



ĐẠI HỌC BÁCH KHOA HÀ NỘI

BÀI THUYẾT TRÌNH

Môn Kho dữ liệu và
Kinh doanh thông minh

ĐỀ TÀI: BẢO HIỂM Ô TÔ

GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN DANH TÚ



NHÓM 9



THÀNH VIÊN CỦA NHÓM 9



Trương Khắc Nam
20210634



Trần Thị Hồng Ngọc
20216862



Nguyễn Trung Nguyên
20216865

MỤC LỤC



- 1. KHẢO SÁT**
- 2. PHÂN TÍCH VÀ THIẾT KẾ**
- 3. XÂY DỰNG HỆ THỐNG**
- 4. TỔNG KẾT**



1. KHẢO SÁT



➤ Tổng quan

- Quy trình nghiệp vụ
- Yêu cầu phân tích
- Quy mô dữ liệu
- ERD hệ thống OLTP
- Hệ thống chỉ số - cây phân tích dashboard

1

KHẢO SÁT

TỔNG QUAN



Bài toán



Hệ thống

1

KHẢO SÁT

TỔNG QUAN

BÀI TOÁN



Dữ liệu ngành bảo hiểm ô tô rất phức tạp bao gồm nhiều loại thông tin khác nhau và các yếu tố ảnh hưởng



Cần hệ thống tổng hợp và phân tích dữ liệu ngành bảo hiểm ô tô



Hệ thống tổng hợp và phân tích dữ liệu ngành bảo hiểm ô tô là gì?

-  Là một hệ thống công nghệ thông tin phức tạp, được nhiều công ty bảo hiểm áp dụng để thu thập, lưu trữ, quản lý và phân tích dữ liệu liên quan đến bảo hiểm ô tô để cải thiện quyết định kinh doanh và dịch vụ khách hàng.
-  Hệ thống này tích hợp các công nghệ như AI, Machine Learning, Big Data và Analytics, Telematics và IoT, ...

Tại sao cần hệ thống tổng hợp và phân tích dữ liệu ngành bảo hiểm ô tô?

-  Đánh giá rủi ro chính xác hơn
-  Tối ưu hóa quyết định kinh doanh
-  Quản lý yêu cầu bồi thường hiệu quả
-  Bảo mật cao
-  Phát hiện và phòng ngừa gian lận
-  Cung cấp dịch vụ cá nhân hóa
-  Tối ưu hóa quản lý tài chính

1. KHẢO SÁT

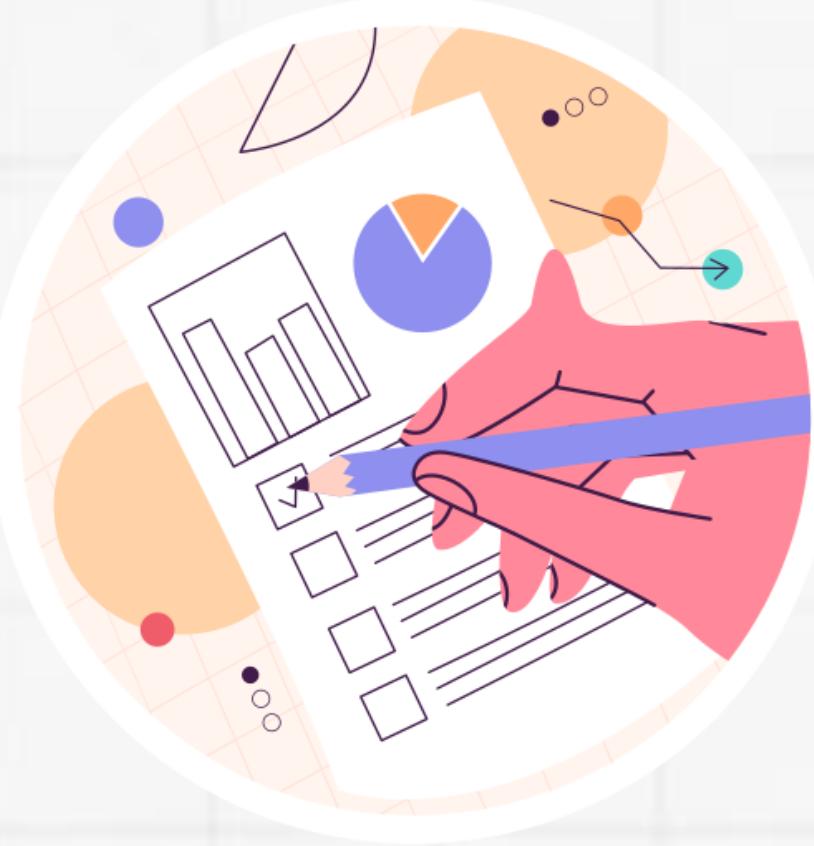


- Tổng quan
- Quy trình nghiệp vụ
- Yêu cầu phân tích
- Quy mô dữ liệu
- ERD hệ thống OLTP
- Hệ thống chỉ số - cây phân tích dashboard

1

KHẢO SÁT

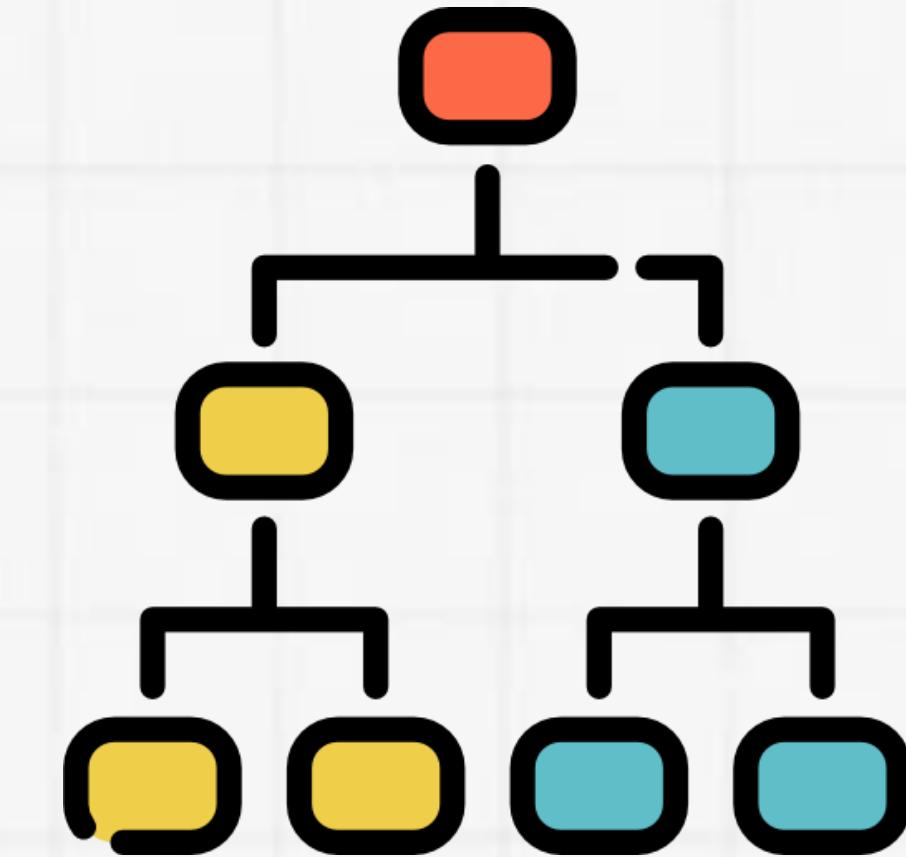
QUY TRÌNH NGHIỆP VỤ



Khảo sát nghiệp vụ



Luồng nghiệp vụ



Cây nghiệp vụ

❖ Các hệ thống đang sử dụng:



Hệ thống chăm sóc khách hàng

Quản lý mối quan hệ với khách hàng, giải quyết thắc mắc, phản hồi và hỗ trợ khách hàng trong quá trình sử dụng dịch vụ bảo hiểm.



Hệ thống quản lý thông tin

Lưu trữ và xử lý thông tin liên quan đến khách hàng, hợp đồng và các giao dịch liên quan đến bảo hiểm.



Hệ thống quản lý thanh toán

Xử lý việc thu và thanh toán phí bảo hiểm và các khoản bồi thường cho khách hàng.



Hệ thống quản lý tài chính

Theo dõi và báo cáo về tình hình tài chính của công ty, quản lý dòng tiền và dự trữ vốn.



Hệ thống quản lý sự cố và tai nạn

Quản lý các yêu cầu bồi thường liên quan đến sự cố và tai nạn, từ việc báo cáo đến xử lý và giải quyết các yêu cầu.



1

KHẢO SÁT

QUY TRÌNH NGHIỆP VỤ

LUÔNG NGHIỆP VỤ



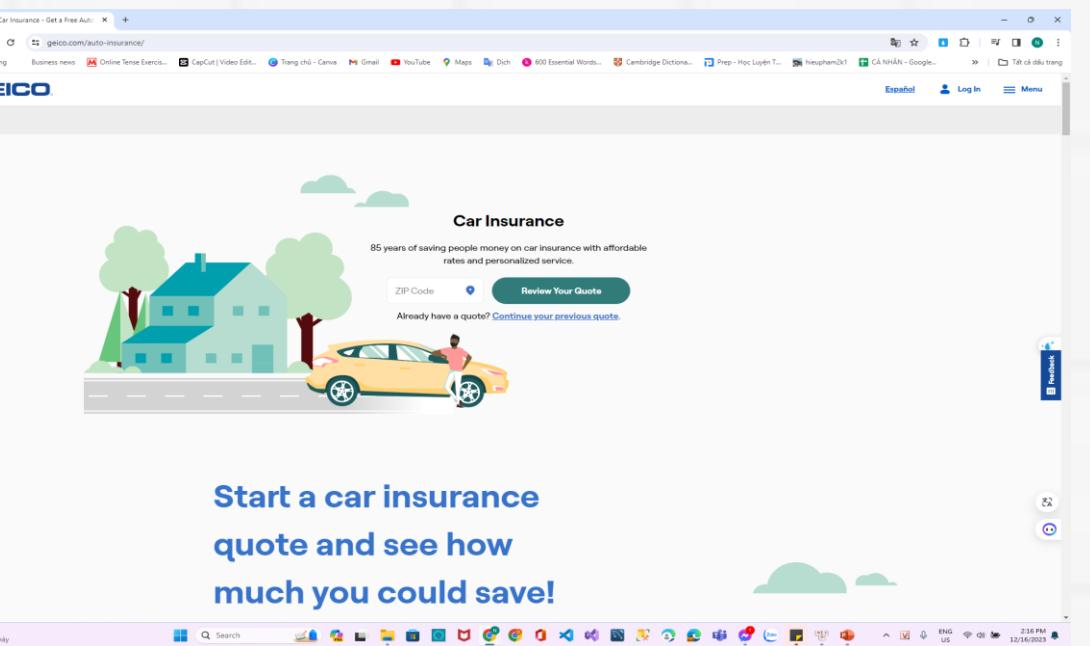
Quy trình đăng ký
mua bảo hiểm



Quy trình
thanh toán



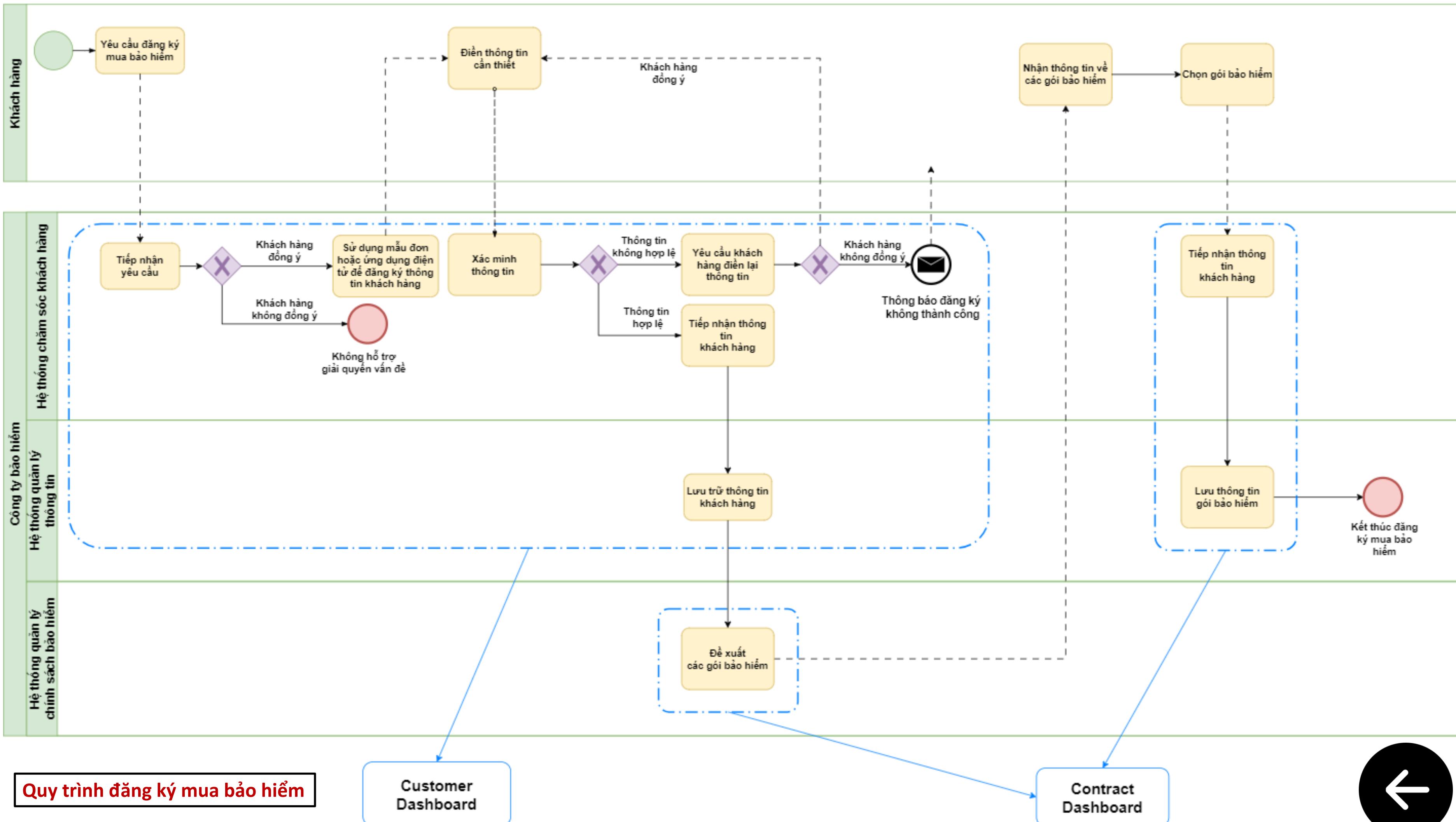
Hợp đồng ký kết
mua bảo hiểm ô tô

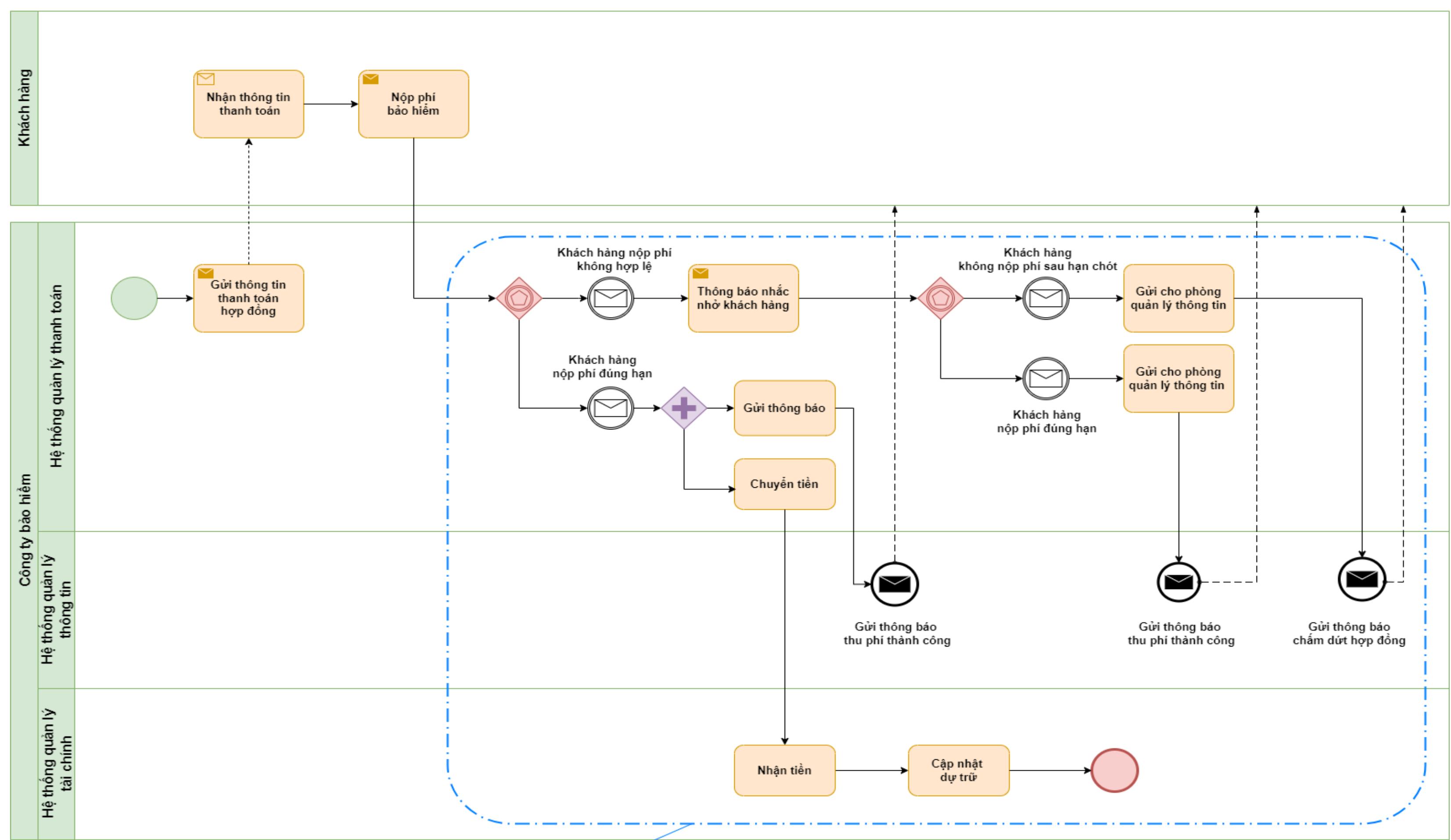


Trang web mua bảo hiểm
ô tô trực tuyến

Quy trình
yêu cầu bồi thường





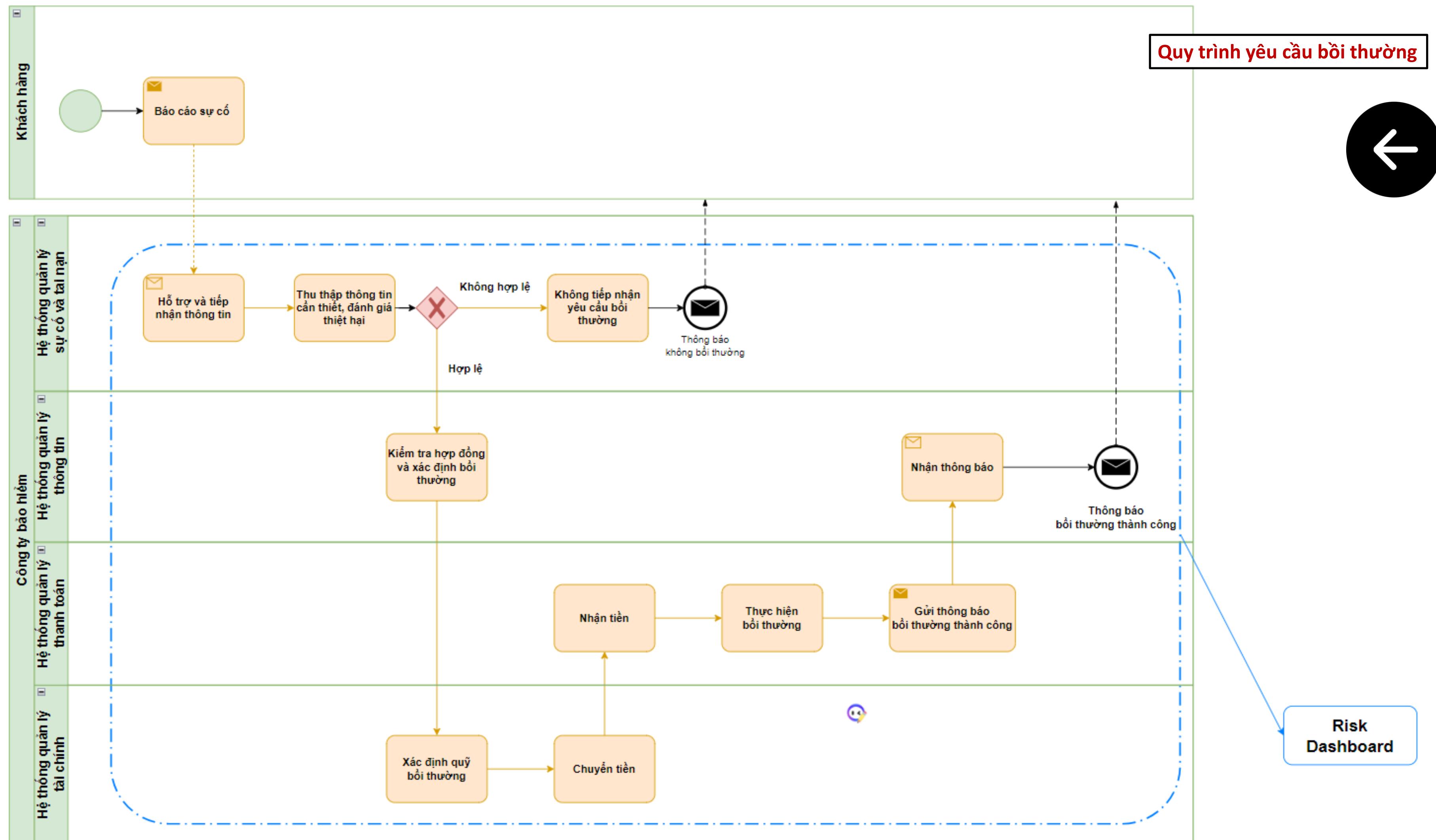


Quy trình thanh toán

Financial
Dashboard



Quy trình yêu cầu bồi thường

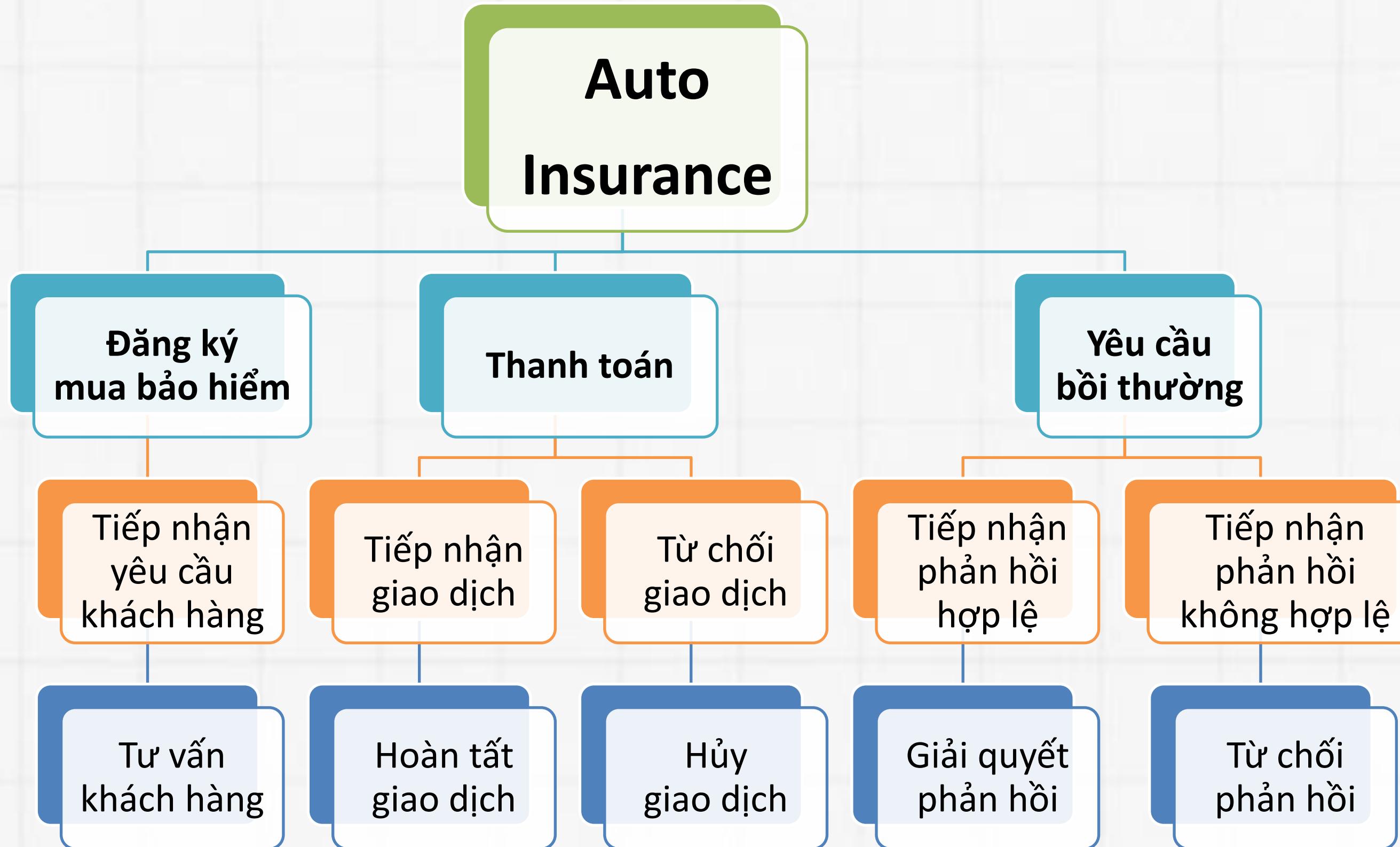


1

KHẢO SÁT

QUY TRÌNH NGHIỆP VỤ

CÂY NGHIỆP VỤ



1. KHẢO SÁT



- Tổng quan
- Quy trình nghiệp vụ
- Yêu cầu phân tích
- Quy mô dữ liệu
- ERD hệ thống OLTP
- Hệ thống chỉ số - cây phân tích dashboard

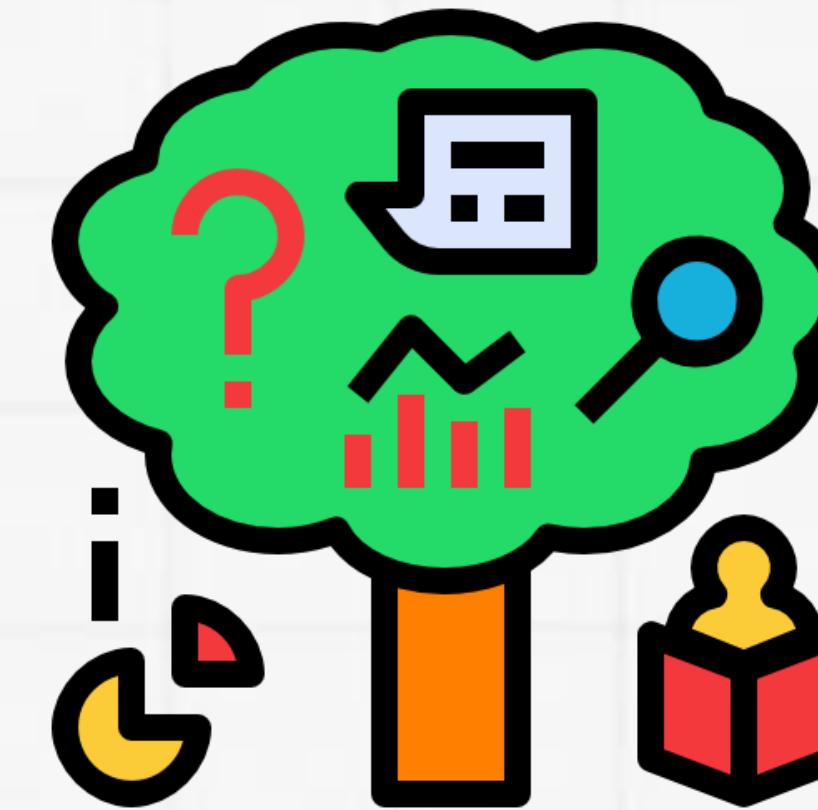
1

KHẢO SÁT

YÊU CẦU PHÂN TÍCH



CHỦ ĐIỂM & KHÍA CẠNH



CÂY PHÂN TÍCH

Chủ điểm:

- 🚗 Số lượng hợp đồng
- 🚗 Số lượng khách hàng
- 🚗 Tài chính

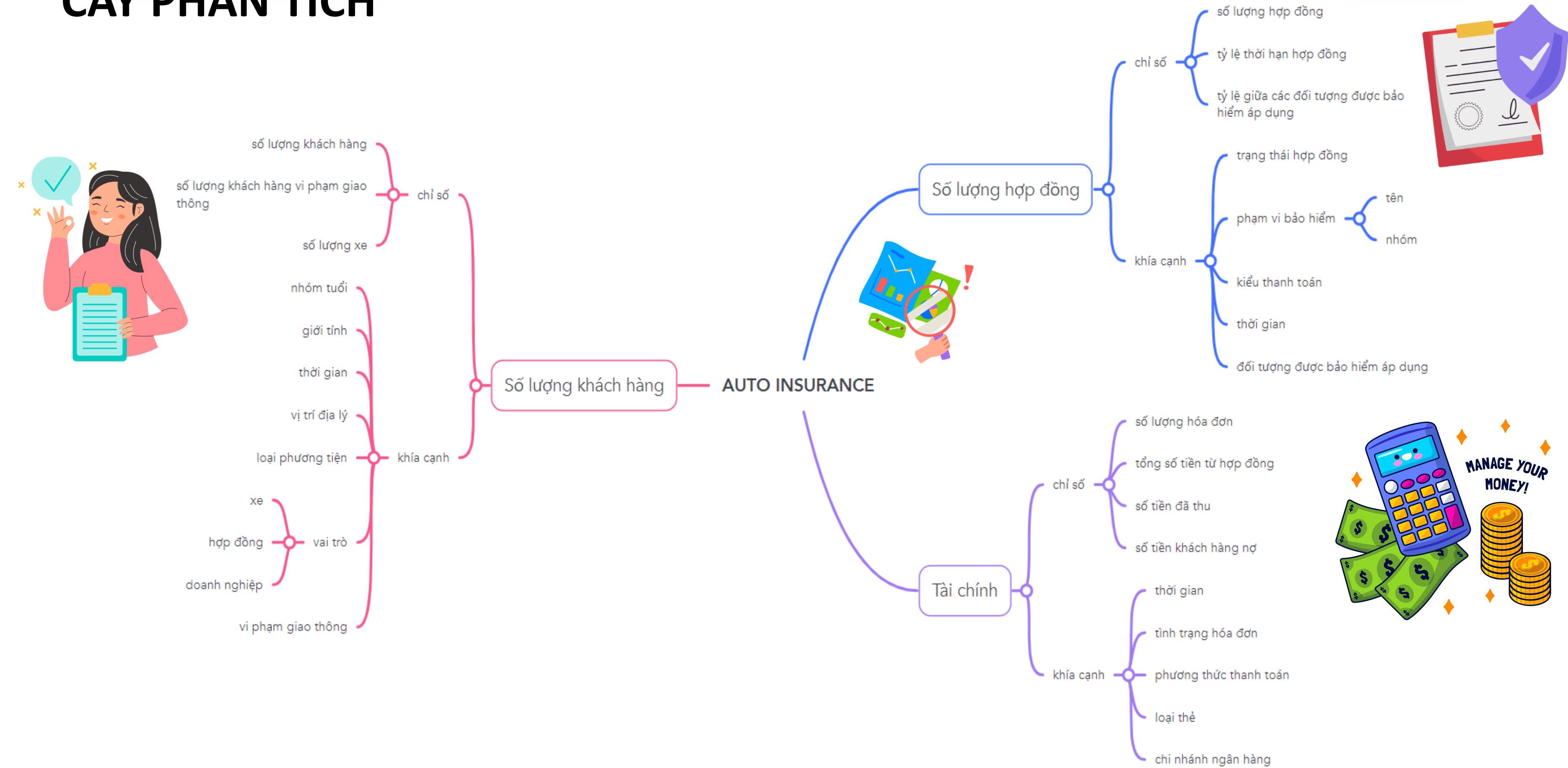


Khía cạnh:

- 🚗 Thời gian
- 🚗 Vị trí địa lý
- 🚗 Đặc điểm của hợp đồng
- 🚗 Thông tin của khách hàng
- 🚗 Thông tin của xe
- 🚗 Hành vi vi phạm giao thông của khách hàng
- 🚗 Giao dịch



CÂY PHÂN TÍCH



1. KHẢO SÁT

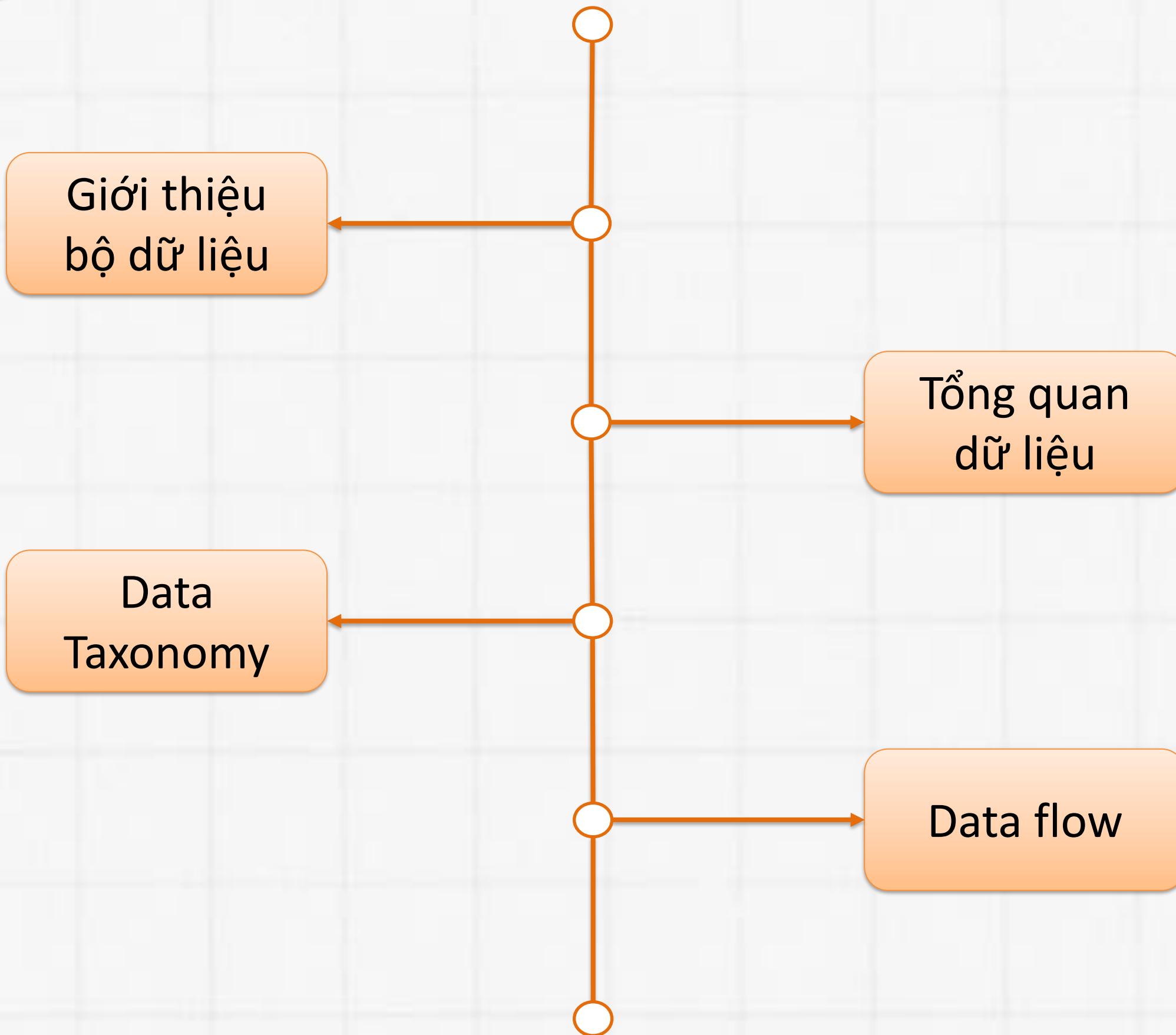


- Tổng quan
- Quy trình nghiệp vụ
- Yêu cầu phân tích
- Quy mô dữ liệu
- ERD hệ thống OLTP
- Hệ thống chỉ số - cây phân tích dashboard

1

KHẢO SÁT

QUY MÔ DỮ LIỆU





Nguồn dữ liệu

Codeproject.com + Geico.com + sinh tự động bằng Python



Tên bộ dữ liệu

Auto Insurance



Nội dung bộ dữ liệu

Bộ dữ liệu lưu giữ thông tin chính sách bảo hiểm ô tô, thông tin hóa đơn và thanh toán, thông tin xe, thông tin chủ hợp đồng, hồ sơ vi phạm của một công ty bảo hiểm nhỏ



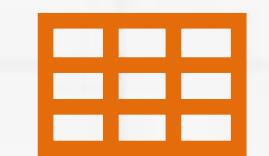
Thời gian

2017 - 2020



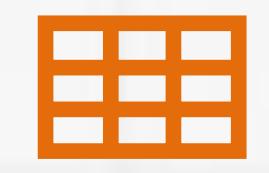
Kích thước bộ dữ liệu

433.8 MB



Số bảng

13 bảng



Số trường

104 trường



Tần suất cập nhật dữ liệu

112.36 MB



	Table Name	Number of Columns	Table Size (MB)
0	bill	7	17.03
1	coverage	7	22.55
2	driver	12	45.08
3	driver_trafficViolation_record	4	17.55
4	driveraddress	9	29.06
5	paymentdetail	17	174.29
6	policy	8	15.52
7	policy_coverage	4	11.55
8	policyeditlog	6	20.03
9	trafficViolationcode	5	15.52
10	vehicle	12	18.03
11	vehicle_coverage	5	13.55
12	vehicle_driver	8	20.55

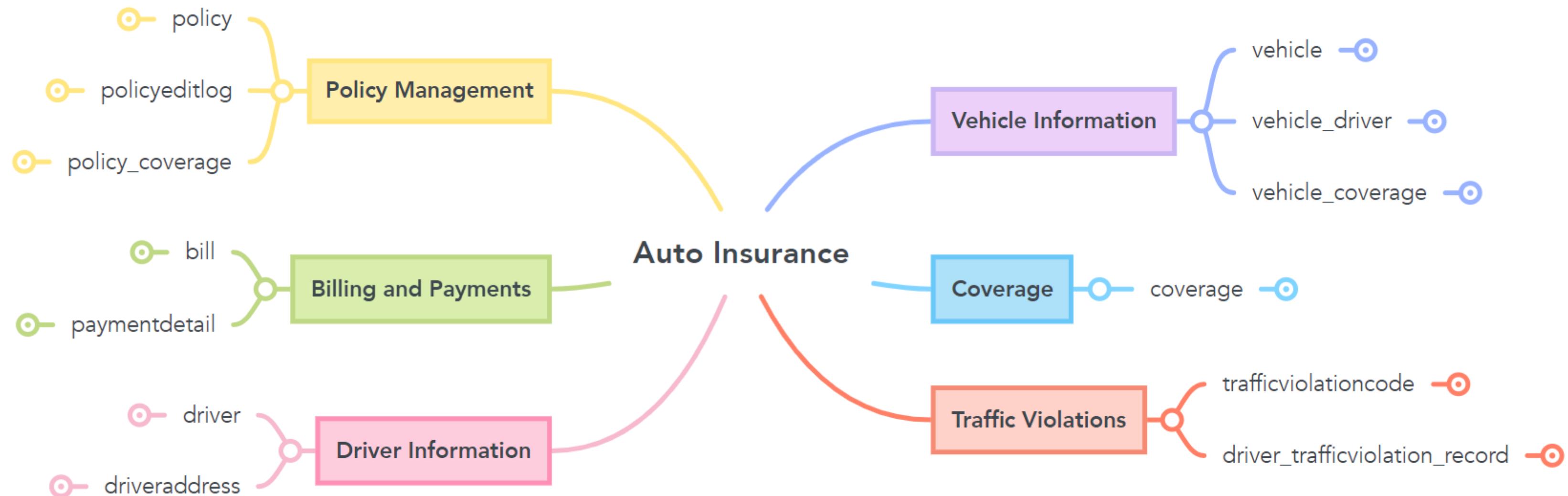


1

KHẢO SÁT

QUY MÔ DỮ LIỆU

DATA TAXONOMY



1

KHẢO SÁT

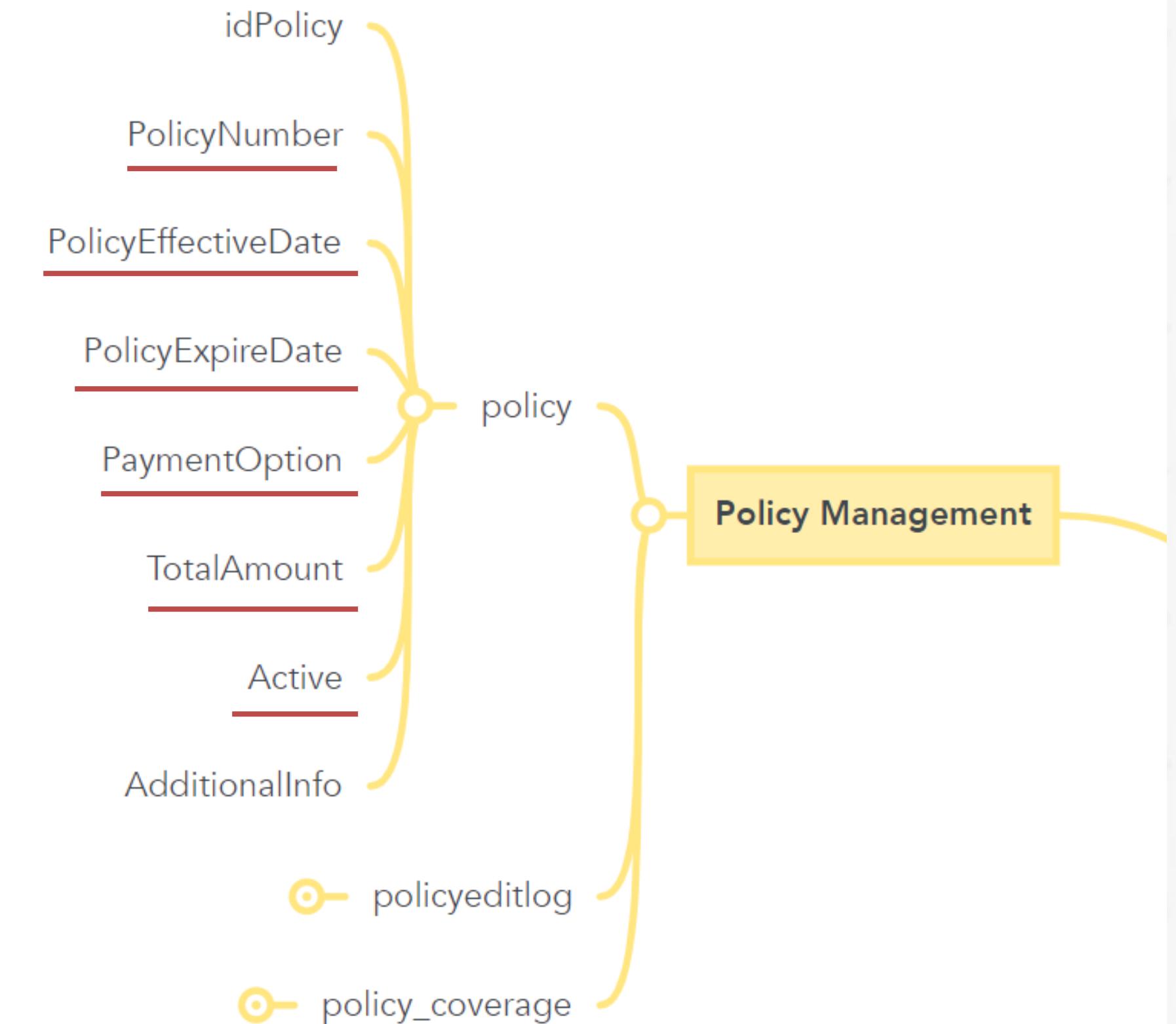
QUY MÔ DỮ LIỆU

DATA TAXONOMY

Bảng policy: chứa thông tin chung về chính sách

Một số thuộc tính quan trọng:

- PolicyNumber: số chính sách
- PolicyEffectiveDate: ngày bắt đầu hiệu lực
- PolicyExpireDate: ngày hết hạn hiệu lực
- PaymentOption: tần suất thanh toán
- TotalAmount: tổng số tiền chính sách
- Active: 0 là chính sách hết hạn, 1 là chính sách còn hạn



1

KHẢO SÁT

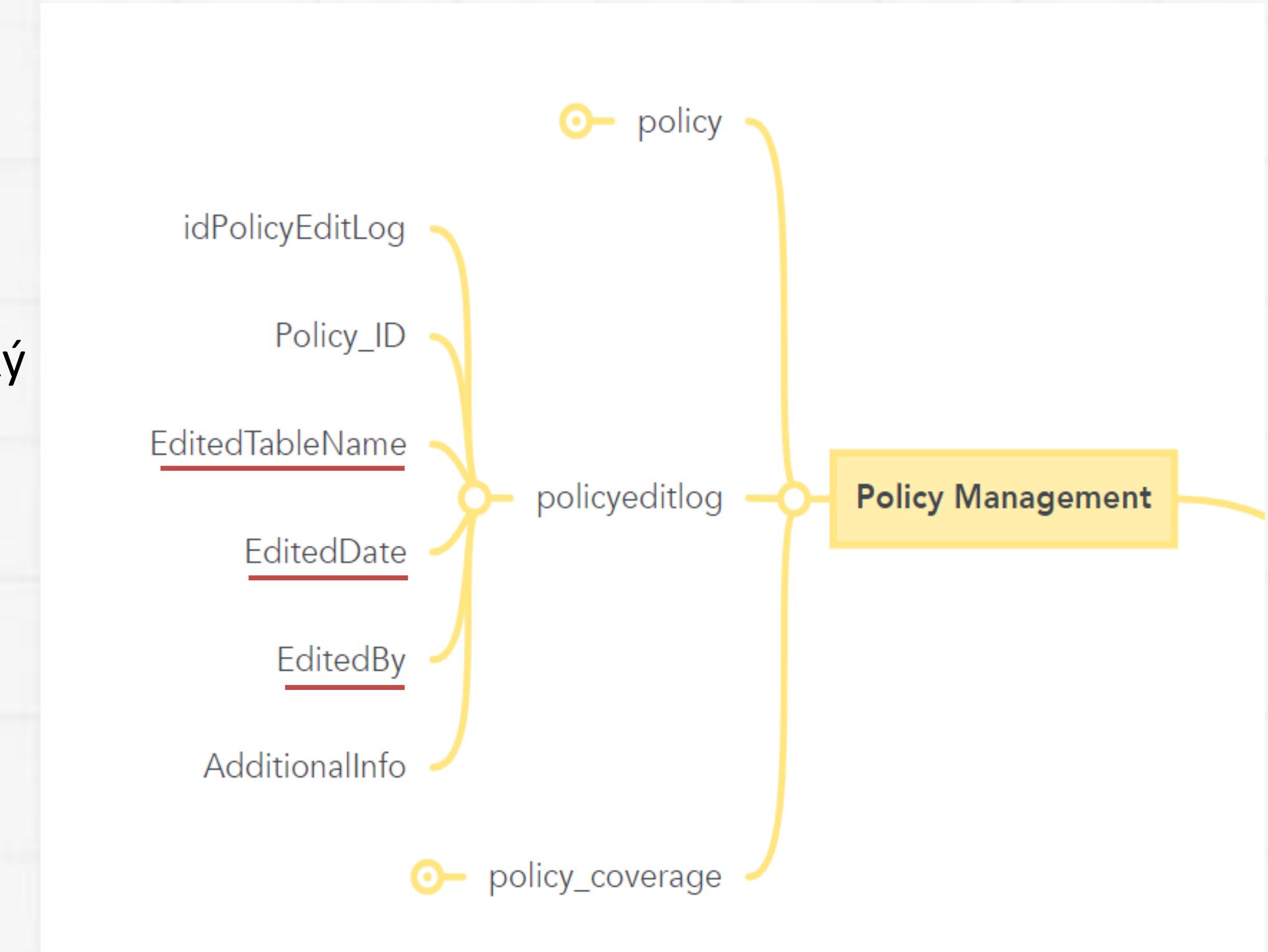
QUY MÔ DỮ LIỆU

DATA TAXONOMY

Bảng policyeditlog: lưu giữ thông tin nhật ký

Một số thuộc tính quan trọng:

- EditedTableName: tên bảng nếu bản ghi được cập nhật
- EditedDate: ngày bản ghi được cập nhật
- EditedBy: người cập nhật bản ghi

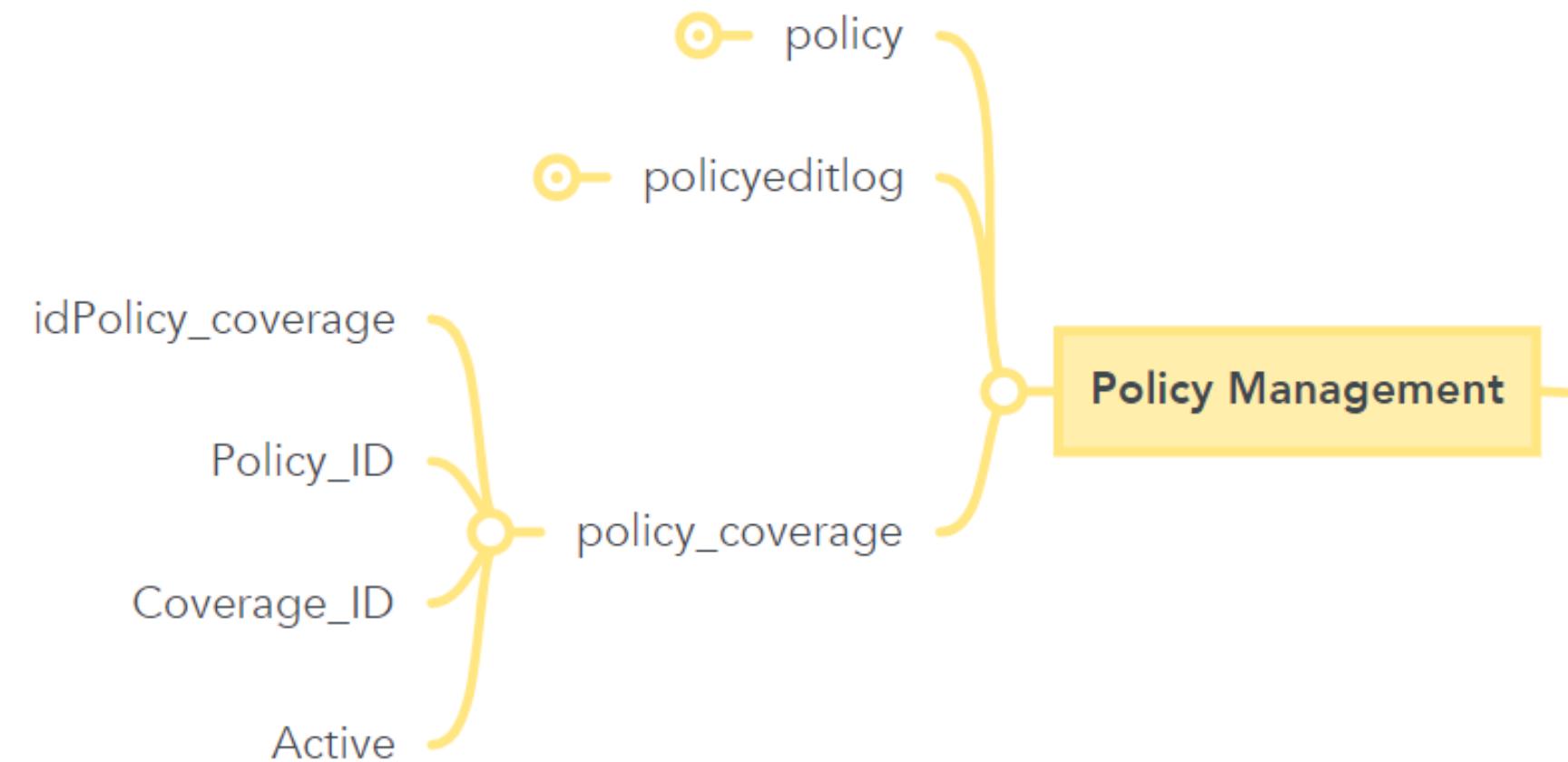


1

KHẢO SÁT

QUY MÔ DỮ LIỆU

DATA TAXONOMY



Bảng policy_coverage: theo dõi các thông tin về việc áp dụng các loại bảo hiểm cụ thể cho từng hợp đồng bảo hiểm

1

KHẢO SÁT

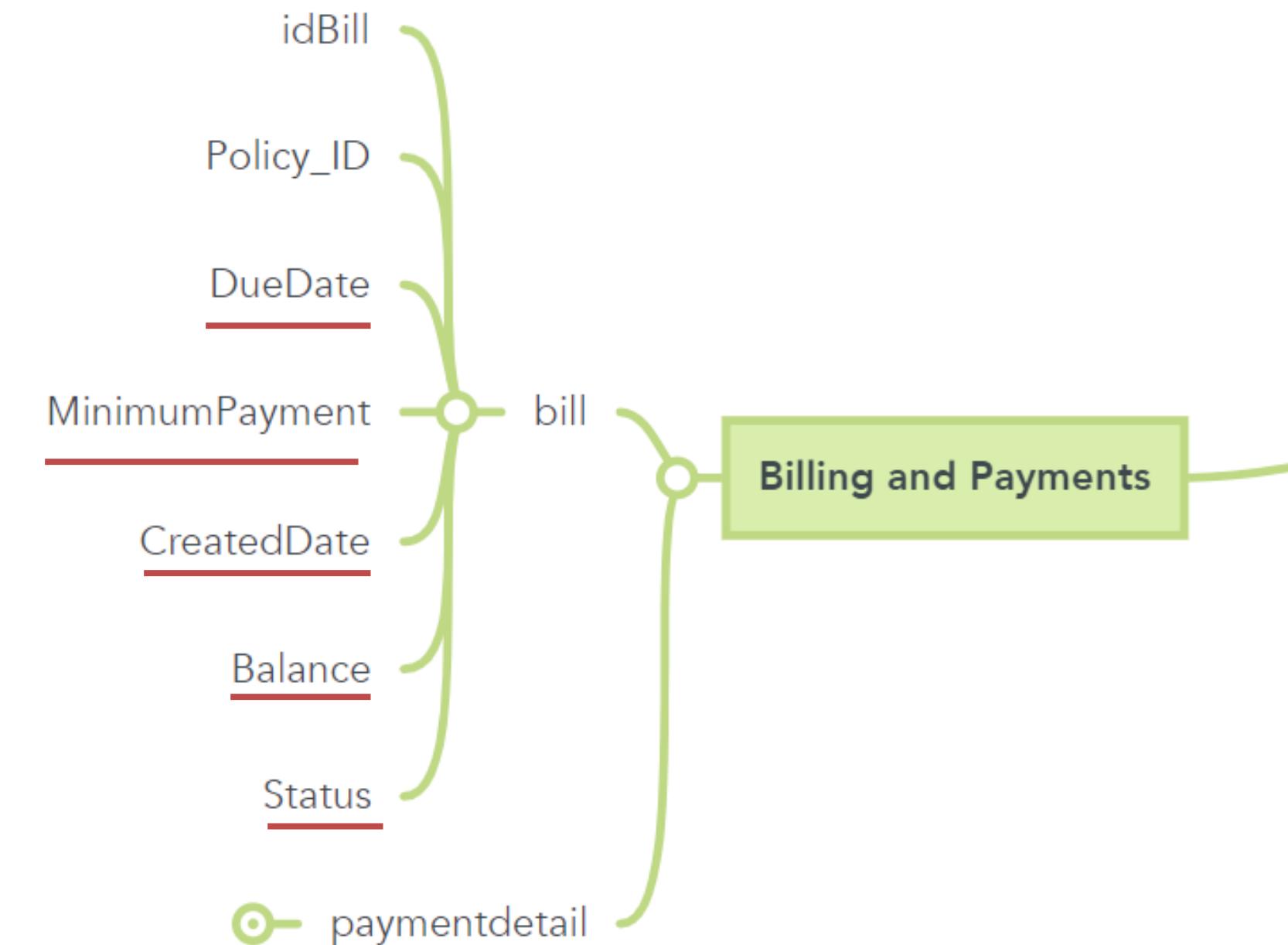
QUY MÔ DỮ LIỆU

DATA TAXONOMY

Bảng bill: lưu thông tin về hóa đơn cho chính sách

Một số thuộc tính quan trọng:

- DueDate: ngày hạn thanh toán
- MinimumPayment: số tiền thanh tối thiểu cho mỗi hóa đơn
- CreatedDate: ngày tạo hóa đơn
- Balance: số tiền còn lại mà khách hàng chưa thanh toán
- Status: trạng thái hóa đơn



1

KHẢO SÁT

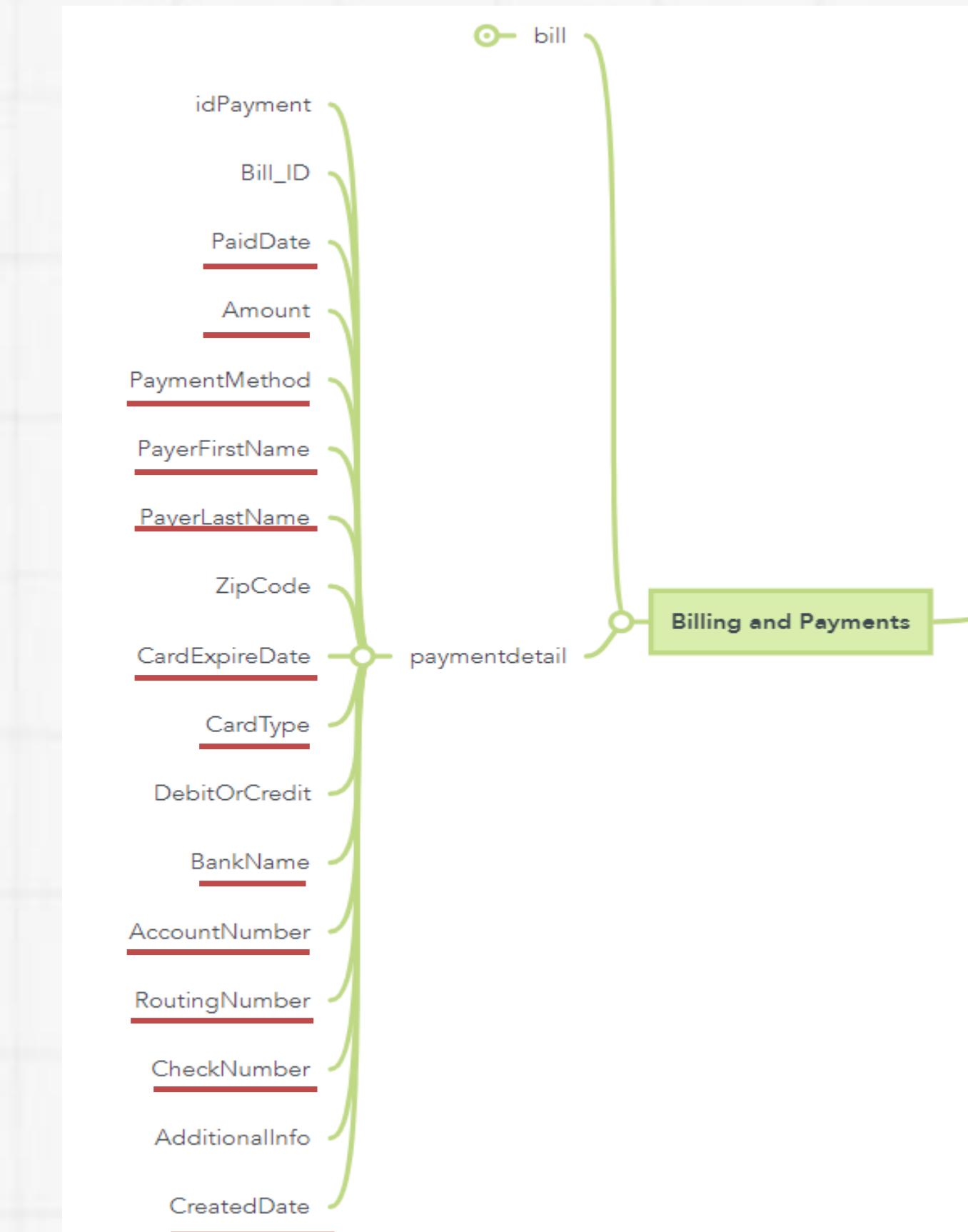
QUY MÔ DỮ LIỆU

DATA FLOW

Bảng paymentdetail: lưu thông tin về việc thực hiện giao dịch thanh toán của khách hàng

Một số thuộc tính quan trọng:

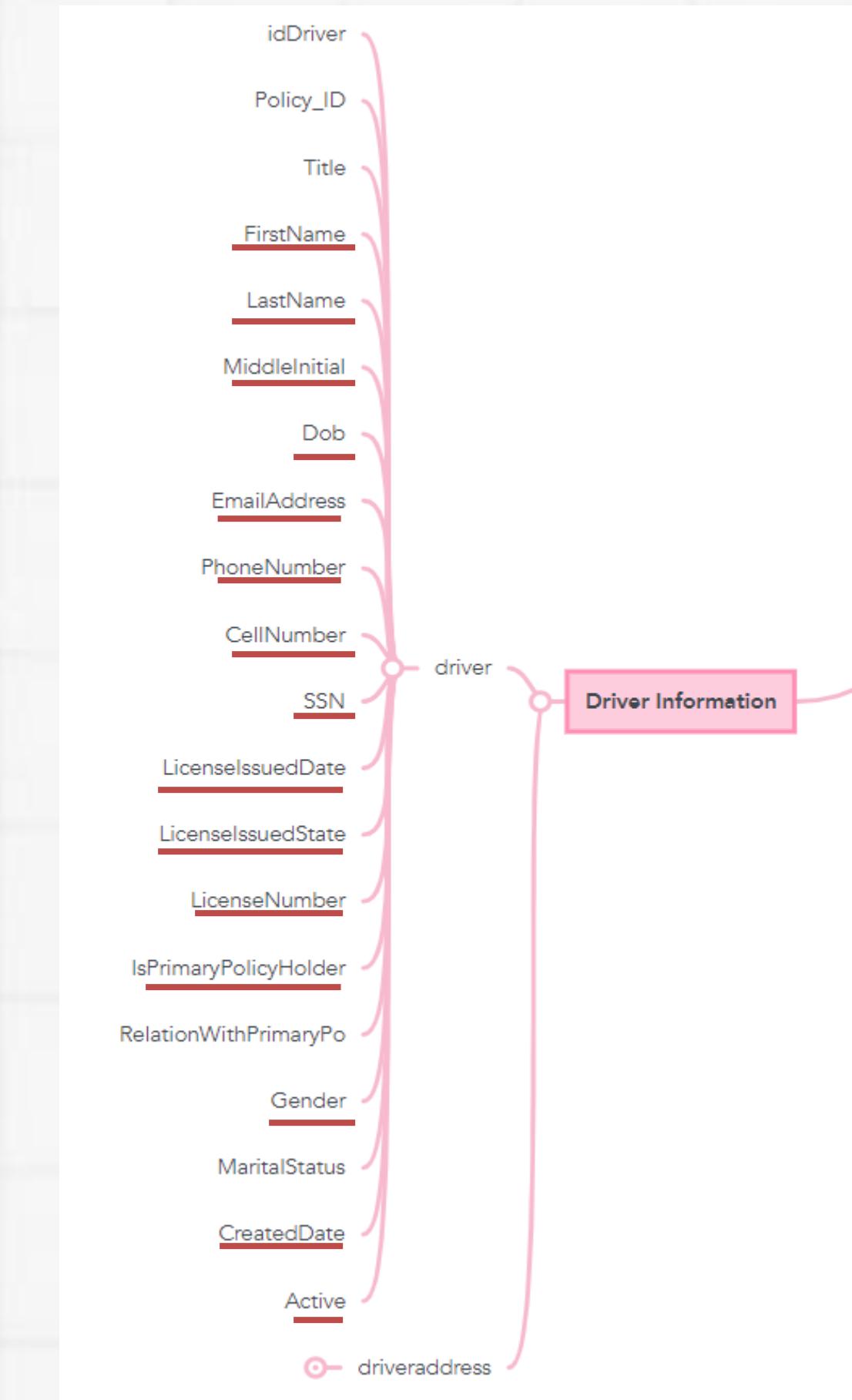
- PaidDate: ngày thực hiện thanh toán
- Amount: số tiền đã thanh toán
- PaymentMethod: phương thức thanh toán
- PayerFirstName, PayerLastName: tên người thanh toán
- CardExpireDate: ngày hết hạn thẻ
- CardType: loại thẻ
- BankName: tên ngân hàng
- AccountNumber: số tài khoản thẻ của người thanh toán
- RoutingNumber: mã định tuyến của ngân hàng
- CheckNumber: người dùng có thể gửi bản cứng của séc để thanh toán
- CreatedDate: ngày tạo hóa đơn



Bảng driver: lưu thông tin về người lái xe

Một số thuộc tính quan trọng:

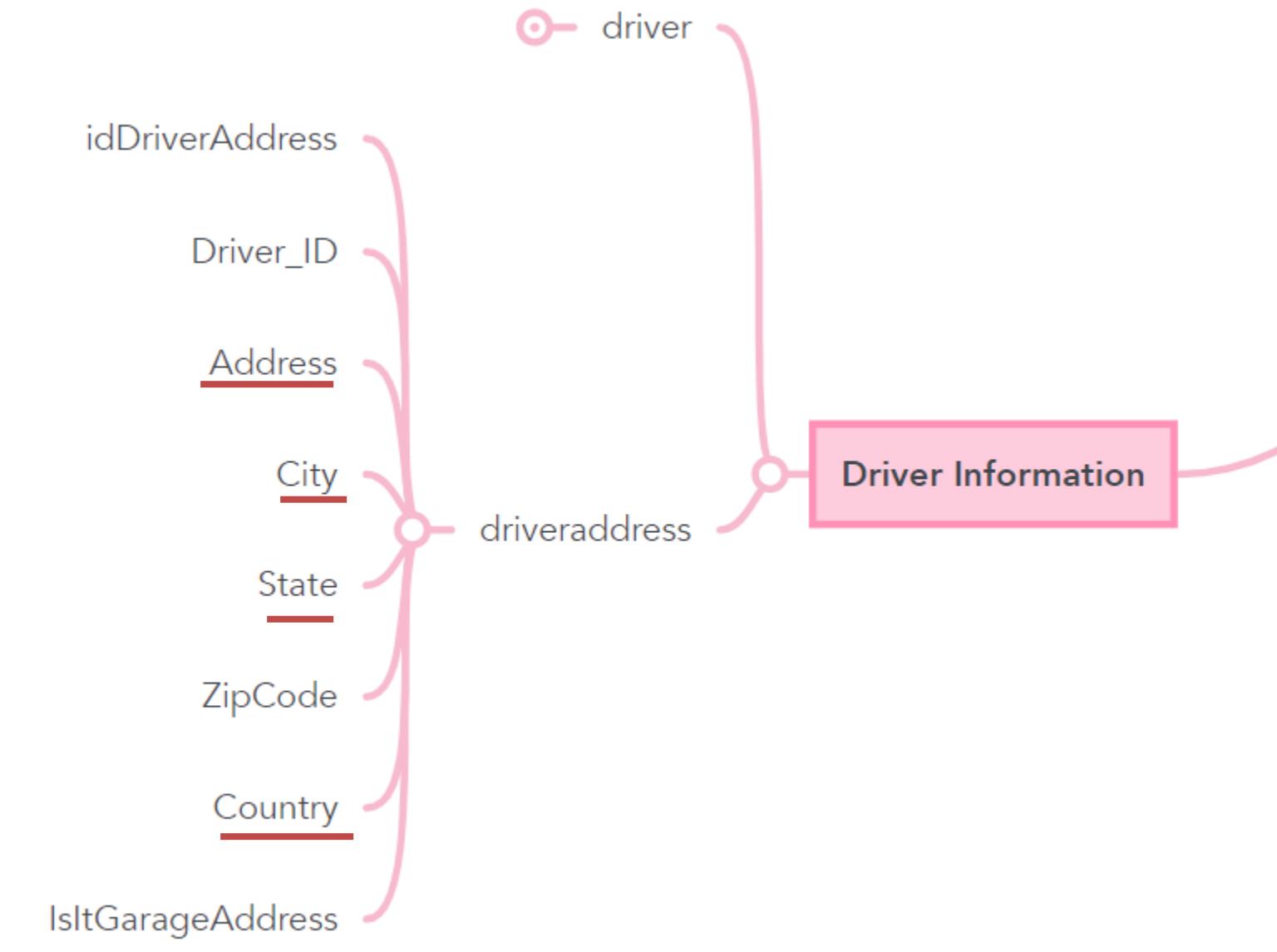
- FirstName, LastName, MiddleInitial: tên người lái xe
- Dob: ngày sinh của người lái xe
- EmailAddress: địa chỉ email của người lái xe
- PhoneNumber: số điện thoại nhà người lái xe
- CellNumber: số điện thoại người lái xe
- SSN: số an sinh xã hội
- LicenseIssuedDate: ngày cấp giấy phép lái xe
- LicenseIssuedState: tiểu bang cấp giấy phép
- LicenseNumber: số giấy phép
- IsPrimaryPolicyHolder: người lái xe là chủ hợp đồng chính (1), người lái xe không là chủ hợp đồng chính (0)
- Gender: giới tính
- CreatedDate: ngày thêm tài xế
- Active: 0 là trình điều khiển bị xóa khỏi chính sách, 1 là trình điều khiển chính



Bảng driveraddress: chứa địa chỉ nhà để xe và gửi thư của người lái xe

Một số thuộc tính quan trọng:

- Address: địa chỉ cụ thể
- City: thành phố
- State: tiểu bang
- Country: đất nước
- IsItGarageAddress: 1 là địa chỉ để xe, 0 là không phải địa chỉ để xe



1

KHẢO SÁT

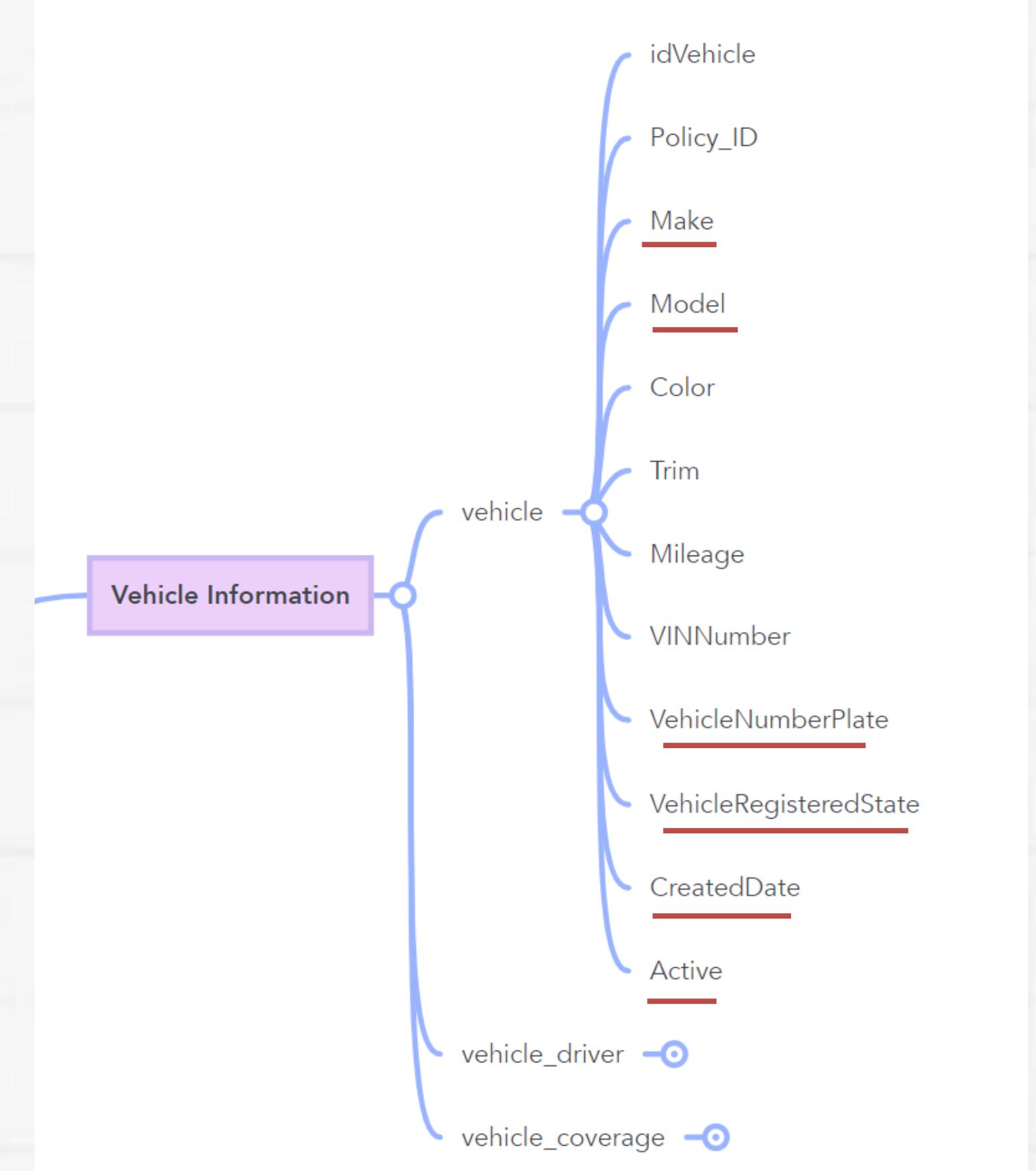
QUY MÔ DỮ LIỆU

DATA TAXONOMY

Bảng vehicle: lưu thông tin xe được bảo hiểm

Một số thuộc tính quan trọng:

- Make: hãng sản xuất xe
- Model: loại xe
- VehicleNumberPlate: biển số xe
- VehicleRegisteredState: tiểu bang đã đăng ký xe
- CreatedDate: ngày tạo
- Active: 1 là xe đã được loại bỏ khỏi chính sách, 0 là xe vẫn ở chính sách



1

KHẢO SÁT

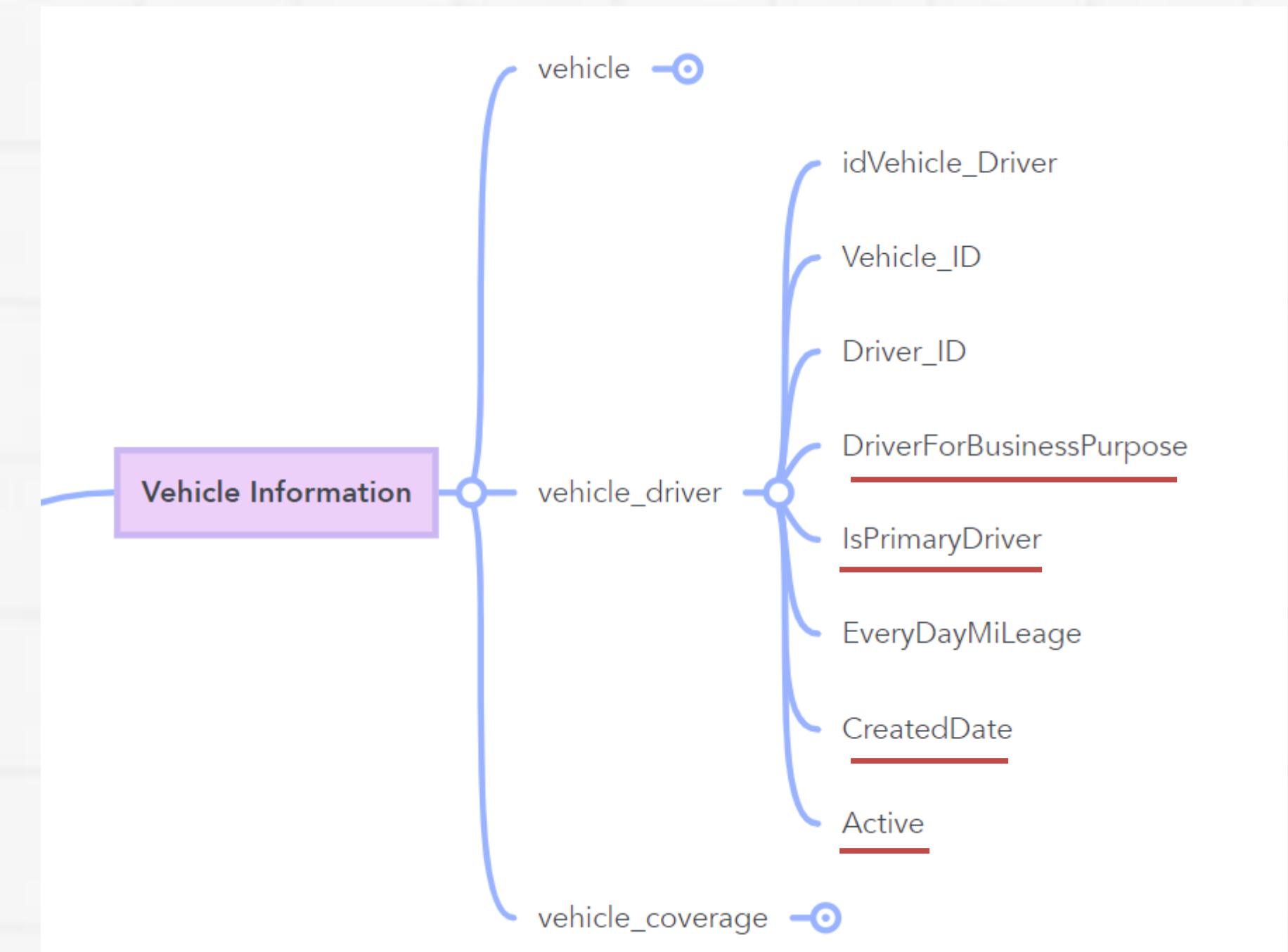
QUY MÔ DỮ LIỆU

DATA TAXONOMY

Bảng vehicle_driver: chứa thông tin về
mỗi quan hệ giữa xe và tài xế của chúng

Một số thuộc tính quan trọng:

- DriverForBusinessPurpose: 1 là tài xế sử dụng xe cho mục đích kinh doanh, 0 là ngược lại
- IsPrimaryDriver: 1 là chủ xe chính, 0 là ngược lại
- CreatedDate: ngày tạo
- Active: 1 – nếu loại bỏ người lái xe khỏi danh sách xe, 0 – ngược lại

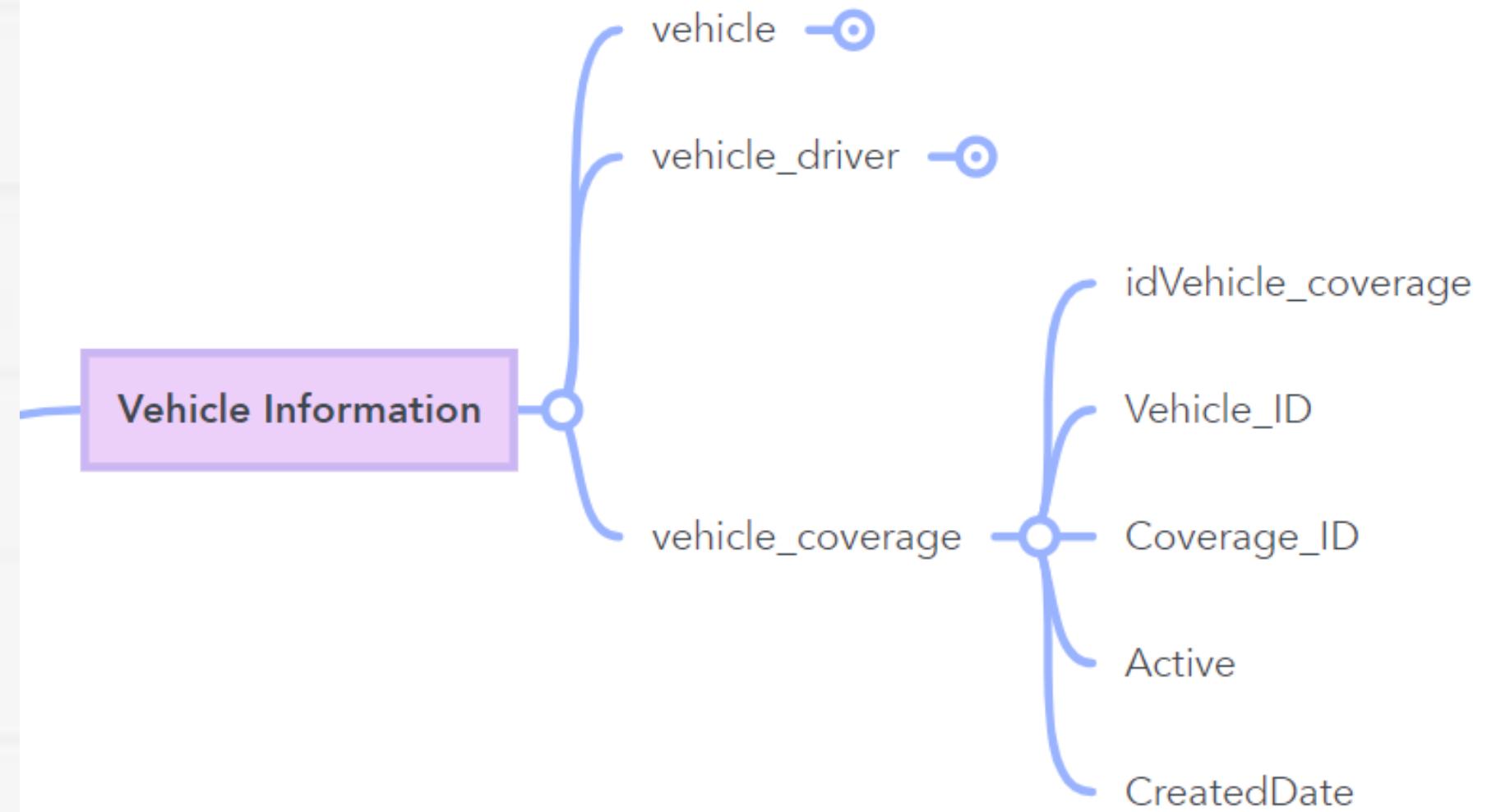


1

KHẢO SÁT

QUY MÔ DỮ LIỆU

DATA TAXONOMY

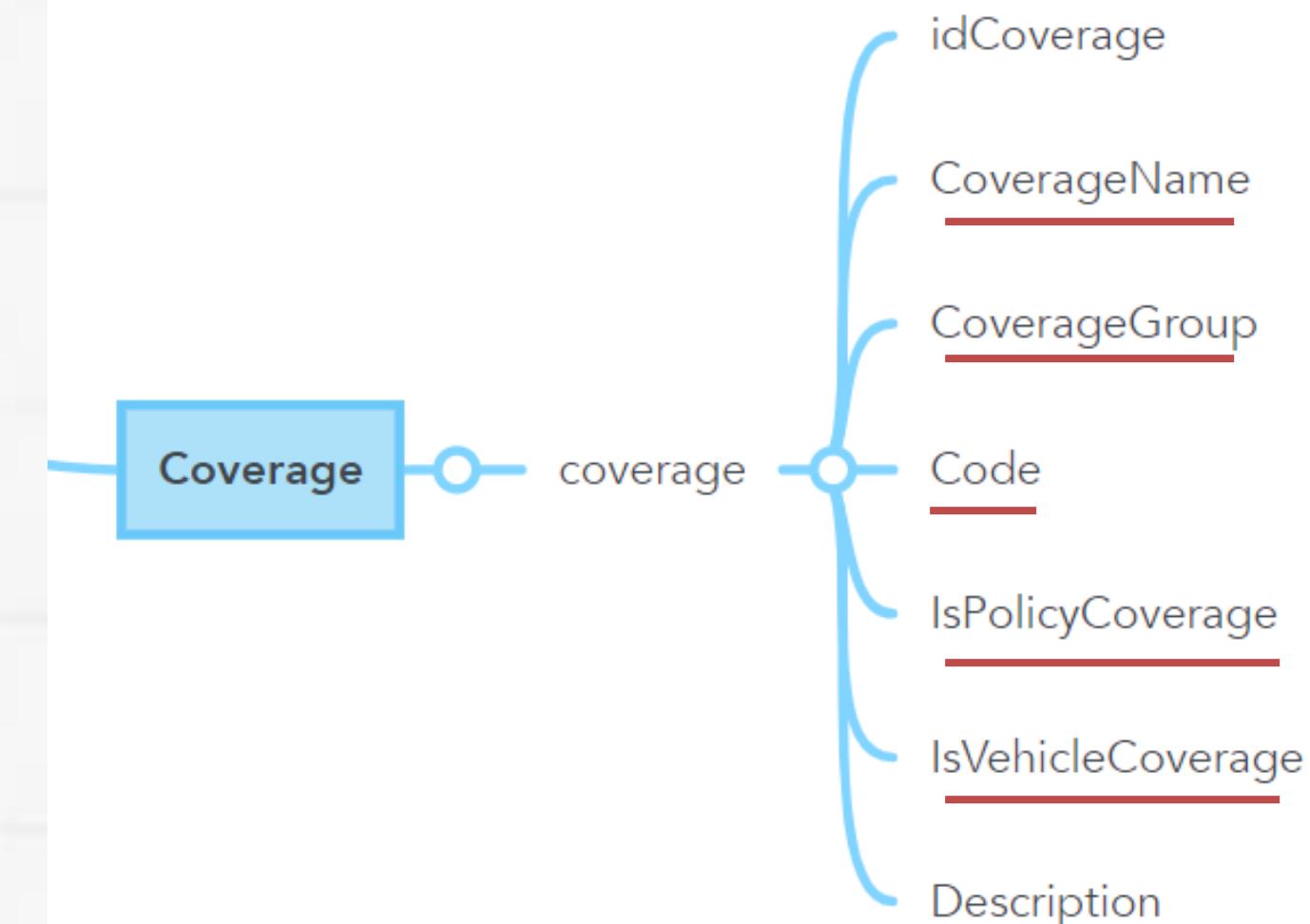


Bảng vehicle_coverage: chứa thông tin về mối quan hệ giữa xe và phạm vi bảo hiểm

Bảng coverage: lưu thông tin về các loại phạm vi bảo hiểm khác nhau

Một số thuộc tính quan trọng:

- CoverageName: tên phạm vi bảo hiểm
- CoverageGroup: loại phạm vi bảo hiểm
- Code: mã tên phạm vi bảo hiểm
- IsPolicyCoverage: 1 – bảo hiểm áp dụng cho chính sách và 0 – ngược lại
- IsVehicleCoverage: 1 – bảo hiểm áp dụng cho phương tiện và 0 – ngược lại



1

KHẢO SÁT

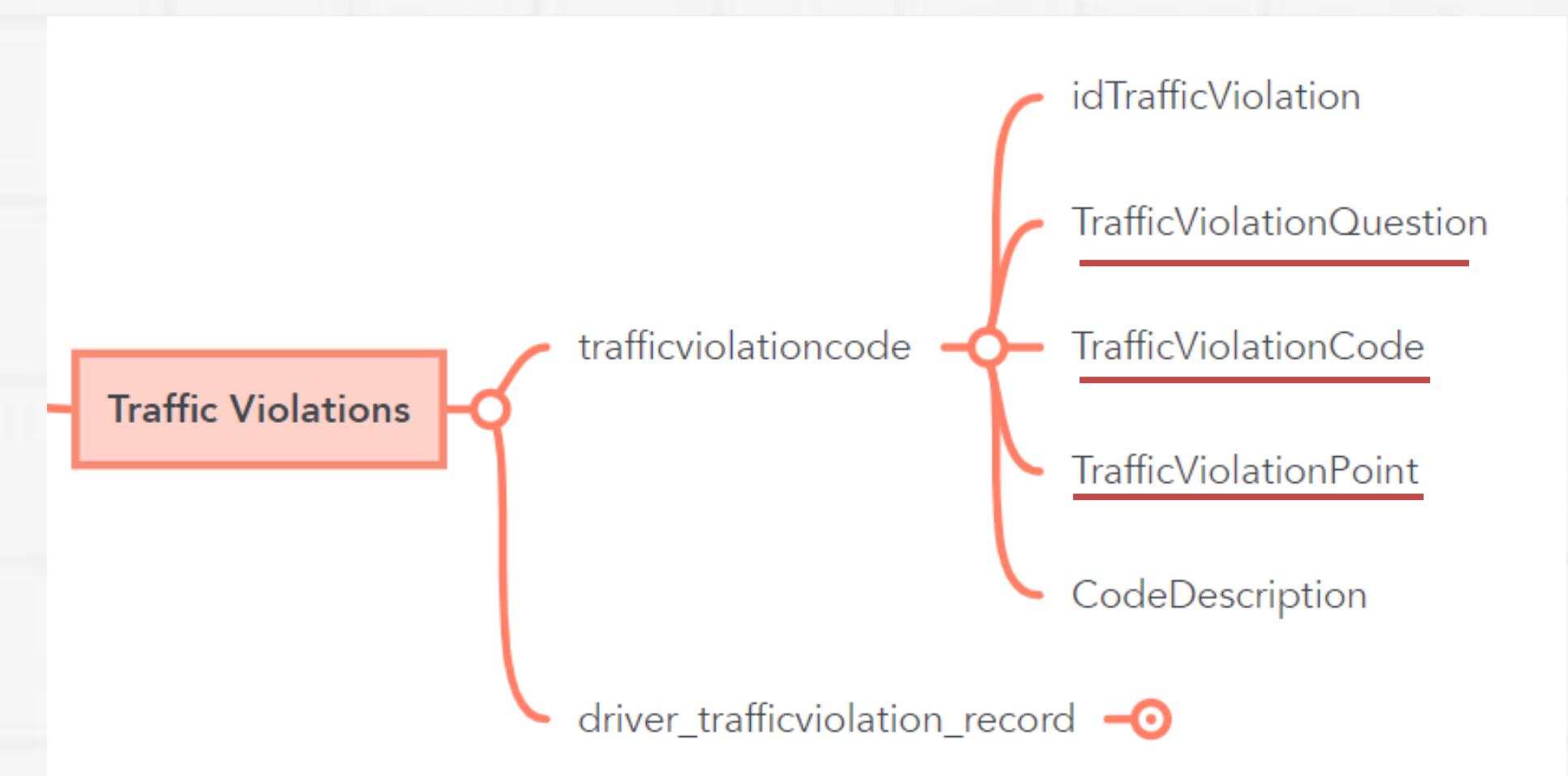
QUY MÔ DỮ LIỆU

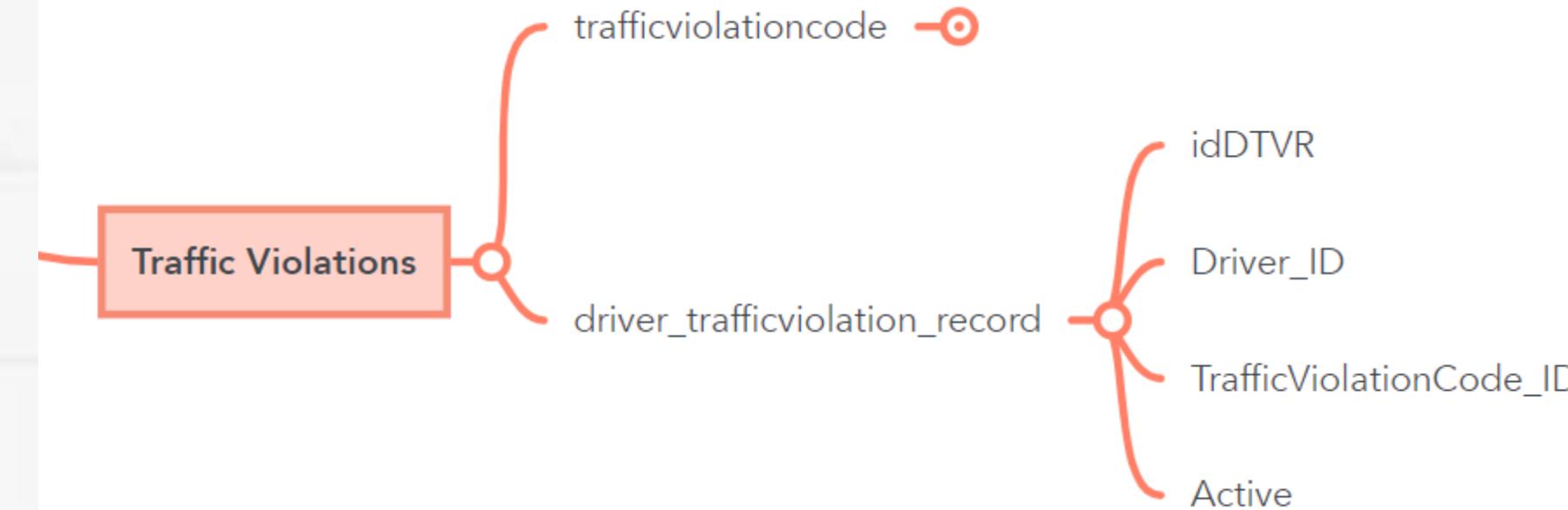
DATA TAXONOMY

Bảng trafficviolationcode: lưu thông tin về các mã vi phạm giao thông và mô tả liên quan đến vi phạm này

Một số thuộc tính quan trọng:

- TrafficViolationQuestion: câu hỏi về vi phạm giao thông
- TrafficViolationCode: mã vi phạm giao thông
- TrafficViolationPoint: điểm đánh giá mức độ nghiêm trọng cho mỗi vi phạm giao thông





Bảng driver_trafficViolation_record: chứa thông tin về các vi phạm giao thông của tài xế

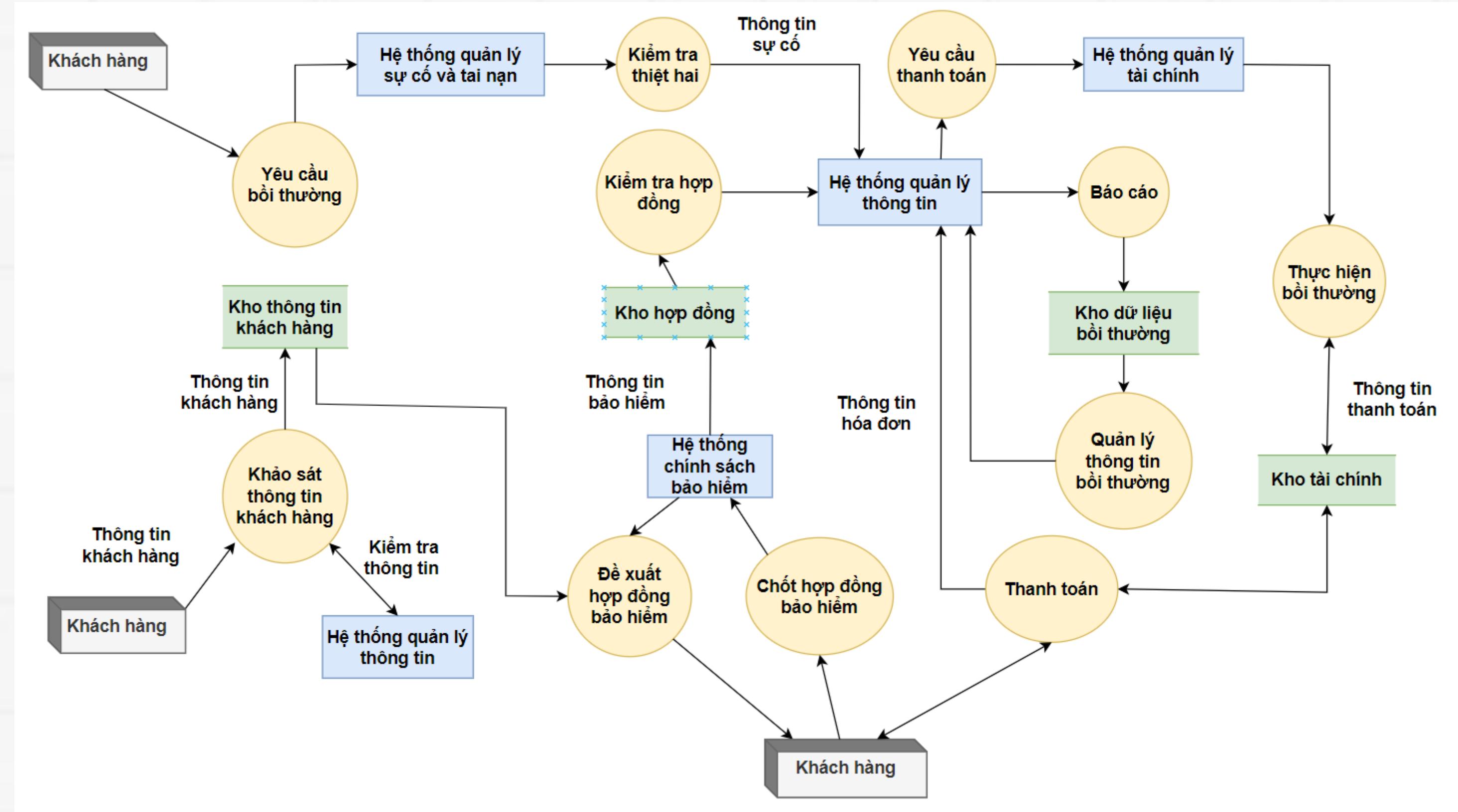


1

KHẢO SÁT

QUY MÔ DỮ LIỆU

DATA FLOW



1. KHẢO SÁT



- Tổng quan
- Quy trình nghiệp vụ
- Yêu cầu phân tích
- Quy mô dữ liệu
- ERD hệ thống OLTP
- Hệ thống chỉ số - cây phân tích dashboard

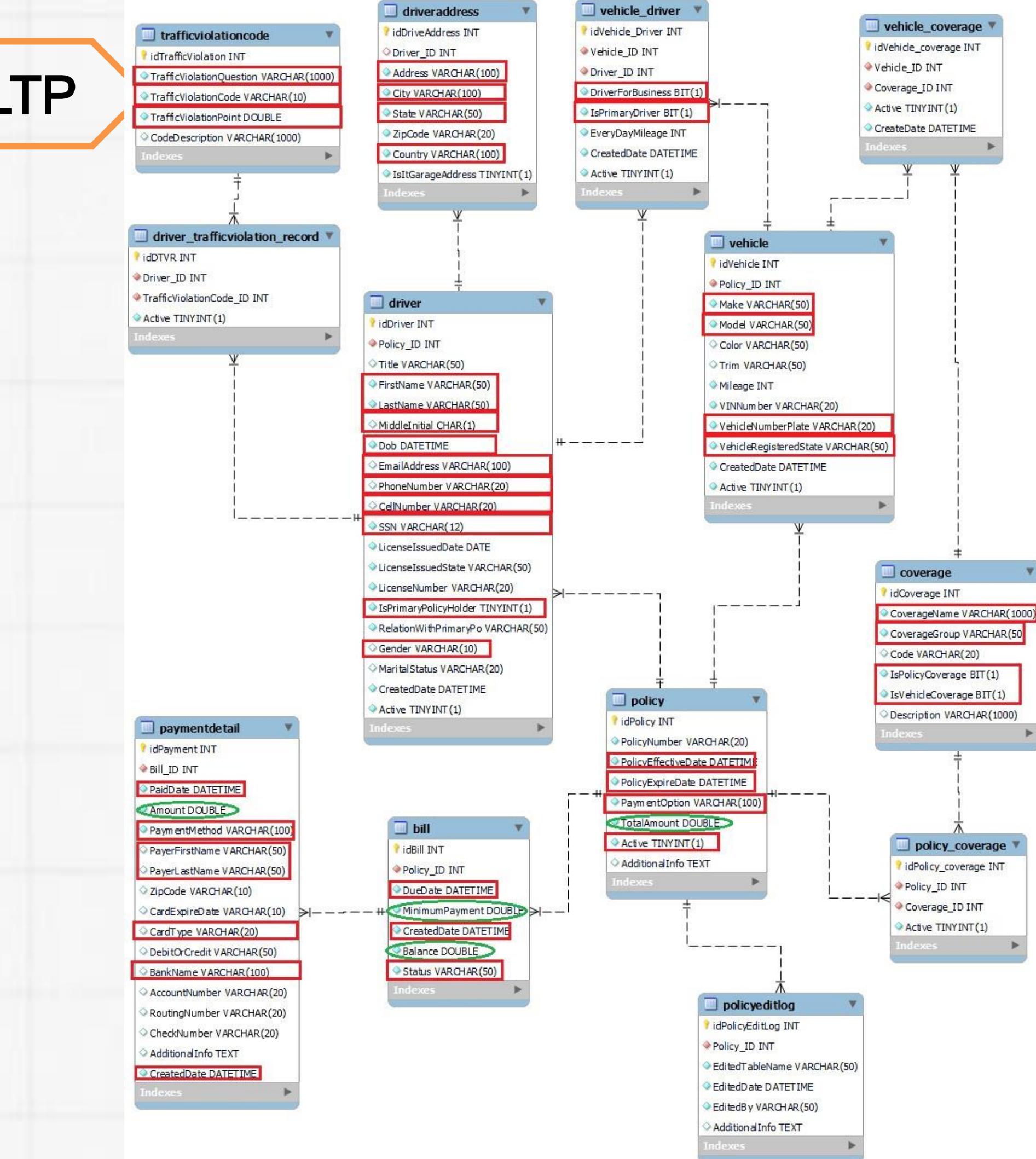
1

KHẢO SÁT

ERD hệ thống OLTP

dimension

fact



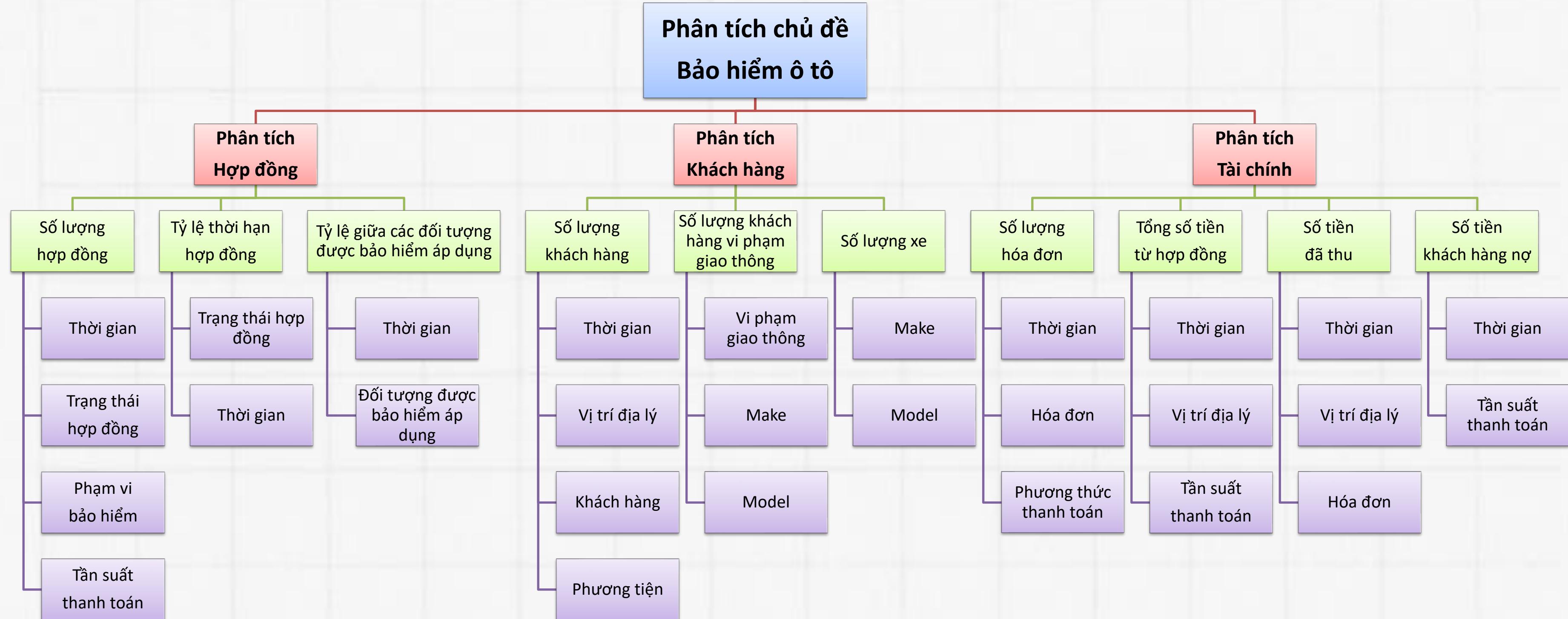
1. KHẢO SÁT



- Tổng quan
- Quy trình nghiệp vụ
- Yêu cầu phân tích
- Quy mô dữ liệu
- ERD hệ thống OLTP
- Hệ thống chỉ số - cây phân tích
dashboard

1 KHẢO SÁT

HỆ THỐNG CHỈ SỐ - CÂY PHÂN TÍCH DASHBOARD

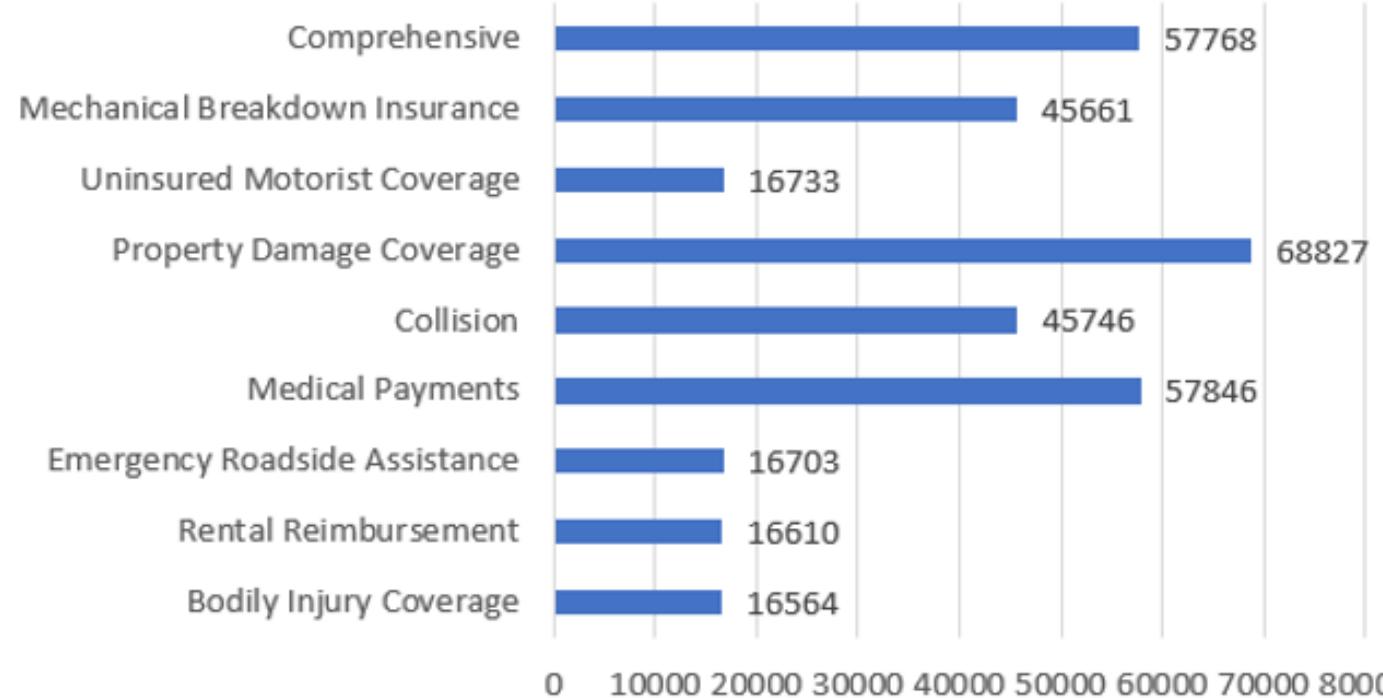


2. PHÂN TÍCH VÀ THIẾT KẾ

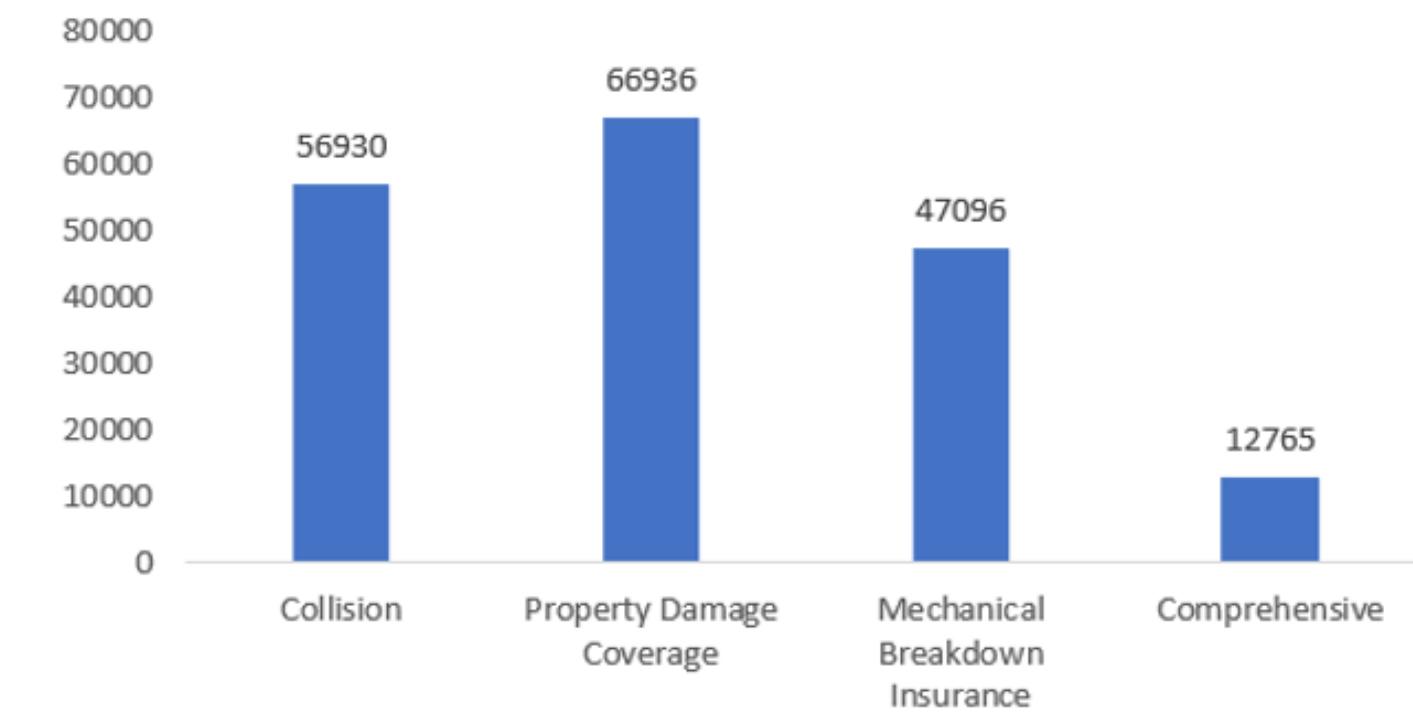
- Khám phá dữ liệu
- Kiến trúc data warehouse
- Nội dung ETL
- Hệ thống dimension
- Data model

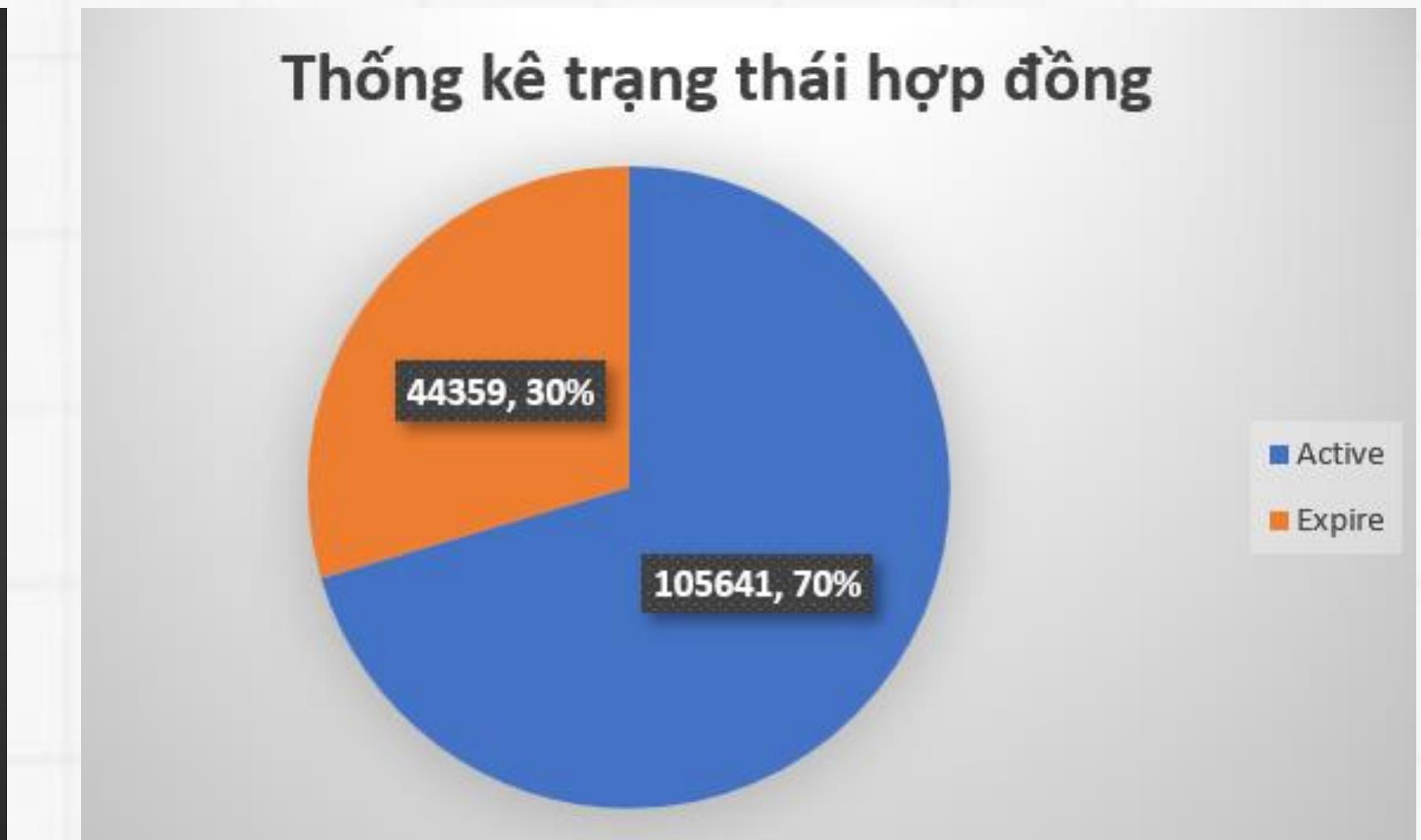


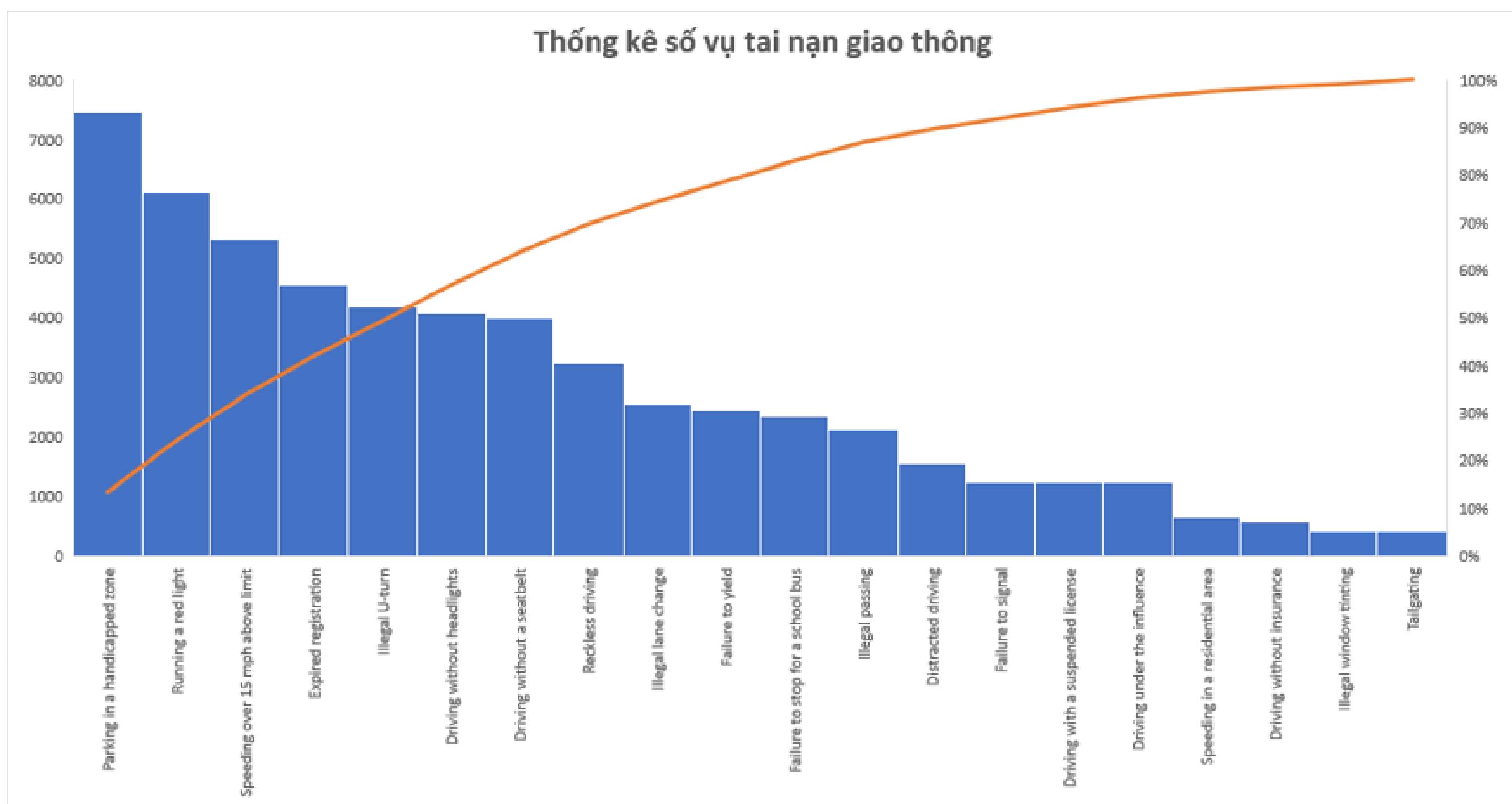
Thống kê hợp đồng theo phạm vi bảo hiểm

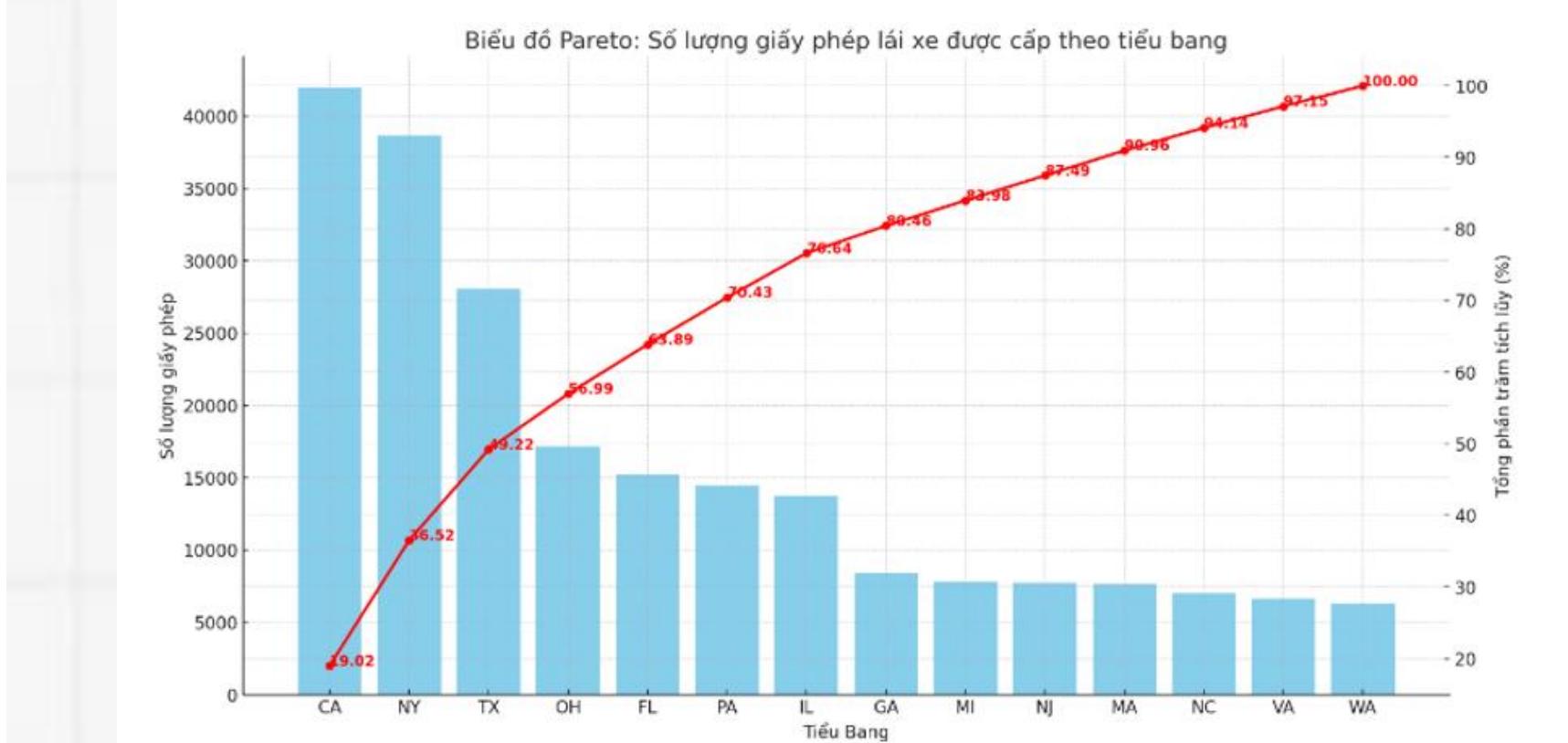
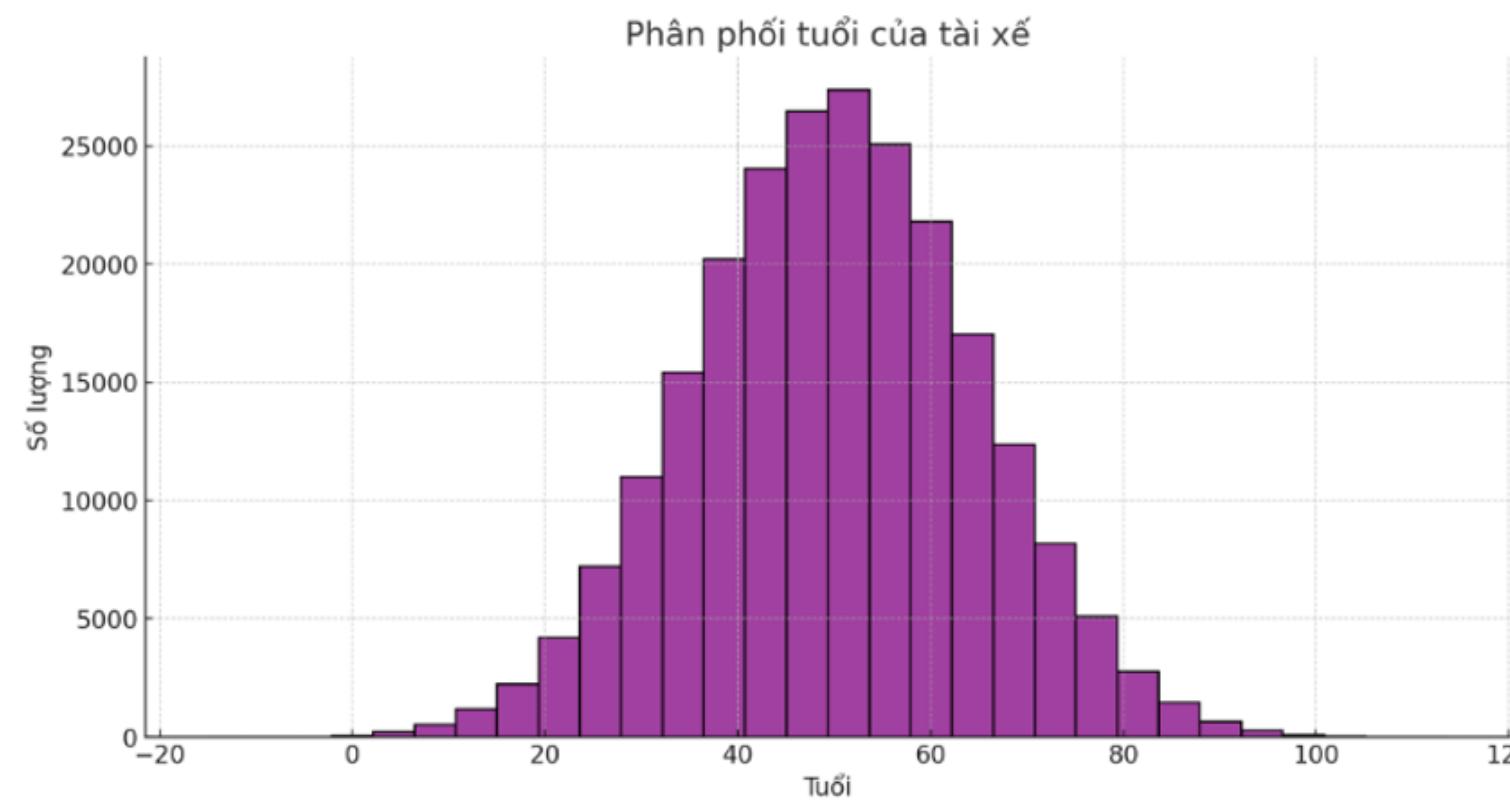


Thống kê số lượng xe theo phạm vi bảo hiểm





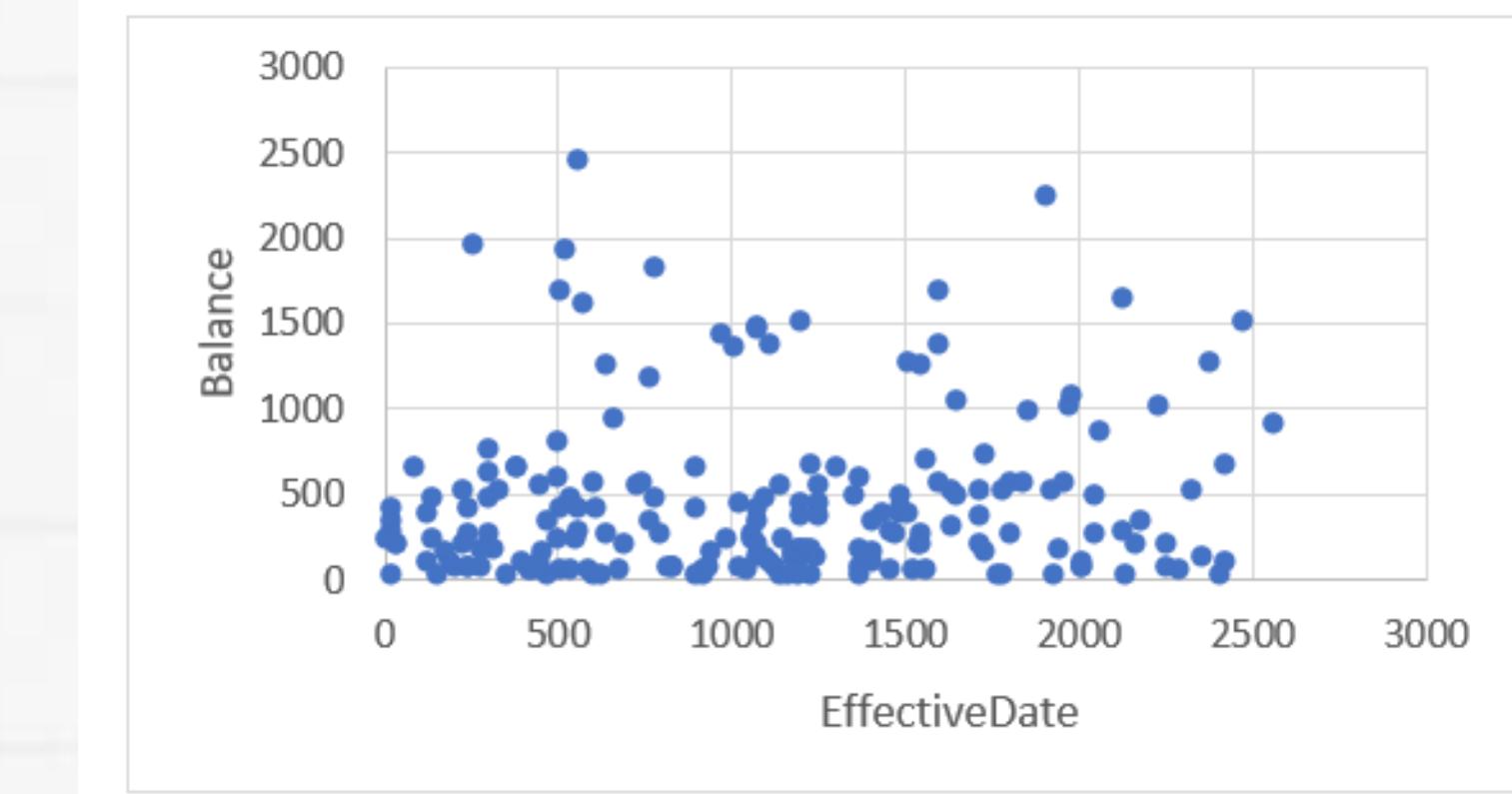
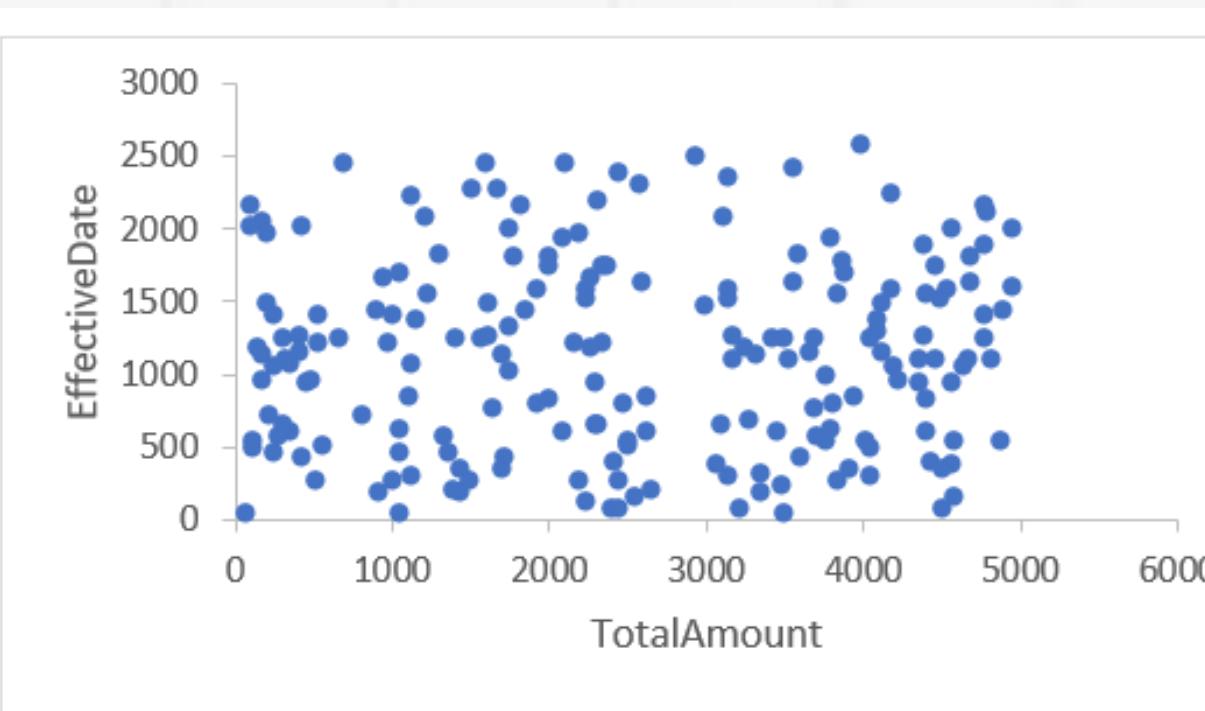
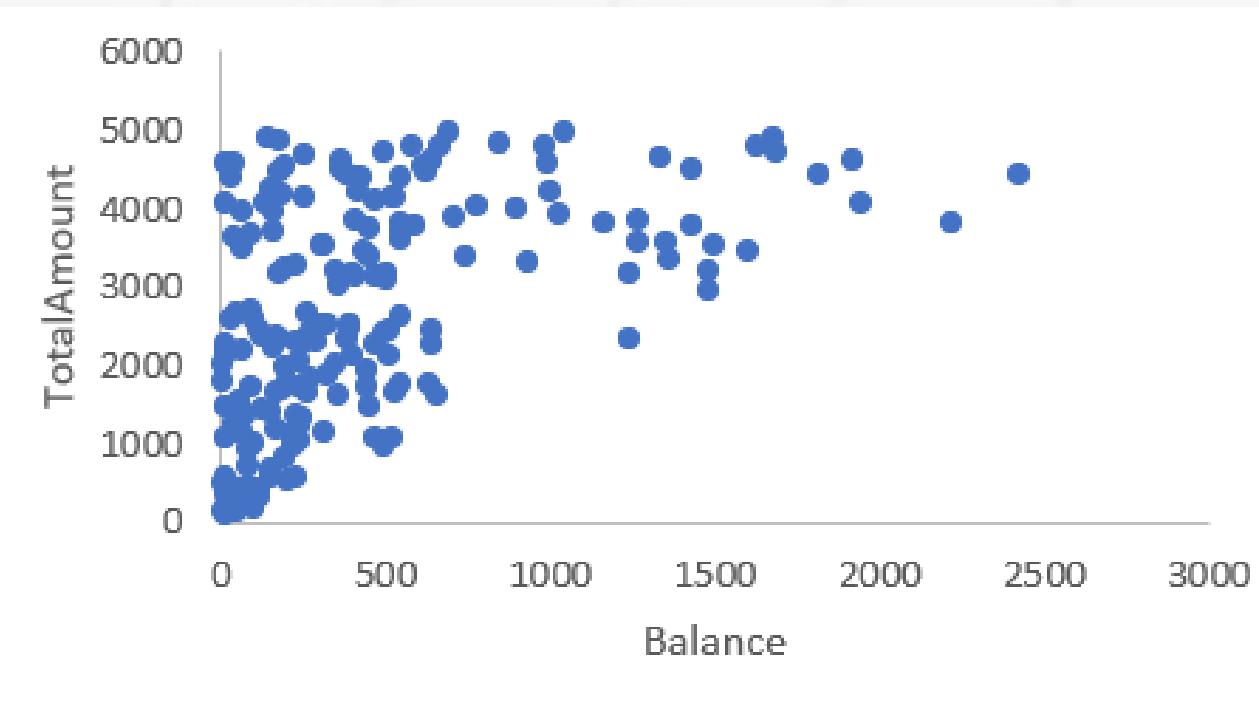




DATA ANALYSIS TOOL

<i>TotalAmount</i>	<i>Balance</i>	<i>EffectiveDate</i>
Mean	2540.1702	Mean
Standard Error	105.4968366	Standard Error
Median	2438.8	Median
Mode	2456.98	Mode
Standard Deviation	1491.950571	Standard Deviation
Sample Variance	2225916.505	Sample Variance
Kurtosis	-1.276452215	Kurtosis
Skewness	-0.052693523	Skewness
Range	4862.26	Range
Minimum	100.85	Minimum
Maximum	4963.11	Maximum
Sum	508034.04	Sum
Count	5046	Count
Confidence Level(95.0%)	208.0351773	Confidence Level(95.0%)
	66.57294981	Confidence Level(95.0%)
		91.68868741

SCATTER DIAGRAM



2. PHÂN TÍCH VÀ THIẾT KẾ

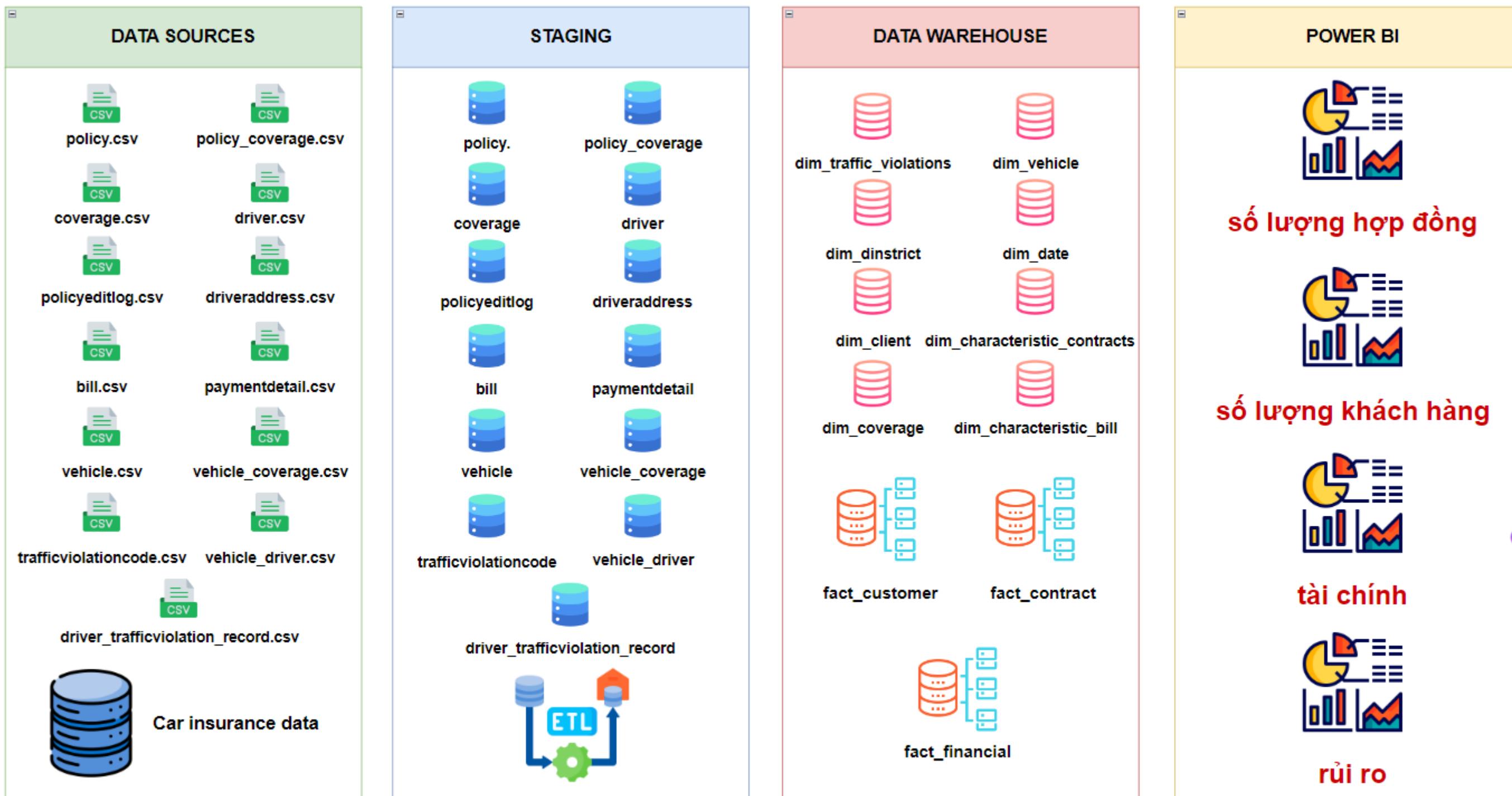
- Khám phá dữ liệu
- **Kiến trúc data warehouse**
- Nội dung ETL
- Hệ thống dimension
- Data model



2

PHÂN TÍCH VÀ THIẾT KẾ

KIẾN TRÚC DATA WAREHOUSE



2. PHÂN TÍCH VÀ THIẾT KẾ

- Khám phá dữ liệu
- Kiến trúc data warehouse
- Nội dung ETL
- Hệ thống dimension
- Data model



Các hoạt động ETL (Data sources → Staging):

Nội dung	Công cụ sử dụng
Loại bỏ các cột thừa	Python + MySQL
Xử lý các giá trị null	MySQL
Thêm cột	Python + MySQL
Định dạng lại dữ liệu	Python
Phân vùng dữ liệu	Python
Thay đổi nội dung dữ liệu	Python

Data sources → Staging



Before

	idPolicy	PolicyNumber	PolicyEffectiveDate	PolicyExpireDate	PaymentOption	TotalAmount	Active	AdditionalInfo
▶	1	PGC122571755	2020-06-02 00:00:00	2024-03-05 00:00:00	Quarterly	1168.55	1	Future ask different trade increase including dir...
	2	PGC853601786	2017-03-22 00:00:00	2023-05-28 00:00:00	Annual	1699.02	1	Unit cause yard.
	3	PGC269831275	2020-03-19 00:00:00	2024-01-29 00:00:00	Annual	4900.92	1	Event usually fill over like radio.
	4	PGC942128509	2020-07-20 00:00:00	2023-12-09 00:00:00	Quarterly	3938.12	1	Ability behind pull modern.
	5	PGC553146963	2018-12-12 00:00:00	2020-12-02 00:00:00	Quarterly	3843.63	0	Through stage fast explain.
	6	PGC777150293	2020-05-15 00:00:00	2020-07-06 00:00:00	OneTime	4556.62	0	Try someone dog really network nor imagine lan...
	7	PGC252588325	2019-12-21 00:00:00	2021-03-21 00:00:00	Semi-Annual	3755.9	1	Both live store foreign yeah born star.

Phân thời gian hiệu lực của Policy thành 4 nhóm
Đổi kiểu dữ liệu

Thay đổi dữ liệu

Xóa cột

Data sources → Staging

Transform



```
def classify_duration(row):
    days = (row['PolicyExpireDate'] - row['PolicyEffectiveDate']).days
    if days < 365:
        return '<1 year'
    elif 365 <= days < 730:
        return '1-2 years'
    elif 730 <= days < 1095:
        return '2-3 years'
    else:
        return '>3 years'
```

Phân thời gian hiệu lực của Policy thành 4 nhóm
tạo thành cột ‘Duration Group’

- Xóa cột ‘AdditionalInfo’
- Thay đổi dữ liệu cột ‘Active’
- Chuyển đổi kiểu dữ liệu của PolicyEffectiveDate và PolicyExpireDate sang date

```
# Xóa cột 'AdditionalInfo'
if 'AdditionalInfo' in df.columns:
    del df['AdditionalInfo']

# Thay đổi dữ liệu cột Active
df['Active'] = df['Active'].map({0: 'Expire', 1: 'Active'})

# Chuyển đổi kiểu dữ liệu của PolicyEffectiveDate và PolicyExpireDate sang date
df['PolicyEffectiveDate'] = pd.to_datetime(df['PolicyEffectiveDate']).dt.date
df['PolicyExpireDate'] = pd.to_datetime(df['PolicyExpireDate']).dt.date

# Phân loại thời gian
df['Duration Group'] = df.apply(classify_duration, axis=1)
```

Data sources → Staging



After

First few rows of the DataFrame:

	<code>idPolicy</code>	<code>PolicyNumber</code>	<code>PolicyEffectiveDate</code>	<code>PolicyExpireDate</code>	<code>PaymentOption</code>	<code>TotalAmount</code>	<code>Active</code>	<code>Duration</code>	<code>Group</code>
0	1	PGC122571755	2020-06-02	2024-03-05	Quarterly	1168.55	Active	>3 years	
1	2	PGC853601786	2017-03-22	2023-05-28	Annual	1699.02	Active	>3 years	
2	3	PGC269831275	2020-03-19	2024-01-29	Annual	4900.92	Active	>3 years	
3	4	PGC942128509	2020-07-20	2023-12-09	Quarterly	3938.12	Active	>3 years	
4	5	PGC553146963	2018-12-12	2020-12-02	Quarterly	3843.63	Expire	1-2 years	

Data sources → Staging



Before

idCoverage	CoverageName	CoverageGroup	Code	IsPolicyCoverage	IsVehicleCoverage	Description
1	Bodily Injury Coverage	Required Auto Insurance Coverages	BI	1	0	Leaving behind sequelae and physical damage
2	Rental Reimbursement	Additional Car Insurance Coverages	NULL	1	0	Car rental customers dont want to rent anymore
3	Emergency Roadside Assistance	Additional Car Insurance Coverages	NULL	1	0	Customers encounter an emergency while parti...
4	Medical Payments	Additional Car Insurance Coverages	MedPay	1	0	Pay for examination and surgery costs
5	Collision	Additional Car Insurance Coverages	NULL	0	1	traffic collision occurs
6	Property Damage Coverage	Required Auto Insurance Coverages	PD	1	1	Compensation for vehicle repair costs
7	Uninsured Motorist Coverage	Required Auto Insurance Coverages	NULL	1	0	Collision with an uninsured driver
8	Mechanical Breakdown Insurance	Additional Car Insurance Coverages	NULL	0	1	The car is damaged due to collision or technical ...
9	Comprehensive	Additional Car Insurance Coverages	NULL	0	1	Comprehensive insurance for cars
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Cập nhật các dữ liệu null

Xóa cột

Data sources → Staging

Transform



```

1 • UPDATE coverage
2   SET Code = CASE
3     WHEN idCoverage = 2 THEN 'RR'
4     WHEN idCoverage = 3 THEN 'ER'
5     WHEN idCoverage = 5 THEN 'Col'
6     WHEN idCoverage = 7 THEN 'UM'
7     WHEN idCoverage = 8 THEN 'MB'
8     WHEN idCoverage = 9 THEN 'Comp'
9     ELSE Code
10 END
11 WHERE idCoverage IN (2, 3, 5, 7, 8, 9);
12
13 • ALTER TABLE coverage
14   MODIFY Code varchar(20) Not NULL;

```

Cập nhật các dữ liệu null của cột Code và
chỉnh sửa kiểu dữ liệu

```

ALTER TABLE coverage
ADD COLUMN SubjectCovered VARCHAR(50);

UPDATE coverage
SET SubjectCovered = CASE
  WHEN IsPolicyCoverage = 1 AND IsVehicleCoverage = 0 THEN 'PolicyCoverage'
  WHEN IsPolicyCoverage = 0 AND IsVehicleCoverage = 1 THEN 'VehicleCoverage'
  WHEN IsPolicyCoverage = 1 AND IsVehicleCoverage = 1 THEN 'Both'
  ELSE NULL
END;

```

Tạo cột SubjectCovered và điều kiện

- ALTER TABLE coverage

DROP COLUMN IsPolicyCoverage,

DROP COLUMN IsVehicleCoverage;

Xóa 2 cột IsPolicyCoverage và IsVehicleCoverage

Data sources → Staging



coverage

After

	idCoverage	CoverageName	CoverageGroup	Code	SubjectCovered	Description
▶	1	Bodily Injury Coverage	Required Auto Insurance Coverages	BI	PolicyCoverage	Leaving behind sequelae and physical damage
	2	Rental Reimbursement	Additional Car Insurance Coverages	RR	PolicyCoverage	Car rental customers dont want to rent anymore
	3	Emergency Roadside Assistance	Additional Car Insurance Coverages	ER	PolicyCoverage	Customers encounter an emergency while parti...
	4	Medical Payments	Additional Car Insurance Coverages	MedPay	PolicyCoverage	Pay for examination and surgery costs
	5	Collision	Additional Car Insurance Coverages	Col	VehideCoverage	traffic collision occurs
	6	Property Damage Coverage	Required Auto Insurance Coverages	PD	Both	Compensation for vehicle repair costs
	7	Uninsured Motorist Coverage	Required Auto Insurance Coverages	UM	PolicyCoverage	Collision with an uninsured driver
	8	Mechanical Breakdown Insurance	Additional Car Insurance Coverages	MB	VehideCoverage	The car is damaged due to collision or technical ...
*	9	Comprehensive	Additional Car Insurance Coverages	Comp	VehideCoverage	Comprehensive insurance for cars
*	NULL	NULL	NULL	NULL	NULL	NULL

Data sources → Staging



bill

Before

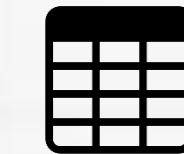
	idBill	Policy_ID	DueDate	MinimumPayment	CreatedDate	Balance	Status
	1	1	2024-05-31 00:00:00	146.06875	2020-05-31 00:00:00	570.1256548060571	Paid
	2	1	2024-05-31 00:00:00	146.06875	2020-08-29 00:00:00	168.67851336785517	Unpaid
	3	2	2021-03-20 00:00:00	212.3775	2017-03-20 00:00:00	426.1133809683317	Paid
	4	2	2021-03-20 00:00:00	212.3775	2018-03-15 00:00:00	191.14014661909226	Unpaid
	5	3	2024-03-17 00:00:00	612.615	2020-03-17 00:00:00	3521.7301935050837	Paid
	6	3	2024-03-17 00:00:00	612.615	2021-03-12 00:00:00	1475.2425886873039	Paid
	7	3	2024-03-17 00:00:00	612.615	2022-03-07 00:00:00	146.08546733362033	Unpaid
▶	8	4	2024-07-18 00:00:00	492.265	2020-10-16 00:00:00	1325.99122174772	Paid
	9	4	2024-07-18 00:00:00	492.265	2020-07-18 00:00:00	1707.876534839988	Paid
	10	4	2024-07-18 00:00:00	492.265	2021-01-14 00:00:00	702.4501224019529	Paid

Chuyển đổi kiểu dữ liệu

Định dạng dữ liệu

Data sources → Staging

Transform



bill

```
# Chuyển đổi kiểu dữ liệu của DueDate và CreatedDate sang date
df['DueDate'] = pd.to_datetime(df['DueDate']).dt.date
df['CreatedDate'] = pd.to_datetime(df['CreatedDate']).dt.date
```

Sửa Data Type của DueDate và CreatedDate từ datetime về date

Data sources → Staging



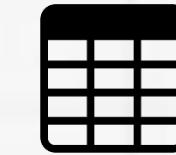
bill

After

	idBill	Policy_ID	DueDate	MinimumPayment	CreatedDate	Balance	Status
▶	1	1	2024-05-31	146.07	2020-05-31	570.13	Paid
	2	1	2024-05-31	146.07	2020-08-29	168.68	Unpaid
	3	2	2021-03-20	212.38	2017-03-20	426.11	Paid
	4	2	2021-03-20	212.38	2018-03-15	191.14	Unpaid
	5	3	2024-03-17	612.62	2020-03-17	3521.73	Paid
	6	3	2024-03-17	612.62	2021-03-12	1475.24	Paid
	7	3	2024-03-17	612.62	2022-03-07	146.09	Unpaid
	8	4	2024-07-18	492.27	2020-10-16	1325.99	Paid
	9	4	2024-07-18	492.27	2020-07-18	1707.88	Paid
	10	4	2024-07-18	492.27	2021-01-14	702.45	Paid

Data sources → Staging

Before



paymentdetail

	idPayment	Bill_ID	PaidDate	Amount	PaymentMethod	PayerFirstName	PayerLastName	ZipCode	CardExpireDate	CardType
▶	1	1	2020-06-04 00:00:00	598.4243451939428	Send a check to company	Timothy	Smith	36491	02/26	Master
	2	3	2017-03-25 00:00:00	1272.9066190316682	Credit	Brandi	Jenkins	93327	08/27	Master
	3	5	2020-03-19 00:00:00	1379.1898064949164	Online banking withdraw	Raymond	Garcia	52252	04/25	Visa
	4	6	2021-03-14 00:00:00	2046.4876048177798	Online banking withdraw	Raymond	Garcia	52252	04/25	Visa
	5	9	2020-07-23 00:00:00	2230.243465160012	Send a check to company	Jose	Hudson	68072	05/29	Discover
	6	8	2020-10-20 00:00:00	381.885313092268	Send a check to company	Jose	Hudson	68072	05/29	Discover
	7	10	2021-01-16 00:00:00	623.5410993457671	Send a check to company	Jose	Hudson	68072	05/29	Discover
	8	11	2018-12-14 00:00:00	2297.654815517587	Online banking withdraw	Amanda	Smith	44899	10/27	American Express
	9	12	2019-03-13 00:00:00	152.27658631598229	Online banking withdraw	Amanda	Smith	44899	10/27	American Express

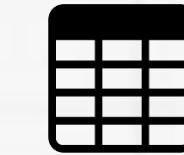
Xóa cột

DebitOrCredit	BankName	AccountNumber	RoutingNumber	CheckNumber	AdditionalInfo	CreatedDate
NULL	New York Community Bank	8407245558509	226071004	0002077869099	Try short sell instead road change. Few worker ...	2020-05-31 00:00:00
NULL	Arvest Bank	2447584809	082900872	3109483888258	Detail again water which hour top where. Docto...	2017-03-20 00:00:00
NULL	Bank OZK	8219550854126168	082907273	8217180557539	Region yeah run when even whatever. Still pro...	2020-03-17 00:00:00
NULL	Bank OZK	8219550854126168	082907273	8217180557539	Region yeah run when even whatever. Still pro...	2021-03-12 00:00:00
NULL	Bank OZK	820443130050	082907273	2734191537363	Those similar huge color necessary. Economy st...	2020-07-18 00:00:00
NULL	Bank OZK	820443130050	082907273	2734191537363	Those similar huge color necessary. Economy st...	2020-10-16 00:00:00
NULL	Bank OZK	820443130050	082907273	2734191537363	Those similar huge color necessary. Economy st...	2021-01-14 00:00:00
NULL	Arvest Bank	570158333	082900872	6075231853797	Clearly before off size culture. Win call direction...	2018-12-10 00:00:00
NULL	Arvest Bank	570158333	082900872	6075231853797	Clearly before off size culture. Win call direction...	2019-03-10 00:00:00
NULL	Arvest Bank	570158333	082900872	6075231853797	Clearly before off size culture. Win call direction...	2019-06-08 00:00:00

Định dạng
kiểu dữ liệu

Data sources → Staging

Transform



paymentdetail

```
# Xóa các cột không cần thiết  
columns_to_drop = ['ZipCode', 'CardExpireDate', 'DebitOrCredit', 'RoutingNumber', 'CheckNumber', 'AdditionalInfo']  
df.drop(columns=columns_to_drop, inplace=True)
```

Xóa các cột ZipCode, CardExpireDate, DebitOrCredit, RoutingNumber, CheckNumber, AdditionalInfo

```
# Chuyển đổi kiểu dữ liệu của cột Amount, CreatedDate và PaidDate  
df['CreatedDate'] = pd.to_datetime(df['CreatedDate']).dt.date  
df['PaidDate'] = pd.to_datetime(df['PaidDate']).dt.date  
df['Amount'] = df['Amount'].apply(Decimal)
```

Chỉnh sửa kiểu dữ liệu của PaidDate và CreatedDate thành Date, Amount thành Decimal

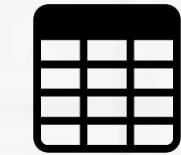
2

PHÂN TÍCH VÀ THIẾT KẾ

NỘI DUNG ETL

Data sources → Staging

After

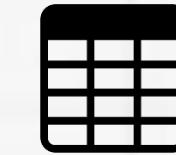


paymentdetail

idPayment	Bill_ID	PaidDate	Amount	PaymentMethod	PayerFirstName	PayerLastName	CardType	BankName	AccountNumber	CreatedDate
1	1	2020-06-04	598.42	Send a check to company	Timothy	Smith	Master	New York Community Bank	8407245558509	2020-05-31
2	3	2017-03-25	1272.91	Credit	Brandi	Jenkins	Master	Arvest Bank	2447584809	2017-03-20
3	5	2020-03-19	1379.19	Online banking withdraw	Raymond	Garcia	Visa	Bank OZK	8219550854126168	2020-03-17
4	6	2021-03-14	2046.49	Online banking withdraw	Raymond	Garcia	Visa	Bank OZK	8219550854126168	2021-03-12
5	9	2020-07-23	2230.24	Send a check to company	Jose	Hudson	Discover	Bank OZK	820443130050	2020-07-18
6	8	2020-10-20	381.89	Send a check to company	Jose	Hudson	Discover	Bank OZK	820443130050	2020-10-16
7	10	2021-01-16	623.54	Send a check to company	Jose	Hudson	Discover	Bank OZK	820443130050	2021-01-14
8	11	2018-12-14	2297.65	Online banking withdraw	Amanda	Smith	American Express	Arvest Bank	570158333	2018-12-10
9	12	2019-03-13	152.28	Online banking withdraw	Amanda	Smith	American Express	Arvest Bank	570158333	2019-03-10
10	13	2019-06-12	1056.58	Online banking withdraw	Amanda	Smith	American Express	Arvest Bank	570158333	2019-06-08
11	14	2020-05-15	4556.62	Send a check to company	Andrew	Reese	Discover	United Bank	17791399	2020-05-13
12	16	2019-12-21	951.70	Credit	Rodney	Johnson	American Express	United Bank	76065147219	2019-12-19

Column Name	Data Type	Null Count
idPayment	int	0
Bill_ID	int	0
PaidDate	date	0
Amount	decimal(10,2)	0
PaymentMethod	varchar(100)	0
PayerFirstName	varchar(50)	0
PayerLastName	varchar(50)	0
CardType	varchar(20)	0
BankName	varchar(100)	0
AccountNumber	varchar(20)	0
CreatedDate	date	0

Data sources → Staging



policy_coverage

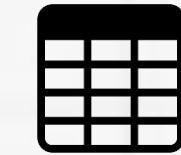
Before

	idPolicy_coverage	Policy_ID	Coverage_ID	Active
▶	1	1	6	1
	2	2	6	1
	3	2	5	1
	4	3	4	1
	5	3	6	1
	6	3	9	1
	7	4	3	1
	8	4	9	1
	9	5	4	0
	10	5	6	0

Column Name	Data Type	Null Count
idPolicy_coverage	int	0
Policy_ID	int	0
Coverage_ID	int	0
Active	tinyint(1)	0

Data sources → Staging

Transform



policy_coverage

```
# Chính sửa cột Active trong DataFrame
df['Active'] = df['Active'].map({1: 'Active', 0: 'Expire'})
```

After

	<u>idPolicy_coverage</u>	<u>Policy_ID</u>	<u>Coverage_ID</u>	<u>Active</u>
▶	1	1	6	Active
	2	2	6	Active
	3	2	5	Active
	4	3	4	Active
	5	3	6	Active
	6	3	9	Active
	7	4	3	Active
	8	4	9	Active
	9	5	4	Expire
	10	5	6	Expire

Column Name	Data Type	Null Count
<u>idPolicy_coverage</u>	int64	0
<u>Policy_ID</u>	int64	0
<u>Coverage_ID</u>	int64	0
<u>Active</u>	object	0

Data sources → Staging

Before



Driver

	idDriver	Policy_ID	Title	FirstName	LastName	MiddleInitial	Dob	EmailAddress	PhoneNumber	CellNumber	SSN	LicenseIssuedDate	LicenseIssuedState	LicenseNumber	IsPrimaryPolicyHolder	RelationWithPrimaryPo	Gender	MaritalStatus	Create
1	24596	Dr	Sonya	Hunt	M	M	1947-05-07 00:00:00	stephanieross@example.net	8381899499	8381899499	826-22-7181	1967-01-13	1967-03-13	530-79-7148	1	Parent	Male	Divorced	2024-01-01
2	99658	Mrs	Amber	Kim	C	C	1979-02-17 00:00:00	tami75@example.org	4069538623	4069538623	670-68-4116	1993-02-18	1993-04-13	355-23-9049	0	Self	Female	Single	2022-01-01
3	105094	Mrs	Sharon	Morris	M	M	1941-10-09 00:00:00	garnold@example.net	2347188243	2347188243	487-35-5077	1984-10-10	1984-12-09	918-21-4744	1	Other	Male	Single	2022-01-01
4	40809	Dr	Christopher	Holder	E	E	1998-11-24 00:00:00	christopher62@example.net	5169103834	5169103834	660-10-6134	2013-04-10	2013-06-10	318-48-9964	1	Other	Female	Married	2024-01-01
5	96604	Dr	Michelle	Rogers	D	D	1927-07-04 00:00:00	scott64@example.com	4778586115	4778586115	113-38-0966	2003-06-25	2003-09-13	460-68-4543	1	Other	Male	Divorced	2021-01-01
6	135868	Dr	Tammy	Navarro	K	K	1988-09-08 00:00:00	millererin@example.org	0018434739	0018434739	679-96-6737	2003-08-17	2003-10-05	232-49-2808	1	Spouse	Female	Married	2020-01-01
7	106788	Mr	Brian	Ramirez	A	A	1925-10-30 00:00:00	marcus17@example.net	1598470323	1598470323	179-04-5859	1939-12-13	1940-02-04	562-85-1843	1	Child	Female	Widowed	2021-01-01
8	54713	Mr	Steve	Foster	B	B	1953-08-20 00:00:00	ygay@example.net	3691357614	3691357614	101-84-2693	1980-09-30	1980-11-23	541-54-7982	0	Self	Male	Single	2024-01-01
9	17964	Dr	Alex	Davis	S	S	1975-07-31 00:00:00	hicksemily@example.com	1379865935	1379865935	128-05-8768	2001-11-03	2002-01-28	785-95-2242	1	Other	Male	Widowed	2023-01-01

Column Name	Data Type	Null Count
idDriver	int	0
Policy_ID	int	0
Title	varchar(50)	0
Dob	datetime	0
EmailAddress	varchar(100)	0
PhoneNumber	varchar(20)	0
SSN	varchar(12)	0
LicenseIssuedDate	date	0
LicenseIssuedState	varchar(50)	0
LicenseNumber	varchar(20)	0
IsPrimaryPolicyHolder	tinyint(1)	0
Gender	tinyint	78400

Data sources → Staging

Transform



Driver

```
# Chuyển kiểu dữ liệu của Gender về text
def convert_gender(gender, title):
    if gender == 1:
        return "Male"
    elif gender == 0:
        return "Female"
    else:
        if title == "Mr":
            return "Male"
        elif title == "Mrs":
            return "Female"
        return None

df['Gender'] = df.apply(lambda row: convert_gender(row['Gender'], row['Title']), axis=1)
```

```
# Chuyển đổi Dob sang kiểu Date
df['Dob'] = pd.to_datetime(df['Dob'])

# Tính tuổi
reference_date = datetime(2020, 12, 31)
df['Age'] = (reference_date - df['Dob']).dt.days // 365

# Phân loại vào nhóm tuổi
def classify_age_group(age):
    if age < 30:
        return '<30'
    elif 30 <= age <= 45:
        return '30-45'
    elif 46 <= age <= 60:
        return '46-60'
    else:
        return '>60'

df['AgeGroup'] = df['Age'].apply(classify_age_group)
```

```
# Xóa các cột không cần thiết
df.drop(columns=['CellNumber', 'RelationWithPrimaryPo', 'MaritalStatus', 'CreatedDate', 'Active'], inplace=True)
```

```
# Tạo cột Name từ FirstName, MiddleInitial và LastName
df['Name'] = df['FirstName'] + ' ' + df['MiddleInitial'].fillna('') + ' ' + df['LastName']
```

```
# Chuyển kiểu dữ liệu của IsPrimaryPolicyHolder về text
df['IsPrimaryPolicyHolder'] = df['IsPrimaryPolicyHolder'].map({1: "Primary", 0: "NonePrimary"})
```

```
# Xóa các cột FirstName, MiddleInitial, LastName
df.drop(columns=['FirstName', 'MiddleInitial', 'LastName'], inplace=True)
```

Data sources → Staging

After



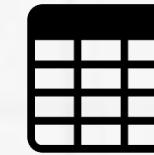
Driver

	idDriver	Policy_ID	Title	Dob	EmailAddress	...	LicenseNumber	IsPrimaryPolicyHolder	Gender	AgeGroup	Name
0	1	24596	Dr	1988-03-11	stephanieross@example.net	...	530-79-7148	NonePrimary	Female	30-45	Sonya M Hunt
1	2	99658	Mrs	1975-12-29	tami75@example.org	...	355-23-9049	NonePrimary	Female	30-45	Amber C Kim
2	3	105094	Mrs	1967-10-12	garnold@example.net	...	918-21-4744	Primary	Female	46-60	Sharon M Morris
3	4	40809	Dr	1989-01-11	christopher62@example.net	...	318-48-9964	Primary	Male	30-45	Christopher E Holder
4	5	96604	Dr	1955-04-26	scott64@example.com	...	460-68-4543	Primary	Male	>60	Michelle D Rogers
...
235384	235385	144862	Mrs	1983-05-02	christianrangel@example.org	...	843-80-8210	Primary	Female	30-45	Benjamin W Holt
235385	235386	62474	Mrs	1996-01-11	angelawoods@example.com	...	361-95-9012	Primary	Female	<30	Mark M Nguyen
235386	235387	123901	Mr	1976-08-16	horncaroline@example.org	...	637-78-5337	Primary	Male	30-45	Erin W Wise
235387	235388	61680	Dr	1964-04-10	bryan51@example.org	...	169-66-6041	Primary	Male	46-60	Donald H Rojas
235388	235389	4642	Dr	1965-06-20	zpalmer@example.com	...	951-94-9987	Primary	Male	46-60	Andres A Copeland

Column Name	Data Type	Null Count
idDriver	int	0
Policy_ID	int	0
Dob	date	0
EmailAddress	varchar(100)	0
PhoneNumber	varchar(20)	0
SSN	varchar(12)	0
LicenseIssuedDate	date	0
LicenseIssuedState	varchar(50)	0
LicenseNumber	varchar(20)	0
IsPrimaryPolicyHolder	varchar(20)	0
Gender	varchar(20)	0
AgeGroup	text	0
Name	varchar(255)	0

Data sources → Staging

Before



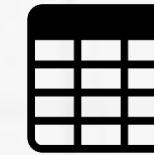
Vehicle

	idVehicle	Policy_ID	Make	Model	Color	Trim	Mileage	VINNumber	VehideNumberPlate	VehideRegisteredState	CreatedDate	Active
	150762	74707	Volkswagen	Tiguan	#99231d	R-Line	45730	VTSPYQV61FRFV56I	LYH-2445	CA	2019-02-05 00:00:00	0
	150761	89828	Hyundai	Elantra	#f9bdd0	Ultimate	145946	HE52MJUCIZVEWWH1R	QBN-9257	VI	2021-04-20 00:00:00	1
	150760	46693	Lexus	RX	#c45a09	Base	193474	LRW3HAE5ELWUYTK1M	YRH-5741	FL	2023-10-03 00:00:00	0
	150759	13982	Volkswagen	Passat	#f1bdf9	SEL	155301	VPHM2JCVES1LWYZHO	FET-8616	NC	2020-10-17 00:00:00	0
	150758	56814	Acura	NSX	#18a59e	Advance	92609	AN771TWMIRMZ9C85C	AYB-4744	DE	2020-01-25 00:00:00	0
	150757	2548	Toyota	Tacoma	#62e06d	LE	133703	TT03TZEZXPBCBNVIZ	SIH-4559	UT	2018-08-10 00:00:00	0
	150756	97827	Jeep	Compass	#f22186	NULL	51314	JCD17LCDZBHTV60G4	DOC-5340	MD	2018-10-18 00:00:00	0
	150755	121865	Cadillac	Escalade	#ef1746	NULL	166365	CEQK8B2XV60RK87PT	ANF-3596	MN	2020-12-29 00:00:00	0
	150754	74142	Audi	A4	#79eae4	S line	18706	AAIOQJHWRFVRBDZR	IBC-0409	MH	2017-11-02 00:00:00	0

Column Name	Data Type	Null Count
idVehicle	int	0
Policy_ID	int	0
Make	varchar(50)	0
Model	varchar(50)	0
Color	varchar(50)	0
Trim	varchar(50)	41200
Mileage	int	0
VINNumber	varchar(20)	0
VehideNumberPlate	varchar(20)	0
VehideRegisteredState	varchar(50)	0
CreatedDate	datetime	0
Active	tinyint(1)	0

Data sources → Staging

Transform



Vehicle

```
# 1. Cập nhật Trim nếu NULL thành Other
df['Trim'] = df['Trim'].fillna('Other')
```

```
# 3. Xóa các cột không cần thiết
df.drop(columns=['CreatedDate', 'Active'], inplace=True)
```

```
# 2. Tạo cột MileageGroup
def classify_mileage_group(mileage):
    if mileage < 2000:
        return '<2000'
    elif 2000 <= mileage < 5000:
        return '2000-5000'
    elif 5000 <= mileage < 10000:
        return '5000-10000'
    elif 10000 <= mileage < 15000:
        return '10000-15000'
    elif 15000 <= mileage < 20000:
        return '15000-20000'
    else:
        return '>20000'

df['MileageGroup'] = df['Mileage'].apply(classify_mileage_group)
```

Data sources → Staging

After



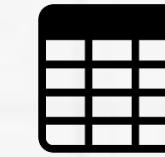
Vehicle

	<u>idVehicle</u>	<u>Policy_ID</u>	Make	Model	Color	Trim	Mileage	VINNumber	VehicleNumberPlate	VehicleRegisteredState	MileageGroup
0	1	51694	Acura	MDX	Yellow	Advance	170270	AM03LCBBY0S1KRNBC	AFN-4961	CA	>20000
1	2	87998	Ford	Mustang	Orange	Platinum	1776	FMJWYB5ZW6MZCB4MA	QNB-1453	OH	<2000
2	3	64767	Tesla	Model S	Silver	Other	51043	TMBKHUIXQ5F8PL2LY	LQI-6563	NY	>20000
3	4	113660	Land Rover	Discovery	Green	HSE	35192	LDS3VAUY45X9NKSFR	QPU-3164	FL	>20000
4	5	24886	Acura	RDX	Black	A-Spec	44302	AR7Y4R0QQ3TN004TR	CGN-6912	CA	>20000

Column Name	Data Type	Null Count
<u>idVehicle</u>	int	0
<u>Policy_ID</u>	int	0
Make	varchar(50)	0
Model	varchar(50)	0
Color	varchar(50)	0
Trim	varchar(50)	0
Mileage	int	0
VINNumber	varchar(20)	0
VehicleNumberPlate	varchar(20)	0
VehicleRegisteredState	varchar(50)	0
MileageGroup	varchar(255)	0

Data sources → Staging

Before



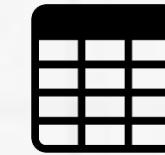
Vehicle_Driver

	<u>IdVehicle_Driver</u>	<u>Vehide_ID</u>	<u>Driver_ID</u>	<u>DriverForBusiness</u>	<u>IsPrimaryDriver</u>	<u>EveryDayMileage</u>	<u>CreatedDate</u>	<u>Active</u>
1	133708	142890	1	1	1	84	2019-01-05 00:00:00	1
2	629	170420	1	1	0	0	2023-03-14 00:00:00	0
3	51387	24664	1	1	0	325	2018-11-30 00:00:00	0
4	57605	176609	1	1	0	177	2018-08-15 00:00:00	0
5	128696	131432	0	0	0	488	2018-04-24 00:00:00	0
6	4086	110954	1	1	0	149	2023-11-07 00:00:00	1
7	124402	82133	1	1	0	41	2022-03-01 00:00:00	1
8	111577	117771	0	0	0	364	2020-01-08 00:00:00	0
9	37720	205907	1	1	0	466	2018-05-02 00:00:00	0

Column Name	Data Type	Null Count
<u>idVehicle_Driver</u>	int	0
<u>Vehicle_ID</u>	int	0
<u>Driver_ID</u>	int	0
<u>DriverForBusiness</u>	bit(1)	0
<u>IsPrimaryDriver</u>	bit(1)	0
<u>EveryDayMileage</u>	int	0
<u>CreatedDate</u>	datetime	0
<u>Active</u>	tinyint(1)	0

Data sources → Staging

After

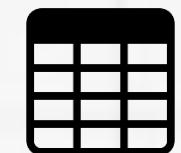


Vehicle_Driver

	<code>idVehicle_Driver</code>	<code>Vehicle_ID</code>	<code>Driver_ID</code>	<code>EveryDayMileage</code>	<code>DrivingPurpose</code>
0	1	133708	142890	32	Both
1	2	629	170420	38	Business
2	3	51387	24664	8	Business
3	4	57605	176609	69	Business
4	5	128696	131432	19	Other

Column Name	Data Type	Null Count
<code>idVehicle_Driver</code>	int	0
<code>Vehicle_ID</code>	int	0
<code>Driver_ID</code>	int	0
<code>EveryDayMileage</code>	int	0
<code>DrivingPurpose</code>	text	0

Data sources → Staging



Vehicle_Coverage

Before

Column Name	Data Type	Null Count
idVehicle_coverage	int	0
Vehicle_ID	int	0
Coverage_ID	int	0
Active	tinyint(1)	0
CreateDate	datetime	0

Transform

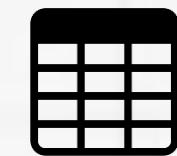


After

Column Name	Data Type	Null Count
idVehicle_coverage	int	0
Vehicle_ID	int	0
Coverage_ID	int	0

```
# Xóa cột Active và CreateDate  
df.drop(['Active', 'CreateDate'], axis=1, inplace=True)
```

Data sources → Staging



DTV

Before

Column Name	Data Type	Null Count
idDTV	int	0
Driver_ID	int	0
TrafficViolationCode_ID	int	0
Active	tinyint(1)	0

Transform

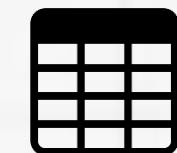


After

Column Name	Data Type	Null Count
idDTV	int	0
Driver_ID	int	0
TrafficViolationCode_ID	int	0

```
# 1. Xóa cột 'active'  
df.drop(columns=['active'], inplace=True)
```

Data sources → Staging

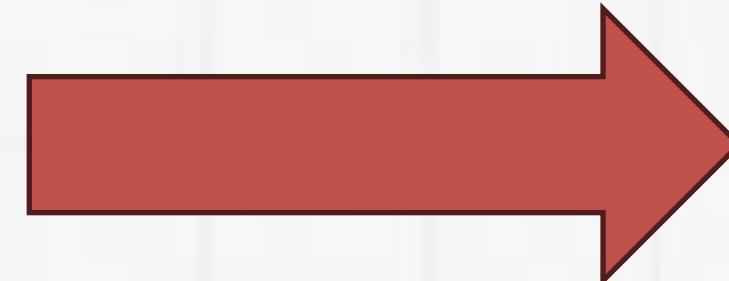


DriverAddress

Before

Column Name	Data Type	Null Count
idDriveAddress	bigint	0
Driver_ID	bigint	0
Address	text	0
City	text	0
State	text	0
ZipCode	text	0
Country	text	0
IsItGarageAddress	bigint	0
Region	varchar(50)	0

Transform



After

Column Name	Data Type	Null Count
idDriveAddress	int	0
Driver_ID	int	0
Address	text	0
City	text	0
State	text	0
Country	text	0
Region	varchar(50)	0

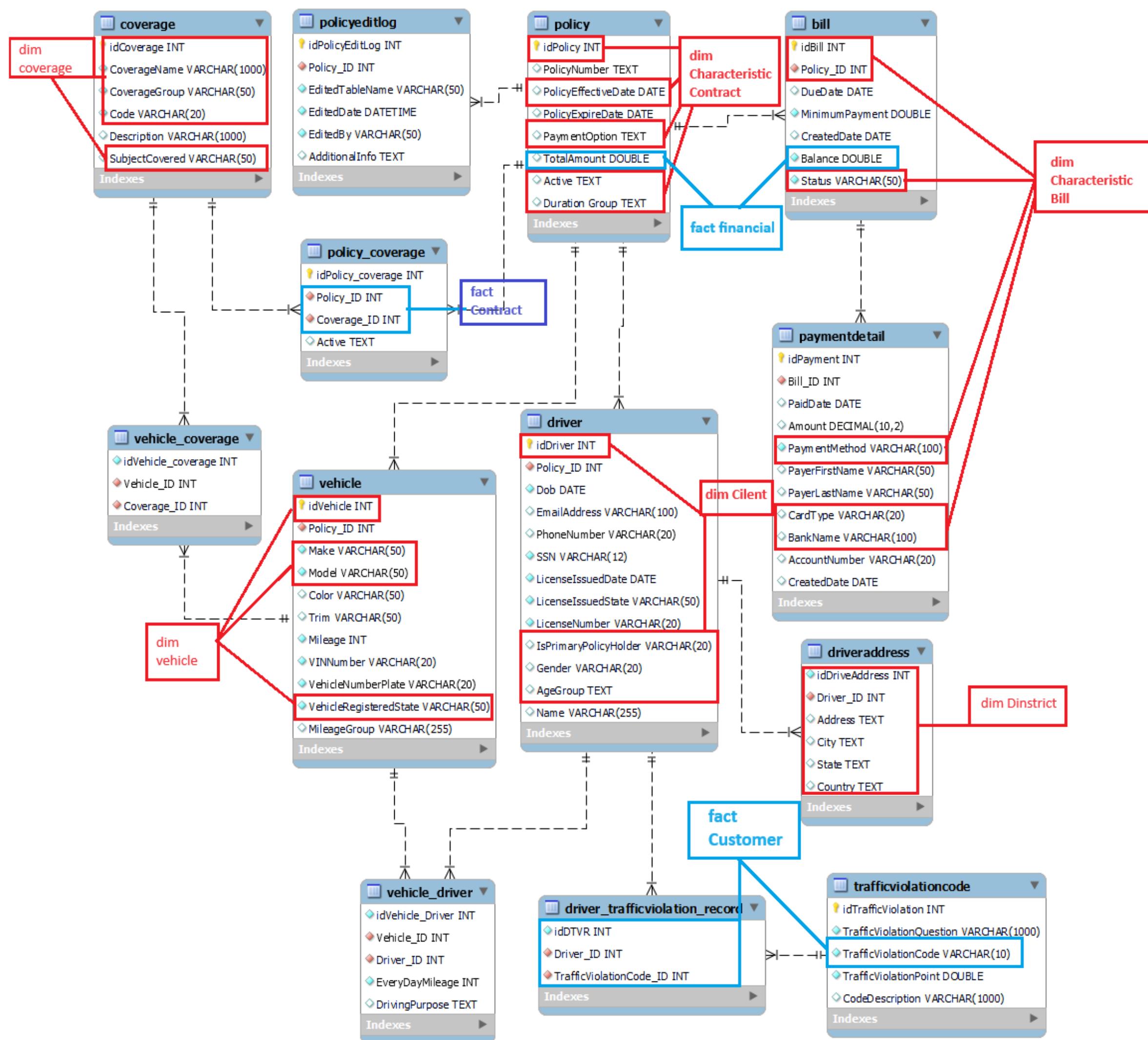
```
# 1. Xóa cột ZipCode và IsItGarageAddress  
df.drop(columns=['ZipCode', 'IsItGarageAddress'], inplace=True)
```

Các hoạt động ETL (OLTP → OLAP):

Nội dung	Công cụ sử dụng
Tạo view dim, fact	MySQL
Lưu dữ liệu vào data warehouse	

Các hoạt động ETL (OLTP → OLAP)

Các thuộc tính của các bảng trong OLTP
→ các thuộc tính của các bảng dim, fact trong OLAP



Các hoạt động ETL (OLTP → OLAP):

Đổ dữ liệu vào OLAP

```

2 •  create view v_dim_coverage as
3   select idCoverage, CoverageName, CoverageGroup, Code, SubjectCovered
4   from coverage;
5
6 •  create view v_dim_district as
7   select idDriveAddress, Driver_ID, City, State, Region, Country
8   from driveraddress;
9
10 •  create table dim_date (
11   idDate date,
12   days int,
13   month int,
14   quarter int,
15   year int,
16   primary key (idDate)
17 );
18
19 •  drop procedure if exists Dates;
20 delimiter |
21 •  create procedure Dates (dateStart date, dateEnd date)
22 begin
23   begin
24     while dateStart <= dateEnd do
25       INSERT INTO dim_date (idDate, days, month, quarter, year)
26       VALUES (dateStart, day(dateStart), MONTH(dateStart), quarter(dateStart), year(dateStart));
27       SET dateStart = DATE_ADD(dateStart, INTERVAL 1 DAY);
28     END WHILE;
29   end;
30
31 •  call Dates('2017-01-01','2020-12-31');
32

```

```

33 •  create view v_dim_date as
34   select * from dim_date;
35
36 •  create view v_dim_vehicle as
37   select idVehicle, vehicle.Make, vehicle.Model, vehicle.VehicleRegisteredState
38   from vehicle;
39
40 •  create view v_dim_client as
41   select driver.idDriver, driver.IsPrimaryPolicyHolder, driver.AgeGroup, driver.Gender
42   from driver;
43
44 •  create view v_dim_characteristic_contract as
45   select policy.idPolicy, policy.PaymentOption, policy.DurationGroup, policy.Active, policy.PolicyEffectiveDate
46   from policy;
47
48 •  create view v_dim_characteristic_bill as
49   select bill.idBill, bill.Policy_ID, bill.Status, paymentdetail.PaymentMethod, paymentdetail.CardType, paymentdetail.BankName
50   from bill join paymentdetail on bill.idBill = paymentdetail.Bill_ID;
51
52 •  create view v_dim_traffic_violations as
53   select trafficviolationcode.idTrafficViolation, trafficviolationcode.TrafficViolationQuestion, trafficviolationcode.TrafficViolationPoint
54   from trafficviolationcode;
55

```

Các hoạt động ETL (OLTP → OLAP):

Đổ dữ liệu vào OLAP

```
73 •  create view v_fact_contract as
74    select policy.idPolicy, policy_coverage.Policy_ID, policy_coverage.Coverage_ID, policy.PolicyNumber, policy.PolicyEffectiveDate, policy.PaymentOption,
75        policy.DurationGroup, policy.Active, coverage.CoverageName, coverage.CoverageGroup, coverage.SubjectCovered, dim_date.idDate
76    from policy join policy_coverage on policy.idPolicy = policy_coverage.Policy_ID
77        join coverage on coverage.idCoverage = policy_coverage.Coverage_ID
78        join dim_date on policy.PolicyEffectiveDate = dim_date.idDate;
79
80 •  create view v_fact_customer as
81    select
82        driver.idDriver, driver.Name, driver.AgeGroup, driver.Gender, driver.IsPrimaryPolicyHolder, driver.Policy_ID, policy.PolicyEffectiveDate, driveraddress.City, driveraddress.State, driveraddress.Country,
83        driveraddress.Region, driveraddress.Driver_ID AS Address_Driver_ID, vehicle.Make, vehicle.Model, vehicle_driver.DrivingPurpose, vehicle_driver.Driver_ID AS VehicleDriver_ID, vehicle.idVehicle,
84        driver_trafficViolation_record.idDTVR,driver_trafficViolation_record.Driver_ID,driver_trafficViolation_record.TrafficViolationCode_ID,trafficViolationcode.idTrafficViolation
85    FROM
86        driver
87    JOIN
88        driveraddress ON driver.idDriver = driveraddress.Driver_ID
89    JOIN
90        vehicle_driver ON driver.idDriver = vehicle_driver.Driver_ID
91    JOIN
92        vehicle ON vehicle_driver.Vehicle_ID = vehicle.idVehicle
93    JOIN
94        policy ON policy.idPolicy = driver.Policy_ID
95    join driver_trafficViolation_record on driver_trafficViolation_record.Driver_ID = driver.idDriver
96    join trafficViolationcode on trafficViolationcode.idTrafficViolation = driver_trafficViolation_record.TrafficViolationCode_ID;
97
98 •  create view v_fact_financial as
99    select
100        policy.idPolicy, policy.TotalAmount,bill.idBill,bill.Policy_ID,bill.Balance,
101        bill.CreatedDate,bill.Status,paymentdetail.PaidDate,paymentdetail.Amount,paymentdetail.PaymentMethod,paymentdetail.CardType,
102        paymentdetail.BankName, paymentdetail.idPayment,paymentdetail.Bill_ID
103    from policy join bill on policy.idPolicy = bill.Policy_ID
104        join paymentdetail on paymentdetail.Bill_id = bill.idBill;
---
```

Các hoạt động ETL (OLTP → OLAP):

Đổ dữ liệu vào OLAP

Views	
▶	v_dim_characteristic_bill
▶	v_dim_characteristic_contract
▶	v_dim_client
▶	v_dim_coverage
▶	v_dim_date
▶	v_dim_district
▶	v_dim_traffic_violations
▶	v_dim_vehicle
▶	v_fact_contract
▶	v_fact_customer
▶	v_fact_edit
▶	v_fact_financial

2. PHÂN TÍCH VÀ THIẾT KẾ

- Khám phá dữ liệu
- Kiến trúc data warehouse
- Nội dung ETL
- Hệ thống dimension
- Data model



2

PHÂN TÍCH VÀ THIẾT KẾ

HỆ THỐNG DIMENSION

31 giá trị	12 giá trị	4 giá trị	3 giá trị	5 giá trị	2 giá trị	9 giá trị	2 giá trị	3 giá trị
date	month	quarter	year	PaymentOption	Active	CoverageName	CoverageGroup	SubjectCovered
			2017	OneTime	Active	Bodily Injury Coverage	Required Auto Insurance	PolicyCoverage
			2018	Month	Expire	Rental Reimbursement	Additional Car Insurance	VehicleCoverage
			2019	Quarterly		Emergency Roadside Assistance		Both
			2020	Semi-Annual		Medical Payments		
				Annual		Collision		
						Property Damage Coverage		
						Uninsured Motorist Coverage		
						Mechanical Breakdown Insurance		
						Comprehensive		

20 giá trị	6 giá trị	29 giá trị	142 giá trị	4 giá trị	2 giá trị	23 giá trị	14 giá trị	4 giá trị	1 giá trị
TrafficViolationQuestion	TrafficViolationPoint	Make	Model	AgeGroup	Gender	City	State	Region	Country
	1	Acura	MDX	16-29	Male	Tallahass ee	FL	West	USA
	2	Ford	Mustang	30-45	Female	San Diego	CA	Northeast	
	3	Tesla	Model S	46-60		Los Angeles	NC	Midwest	
	4	Land Rover	Discovery	older than 60		Charlotte	NY	South	
	5	Volkswage n	RDX			Albany	PA		
	6	Porsche	Jetta			San Jose	IL		
		Lexus	Panamera			York	TX		
		Buick	NX			Chicago	VA		
		Lincoln	Enclave			Houston	WA		
		Chevrolet	...			Richmon d	MA		
		Infiniti				Harvard	GA		
		...				Miami	OH		
						Auburn	MI		
						Boston	NJ		

2. PHÂN TÍCH VÀ THIẾT KẾ

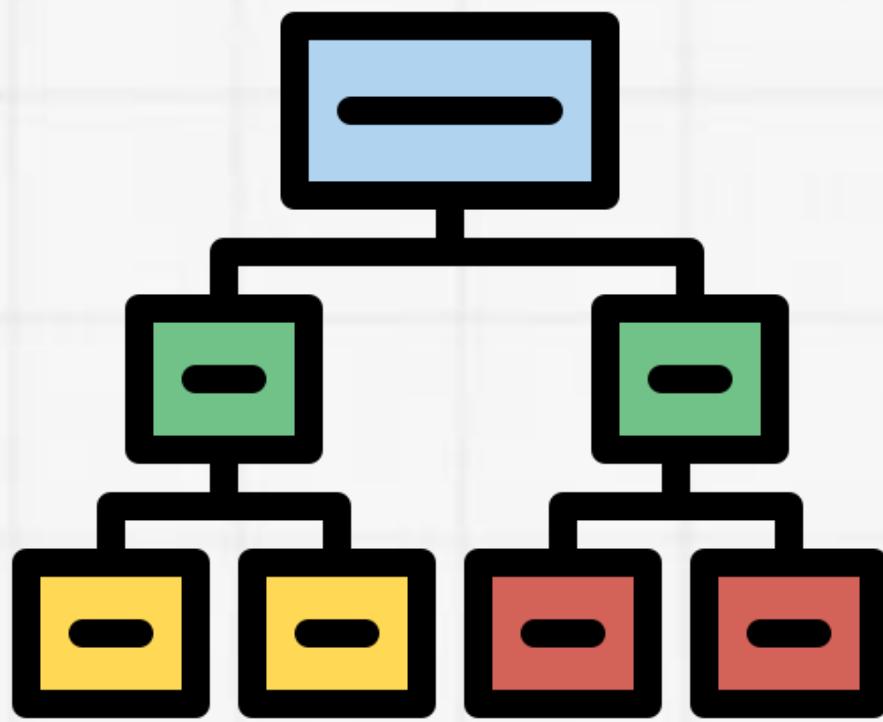
- Khám phá dữ liệu
- Kiến trúc data warehouse
- Nội dung ETL
- Hệ thống dimension
- Data model



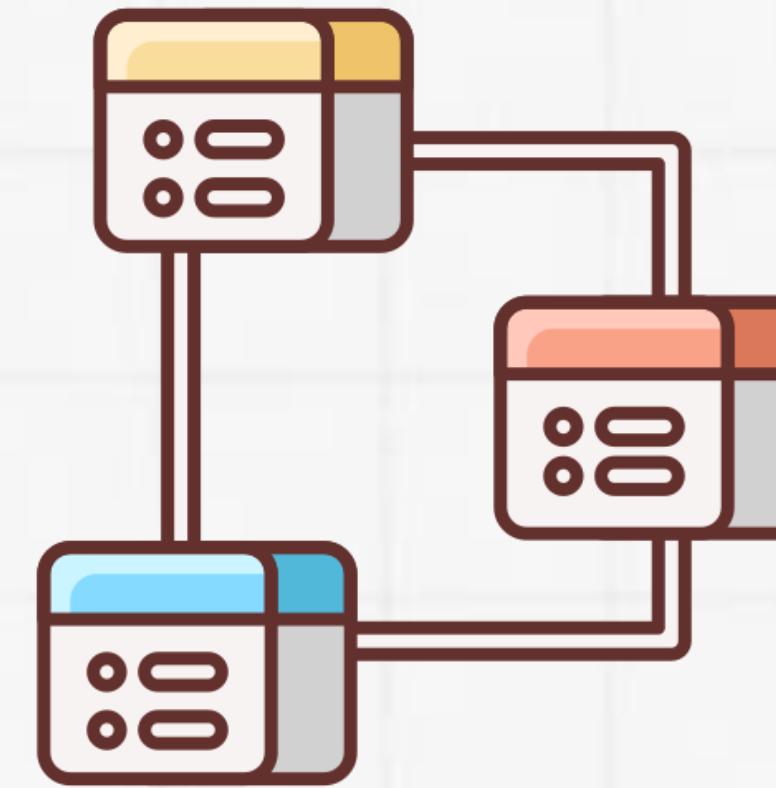
2

PHÂN TÍCH VÀ THIẾT KẾ

DATA MODEL

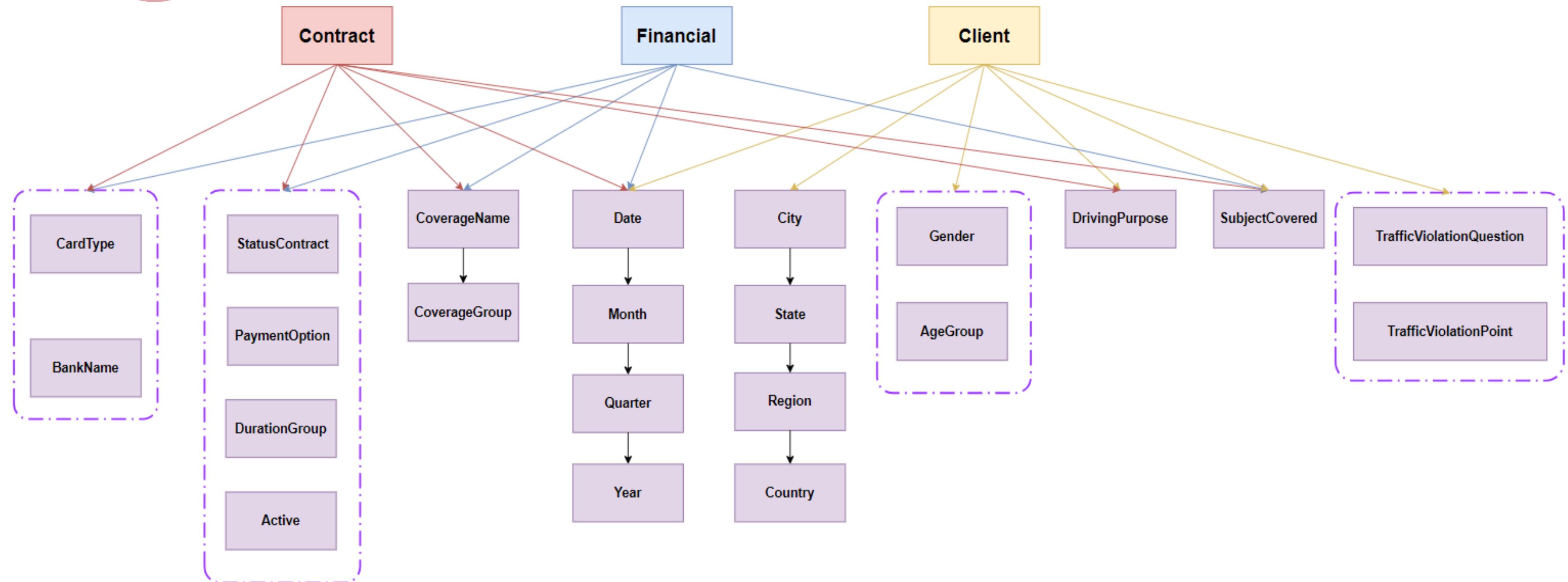


Data model logic

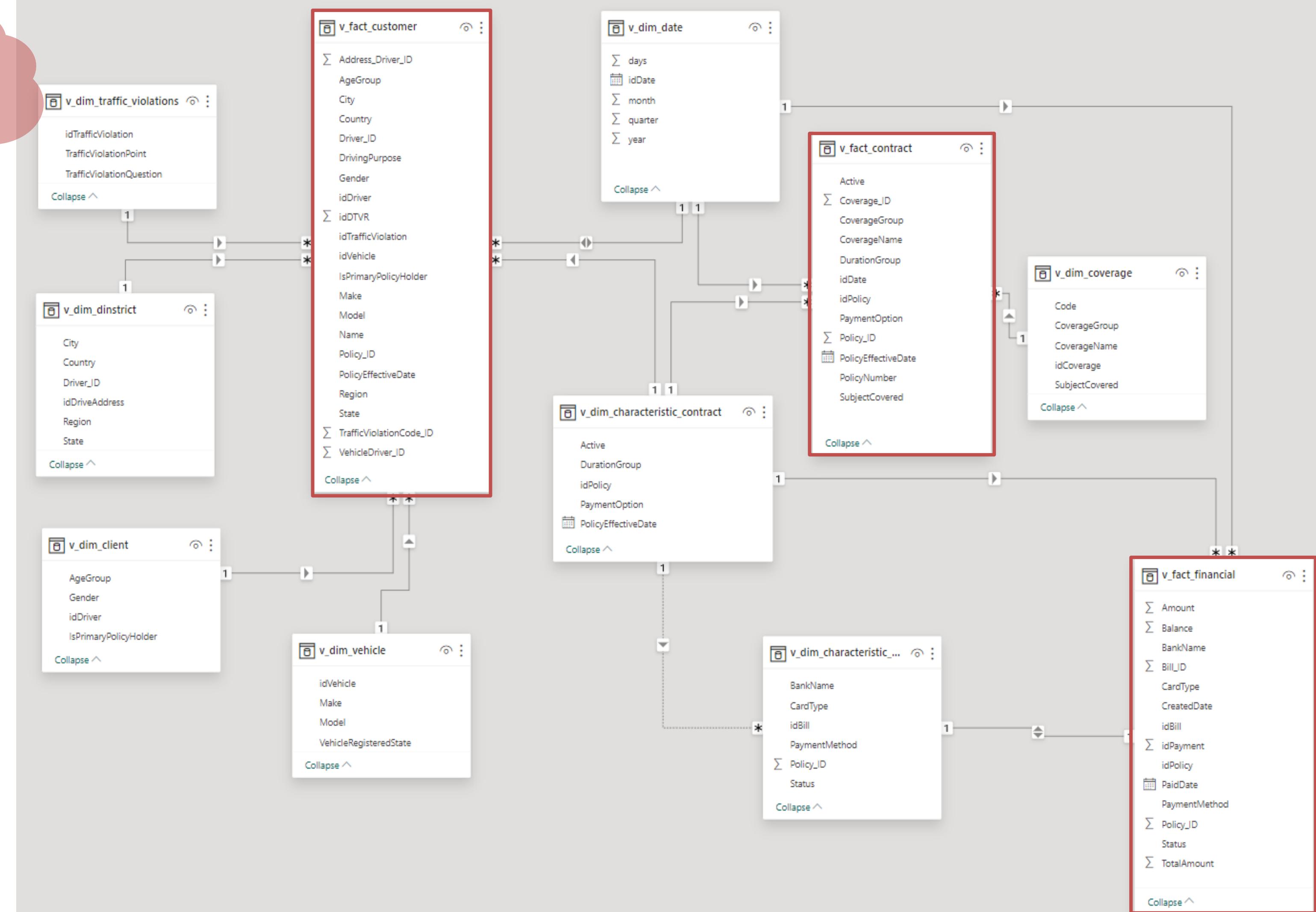


Data model vật lý

DATA MODEL LOGIC



DATA MODEL VẬT LÝ



3. XÂY DỰNG HỆ THỐNG



DASHBOARD



CONTRACT
DASHBOARD



FINANCIAL
DASHBOARD



3. XÂY DỰNG HỆ THỐNG



DASHBOARD



Một số hàm Dax đã sử dụng:

- 👉 **SUM()**: tính tổng số tiền
- 👉 **CALCULATE()**: tính tổng, tính số lượng với điều kiện bộc lọc năm 2017
- 👉 **COUNT()**: đếm số lượng người lái xe
- 👉 **YEAR(), MONTH(), QUARTER()**: lấy tháng, quý, năm từ Date

Một số measure đã sử dụng:

- 👉 **TargetCustomer2017**: để xác định mục tiêu số lượng khách hàng năm 2017
- 👉 **TargetPaidAmount2017**: để xác định mục tiêu tổng số tiền đã thanh toán năm 2017

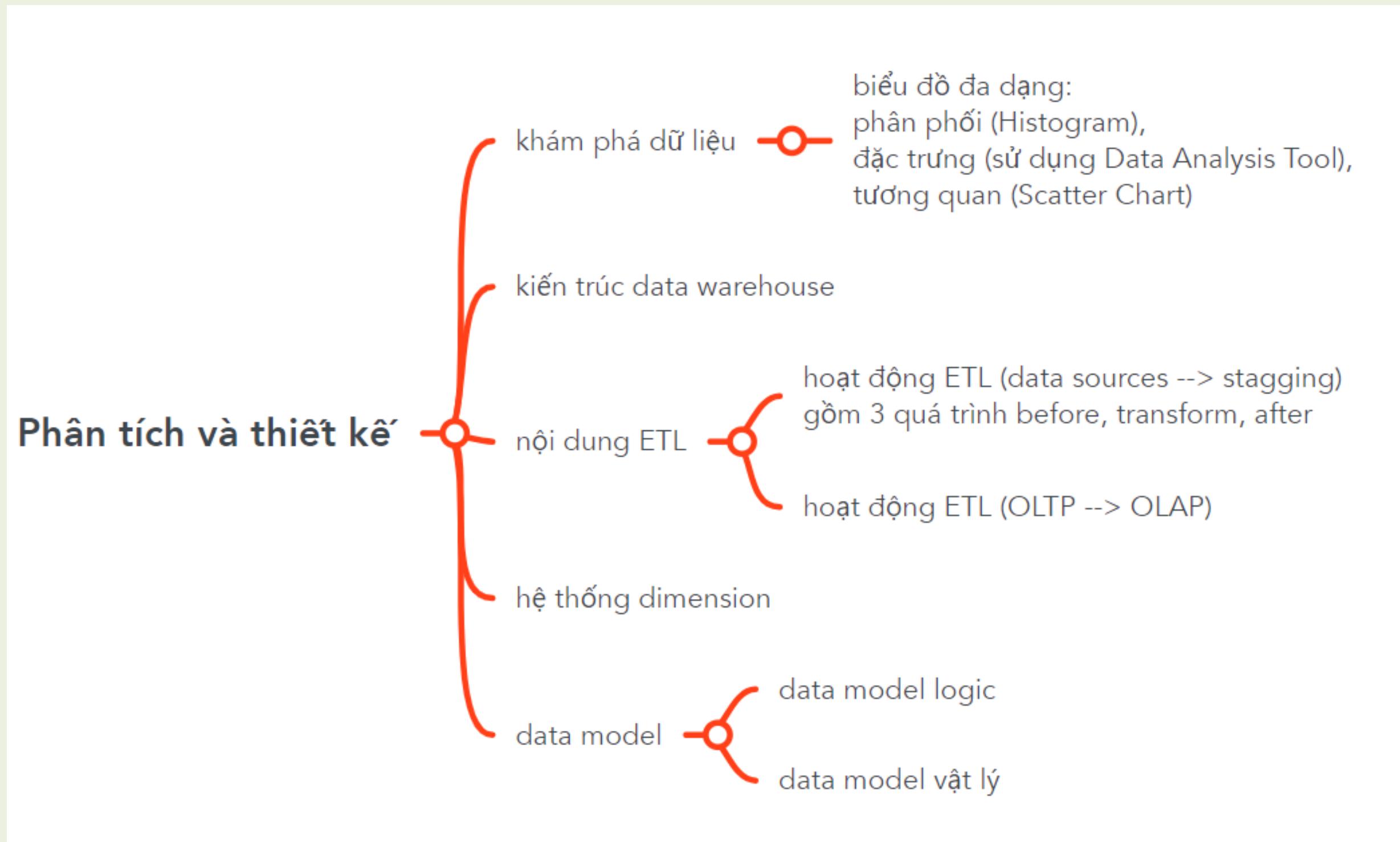
4. TỔNG KẾT

👍 ĐÃ LÀM ĐƯỢC:



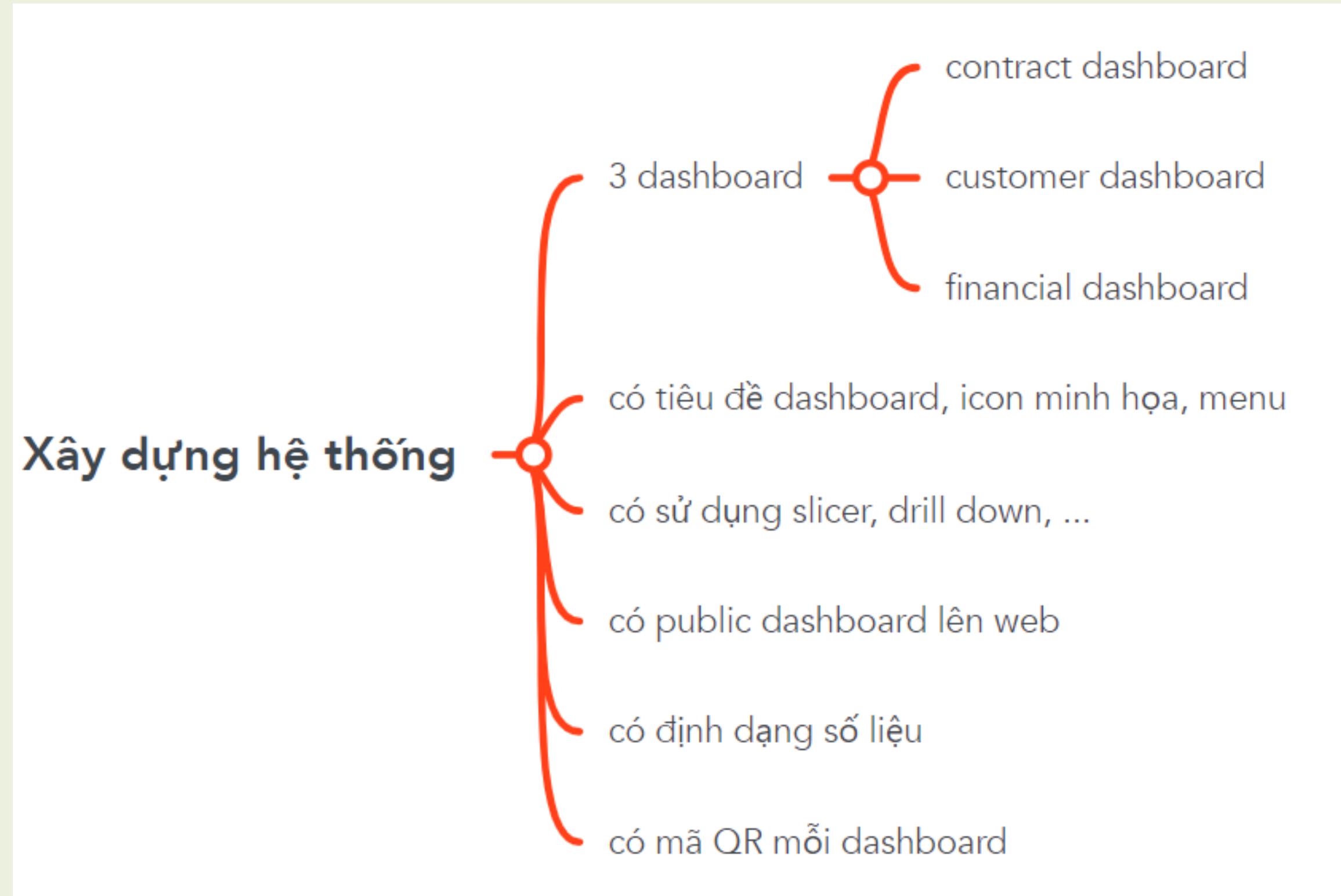
4. TỔNG KẾT

👍 ĐÃ LÀM ĐƯỢC:



4. TỔNG KẾT

👍 ĐÃ LÀM ĐƯỢC:



4. TỔNG KẾT

👎 CHƯA LÀM ĐƯỢC:

- Chưa xây dựng được hệ thống tự động cập nhật dữ liệu vào OLAP
- Dữ liệu tự sinh chưa sát với thực tế



4. TỔNG KẾT

