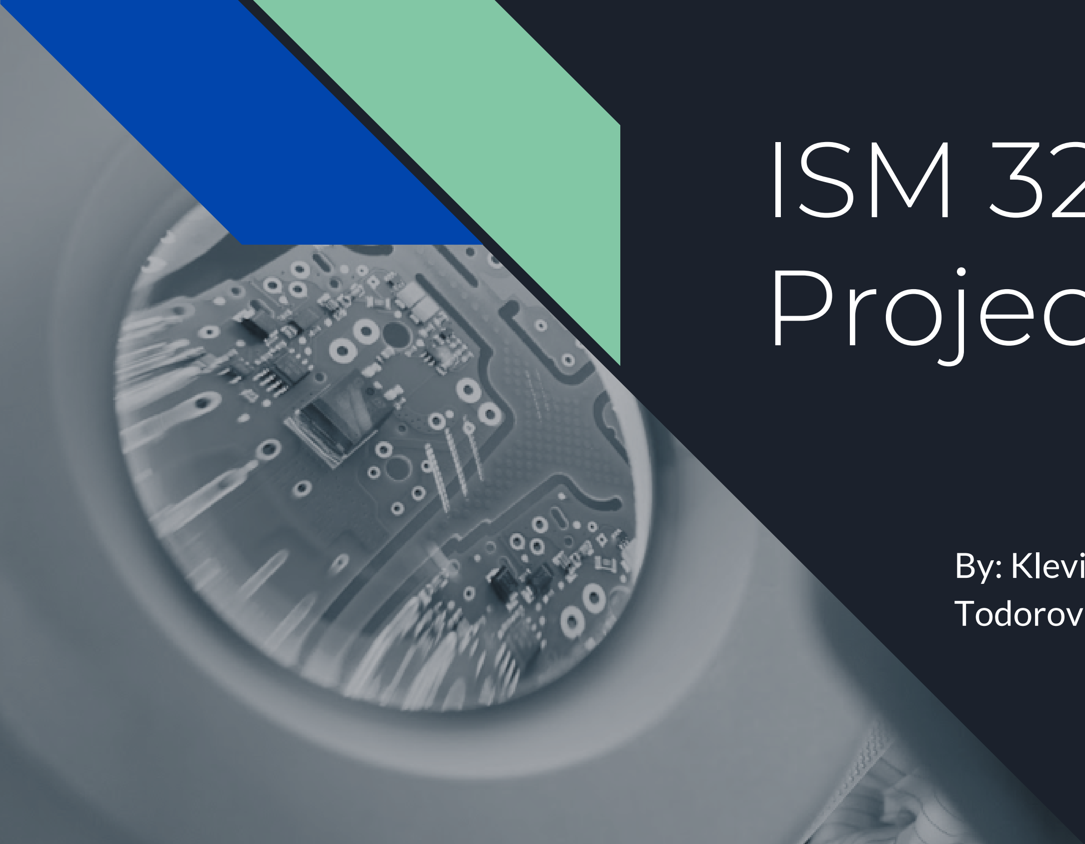


ISM 321 Group Project

By: Klevis Todi, Yusif Affara, Tommy Olokesusi, Lina Bazzi, Evan Todorov , Mariam Obeid

Project Overview

- An insurance company, Progressive Insurance Corporation, will utilize the group's database system.
- The database is being designed for the marketing department of Progressive Insurance Corporation, specifically its customer service.
- The company's primary goal is to improve the quality of its customer service, as there have been numerous complaints about inadequate, unhelpful, and incomplete information provided to customers.
- Progressive Insurance has a problem with maintaining and retaining its customer base, which the database system aims to help eliminate.
- The database system will analyze and display information about a sample set of 80 customers who made an inquiry to Progressive Insurance's customer service hotline in February 2023.
- The system will measure how well their needs were met.

Data To Be Stored In Our Database System

-CUSTOMER data (attributes/columns/fields)

Customer Call/Email ID (number) - field

Customer Name (first, middle, last) - field

Customer Address (street, city, state, zip code) - field

Customer Age (date of birth) - field

Customer Monthly insurance coverage plan rate - field

Customer Issue’s Reason - field

Customer Date of Call - field

Customer Issue Solved (yes/no checkmark) - field

Customer Customer Service Rating (out of 5) - field

-EMPLOYEES data

Employee ID (number) - field

Employee Name (first, middle, last) - field

Employee Address (street, city, state, zip code) - field

Employee Age (date of birth) - field

Employee Date of Call Receival - field

Employee Solved Customer Issue (yes/no checkmark) - field

Employee Rating (out of 5) - field

-MANAGERS data

Manager ID (number) - field

Manager Name (first, middle, last) - field

Manager Address (street, city, state, zip code) - field

Manager Age (date of birth) - field

-BOARD_OF_DIRECTORS data

Board of Director ID (number) - field

Board of Director Name (first, middle, last) - field

Board of Director Address (street, city, state, zip code) - field

Board of Director Age (date of birth) - field

-DEPARTMENT data

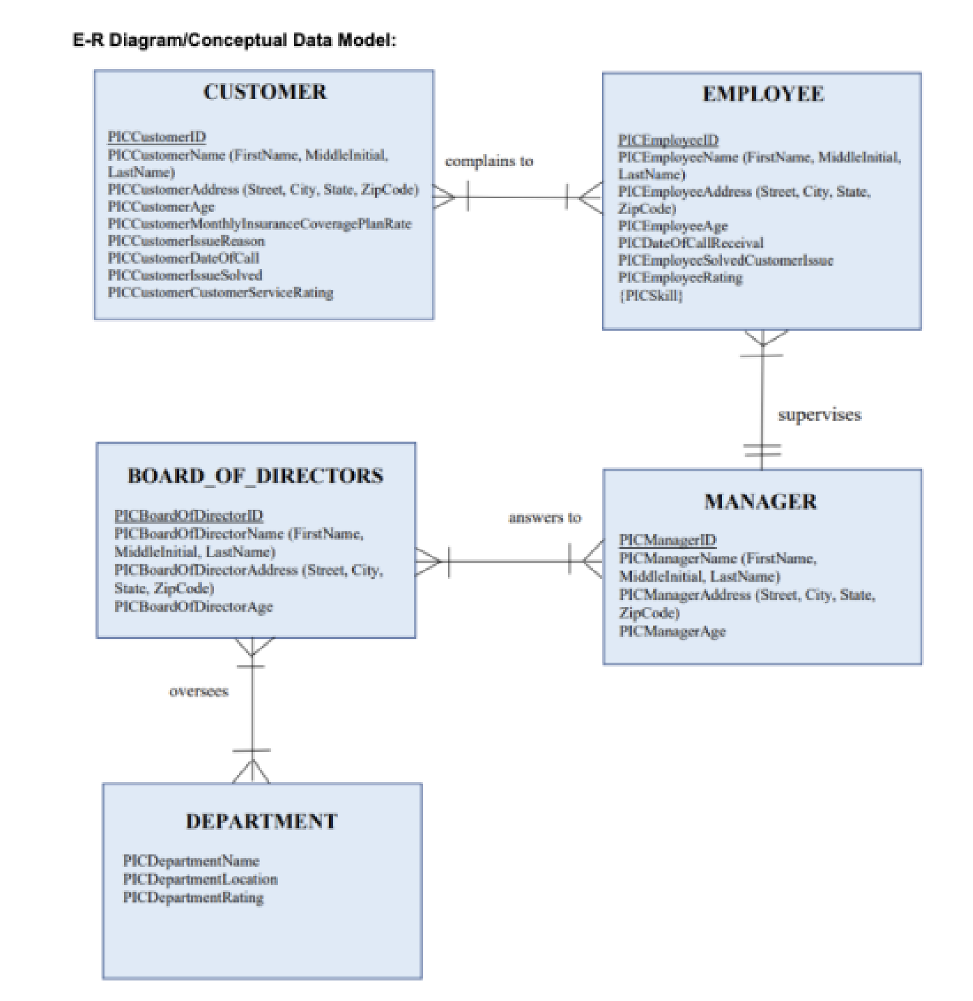
Department Name - field

Department Location - field

Department Rating (number) - field

E-R Diagram/Conceptual Data Model Scenario

- Progressive Insurance Company (PIC) offers customer service to various customers.
- 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 all manage and oversee the affairs of providing the best quality customer service to their clients/customers.
- 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.



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

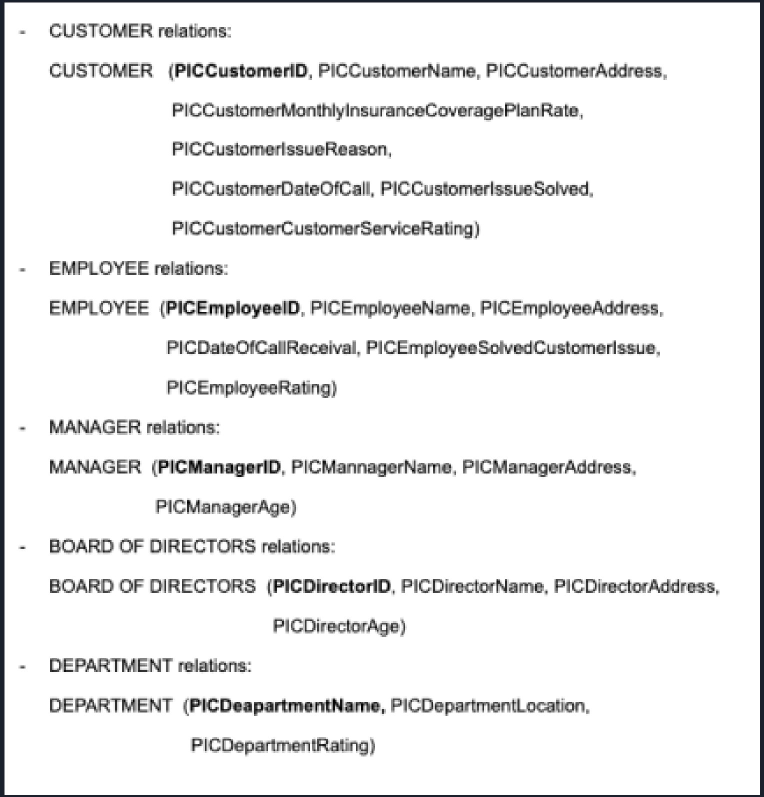
PICManagerID
PICManagerName (FirstName, MiddleInitial, LastName)
PICManagerAddress (Street, City, State, ZipCode)
PICManagerAge

oversees

DEPARTMENT

PICDepartmentName
PICDepartmentLocation
PICDepartmentRating

Logical Database Design



- CUSTOMER relations:

CUSTOMER (**PICCustomerID**, PICCustomerName, PICCustomerAddress,
PICCustomerMonthlyInsuranceCoveragePlanRate,
PICCustomerIssueReason,
PICCustomerDateOfCall, PICCustomerIssueSolved,
PICCustomerCustomerServiceRating)

- EMPLOYEE relations:

EMPLOYEE (**PICEmployeeID**, PICEmployeeName, PICEmployeeAddress,
PICDateOfCallReceival, PICEmployeeSolvedCustomerIssue,
PICEmployeeRating)

- MANAGER relations:

MANAGER (**PICManagerID**, PICMannagerName, PICManagerAddress,
PICManagerAge)

- BOARD OF DIRECTORS relations:

BOARD OF DIRECTORS (**PICDirectorID**, PICDirectorName, PICDirectorAddress,
PICDirectorAge)

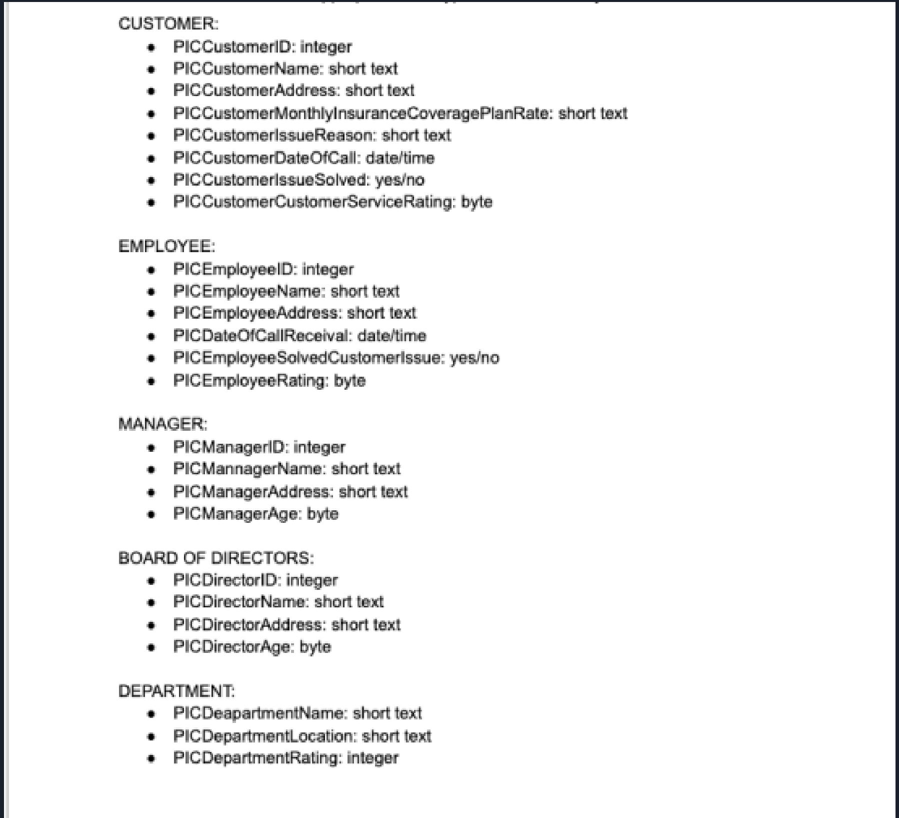
- DEPARTMENT relations:

DEPARTMENT (**PICDeapartmentName,** PICDepartmentLocation,
PICDepartmentRating)

Logical Database Design

- All relations are in third normal form (no transitive dependencies and partial functional dependencies exist)
- Primary keys are identified in bold and no foreign keys exist.

Physical Database Design



CUSTOMER:

- PICCustomerID: integer
- PICCustomerName: short text
- PICCustomerAddress: short text
- PICCustomerMonthlyInsuranceCoveragePlanRate: short text
- PICCustomerIssueReason: short text
- PICCustomerDateOfCall: date/time
- PICCustomerIssueSolved: yes/no
- PICCustomerCustomerServiceRating: byte

EMPLOYEE:

- PICEmployeeID: integer
- PICEmployeeName: short text
- PICEmployeeAddress: short text
- PICDateOfCallReceival: date/time
- PICEmployeeSolvedCustomerIssue: yes/no
- PICEmployeeRating: byte

MANAGER:

- PICManagerID: integer
- PICMannagerName: short text
- PICManagerAddress: short text
- PICManagerAge: byte

BOARD OF DIRECTORS:

- PICDirectorID: integer
- PICDirectorName: short text
- PICDirectorAddress: short text
- PICDirectorAge: byte

DEPARTMENT:

- PICDeapartmentName: short text
- PICDepartmentLocation: short text
- PICDepartmentRating: integer

Determining data integrity controls for the fields in your database

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

Determining data integrity controls for the fields in your database

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

Determining data integrity controls for the fields in your database

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 value, range control = 18 - 65

Determining data integrity controls for the fields in your database

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

Determining data integrity controls for the fields in your database

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

Referential Integrity Constraints

1. Referential integrity constraint exists between the PICCustomerID field (primary key) in the customer table and the PICCustomerID field (foreign key) in the employee table.
2. Referential integrity constraint exists between the PICManagerID field (primary key) in the manager table and the PICManagerID field (foreign key) in the employee table.

Set of normalized relations for your database.

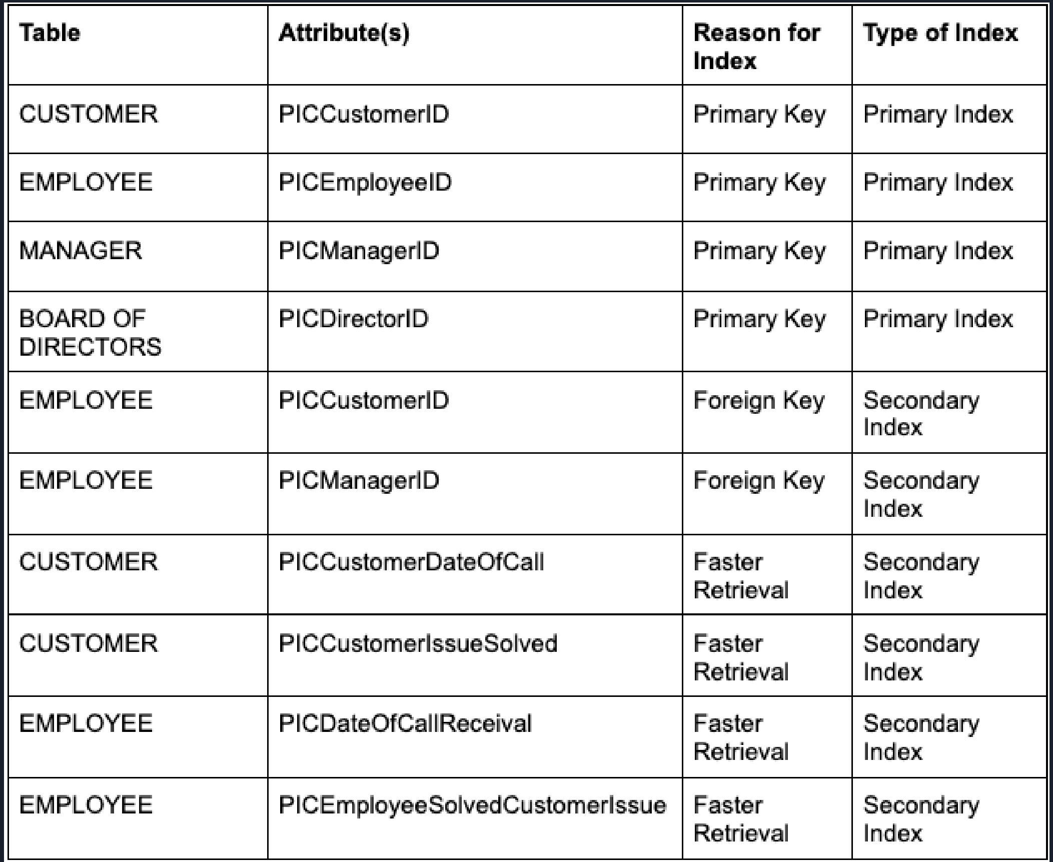
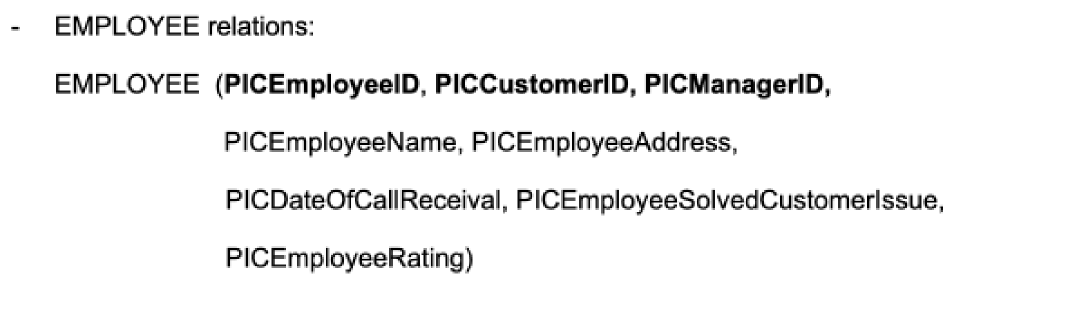


Table	**Attribute(s)**	**Reason for Index**	**Type of Index**
CUSTOMER	PICCustomerID	Primary Key	Primary Index
EMPLOYEE	PICEmployeeID	Primary Key	Primary Index
MANAGER	PICManagerID	Primary Key	Primary Index
BOARD OF DIRECTORS	PICDirectorID	Primary Key	Primary Index
EMPLOYEE	PICCustomerID	Foreign Key	Secondary Index
EMPLOYEE	PICManagerID	Foreign Key	Secondary 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

Denormalization

- 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.



- EMPLOYEE relations:

EMPLOYEE (**PICEmployeeID, PICCustomerID, PICManagerID,**
PICEmployeeName, PICEmployeeAddress,
PICDateOfCallReceival, PICEmployeeSolvedCustomerIssue,
PICEmployeeRating)

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 succesfully as 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 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*

Database Implementation (cont'd.)

- EMPLOYEE table:
  - ○ PICEmployeeID: "1500" - was used to test the field's range control of permitted numbers between 2000 and 2999; worked successfully as 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 1and 5, worked successfully as 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*

Database Implementation (cont’d.)

- 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 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 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 user saves and closes the EMPLOYEE table.

Database Implementation (cont’d.)

- 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 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.

Database Implementation (cont'd.) 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.

Database Implementation (cont’d.)

- 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 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 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 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 each employee’s rating of themselves.

Thank you!