**Business Requirement Summary:**

This exercise is a pre-cursor to the Capstone project. The Capstone project involves building a “mini” module for a pizza franchise. In this exercise set, you will build the base table and application for the Products that will be sold thru franchise stores. The Products table will be used by all stores. You will also build the base tables and application that will allow a store to take Orders. You will be able to use these components as the basis for more advanced phases of the Capstone project.

In the Pizza franchise, products will be grouped into categories such as Pizza, Pasta, Salad, Dessert. There is also a subcategory for each – (Pizza) One-Topping, (Pizza)Two-Topping, (Pizza)Custom, (Pasta)White Sauce, (Pasta)Red Sauce, (Salad)Ceasar, etc.. You can create your own product categories and subscategories.

Orders types will be Dine-In, Delivery, Cater, and Takeout. Use the following value/description type pairs when defining OrderType:

* + - CATER / Cater
    - DINEIN / Dine In
    - DEL / Delivery
    - TAKE / Takeout

*Naming:* To avoid name conflicts, **add your initials to the beginning of each object you create**. For example, nsTrnRWOrdersHeader.

*Lifecycle:* Create all objects in your individual student lifecycle

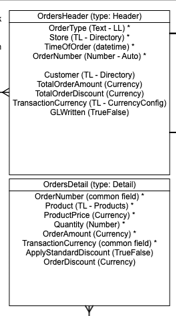
*Family:* Create all objects in thePlaypen product family

*Module:* Personal

**Products:**

|  |  |
| --- | --- |
| **Task** | **Notes** |
| **Products:** Create the Data Items, List Lookups, and table for Products | * Think through your design of the table.   + What table type should it be?   + If the Orders application is going search the products table for products, what should the type ahead fields be?   + What about List Lookups needed? * The Product Table contains the following fields:   + ProductName (Text)   + ProductCategory (Text – List Lookup)   + ProductPrice (Currency)   + ProductSubCategory(Text – List Lookup)   + Description (Textarea) * Create the Data Items for the products table * Create the List Lookups required   + What kind of List Lookup is the ProductSubCategory? * Create the Products Table. Include the following attributes:   + Create the primary and secondary typeaheads (If someone was going to search the product table for a product, what is the most likely field they would use?)   + Product Name should be sorted A-Z   + ProductName should be unique |
| Create the Products Application  *This is where stores create and manage their products.* | * What Application Type and Application Style would you choose?   Layout:   * Header Row should have the four main fields: name, category, price, subcategory * Description should be across the whole page   Test your application and make sure:  You can define various products with different categories and subcategories   * ProductName is unique * ProductName sort A - Z |
| Create a Product Search Application | All table lookups can have a search application. Often it makes sense to show the same search application for every table lookup on a table without having to define it everywhere it's used. By defining a default search application the table, any table lookups will use that search app unless it's overridden at the application.   * Create App   + App Name: ProductsSearch   + App Type - Standard   + App Style - Mini   + Table: your products table * Attach it as the default search app (in the Relationships page) of your Products table |

**Orders:**



|  |  |
| --- | --- |
| **Task** | **Details** |
| Create Data Items | * Notes:   + OrderType     - List Lookup – Create OrderType LL   + OrderNumber – use Automatic Numbers   + TimeOfOrder - define it to default to the current time   + TotalOrderAmount and TotalOrderDiscount     - No negative values |
| Create Tables | **Header**   * Attributes:   + Store filters to only show internal orgs   + Customer filters to only show customers (hint: filter on the ContactRoleGrouping field) * Indexes:   + Sort TimeOfOrder (Z - A)   + OrderNumber is unique   **Detail**   * ~~Requirements~~   + ~~Product filters to only show ACTIVE products~~ * Indexes:   + Can only have a Product once per order (hint: how are detail record tied to the header?)   **Header/Detail**   * Ensure the detail table is added to the detail tables subtable and not the single detail field * Add common fields for:   + OrderNumber (header) to OrderNumber (detail). This defaults the value.   + Store (header) to Store (detail). This defaults the value. We are adding this so we can filter discounts for the specific store.   + TransactionCurrency (header) to TransactionCurrency (detail). We need it in the detail so we can do currency calculations. |
| Create Automatic Numbers | * Create automatic number definition for OrderNumber |
| Create Applications | (info)When creating the applications, you'll notice that there are many fields that come through as related table lookups fields for the Store and Customer fields and many aren't applicable. To exclude them from showing up in the application, what should you do????  **Header App**   * Layout:   + These fields should be disabled:     - Order Number     - Total Discount     - Total Order     - TransactionCurrency     - GL Written   + Two pages: 1) Details and 2) Support     - The header row should have TimeOfOrder, ~~Customer~~, OrderType,, and Store     - The Support page should have Customer, TransactionCurrency, GL Written     - When defining the pages and rows, ensure "Show When Empty" is selected for the Details page     - Also define a # of columns per row to add a consistent look * Attributes:   + Store should have a search action. Use the base search app that already exists. Add a search action:     - Search Field: Store field     - Search App: OrganizationalUnitsSearch     - Setting Name: InternalOrganizationFilter     - Form: List     - Data Mappings: the title should say “Search Stores”   + Customer should also have a search action - search the available search apps over the Directory and pick the customer one. The   **Detail App:**   * Layout   + These fields should be disabled:     - Discount Amount     - Order Amount   + List View:     - In this order: Product, Price, Quantity, ApplyStandardDiscount, DiscountAmount, OrderAmount   + Field Dependencies:     - Since we are storing the transaction currency on the record, we can default the currency code for the currency fields. Use Field Dependencies (hint: Field Control) to default the currency. (info)This is a best practice: a transaction has a currency which defaults the code for the currency fields. This prevents users from inserting invalid data. |

|  |  |
| --- | --- |
| Create the OrdersJoin table | Create a Join between Orders Detail and Orders Header  Hint: similar to GeneralJournalJoin |
| Create a Join Application over the OrdersJoin table | The OrdersJoin application provides a view of the detail records of all orders so the user can quickly find information in the order. The double-click and edit are overridden with an application link to the Orders H/D the detail record is attached to. Therefore, when the user double-clicks/edits a record, they will be taken to the detail view of the Orders Header/Detail application where they can view the whole transaction.  Hint: From the menu, look for General Ledger. If you have the Information toggle turned on, you will see that the General Ledger application on the menu is a list application called GeneralJournalJoin built over the GeneralJournalJoin table. This is a similar application to what you are building with the OrdersJoin. |

**Create a Join Table and a Join Application for Orders:**

|  |  |
| --- | --- |
| EXTRA: Create a field action to Customer Search  (Note – this can be done during the Capstone) | Since orders are a quick process, we need an easy way for the stores to create a new customer.   * Add a field action on CustomerName\_Name. It will show a mini app and go to the detail view. * Setup an application link with the CustomerSearch app. * Use the CustomerSearchSettings app settings. |

**Customer Search**