

Project Concept:

Personalized family history website designs to purchase. The e-commerce site will contain: Layout design options and themes to choose from.

Customer ads could run on social media and the web to an audience that has interests in genealogy, ancestry, dna, biographies, history, and 18 to 65.

The unique feature is once created, it will allow customers to share with family history and others without “information seekers” having to go through other less detailed larger database focused sites that require memberships and sifting through a lot of information not related to their specific research.

Database Design Draft with entities and attributes:

The Department entity is the department that became each task for Employees to do. They are listed under Department Data. It has attributes listed below to include those department names. There are also entities of Customers, Employees, and Payments with the listed attributes and data types below. I have included the ER diagram, the ER diagram coding and the sql script code for the database I created.

Entities and Attributes: **Department:** EmployeeID number(3) department varchar30 [primary key] departmentID number(3) cost varchar6 Department Data: (style, mobile view enhanced, desktop view enhanced, Google mapping , Biographies, Pedigree search, main galleries, color scheme, font, Video on homepage, Extra Links) **Customers** custID number (3) [primary key, name varchar40, gender char(1), year_of_birth number(4), address varchar40, postal_code varchar10, city varchar30, phone varchar10, comment varchar200 **Employees** EmployeeID number(3) [primary key, Employee_name varchar30, commission varchar6, departmentID number(3), department varchar30 **Payments** custID number (3), postal_code varchar10, department varchar30, amount varchar6.

One-to-one relationships

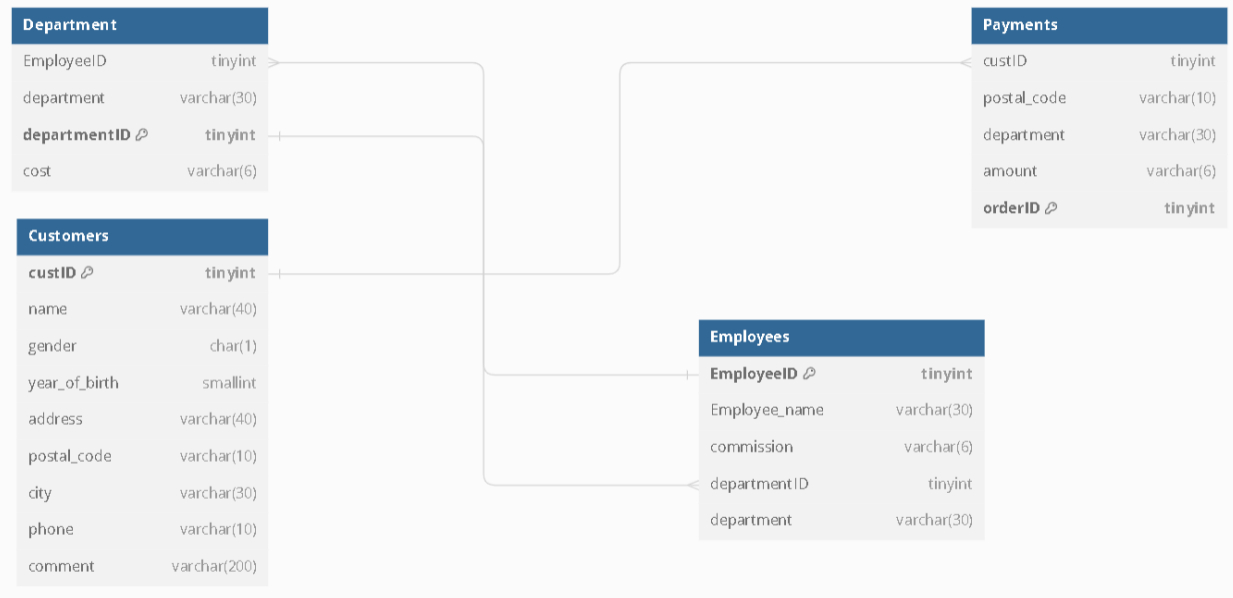
Employees have one employee ID, Customers have one customer ID

One-to-many

One employee is assigned to each department. Many employees could work in a department.

Many-to-many relationships

Department has many to many relationship between department, employees and payments.



```
Table "Department" {
  "EmployeeID" tinyint
  "department" varchar(30)
  "departmentID" tinyint [pk]
  "cost" varchar(6)
}
```

```
Table "Customers" {
  "custID" tinyint [pk]
  "name" varchar(40)
  "gender" char(1)
  "year_of_birth" smallint
  "address" varchar(40)
  "postal_code" varchar(10)
  "city" varchar(30)
  "phone" varchar(10)
  "comment" varchar(200)
}
```

```
Table "Employees" {
  "EmployeeID" tinyint [pk]
  "Employee_name" varchar(30)
  "commission" varchar(6)
}
```

```

    "departmentID" tinyint
    "department" varchar(30)
}

Table "Payments" {
    "custID" tinyint
    "postal_code" varchar(10)
    "department" varchar(30)
    "amount" varchar(6)
    "orderID" tinyint [pk]
}

Ref:"Employees"."EmployeeID" < "Department"."EmployeeID"

Ref:"Department"."departmentID" < "Employees"."departmentID"

Ref:"Customers"."custID" < "Payments"."custID"

```

```

CREATE TABLE `Department` (
    `EmployeeID` tinyint,
    `department` varchar(30),
    `departmentID` tinyint PRIMARY KEY,
    `cost` varchar(6),
    FOREIGN KEY (EmployeeID) References Employees(EmployeeID)
);

```

```

CREATE TABLE `Customers` (
    `custID` tinyint PRIMARY KEY,
    `name` varchar(40),
    `gender` char(1),
    `year_of_birth` smallint,
    `address` varchar(40),
    `postal_code` varchar(10),

```

```
`city` varchar(30),  
`phone` varchar(10),  
`comment` varchar(200)  
);
```

```
CREATE TABLE `Employees` (  
  `EmployeeID` tinyint PRIMARY KEY,  
  `Employee_name` varchar(30),  
  `commission` varchar(6),  
  `departmentID` tinyint,  
  `department` varchar(30),  
  FOREIGN KEY (departmentID) References Department(departmentID)  
);
```

```
CREATE TABLE `Payments` (  
  `custID` tinyint,  
  `postal_code` varchar(10),  
  `department` varchar(30),  
  `amount` varchar(6),  
  `orderID` tinyint PRIMARY KEY,  
  FOREIGN KEY (custID) References Customers(custID)  
);
```