Northeastern University
College *of* Engineering

IE 6750 Data Warehousing and Integration

Project Title

Policy Lapsation Trends and Insights in Life Insurance

Milestone-2

Group 4

Aayush Amrute – amrute.a@northeastern.edu
(NUID : 002838262)

Saurabh Chavan - chavan.sau@northeastern.edu
(NUID : 002479083)

Paritosh Vyawahare - vyawahare.p@northeastern.edu
(NUID : 002416079)

Submission Date : 2/05/2025

Table of Contents

Problem Definition:

Life insurance companies face a big challenge when it comes to managing policy lapsation. This happens because the data they rely on is often incomplete or inconsistent, coming from multiple sources like web sales, agents, and bank relationship managers. As a result, companies struggle to intervene on time, leading to delays and higher policy churn. For example, customers who miss their premium payments whether they are on autopay or non-autopay often end up lapsing their policies without receiving proper follow-ups.

Entity Relationship Diagram (ERD)

Entities and Their Attributes

The following entities are included in the ERD along with their key attributes:

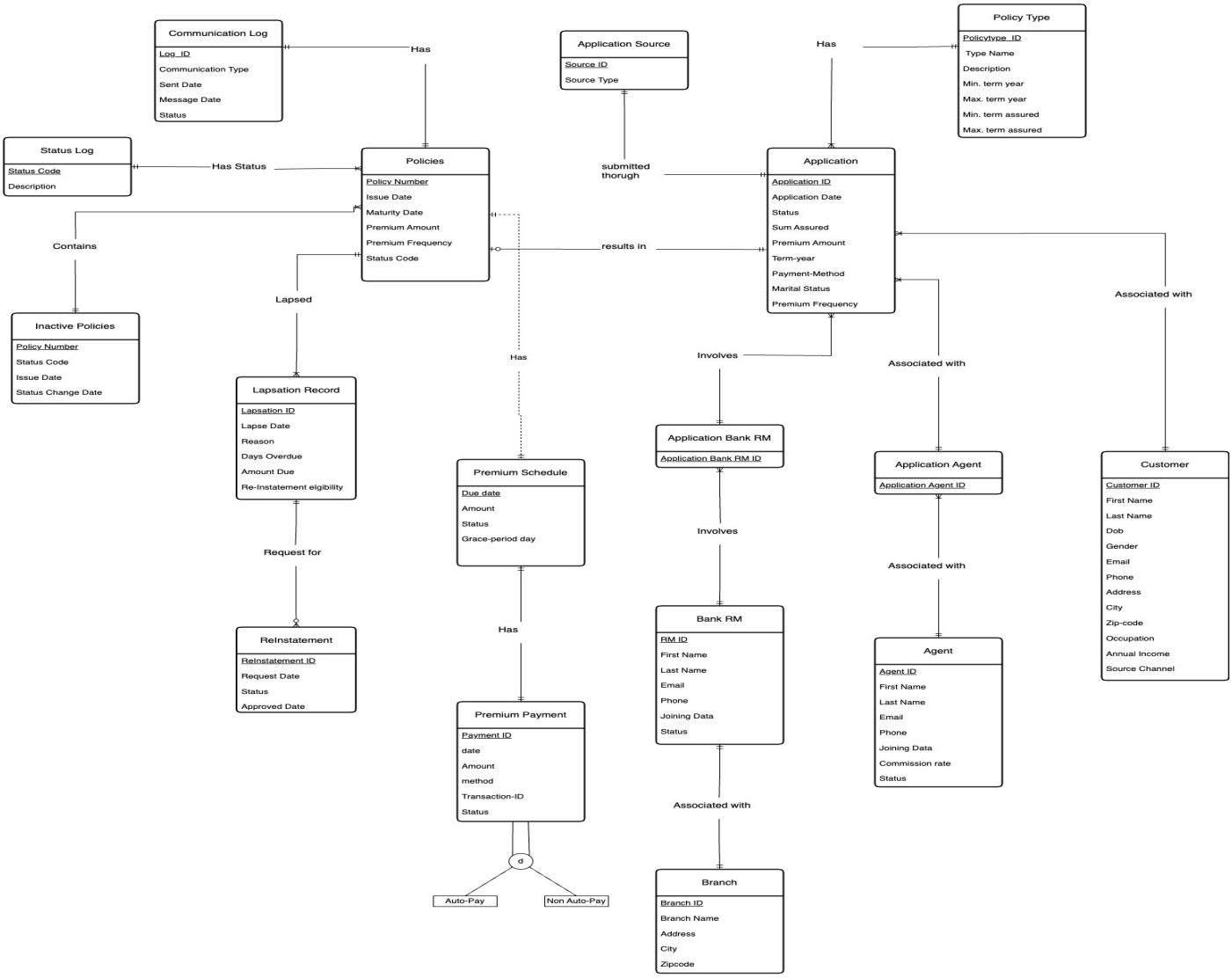
1. **Customer**
   - **Primary Key:** Customer_ID
   - **Attributes:** First_Name, Last_Name, Gender, DOB, Email, Phone, Address, City, Zip_Code, Occupation, Annual_Income, Created_At
   - The central entity that stores customer data relevant to applications and policies.
2. **Agent**
   - **Primary Key:** Agent_ID
   - **Attributes:** First_Name, Last_Name, Email, Phone, Joining_Date, Status, Commission_Rate
   - Represents agents who assist customers with policy applications and management.
3. **Branch**
   - **Primary Key:** Branch_ID
   - **Attributes:** Branch_Name, Address, City, Zip_Code
   - Tracks insurance company branch locations associated with agents and relationship managers.
4. **Bank Relationship Manager (RM)**
   - **Primary Key:** RM_ID
   - **Attributes:** First_Name, Last_Name, Email, Phone, Joining_Date, Status, Branch_ID (FK)
   - Managers who oversee policy applications and payments through bank channels.
5. **Policy Type**
   - **Primary Key:** Policytype_ID
   - **Attributes:** Type_Name, Description, Min_Term_Year, Max_Term_Year, Min_Sum_Assured, Max_Sum_Assured
   - Defines the various types of insurance policies (e.g., Term Life, Whole Life).
6. **Application**
   - **Primary Key:** Application_ID
   - **Foreign Keys:** Customer_ID (FK), Policytype_ID (FK)
   - **Attributes:** Application_Date, Status, Sum_Assured, Premium_Amount, Premium_Frequency, Term_Years, Payment_Method, Marital_Status
   - Captures details of customer applications for insurance policies.
7. **Application Source**
   - **Primary Key:** Source_ID
   - **Foreign Key:** Application_ID (FK)
   - **Attributes:** Source_Type

- Tracks the channel through which the application was submitted (e.g., Web, Agent, or Bank).

8. **Application Agent**
   - **Primary Key:** App_Agent_ID
   - **Foreign Keys:** Application_ID (FK), Agent_ID (FK)
   - **Attributes:** None
   - Associates applications with the agents responsible for processing them.
9. **Application Bank RM**
   - **Primary Key:** App_BankRM_ID
   - **Foreign Keys:** Application_ID (FK), RM_ID (FK)
   - **Attributes:** None
   - Associates applications with bank relationship managers.
10. **Status Log**
    - **Primary Key:** Status_Code
    - **Attributes:** Description
    - Lists different statuses that a policy can have, such as Active, Lapsed, or Withdrawn.
11. **Policy**
    - **Primary Key:** Policy_Number
    - **Foreign Keys:** Application_ID (FK), Status_Code (FK)
    - **Attributes:** Issue_Date, Maturity_Date, Premium_Amount
    - Represents the policies issued after applications are approved.
12. **Inactive Policies**
    - **Primary Key:** Inactive_ID
    - **Foreign Keys:** Policy_Number (FK), Status_Code (FK)
    - **Attributes:** Status_Change_Date
    - Tracks policies that have become inactive due to lapsation or withdrawal.
13. **Premium Payment**
    - **Primary Key:** Transaction_ID
    - **Foreign Key:** Policy_Number (FK)
    - **Attributes:** Payment_Date, Amount, Status
    - Contains records of premium payments made on policies.
14. **Premium Schedule**
    - **Attributes:** Policy_Number (FK), Due_Date, Amount, Status, Grace_Period_Days
    - Defines the schedule of premium payments for each policy.
    - This is a weak entity that depends on the **Policies** entity. It has a composite key composed of **Policy_Number** (foreign key) and **Due_Date**, representing each premium payment schedule for a specific policy.
15. **Communication Log**
    - **Primary Key:** Log_ID
    - **Foreign Key:** Policy_Number (FK)
    - **Attributes:** Communication_Type, Sent_Date, Message_Type
    - Logs reminders and warnings sent to customers regarding premiums.
16. **Lapsation Record**
    - **Primary Key:** Lapsation_ID
    - **Foreign Key:** Policy_Number (FK)

- **Attributes:** Lapsation_Date, Reason, Days_Overdue, Amount_Due, Reinstatement_Eligibility
- Tracks details of policies that have lapsed.

17. **Reinstatement**
    - **Primary Key:** Reinstatement_ID
    - **Foreign Key:** Lapsation_ID (FK)
    - **Attributes:** Request_Date, Status, Approved_Date
    - Captures requests to reinstate lapsed policies.



Relationships and Cardinalities

The relationships between entities and their cardinalities are defined as follows:

1. **Customer ↔ Application:**
    - One customer can submit multiple applications. ***(1:M)***
    - Each application is linked to one customer. ***(M:1)***
2. **Bank RM ↔ Branch:**
    - Each bank RM is assigned to exactly one branch. ***(1:1)***
    - A branch has exactly one bank RM associated with it. ***(1:1)***
3. **Application ↔ Policy Type:**
    - Each application is tied to one policy type. ***(1:1)***
    - A policy type can be referenced by multiple applications. ***(1:M)***
4. **Application ↔ Policies:**
    - An application may not always result in a policy, but each policy is tied to exactly one application. ***(0:1)***
    - A policy is always linked to one approved application. ***(1:1)***
5. **Policies ↔ Status Code:**
    - Each policy has exactly one status code. ***(1:1)***
    - A single status code can apply to multiple policies. ***(1:M)***
6. **Policies ↔ Inactive Policies:**
    - A policy may not become inactive, but if it does, there can be multiple inactive entries for policies. ***(0:M)***
    - Each inactive policy record is tied to a specific policy. ***(1:1)***
7. **Policies ↔ Premium Schedule:**
    - Each policy has exactly one premium schedule. ***(1:1)***
    - A premium schedule is linked to one specific policy. ***(1:1)***
8. **Premium Schedule ↔ Premium Payment:**
    - Each premium schedule has one corresponding premium payment. ***(1:1)***
    - Each premium payment is tied to one premium schedule. ***(1:1)***
9. **Policies ↔ Communication Log:**
    - A policy can have exactly one communication log. ***(1:1)***
    - Each communication log entry corresponds to one policy. ***(1:1)***
10. **Policies ↔ Lapsation Record:**
    - One policy can have multiple lapsation records. ***(0:M)***
    - Each lapsation record corresponds to one policy. ***(1:1)***

11. **Lapsation Record ↔ Reinstatement:**
    - A lapsation may not have any reinstatement requests, but if it does, there can be multiple requests. ***(0:M)***
    - Each reinstatement request is linked to a specific lapsation record. ***(1:1)***
12. **Application ↔ Application Source:**
    - **1:1:** Each application is submitted through one application source. *(1:1)*
    - **1:1:** Each application source entry is tied to one application. *(1:1)*
13. **Application ↔ Application Bank RM:**
    - **1:1:** An application can have one application bank RM entry. *(1:1)*
    - **1:M:** An application bank RM can be linked to multiple applications. *(1:M)*
14. **Application Bank RM ↔ Bank RM:**
    - **1:1:** Each application bank RM entry is linked to one bank RM. *(1:1)*
    - **1:M:** A bank RM can be linked to multiple application bank RM entries. *(1:M)*
15. **Application ↔ Application Agent:**
    - **1:1:** Each application is linked to one application agent entry. *(1:1)*
    - **1:M:** An application agent can handle multiple applications. *(1:M)*
16. **Application Agent ↔ Agent:**
    - **1:1:** Each application agent entry is tied to one agent. *(1:1)*
    - **1:M:** An agent can handle multiple application agent entries. *(1:M)*

Assumptions Made:

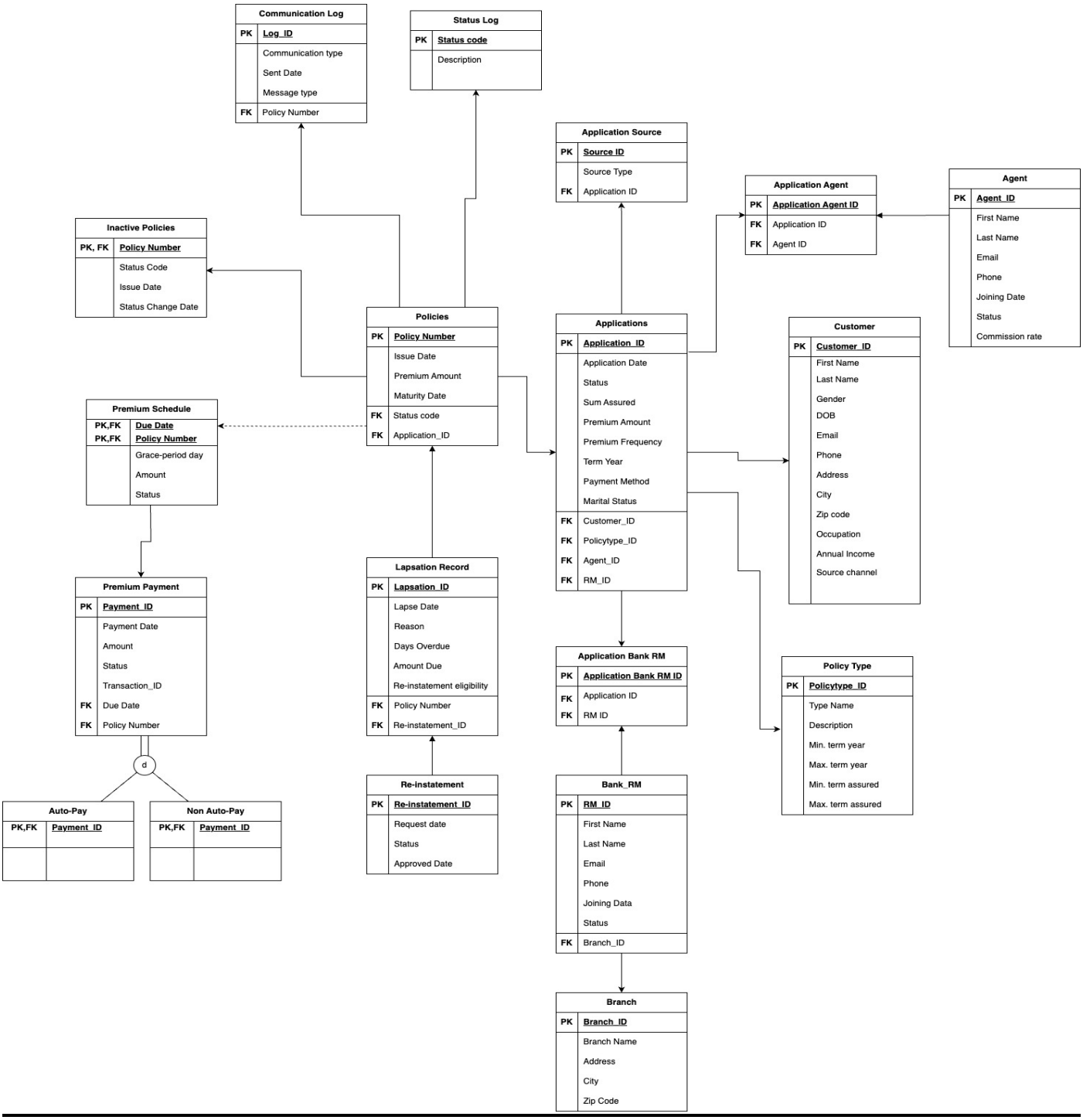
1. We assume that an agent's commission rate may vary depending on the application they handle.
2. There may be agents who have not yet filed any applications. These agents can either be active or inactive.
3. The commission rate for a particular agent is constant. For example, if an agent charges a 1% commission, that rate will apply to all applications they handle.
4. The same rule applies to bank relationship managers (RMs), where their commission rate remains consistent across all applications.
5. Each bank branch is assigned a single RM (one-to-one relationship).
6. A **Policy Number** is generated when an application is approved.
7. Policy reminders are sent only once, and further reminders are not scheduled.
8. In both **Premium Schedule** and **Premium Payment**, we assume that there are no partial payments. Each premium payment aligns fully with the premium schedule, meaning there is a minimum of one and a maximum of one payment per schedule entry.
9. Due dates for premium payments may change each month or year. As such, no historical record of previous due dates is maintained.

Relational Model

Bold -> Primary Key; *Italic -> Foreign Key*.

- **Customer** (**Customer_ID**, First_Name, Last_Name, Gender, DOB, Email, Phone, Address, City, Zip_Code, Occupation, Annual_Income, Created_At)
- **Agent** (**Agent_ID**, First_Name, Last_Name, Email, Phone, Joining_Date, Status, Comission_Rate)
- **Branch** (**Branch_ID**, Branch_Name, Address, City, Zip_Code)
- **Bank Relationship Manager (RM)** (**RM_ID**, First_Name, Last_Name, Email, Phone, Joining_Date, Status, *Branch_ID*)
- **Policy Type** (**Policytype_ID**, Type_Name, Description, Min_Term_Year, Max_Term_Year, Min_Sum_Assured, Max_Sum_Assured)
- **Application** (**Application_ID**, Application_Date, Status, Sum_Assured, Premium_Amount, Premium_Frequency, Term_Year, Payment_Method, Marital_Status, *Customer_ID*, *Policytype_ID*)
- **Application Source** (**Source_ID**, Source_Type, *Application_ID*)
- **Application Agent** (**Application_Agent_ID**, *Application_ID*, *Agent_ID*)
- **Application Bank RM** (**Application_BankRM_ID**, *Application_ID*, *RM_ID*)
- **Status Log** (**Status_Code**, Description)
- **Policy** (**Policy_Number**, Issue_Date, Maturity_Date, Premium_Amount, *Application_ID*, *Status_Code*)
- **Inactive Policies** (**Inactive_ID**, Status_Change_Date, *Policy_Number*, *Status_Code*)
- **Premium Payment** (**Transaction_ID**, Payment_Date, Amount, Status, *Policy_Number*)
- **Premium Schedule** (**(*Policy_Number, Due_Date*)**, Amount, Status, Grace_Period_Days)
- **Communication Log** (**Log_ID**, Comunication_Type, Sent_Date, Message_Type, *Policy_Number*)
- **Lapsation Record** (**Lapsation_ID**, Lapsation_Date, Reason, Days_Overdue, Amount_Due, Reinstatement_Eligibility, *Policy_Number*)
- **Reinstatement** (**Reinstatement_ID**, Request_Date, Status, Approved_Date, *Lapsation_ID*)

Relational Schema:



We noticed that a few tables in our lifeinsurance database contain attributes that can change over time, such as customer details, policy premiums, and lapsation records. To accommodate these changes and maintain historical data, we have implemented **Slowly Changing Dimensions - Type 2**.

This approach ensures that whenever an update occurs, such as changes in a customer's address or annual income, a new record is inserted with updated details, while the old record is retained with an End_Date and an Is_Current flag.

The factors that could change and require tracking include customer contact details, policy premium amounts, and lapsation statuses. Implementing SCD Type 2 allows us to analyze changes over time, providing valuable insights

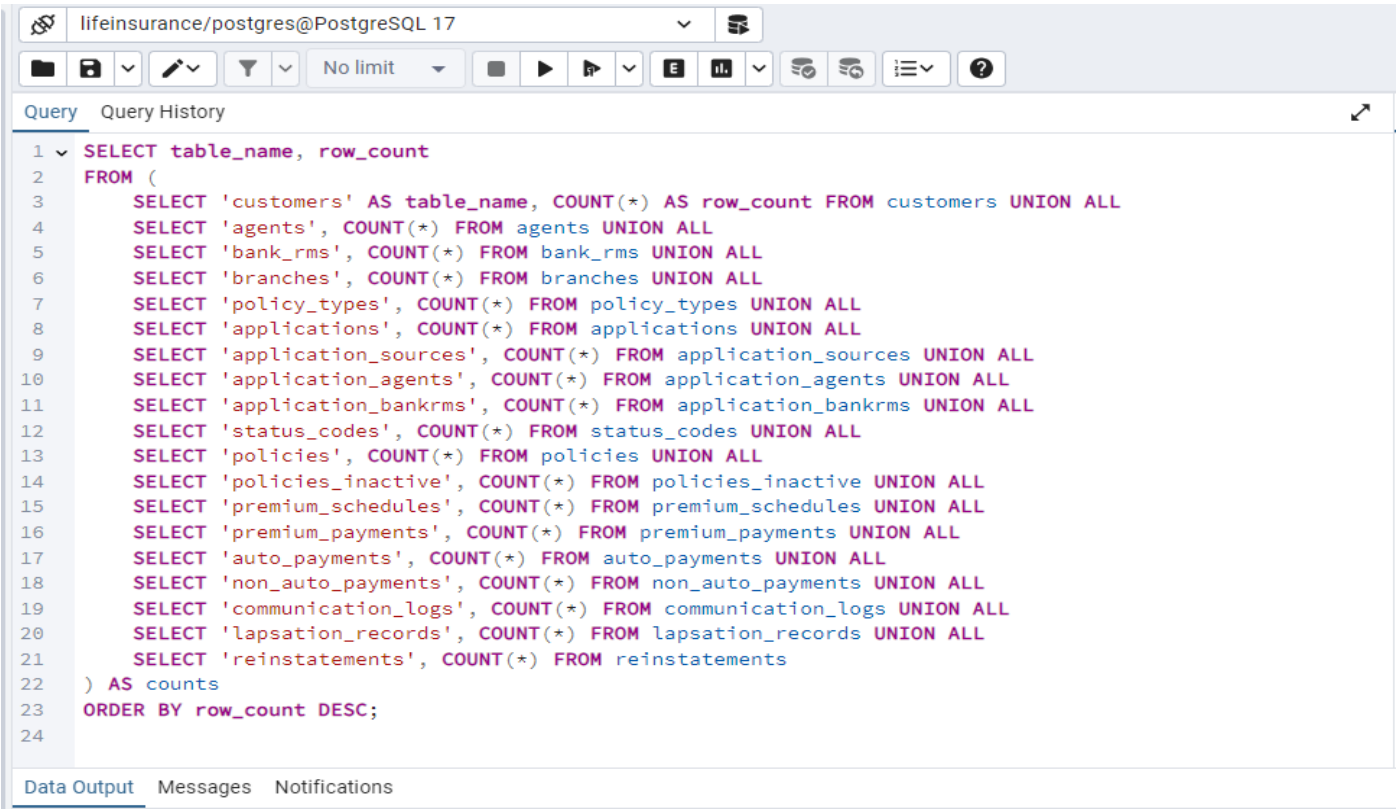
Database Implementation:

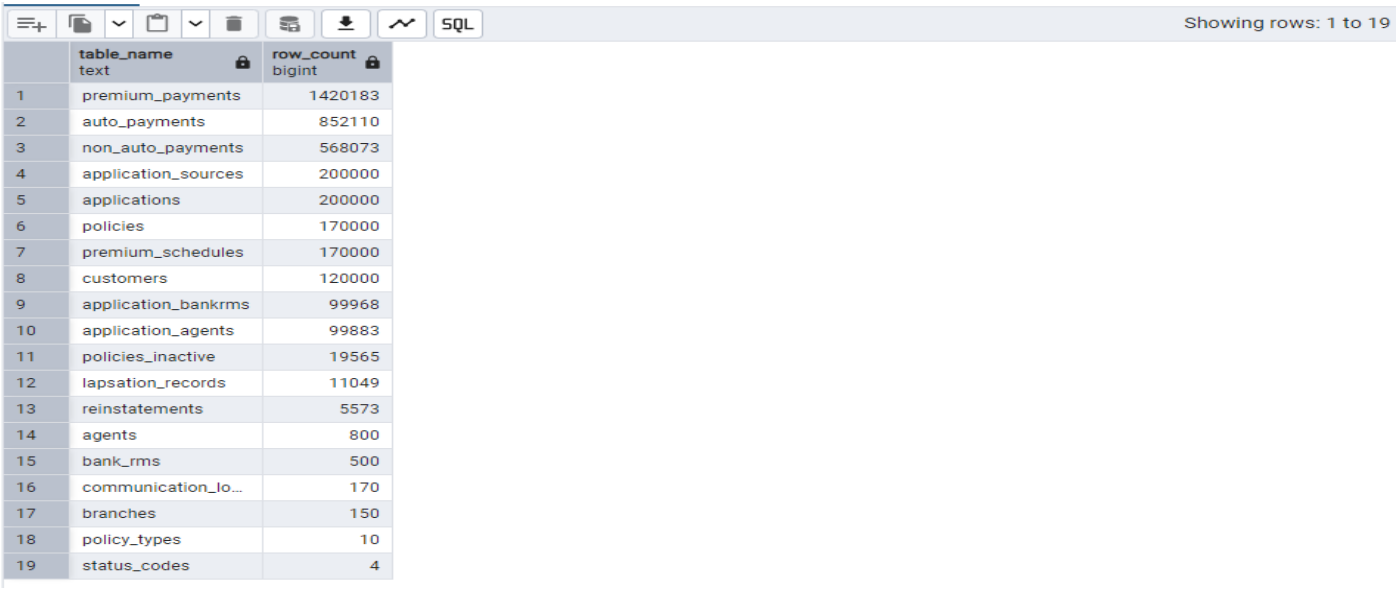
Source:

We used the **Faker** library in Python to generate sample data for the database. The data was exported to **CSV** files, which were then imported into the **PostgreSQL** database. This allowed us to test and validate the database structure, queries, and relationships.

Tables:

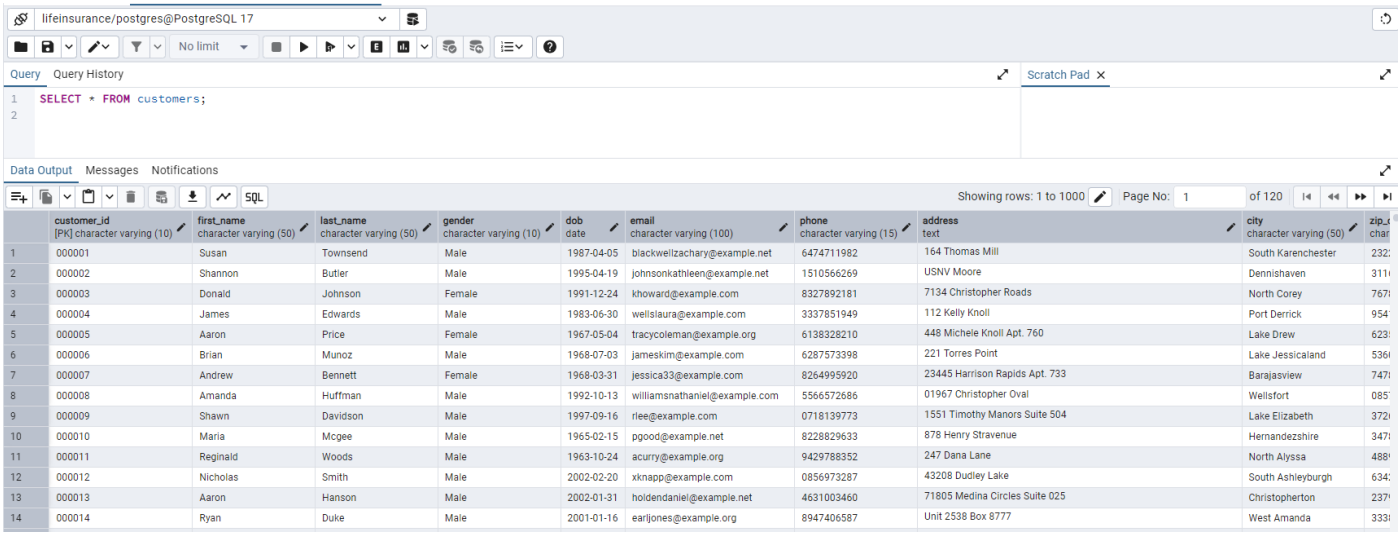
1. **All Tables**: This query retrieves the number of records in each table, displaying results in descending order of volume.





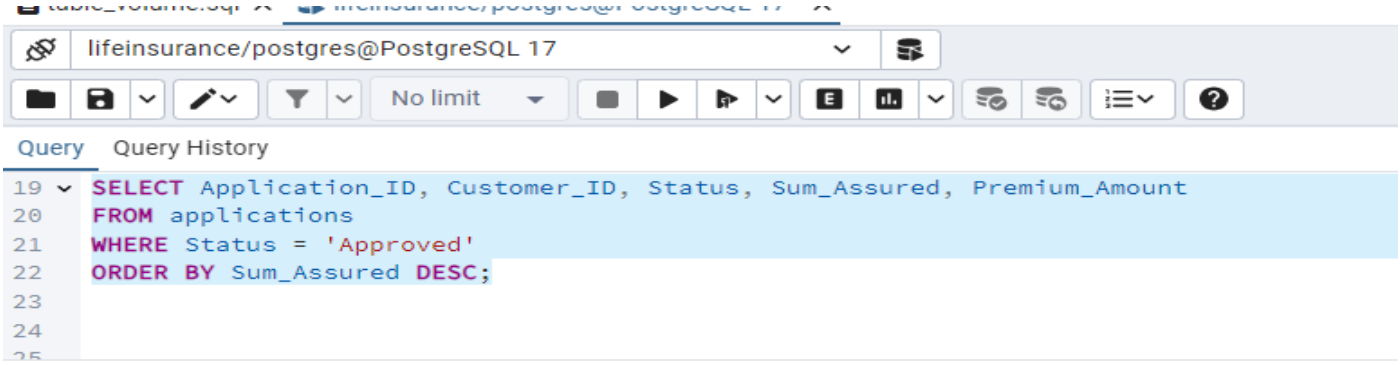
	table_name text	row_count bigint
1	premium_payments	1420183
2	auto_payments	852110
3	non_auto_payments	568073
4	application_sources	200000
5	applications	200000
6	policies	170000
7	premium_schedules	170000
8	customers	120000
9	application_bankrms	99968
10	application_agents	99883
11	policies_inactive	19565
12	lapsation_records	11049
13	reinstatements	5573
14	agents	800
15	bank_rms	500
16	communication_lo...	170
17	branches	150
18	policy_types	10
19	status_codes	4

2. **Customers:** This query fetches random customers from the customers table.



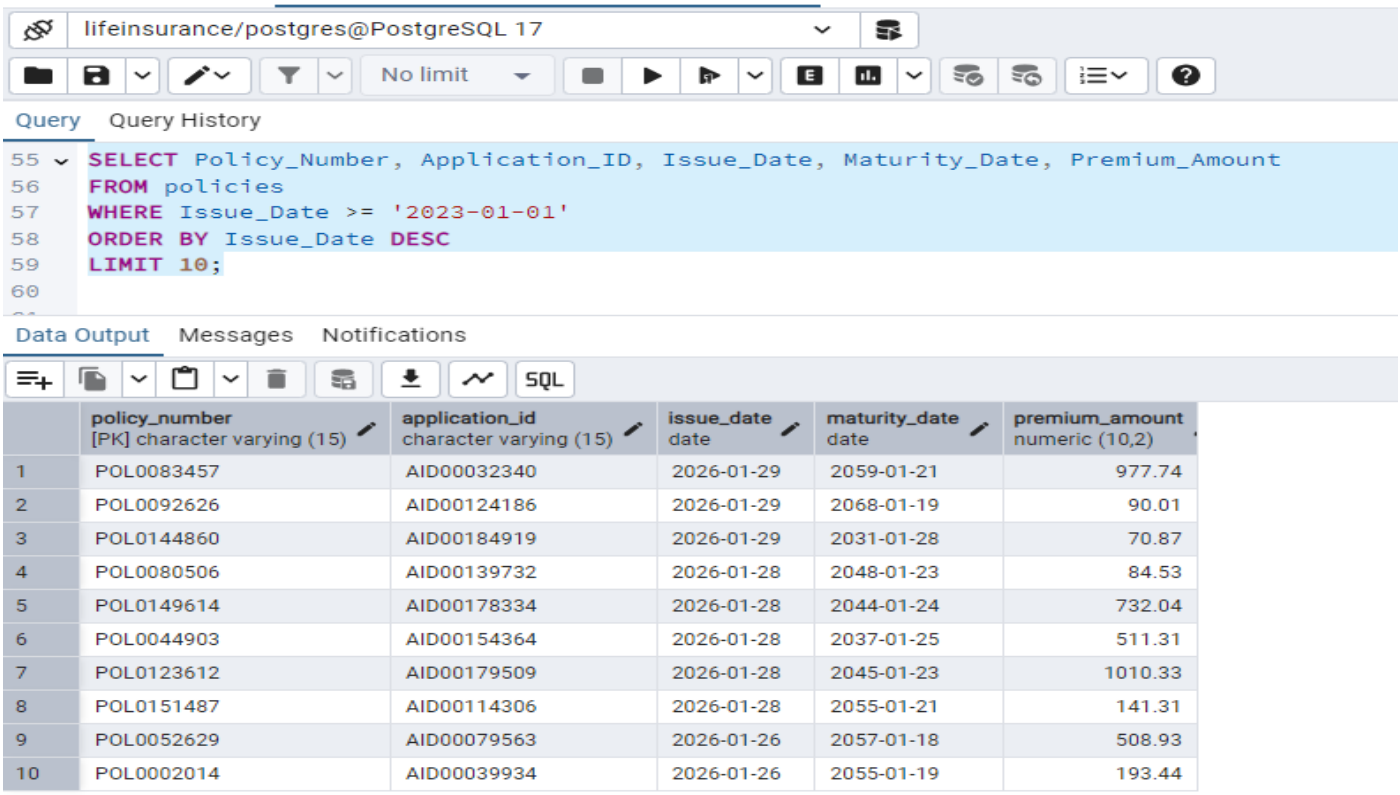
	customer_id [PK] character varying (10)	first_name character varying (50)	last_name character varying (50)	gender character varying (10)	dob date	email character varying (100)	phone character varying (15)	address text	city character varying (50)	zip_c char
1	000001	Susan	Townsend	Male	1987-04-05	blackwellzachary@example.net	6474711982	164 Thomas Mill	South Karenchester	232
2	000002	Shannon	Butler	Male	1995-04-19	johnsonkathleen@example.net	1510566269	USNV Moore	Dennishaven	311
3	000003	Donald	Johnson	Female	1991-12-24	khoward@example.com	8327892181	7134 Christopher Roads	North Corey	767
4	000004	James	Edwards	Male	1983-06-30	wellslaura@example.com	3337851949	112 Kelly Knoll	Port Derrick	954
5	000005	Aaron	Price	Female	1967-05-04	tracycoleman@example.org	6138328210	448 Michele Knoll Apt. 760	Lake Drew	623
6	000006	Brian	Munoz	Male	1968-07-03	jameskim@example.com	6287573398	221 Torres Point	Lake Jessicaland	536
7	000007	Andrew	Bennett	Female	1968-03-31	jessica33@example.com	8264995920	23445 Harrison Rapids Apt. 733	Barajasview	747
8	000008	Amanda	Huffman	Male	1992-10-13	williamsnathaniel@example.com	5566572686	01967 Christopher Oval	Wellsfort	085
9	000009	Shawn	Davidson	Male	1997-09-16	rlee@example.com	0718139773	1551 Timothy Manors Suite 504	Lake Elizabeth	372
10	000010	Maria	Mcgee	Male	1965-02-15	pgood@example.net	8228829633	878 Henry Stravenue	Hernandezshire	347
11	000011	Reginald	Woods	Male	1963-10-24	acurry@example.org	9429788352	247 Dana Lane	North Alyssa	488
12	000012	Nicholas	Smith	Male	2002-02-20	xknapp@example.com	0856973287	43208 Dudley Lake	South Ashleyburgh	634
13	000013	Aaron	Hanson	Male	2002-01-31	holdendaniel@example.net	4631003460	71805 Medina Circles Suite 025	Christopherton	237
14	000014	Ryan	Duke	Male	2001-01-16	earljones@example.org	8947406587	Unit 2538 Box 8777	West Amanda	333

3. **Applications:** Retrieves all applications where the status is "Approved".



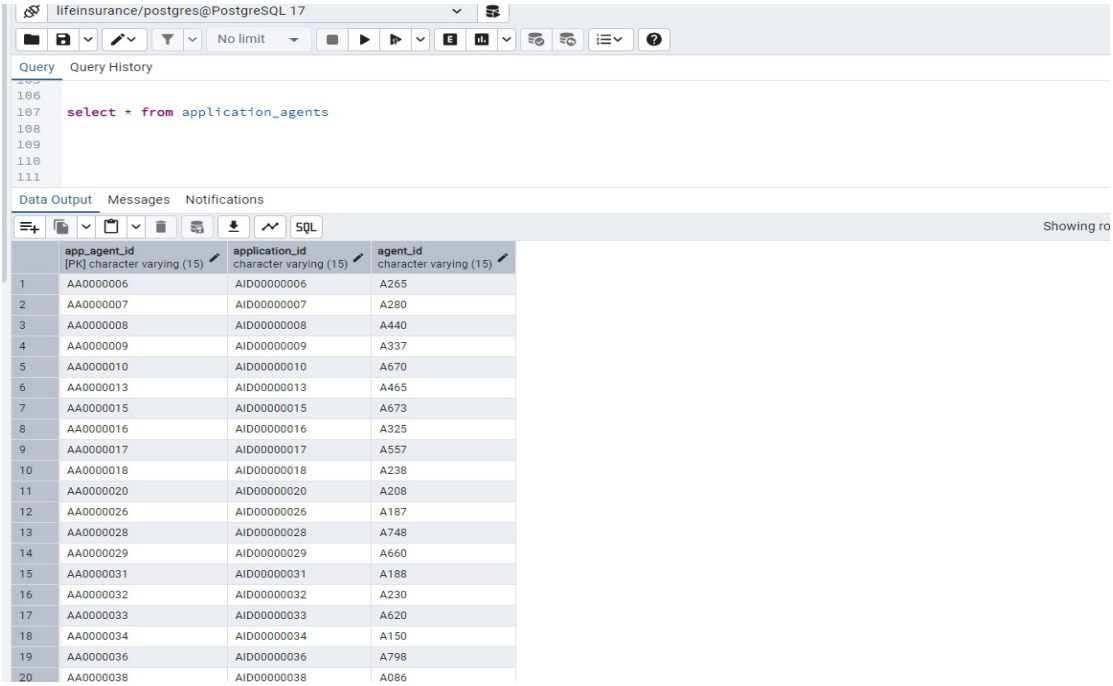
	application_id [PK] character varying (15)	customer_id character varying (10)	status character varying (20)	sum_assured numeric (12,2)	premium_amount numeric (10,2)
1	AID00107752	013368	Approved	999994.40	2666.65
2	AID00079292	046290	Approved	999945.59	465.09
3	AID00000122	018836	Approved	999941.06	555.52
4	AID00082404	074758	Approved	999937.46	689.61
5	AID00026945	054848	Approved	999900.47	999.90
6	AID00076983	109795	Approved	999892.19	107.52
7	AID00074972	110220	Approved	999886.12	3635.95
8	AID00000069	115456	Approved	999855.06	1599.77
9	AID00163256	018574	Approved	999807.66	1290.07
10	AID00093985	077725	Approved	999799.86	270.22
11	AID00090660	014659	Approved	999790.65	605.93
12	AID00156677	079606	Approved	999787.53	1025.42

4. **Policies:** Fetches the 10 newest policies issued after January 1, 2023, sorted by the most recent issue date.



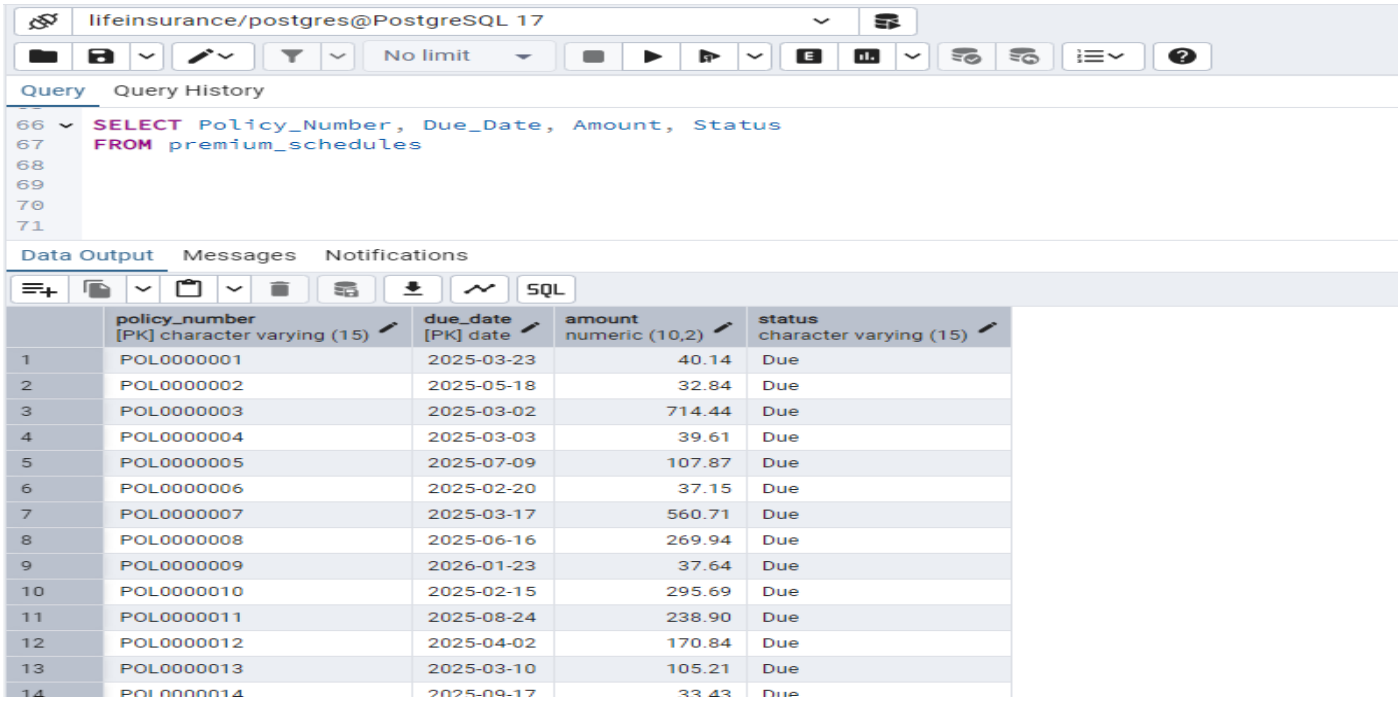
	policy_number [PK] character varying (15)	application_id character varying (15)	issue_date date	maturity_date date	premium_amount numeric (10,2)
1	POL0083457	AID00032340	2026-01-29	2059-01-21	977.74
2	POL0092626	AID00124186	2026-01-29	2068-01-19	90.01
3	POL0144860	AID00184919	2026-01-29	2031-01-28	70.87
4	POL0080506	AID00139732	2026-01-28	2048-01-23	84.53
5	POL0149614	AID00178334	2026-01-28	2044-01-24	732.04
6	POL0044903	AID00154364	2026-01-28	2037-01-25	511.31
7	POL0123612	AID00179509	2026-01-28	2045-01-23	1010.33
8	POL0151487	AID00114306	2026-01-28	2055-01-21	141.31
9	POL0052629	AID00079563	2026-01-26	2057-01-18	508.93
10	POL0002014	AID00039934	2026-01-26	2055-01-19	193.44

5. **Application Agents:** Fetches all entries from the application-agent table.



	app_agent_id [PK] character varying (15)	application_id character varying (15)	agent_id character varying (15)
1	AA0000006	AID00000006	A265
2	AA0000007	AID00000007	A280
3	AA0000008	AID00000008	A440
4	AA0000009	AID00000009	A337
5	AA0000010	AID00000010	A670
6	AA0000013	AID00000013	A465
7	AA0000015	AID00000015	A673
8	AA0000016	AID00000016	A325
9	AA0000017	AID00000017	A557
10	AA0000018	AID00000018	A238
11	AA0000020	AID00000020	A208
12	AA0000026	AID00000026	A187
13	AA0000028	AID00000028	A748
14	AA0000029	AID00000029	A660
15	AA0000031	AID00000031	A188
16	AA0000032	AID00000032	A230
17	AA0000033	AID00000033	A620
18	AA0000034	AID00000034	A150
19	AA0000036	AID00000036	A798
20	AA0000038	AID00000038	A086

6. **Premium Schedules:** Displays the premium payment schedule, including policy number, due date, amount, and status.



lifeinsurance/postgres@PostgreSQL 17

No limit

Query | Query History

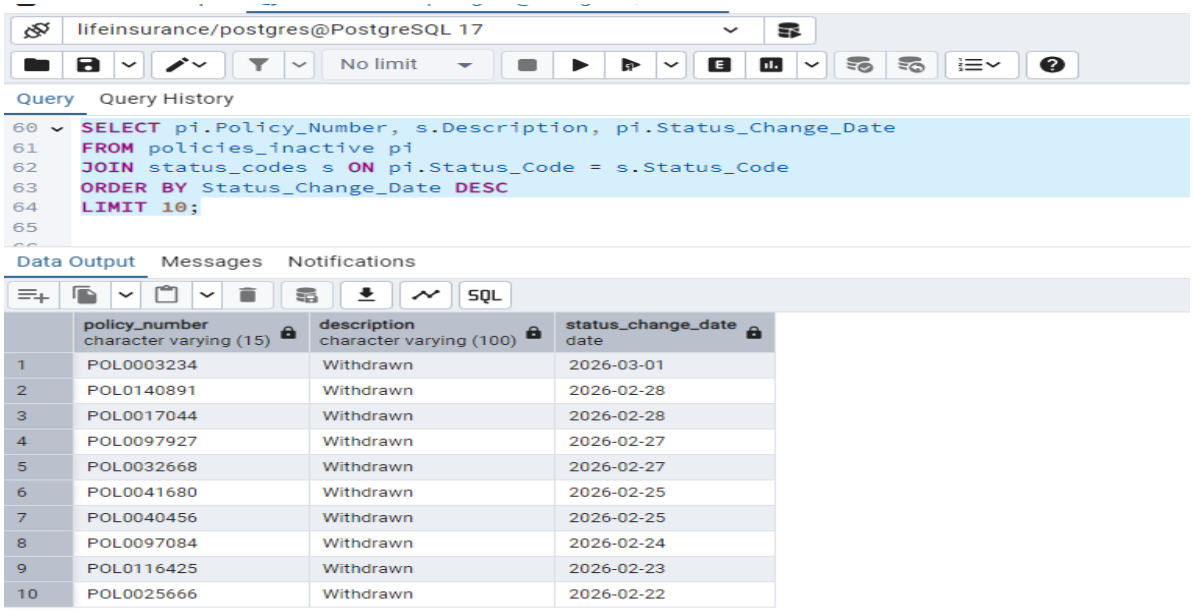
```
66  SELECT Policy_Number, Due_Date, Amount, Status
67  FROM premium_schedules
68
69
70
71
```

Data Output | Messages | Notifications

SQL

	policy_number [PK] character varying (15)	due_date [PK] date	amount numeric (10,2)	status character varying (15)
1	POL0000001	2025-03-23	40.14	Due
2	POL0000002	2025-05-18	32.84	Due
3	POL0000003	2025-03-02	714.44	Due
4	POL0000004	2025-03-03	39.61	Due
5	POL0000005	2025-07-09	107.87	Due
6	POL0000006	2025-02-20	37.15	Due
7	POL0000007	2025-03-17	560.71	Due
8	POL0000008	2025-06-16	269.94	Due
9	POL0000009	2026-01-23	37.64	Due
10	POL0000010	2025-02-15	295.69	Due
11	POL0000011	2025-08-24	238.90	Due
12	POL0000012	2025-04-02	170.84	Due
13	POL0000013	2025-03-10	105.21	Due
14	POL0000014	2025-09-17	33.43	Due

7. **Inactive Policies:** Lists the 10 most recent inactive policies with their status and change date.



lifeinsurance/postgres@PostgreSQL 17

No limit

Query | Query History

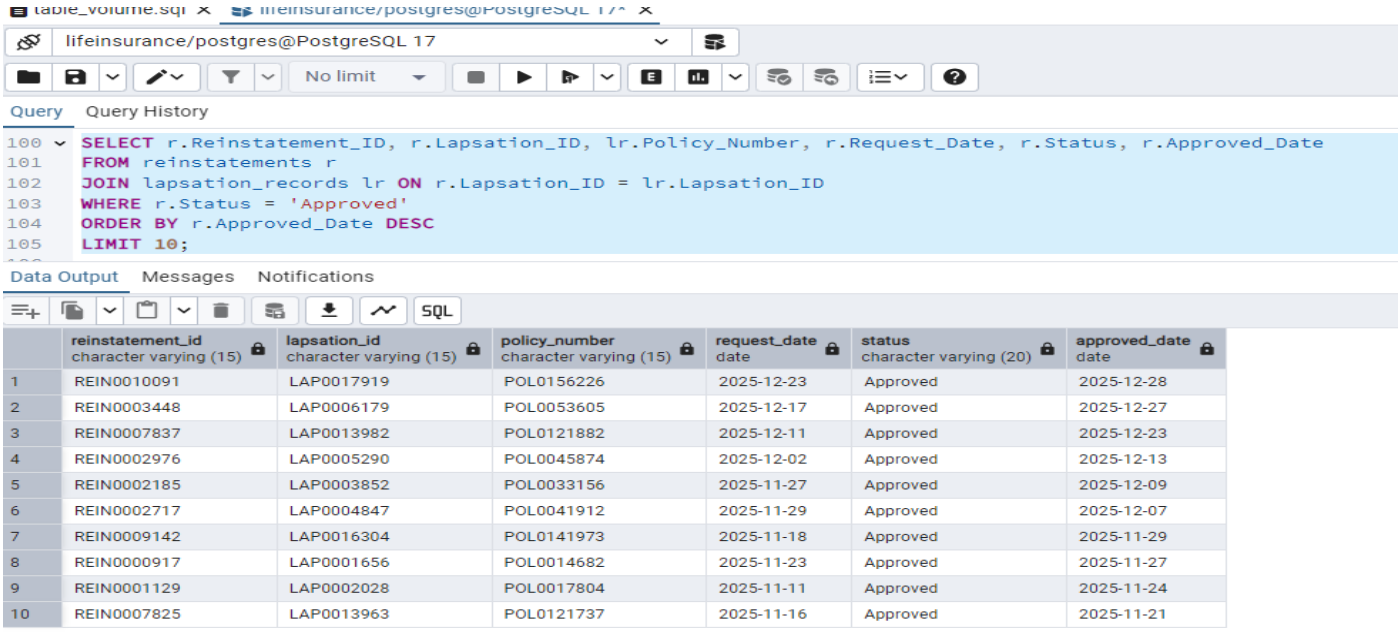
```
60  SELECT pi.Policy_Number, s.Description, pi.Status_Change_Date
61  FROM policies_inactive pi
62  JOIN status_codes s ON pi.Status_Code = s.Status_Code
63  ORDER BY Status_Change_Date DESC
64  LIMIT 10;
65
```

Data Output | Messages | Notifications

SQL

	policy_number character varying (15)	description character varying (100)	status_change_date date
1	POL0003234	Withdrawn	2026-03-01
2	POL0140891	Withdrawn	2026-02-28
3	POL0017044	Withdrawn	2026-02-28
4	POL0097927	Withdrawn	2026-02-27
5	POL0032668	Withdrawn	2026-02-27
6	POL0041680	Withdrawn	2026-02-25
7	POL0040456	Withdrawn	2026-02-25
8	POL0097084	Withdrawn	2026-02-24
9	POL0116425	Withdrawn	2026-02-23
10	POL0025666	Withdrawn	2026-02-22

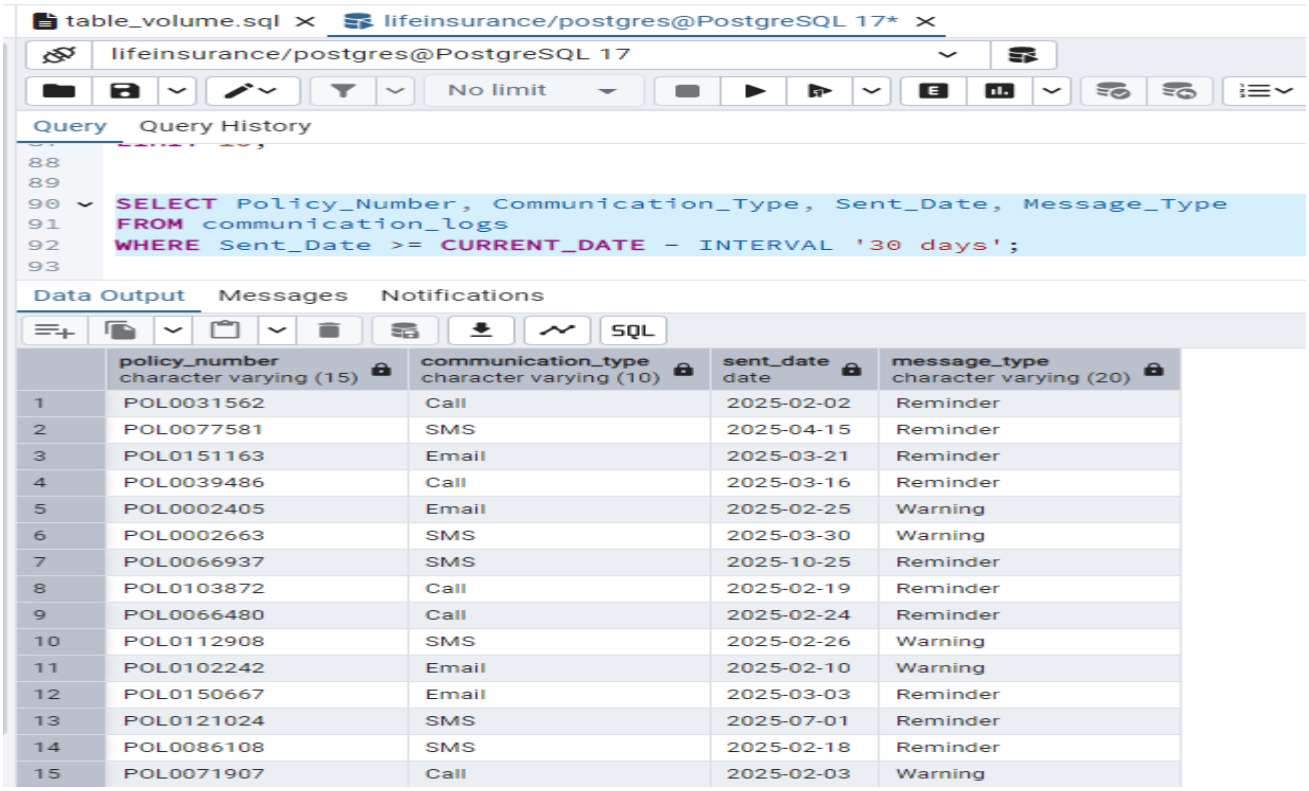
8. **Reinstatements:** Shows the 10 most recent approved reinstatements with their request and approval dates**.**



```
SELECT r.Reinstatement_ID, r.Lapsation_ID, lr.Policy_Number, r.Request_Date, r.Status, r.Approved_Date
FROM reinstatements r
JOIN lapsation_records lr ON r.Lapsation_ID = lr.Lapsation_ID
WHERE r.Status = 'Approved'
ORDER BY r.Approved_Date DESC
LIMIT 10;
```

	reinstatement_id character varying (15)	lapsation_id character varying (15)	policy_number character varying (15)	request_date date	status character varying (20)	approved_date date
1	REIN0010091	LAP0017919	POL0156226	2025-12-23	Approved	2025-12-28
2	REIN0003448	LAP0006179	POL0053605	2025-12-17	Approved	2025-12-27
3	REIN0007837	LAP0013982	POL0121882	2025-12-11	Approved	2025-12-23
4	REIN0002976	LAP0005290	POL0045874	2025-12-02	Approved	2025-12-13
5	REIN0002185	LAP0003852	POL0033156	2025-11-27	Approved	2025-12-09
6	REIN0002717	LAP0004847	POL0041912	2025-11-29	Approved	2025-12-07
7	REIN0009142	LAP0016304	POL0141973	2025-11-18	Approved	2025-11-29
8	REIN0000917	LAP0001656	POL0014682	2025-11-23	Approved	2025-11-27
9	REIN0001129	LAP0002028	POL0017804	2025-11-11	Approved	2025-11-24
10	REIN0007825	LAP0013963	POL0121737	2025-11-16	Approved	2025-11-21

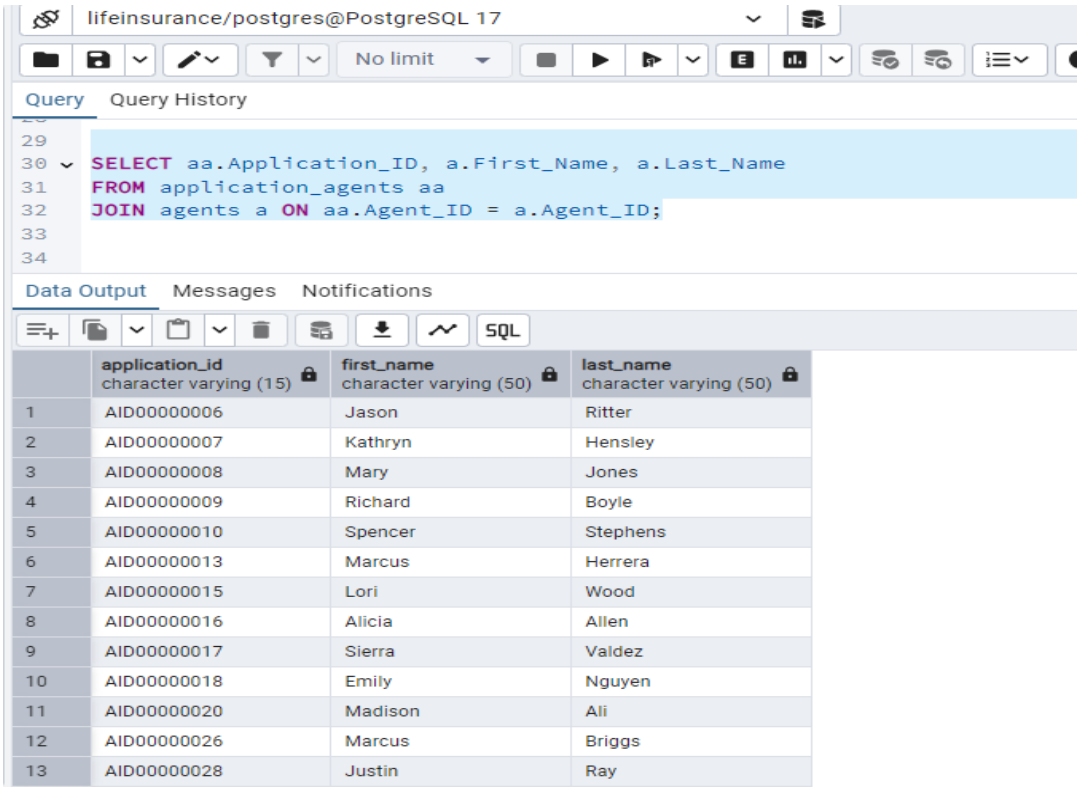
9. **Communication Logs:** Lists communication logs for policies with messages sent within the last 30 days.



```
SELECT Policy_Number, Communication_Type, Sent_Date, Message_Type
FROM communication_logs
WHERE Sent_Date >= CURRENT_DATE - INTERVAL '30 days';
```

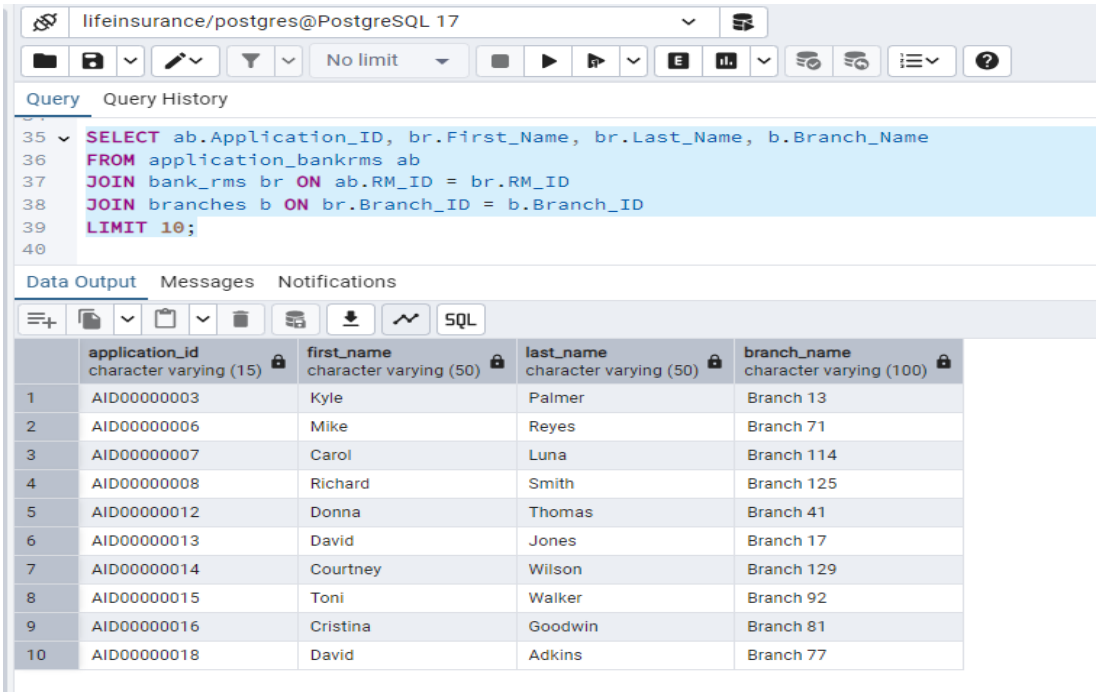
	policy_number character varying (15)	communication_type character varying (10)	sent_date date	message_type character varying (20)
1	POL0031562	Call	2025-02-02	Reminder
2	POL0077581	SMS	2025-04-15	Reminder
3	POL0151163	Email	2025-03-21	Reminder
4	POL0039486	Call	2025-03-16	Reminder
5	POL0002405	Email	2025-02-25	Warning
6	POL0002663	SMS	2025-03-30	Warning
7	POL0066937	SMS	2025-10-25	Reminder
8	POL0103872	Call	2025-02-19	Reminder
9	POL0066480	Call	2025-02-24	Reminder
10	POL0112908	SMS	2025-02-26	Warning
11	POL0102242	Email	2025-02-10	Warning
12	POL0150667	Email	2025-03-03	Reminder
13	POL0121024	SMS	2025-07-01	Reminder
14	POL0086108	SMS	2025-02-18	Reminder
15	POL0071907	Call	2025-02-03	Warning

10. **Application Agents :** Lists application IDs with associated agent names.



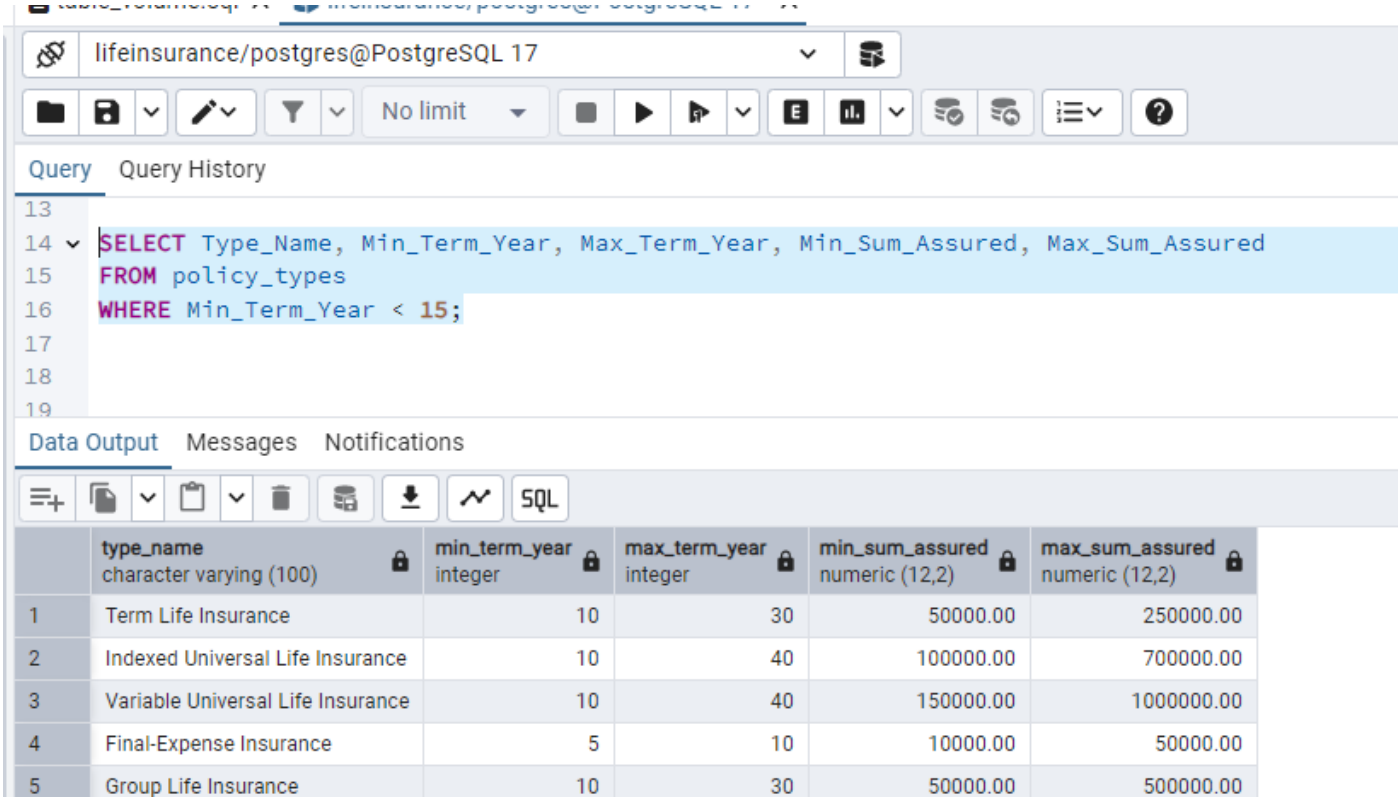
	application_id character varying (15)	first_name character varying (50)	last_name character varying (50)
1	AID00000006	Jason	Ritter
2	AID00000007	Kathryn	Hensley
3	AID00000008	Mary	Jones
4	AID00000009	Richard	Boyle
5	AID00000010	Spencer	Stephens
6	AID00000013	Marcus	Herrera
7	AID00000015	Lori	Wood
8	AID00000016	Alicia	Allen
9	AID00000017	Sierra	Valdez
10	AID00000018	Emily	Nguyen
11	AID00000020	Madison	Ali
12	AID00000026	Marcus	Briggs
13	AID00000028	Justin	Ray

11. **Bank RM, Branches:** Shows application IDs, bank relationship managers' names, and their associated branch names.



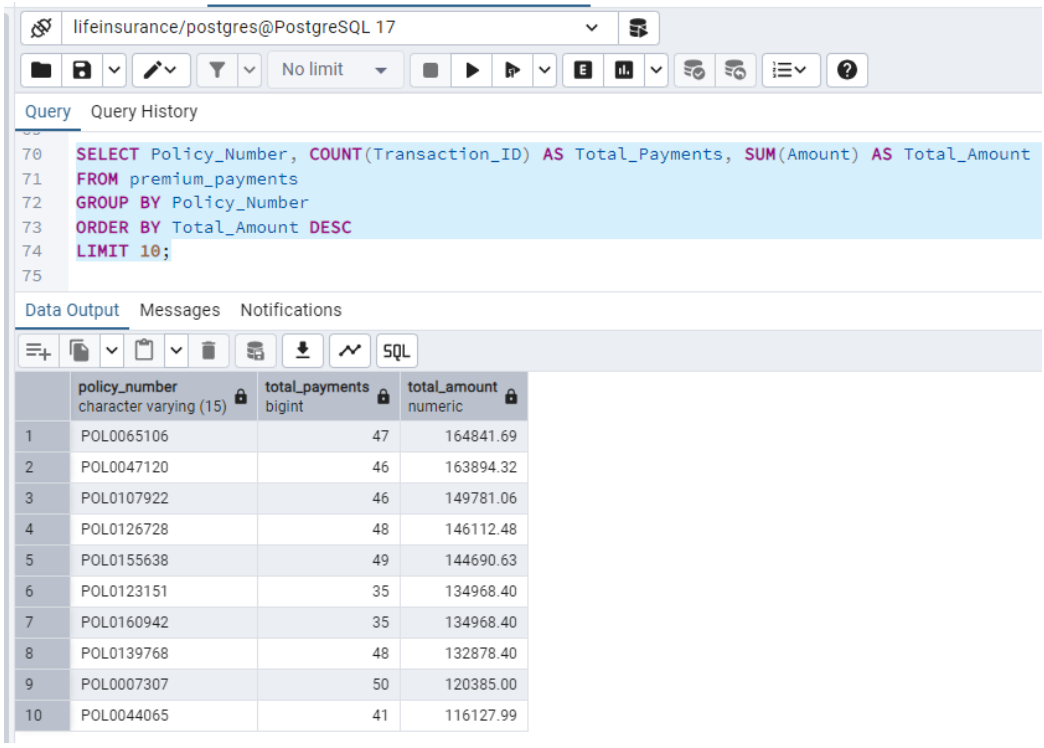
	application_id character varying (15)	first_name character varying (50)	last_name character varying (50)	branch_name character varying (100)
1	AID00000003	Kyle	Palmer	Branch 13
2	AID00000006	Mike	Reyes	Branch 71
3	AID00000007	Carol	Luna	Branch 114
4	AID00000008	Richard	Smith	Branch 125
5	AID00000012	Donna	Thomas	Branch 41
6	AID00000013	David	Jones	Branch 17
7	AID00000014	Courtney	Wilson	Branch 129
8	AID00000015	Toni	Walker	Branch 92
9	AID00000016	Cristina	Goodwin	Branch 81
10	AID00000018	David	Adkins	Branch 77

12. **Policy types :** Lists policy types with a minimum term year of less than 15.



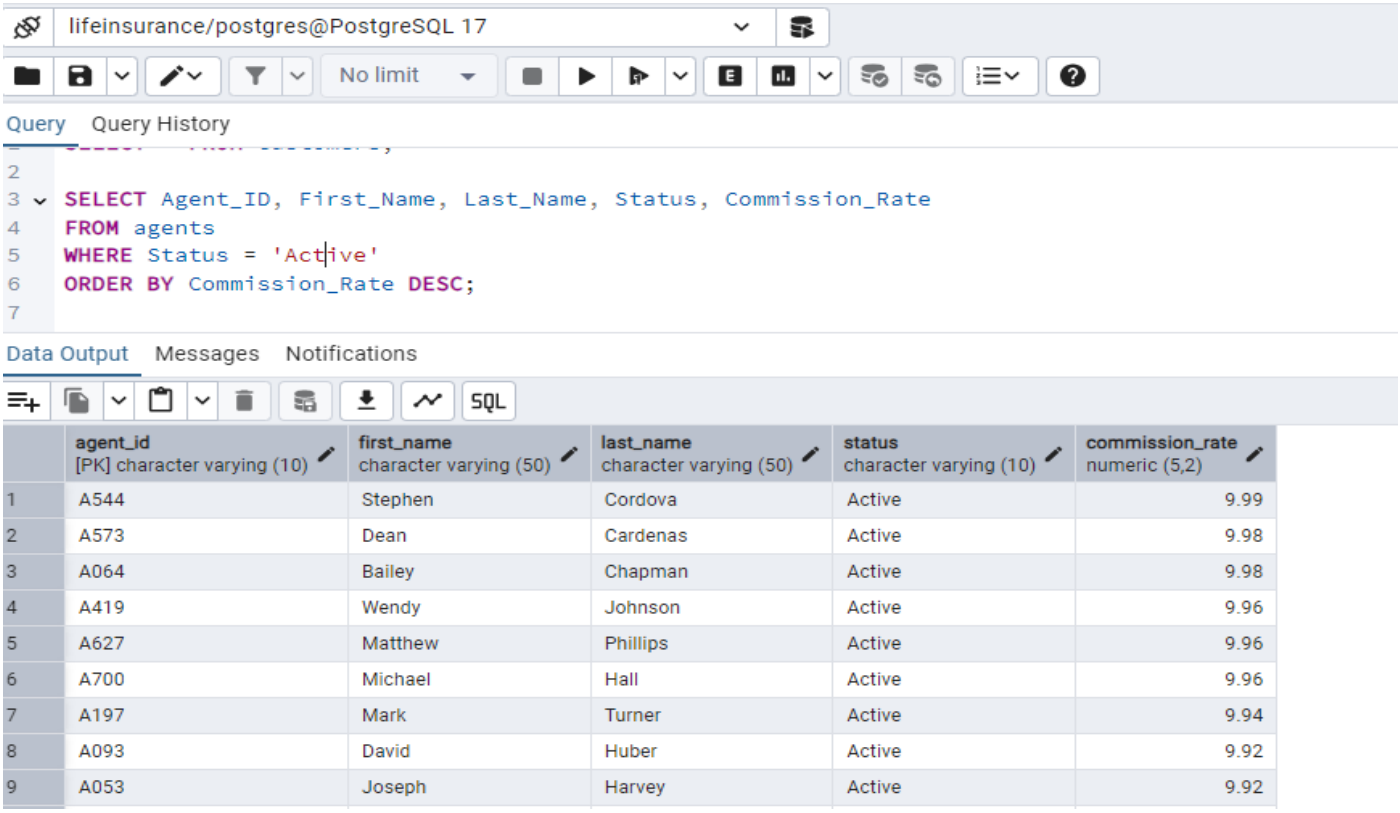
	type_name character varying (100)	min_term_year integer	max_term_year integer	min_sum_assured numeric (12,2)	max_sum_assured numeric (12,2)
1	Term Life Insurance	10	30	50000.00	250000.00
2	Indexed Universal Life Insurance	10	40	100000.00	700000.00
3	Variable Universal Life Insurance	10	40	150000.00	1000000.00
4	Final-Expense Insurance	5	10	10000.00	50000.00
5	Group Life Insurance	10	30	50000.00	500000.00

13. **Premium Payments:** Shows top 10 policies by total premium payments.



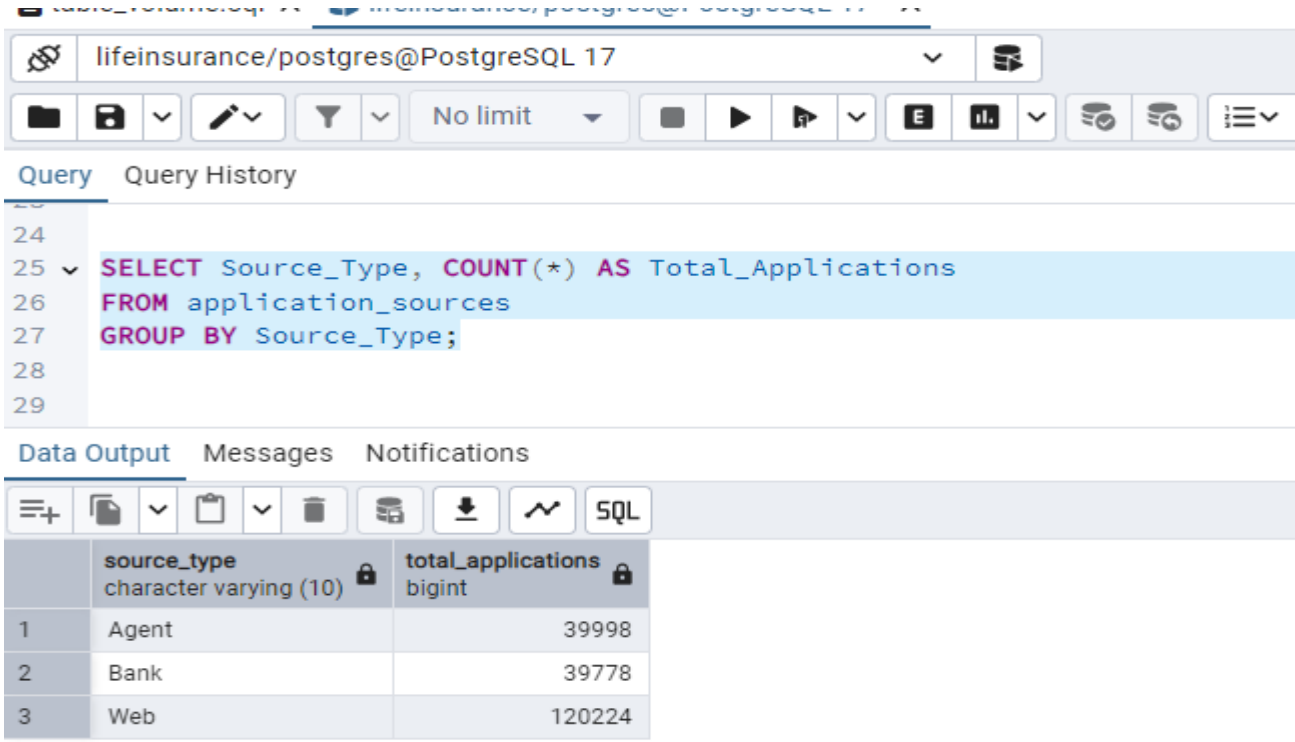
	policy_number character varying (15)	total_payments bigint	total_amount numeric
1	POL0065106	47	164841.69
2	POL0047120	46	163894.32
3	POL0107922	46	149781.06
4	POL0126728	48	146112.48
5	POL0155638	49	144690.63
6	POL0123151	35	134968.40
7	POL0160942	35	134968.40
8	POL0139768	48	132878.40
9	POL0007307	50	120385.00
10	POL0044065	41	116127.99

14. **Agents**: Lists active agents sorted by the highest commission rates.



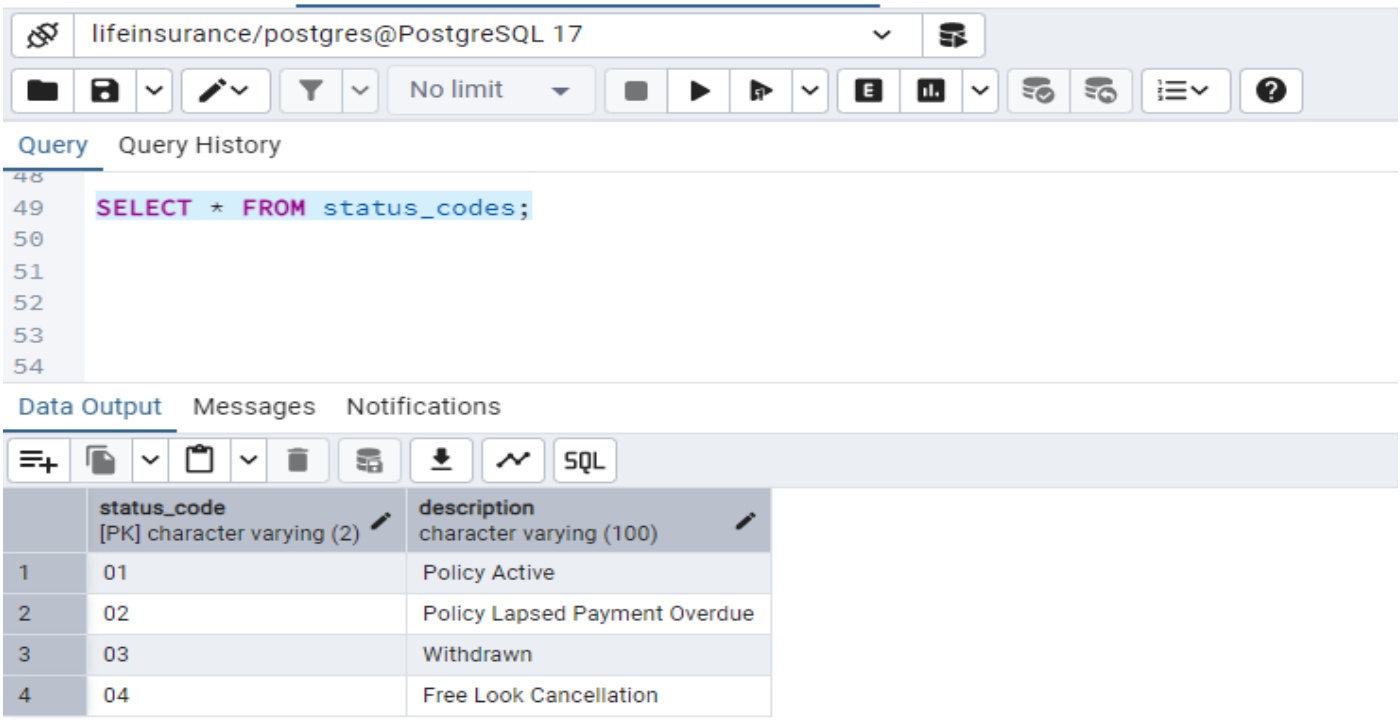
	agent_id [PK] character varying (10)	first_name character varying (50)	last_name character varying (50)	status character varying (10)	commission_rate numeric (5,2)
1	A544	Stephen	Cordova	Active	9.99
2	A573	Dean	Cardenas	Active	9.98
3	A064	Bailey	Chapman	Active	9.98
4	A419	Wendy	Johnson	Active	9.96
5	A627	Matthew	Phillips	Active	9.96
6	A700	Michael	Hall	Active	9.96
7	A197	Mark	Turner	Active	9.94
8	A093	David	Huber	Active	9.92
9	A053	Joseph	Harvey	Active	9.92

15. Application Sources: Counts total applications grouped by their source type (Agent, Bank, Web).



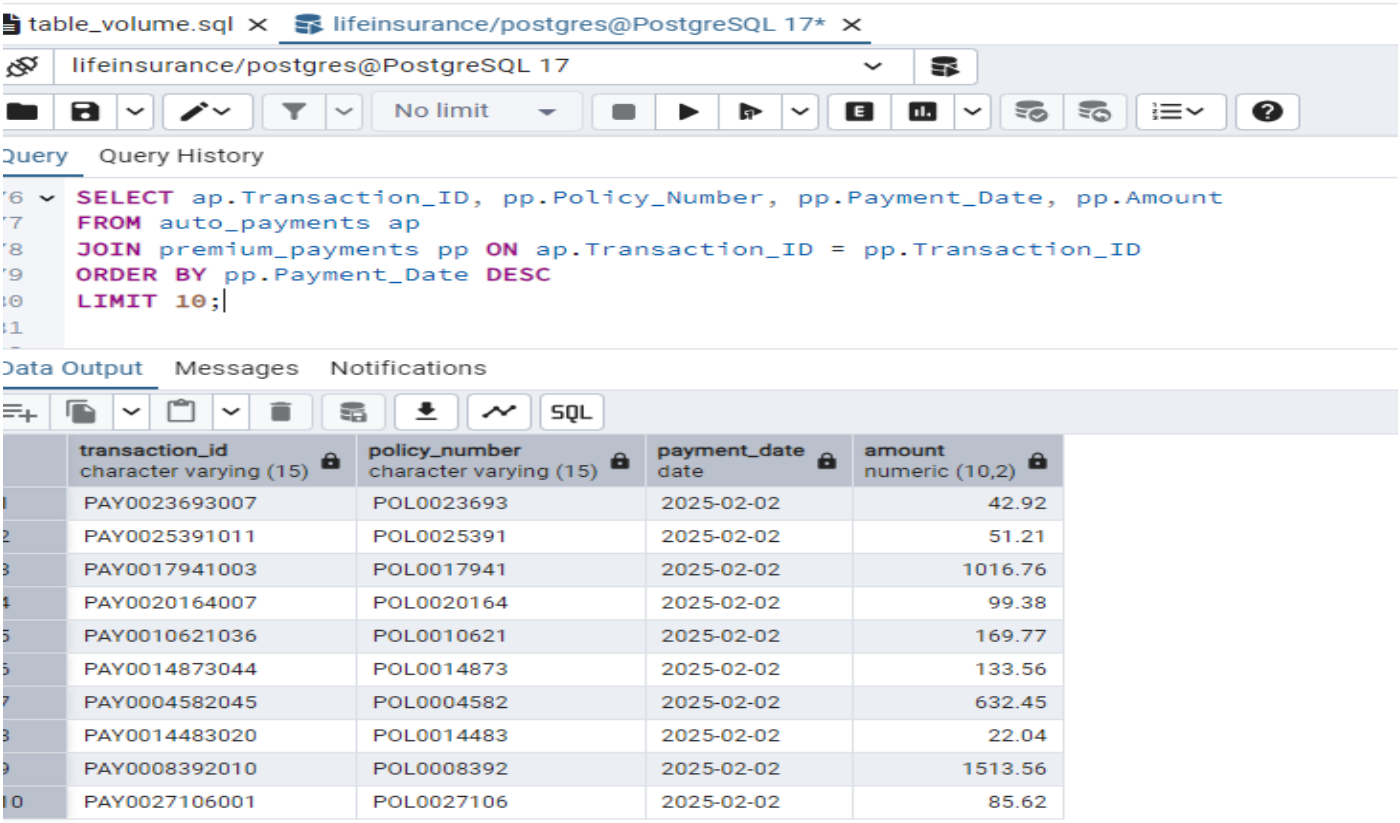
	source_type character varying (10)	total_applications bigint
1	Agent	39998
2	Bank	39778
3	Web	120224

16. Status Code: Displays all status codes with their corresponding descriptions



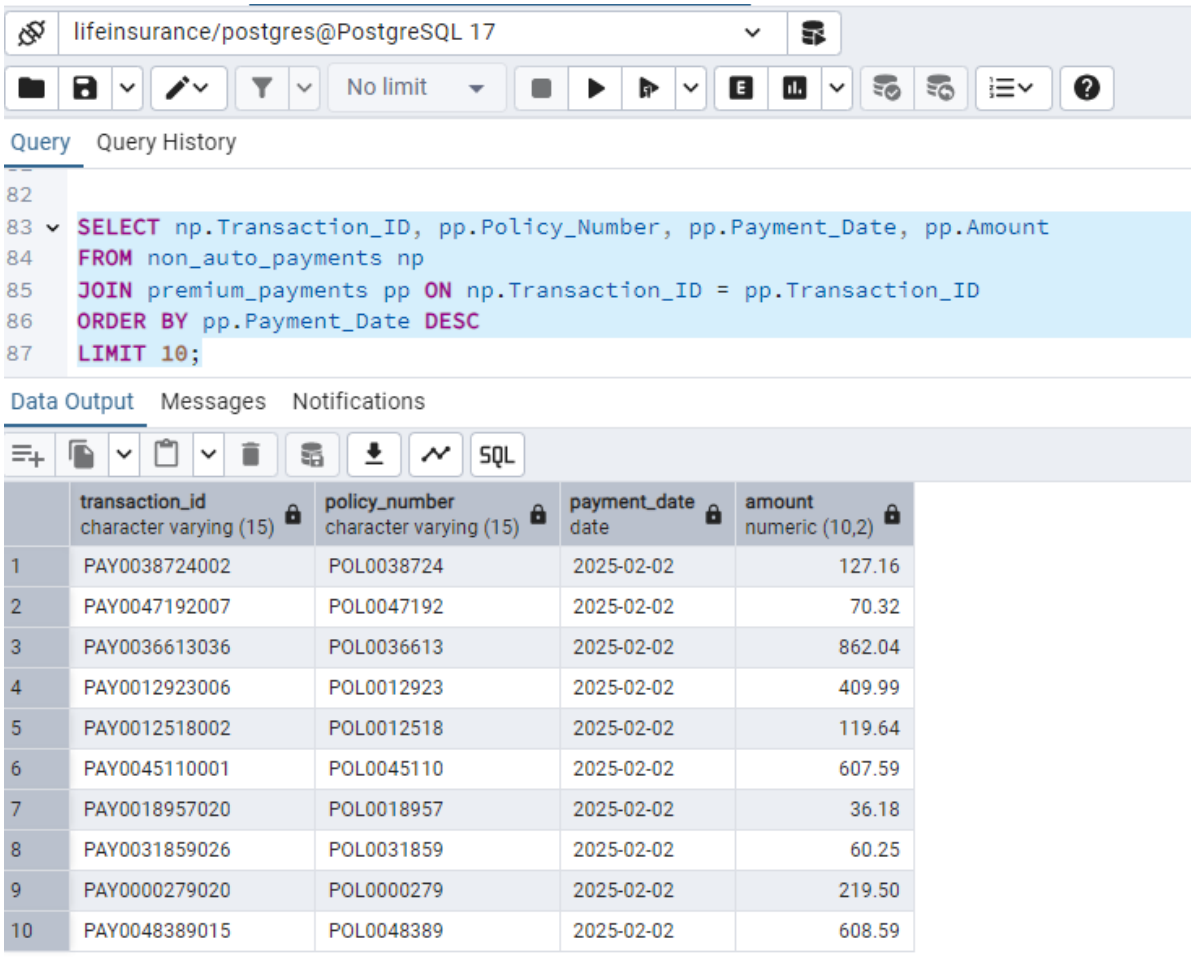
	status_code [PK] character varying (2)	description character varying (100)
1	01	Policy Active
2	02	Policy Lapsed Payment Overdue
3	03	Withdrawn
4	04	Free Look Cancellation

17. Auto Payments: Lists 10 recent auto-pay premium transactions



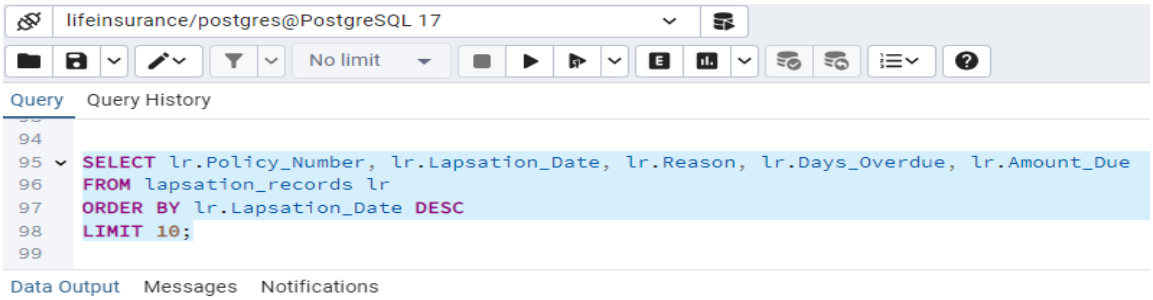
transaction_id character varying (15)	policy_number character varying (15)	payment_date date	amount numeric (10,2)
PAY0023693007	POL0023693	2025-02-02	42.92
PAY0025391011	POL0025391	2025-02-02	51.21
PAY0017941003	POL0017941	2025-02-02	1016.76
PAY0020164007	POL0020164	2025-02-02	99.38
PAY0010621036	POL0010621	2025-02-02	169.77
PAY0014873044	POL0014873	2025-02-02	133.56
PAY0004582045	POL0004582	2025-02-02	632.45
PAY0014483020	POL0014483	2025-02-02	22.04
PAY0008392010	POL0008392	2025-02-02	1513.56
PAY0027106001	POL0027106	2025-02-02	85.62

18. Non-Auto Payments: Lists 10 recent non-auto-pay premium transactions.



	transaction_id character varying (15)	policy_number character varying (15)	payment_date date	amount numeric (10,2)
1	PAY0038724002	POL0038724	2025-02-02	127.16
2	PAY0047192007	POL0047192	2025-02-02	70.32
3	PAY0036613036	POL0036613	2025-02-02	862.04
4	PAY0012923006	POL0012923	2025-02-02	409.99
5	PAY0012518002	POL0012518	2025-02-02	119.64
6	PAY0045110001	POL0045110	2025-02-02	607.59
7	PAY0018957020	POL0018957	2025-02-02	36.18
8	PAY0031859026	POL0031859	2025-02-02	60.25
9	PAY0000279020	POL0000279	2025-02-02	219.50
10	PAY0048389015	POL0048389	2025-02-02	608.59

19. Lapsation Records: Lists 10 recent policy lapsations with overdue detail



	policy_number character varying (15)	lapsation_date date	reason character varying (200)	days_overdue integer	amount_due numeric (10,2)
1	POL0108334	2025-12-20	Missed Payments	62	1900.65
2	POL0068238	2025-12-14	Missed Payments	52	856.21
3	POL0156226	2025-12-10	Missed Payments	87	1427.02
4	POL0062991	2025-12-04	Financial Hardship	46	547.82
5	POL0119800	2025-12-03	Financial Hardship	76	748.35
6	POL0147339	2025-11-25	Missed Payments	91	287.99
7	POL0092702	2025-11-23	Financial Hardship	101	1573.33
8	POL0053605	2025-11-23	Missed Payments	54	962.53
9	POL0084339	2025-11-17	Missed Payments	97	1337.88
10	POL0041912	2025-11-14	Missed Payments	71	1586.39

Thinking Ahead:

In our project, we have incorporated key **dimensions, hierarchies, and measures** that allow for better organization and insights into the life insurance system.

Dimensions:

1. **Time Dimension:** Tracks events like application date, policy issue date, premium due date, lapsation date, and reinstatement date.
    - **Hierarchy:** Year → Quarter → Month → Day
2. **Customer Dimension:** Captures customer demographics and financial details.
    - **Hierarchy:** Customer → Occupation → Income Group
3. **Geographical Dimension:** Helps analyze policy distribution and trends based on location.
    - **Hierarchy:** Country → State → City → Zip Code
4. **Policy Dimension:** Contains details about policy type, premium frequency, and sum assured.
    - **Hierarchy:** Policy Type → Term Duration → Premium Frequency
5. **Agent & RM Dimension:** Tracks performance of agents and relationship managers.
    - **Hierarchy:** Agent/RM → Branch → Region

Measures:

1. **Total Premium Collected** – Sum of all premium payments.
2. **Policy Lapsation Rate** – Percentage of policies that have lapsed.
3. **Reinstatement Success Rate** – Number of reinstated policies vs. total lapsed policies.
4. **Customer Retention Rate** – Number of customers renewing policies.
5. **Agent Performance Score** – Policies sold and commissions earned by agents.