 3

While a NoSQL database has its great advantage in schema flexibility and scalability, it also comes with its drawbacks, i.e. it has very low flexibility in queries. As the query patterns of a NoSQL database is very much predetermined by its database design, it is not as flexible as SQL that can be used to explore patterns and relationships freely with the join function. While there is \$lookup available, it has significantly slower performance compared to that of SQL's, while also being more brittle.

In the case of Project 2, we can see that the size of data is still relatively small, thus there is no scaling problem. As the goal of our query is to find out client behaviors and patterns to improve service and marketing, it is important to retain the ability of flexible query provided by SQL to allow for more comprehensive analysis instead of iterating on well-understood parameters that might bring confirmation bias. So, for this project, SQL is preferred over NoSQL.

PRESENTATION VIDEO LINK

- <https://www.youtube.com/watch?v=IdgCBQlpQ2g>