1. Collection and Documenting relationships

   - Collection “job”: Contain all job information
   _id: String (surrogate key)
   workclass: String
   occupation: String

   - Collection “education”: Contain all education’s levels
   _id: String (surrogate key)
   education: String
   education_num: Int32

   - Collection “relation”: Contain all individual relation
   _id: String (surrogate key)
   marital_status: String
   relationship: String

   - Collection “national”: Contain all ethnicities
   _id: String (surrogate key)
   race: String
   native_country: String

   - Collection “adult”: Contain every single person’s data
   _id: String (surrogate key)
   age: Int32
   total: Int32 (total of assets)
   gender: String
   capital_gain: Int32 (top-up in balance)
   capital_loss: Int32 (charge in balance
   hours_per_week: Int32 (work hours per week)
   income_bracket: String
   job: String (Foreign Key)
   education: String (Foreign Key)
   relation: String (Foreign Key)
   national: String (Foreign Key)

2. Indexing the Collection

   - Based on data sources and business queries, the data seem to be focused on individual finance status: assets, capital gain, capital loss, and income.
   - Indexing the “adult.total” will provide high performance for finance queries, to check if it works, use the command:
   db.adult.explain().find()