# Lesture 7

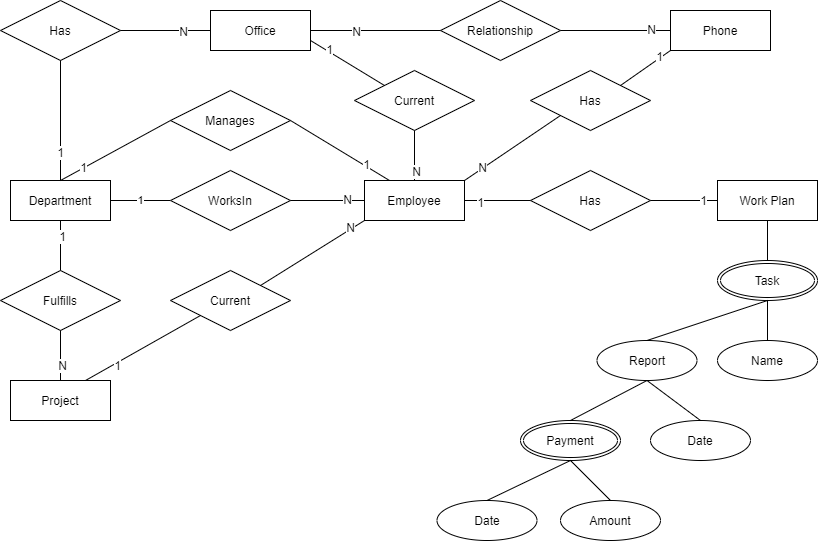
Create database schema according to these criteria. (Crate conceptual model, logical model, physical model, FD graphs for each table, CREATE TABLE script for each table.) You should describe normalform for each table. Tables must be in at least BCNF or you must justify why it is not so.

## Excercise 1

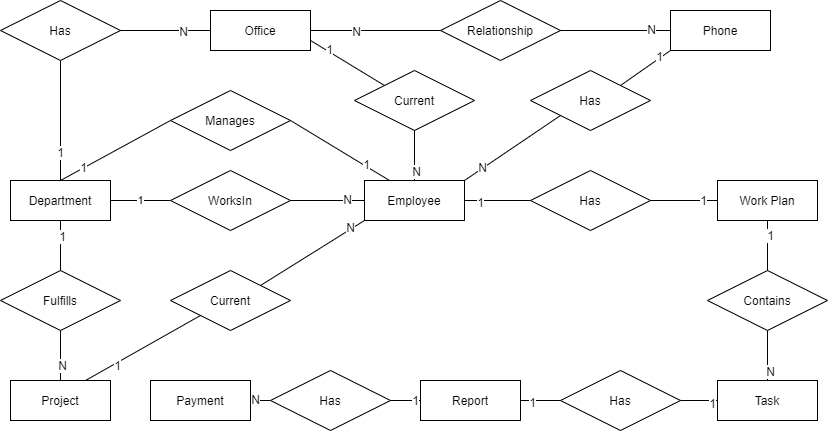
The firm has multiple departments

1. Each department contains several employees, multiple projects, multiple offices
2. Each employee has a work plan (i.e. a number of tasks to be fulfilled). Each task has an accounting report in which amount of payments have been listed (list may consist of several amount values) payed to an employee for the performance of that task.
3. There are several telephones in each office
4. The database must store
   1. For each department: ID (unique number), budget and manager ID
   2. For each employee, ID (unique number), current project ID, office ID, telephone number, and job list with all payments and dates they have been made for the job
   3. Each project: unique ID and budget
   4. Each office: unique ID, size in square meters, all unique phone numbers in this office
5. Assumptions:
   1. Not one employee leads multiple departments
   2. No employees work in multiple departments
   3. No employee uses more than one office
   4. No employee has more than one phone
   5. No project is done by more than one department
   6. No office is allocated for more than one department

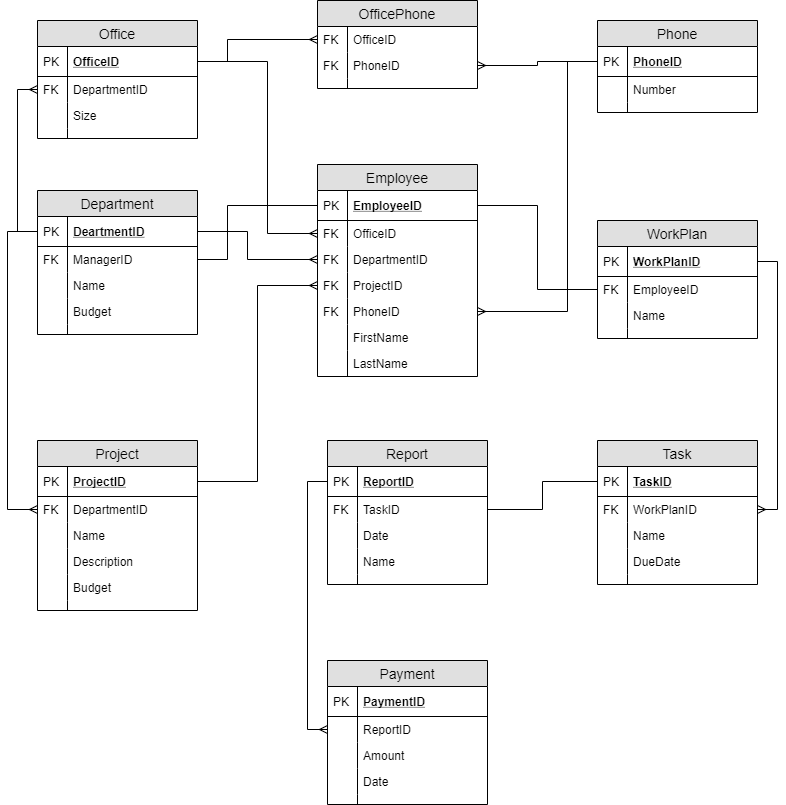
### Conceptual Model, Level 0



### Conceptual Model Level 1

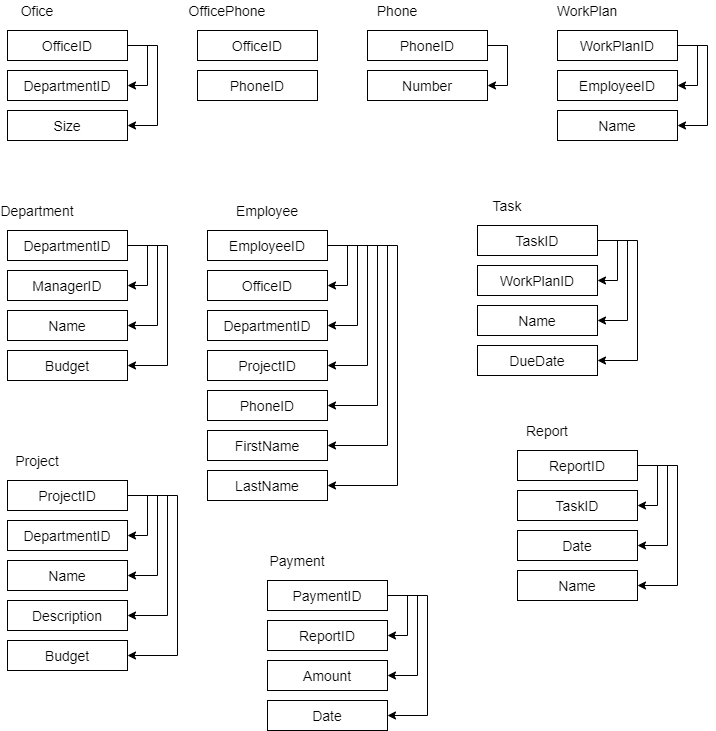


### Logical Model



All fields of relation OffizePhone are foreign keys and also form primary key.

### FD Graphs



### Normalforms

All tables are in Normalform 1 because all fields are atomic.

All tables has primary keys (Primary key for table OfficePhone is combination of both fields of that table); therefore all tables are in normalform 2.

All functional dependencies begins from full primary or potential key; therefore all tables are in BCNF.

### Table Create Scripts

CREATE DATABASE Office

GO

USE Office

GO

CREATE TABLE Phone(

PhoneID int IDENTITY(1,1) PRIMARY KEY,

Number char(15))

CREATE TABLE Department(

DepartmentID int IDENTITY(1,1) PRIMARY KEY,

ManagerID int,

Name nvarchar(50) NOT NULL,

Budget money)

CREATE TABLE Office(

OfficeID int IDENTITY(1,1) PRIMARY KEY,

DepartmentID int FOREIGN KEY REFERENCES Department(DepartmentID),

Size numeric(8,2))

CREATE TABLE OfficePhone(

OfficeID int NOT NULL FOREIGN KEY REFERENCES Office(OfficeID),

PhoneID int NOT NULL FOREIGN KEY REFERENCES Phone(PhoneID))

CREATE TABLE Project(

ProjectID int IDENTITY(1,1) PRIMARY KEY,

DepartmentID int NOT NULL FOREIGN KEY REFERENCES Department(DepartmentID),

Name nvarchar(100) NOT NULL,

Description nvarchar(4000),

Budget Money)

CREATE TABLE Employee(

EmployeeID int IDENTITY(1,1) PRIMARY KEY,

OfficeID int FOREIGN KEY REFERENCES Office(OfficeID),

DepartmentID int FOREIGN KEY REFERENCES Department(DepartmentID),

ProjectID int FOREIGN KEY REFERENCES Project(ProjectID),

PhoneID int FOREIGN KEY REFERENCES Phone(PhoneID),

FirstName nvarchar(20) NOT NULL,

LastName nvarchar(20) NOT NULL)

ALTER TABLE Department

ADD FOREIGN KEY (ManagerID) REFERENCES Employee(EmployeeID)

CREATE TABLE WorkPlan(

WorkPlanID int IDENTITY(1,1) PRIMARY KEY,

EmployeeID int FOREIGN KEY REFERENCES Employee(EmployeeID),

Name nvarchar(50))

CREATE TABLE Task(

TaskID int IDENTITY(1,1) PRIMARY KEY,

WorkPlanID int FOREIGN KEY REFERENCES WorkPlan(WorkPlanID),

Name nvarchar(50),

DueDate date)

CREATE TABLE Report(

ReportID int IDENTITY(1,1) PRIMARY KEY,

TaskID int FOREIGN KEY REFERENCES Task(TaskID),

Date date,

Name nvarchar(50))

CREATE TABLE Payment(

PaymentID int IDENTITY(1,1) PRIMARY KEY,

ReportID int FOREIGN KEY REFERENCES Report(ReportID),

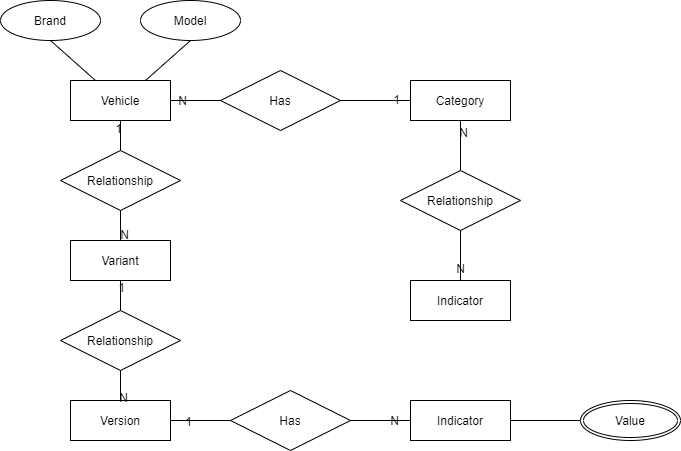
Amount money,

Date date)

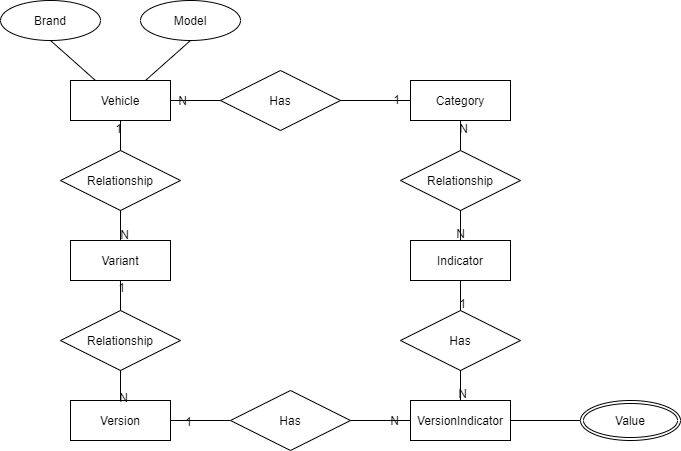
## Excercise 2

When manufacturing vehicles, the factory has previously issued ‘Type Approvals’ (TA). The TA document is meant to describe the technical characteristics of the variant and version of the vehicle (the model has multiple variations, the variant has multiple versions). The vehicle has a brand and model. TA contains several hundred technical indicators, such as fuel, engine volume, unladen mass, full mass, length, width, etc. Each indicator can have either one or more values. Each technical indicator may be for either one or more variants or versions. Values can either be entered manually or be in a system-coded classifier. Vehicles break down by category (motorcycles, cars, trucks, etc.). There is a slightly different set of technical indicators for each category - some of the indicators are for several categories, some are for one specific, some are for all categories. Create an ER model where Type Approvals can be stored with all of the above information.

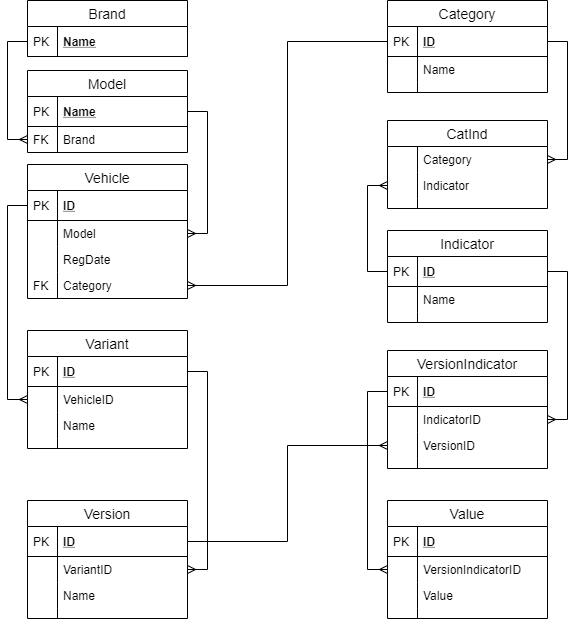
### Conceptual Model Level 0



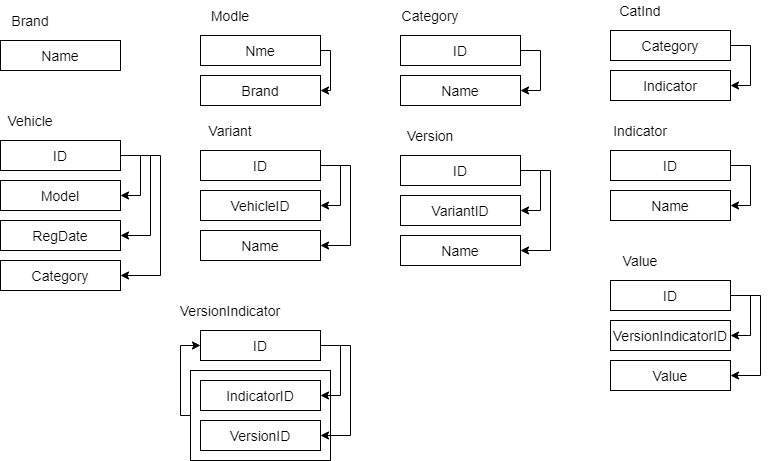
### Conceptual Model Level 1



### Logical Model



### FD Graphs



### Normalforms

All tables are in Normalform 1 because all fields are atomic.

All tables has primary keys; therefore all tables are in normalform 2.

All functional dependencies begins from full primary or potential key; therefore all tables are in BCNF.

### Table Create Scripts

### CREATE DATABASE AutoTA

### GO

### USE AutoTA

### GO

### CREATE TABLE Brand(

### Name char(30) PRIMARY KEY)

### CREATE TABLE Model(

### Name char(30) PRIMARY KEY,

### Brand char(30) NOT NULL FOREIGN KEY REFERENCES Brand(Name))

### CREATE TABLE Category(

### ID int IDENTITY(1,1) PRIMARY KEY,

### Name nvarchar(50))

### CREATE TABLE Vehicle(

### ID int IDENTITY(1,1) PRIMARY KEY,

### Model char(30) NOT NULL FOREIGN KEY REFERENCES Model(Name),

### RegDate Date NOT NULL,

### Category int NOT NULL FOREIGN KEY REFERENCES Category(ID))

### CREATE TABLE Variant(

### ID int IDENTITY(1,1) PRIMARY KEY,

### VehicleID int FOREIGN KEY REFERENCES Vehicle(ID),

### Name nvarchar(50))

### CREATE TABLE Version(

### ID int IDENTITY(1,1) PRIMARY KEY,

### VariantID int FOREIGN KEY REFERENCES Variant(ID),

### Name nvarchar(50))

### CREATE TABLE Indicator(

### ID int IDENTITY(1,1) PRIMARY KEY,

### Name nvarchar(50))

### CREATE TABLE CatInd(

### Category int FOREIGN KEY REFERENCES Category(ID),

### Indicator int FOREIGN KEY REFERENCES Indicator(ID))

### CREATE TABLE VersionIndicator(

### ID int IDENTITY(1,1) PRIMARY KEY,

### IndicatorID int FOREIGN KEY REFERENCES Indicator(ID),

### VersionID int FOREIGN KEY REFERENCES Version(ID))

### CREATE TABLE Value(

### ID int IDENTITY(1,1) PRIMARY KEY,

### VersionIndicatorID int FOREIGN KEY REFERENCES VersionIndicator(ID),

Value nvarchar(500))