

Relational Schema

user = (username, address, birthdate, first_name, last_name)

owner = (username[fk1])

fk1: username → user.username

employee = (username[fk2], taxID, experience, hired, salary)

fk2: username → user.username

pilot = (username[fk3], taxID[fk4], license_type, experience)

fk3: username → user.username, fk4: taxID → employee.taxID

worker = (username[fk5], taxID[fk6])

fk5: username → user.username, fk6: taxID → employee.taxID

location = (label, x_coord, y_coord, space)

restaurant = (name, spent, rating)

service = (ID, name, manager[fk7])

fk7: manger → worker.username

ingredient = (barcode, iname, weight)

drone = (ID[fk8], tag, fuel, capacity, sales, pilot_user[fk9], swarm_lead_ID[fk10],
swarm_lead_tag, hover[fk11])

fk8: ID → service.ID, fk9: control → pilot.username, fk10: lead_ID → drone.ID, fk11:
hover → location.label

contain = (barcode[fk12], ID, tag[fk13], price, quantity)

fk12: barcode → ingredient.barcode, fk13: ID, tag → service.ID, drone.tag

fund = (username[fk14], name[fk15], invested, dt_made)

fk14: username → user.username, fk16: name → restaurant.name

Work_for = (username[fk16], ID[fk17])

fk16: username → user.username, fk17: ID → service.ID

Unhandled Constraints

The previous job of employee is stored.

The derived attributes cannot be tracked.

Ensure that a user is employee or owner or both.

Ensure that an employee is either a pilot or a worker.

Ensure that service has at least one worker that works for the service.

Ensure that drone has at least one service that own the drone.

Ensure that drone has at least one location to hover.

Ensure the location is either a service or a restaurant.

