

Part 1 done by Monday September 21st

- ☐ Use ES6 class syntax to create classes with constructors (and extends keyword when appropriate) for
 - ☐ Person
 - ☐ Employee (inherit from Person)
 - ☐ Manager (inherit from Employee) (remember that manager keeps track of directReports)
 - ☐ Executive (inherit from Manager)
 - ☐ Nonemployee (inherit from Person)
 - ☐ Contractor (inherit from Nonemployee)
 - ☐ Vendor (inherit from Nonemployee)
 - ☐ Customer (inherit from Nonemployee)
- ☐ Use faker.js to create fake data for
 - ☐ 5 vendors
 - ☐ 5 customers
 - ☐ 5 contractors
 - ☐ 20 employees
 - ☐ 5 managers
 - ☐ 3 executives
- ☐ Call the constructors with fake data for
 - ☐ 5 vendor objects of type Vendor
 - ☐ 5 customer objects of type Customer
 - ☐ 5 contractor objects of type Contractor
 - ☐ 20 employee objects of type Employee
 - ☐ 5 manager objects of type Manager
 - ☐ 3 executive objects of type Executive
- ☐ Modify the objects so that
 - ☐ Each employee reports to a random manager.
 - ☐ First two managers report to the first executive.
 - ☐ Last three managers report to the second executive.
 - ☐ First and second executives report to the third executive.
 - ☐ Third executive reports to itself. (Yes, I've seen this done.)
- ☐ Put your objects in arrays called
 - ☐ vendors
 - ☐ customers
 - ☐ contractors
 - ☐ employees
 - ☐ managers
 - ☐ executives

Part 2 done by Friday September 25th

- ☐ Create DBs in SQLite3 (sql script, don't use JS)
 - ☐ create DB ORM1
 - ☐ create tables
 - ☐ create DB ORM2
 - ☐ create tables
 - ☐ create DB ORM3
 - ☐ create tables
 - ☐ create DB ORM4
 - ☐ create tables
- ☐ Create DBs in Postgres (sql script, don't use JS)
 - ☐ create DB ORM1
 - ☐ create tables
 - ☐ create DB ORM2
 - ☐ create tables
 - ☐ create DB ORM3
 - ☐ create tables
 - ☐ create DB ORM4
 - ☐ create tables
- ☐ Create DBs in MSSQL (optional) (sql script, don't use JS)
 - ☐ create DB ORM1
 - ☐ create tables
 - ☐ create DB ORM2
 - ☐ create tables
 - ☐ create DB ORM3
 - ☐ create tables
 - ☐ create DB ORM4
 - ☐ create tables
- ☐ Create DBs in MySQL (optional) (sql script, don't use JS)
 - ☐ create DB ORM1
 - ☐ create tables
 - ☐ create DB ORM2
 - ☐ create tables
 - ☐ create DB ORM3
 - ☐ create tables
 - ☐ create DB ORM4
 - ☐ create tables

Part 3 done by Wednesday September 30th

- ☐ Use the strategy in subsheet ORM1 to create DAO modules. All DAOs must use prepared statements.
 - ☐ **vendorDAO1Sqlite.js** (Note: there is no PersonDAO or NonEmployeeDAO)
 - ☐ C
 - ☐ R
 - ☐ U
 - ☐ D
 - ☐ L
 - ☐ **customerDAO1Sqlite.js**
 - ☐ C
 - ☐ R
 - ☐ U
 - ☐ D
 - ☐ L
 - ☐ **contractorDAO1Sqlite.js**
 - ☐ C
 - ☐ R
 - ☐ U
 - ☐ D
 - ☐ L
 - ☐ **employeeDAO1Sqlite.js**
 - ☐ C
 - ☐ R
 - ☐ U
 - ☐ D
 - ☐ L
 - ☐ **managerDAO1Sqlite.js**
 - ☐ C
 - ☐ R
 - ☐ U
 - ☐ D
 - ☐ L
 - ☐ **executiveDAO1Sqlite.js**
 - ☐ C
 - ☐ R
 - ☐ U
 - ☐ D
 - ☐ L
- ☐ Write a script to iterate over vendors to create() in vendorDAO1Sqlite.js to populate the right table(s).
- ☐ Write a script to iterate over customers. Use create() in vendorDAO1Sqlite.js to populate the right table(s).

- ❑ Write a script to iterate over contractors. Use create() in vendorDAO1Sqlite.js to populate the right table(s).
- ❑ Write a script to iterate over employees. Use create() in vendorDAO1Sqlite.js to populate the right table(s).
- ❑ Write a script to iterate over managers. Use create() in vendorDAO1Sqlite.js to populate the right table(s).
- ❑ Write a script to iterate over executives. Use create() in vendorDAO1Sqlite.js to populate the right table(s).
- ❑ The DAO on the previous page is just one tuple from this cartesian product

$\{\text{ORM1, ORM2, ORM3, ORM4}\} \times \{\text{SQLite, Postgres, *MSSQL, *MySQL}\} \times \{\text{raw DAO, knex DAO}\}$

Repeat the previous page for the rest of the DAOs.

- ❑ Do the tuples for

$\{\text{SQLite, Postgres, *MSSQL, *MySQL}\} \times \{\text{sequelize}\}$ (Sequelize makes its own ORM)

MSSQL and MySQL are optional. If you do them, you must complete them. No partial credit.

If you didn't pay attention during set theory in discrete math, then it sucks to be you because the terms "tuple" and "cartesian product" are used all the time in database work.