Thomas Magnusson

Alan Labouseur

Database Management

19 November 2017

NASA Functional Dependencies

[People] (pid) -> firstName, lastName, dateOfBirth

[Engineers] (pid) -> highestDegree, favoriteVideoGame

[Astronauts] (pid) -> firstFlightDate, golfHandicap

[FlightControlOperators] (pid) -> chairPref, drinkPref, hangoverCure

[Suppliers] (pid) -> address, address2, city, state, zip, country

[Marriages] (pid1, pid2) -> [The relation itself indicates one person is married to another]

[Crew] (pid, scid, missionStartDate) -> [a given astronaut is part of a crew on a specific spacecraft on a mission that starts on a particular day]

[Spacecrafts] (scid) -> tailNumber, fuelType, crewCapacity, weightTons

[SpacecraftComponents] (cid) -> description, costUSD

[Parts] (cid) -> [parts only have a description and a cost]

[Systems] (cid) -> [systems have a description and cost]

[Catalog] -> (supplierPid, partCid) -> paymentTerms

[SpacecraftConstructs] (partCid, systemCid) -> [systems have many parts, many of the same parts can belong to different systems]

[SpacecraftInnards] (scid, systemCid) -> [A spacecraft can have many parts, one part can belong to many spacecrafts]

My NASA database design is in first normal form because all of the fields are atomic, that is to say that they cannot be logically decomposed further.

My database is in second normal form because it is in first normal form and no non-key attributes rely on a portion of the primary key, that is to say that there exist no partial key dependencies.

My database is in third normal form because it is in second normal form and it contains no multikey dependencies. Every non-key attribute is determined entirely by the primary key.

My database is in Boyce-Codd normal form because it is in third normal form and there exist no transitive dependencies among candidate keys. That is to say, the only candidate keys are the primary keys.