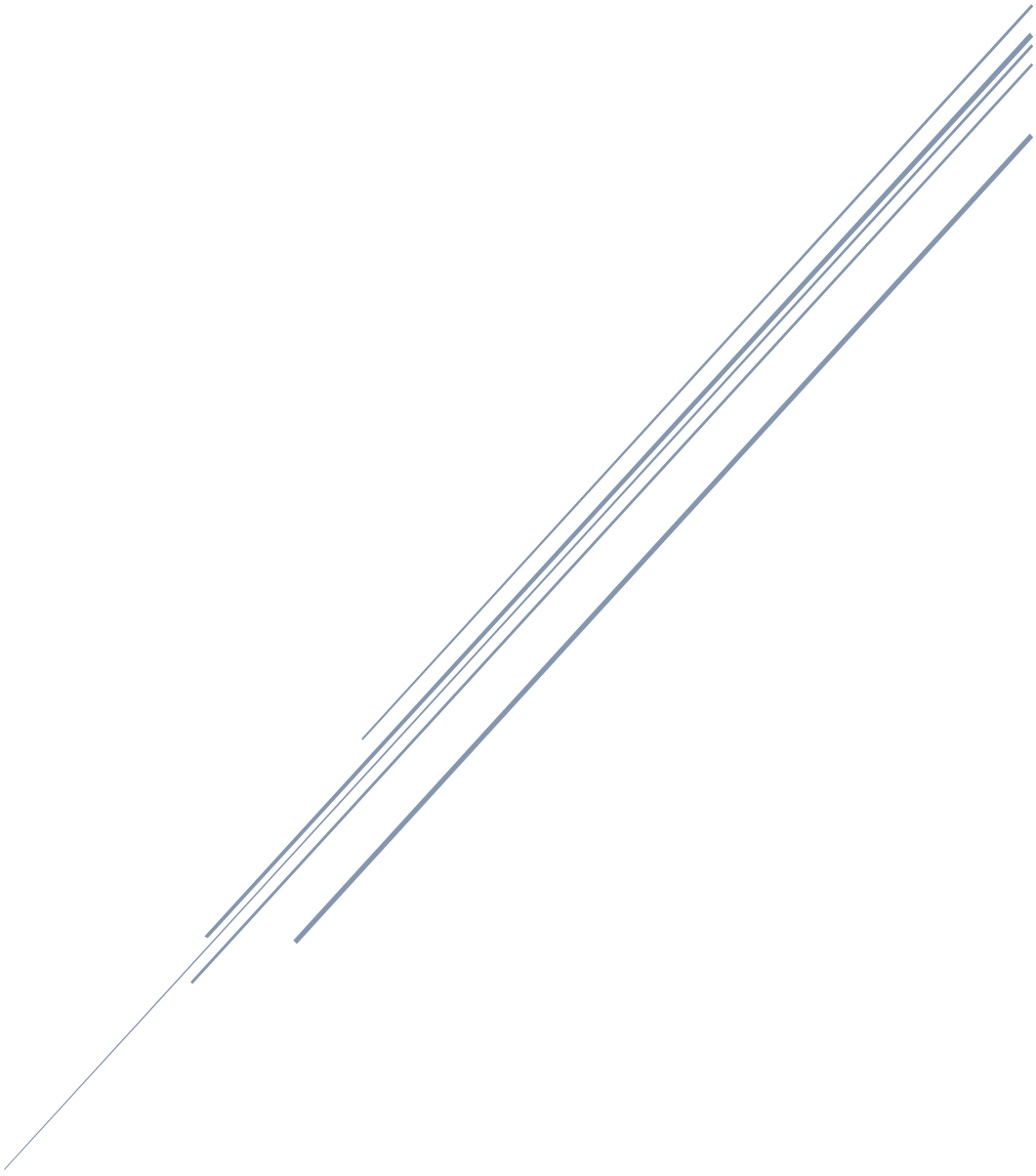
DBD281 PROJECT

Elmar Jacobs 576901



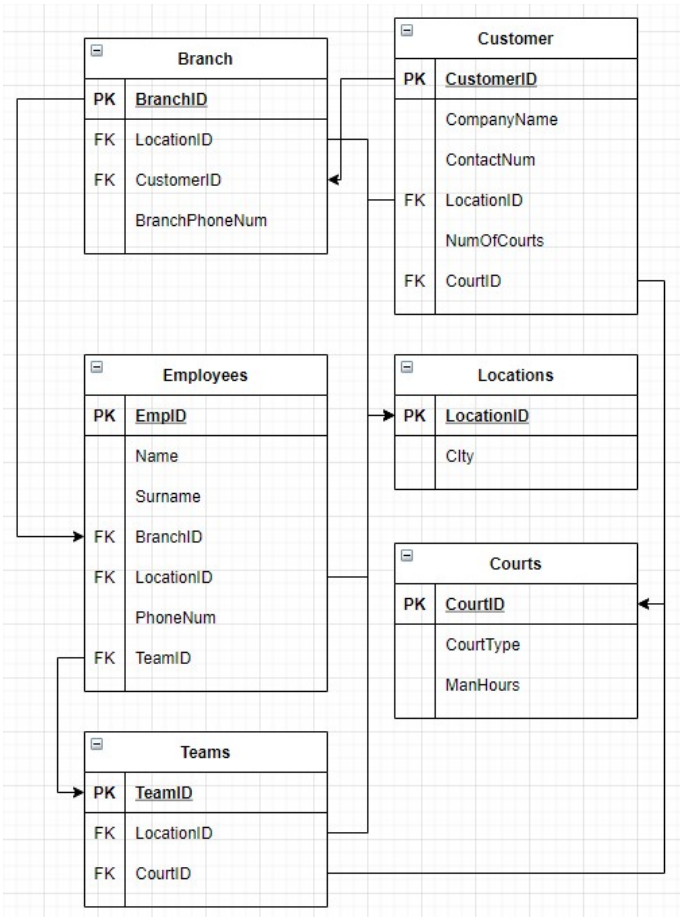
PE campus
2021/03/07

Introduction and background:

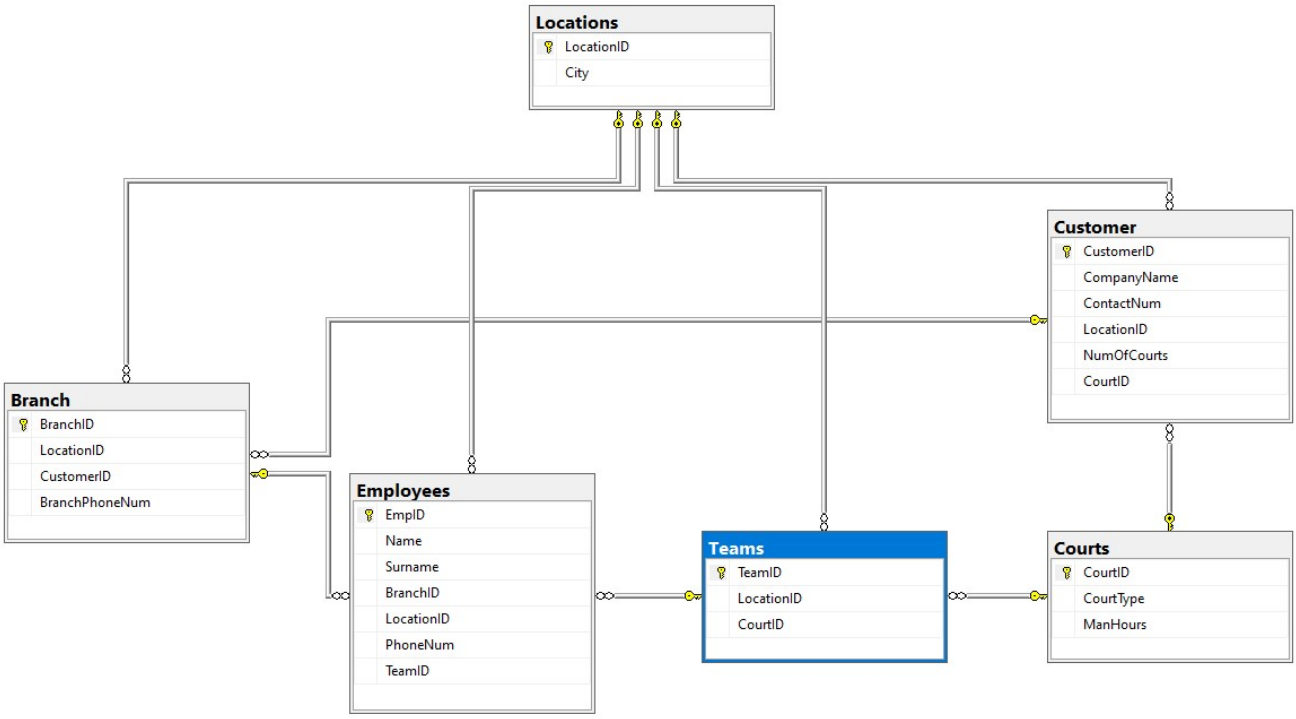
For the scenario that I have chosen I will make a new database for Synsport. Synsport is a company based in South Africa that specialises in sport court resurfacing, and they do a very wide range of courts, from tennis to golf, even gym flooring. They offer their service to normal residences with courts and they also do professional courts. Synsport does hard construction of courts on the side too. Synsport looks to be a pretty small company with only three branches, one in Cape Town, one in Johannesburg and one in the Garden Route area. This is why I chose to do their database, a company like this could massively benefit from having a proper database that they can store all their employee's information alongside their customer's. A company like this works in teams, certain teams do certain court surfaces on separate areas, the company could greatly benefit from having all their teams' data on their database while it is linked to the team's area and every employee in the team.

ERD:

This is the initial ERD that I drew up before creating the database:



This is the final ERD/Database diagram that was created in SQL Server:



Normalization:

Normalization of my database started by making sure that each table cell had only a single value, and by doing this each record was unique. Every instance of data could link together in a way, but no double values were inputted. Thus achieving 1NF.

Primary and foreign keys were added to tables so that they could link in with each other while keeping data contained in the tables straight forward and to the point without having any data in that could relate to something off topic or different, for example I split the Employee table by creating a Teams table, by doing this the employee has a TeamID and all the information about the team can be found in the Teams table keeping the Employees table only about the employees. Thus achieving 2NF.

Tables were split up even further to achieve 3NF, for example I split the Employees table by creating a Branch, Locations and Teams table. All these information directly relate to the Employee's details but keeps the Employee table neat and the data about his location can be found in a different table, all liked with primary and foreign keys. Thus achieving 3NF.

List of objects:

Tables:
-Branch, table created to store branch information.
-Customer, table created to store all information about customers.
-Employees, table created to store all information about Employees.
-Locations, table to store different locations and be linked with an ID.
-Teams, table to store the team's location and what courts they work on.
-Courts, list of all courts that the company can resurface and how long the actual resurfacing takes to complete.

Stored Procedures:
-sp_ShowBranchEmp, this procedure shows the details of an employee working at a certain branch, the branch is given by the user.
-sp_ShowLocationTeams, shows the locations and how many teams are in that location.
-sp_CourtCount, shows how many courts have been resurfaced by the company in total.

Views:
-vw_CustFrmCPT, shows all customers from Cape Town.
-vw_CustFrmJHB, shows all customers from Johannesburg.
-vw_CustFrmGDR, shows all customers from Garden Route area.
-vw_CustomerNames, shows all customer names, phone number and locations.
-vw_CusCourts, shows how many courts a customer has of a certain kind.

Triggers:
-tr_EmpChange, prints that something has changed when a change is made to employees table.
-tr_CusChange, prints that something has changed when a change is made to customer table.

Queries questions in plain language:

In the business model that Synsport has they can hugely benefit from having all of their customer's or employee's data/information.
For instance, a manager might need to urgently contact a certain employee, the answer to this would be to just run the sp_ShowBranchEmp procedure and he/she will be able to find all the information relevant to their employees at a certain branch.
Another problem that can be solved by the database is timing. When a customer wants to resurface their court the company can see exactly how long it takes to resurface a court of that kind. Thus they can give the customer an accurate timeframe for when they will have completed their work.
The database makes it so easy to do anything admin in the Synsport company, if a new branch opens its details can simply be inserted into the database and all queries created can be used to find or select grouped data for research.