Tommy Olokesusi
Lina Bazzi
Evan Todorov
Mariam Obeid
Yusif Affara
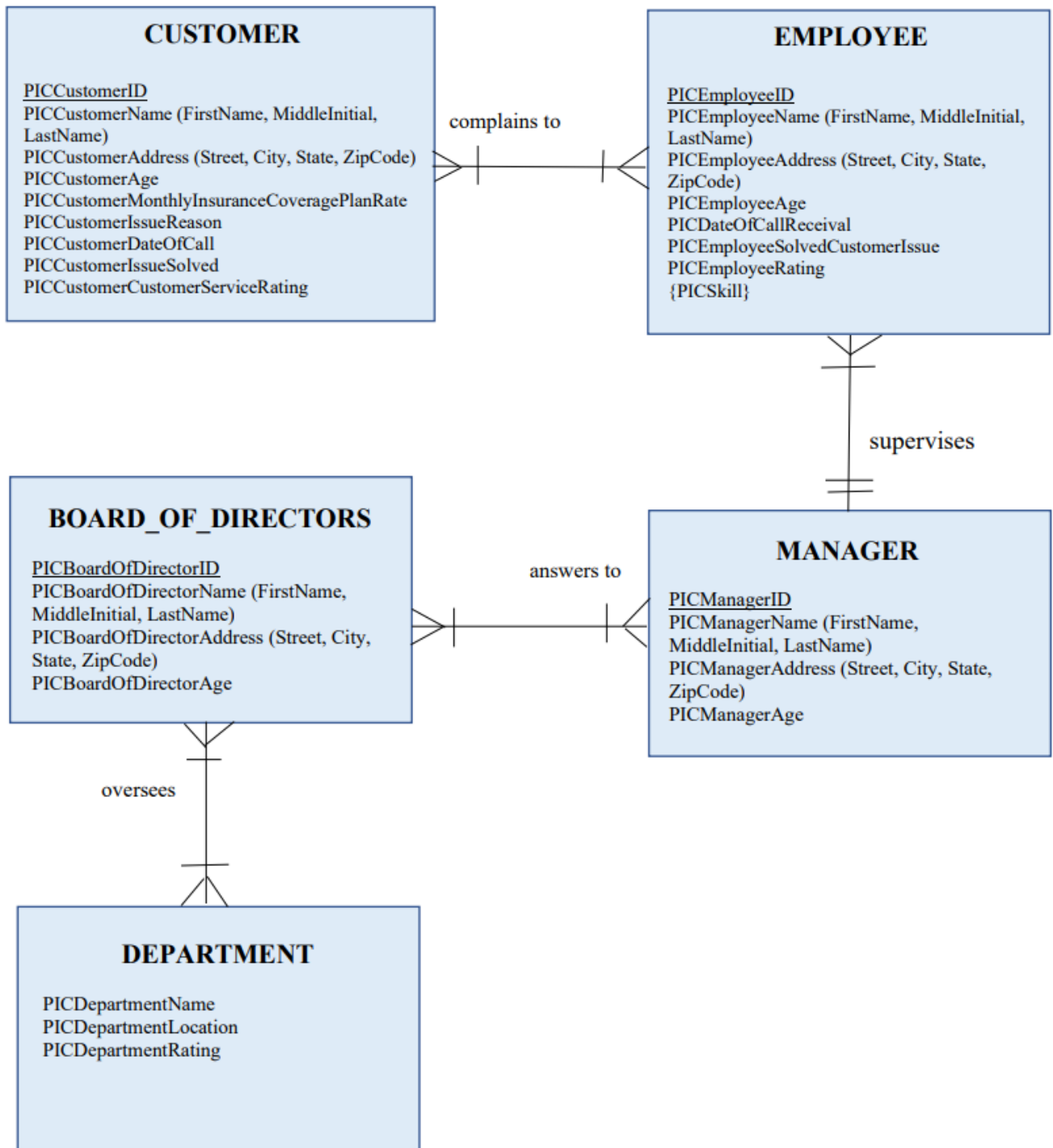**Klevis Todi**

—---

      Progressive Insurance Company (PIC) was started in 1937 by Joseph Lewis and Jack Green. Their goal was to provide vehicle owners with protection and security. Progressive Insurance Company (PIC) offers customers a variety of services. PIC keeps track of the customers it offers service to as well as its employees, managers, board of directors, and the whole marketing department who manage and oversee the affairs of providing the best quality customer service to their clients/customers. When creating our database on PIC the information that was needed to support our system was information about their customers, employees, managers, board_of_directors, and department. We gathered information on what each category does and how each one works with one another to satisfy the customers' needs.

      The organization that will utilize our group's database system is an insurance company. Our group will be designing a database for the marketing department of Progressive Insurance Corporation, specifically its customer service. In our case, the company's main goal is to improve its customer service quality, as there have been numerous reports on the 'Better Business Bureau' (a non-profit organization that provides information on businesses and manages consumer complaints about firms) that it is lacking in providing its customers with adequate, helpful, and thorough information regarding their inquiries. From unexpectedly ending customers' coverages to being reluctant to solve their problems and a few others, Progressive Insurance evidently has an ongoing issue with maintaining and retaining its customer base, and we hope that our database system design can possibly help with eliminating such. Thus, our database will be focused on analyzing and displaying information about a sample set of 50 customers who made an inquiry to Progressive Insurance's customer service hotline and measuring how well their needs were met, for the month of February 2023 alone.

E-R Diagram/Conceptual Data Model:

CUSTOMER

PICCustomerID
PICCustomerName (FirstName, MiddleInitial, LastName)
PICCustomerAddress (Street, City, State, ZipCode)
PICCustomerAge
PICCustomerMonthlyInsuranceCoveragePlanRate
PICCustomerIssueReason
PICCustomerDateOfCall
PICCustomerIssueSolved
PICCustomerCustomerServiceRating

complains to

EMPLOYEE

PICEmployeeID
PICEmployeeName (FirstName, MiddleInitial, LastName)
PICEmployeeAddress (Street, City, State, ZipCode)
PICEmployeeAge
PICDateOfCallReceival
PICEmployeeSolvedCustomerIssue
PICEmployeeRating
{PICSkill}

supervises

BOARD_OF_DIRECTORS

PICBoardOfDirectorID
PICBoardOfDirectorName (FirstName, MiddleInitial, LastName)
PICBoardOfDirectorAddress (Street, City, State, ZipCode)
PICBoardOfDirectorAge

answers to

MANAGER

PICManagerID
PICManagerName (FirstName, MiddleInitial, LastName)
PICManagerAddress (Street, City, State, ZipCode)
PICManagerAge

oversees

DEPARTMENT

PICDepartmentName
PICDepartmentLocation
PICDepartmentRating

With the information gathered we were able to construct an entity-relationship diagram that included attributes of customers include PICCustomerID, PICCustomerName (composed of FirstName, MiddleInitial, and LastName), PICCustomerAddress (composed of Street, City, State, and ZipCode), PICCustomerAge, PICCustomerMonthlyInsuranceCoveragePlanRate, PICCustomerIssueReason, PICCustomerDateOfCall, PICCustomerIssueSolved, and PICCustomerCustomerServiceRating. Attributes of employees include PICEmployeeID, PICEmployeeName (composed of FirstName, MiddleInitial, and LastName), PICEmployeeAddress (composed of Street, City, State, and ZipCode), PICEmployeeAge, PICDateOfCallReceival, PICEmployeeSolvedCustomerIssue, and PICEmployeeRating. Attributes of managers include PICManagerID, PICManagerName (composed of FirstName, MiddleInitial, and LastName), PICManagerAddress (composed of Street, City, State, and ZipCode), and PICManagerAge. Attributes of board_of_directors include PICBoardOfDirectorID, PICBoardOfDirectorName (composed of FirstName, MiddleInitial, and LastName), PICBoardOfDirectorAddress (composed of Street, City, State, and ZipCode), and PICBoardOfDirectorAge. Many employees have more than one skill/area of expertise. Attributes of the department include PICDepartmentName, PICDepartmentLocation, and PICDepartmentRating. Customers can take their complaints to more than one employee, but must take them to at least one employee. Each employee can have multiple customers, but must have at least one customer. A single manager supervises each employee, and managers can manage multiple employees. Managers can answer to many directors, but each manager must answer to at least one director. Board of directors must oversee the affairs of the whole marketing department, but at least one member of the board of directors must oversee the department.

From the information collected we were able to create a set of normalized relations for our database, in the third normal form. Which means no transitive dependencies and partial functional dependencies exist. We bolded the primary keys so they can be easily identified. There are no foreign keys included. Our CUSTOMER relations includes (**PICCustomerID**, PICCustomerName, PICCustomerAddress, PICCustomerMonthlyInsuranceCoveragePlanRate, PICCustomerIssueReason, PICCustomerDateOfCall, PICCustomerIssueSolved, PICCustomerCustomerServiceRating). For our EMPLOYEE relations we included (**PICEmployeeID**, PICEmployeeName, PICEmployeeAddress, PICDateOfCallReceival, PICEmployeeSolvedCustomerIssue, PICEmployeeRating). Our MANAGER relations consist of (**PICManagerID**, PICMannagerName, PICManagerAddress, PICManagerAge). The BOARD OF DIRECTORS relations contains (**PICDirectorID**, PICDirectorName, PICDirectorAddress, PICDirectorAge). Lastly, the DEPARTMENT relations consists of (**PICDeapartmentName,** PICDepartmentLocation, PICDepartmentRating). Looking at the data that has been added to the normalized relations, we decided to denormalize the employee relations. We thought it

would be best to add **PICCustomerID, PICManagerID** as foreign keys. The justification behind the decision to denormalize this relation is to ensure that each customer is assigned to each employee to resolve their respective claims, and that each employee is assigned to each manager to oversee their performances, all one-to-one each.

To guarantee the integrity of our database, referential integrity constraints were identified. The first constraint identified is, if an employee solved a customer issue, then the PICEmployeeID in the employee table should match the PICEmployeeSolvedCustomerIssue in the customer table. The second constraint identified is, if an employee solved a customer issue, then the PICDateOfCallReceival in the employee table should match the PICCustomerDateOfCall in the customer table. The third constraint identified is, If a customer rates an employee in the PICCustomerCustomerServiceRating of the customer table, then such specific rating should match that of the employee in the PICEmployeeRating of the employee table, where each employee gets to sincerely rate themselves. The fourth constraint identified is, if a customer rates an employee in the PICCustomerCustomerServiceRating of the customer table, then such specific rating should also match that of the department in the PICDepartmentRating of the department table, where the marketing department gets to rate the company's overall customer service quality, as suggested by each customer. The fifth constraint identified is if a customer issue is marked as solved, then the PICCustomerIssueSolved field should be set to true/yes. The sixth constraint identified is, if a customer is associated with a department, then the PICDeapartmentName in the department relation should match the department of the corresponding employee who solved the issue. Lastly, the seventh constraint identified is, if a customer is associated with a manager, then the PICManagerID in the manager relation should match the manager of the corresponding employee who solved the issue. Each one of these constraints follows one another. This ensures that the information entered is correct and accurate. For the integrity controls, we defined default values, range controls, and null value controls in all the fields for each table in the database.

The indexes needed for the database are the primary index and the secondary index. For customers, its attributes are PICCustomerID with a primary key resulting in a primary index. Also, PICCustomerDateOfCall and PICCustomerIssueSolved with a faster retrieval resulting in a secondary index. The attributes for employees are PICEmployeeID with a primary key resulting in a primary index. In addition to, PICDateOfCallReceival and PICEmployeeSolvedCustomerIssue which have a faster retrieval resulting in a secondary index. The attributes for manager are PICManagerID with a primary key resulting in a primary index. The attributes of the Board of Directors are PICDirectorID with a primary key resulting in a primary index.

In conclusion, this database helps ensure that Progressive Insurance Company provides the best care and services for its customers. An important matter to take into consideration

when working with customers is to make sure that their needs are satisfied. In this paper a database of PIC customers, employees, managers, board of directors, and department were created to build a relationship between the customers and the company.

Appendices

CUSTOMER:

Attributes	Data Types
PICCustomerID	integer
PICCustomerName	short text
PICCustomerAddress	short text
PICCustomerMonthlyInsuranceCoveragePlanRate	short text
PICCustomerIssueReason	short text
PICCustomerDateOfCall	date/time
PICCustomerIssueSolved	yes/no
PICCustomerCustomerServiceRating	byte

EMPLOYEE:

Attributes	Data Types
PICEmployeeID	integer
PICEmployeeName	short text
PICEmployeeAddress	short text
PICDateOfCallReceival	date/time
PICEmployeeSolvedCustomerIssue	yes/no
PICEmployeeRating	byte

MANAGER:

Attributes	Data Types
PICManagerID	integer
PICMannagerName	short text
PICManagerAddress	short text
PICManagerAge	byte

BOARD OF DIRECTORS:

Attributes	Data Types
PICDirectorID	integer
PICDirectorName	short text
PICDirectorAddress	short text
PICDirectorAge	byte

DEPARTMENT:

Attributes	Data Types
PICDeapartmentName	short text
PICDepartmentLocation	short text
PICDepartmentRating	integer

Table	Attribute(s)	Reason for Index	Type of Index
CUSTOMER	PICCustomerID	Primary Key	Primary Index
EMPLOYEE	PICEmployeeID	Primary Key	Primary Index
MANAGER	PICManagerID	Primary Key	Primary Index
BOARD OF DIRECTORS	PICDirectorID	Primary Key	Primary Index
CUSTOMER	PICCustomerDateOfCall	Faster Retrieval	Secondary Index
CUSTOMER	PICCustomerIssueSolved	Faster Retrieval	Secondary Index
EMPLOYEE	PICDateOfCallReceival	Faster Retrieval	Secondary Index
EMPLOYEE	PICEmployeeSolvedCustomerIssue	Faster Retrieval	Secondary Index

Integrity controls

CUSTOMER:
- PICCustomerID: default value: 0, must be unique and not null, range control = 1000 - 1999
- PICCustomerName: no default value, no range control, not null
- PICCustomerAddress: , no default value, no range control, not null
- PICCustomerMonthlyInsuranceCoveragePlanRate: no default value, no range control, not null
- PICCustomerIssueReason: no default value, no range control, not null
- PICCustomerDateOfCall: not null, check for valid date format
- PICCustomerIssueSolved: no default value, no range control, not null
- PICCustomerCustomerServiceRating: default value: 0, range control = 1 - 5, not null

EMPLOYEE:
- PICEmployeeID: default value: 0, range control: 2000 - 2999, must be unique and not null
- PICEmployeeName: not null, check for valid name format (e.g. no special characters)
- PICEmployeeAddress: no default value, no range control, not null
- PICDateOfCallReceival: no default value, no range control, not null, check for valid date format
- PICEmployeeSolvedCustomerIssue: not null
- PICEmployeeRating: default value: 0, check for valid rating values, range control = 1 - 5

MANAGER:
- PICManagerID: default value: 0, must be unique and not null, range control = 3000 - 3999
- PICMannagerName: not null, check for valid name format (e.g. no special characters)
- PICManagerAddress: not null
- PICManagerAge: default value: 0, not null, check for non-negative values,
             range control = 18 - 65

BOARD OF DIRECTORS:
- PICDirectorID: default value: 0, must be unique and not null, range control = 4000 - 4999
- PICDirectorName: not null, check for valid name format (e.g. no special characters)
- PICDirectorAddress: not null
- PICDirectorAge: default value: 0, not null, check for non-negative values,
             range control = 30 - 65

DEPARTMENT:
- PICDeapartmentName: not null, check for valid department name format (e.g. no special characters)
- PICDepartmentLocation: not null
- PICDepartmentRating: default value:0, check for valid rating values, range control: 1 - 5

Database Implementation

Test Data:
List of test data used to test database's data integrity controls and referential integrity constraints:

- Data integrity control:
  - CUSTOMER table:
    - PICCustomerID: "950" - was used to test the field's range control of permitted numbers between 1000 and 1999, worked successfully as the user was not allowed to advance to other columns/rows without inputting a number within the specified range, validation message prompt: "Number is not between 1000 and 1999!"
    - PICCustomerCustomerServiceRating: "7" - was used to test the field's range control of permitted numbers between 1 and 5, worked successfully as the user was not allowed to advance to other columns/rows without inputting a number within the specified range, validation message prompt: "Number is not between 1 and 5!"
    - *other fields in this table do not have definitive and measurable data integrity controls, therefore only the above two fields will be presented for justification and to describe their purposes*

  - EMPLOYEE table:
    - PICEmployeeID: "1500" - was used to test the field's range control of permitted numbers between 2000 and 2999; worked successfully as the user was not allowed to advance to other columns/rows without inputting a number within the specified range, validation message prompt: "Number is not between 2000 and 2999!"

- ○ PICEmployeeRating: "6" - was used to test the field's range control of permitted numbers between 1 and 5, worked successfully as the user was not allowed to advance to other columns/rows without inputting a number within the specified range, validation message prompt: "Number is not between 1 and 5!"
  - ○ *other fields in this table do not have definitive and measurable data integrity controls, therefore only the above tow fields will be presented for justification and to describe their purposes*

- ● Referential integrity control:
  - - CUSTOMER table:
  - ○ PICCustomerID: "1111" - was used to ensure and test that the value in each row of this field in this table matches that of the same corresponding row of the same field name in the EMPLOYEE table, worked successfully as the user was not allowed to advance to other columns/rows without being warned by access that such invalid value cannot be added or changed because a related record is required in the EMPLOYEE table.

  - - EMPLOYEE table:
  - ○ PICCustomerID: "1045" - was used to ensure and test that the value in each row of this field in this table matches that of the same corresponding row of the same field name in the CUSTOMER table, worked successfully as the record in the CUSTOMER table also changed/updated automatically to the same value in the EMPLOYEE table after the user saves and closes the EMPLOYEE table.
  - ○ PICManagerID: "3023" - was used to ensure and test that the value in each row of this field in this table matches that of the same corresponding row of the same field name in the MANAGER table, worked successfully as the record in the MANAGER table also changed/updated automatically to the same value in the EMPLOYEE table after the user saves and closes the EMPLOYEE table.

  - - MANAGER table:
  - ○ PICManagerID: "3333" - was used to ensure and test that the value in each row of this field in this table matches that of the same corresponding row of the same field name in the EMPLOYEE table, worked successfully as the user was not allowed to advance to other columns/rows without being warned by access that such invalid value cannot be added or changed because a related record is required in the EMPLOYEE table.

Queries:

- Customer First Query: This query was designed and executed to request information from the database based on every customer of Progressive Insurance Company in this sample data and their issues' reasons reported to the company's customer service hotline, presented using their IDs, names, dates of calls, and the issues' reasons themselves.
- Customer Second Query: This query was designed and executed to request information from the database based on every customer of Progressive Insurance Company in this sample data and their issues' reasons reported to the company's customer service hotline, and to essentially find out if each issue was resolved or not, presented using their IDs, names, dates of calls, and a question representing the results of asking if the customers believed their issues were solved or not.
- Customer Third Query: This query was designed and executed to request information from the database based on every customer of Progressive Insurance Company in this sample data and their issues' reasons reported to the company's customer service hotline, and to find out what each customer's customer service rating was after their claims were attended to, presented using their IDs, names, dates of calls, and each customer's customer service ratings.
- Employee First Query: This query was designed and executed to request information from the database based on every employee of Progressive Insurance Company in this sample data and the measure of if the customer issues were resolved or not. It is to be noted that the answers are clearly not sincere and accurate (not matching with that of the customers), therefore, this database serves only as a model for measuring Progressive Insurance's customer service performance and making sure the customer's judgment of the company's customer service quality aligns with that of the company's employees as well, presented using their IDs, names, dates of calls, and a question representing the results of asking if the employees believed the customers' issues were solved or not.
- Employee Second Query: This query was designed and executed to request information from the database based on every employee of Progressive Insurance Company in this sample data and the customer service ratings each employee gave themselves. It is to be noted that the ratings are clearly not sincere and accurate (not matching with that of the customers), therefore, this database serves only as a model for measuring Progressive Insuracnce's customer service performance and making sure the customer's judgment of the company's customer service quality aligns with that of the company's employees as well, presented using their IDs, names, dates of calls, and each employee's rating of themselves.