1. Добавить страницу для вывода списка материалов **PageMaterialList**.
2. Задать Title для страницы.
3. Сделать кнопку «Назад».
4. Добавить StackPanel в качестве глобального контейнера для всех элементов страницы.
5. **Фильтрация списка по типу материала**Внутри глобальной StackPanel создать еще одну, с горизонтальной ориентацией, в нее разместить TextBlock и ComboBox для выбора типа материала, а также кнопку «Показать».
6. **Вывод данных в таблицу**  
   Под стекпанелью разместить **DataGrid** для вывода табличных данных, задать ей имя GridList и свойство только для чтения IsReadOnly=true (для запрета редактирования), а также AutoGenerateColumns=”False” (чтобы не генерировалось лишних столбцов со служебной информацией).

Внутри DataGrid в теге <Data.Columns> прописать все столбцы из нужной таблицы с помощью вложенных тегов   
<DataGridTextColumn Header=”Заголовок столбца” Width=”\*”/>.

1. В коде страницы PageMaterialList, после InitializeComponent() установить источник данных для таблицы данных:

GridList.ItemSource = OdbConnectHelper.entObj.<имя таблицы-источника>.ToList()

1. В разметке страницы PageMaterialList выполнить привязку столбцов DataGrid к столбцам таблицы-источника:

Binding = {Binding <Наименование столбца в таблице-источнике>}

1. Для тех полей, где вместо ключей желательно выводить соответствующие текстовые данные, надо прописать следующее:

Binding = {Binding <Связанная таблица.Поле подстановки>}

1. Добавить в конец DataGrid еще один столбец, в нем будут содержаться кнопки для выполнения функционала (просмотр, редактирование, удаление):

<DataGridTemplateColumn Width=”\*”>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button

Name=”ИмяКнопки”

Click=”Обработчик\_Клика”

Content=”Надпись на кнопке” />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn Width=”\*”>

1. Для тех полей, где вместо ключей желательно выводить соответствующие текстовые данные, надо прописать следующее:

Binding = {Binding <Связанная таблица.Поле подстановки>}