Theory Assignment#4

**4.1 E.R-Diagram with one case study:**

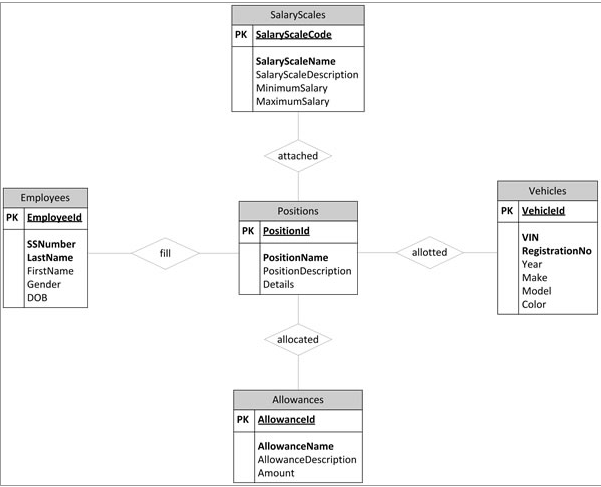
This is the first of three case studies that are used to guide the reader through the six steps outlined in Six-Step Relational Database Design. Some of the outputs of the six step database design process are depicted here.

**Scenario:**

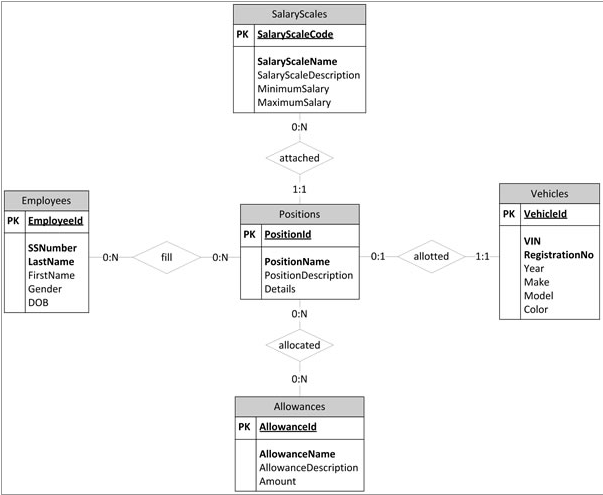
A small accounting firm wants a simple HR application that will help it to keep track of its employees, their positions, allowances, salary scales, and which company vehicles their employees drive.   
The application must keep track of all the positions at the firm, the employees filling these positions, the allowances for these positions, the salary scales for these positions, and the company vehicles assigned to these positions.

**Entity-Relationship diagrams**

Below is the Simplified Entity-Relationship diagram that is output from Step 3 of the six step database design process as described in Six-Step Relational Database Design:

****

Below is the Detailed Entity-Relationship diagram that is output from Step 5 of the six step database design process as described in Six-Step Relational Database Design:

****

**4.2 Design:**

**4.2.1 Functional Design:**

It is form of constraint (hence, part of the schema) finding them is part of the database design that is also used in normalizing the relations.

If two tuples agree on the attributes

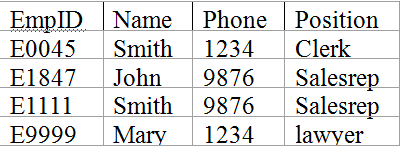
A , A , … An

then they must also agree on the attributes

B , B , … Bn

A , A , … An  B , B , … Bn

Examples



* EmpID Name, Phone, Position
* Position Phone
* but Phone Position

**4.2.2 Database Design:**

**4.2.2.1 Conceptual DB design:**

**ER Modeling (Top down Approach)**

What is ER Modeling?

A graphical technique for understanding and organizing the data independent of the actual database implementation

We need to be familiar with the following terms to go further.

Entity

Anything that has an independent existence and about which we collect data. It is also known as entity type.

**Normalization (Bottom Up approach)**

**Purpose:**

Avoid anomalies

Reduce redundancy

**Process:**

Successive application of rules

Bottom-up (data drives process)

Move from first through fifth normal form

**4.2.2.2 Logical DB Design:**

Using Microsoft® SQL Server™ 2000 effectively begins with normalized database design. Normalization is the process of removing redundancies from the data. For example, when you convert from an indexed sequence access method (ISAM) style application, normalization often involves breaking data in a single file into two or more logical tables in a relational database. Transact-SQL queries then recombine the table data by using relational join operations. By avoiding the need to update the same data in multiple places, normalization improves the efficiency of an application and reduces the opportunities for introducing errors due to inconsistent data.

**4.2.2.3 Physical DB design:**

What was the point in creating the logical data model? You want to create a database to store data about CDs. The data model is only an intermediate step along the way. Ultimately, you would like to end up with a MySQL or mSQL database where you can store data. How do you get there? Physical database design translates your logical data model into a set of SQL statements that define your MySQL or mSQL database.

**4.3 Characters of relation:**

**4.4 ER to relational mapping algorithm:**

**4.4.1 Mapping of regular entity Type:**

**4.4.2 Mapping of week entity Type:**

**4.4.3 Mapping of binary 1:1 relational type:**

**4.4.4 Mapping of binary 1: n relational type:**

**4.4.5 Mapping of binary m: n relational type:**

**4.4.6 Mapping of multivalve attributes:**

**4.6.7 Mapping of N-array relationship type:**

**REFERENCE:**

[1]. http://www.fidelcaptain.com/casestudy1/erdcs1.html

[2]. https://technet.microsoft.com/en-us/library/aa178427(v=sql.80).aspx

[3].http://www.careerbless.com/db/rdbms/c1/design.php