Caleb Rogers

Professor Alan Labouseur

November 17th, 2020

Database Systems

Lab 9: Normalization 3

1)

Engineers: pID --> highestDegree, favVideoGame

FlightControlOperators: pID --> favChair, favDrink, hangoverCure

Astronauts: pID --> yrsFlying, golfHandicap, spouseID

People: pID --> firstName, lastName, yrsOld

Crew: pID, scID

Spacecraft: scID --> shuttleName, tailNum, weightTons, fuelType, crewCapacity

Systems: scID --> systemName, description, costUSD

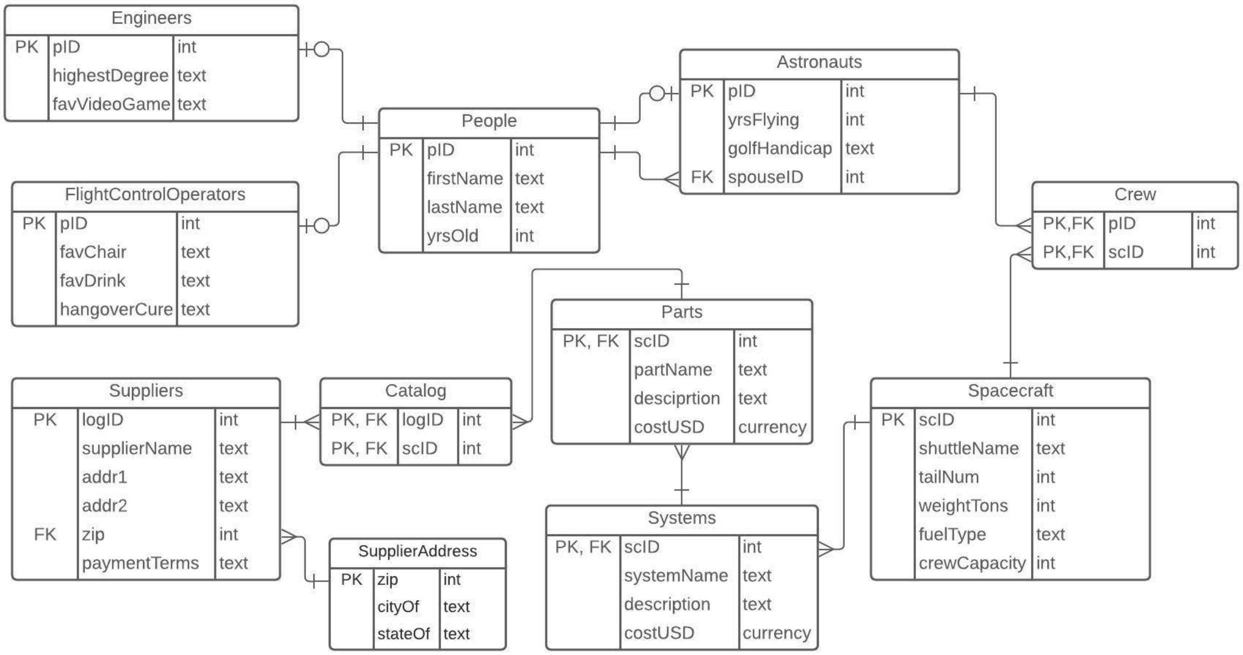
Parts: scID --> partName, description, costUSD

Catalog: logID, scID

Suppliers: logID --> supplierName, addr1, addr2, zip, paymentTerms

SuppliersAddress: zip --> cityOf, stateOf

2)



3) This database fulfills the three normal rules required to normalize a relational database. The first normal rule that requires all intersecting values in any table are atomic, and this is accomplished through the split up of a supplier's address and the split up of an astronaut's spouse's name. The second normal rule calls that the first normal rule is fulfilled and that there are no partial key dependencies. Adding a ‘People’ table solves this issue by partitioning the similar data between engineers, flight control operators, and astronauts in the ‘People’ table; and then ‘Engineers’ table has its specific engineer's data, the ‘Astronauts’ table has its specific astronaut's data, and the ‘FlightControlOperators’ has its specific flight control operator's data. Splitting the primary key between these table eliminated the need of this varying data to rely on a single primary key causing partial key dependencies. The third normal rule is accomplished by fulfilling the second normal rule and eliminating any transitive dependencies. There does not exist any transitive dependencies therefore this database is normalized. This database also fulfills

Boyce-Codd Normal Form (BCNF) as it is protected from insert, update, and delete anomalies through appropriate normalization and use of referential integrity.