House Insurance Database Design

Group 3: Chunyan Hao, Ming Luo, Mugdha Parbat, Pranav Pulkundwar

Database Specifications:

Database Purpose, Business Problems Addressed, Business Rules, Design Decision, ER Diagram

Database Purpose:

The objective of the House Insurance Database is to maintain acquired data and establish a foundation for insurance companies to strategically analyze potential client information. This will speed up the underwriting department's risk evaluation and decision-making process. The database can be used by various departments: finance, legal, claims, underwriting, marketing, etc. The database can assist in the diversification of finances, investments and calculating policy premiums, evaluating property conditions, and analyzing policy features resulting in continued growth and sustainability.

Business Problem Addressed:

- Maintain (enter, update, and delete), track, and report detailed information for each potential new case, enable the easy access and detailed analysis of the database.
- Evaluate house conditions and risk of getting damaged based on the company's internal rating system. Provide an initial insurance price combining information of internal possessions values, community environment as well as house insurance history.
- Query the information of any popular insurance coverage across a certain demographic from the database which can be helpful to provide valuable information to build and expand product marketing strategies
- Optimize the insurance coverage pricing strategy and maintain a balance between a high gross profit rate and a high customer engagement rate.
- Extract customer source reports from the database and estimate advertisement efficiency in different platforms like social media, house agencies and flyers.
- Generate data to analyze customer's features like address, age, education, and credit scores to figure out insurance preference in different customer segments. Then design more effective marketing strategies.

Business Rules:

- One case is mandatory to have one and only one customer. One customer is mandatory to have at least one case record. (non-identifying)
- One case is mandatory to have at least one CaseCoverage record. One case coverage is mandatory to have only one case record. (identifying)
- One CustomerSource is mandatory to have at least one Customer. One Customer is mandatory to have only one CustomerSource record. (non-identifying)
- One CaseCoverage is mandatory to have one and only one CoverageOptions. One CoverageOptions is optional to have many CaseCoverage. (identifying)

- One house is mandatory to have at least one case record. One case is mandatory to have one and only one house. (non-identifying)
- One house is mandatory to have one and only one address. One address is optional to have one house. (non-identifying)
- One customer is mandatory to have one and only one address. One address is optional to have one customer. (non-identifying)
- One CommunityEnviroment record is optional to have many addresses. One address is mandatory to have only one CommunityEnviroment. (non-identifying)
- One ClaimHistory is optional to have one HouseInfo record. One HouesInfo record is optional to have many ClaimHistory. (non-identifying).
- One Payment record is mandatory to have only one ClaimHistory. One ClaimHistory is optional to have many Payment records. (non-identifying)
- One Payment record is mandatory to have only one CoverageOption. One CoverageOption is optional to have many payment records. (non-identifying)
- One ClaimHistory is optional to have one CustomerInfo record. One CustomerInfo is optional to have many ClaimHistory. (non-identifying)
- One InternalPossessions is mandatory to have one HouseInfo. One HouseInfo is optional to have many InternalPossessions. (identifying)
- One SecuritySystem is mandatory to have only one HouseInfo. One HouseInfo is optional to have many SecuritySystem. (identifying)
- If any address in the area whose crime rate is equal to more than 65%, such record won't be added to Address table
- If the payment amount is 0, such record won't be added in the Payment table
- If a house has more than 5 claim history records, it will be prevented from enrolling new case record

Design Decisions:

Entity Name	Why is this entity included?	How is this entity related to other entities?
CasePortal	"CasePortal" holds general information about the new cases. This is a basic entity in the database which contains "start_date", "end_date", "house_id", and "customer_id" of the enquiry. It also collects information about deductibles and limits. Deductible is the amount of money a houseowner must pay from their pocket before house insurance coverage is claimed. When the insurance company pays the claim, it will be for the total damage amount minus the amount of the deductible. A limit is the highest	The "CasePortal" entity is related to the "CustomerInfo" via its primary key, "customer_id" to store all the basic information about the customer.

Similarly, "CasePortal" has foreign key "house_id" which ties to "HouseInfo". The company will be able to view detailed house information by using the foreign key "house_id" for investigation of house. |

	amount the insurance company will have to pay for a claim that the insurance policy covers. Hence, this information is crucial to be stored in the database for future usage when the company must calculate the amount of claim to be paid.	"CasePortal" is connected to CoverageOptions via an associative entity "CaseCoverage" to maintain the many to many relationships.
CustomerInfo	This is one of the most important entities in the insurance database as it contains comprehensive information about all customers. It is made up of personal information of the customer along with vital information such as "credit score", "highest_education", and smoking habits. This information plays a significant role as certain incentives is given to the customer while calculating policy premiums.	The "CustomerInfo" entity is related to "CustomerSource" via "customer_source_id" and has a one-to-many relationship. This is because, there would be one or multiple customers who might have approached the insurance company through the same "CustomerSource". "CustomerInfo" entity is connected to the "Address" entity. It is important to note that this address entity gives us the mail address of the customer, and the house to be insured. It is possible that the customer stays in a different location with the house to be insured. The "CustomerInfo" can retrive "ClaimsHistory" via "customer_id" to check whether and when the person had a house insurance claim before.
CustomerSource	This entity supports the insurance company by recording various sources the customers have approached the insurance company. This entity will be particularly helpful for the marketing and advertising department to identify the best and least performing advertising channels.	The "CustomerSource" is related to "CustomerInfo" as every customer will have a source from which they came to know about the insurance company.
CaseCoverage	"CaseCoverage" is an associative entity which carries the information about "case_id" and "coverage_id". It connects "CasePortal" and "CoverageOptions" with a many to many relationships.	

CoverageOptions	This entity is joined by "CaseCoverage", to the "CasePortal". Coverage option describes the type of incident it is covering. For example, real property damage, damage to private property, or damage occurred to the house due to natural calamities such as floods, earthquakes, mudslides, etc.	
HouseInfo	This is another most important entity in this database, as our database is focused on managing data for a house insurance company. It has basic details such as "house_id" and "zip-code" along with descriptive information such as the number of bedrooms, type of construction and distance of the house from river and fire station. This detailed information is crucial for the insurance company to calculate premiums and build an insurance policy.	"HouseInfo" is connected to the "CasePortal", and "Address". "InternalPossissions" is connected solely to the "HouseInfo" which contains information about valuable personal items in the house which will be covered in the insurance. It is a zero-to-many relationships as some customers would not have any precious items whereas some might have several. This entity is also connected directly to "SecuritySystem" which also has zero-to-many relationships. It describes what type of smoke alarm, burglar alarms, and deadbolts a house has or if it has any at all.
Address	The insurance company needs to keep records of all house addresses to assess the related environment information that would be essential in calculating the policy premium. Since the premium amount depends on the location and the surroundings of the house. This entity also contains the customer's mail address which might be different from the house to be insured.	This entity is connected to "CustomerInfo" through the primary key "address_id". It is also connected to the "CommunityEnvironment" to record environment conditions in the community.
CommunityEnvironment	This entity is useful to keep information about the environment conditions of the house that directly affect the policy premium rate a house might need, for it to be insured from attributes like the crime rate, population, earthquakes rate,	This entity is connected only to "Address" entity via zip-code.

	floods, and wildfires or other natural or manmade calamities in the area.	
ClaimHistory	This entity helps store and track past claims records. Each claim is given a unique "claim_id". Insurance claim history is crucial when evaluating the potential damage involved in the house or the customer or both.	This entity is directly connected to "HouseInfo" and "CustomerInfo" entities with their foreign keys. These connections help track the detailed information about the customer demographics and the house for which the claim was made. This entity is also connected to the "Payment" entity through "claim_id" to keep a track of the payment's records received with a house insurance claim.
Payments	Payment entity is created to help track the past payment records for any claim history. Each payment is given a payment_id to distinguish all payments from one another. The attribute "coverage_id" will note the corresponding type of coverage. Amount paid for this claim is also stored in an attribute called "amount_paid". The status of the payment is tracked to know if the payment is completed or if it is in process.	This entity is directly connected to the "ClaimsHistory" entity. It is also connected to the "CoverageOption" entity by "coverage_id" as a foreign key.
InternalPossession	When a customer files for insurance of a house, the insurance company also needs to check the actual value of internal valuable possessions which the customer wants insuring. This possession check includes a thorough analysis of all items and areas of the house which helps calculate the correct policy premium.	This entity is connected "HouseInfo" by its primary key. It is a zero-to-many relationships as some customers would not have any precious items in the house whereas some might have several.
SecuritySystem	This entity is needed to consider which type of security system a house has in case of a break-in or any other type of incident. This entity helps to decide how safe the house is irrespective of external factors like surroundings and natural calamities.	This Entity is connected to the "HouseInfo" entity by a one-to-many relationship which means that it records all the security system backups a single house has.