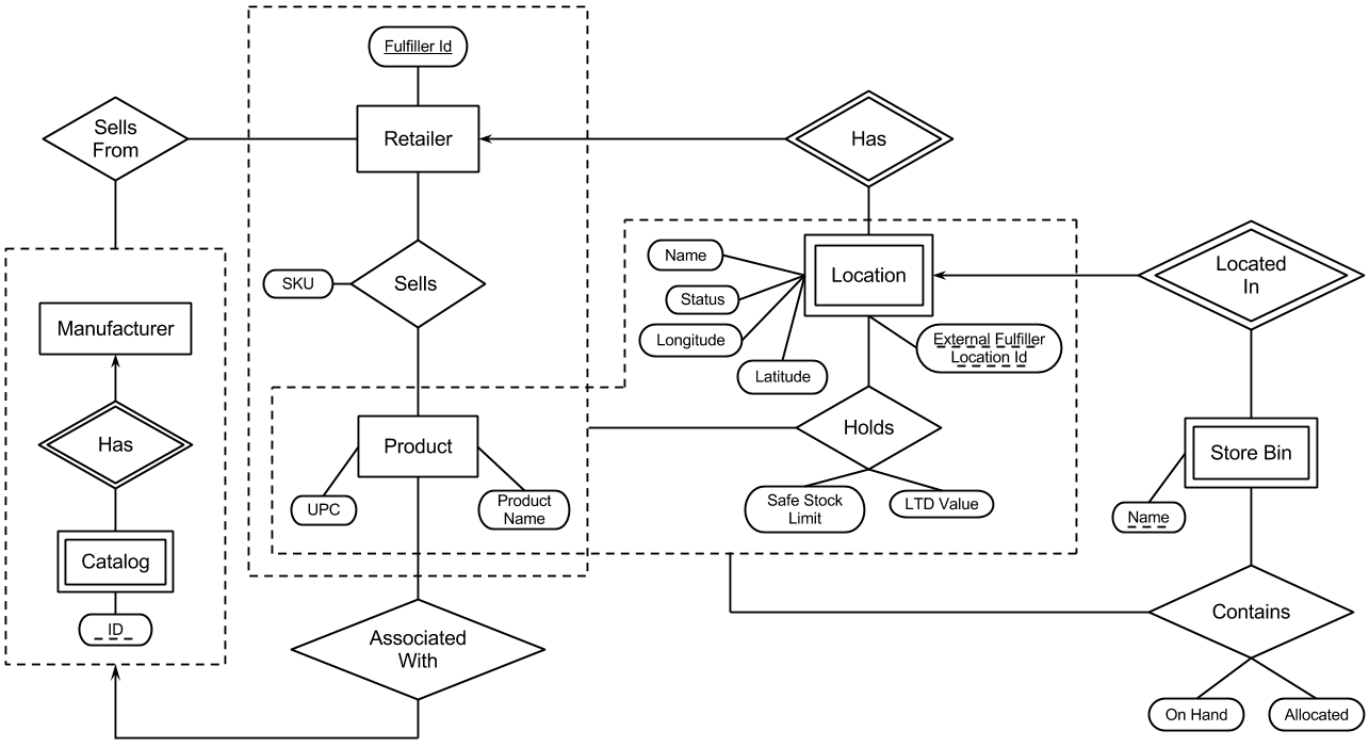
CSC 366
GLADE
Diego Torres
Adin Miller
Lauren Thurston
George Reyes
Elizabeth Loui

Lab 4

ER Diagram



Change Log

- Changed the Catalog entity set to a weak entity set with ID as its discriminator
- Changed the many-to-one Has relationship set into an identifying relationship set, which links Manufacturer and Catalog (many catalogs for one manufacturer)
- Added the Fulfiller Id as the primary key of the Retailer entity set
- Added the many-to-many Sells From relationship set, which links Retailer and the

aggregation containing Manufacturer, Catalog, and Has
- Changed the Inventory Record entity set to the Product entity and added UPC and Product Name attributes to it
- Added the many-to-many Sells relationship set, which links Retailer and Product, and added a SKU attribute to it
- Aggregated Retailer, Sells, and Product
- Changed the many-to-one Associated With relationship set to link Product and Catalog (many products to one catalog)
- Changed the Location entity set to be a weak entity set with External Fulfiller Location Id as its discriminator and it now also has the following attributes: Name, Status, Longitude, and Latitude
- Changed the many-to-one Has relationship set into an identifying relationship set, which links Location and the aggregation of Retailer, Sells, and Product (Many locations to one combination of retailers and products)
- Added Safe Stock Limit and LTD Value attributes to the many-to-many Holds relationship, which links Location and the aggregation of Retailer, Sells, and Product
- Aggregated Location, Holds, and Product
- Changed the Located In identifying relationship set into a many-to-one relationship set, which links Store Bin and Location
- Added the On Hand and Allocated attributes to the many-to-many Contains relationship set, which has been updated to link Store Bin and the aggregation of Location, Holds, and Product

Relational Tables

Retailer
- FulfillerId PRIMARY KEY
- Status

Location
- FulfillerId
- ExternalFulfillerLocationId
- InternalFulfillerLocationId PRIMARY KEY
- Type
- Name
- Description
- Latitude
- Longitude
- Status

- SafetyStockLimit
- UNIQUE(FulfillerId, ExternalFulfillerLocationId)
- Attribute FulfillerId is a foreign key referencing table Retailer

StoreBin
- Id PRIMARY KEY
- InternalFulfillerLocationId
- Status
- Type
- Name
- Description
- UNIQUE (InternalFulfillerLocationId, Name)
- Attribute InternalFulfillerLocationId is a foreign key referencing table Location

ContainedInBin
- BinId
- LocationProductId
- OnHand
- Allocated
- PRIMARY KEY(BinId, RetailerProductId)
- Attribute BinId is a foreign key referencing table StoreBin
- Attribute LocationProductId is a foreign key referencing table LocationProduct

Product
- Name
- UPC PRIMARY KEY
- ManufacturerId
- CatalogId
- Attribute ManufacturerId is a foreign key referencing table Manufacturer
- Attribute CatalogId is a foreign key referencing table Catalog

RetailerProduct
- Id PRIMARY KEY
- FulfillerId
- UPC
- SKU
- UNIQUE(FulfillerId, UPC)
- Attribute FulfillerId is a foreign key referencing table Retailer
- Attribute UPC is a foreign key referencing table Product

LocationProduct
- Id PRIMARY KEY
- InternalFulfillerLocationId

- RetailerProductId
- LTD
- SafeStockLimit
- Attribute InternalFulfillerLocationId is a foreign key referencing table Location
- Attribute RetailerProductId is a foreign key referencing table RetailerProduct

CatalogServedByLocation
- Id PRIMARY KEY
- InternalFulfillerLocationId
- ManufacturerId
- CatalogId
- UNIQUE(InteralFulfillerLocationId, ManufacturerId, CatalogId)
- Attribute InternalFulfillerLocationId is a foreign key referencing table Location
- Attribute ManufacturerId is a foregin key referencing table Manufacturer
- Attribute CatalogId is a foreign key referencing table Catalog

Manufactuer
- ManufacturerId PRIMARY KEY

Catalog
- CatalogId
- ManufacturerId
- PRIMARY KEY(CatalogId, ManufacturerId)
- Attribute ManufacturerId is a foreign key referencing table Manufacturer

Constraints

OnHand >= LocationProduct.SafeStockLimit >= 0
Allocated >= 0
Location.SafetyStockLimit >= 0
LTD >= 0

Use Cases

Define Fulfiller (Retailer)
- API call: createFulfiller
- input: createFulfillerRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID)
  FulfillerRequest(String Name, +int ManufacturerID, +int RetailerID)
- output: createFulfillerResponse(int createFulfillerReturn)
- SQL:
  INSERT INTO Retailer VALUES (FulfillerID, Status);

Get Fulfiller Status

- API call: getFulfillerStatus
- input: getFulfillerStatusRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID)
  int FulfillerID
- output: getFulfillerStatusResponse(int Status)
- SQL:
  SELECT Status
  FROM Retailer r
  WHERE r.FulfillerId = FulfillerID;

Define Store Location

- API call: createFulfillmentLocation
- input: createFulfillmentLocationRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID)
  FulfillmentLocation(+int FulfillerID, +int ManufacturerLocationID,
  +int RetailerLocationID, String ExternalLocationID,
  String LocationName, +int TypeID, double Latitude,
  double Longitude, int Status, String CountryCode)
- output: int createFulfillmentLocationReturn
- SQL:
  INSERT INTO Location VALUES (FulfillerID, ExternalLocationID, RetailerLocationID,
  Location Name, TypeID, Latitude, Longitude, Status);

Get Fulfillment Locations

- API call: getFulfillmentLocations
- input: getFulfillmentLocationsRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID)
  FulfillmentLocation(+int FulfillerID, +int ManufacturerLocationID,
  +int RetailerLocationID, String ExternalLocationID,
  String LocationName, +int TypeID, double Latitude,
  double Longitude, int Status, String CountryCode)
- output: getFulfillmentLocationsResponse(+int FulfillerID, +int FulfillerLocationID)
- SQL:
  SELECT FulfillerInternalLocationID
  FROM Location L, Retailer R
  WHERE L.FulfillerID = R.FulfillerID;

Get Fulfillment Location Types

- API call: getFulfillmentLocationTypes
- input: createFulfillmmentLocationRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID)

FulfillmentLocation(+int FulfillerID, +int ManufacturerLocationID,
+int InternalFulfillerLocationID,
String ExternalFulfillerLocationID, String LocationName,
+int TypeID, double Latitude, double Longitude, int Status,
String CountryCode)

- output: FulfillmentLocationType(+int FulfillmentLocationTypeID, String Description)
- SQL:
  SELECT Type, Description
  FROM Location l
  WHERE l.InternalFulfillerLocationId = InternalFulfillerLocationID;

Define Store Bin

- API call: createBin
- input: createBinRequest
  - AuthenticationHeader(+int UserID, +int FulfillerID),
  - createBin(+int BinId, +int FulfillerLocationId, int BinStatusId, int BinTypeId, String Name)
- output: createBinReturn (positiveInteger)
  - createBinResponse(int positiveInteger)
- SQL:
  INSERT INTO StoreBin VALUES (BinId, FulfillerLocationId, BinStatusId, BinTypeId, Name, 'Description');

Get Bins

- API call: getBins
- input: getBinsRequest
  - AuthenticationHeader(+int UserID, +int FulfillerID),
  - BinRequest(+int FulfillerLocationID, String SearchTerm, +int NumResults, +int ResultsStart)
- output: BinResponse:
  Bin[] bins
  int ResultCount
  - Bin(+int BinID, +int FulfillerLocationID, +int BinTypeID, +int BinStatusID, String Name)
- SQL:
  SELECT *
  FROM StoreBin b, (SELECT COUNT(*) AS ResultCount
  FROM StoreBin b
  WHERE b.FulfillerLocationId = FulfillerLocationID) c
  WHERE b.FulfillerLocationId = FulfillerLocationID;

Get Bin Types

- API call: getBinTypes

- input: getBinTypesRequest
    - AuthenticationHeader(+int UserID, +int FulfillerID),
    - getBinTypes(complexType)
- output: BinType(+int BinTypeID, String Description)
- SQL:

```
SELECT Type, Description
FROM StoreBin b
WHERE b.FulfillerLocationId = FulfillerLocationID;
```

Get Bin Statuses

- API call: getBinStatuses
- input: getBinStatusesRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID),
  getBinStatuses(complexType)
- output: BinStatus(+int BinStatusID, String Description)
- SQL:

```
SELECT Status, Description
FROM Location l, StoreBin b
WHERE  l.FulfillerId = FulfillerID AND l.InternalFulfillerLocationId = b.InternalFulfillerLocationID;
```

Define Manufacturer/Catalog

- API call: createManufacturerCatalog
- input: createManufacturerCatalogRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID),
  ManufacturerCatalogRequest(+int ManufacturerID, +int CatalogID,
  +int FulfillerLocationID, int Status)
- output: createManufacturerCatalogResponse (complexType)
- SQL:

```
INSERT INTO Manufacturer VALUES (ManufacturerID);
INSERT INTO Catalog VALUES (CatalogID, ManufacturerID);
```

Fulfill Inventory

- API call: fulfillInventory
- input: FulfillInventoryRequest(
  Authentication Header,
  fulfillInventory(UpdateRequest request(
  FulfillerLocationCatalog FulfillmentLocationCatalog(
  ManufacturerCatalog ManufacturerCatalog(
  +int ManufacturerID, +int CatalogID),
  +int FulfillerLocationID,
  ManufacturerLocation ManufacturerLocation(
  +int ManufacturerID, +int ManufacturerLocationID),
  RetailerLocation RetailerLocation(

+int RetailerID, +int RetailerLocationID)),
UpdateItem[] Items)))
- UpdateItem(String PartNumber, String UPC, String LocationUPC, int Quantity, +int OrderID, +int OrderItemID, +int ShipmentID, +int FulfillerLocationID)))

- output: FulfillInventoryResponse()
- SQL:
  UPDATE ContainedInBin CB, CatalogServedByLocation CL, LocationProduct LP, RetailerProduct RP
  SET CB.Allocated = CB.Allocated - Quantity AND CB.OnHand = CB.OnHand - Quantity
  WHERE ManufacturerID = CL.ManufacturerId AND CatalogID = CL.CatalogId AND
  CL.InternalFulfillerLocationId = FulfillerLocationID AND
  CL.InternalFulfillerLocationId = LP.InternalFulfillerLocationId AND
  LP.RetailerProductId = RP.Id AND RP.UPC = UPC AND
  LP.Id = CB.LocationProductId;

Bulk Inventory Update

- API call: adjustInventory
- input: AdjustInventorySoapIn(AuthenticationHeader, AdjustRequest(String LocationName, AdjustItem[] Items))
  - AdjustItem(String PartNumber, String UPC, String LocationUPC, int BinID, int Quantity)
- output: AdjustInventorySoapOut(String AdjustResponse)
- SQL:
  UPDATE ContainedInBin C, LocationProduct LP, RetailerProduct RP
  SET RP.UPC = UPC AND C.BinId = BinId AND C.OnHand = Quantity
  WHERE C.LocationProductId = LP.Id AND LP.RetailerProductId = RP.Id;

Trickle Inventory Update

- API call: refreshInventory
- input: RefreshIventorySoapIn(AuthenicationHeader, RefreshRequest(String LocationName, RefreshItem[] Items))
  - RefreshItem(String PartNumber,String UPC,String LocationUPC,Int BinId,Int Quantity,Double LTD,int Floor,int SafetyStock)
- output:refreshInventorySoapOut(String RefreshResponse)
- SQL: UPDATE LocationProduct LP,RetailerProduct RP,ContainedInBin CB
  SET LP.LTD = LTD AND LP.SafeStockLimit = SafetyStock AND CB.BinId = BinId AND RP.UPC = UPC AND CB.OnHand = Quantity
  WHERE LP.RetailerProductId = RP.Id AND CB.LocationProductId = LP.Id;

Get Inventory

- API call: getInventory
- input: createFulfillerRequest:
  AuthenticationHeader(+int UserID, +int FulfillerID)

InventoryRequest(ManufacturerCatalog catalog,
ItemQuantities[] Quantities, String [] LocationNames,
RequestLocation location, InventoryRequestType Type,
int Limit, boolean IgnoreSafetyStock,
boolean IncludeNegativeInventory, boolean OrderByLTD)

- ItemQuantity(String PartNumber, String UPC, String LocationUPC, int Quantity)
- RequestLocation(String Unit, +int Radius, String PostalCode, double Latitude,
double Longitude, String CountryCode)
- InventoryRequestType (“ALL” || “PARTIAL” || “ANY” || “ALL_STORES”)

- output: InventoryResponse:
String LocationName, int CatalogID, int ManufacturerID, int OnHand, int Available,
String PartNumber, String UPC, String LocationUPC, double LTD, int Floor,
int SafetyStock, boolean STHEnabled, boolean, RestockEnabled,
boolean PickupEnabled, String CountryCode, double Distance
- SQL:
SELECT L.Name, P.ManufacturerId, P.CatalogId, CB.OnHand,
(CB.OnHand - CB.Allocated), ?, RP.UPC, LP.LTD, ?, LP.SafeStockLimit, ?, ?, ?, ?,
(Insert "Distance Between 2 Longitudes and Latitudes" Formula here)

FROM Location L, LocationProduct LP, RetailerProdut RP, Product P,
CatalogServedByLocation CL, ContainedInBin CB,
(SELECT Name FROM LocationNames LN),
(SELECT UPC, Quantity FROM ItemQuantity IQ)

WHERE LN.Name = L.Name AND
L.InternalFulfillerLocationId = CL.InternalFulfillerLocationId AND
CatalogID = CL.CatalogId AND ManufacturerID = CL.ManufacturerId AND
CL.InternalFulfillerLocationId = LP.InternalFulfillerLocationId AND
LP.RetailerProductId = RP.Id AND RP.UPC = P.UPC AND
LP.Id = CB.LocationProductId AND CB.OnHand >= IQ.Quantity AND
"Calculated Distance" < Radius AND
CB.OnHand >= 0 (if IncludeNegativeInventory == false) AND
CB.OnHand >= LP.SafeStockLimit (if IgnoreSafetyStock == false)

ORDER BY OrderByLTD (if OrderByLTD == true);

Get Item Locations By Fulfiller ()

- API call: getItemLocationsByFulfiller
- input: getItemLocationsByFulfillerRequest
AuthenticationHeader(+int UserID, +int FulfillerID)
getItemLocationsByFulfiller->InventorySearchRequest(ArrayofFulfillerId
FulfillerIds,int LocationID,String PostalCode,String PartNumber,String UPC,String LocationUPC)
FulfillerId(int fulfillerId)

- output:getItemLocationsByFulfillerResponse
  GetItemLocationByFulfillerReturn
  InventorySearchResponse(int FulfillerId,int FulfillerLocationID,int RetailerID,int RetailerLocationID,String LocationName,String PartNumber,String UPC,String LocationUPC,int Available,int OnHand,int Allocated,Double Distance)
- SQL

  select R.FulfillerId,L.InternalFulfillerLocationId,RP.Id,L.Name, RP.UPC, CB.OnHand,CB.Allocated, (Insert "Distance Between 2 Longitudes and Latitudes" Formula here)

  from RetailerProduct RP, LocationProduct LP, Retailer R,Location L,ContainedInBin CB

  where LP.InternalFulfillerLocationId = LocationID AND RP.UPC = RP.UPC AND RP.ID = LP.RetailerProductId AND R.FulfillerId = RP.FulfillerId AND L.InternalFulfillerLocationId = LP.InternalFulfillerLocationId AND CB.LocationProductId = LP.Id;

Allocate Inventory
- API call: allocateInventory
- input: allocateInventoryRequest(AuthenticationHeader, allocateInventory(UpdateRequest request(FullfillmentLocationCatalog FulfillerLocationCatalog, UpdateItem[] Items)))
  - UpdateItem(String PartNumber, String UPC, String LocationUPC, int Quantity, +int OrderID, +int OrderItemID, +int ShipmentID, +int FulfillerLocationID)
- output: allocateInventoryResponse()
  UPDATE ContainedInBin CB,
  (select LP.InternalFulfillerLocationId from LocationProduct LP)Loc,
  (select RP.UPC from RetailerProduct)Retail
  SET CB.Allocated = Items AND CB.OnHand = Quantity
  WHERE Loc = FulfillerLocationId AND Retail = UPC;

De-allocate inventory
- API call: deallocateInventory
- input: deallocateInventoryRequest(AuthenticationHeader, deallocateInventory(updateRequest request (FullfillmentLocationCatalog FulfillerLocationCatalog, UpdateItem[] Items)))
  - UpdateItem(String PartNumber, String UPC, String LocationUPC, int Quantity, +int OrderID, +int OrderItemID, +int ShipmentID, +int FulfillerLocationID)))
- output: deallocateInventoryResponse()
- SQL:
  UPDATE ContainedInBin CB,
  (select LP.InternalFulfillerLocationId from LocationProduct LP)Loc,
  (select RP.UPC from RetailerProduct)Retail
  SET CB.Allocated = Items AND CB.OnHand = Quantity

```
WHERE Loc = FulfillerLocationId AND Retail = UPC;
```