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.

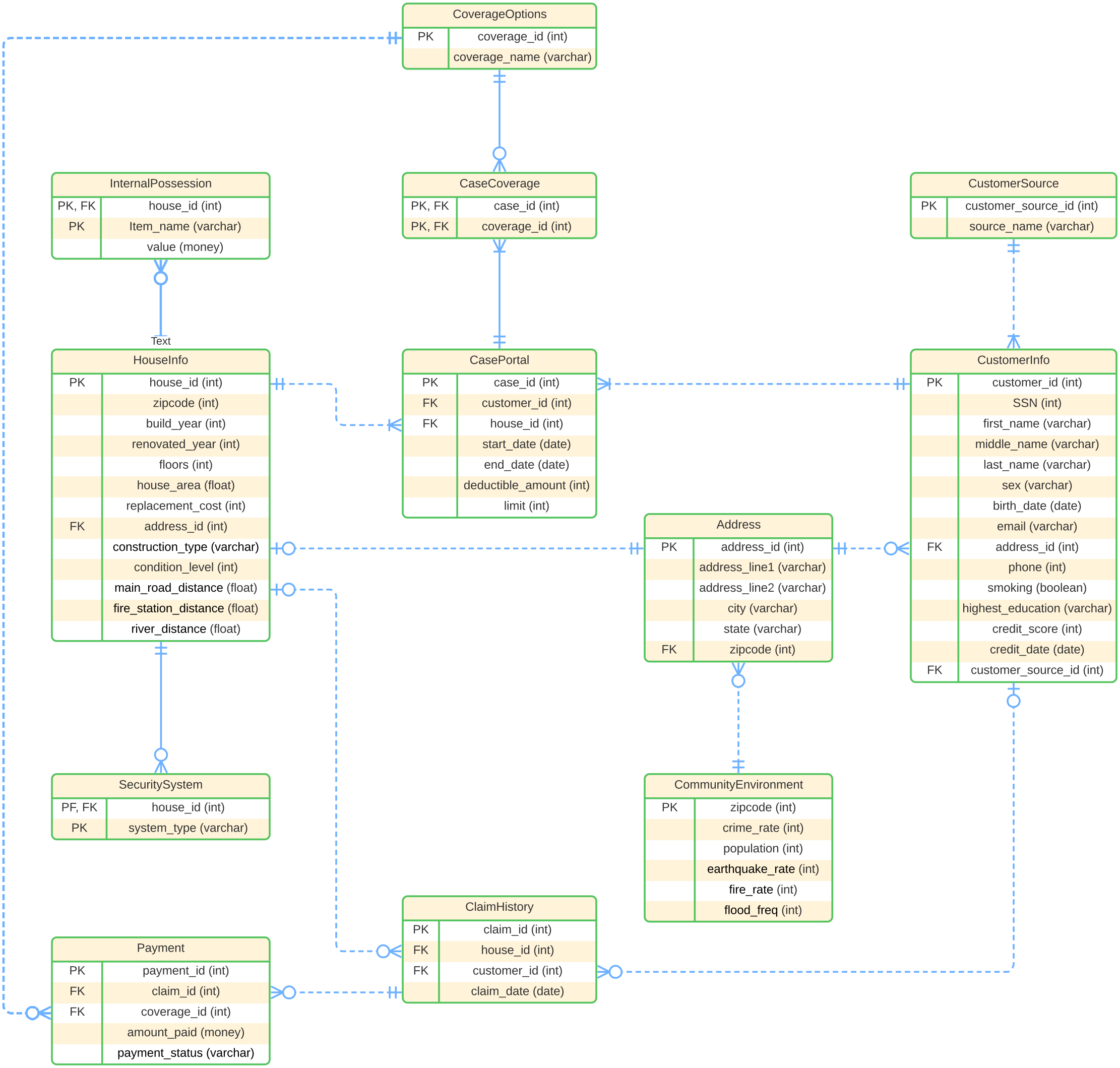
Business Rules:

- 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)

Feedback:

No.	Given Feedback	Feedback Implemented
1.	The PK in entities "HouseInfo", "Payment", "CoverageOptions", "CasePortal", "Address", "CommunityEnvironment", "CustomerInfo" is not an FK. Please remove the FK label.	Removed FK labels from PK in entities "HouseInfo", "Payment", "CoverageOptions", "CasePortal", "Address", "CommunityEnvironment", "CustomerInfo" as it was not a foreign key.
2.	The relationship between entity "CustomerInfo" and entity "CustomerSource" should be non-identifying.	Established a non-identifying relationship between the entities – "CustomerInfo" and "CustomerSource" by replacing connection with a dotted line. Established it as a non-identifying relationship because – 1. Both entities exist independently. 2. PK of CustomerInfo doesn't contain PK component of CustomerSource.
3.	For the one-to-many relationship between "Payment" and "ClaimHistory", many is on "Payment". Therefore, claim_id should be put in "Payment" as the FK. Please remove payment_id from "ClaimHistory".	Since entities "ClaimHistory" and "Payment" have a one-to-many relationship with many on "Payment", we therefore put claim_id "ClaimHistory" as an FK in "Payment" and removed payment_id from the ClaimHistory. This was implemented because there is one-to-many relationship present between these tables. The primary key field in the "ClaimHistory", claim_id, is designed to contain unique values. The foreign key field in the "Payment", claim_id, is designed to allow multiple instances of the same value.

The updated ERD is given below:


CoverageOptions
PK
coverage_id (int)
coverage_name (varchar)
InternalPossession
PK, FK
house_id (int)
PK
Item_name (varchar)
value (money)
Text
CaseCoverage
PK, FK
case_id (int)
PK, FK
coverage_id (int)
CustomerSource
PK
customer_source_id (int)
source_name (varchar)
HouseInfo
PK
house_id (int)
zipcode (int)
build_year (int)
renovated_year (int)
floors (int)
house_area (float)
replacement_cost (int)
FK
address_id (int)
construction_type (varchar)
condition_level (int)
main_road_distance (float)
fire_station_distance (float)
river_distance (float)
CasePortal
PK
case_id (int)
FK
customer_id (int)
FK
house_id (int)
start_date (date)
end_date (date)
deductible_amount (int)
limit (int)
Address
PK
address_id (int)
address_line1 (varchar)
address_line2 (varchar)
city (varchar)
state (varchar)
FK
zipcode (int)
CustomerInfo
PK
customer_id (int)
SSN (int)
first_name (varchar)
middle_name (varchar)
last_name (varchar)
sex (varchar)
birth_date (date)
email (varchar)
FK
address_id (int)
phone (int)
smoking (boolean)
highest_education (varchar)
credit_score (int)
credit_date (date)
FK
customer_source_id (int)
SecuritySystem
PF, FK
house_id (int)
PK
system_type (varchar)
CommunityEnvironment
PK
zipcode (int)
crime_rate (int)
population (int)
earthquake_rate (int)
fire_rate (int)
flood_freq (int)
ClaimHistory
PK
claim_id (int)
FK
house_id (int)
FK
customer_id (int)
claim_date (date)
Payment
PK
payment_id (int)
FK
claim_id (int)
FK
coverage_id (int)
amount_paid (money)
payment_status (varchar)