EERD to Relational Schema

Bryson Bien - Elliot Willner - Max Sikora

USERS = (uname, address, birthdate, fname, lname)

CUSTOMER = (uname [fk1], rating, credit)

fk1: uname -> USERS.uname

EMPLOYEE = (uname [fk2], taxID, hired, salary, service)

fk2: uname -> USERS.uname

ITEM = (barcode, iname, weight)

DRONE_PILOT = (uname, taxID [fk8], licenseID, experience)

fk8: uname, taxID -> EMPLOYEE.uname, EMPLOYEE.taxID

FLOOR_WORKER = (uname, taxID [fk9])

fk9: uname, taxID -> EMPLOYEEe.uname, EMPLOYEE.taxID

STORE = (storeID, sname, revenue, manager [fk7])

fk7: manager -> FLOOR_WORKER.uname, manager is non-null

DRONE = (storeID [fk3], droneTag, rem_trips, capacity, serve [fk6], control [fk14])

fk3: storeID -> STORE.storeID

fk6: serve -> STORE.storeID, serve is non-null

fk14: control -> DRONE_PILOT.licenseID, control is non-null

ORDERS = (orderID, sold_on, request [fk4], deliver [fk5])

fk4: request -> CUSTOMER.uname, request is non-null

fk5: deliver -> DRONE.droneTag, deliver is non-null

CONTAIN = (barcode [fk10], orderID [fk11], price, quantity)

fk10: barcode -> ITEM.barcode

fk11: orderID -> ORDER.orderID

EMPLOY = (storeID [fk12], uname, taxID [fk13])

fk12: storeID -> STORE.storeID

fk13: uname, taxID -> FLOOR_WORKER.uname, FLOOR_WORKER.taxID

Unhandled Constraints, Assumptions, & Comments

- **RENAMES:** USER is already a SQL keyword, so we switched it to USERS to allow us to create a table, and the same applied from ORDER to ORDERS.
- **RELATIONSHIPS:** We made the relationship manage become manager to match the data from canvas more. M:N relationships became tables in SQL while the others because attributes.
- **DERIVED ATTRIBUTE:** Any derived attribute will be accessed with a query so we did not place it into our database initialization.
- **DISJOINTED CONSTRAINTS:** We do not have an implemented disjoint constraint to assume floor workers and drone pilots are not the same user.
- **FOREIGN KEYS:** When accessing foreign keys in SQL, we assume the furthest attribute is derived from the nearest entity. We also presume that the UNIQUE characteristic is defined upon initialization so we don't re-establish it while we do re-establish if it is a PRIMARY KEY again.
- **ORDER:** The order of the relational schema is consistent with SQL.
- **SQL DATA**: The data inserted into the database is according to the relational schema and it is assumed that the data can be pulled out to form the canvas data. Constraints for all foreign keys so the data can be accessed without error.
- **UNAME:** Uname is used over Fname when referenced as it is unique.
- **ITEMS:** Multiple items can pass through the contain relationship during orders.
- **DRONE TABLE:** Serve referencing storeID and storeID are both referenced, so to ensure no missing data, we included them both despite the supposed overlap.