-
-
-
-
-

Access 2016: Export Data from Microsoft Access

You can export data from Access to a variety of formats by using the External Data tab on the Ribbon.

Access 2016 includes options for exporting data to other sources. You can export to a variety of formats, including:

- Excel file
- Text file (CSV, TSV, etc)
- XML file
- PDF file
- XPS file
- Email
- Word document
- Sharepoint list
- HTML document
- Another Access database
- ODBC datasource (such as SQL Server)

You can export from a table, a query, a form, or a report.

Today we'll export two different queries to two different formats:

1. We'll export our `Iron Maiden Albums` query to an Excel spreadsheet.
2. We'll export our `Albums from the last 25 years` query to an XML file.

Export a Query to Excel

Let's export the results of the `Iron Maiden Albums` query that we saved [previously](#).
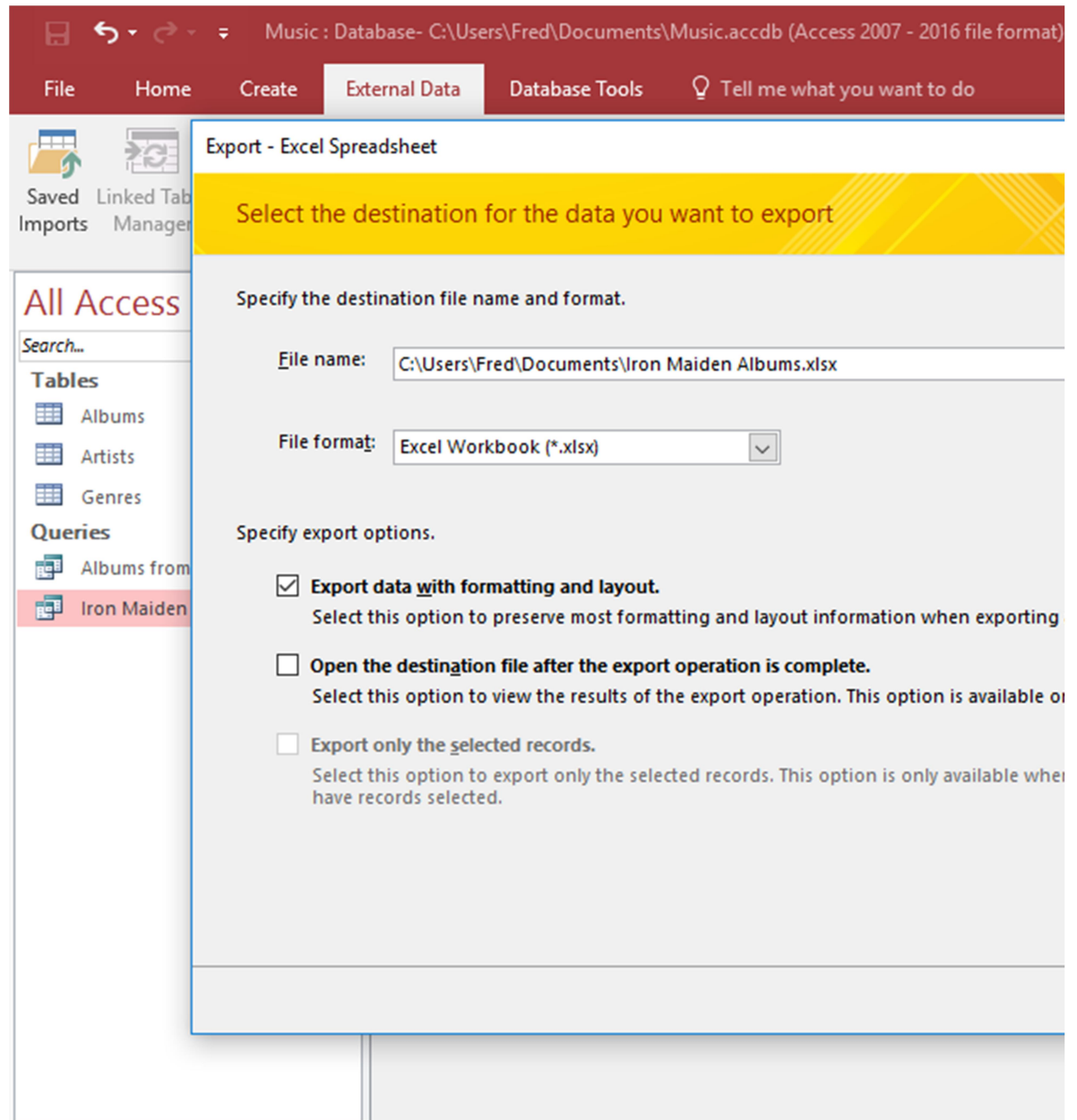
In this case, we'll export the query results to an Excel spreadsheet.

1.
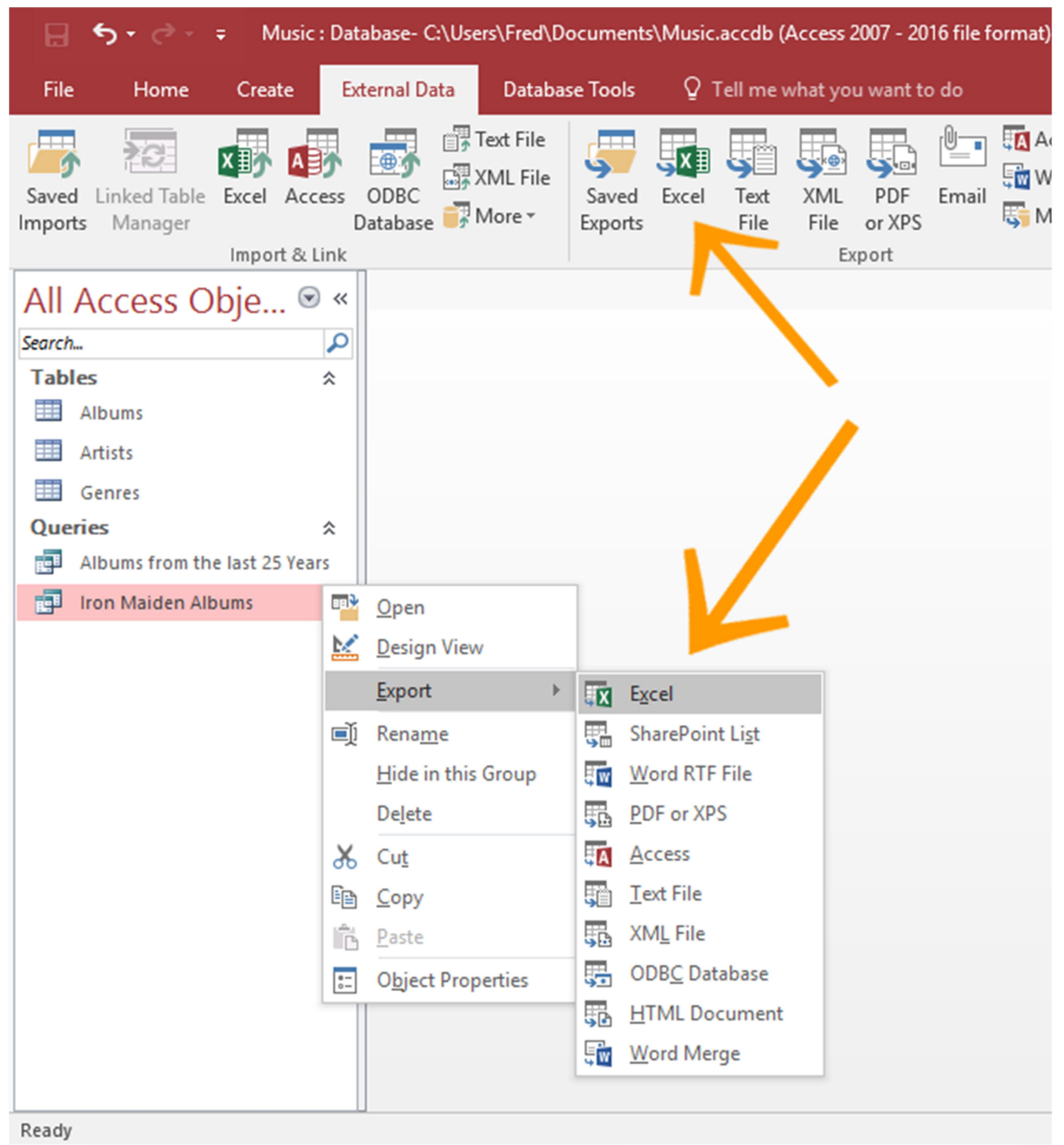
Launch the Export Dialog

Right-click on the query in the left menu (in our case, the `Iron Maiden Albums` query) and select `Export > Excel` from the contextual menu.

Alternatively, you can select the query in the left menu, then click `Excel` from the export options in the Ribbon (on the `External Data` tab).
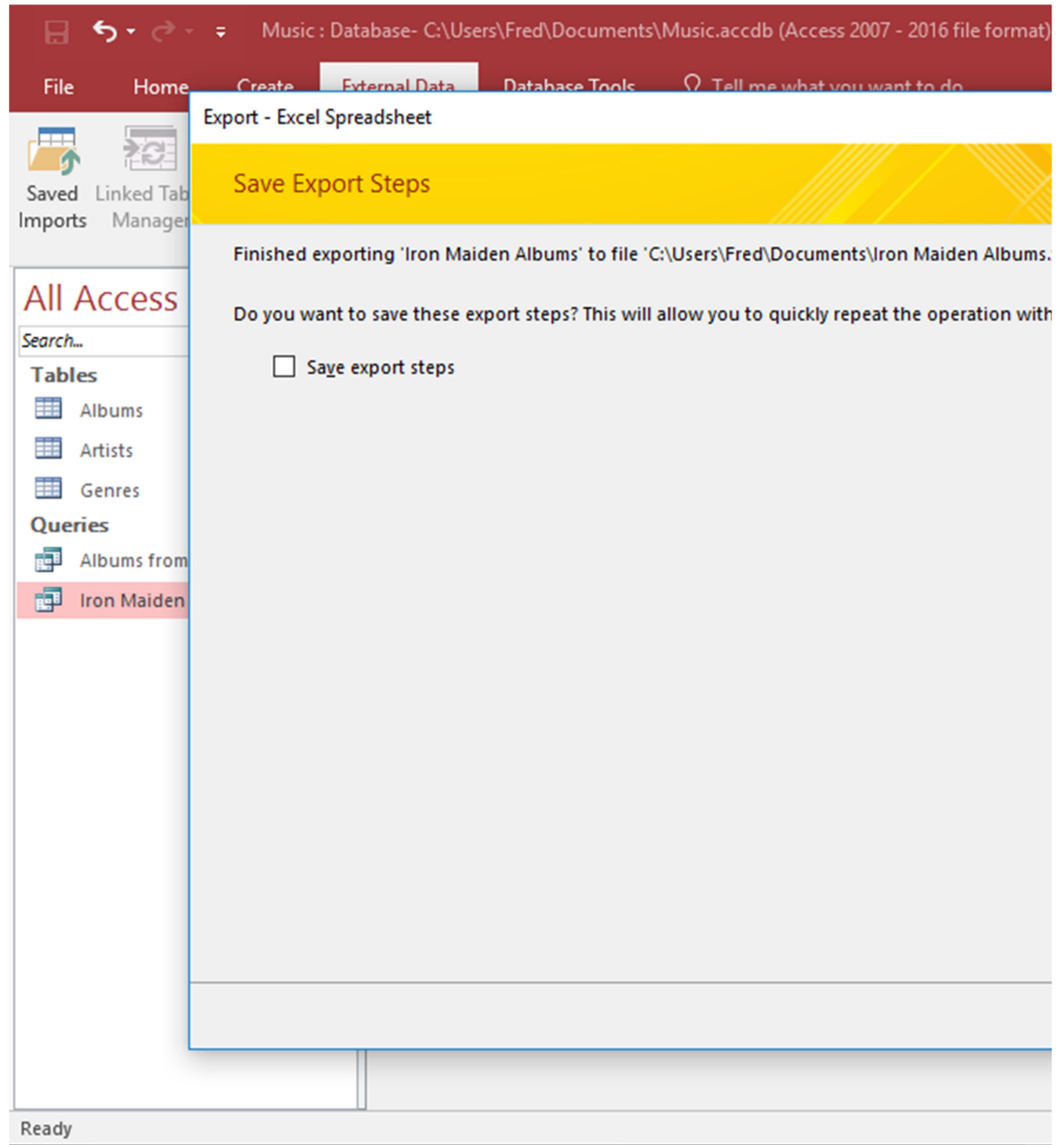

2.

Export Options

Review the options. In our case, we'll keep the file name as Access has suggested. However, we'll put a tick next to `Export data with formatting and layout`.
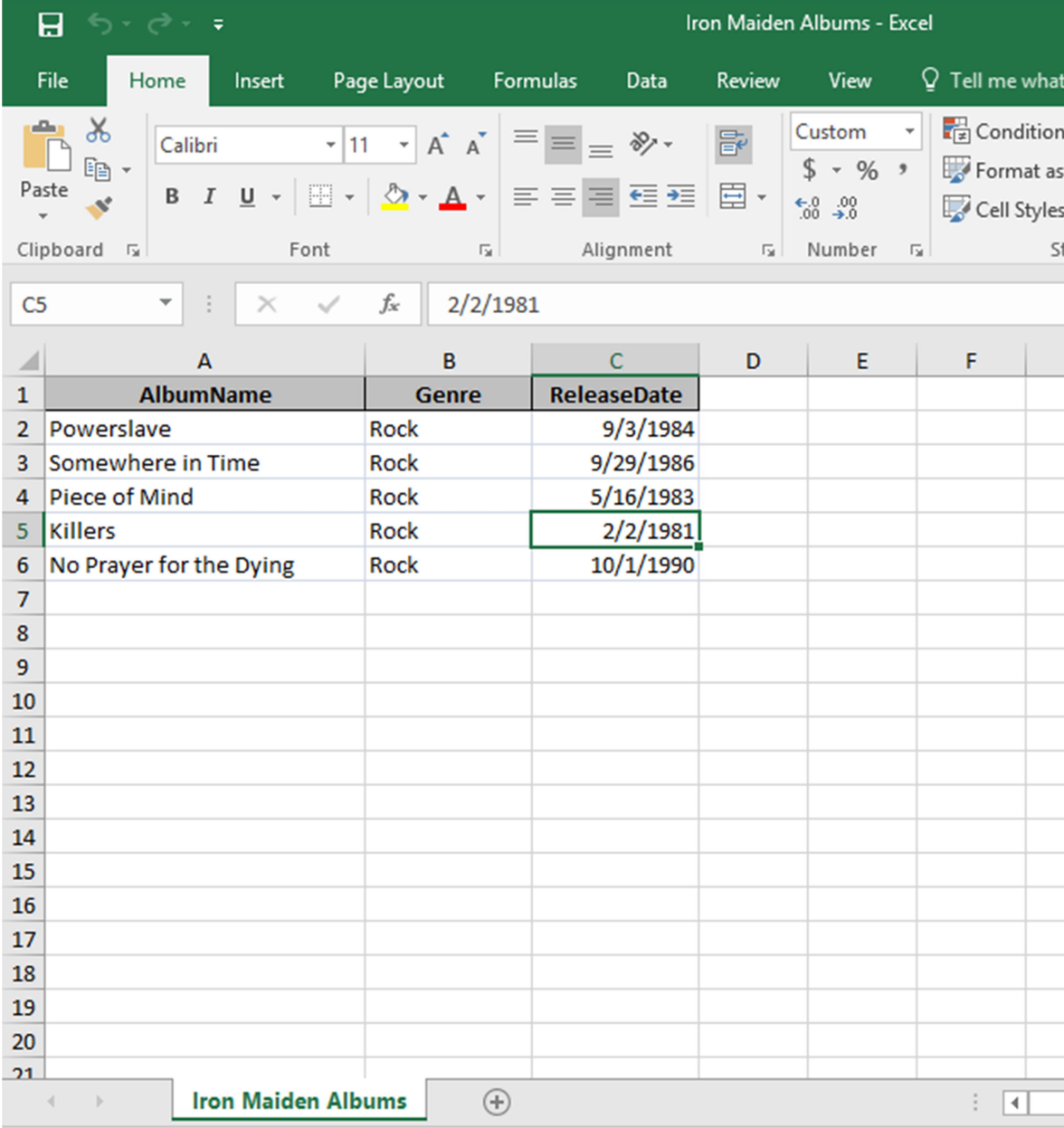
Click `OK` to export the data.

3.

Save Export Steps?

Access gives you the option of saving the steps you just took, so that you can re-run them again later.

In our case, we'll give it a miss. Click `Close` to finish.

4.

View the Exported File

You can now check your exported file in Excel. Navigate to the location you chose at step 2 and open the file.

You can save yourself from having to navigate to the file by checking `Open the destination file after the export operation is complete` at step 2.

Export to an XML File

Now we'll export our `Albums from the last 25 Years` query (that we [built with the Expression Builder](#)) to an XML file.
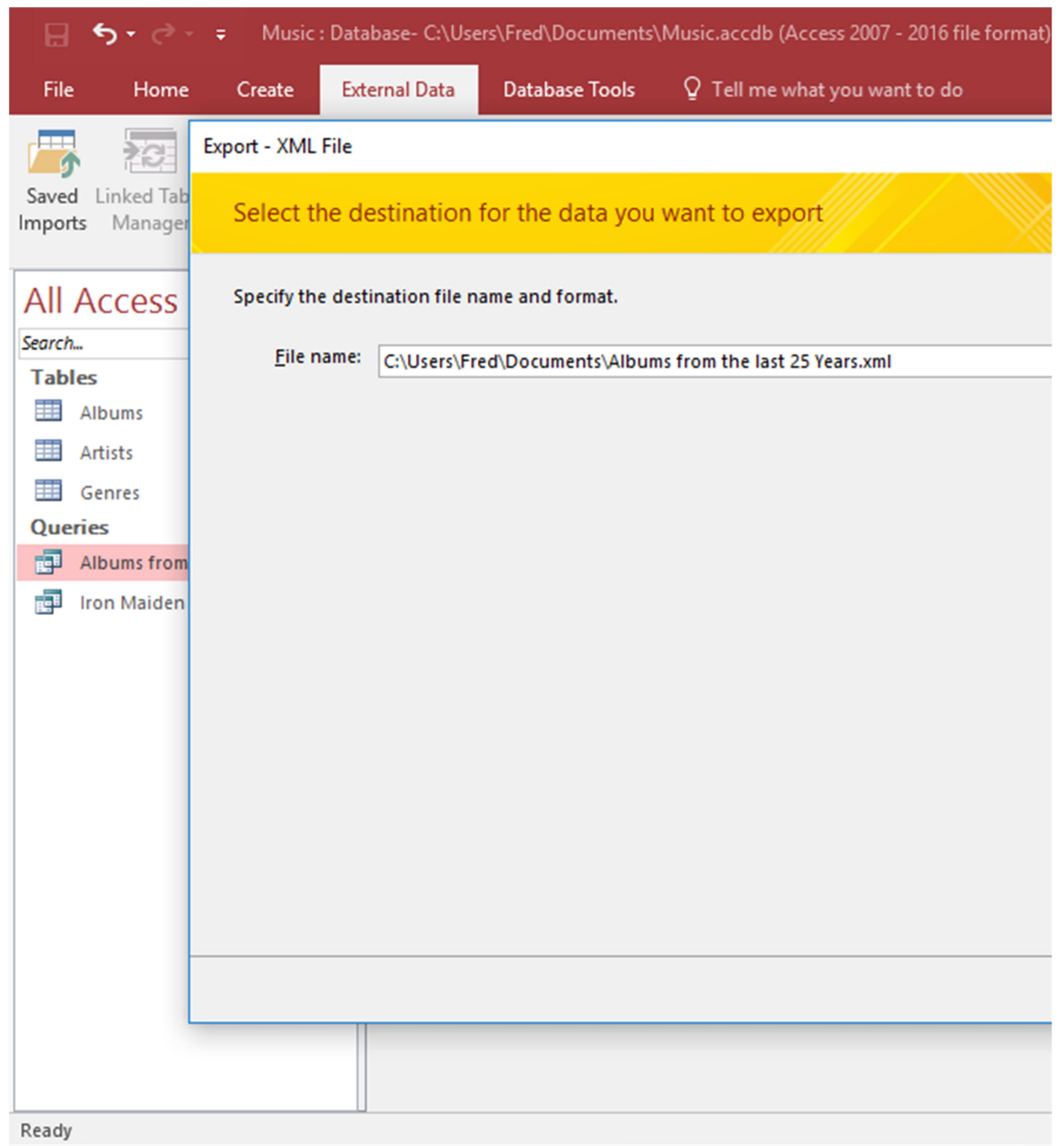
1.
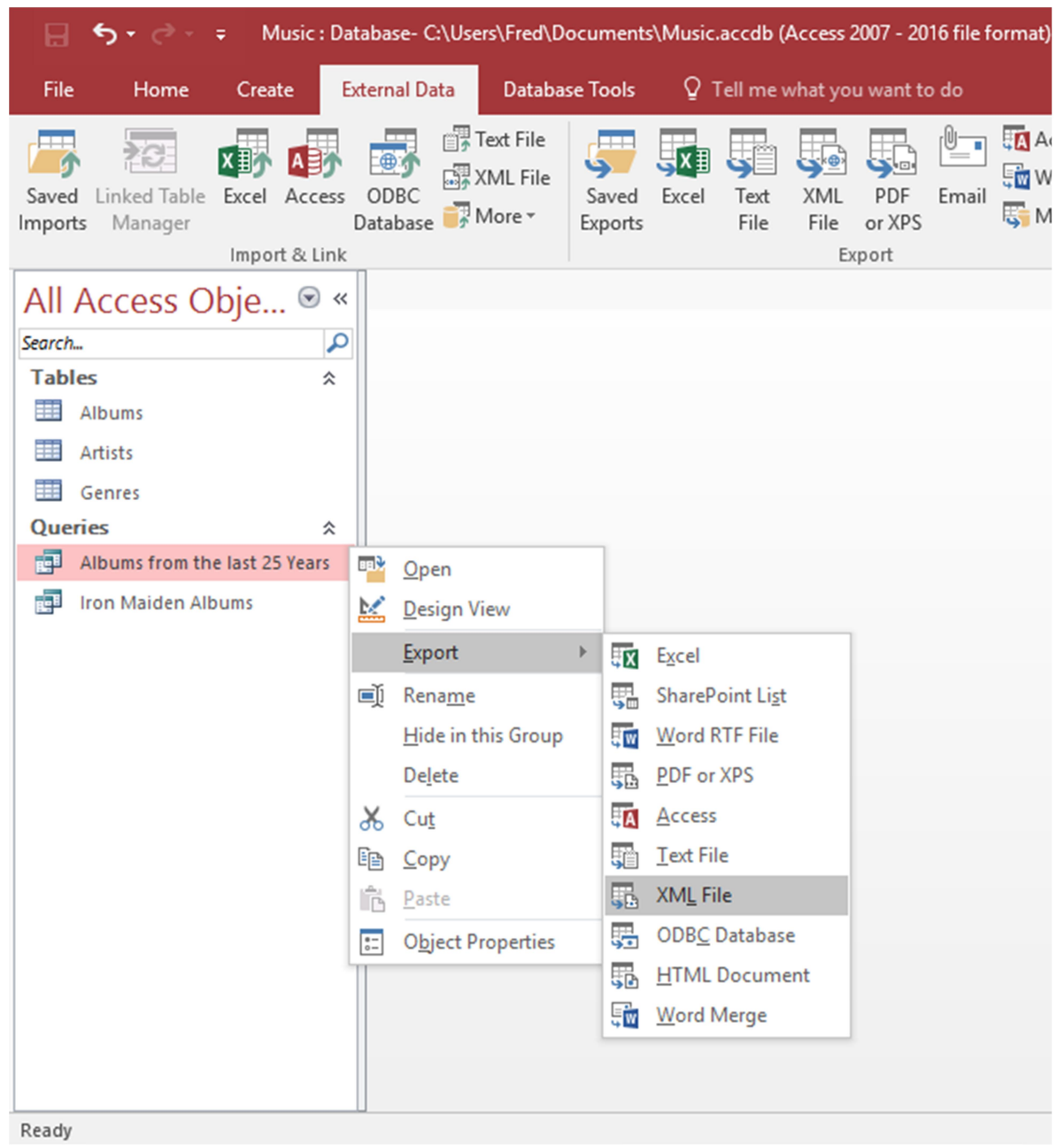
Launch the Export Dialog

Right-click on the query in the left menu (in our case, the `Albums from the last 25 Years` query) and select `Export > XML File` from the contextual menu.

Alternatively, you can select the query in the left menu, then click `XML File` from the export options in the Ribbon (on the `External Data` tab).

2.

File Destination

Review the file destination. In our case, we'll keep the file name and destination as Access has suggested.

Click OK.
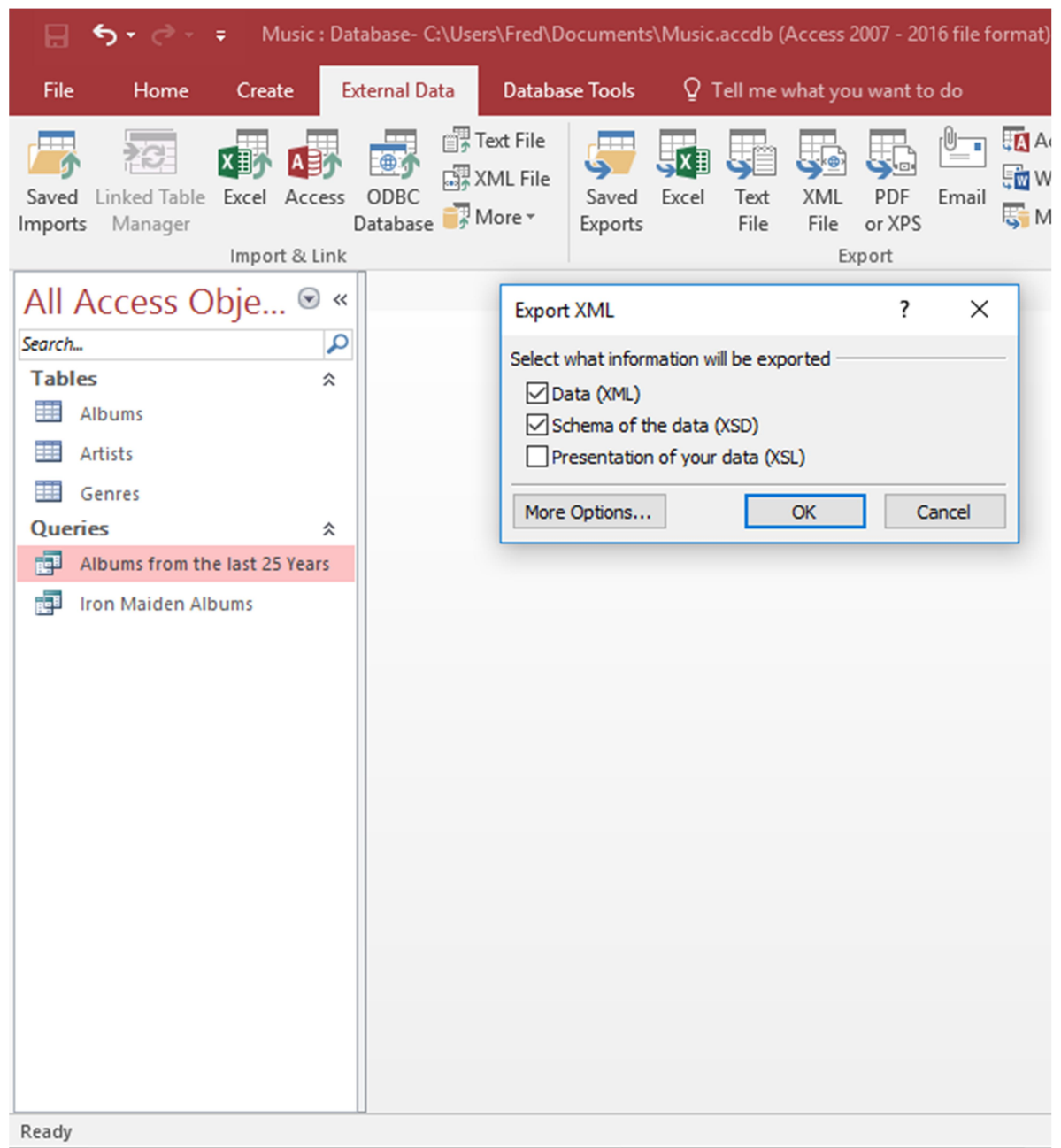
3.

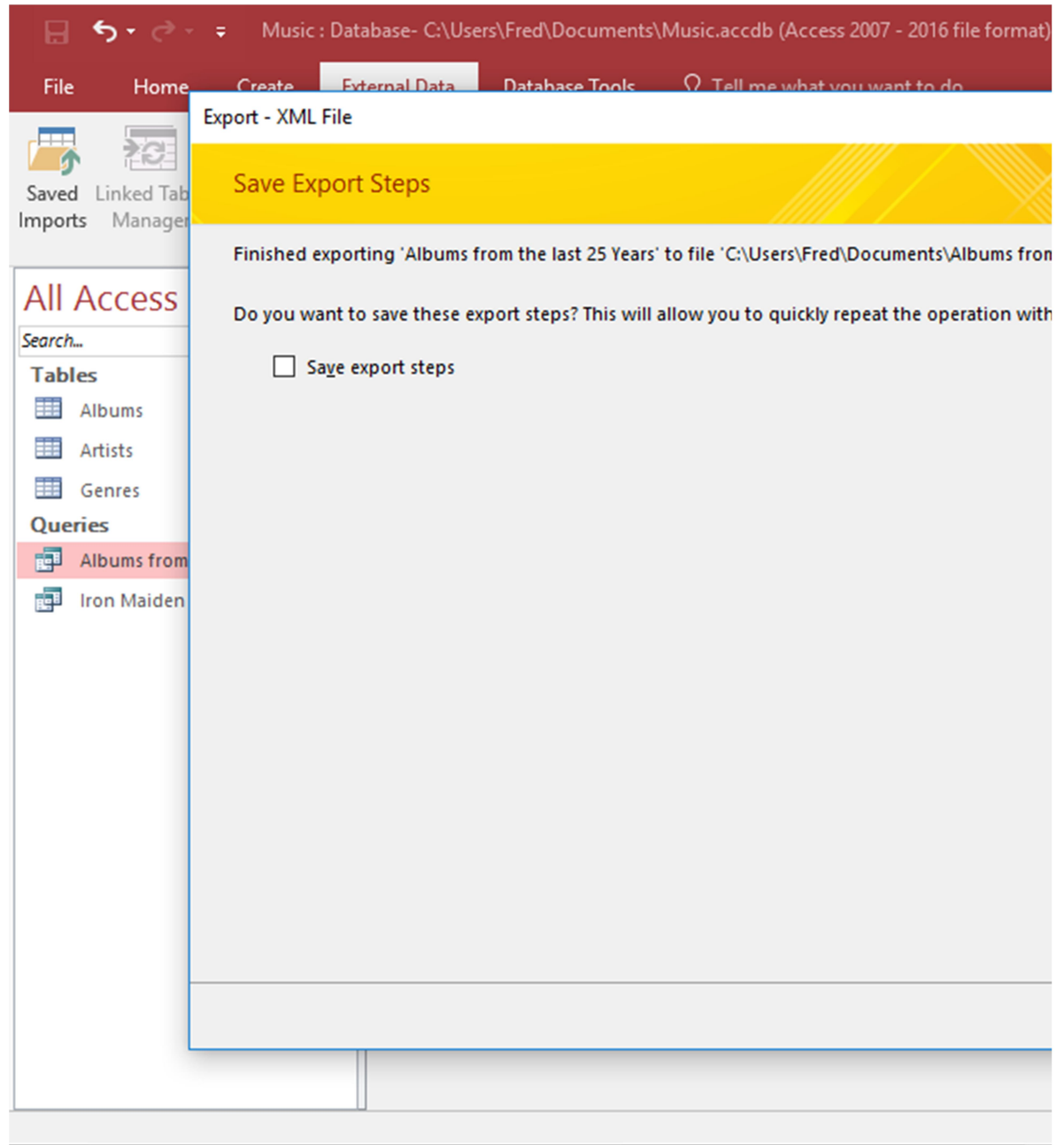# Select Export Information

Review/change the export information as required. In our case, we'll just keep the first two options checked. This

will generate two files: the XML file and an XSD file (which will hold rules, such as the data types).

Click OK to export the files.

4.

Save Export Steps?

Access gives you the option of saving the steps you just took, so that you can re-run them again later.

In our case, we'll give it a miss. Click `Close` to finish.

5.   Review the File/s

Navigate to the location you chose for the files and open them.

The XML File

Here's what the resulting XML file contains — all two albums from the last 25 years:

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Albums%20from%20the%20last
%2025%20Years.xsd" generated="2016-07-17T23:39:57">
<Albums_x0020_from_x0020_the_x0020_last_x0020_25_x0020_Ye
ars>
<ReleaseDate>2005-09-28T00:00:00</ReleaseDate>
<AlbumName>Snoopified</AlbumName>
<Genre>5</Genre>
<ArtistName>Snoop Dogg</ArtistName>
</Albums_x0020_from_x0020_the_x0020_last_x0020_25_x0020_Y
ears>
<Albums_x0020_from_x0020_the_x0020_last_x0020_25_x0020_Ye
ars>
<ReleaseDate>1996-11-12T00:00:00</ReleaseDate>
<AlbumName>Tha Doggfather</AlbumName>
<Genre>5</Genre>
<ArtistName>Snoop Dogg</ArtistName>
</Albums_x0020_from_x0020_the_x0020_last_x0020_25_x0020_Y
ears>
</dataroot>
```

The XSD File

The XSD file contains the schema for the XML file. The schema defines rules about the structure of the data. Here's what the resulting XSD file contains:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:od="urn:schemas-microsoft-com:officedata">
<xsd:element name="dataroot">
<xsd:complexType>
<xsd:sequence>
<xsd:element
ref="Albums_x0020_from_x0020_the_x0020_last_x0020_25_x002
0_Years" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="generated" type="xsd:dateTime"/>
</xsd:complexType>
</xsd:element>
<xsd:element
name="Albums_x0020_from_x0020_the_x0020_last_x0020_25_x00
20_Years">
<xsd:annotation>
<xsd:appinfo/>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ReleaseDate" minOccurs="0"
od:jetType="datetime" od:sqlSType="datetime"
type="xsd:dateTime"/>
<xsd:element name="AlbumName" minOccurs="0"
od:jetType="text" od:sqlSType="nvarchar">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:maxLength value="255"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Genre" minOccurs="0"
od:jetType="longinteger" od:sqlSType="int"
type="xsd:int"/>
<xsd:element name="ArtistName" minOccurs="0"
od:jetType="text" od:sqlSType="nvarchar">
```

```xml
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:maxLength value="255"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```