

## **Making Modifications to IMPORT\_IBR\_DATA97.mdb Queries in ACCESS 97**

Some local IBR systems include additional local-use only data elements. This appendix provides instructions on how to modify the queries in IMPORT\_IBR\_DATA97.mdb to import additional IBR data elements into Access database tables.

Appendix A lists and describes the queries used to create IBR tables and load data into them. There are two queries per IBR segment that need to be modified to accommodate additional data elements. They are as follows:

### **ADMINISTRATIVE**

mkARRESTEEA\_Table  
appADMINISTRATIVE\_Data

### **OFFENSE**

mkOFFENSE\_Table  
appOFFENSE\_Data

### **PROPERTY**

mkPROPERTY\_Table  
appPROPERTY\_Data

### **VICTIM**

mkVICTIM\_Table  
appVICTIM\_Data

### **OFFENDER**

mkOFFENDER\_Table  
appOFFENDER\_Data

### **ARRESTEEA**

mkARRESTEEA\_Table  
appARRESTEEA\_Data

### **ARRESTEEB**

mkARRESTEEB\_Table  
appARRESTEEB\_Data

Each mk--\_Table query is used to declare the data elements stored in each table by assigning a name and data type for each data element. The general form of the mk--\_Table query is:

```
CREATE TABLE TableName(DataElementName1 DATA_TYPE1, DataElementName2 DATA_TYPE2...)
```

The data type of the additional data element will be listed on the IBR data file record layout. IBR data types are usually text (or character) or numeric (integer or decimal).

*To add a text data element to a table, the modification to the mk--\_Table query will be of the form:*

**,DataElementName TEXT(HH),**

where DataElementName is the name of the data element to be added, TEXT is the required data type declaration keyword, and HH is the length of the text data element. The length of the text data element will be listed on the IBR data file record layout.

*To add an integer numeric data element to a table, the modification to the mk--\_Table query will be of the form:*

**,DataElementName LONG,**

where DataElementName is the name of the data element to be added and LONG is the required integer data type declaration keyword.

*To add a decimal numeric data element to a table, the modification to the mk--\_Table query will be of the form:*

**,DataElementName DOUBLE,**

where DataElementName is the name of the data element to be added and DOUBLE is the required decimal data type declaration keyword.

Each app--\_Data query is used to extract data from a particular IBR data record and append the data to a particular IBR data table. The general form of the app--\_Data query is:

```
INSERT INTO TableName ( DataElement1, DataElement2...)  
SELECT IIf(MID([Field1],StartPositionData1,LengthOfData1)=' ',Null,MID([Field1],StartPositionData1,LengthOfData1)),  
IIf(MID([Field1],StartPositionData2,LengthOfData2)=' ',Null,MID([Field1],StartPositionData2,LengthOfData2)),...  
FROM LinkedTableName
```

The app--\_Data query will be modified to declare where the additional data element is located in the IBR data record and where to append the additional data element in the IBR table.

First, the name of the additional data element must be added to the INSERT INTO SQL statement. For example, a new data element called DataElement3 would be added as follows:

**INSERT INTO TableName(DataElement1, DataElement2, DataElement3)**

Next we modify the SELECT SQL statement to locate the additional data element in the IBR data record. *If the data element is a text or character data type, the SQL will be of the form:*

**,IIF(MID([Field1],AA,BB)='CC', NULL, MID([Field1],AA,BB))**

where

AA = starting position of the data element in the record,  
BB = length of the data element in the record, and  
CC = blank spaces the number of which is equal to the value of BB.

*If the data element is a numeric data type, the SQL will be of the form:*

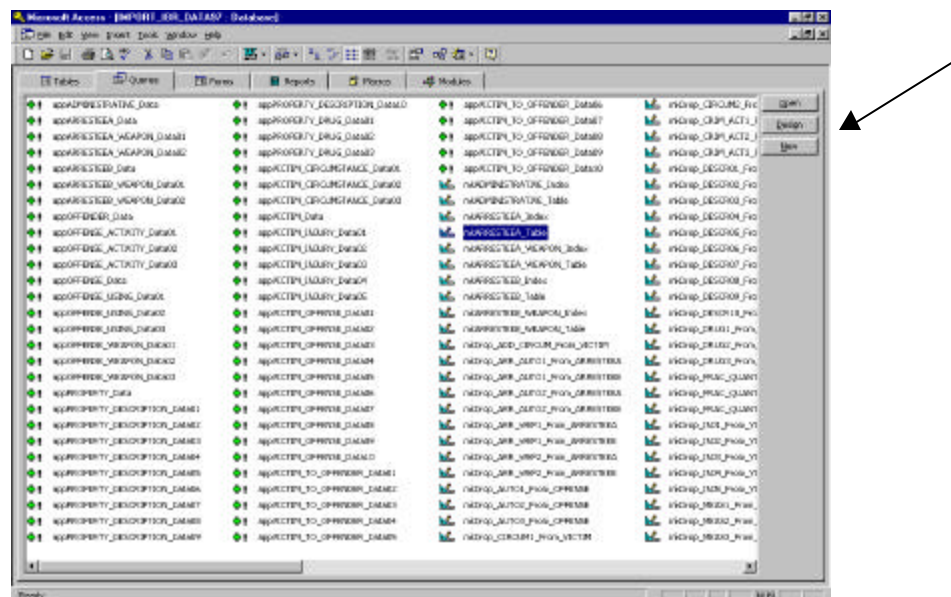
**,IIF(MID([Field1],AA,BB)='CC', NULL, VAL( MID([Field1],AA,BB)))**

*Make modifications to all queries before running macros M01 through M12.*

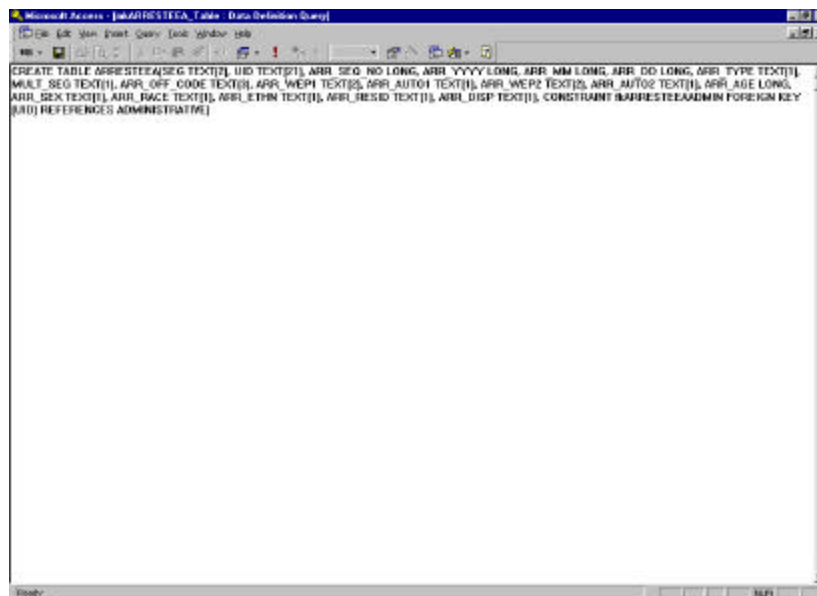
### Example: Add additional data element to ARRESTEEA Segment table

A local IBR data file includes a data element called ARR\_DRG\_TYPE in the ARRESTEEA Segment data record. The local IBR data file record layout shows that data element ARR\_DRG\_TYPE begins as position 76 in the data record and is represented by a value two-characters in length.

We must first add data element ARR\_DRG\_TYPE to the ARRESTEEA table structure by modifying table creation query mkARRESTEEA\_Table. To open the query, highlight the query name in the Query View window, then press the Design button (indicated by arrow below):



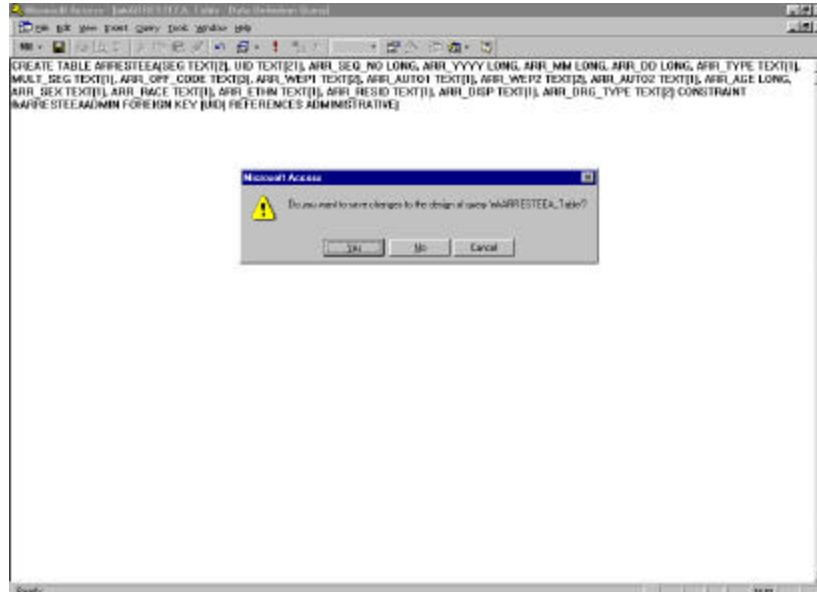
The SQL window appears:



To add a field for data element ARR\_DRG\_TYPE, enter the field name, field data type, and data length as follows:

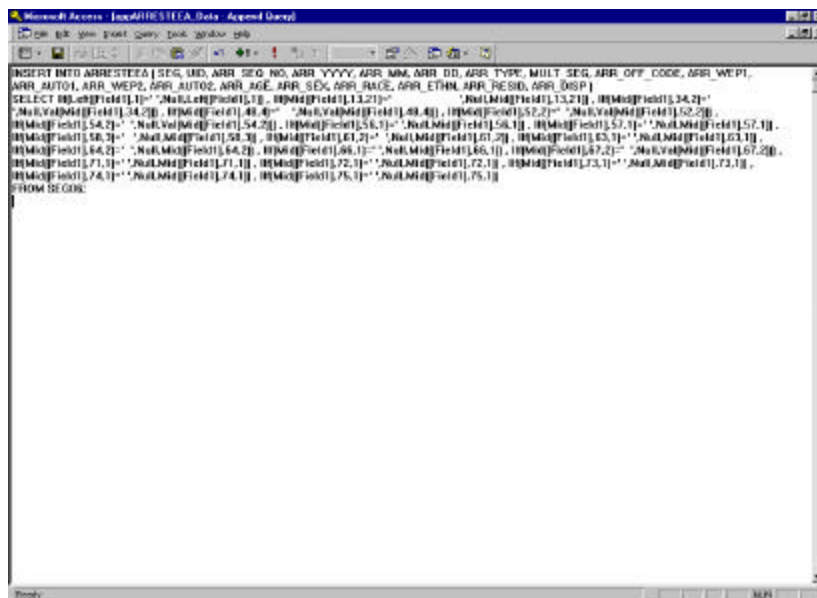
```
CREATE TABLE ARRESTEEA(SEG TEXT(2), UID TEXT(21), ARR_SEQ_NO LONG, ARR_YYYY LONG, ARR_MM LONG,  
ARR_DD LONG, ARR_TYPE TEXT(1), MULT_SEG TEXT(1), ARR_OFF_CODE TEXT(3), ARR_WEP1 TEXT(2), ARR_AUTO1  
TEXT(1), ARR_WEP2 TEXT(2), ARR_AUTO2 TEXT(1), ARR_AGE LONG, ARR_SEX TEXT(1), ARR_RACE TEXT(1),  
ARR_ETHN TEXT(1), ARR_RESID TEXT(1), ARR_DISP TEXT(1), ARR_DRG_TYPE TEXT(2), CONSTRAINT  
fkARRESTEEAADMIN FOREIGN KEY (UID) REFERENCES ADMINISTRATIVE)
```

After entering the information for the new data element, press the query close button (the lower 'X' on the right of the screen indicated by the arrow above). A dialog box will appear:



Since you want to save the changes, click the 'Yes' button or press 'Enter.'

Next we modify append query appARRESTEEA\_Data to load the additional data element to the ARRESTEEA table. Open the SQL window for query appARRESTEEA\_DATA:



Since table ARRESTEEA will now include an additional data field called ARR\_DRG\_TYPE and we want to load data into it, enter ,ARR\_DRG\_TYPE (must include the leading comma) at the end of the "INSERT INTO" line as follows

```
INSERT INTO ARRESTEEA ( SEG, UID, ARR_SEQ_NO, ARR_YYYY, ARR_MM, ARR_DD, ARR_TYPE, MULT_SEG,  
ARR_OFF_CODE, ARR_WEP1, ARR_AUTO1, ARR_WEP2, ARR_AUTO2, ARR_AGE, ARR_SEX, ARR_RACE, ARR_ETHN,  
ARR_RESID, ARR_DISP, ARR_DRG_TYPE )
```

Next we must tell the query where the new data is located in the data record. This is done by adding the SQL line as indicated below to the end of the SELECT statement:

```
SELECT IIf(LEFT([Field1],1)=' ',Null,LEFT([Field1],1)), IIf(MID([Field1],13,21)='          ',Null,MID([Field1],13,21)),  
IIf(MID([Field1],34,2)=' ',Null,VAL(MID([Field1],34,2))), IIf(Mid([Field1],48,4)=' ',Null,VAL(MID([Field1],48,4))),  
IIf(MID([Field1],52,2)=' ',NULL,VAL(MID([Field1],52,2))), IIf(MID([Field1],54,2)=' ',NULL,VAL(MID([Field1],54,2))),  
IIf(Mid([Field1],56,1)=' ',Null,MID([Field1],56,1)), IIf(Mid([Field1],57,1)=' ',Null,MID([Field1],57,1)), IIf(MID([Field1],58,3)='  
' ,Null,MID([Field1],58,3)), IIf(MID([Field1],61,2)=' ',Null,MID([Field1],61,2)), IIf(Mid([Field1],63,1)=' ',Null,MID([Field1],63,1)),  
IIf(MID([Field1],64,2)=' ',Null,MID([Field1],64,2)), IIf(Mid([Field1],66,1)=' ',Null,MID([Field1],66,1)), IIf(Mid([Field1],67,2)='  
' ,Null,VAL(MID([Field1],67,2))), IIf(Mid([Field1],71,1)=' ',Null,MID([Field1],71,1)), IIf(Mid([Field1],72,1)=' ',Null,MID([Field1],72,1)),  
IIf(Mid([Field1],73,1)=' ',Null,MID([Field1],73,1)), IIf(Mid([Field1],74,1)=' ',Null,MID([Field1],74,1)), IIf(Mid([Field1],75,1)='  
' ,Null,MID([Field1],75,1)), IIf(Mid([Field1],76,2)=' ',Null,MID([Field1],76,2))  
FROM SEG06;
```

The SQL that was added; ,IIf(Mid([Field1],76,2)=' ',Null,MID([Field1],76,2)); tells Access that 76 is the beginning position of data element ARR\_DRG\_TYPE and is 2 characters in length.

Close and save the query.

After all pertinent queries have been modified run macros m01 through m12 as indicated in *Reading State-Level Data into ACCESS*.