

Data And Application

Group Project Phase - 1

Team No. - 57

Team Name - Demolishers

Team members Details

Name	Roll no.	Email ID
Abhinav Anand	2020101054	abhinav.a@students.iiit.ac.in
Lavisha Bhambri	2020101088	lavisha.bhambri@students.iiit.ac.in
Mayank Bhardwaj	2020101068	mayank.bhardwaj@students.iiit.ac.in

Introduction

1.1 The Mini world

The mini-world we are making a database for is an insurance company. It sells various types of insurances like life insurance, vehicle insurance, travel insurance, medical insurance, house insurance to different customers on the basis of the insurance policy. It also involves third-party administrators to settle the claims or cases.

1.2 Purpose

The purpose of this database is to provide a general synopsis of the various aspects of the selling, buying and resolution of insurances of the company. It also simplifies the management of the details of the users and their claims in the Insurance company. The application has complete access for the CRUD operations that are to create, read, update and delete the database entries.

1.3 Users

The application has different uses based on the role of the user:-

1. Management - Management can use this application to keep track of the overall insurance status.
2. Employees - Employees can use the application to verify the client's information about the different types of their insurances. Employees can also look at the number of insurances they have sold to their clients and how many cases were resolved by them in a given period.
3. Client (End-users) - Clients are the end-users of the application. They can look at their insurance details and their resolved claims. They can also know the employee who sold the insurance to them.

Both the employees and the clients can also see their respective dependent information.

2.1 Database Requirements

2.1.1 Insurance policy

This entity stores information about the insurance being sold.

Attribute	Description
Policy id (Primary Key)	A unique number given to every insurance policy.
Date of issue	The date when the insurance policy was sold.
Duration	Number of years for which the insurance policy is active.
Date of expiry (Derived Attribute)	The date after which the insurance policy expires, derived from Date of Issue and Duration.
Sum assured	The maximum amount of money that can be claimed under this insurance policy.
Premium value	The premium required for the insurance policy.
Terms and Conditions	Contains the terms and conditions for the policy.

2.1.2 Third Party Administrator (TPA)

This entity stores information of the various third party administrators used by the company.

Attribute	Description
TPA id (Primary Key)	A unique number given to every TPA.
TPA name	Stores the name of the TPA.
TPA contact info	Multivalued attribute storing contact info of the TPA.
Types of investigations	Multivalued attribute that stores the types conducted of insurance claims a TPA processes.
TPA address	A composite attribute that stores the TPA address composed of the four attributes: street address + zip code + city + state.
Street address	Stores the street address of the TPA.

Zip code	Stores the zip code of the TPA.
City	Stores the city of TPA .
State	Stores the state of TPA.

2.1.3 Customer

This entity stores information about the customers of the company.

Attribute	Description
Aadhar number (Primary Key)	Stores the Aadhar number of the customer.
Date of Birth	Stores the date of birth of the customer.
Age	Derived Attribute from date of birth.
Contact Numbers	Multivalued Attribute storing the contact numbers of the customer.
Email id	Stores the email id of the customer.
Full name	Composite Attribute storing the full name of the customer made of first name, middle name and last name.
First name	Stores the first name of the customer.
Middle name	Stores the middle name of the customer.
Last name	Stores the last name of the customer.
Customer status	Stores the customer loyalty level (Platinum, Gold, Silver, Bronze, Normal).
Residential address	A composite attribute that stores the address composed of the four attributes: street address + zip code + city + state.
Street address	Stores the street address of the customer's location.
Zip code	Stores the zip code of the customer's location.
City	Stores the city of the customer's location.
State	Stores the state of the customer's location.

2.1.4 Employee

This entity stores information about the employees of the company.

Attribute	Description
Employee id (Primary key)	Made up of two attributes, Department number and serial number.
Department number	Unique number given to every department.
Serial number	A Number given to every employee in a department.
Aadhar number (Secondary Key)	Stores the Aadhar number of the employee.
Date of Birth	Stores the date of Birth of the employee.
Age	Derived attribute from date of birth.
Contact Numbers	Multivalued attribute storing the contact numbers of the employee.
Email id	Stores the email id of the employee.
Full name	Composite attribute storing the full name of the employee made of first name, middle name and last name.
First name	Stores the first name of the employee.
Middle name	Stores the middle name of the employee.
Last name	Stores the last name of the employee.
Residential address	A composite attribute that stores the address composed of the four attributes: street address + zip code + city + state.
Street address	Stores the street address of the employee's location.
Zip code	Stores the zip code of the employee's location.
City	Stores the city of the employee's location.
State	Stores the state of the employee's location.

2.2 Subclass

There are 5 subclasses for the entity insurance which are disjoint and the participation of the insurance entity is total. They are as follows:-

2.2.1 Life Insurance

If the insurance being issued is life insurance.

Attribute	Description
Death Benefit value	Money received by the beneficiaries in case of an untimely death of the person.
Beneficiaries	Multi valued attribute which stores the beneficiaries
Medical History	Stores the medical history of the customer.

2.2.2 Medical Insurance

If the insurance being issued is medical insurance.

Attribute	Description
Medical History	Stores the medical history of the customer.
Medical conditions covered	Multi value attribute which stores medical conditions covered.
Cashless hospitals	Multi value attribute which stores hospitals that will accept the insurance cashless.

2.2.3 Vehicle Insurance

If the insurance being issued is vehicle insurance.

Attribute	Description
Vehicle colours	Multi value attribute that Stores the colours of the vehicle.
Vehicle license number (Primary Key)	Stores the vehicle license number.

Customer license number (Primary Key)	Stores the customer license number.
---	-------------------------------------

2.2.4 House Insurance

If the insurance being issued is house insurance.

Attribute	Description
Replacement cost	Stores the inspected value of the house.
House address	A composite attribute that stores the address composed of the four attributes: street address + zip code + city + state.
Street address	Stores the street address.
Zip code	Stores the zip code.
City	Stores the city.
State	Stores the state.

2.2.5 Travel Insurance

If the insurance being issued is travel insurance.

Attribute	Description
Travel Destinations	Multi value attribute storing the places being visited
Itinerary	Stores the itinerary of the trip
Airline and Hotel Booking	Multi value attribute storing the airline bookings and hotel bookings used for the trip.

2.3 Weak Entity

2.3.1 Employee dependent

This particular entity type would store all the necessary information related to the dependents of an employee. It is a weak entity because it is associated with an employee, and if the employee doesn't exist, the employee dependent would not exist. Being a weak entity this does not have a primary key, rather it is identified using the employee it is associated with.

Attribute	Description
Name (Weak Key Attribute)	Composite attribute made of first name, middle name, and last name.
First name	Stores the first name of the dependent.
Middle name	Stores the middle name of the dependent.
Last name	Stores the last name of the dependent.
Date of Birth	Stores the date of birth of the dependent.
Age	Derived attribute from date of birth.

2.3.2 Customer dependent

This particular entity type would store all the necessary information related to the dependents of a customer. It is a weak entity because it is associated with a customer, and if the customer doesn't exist, the customer dependent would not exist. Being a weak entity this does not have a primary key, rather it is identified using the customer it is associated with.

Attribute	Description
Name (Weak Key Attribute)	Composite attribute made of first name, middle name, and last name
First name	Stores the first name of the dependent
Middle name	Stores the middle name of the dependent
Last name	Stores the last name of the dependent
Date of Birth	Stores the date of birth of the dependent
Age	Derived attribute from date of birth

2.4 Relationship Types

2.4.1. Supervision

1. A 1:n relationship type between employee(as Supervisor) and employee(as Supervisee).
2. Employee (as Supervisor) participation is partial.
3. Employee (as Supervisee) participation is partial.
4. An employee (as supervisee) can have a minimum 0 supervisor to at max 1 supervisor.
5. An Employee (as supervisor) can command minimum 0 to multiple numbers of Employees.

2.4.2 Provided by

1. A m:n relationship type between employee dependent and employee.
2. Employee dependent participation is total.
3. Employee participation is partial.
4. Each Employee dependent can have 1 to multiple number of employees to depend on.
5. Each Employee can have 0 to multiple number of employee dependents to take care of.

2.4.3 Issued by

1. A n:1 relationship type between Insurance policy and employee.
2. Insurance policy participation is total.
3. Employee participation is partial.
4. Each insurance policy can be issued by exactly 1 employee.
5. Each employee can issue 0 to multiple numbers of policies.

2.4.4 Bought by

1. A 1:n relationship type between Insurance policy and customer.
2. Insurance policy participation is total.
3. Customer participation is total.
4. Every insurance policy is bought by 1 customer.
5. Each customer has 1 or more insurance policies.

2.4.5 Depends on

1. A m:n relationship type between customer dependent and customer.
2. customer dependent participation is total.
3. Customer participation is partial.
4. Each customer can provide for 0 or some number of people.
5. Every dependent depends on 1 or more customers.

2.4.6 Resolves Claim

1. A 4-ary relationship between TPA, Insurance policy, Customer, Employee.
2. TPA participation is partial.
3. Insurance policy participation is partial.
4. Customer participation is partial.
5. Employee participation is partial.
6. It is a (1,1,1,n) relationship type.
7. The TPA investigates cases or claims made by customers and helps in finding information for the company(represented by employees) regarding a policy.
8. Not all TPA's investigate, and only one works on 1 case.
9. Not all policies get involved, and only 1 is claimed at a time.
10. Not all customers make claims, and only 1 customer's claim is in a case.
11. Not all employees help in resolution, and multiple employees can help in them.

Functional Requirements

Modifications

1. Insert:

1.1 AddCustomer():

Input- Aadhar number, Full name, Date of Birth, Contact number/s, email id, customer status(platinum, gold, silver, bronze or normal), Residential Address.

Output- If the entered details are valid, then this data is added to the database.

1.2 AddCustomerDependent():

Input- Aadhar number of the customer this person is dependent on, Full name, Date of Birth.

Output- If the entered details are valid, then this data is added to the database.

1.3 AddEmployee():

Input- Aadhar number, Full name, Date of Birth, Contact number/s, email id, Residential Address, Department number of where the employee is to be assigned.

Output- If the entered details are valid, then this data is added to the database and the employee is assigned a unique employee id consisting of a department number and a serial number.

1.4 AddEmployeeDependent():

Input- Employee Department number of the employee that provides for this person, Employee Serial number of the employee that provides for this person, Full name, Date of Birth.

Output- If the entered details are valid, then this data is added to the database.

1.5 AddLifeInsurance():

Input- Aadhar number of the customer buying the policy, Date of issue, Duration, Assured sum, Premium value, beneficiaries, death benefit value, Customer medical history.

Output- If the information is valid, then this data is entered in the database along with the standard terms and conditions of a Life Insurance policy issued by the company. And, a unique policy id is assigned. If the addition of this policy affects the status of the customer buying this policy, then that is updated too.

1.6 AddMedicalInsurance():

Input- Aadhar number of the customer buying the policy, Date of issue, Duration, Assured sum, Premium value, Medical conditions covered by this policy, Names of cashless hospitals, Customer medical history.

Output- If the information is valid, then this data is entered in the database along with the standard terms and conditions of a Medical Insurance policy issued by the company. And, a unique policy id is assigned. If the addition of this policy affects the status of the customer buying this policy, then that is updated too.

1.7 AddVehicleInsurance():

Input- Aadhar number of the customer buying the policy, Date of issue, Duration, Assured sum, Premium value, Colours of the vehicle, Vehicle license plate number, Driving license number of the customer buying this policy.

Output- If the information is valid, then this data is entered in the database along with the standard terms and conditions of a Vehicle Insurance policy issued by the company. And, a unique policy id is assigned. If the addition of this policy affects the status of the customer buying this policy, then that is updated too.

1.8 AddHouseInsurance():

Input- Aadhar number of the customer buying the policy, Date of issue, Duration, Assured sum, Premium value, Replacement cost of the house, Full address of the insured property.

Output- If the information is valid, then this data is entered in the database along with the standard terms and conditions of a House Insurance policy issued by the company. And, a unique policy id is assigned. If the addition of this policy affects the status of the customer buying this policy, then that is updated too.

1.9 AddTravelInsurance():

Input- Aadhar number of the customer buying the policy, Date of issue, Duration, Assured sum, Premium value, Customer's travel destination/s, Travel itinerary of the customer, Customer's Air- line and Hotel booking.

Output- If the information is valid, then this data is entered in the database along with the standard terms and conditions of a Travel Insurance policy issued by the company. And, a unique policy id is assigned. If the addition of this policy affects the status of the customer buying this policy, then that is updated too.

1.10 AddTPA():

Input- TPA name, TPA Address, TPA Contact Number/s, Type of Investigations conducted by the TPA.

Output- If the information is valid, then this data is added to the database and a unique TPA id is assigned.

2. Delete:

2.1 DeleteCustomer(aadhar number)

Input- Customer Aadhar number.

Output- If the input is valid, the customer data, data of the people that depend on this customer and the data of the policies bought by this customer is deleted from the database. And a confirmation response of the completion of the task is displayed.

2.2 DeleteCustomerDependent(aadhar number, full name)

Input- Aadhar number of the customer this person is dependent on, the full name of the person.

Output- If the input is valid, the data of this person is deleted from the database. And a confir- mation response of the completion of the task is displayed.

2.3 DeleteEmployee(department no, serial no)

Input- Department number of the employee, serial number of the employee.

Output- If the input is valid, the employee data and the data of the people the employee provides for is deleted from the database. And a confirmation response of the completion of the task is displayed.

2.4 DeleteEmployeeDependent(department no, serial no, full name)

Input- Department number of the employee that provides for this person, Serial number of the employee that provides for this person, the full name of the person.

Output- If the input is valid, the data of this person is deleted from the database. And a confirmation response of the completion of the task is displayed.

2.5 DeletePolicy(policy id)

Input- Policy Id.

Output- If the information is valid then the data of the policy is deleted from the database, and if this deletion affects customer status of the customer that availed it, then it is updated too. And a confirmation response of the completion of the task is displayed.

2.6 DeleteTPA(TPA id)

Input- TPA id.

Output- If the information is valid, then the data of the TPA is deleted from the database. And a confirmation response of the completion of the task is displayed.

3. Update:

3.1 UpdateCustomerInfo(Aadhar number)

Input- Customer Aadhar number, Updated customer information.

Output- If the updated information is valid, then this data is updated in the database.

3.2 UpdateCustomerDependentInfo(Aadhar number, full name)

Input- Aadhar number of customer that the person depends on, full name of the person, updated person information.

Output- If the updated information is valid, then this data is updated in the database.

3.3 UpdateEmployeeInfo(department no,serial no)

Input- Employee department number, Employee Serial number, updated employee information.

Output- If the updated information is valid, then this data is updated in the database.

3.4 UpdateEmployeeDepartment(department no,serial no)

Input- Current Department number of the employee, Current Serial number of the employee, Up- dated Department number of the employee.

Output- If the information is valid, then the employee is assigned a new serial number in the up- dated department, and the employee data is copied to this new employee id and the data in the previous location is deleted from the database. Also the data of the people the employee provides for is linked to this new employee id.

3.5 UpdateEmployeeDependentInfo(department no, serial no, full name)

Input- Department number number of the employee that provides for the person, Serial number number of the employee that provides for the person, full name of the person, updated person information.

Output- If the updated information is valid, then this data is updated in the database.

3.6 UpdateTPAInfo(TPA id)

Input- TPA id, Updated TPA Information.

Output- If the updated information is valid, then this data is updated in the database.

3.7 UpdatePolicy(policy id)

Input- Policy id, updated policy information.

Output- If the policy id is valid, then the system provides fields of data of the subclass that policy belongs to, with the current information displayed in those fields. After the user updates this information and if the updated information is valid, then this data is added to the database.

3.8 UpdateTnC(policy type)

Input- Type of policy whose terms and conditions are to be updated, updated terms and conditions.

Output- If the updated information is valid, then this change is carried out in all the policies of the same type whose terms and conditions were not changed since the policy was issued in the database.

Retrievals

1. Selection:

1.1 GetAllCustomers(customer status)

Input- Customer status.

Output- If the status is valid, data tuples of data of all customers having that status is returned.

1.2 GetAllPoliciesOfCustomer(aadhar number)

Input- Customer Aadhar number.

Output- If the aadhar number is valid, then a data tuple of all the policies bought by the customer is returned.

1.3 GetAllDependentsofCustomers(aadhar number)

Input- Aadhar Number of the customer.

Output- If the aadhar number is valid, the data tuple of all the people dependent on the customer is returned.

1.4 GetAllDependentsofEmployees(department no, serial no)

Input- Department number of the employee, Serial number of the employee.

Output- If the employee id is valid, data tuple of all the people the employee provides for is returned.

1.5 GetAllTPAsInCity(city)

Input- City.

Output- If the city is valid, then a data tuple of the data of all TPAs in the city is returned.

2. Projection:

2.1 GetInsurancePolicies(policy subclasses)

Input- List of policy subclasses.

Output- If the policy subclasses are valid, the data tuple of all the data of the policies of the policy subclasses queried for is returned.

2.2 GetInsuranceCustomers(policy subclasses, customer status)

Input- List of policy subclasses, List of Customer Status.

Output- If the policy subclasses and customer status are valid, the data tuple of all the data of the customers who bought policies of the queried policy subclasses and have status as one of the queried customer status is returned.

2.3 GetPoliciesWithPremium(min, max)

Input- Minimum Premium value, Maximum Premium Value.

Output- If the values are valid, then a data tuple of all policies having premium value between the given range is returned.

2.4 GetCustomersOfAge(age)

Input- Age.

Output- If the age is valid, then a data tuple of all the data of customers and their dependents of the given age is returned.

3. Aggregate:

3.1 GetTotalClaim(policy subclasses)

Input- List of policy subclasses.

Output- If the policy subclasses are valid, then a list of values of total claim for each queried subclass is returned in the order of the subclasses in the list.

3.2 GetMaxClaim(policy subclass)

Input- Policy Subclass.

Output- If the policy subclass is valid, then the value of the maximum claim of that policy subclass is returned.

3.3 GetMinClaim(policy subclass)

Input- Policy Subclass.

Output- If the policy subclass is valid, then the value of the minimum claim of that policy subclass is returned.

3.4 GetAverageClaim(policy subclass)

Input- Policy Subclass.

Output- If the policy subclass is valid, then the value of the average of claims of that policy subclass is returned.

4. Search:

4.1 SearchCustomerByName(name)

Input- Name.

Output- A data tuple of the data of all people who are customers or their dependents and have a partial or complete match with the queried name is returned.

4.2 SearchEmployeeByName(name)

Input- Name.

Output- A data tuple of the data of all people who are employees or their dependents and have a partial or complete match with the queried name is returned.

4.3 SearchTPAByName(name)

Input- Name.

Output- A data tuple of the data of all TPAs that have a partial or complete match with the queried name is returned.

5. Analysis:

5.1 GenerateEmployeePerformanceReport(department no, serial no)

Input- Employee Department Number, Employee Serial Number.

Output- If the employee id is valid, a report of all the work done by the employee like sales, supervision, processing, etc. is generated.

5.2 GenerateClaimedPolicyReports(policy subclasses)

Input- List of policy subclasses.

Output- If the policy subclasses are valid, then a report of all policies claimed that belong to these subclasses is generated. This report includes the claimed amount, date of claim, investigating TPA, etc.

5.3 GenerateCustomerReport(aadhar number)

Input- Customer Aadhar number.

Output- If the Aadhar number is valid, then a report of the customer is generated. This report includes policies bought by customers, monthly premium paid, amount claimed, etc.