Assignment 3

For this assignment I had to design an ER diagram for each case, convert the ER diagram to the relational model, and convert the relational notation to SQL (correct syntax was not required).

DB1 corresponds to pages 2 and 3, DB2 corresponds to pages 4 and 5.

DB1. Car Inventory. (20 marks)

This database will hold inventory for a car dealership. There are three important entities: Manufacturers, Cars, and Engines. Name, Country of origin, Year Founded, and a unique identifier should be stored for Manufacturers. Manufacturers have a one-to-many relationship with Cars, each of which consist of an Stock Keeping Unit # (SKU), Name, and up to 5 dealerships where the car is sold. A car can have only one engine that also has an SKU#. Information about the Fuel Type and Engine Manufacturer Name, and Engine Type (I4, V6, V8) should be stored.

DB2. University Professors. (20 marks)

This database will be used to keep track of Professors at a University. There are six important entities: Professors, Degrees, Departments, Courses, Research, and Graduate Students.

Professors should have a unique Employee ID, a name, age, and their hire date saved in the database. One Professor may hold many degrees. The primary key of a Degree consists of the Confer Date, Title (ie. PhD, MBA, etc), Major, and Institution. There should also be a marker to show if the degree is a terminal degree or not.

One Professor may supervise many graduate students. A graduate student has a name, program, and student number. A Professor may also supervise many research projects. Research projects should have an id number, how much funding was allocated, how much funding is left, the name of the funding agency, and the title of the project.

One Professor may teach many courses. The Enrollment Count, Title, and Description of the course should be saved; each course can be identified with its unique course code. A close up of a map

Description automatically generatedManufacturer (1-side)

|  |  |  |  |
| --- | --- | --- | --- |
| ManufacturerID | Name | CountryOfOrigin | YearFounded |

Car&Engine (n-side)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CarSKU | CarName | Dealership1 | Dealership2 | Dealership3 | Dealership4 |
| Dealership5 | FuelType | EngineManufacturer | EngineType | EngineSKU | ManufacturerID |

SQL  
  
Create table manufacturer(manufacturerid integer primary key, name text, countryoforigin text, yearfounded text);  
  
Create table car&engine(carsku text primary key, carname text, dealership1 text, dealership2 text, dealership3 text, dealership4 text, fueltype text, enginemanufacturer text, enginetype text, enginesku text, manufacturerid FK referencing manufacturer.manufacturerid);

A close up of a map

Description automatically generated

Notes: I added an ID to the degrees column to get it into second normal form. Confer Date may not work because people can receive the same degrees from the same institution with the same major, title and all, to then get them at the same day. Thus, in case, I added an ID. Also, the Courses relation is a merged relation between the former Courses and Departments relations. They had aa 1:1 relationship. For all relations except the Professor relation I added the professors’ IDs as foreign keys.  
  
Professor (1-side)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| EmployeeID | HireDay | HireMonth | HireYear | Fname | Lname | Age |

ResearchProjects (n-side)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID# | ProjectTitle | FundingAlloc | FundingLeft | FundingAgency | SupervisorID |

GraduateStudents (n-side)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student# | Fname | Lname | Program | SupervisorID |

Degrees (n-side)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | ConferDay | ConferMonth | ConferYear | Institution | Major | Title |
| TerminalDegree? | SupervisorID |  |  |  |  |  |

Courses (n-side)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Code | EnrollmentCount | Title | Description | Department | ProfessorID |

SQL  
Create table professor(employeeid integer primary key, hireday integer, hiremonth integer, hireyear integer, fname text, lname text, age integer);

Create table researchprojects(id# integer primary key, projecttitle text, fundingalloc integer, fundingleft integer, fundingagency text, supervisorid FK referencing professor.employeeid);

Create table graduatestudents(student# integer primary key, fname text, lname text, program text, supervisorid FK referencing professor.employeeid);

Create table degrees(id integer primary key, conferday integer, confermonth integer, conferyear integer, institution text, major text, title text, terminaldegree? text, supervisorid FK referencing professor.employeeid);

Create table courses(code integer primary key, enrollmentcount integer, title text, description text, department text, professorid FK referencing professor.employeeid);