AUTO INSURANCE POLICY DATABASE

Database Specification: Purpose, Business Problems Addressed, Business Rules, Design Requirements, and Design Decisions.

Team: Himanshu Saxena, Komal Matam, Harsh Goswami

Database Purpose

XYZ insurance company wants to setup a database for automating the flow of information between its business units. The purpose of the database is to ingest the information related to drivers, policies, coverage, financials like bills, payments, and claims. Ensure data integrity and query the database for information retrieval. Connect the database to Power BI and generate reports to provide analytics to the business.

Business Problems Addressed

1. Report development for the key stakeholders and other business executives for business management and development
2. Enable Sales Managers to gain analytics from the insurance database for business management
3. Enable Sales Executives to keep a track of the new leads
4. Enable Operations team to keep a track of Claims

Business Rules

1. One Policy can have one or more registered Drivers
2. There can be more than one Vehicle under a single Policy
3. One Driver can have zero or more Claims
4. One Policy can have zero or more policy edit logs
5. One Driver can have one or more Addresses
6. One Policy will have multiple Bills
7. There are different types of Coverage that apply to either Policy or to Vehicle. Some apply to both of them as well.

Design Requirements (Credit to Prof. Simon)

- Use Crow's Foot notation
- Specify the primary key fields in each table by specifying PK beside the fields
- Draw a line between the fields of each table to show the relationships between each table. This line should be pointed directly to the fields in each table that are used to form the relationship
- Specify which table is on the many side of the relationship by placing a Crow's Feet symbol next to the field where the line ends.

Design decisions

Entity Name	**What information is contained in the Entity/Table**	**How Entity is related to other Entities**
Policy	The policy table contains all policy information about the customer/driver	Policy is the parent entity. It is related to Bills with One-To-Many relationship.
Vehicle	This table contains data of all registered vehicles under the given policy for the drivers	There is One-To-Many relationship between Policy and Vehicle table.
Driver	This table contains data of all the registered drivers for a given policy.	There is One-To-Many relationship between policy and driver table.
Vehicle_Driver	This is the bridge table between vehicle and driver	There is Many-To-Many relationship between vehicle and driver table, hence this table acts as the bridge table/associative entity.
DriverAddress	This table holds the driver's mailing and garage address,etc..	It has one to many relationship with the driver entity.
TrafficViolationCode	This table holds the different Traffic Violation Code information.	There is Zero-To-Many relationship between this table and Driver_TrafficViolation_Code. Similarly there is Zero-To-Many relationship between driver and Driver_TrafficViolation_Code. We have zero to many instead of One-To-Many because some drivers have never violated the traffic rule.
Driver_TrafficViolation_Code	This is the bridge table between driver and traffic violation code.	There is Many-To-Many relationship between driver and traffic violation code table, hence this table acts as the bridge table/associative

		entity.
Bill	This table contains monthly bills that need to be paid by the customer to the insurance company	It has one to many relationship with the policy table
PaymentDetail	This table contains the payment details of the customer like multiple transaction details.	It has one to many relationship with the billing table.
Coverage	There are different type of coverage in the Auto Insurance like Liability, Collision, Comprehensive, Rental, Medical Payment, Towing, Mechanical Break Down. Some of them apply only to policy, some only to vehicle and some of them to both. IsPolicyCoverage and IsVehicleCoverage tell, is it applicable to Vehicle or Policy and if it is applicable to both, then these two boolean columns will have true value.	Coverage has a One-To-Many relationship with Policy_Coverage and Vehicle_Coverage
Claim	This table contains any claims submitted by the customer like an accident that caused damage to the vehicle or	It has one to many relationship with the driver table.
Policy_Coverage	This is the bridge table between policy and coverage	There is Many-To-Many relationship between coverage and policy table, hence this table acts as the bridge table/associative entity.
Vehicle_Coverage	This is the bridge table between vehicle and coverage	There is Many-To-Many relationship between vehicle and coverage table, hence this table acts as the bridge table/associative entity.
PolicyEditLog	This table keeps the log	It has one to many

	information if customer updates the policy information like remove vehicle or add driver, etc.	relationship with the policy table.