Analysis of Vehicle and Home Claims

**Group Project - BUAN 6320.002 Professor: Engin Calisir**

* **Arshiya Banu –** [**AXB200095@utdallas.edu**](mailto:AXB200095@utdallas.edu)
* **Naina Chowdhary Vallurupalli–** [**NVC20001@utdallas.edu**](mailto:NVC20001@utdallas.edu)
* **Hrishika Dubey –** [**HXD200013@utdallas.edu**](mailto:HXD200013@utdallas.edu)

# **Summary:**

The insurance claim is a crucial concept of the insurance industry. The insurance process models lay out each work step in a specific process to identify improvement to employee productivity, customer experience and/or risk management. Certain insurance processes can be complex, and steps for the same process (e.g., claims processing) vary based on the insurance product. For example, processing a life insurance claim is much different that processing an auto insurance claim. Other nuances, such as distribution channel (i.e., agent, broker, direct) and customer group (i.e., retail, group) should also be considered.

In this report we are focusing on the **Vehicle and Home claims**.

# **Business Rules:**

1. Every Policy is linked to a either Vehicle or Property as the coverable.
2. Every Policy has a Policy Type associated to it.
3. Every Policy has a Policy Status associated to it.
4. Multiple Policies can have the same Policy Status.
5. Multiple Policies can have the same Policy Type.
6. There can be multiple vehicles under a single policy.
7. There can be multiple properties on a single policy.
8. A given vehicle or property can only be covered on a single policy at a time.
9. Same type of policy can be purchased for different vehicles.
10. Same type of policy can be purchased for different properties.
11. A policy can have many claims.
12. A claim is associated for a single type of policy.
13. A vehicle or property can have many claims overall.
14. Every Policy has One or More Customers as policy Insureds.
15. There can be multiple customers on a Single Policy.
16. The same Customer can be part of Multiple Polices.
17. The Policy and Customer has Many-to-Many relationship.
18. A given claim is associated to a Single Loss Cause Type
19. A claim is associated to a single Claim Status.
20. A loss cause or loss type can be associated to multiple claims.
21. Multiple claims can be associated to the same Loss cause Type.
22. Multiple claims can have the same claim status.
23. A claim can be associated to multiple customers or insureds.
24. The same customer or Insured can be part of multiple claims. This means that the customer can have multiple claims.
25. The Claim and Customer has Many-to-Many relationship.
26. Multiple contacts can be associated with a vehicle or property or claims.
27. The customer can be a driver or Non-driver on an Auto Policy.
28. The Policy entity must include unique Policy, Policy Type, Policy Status, Customers, associated Claims, Property details (for a Home Policy), Vehicle Details(for an Auto policy), unique Policy Number, Account Number, Policy Effective date and Policy Expiration date.
29. The Customer entity must include unique CustomerID, customer personal details having First Name, Last Name, Middle Name, Driver Indicator, and Address related information having AddressLine1, City, State and Zipcode. The Table should also include the associated list of claims and Policies.
30. Policy which corresponds to a property should contain details including unique PropertyID, Location, Roof Type, Number of Stories, Number of units, the Square Footage Area, Construction Year. The table should also include the associated PolicyID.
31. For an Auto Policy, the Vehicle Entity must include unique VehicleID, the associated PolicyID, vehicle specific details such as Vehicle Type, Vehicle Style, Vehicle Model, Vehicle Make, VIN, License Plate number and Vehicle Year.
32. The Claim entity must include unique ClaimID, the associated Loss Cause details, Policy details, claim status details, the associated customer details, and claim specific details having Loss Date, unique Claim Number, claim closed Date, Closed Reason, the Loss Location Zipcode, the Loss Location State and the Line of Business indicating if the claim is an Auto or Property Claim.
33. The Policy Type entity is a standalone table comprising of default policy types available. The attributes comprise of the unique PolicyTypeID, Name and Description of the Policy Type. This table has a One-to-Many relationship with the Policy Entity.
34. The Policy Status Table is a standalone table consisting of a unique PolicyStatusID, and Name and Description of the status. This table has a One-to-Many relationship with the Policy Entity.
35. The Loss Cause Type entity must include the unique lossCauseID and the details of the lossCause like the Name and Description. This table has a One-to-Many relationship with the Claim Entity.
36. The Claim Status entity is a standalone table comprising of default claim statuses available. It must include the unique claimStatusID, Name and Description. This table has a One-to-Many relationship with the Claim Entity.

# **ERD Components:**

## **Entities:**

* Claim
* Policy
* Vehicle
* Property
* Contact
* Policy Status
* Policy Type
* Loss Cause Type
* Claim Status
* Assoc\_Policy\_Customer
* Assoc\_Claim\_Customer

## **List of Attributes:**

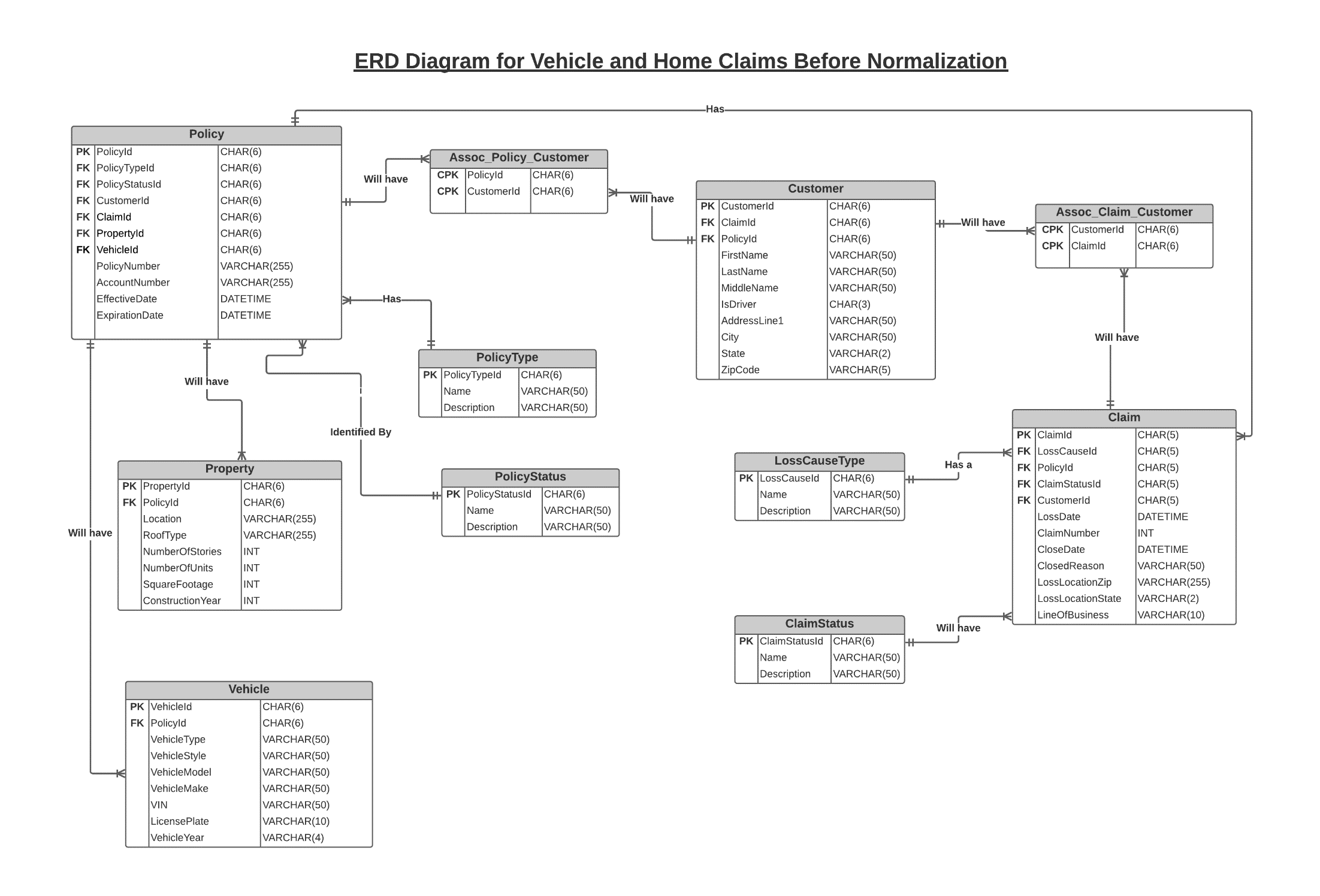
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Claim** | **Policy** | **Vehicle** | **Policy Status** | **Policy Type** |
| ClaimId  LossCauseId  PolicyId  ClaimStatusId  CustomerId  LossDate  ClaimNumber  CloseDate  ClosedReason  LossLocation  LossLocationState  LinesOfBusiness | PolicyId  PolicyTypeId  PolicyStatusId  CustomerId  ClaimId  PropertyId  VehicleId  PolicyNumber  AccountNumber  EffectiveDate  ExpirationDate | VehicleId  PolicyId  InsuredId  VehicleType  VehicleStyle  VehicleModel  VehicleMake  VIN  LiscensePlate  LiscenceNumber | PolicyStatusId  Name  Description | PolicyTypeId  Name  Description |

|  |  |
| --- | --- |
| **Assoc\_Policy\_Customer** | **Assoc\_Claim\_Customer** |
| PolicyId  CustomerId | CustomerId  ClaimId |

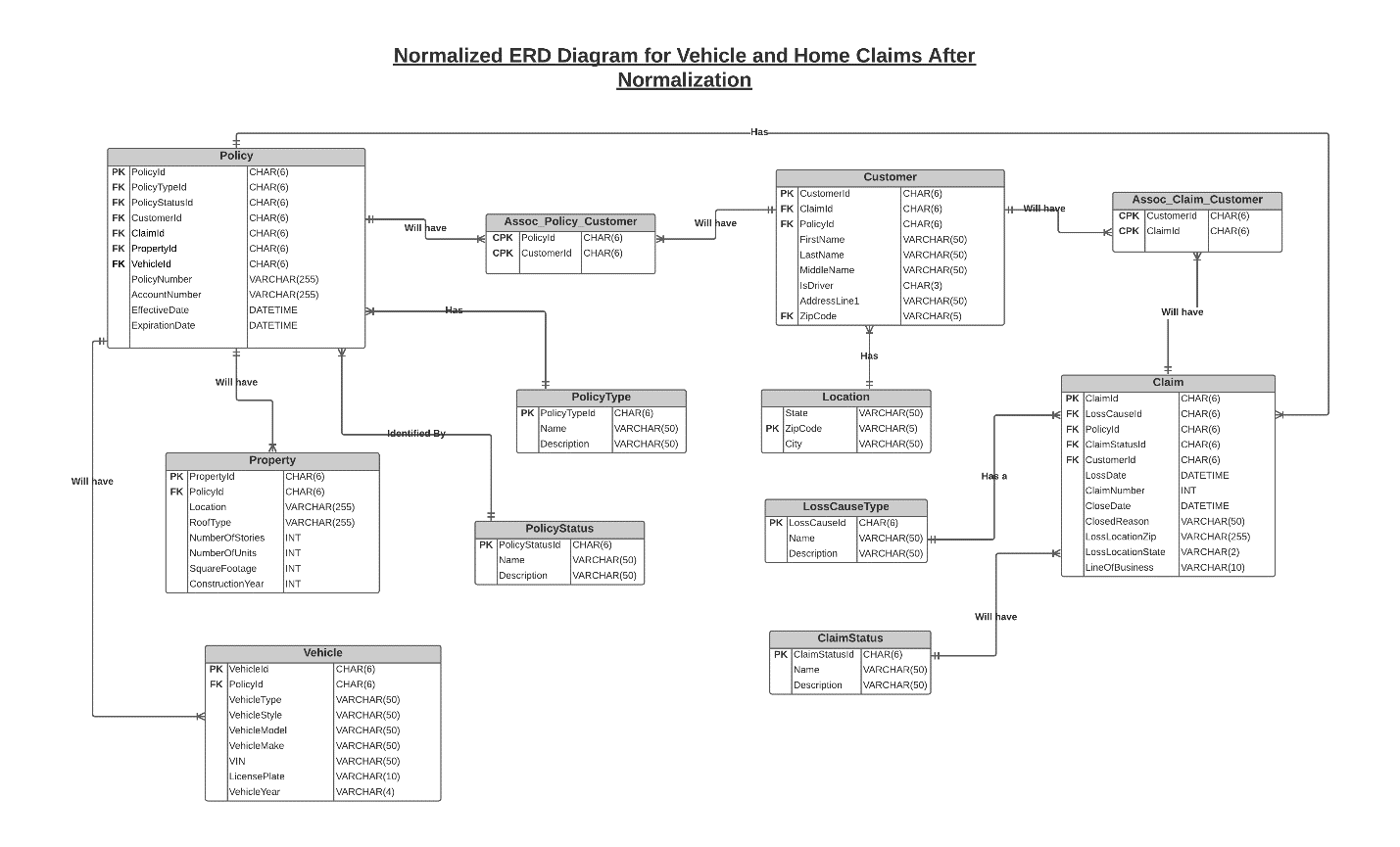
|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Customer** | **LossCauseType** | **ClaimStatus** |
| PropertyId  PolicyId  InsuredId  Location  RoofType  NumberOfStories  NumberOfUnits  SquareFootage  ConatructionYear | CustomerId  ClaimId  PolicyId  FirstName  LastName  MiddleName  IsDriver  AddressLine1  City  State  ZipCode | LosscauseId  Name  Description | ClaimStatusId  Name  Description |

# **Normalization:**

# **ERD For Analysis of Vehicle and Home Claims Before Normalization:**



# **ERD For Analysis of Vehicle and Home Claims After Normalization:**

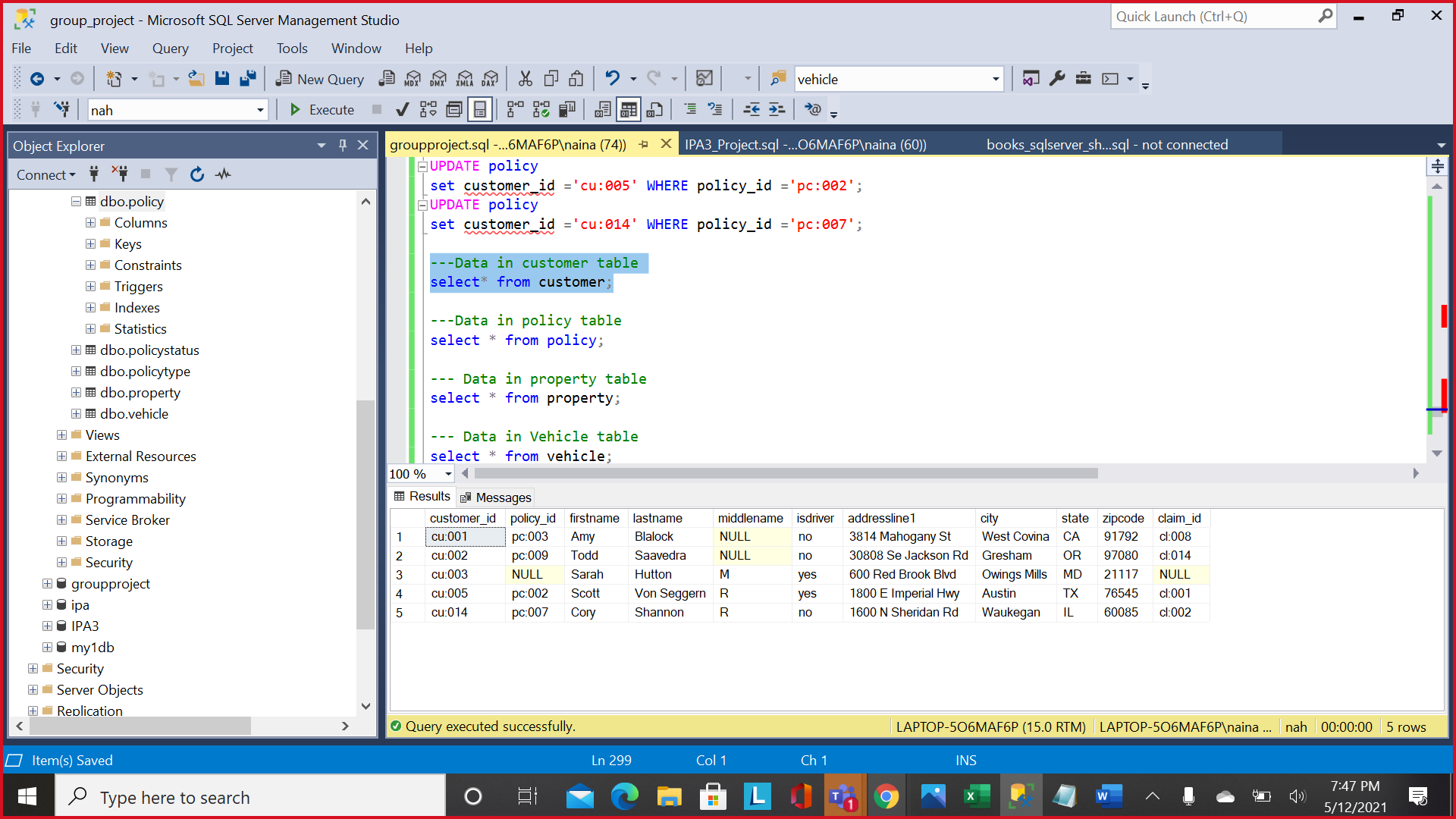


# **Note: We are adding normalization carried out on data with 2 sets of entries. Both follow 1nf, 2nf and 3nf steps from the master table created from sample data. One set has more complex scenarios of one person having multiple insurances in both property and vehicle category. The added scenarios present more ways to normalize the data. The Normalization excel files are attached.**

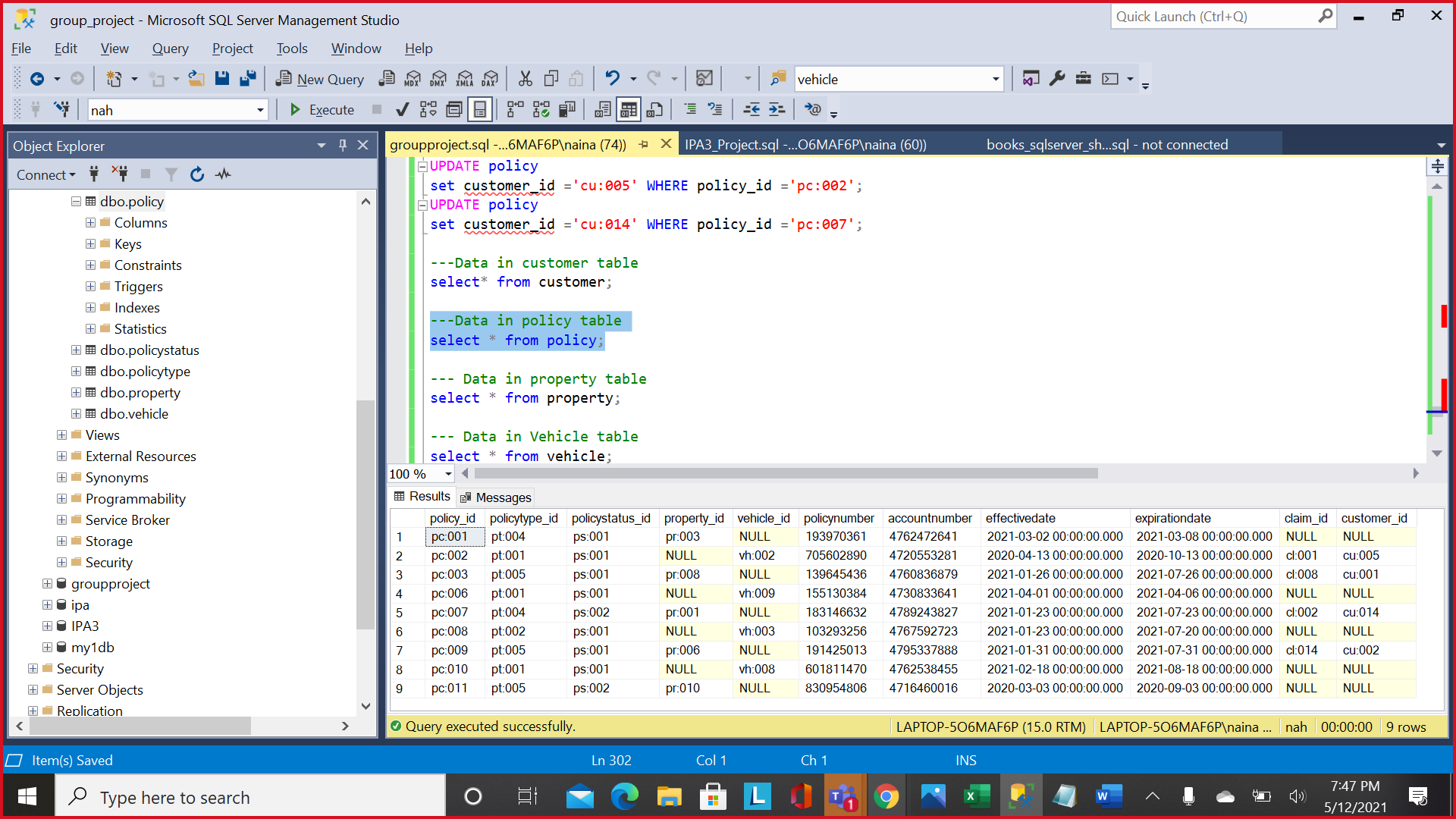
# **SQL: We have considered the ERD before normalization and written SQL queries using it.**

**Note: All queries are attached in dbo file**

**Customer Table**



**Policy Table**



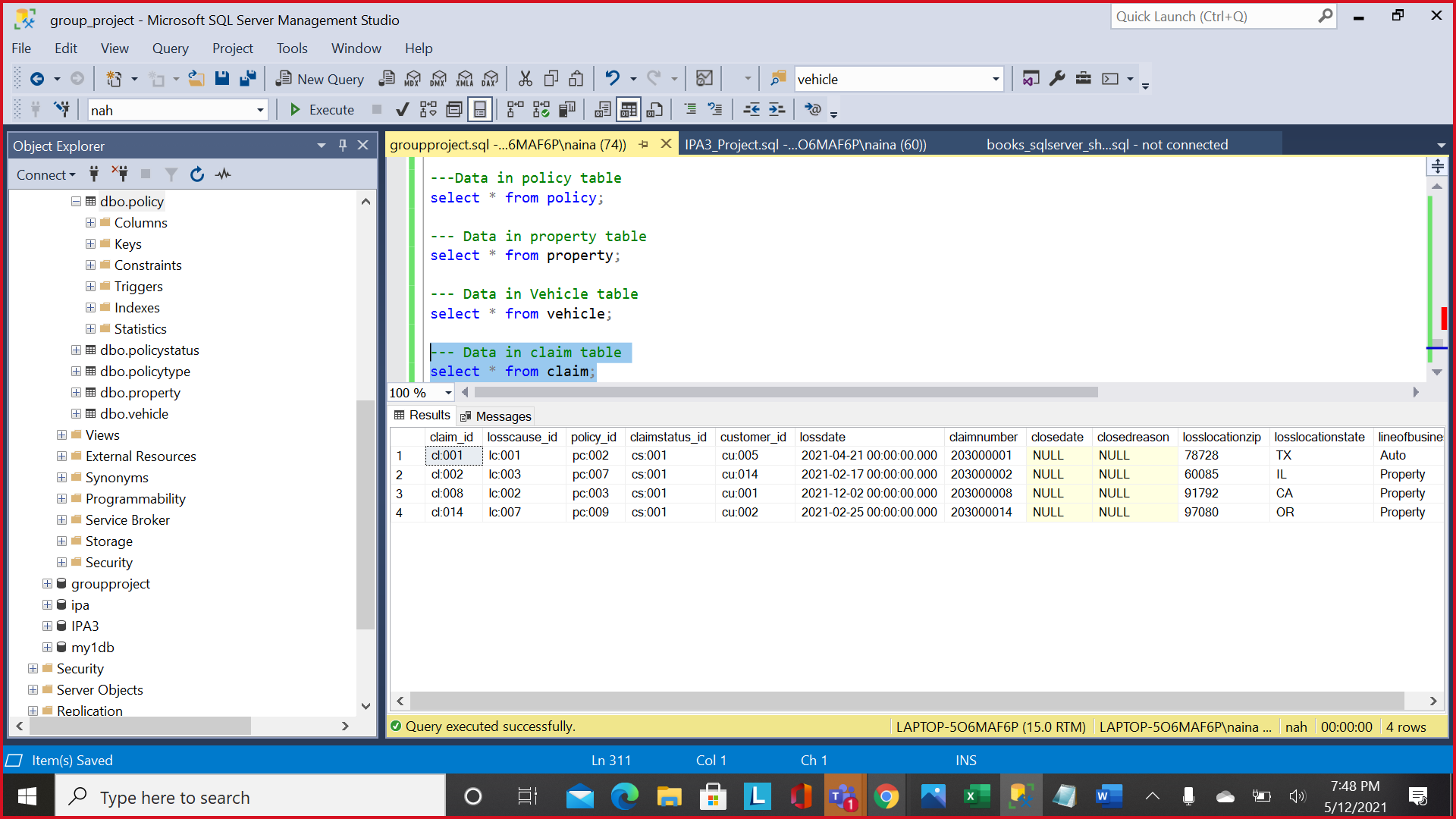
**Property Table**



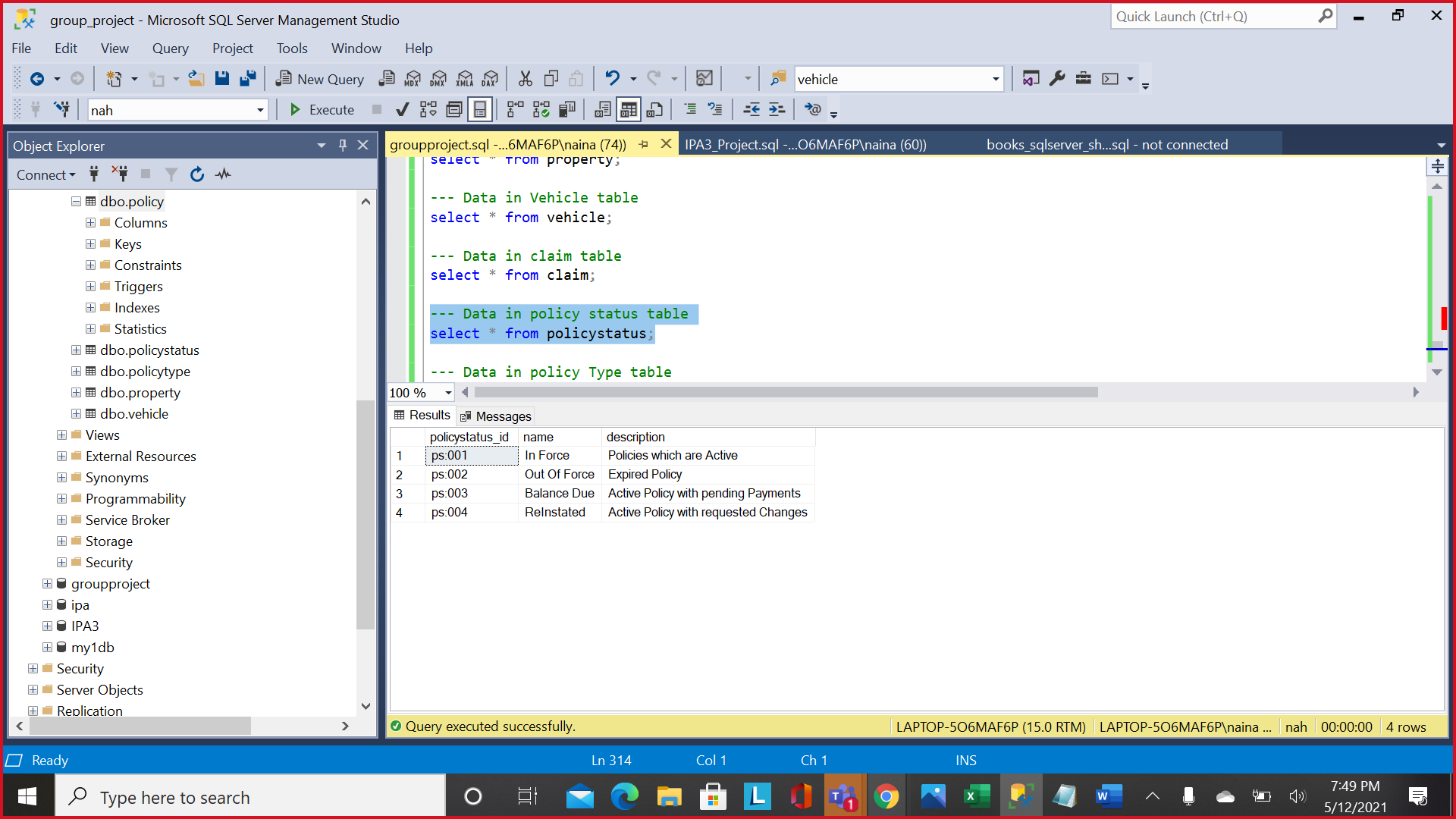
**Vehicle Table**



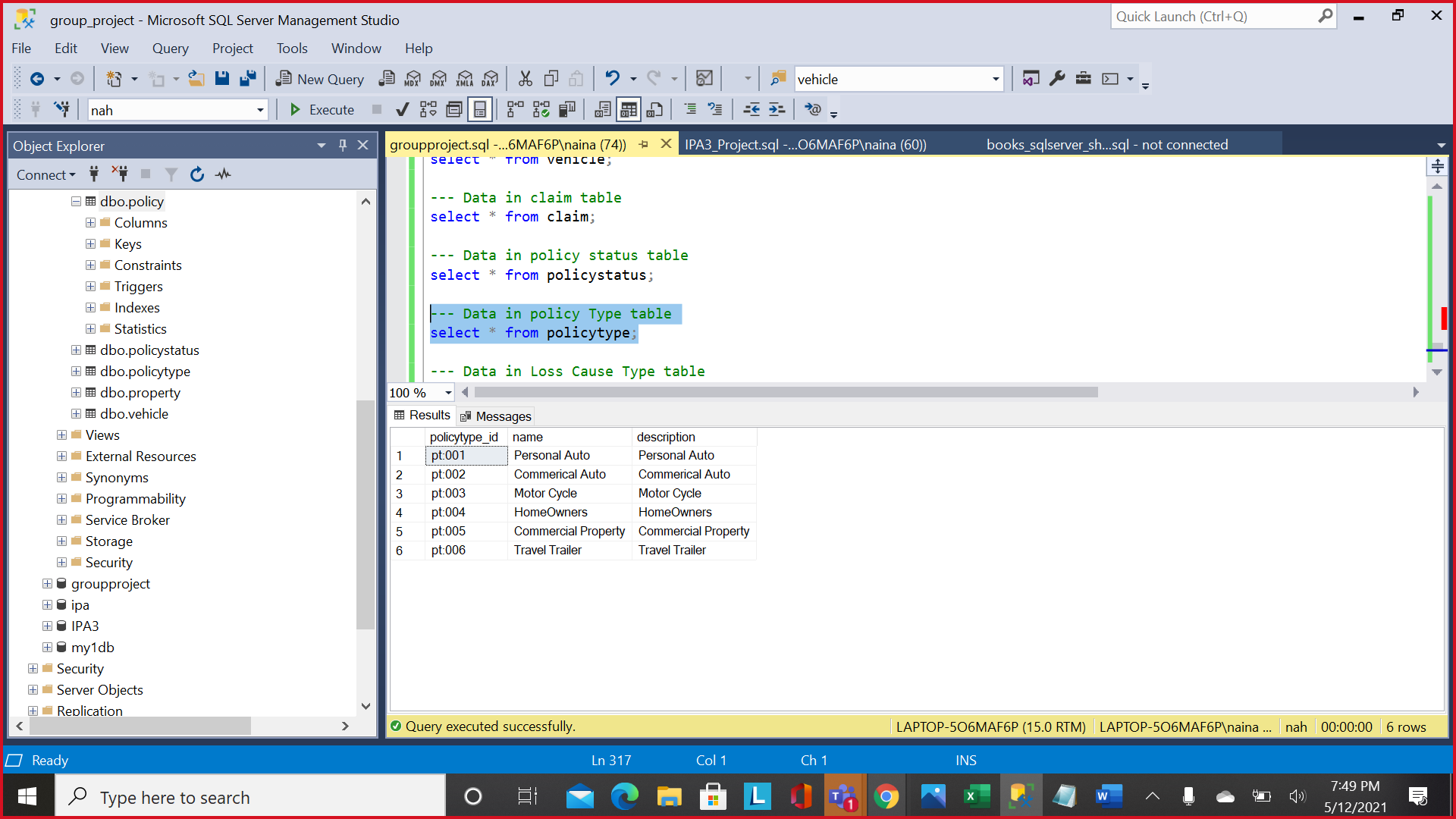
**Claim Table**



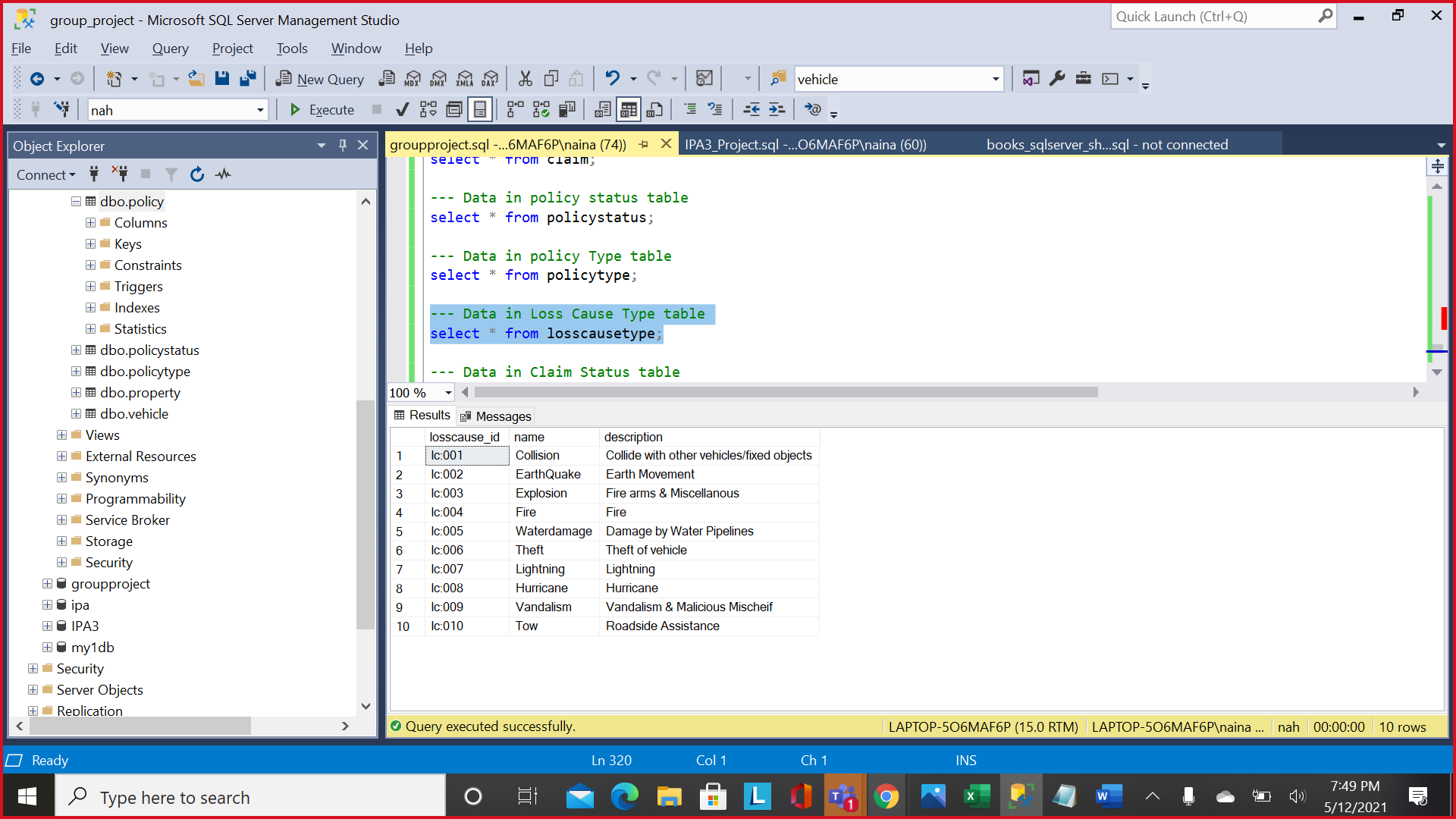
**Policy Status Table**



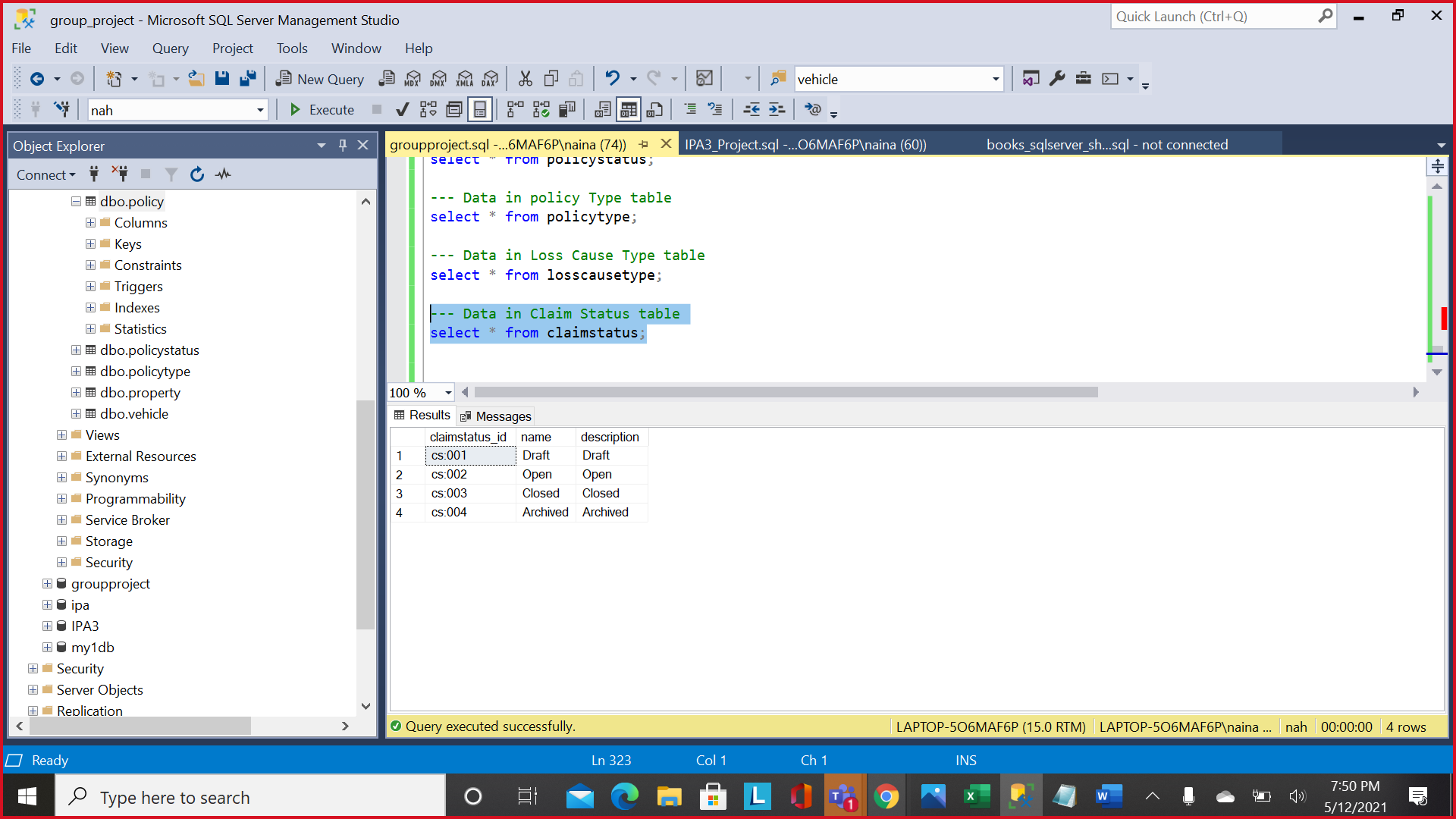
**Policy Type Table**



**Loss Cause Table**



**Claim Status Table**



**Assoc\_Claim\_Customer Table:**

Graphical user interface, text, application

Description automatically generated

**Assoc\_Policy\_Customer Table:**

Graphical user interface, text

Description automatically generated

1. Chose at least two (2) tables from your project that has relationship:

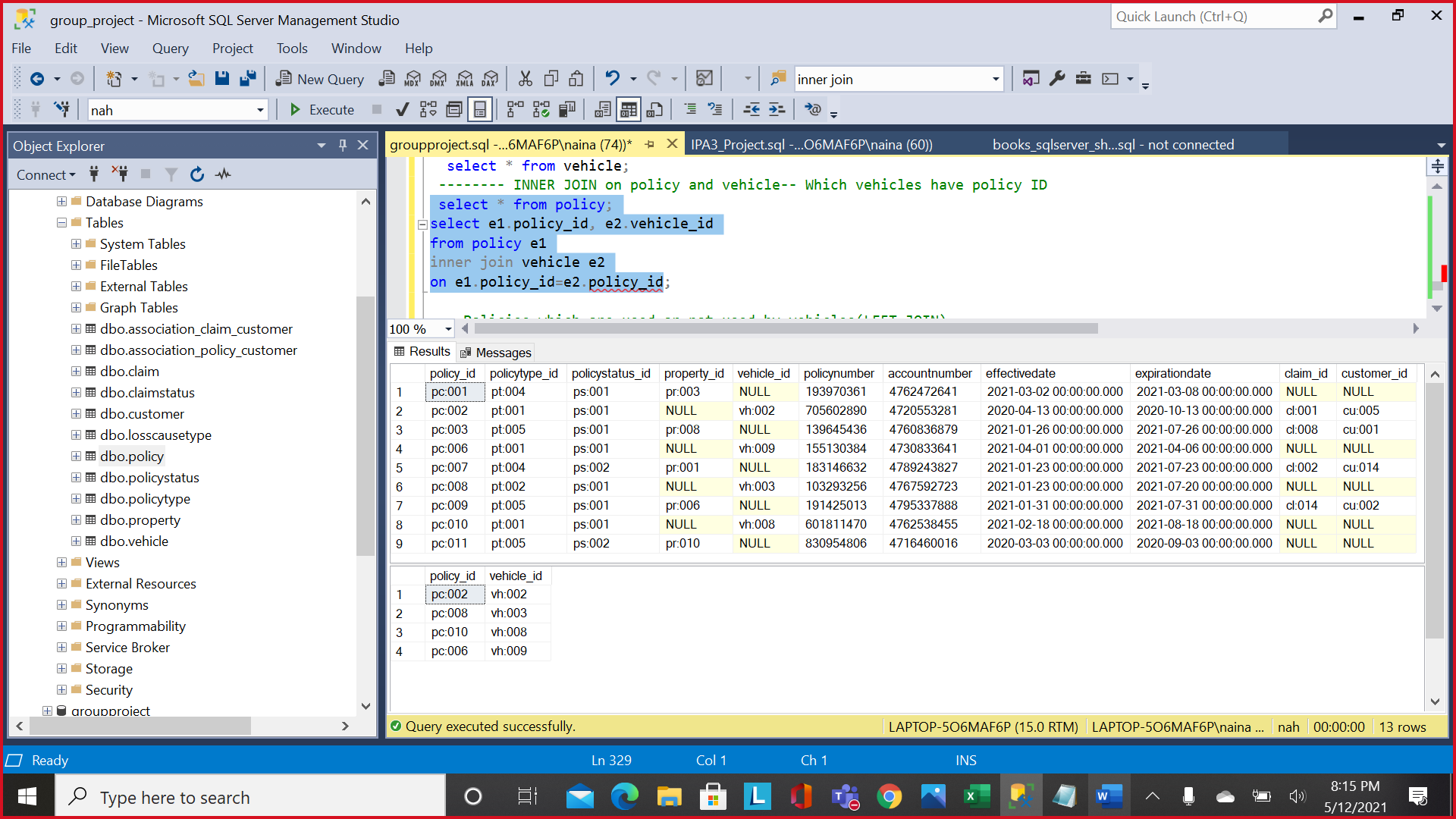
**Two Different tables have been selected for each type of join.**

1. Show your table records.

Attached screenshots above.

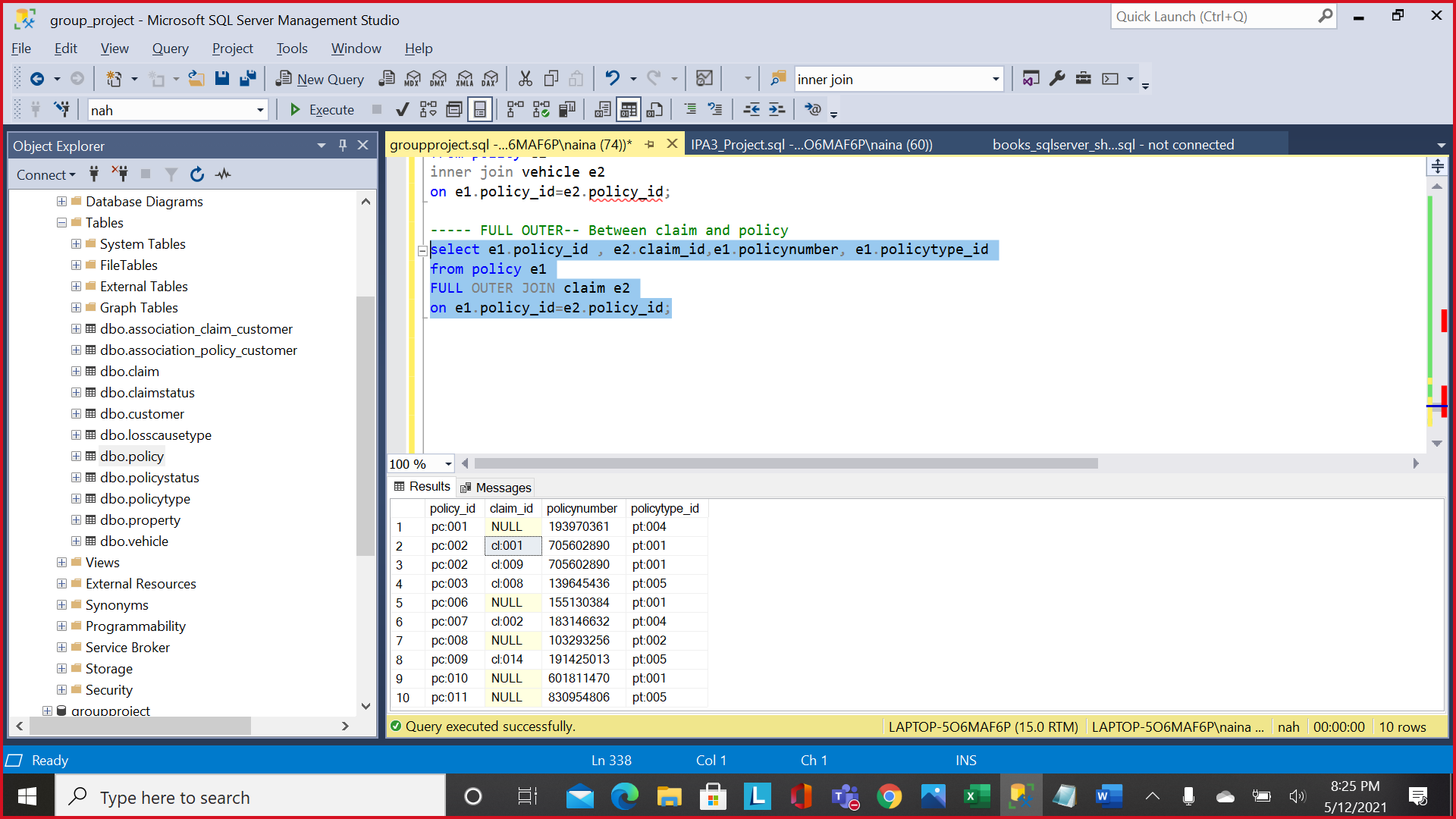
1. Create inner joins between two tables and explain what is the purpose of your join?

Ans: **This join fetches all the auto policies from the policy table.**

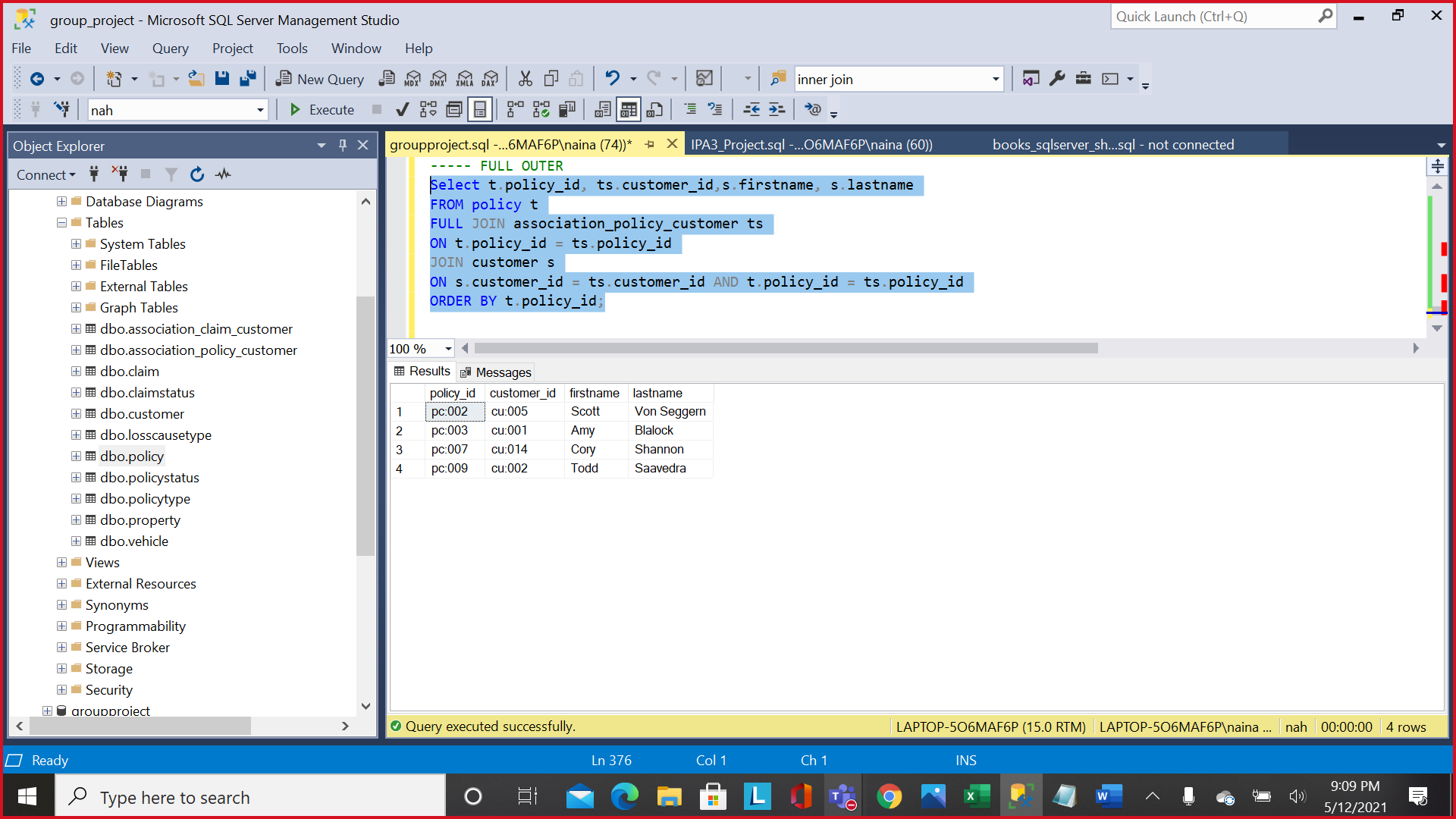


1. Create full joins between two tables and explain what is the purpose of your join?

Ans: **This join fetches all the policies and its associated claims**

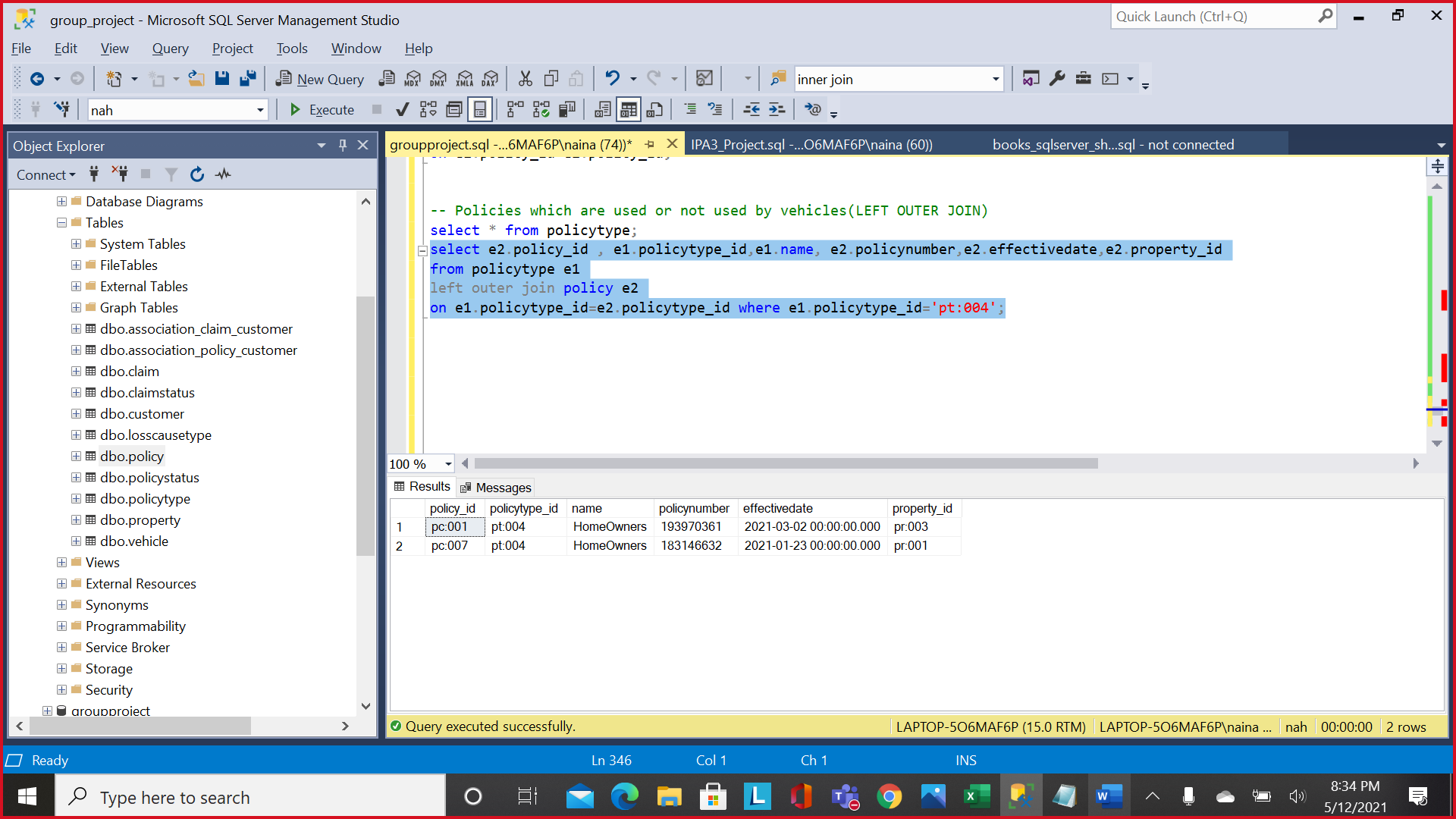


In another query, we are fetching policies associated with customers.



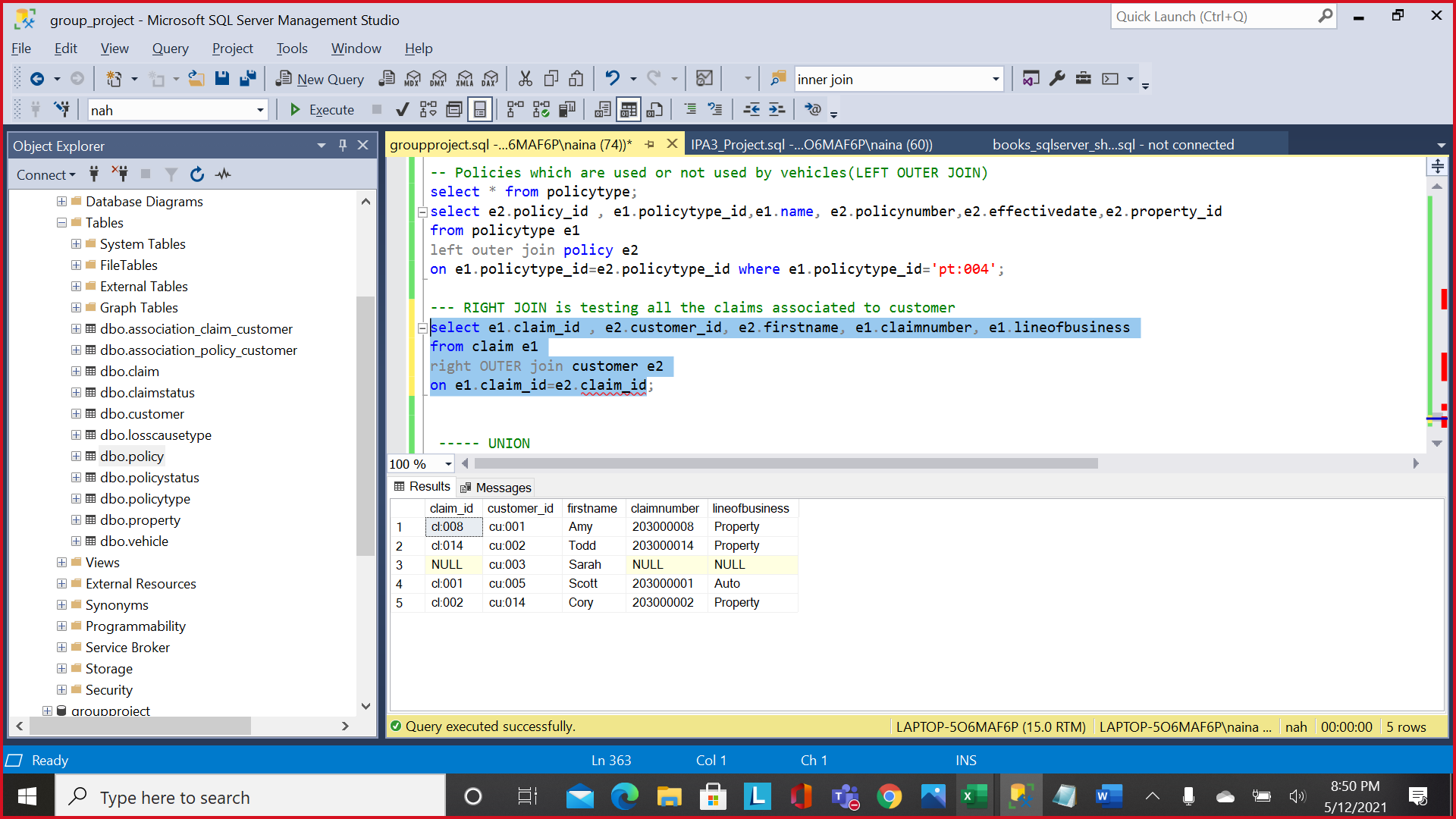
1. Create left outer joins between two tables and explain what is the purpose of your join?

Ans**: To fetch all available policies from policy table corresponding to policy type property (here, policy type is pt:004)**



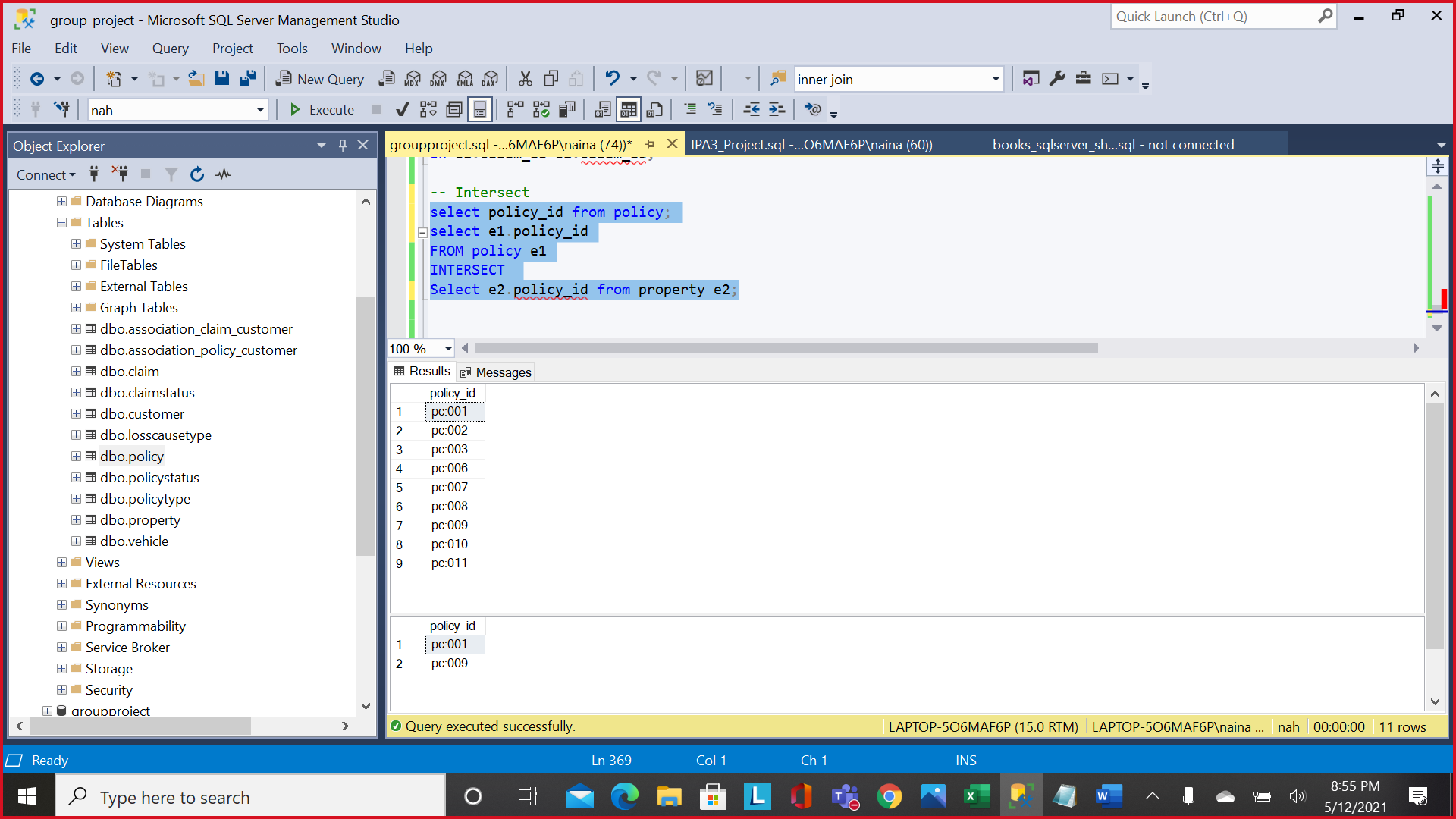
1. Create right outer joins between two tables and explain what is the purpose of your join?

Ans: **RIGHT JOIN is testing all the claims associated to customers.**



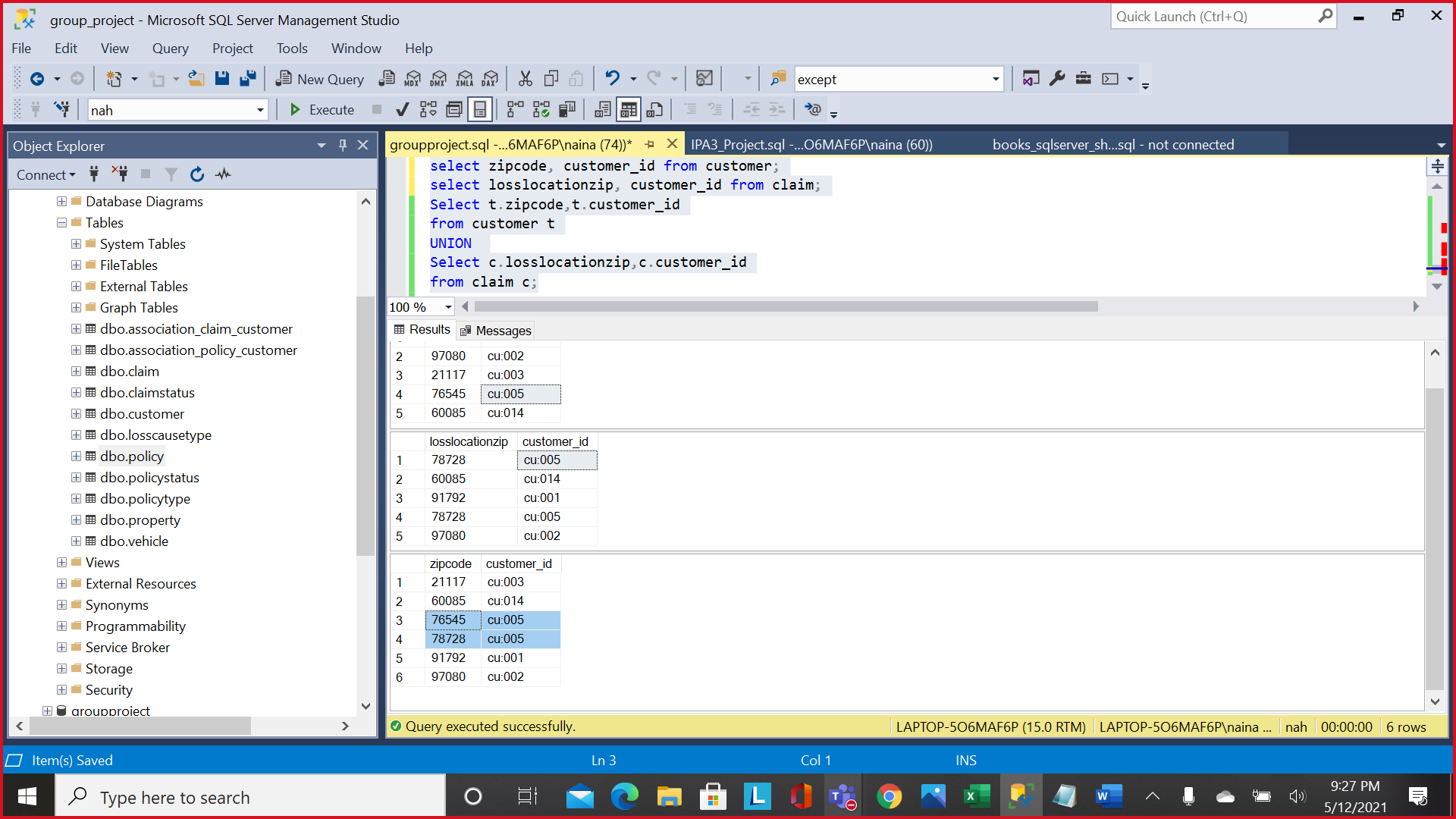
1. Create intersect between two tables and explain what is the purpose of your intersect?

Ans: **It returns common policy Id’s in both policy and property tables. It fetches all the property policies.**



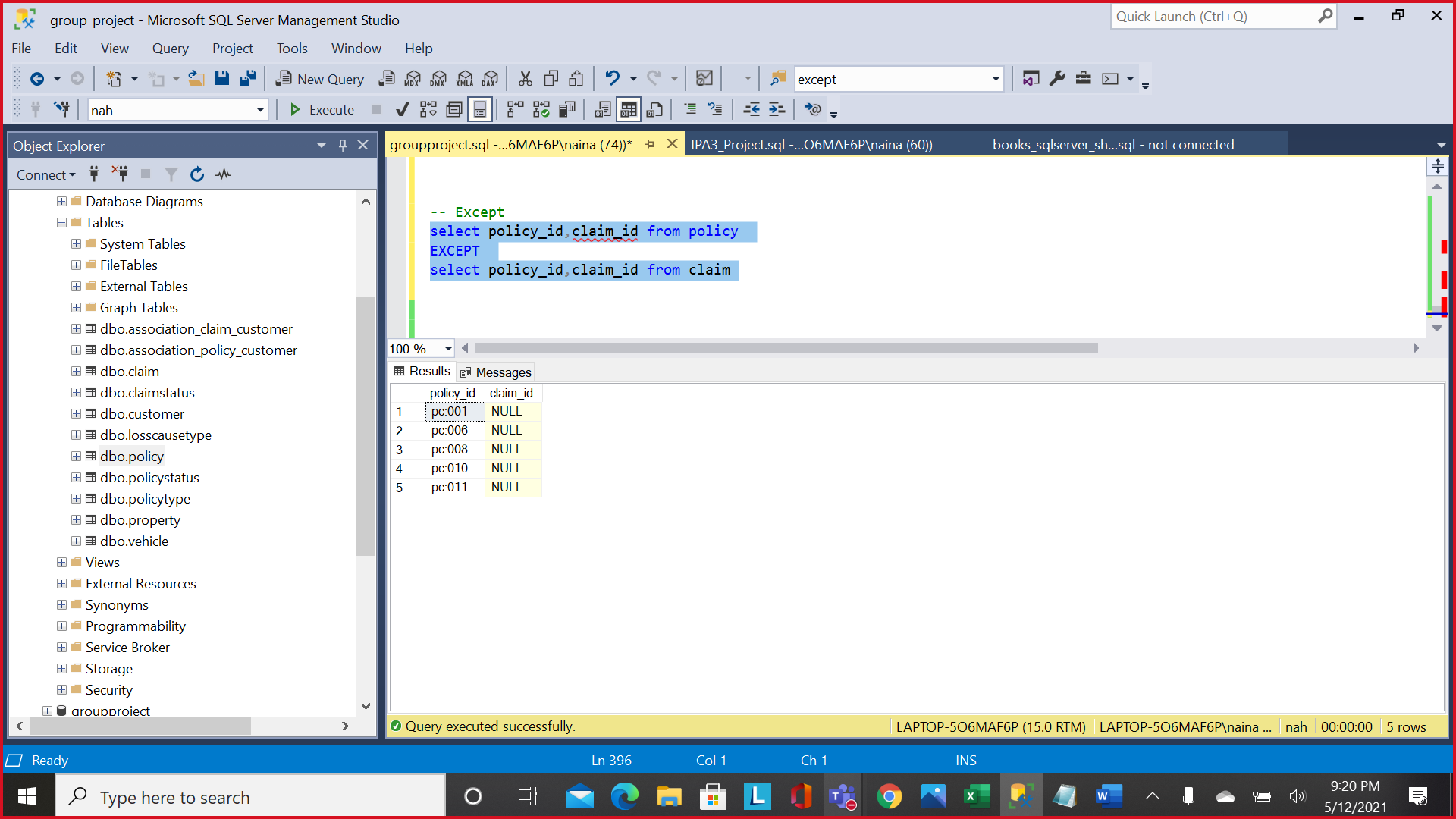
1. Create union between two tables and explain what is the purpose of your union?

Ans: **Fetching all results of all customers having loss location different than their insured location. Highlighted cu:005 has loss location different than insured location.**



1. Create except between two tables and explain what is the purpose of your except?

Ans: **Fetch all policies which have no claim associated to it.**



**Conclusion:**

With sample data and ERD diagram, we saw that performing normalization makes data and schema organization much more organized and efficient. Data storage becomes streamlined with repetitions being removed while still allowing retrieval with joins and other SQL query methods. This also improves performance of queries by allowing only the data needed to be queried faster.