CS-426 Database Development
Professor Ricardo Jimenez

Final Project:

# Football Equipment Database

By Todd J.
Wed, 6:00pm Summer 2009

Table of Contents

# I – Database Requirements

## A. Background

The <unnamed football team> is a professional NFL sports team in South Florida. They are an NFL franchise with a training facility located in Davie, Florida. Within the training facility exists an equipment department. Considering the nature of the staff and the players, it's become extremely important for the equipment department to have the ability to keep track of everything that goes in and out of their inventory. The players and staff rely on the equipment department to ensure that they have the necessary gear at the stadium and on the practice field. The game day field and support staff also rely on the equipment department to ensure that they have the latest NFL standard issue for game time.

The equipment department currently use an outdated system written originally in Access 95. The original developer of the system was not a programmer or an analyst, but a very hands-on computer user who converted their Lotus 1-2-3 spreadsheets into a useable database system. The system was simply designed ad-hoc in order to serve an immediate need. Although already written in Microsoft Access, the table structure and layout were done without a case study. The tables lack normalization and development for the system is excruciatingly difficult due to the poor table and structure layout. The old system exists as nothing more than a series of spreadsheets imported into access with few, if any relational keys.

The scope of this project will be to provide the equipment department with a solid foundation that their department can grow and expand with. It should fill their immediate needs that aren't currently being met with their existing system. It will provide them with a good platform with which they can expand should the need arise.

## B. Business Roles – By Group

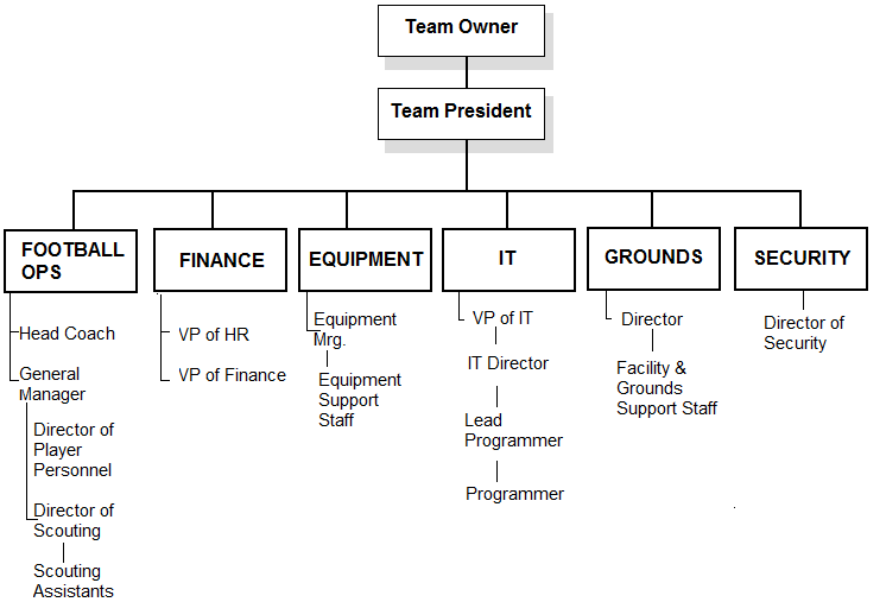
1. <u>Staff / Players:</u> The system should keep track of all current staff members. This includes players, coaches, and support staff. The system should be efficient enough to expand, if needed, the number of departments and types of staff. The system should be generic enough to handle all types of personnel data, while being specific enough to meet all the requirements needed for the player information. It would be ideal to make the system configurable enough to allow for the retainment of terminated or waived employees or players. This would allow the equipment staff to maintain the player and personnel sizes if this player or staff member should return to employment.

2. <u>Items:</u> The equipment department needs to be able to keep track of what items they purchase, the types of items they purchase, and how to properly categorize them. There needs to be a distinction between players, coaching staff, and support staff.

3. Inventory / Issuance: The equipment department needs to be able to keep track of the inventory in their department. Specifically, they need to be able to determine who has what so they can determine how many the staff or player has already been issued of that particular item. It's also imperative that they be able to provide detailed lists to operations to better facilitate their annual budget. This will provide an additional feature to them as they'll be able to determine what items they have surplus or excess of. This will reduce the department's costs in terms of donation / auction / charity items.

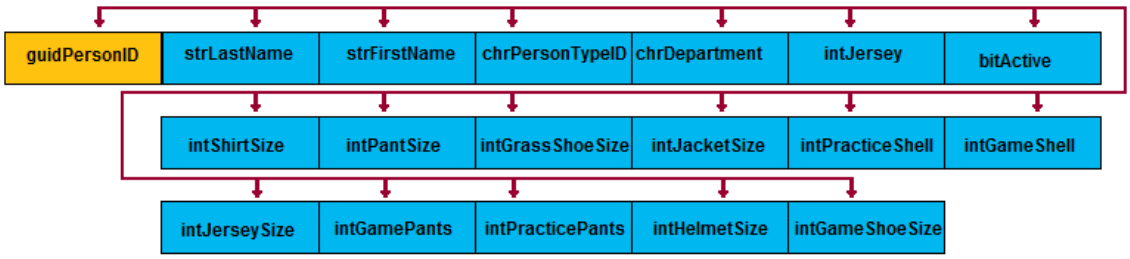
4. Vendors: Vendor relations are important to the equipment department. Purchasing your items from a common source can provide the team with discounts and savings. A benefit to storing the vendor's contact information in the system is the ability to see where the items in inventory came from.

## II – Team Organizational Chart



## III – Equipment Database and Functional Dependencies

1. **Persons Table** - No transitional or partial dependencies, 3NF
   tblPersons (guidPersonID, strLastName, strFirstName, chrPersonTypeID, chrDepartmentCodeID, intJersey, bitActive, intShirtSize, intPantSize, intGrassShoeSize, intJacketSize, intPracticeShell, intGameShell, intJerseySize, intGamePants, intPracticePants, intHelmetSize, intGameShoeSize)

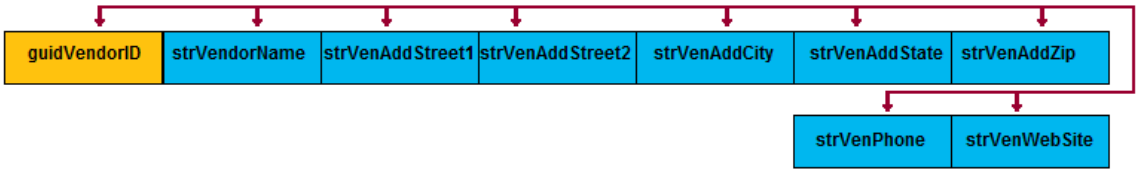



There are no transient or partial dependencies in the Persons Table. Although another table could be added which would be responsible purely for the item sizes, it would create a 1 to 1 relationship. It is therefore the most responsible thing to store this information directly to the person table as it is directly attributed to the person.

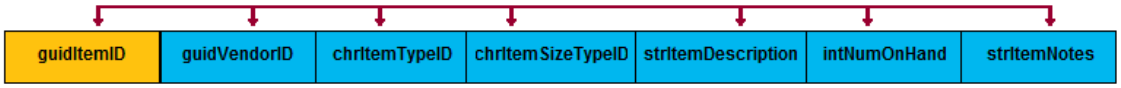
**Persons Type Table** - No transitional or partial dependencies, 3NF
tblPersonsType (chrPersonType, strPersonTypeDesc)


**Department Table** - No transitional or partial dependencies, 3NF
tblDepartment (chrDepartmentCodeID, strDepartmentDesc)

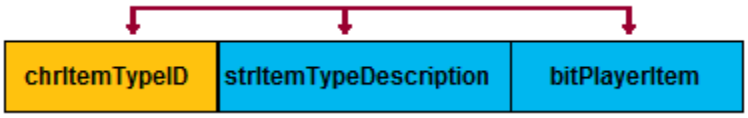
2. **Vendor Table** - No transitional or partial dependencies, 3NF
   tblVendor (guidVendorID, strVendorName, strVenAddStreet1, strVenAddStreet2, strVenAddCity, strVenAddState, strVenAddZip, strVenPhone, strVenWebSite)



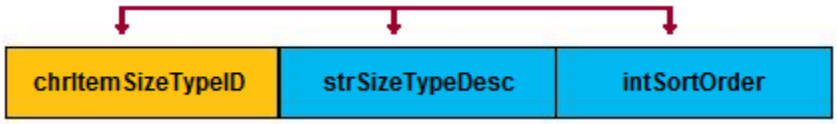

3. **Items Table** - No transitional or partial dependencies, 3NF
   tblItems (guidItemID, guidVendorID, chrItemTypeID, chrItemSizeTypeID, strItemDescription, intNumOnHand, strItemNotes)



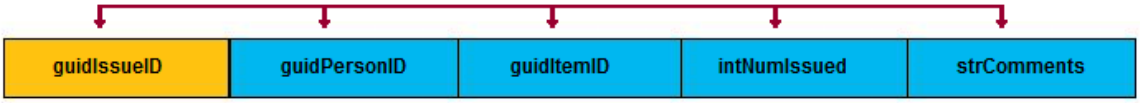

**Item Category Type** - No transitional or partial dependencies, 3NF
tblItemCategoryType (chrItemTypeID, strItemTypeDescription, bitPlayerItem)




**Item Size Type** - No transitional or partial dependencies, 3NF
tblItemSizeTypes (chrItemSizeTypeID, strSizeTypeDesc, intSortOrder)



4. **Item Issue Table** - No transitional or partial dependencies, 3NF
   tblItemIssue (guidIssueID, guidPersonID, guidItemID, intNumIssued, strComments)



## IV – Equipment Database Normalization, Dependencies and Entity Analysis

**1.** **User Identification**
**Last Name**
**First Name**
**Group Type**
**Department Name**
**Status**
**Clothing Sizes**

- User Identifications are assigned by the equipment manager. Domain / Account IDs are assigned by the Administrator.
- It is important to know which "group" the person belongs to. This would include three main categories, Player, Coach, and Staff.
- It is also important to know which departments the person falls under. Although Players and Coaches automatically fall under Football Operations, the staff departments are diverse. It's important for the equipment manager to know this as the type of equipment used or issued is dependant upon this.
- As each person has a specific set of sizes, dependant upon the equipment they use, it would likely be useful to store this information directly in the Persons table, rather than in a separate table. A 1 to 1 relation to another table would not be ideal.

**<u>Functional Dependencies Analysis:</u>**

tblPersons (<u>guidPersonID</u>, strLastName, strFirstName, chrPersonTypeID, chrDepartmentCodeID, intJersey, bitActive, intShirtSize, intPantSize, intGrassShoeSize, intJacketSize, intPracticeShell, intGameShell, intJerseySize, intGamePants, intPracticePants, intHelmetSize, intGameShoeSize)

**There is a dependency between the Persons Table, Group Type, and Department. It is therefore recommended that a separate table for GROUP and DEPARTMENT be created. The chrPersonTypeID is left as a foreign key to the tblPersonType table, and chrDepartmentCodeID is left as a foreign key to the tblDepartment table.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Person | Person ID | PersonTypeID, DepartmentID |
| Person Group | PersonTypeID | |
| Department | DepartmentID | |

**2.** **Person Group Type ID**
**Person Group Description**

- User Group Type is defined by the role or position that the employee meets within the organization. This is broken up into three main groups, PLAYERS, STAFF, COACHES.
- Person Group Description will be the full description of the group, while the ID will simply be a referential three-character key.

**Functional Dependencies Analysis:**

tblPersonsType (chrPersonType, strPersonTypeDesc)

**The Group table is referential to the Persons table. It's primary key, chrPersonTypeID is a foreign key within the Persons Table.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Person | Person ID | PersonTypeID |
| Person Group | PersonTypeID | |

**3.** **Department ID**
**Department Description**

- User Department is defined by the role or position that the employee meets within the organization. This is broken up into many types.
- Department Description will be the full description of the Department, while the ID will simply be a referential three-character key.

## Functional Dependencies Analysis:

tblDepartment (chrDepartmentCodeID, strDepartmentDesc)

**The Department table is referential to the Persons table. It's primary key, chrDepartmentCodeID is a foreign key within the Persons Table.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Person | Person ID | DepartmentID |
| Department | DepartmentID | |

**4.** **Vendor ID**
**Vendor Name**
**Vendor Address**
**Vendor Phone**
**Vendor Website**

- The Vendor is defined as the supplier of the items for the item table. It will be beneficial for the equipment department to store this data in a single repository.
- Complete vendor information should be stored. Address, City, State, and Zip, as well as contact phone, and the internet web address.

## Functional Dependencies Analysis:

tblVendor (guidVendorID, strVendorName, strVenAddStreet1, strVenAddStreet2, strVenAddCity, strVenAddState, strVenAddZip, strVenPhone, strVenWebSite)

**The Vendor table is referential to the Items table. It's primary key, guidVendorID is a foreign key within the Items Table.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Items | Item ID | Vendor ID |
| Vendor | Vendor ID | |

**5.** **Item**
**Item Type**
**Vendor Name**
**Item Size Type**
**Item Description**
**Inventory**
**Item Notes**

- The Item table is defined as the purchased inventory items. They would be stored in this table, keeping track of how many, and from where they were purchased. This would be linked by three referential tables. In total, this table would combine the type of item, the item size, the quantity, and from where it was purchased.

- Complete vendor information would be accessed by the foreign key, Vendor ID.

- Item Size Type information would be accessed by the foreign key, Item Size Type ID.

- Item Type ID information would be accessed by the foreign key, Item Type ID.

**Functional Dependencies Analysis:**

tblItems (guidItemID, guidVendorID, chrItemTypeID, chrItemSizeTypeID, strItemDescription, intNumOnHand, strItemNotes)

**The Item table has three referential tables; Vendor, Item Size Type, and Item Type. It's primary key, guidItemID is a foreign key within the Items Issued Table, which includes a list of the items that have been issued to the people defined in the Persons Table.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Items | Item ID | Vendor ID, Item Type ID, Item Size ID |
| Vendor | Vendor ID | |
| Item Type | Item Type ID | |
| Item Size | Item Size ID | |

**6.** **Item Type**
**Item Type Description**

- The Item Type table defines the total available item types. These are the categories that the items purchased will fall into. It's referential to the Items table.

**Functional Dependencies Analysis:**

tblItemCategoryType (chrItemTypeID, strItemTypeDescription, bitPlayerItem)

**The Item Category Type table is referential to the Items table. It's primary key, chrItemTypeID is a foreign key within the Items Table.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Items | Item ID | Item Type ID |
| Item Type | Item Type ID | |

**7.** **Item Size Type**
**Item Size Type Description**
**Sort Order**

- The Item Size Type table defines the total available sizes. These are the sizes that the items purchased will need to fall into. It's referential to the Items table.
- It will also include a Sort Order so that the display of the sizes will be in a respective order.

**Functional Dependencies Analysis:**

tblItemSizeTypes (chrItemSizeTypeID, strSizeTypeDesc, intSortOrder)

**The Size Item Type table is referential to the Items table. It's primary key, chrItemSizeTypeID is a foreign key within the Items Table.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Items | Item ID | Item Size Type ID |
| Item Size Type | Item Size Type ID | |

**8.** **Item Issue ID**
**Person ID**
**Item ID**
**Quantity Issued**
**Comments**

- The Items Issued table will keep track of the number of items and the type of items that were issued. In addition, it will also keep track of who the items were issued to.
- The items issued table is the top of the tree whereby all other tables branch from it. It is the cornerstone table.

**Functional Dependencies Analysis:**

tblItemIssue (guidIssueID, guidPersonID, guidItemID, intNumIssued, strComments)

**The Item Issue table is referential to the two main tables, Items and Persons. This is the key table that brings the two branches of tables together.**

| Relation | Primary Key | Foreign Key |
|---|---|---|
| Item Issue | Issue ID | Item ID, Person ID |
| Items | Item ID | |
| Persons | Person ID | |

## V – Equipment Database Conceptual Schema

### 1. TBLPERSONS – Entity

| Attribute | Type | Format / Example | Req. | PK or FK | Ref. Table | Comments |
|---|---|---|---|---|---|---|
| guidPersonID | AutoNumber | 12 | Y | PK | | Person ID |
| strLastName | Text(50) | Jaspers | Y | | | Last Name |
| strFirstName | Text(50) | Todd | N | | | First Name |
| chrPersonTypeID | Text(1) | S | Y | FK | tblPersonType | Group Type |
| chrDepartmentCodeID | Text(3) | SIT | Y | FK | tblDepartment | Department ID |
| intJersey | Number | | N | | | Jersey Number |
| bitActive | Yes/No | True | Y | | | Status |
| intShirtSize | Number | 22 | N | | | Size |
| intPantSize | Number | 36 | N | | | Size |
| intGrassShoeSize | Number | 12 | N | | | Size |
| intJacketSize | Number | 44 | N | | | Size |
| intHatSize | Number | 41 | N | | | Size |
| intPracticeShell | Number | | N | | | Size |
| intGameShell | Number | | N | | | Size |
| intJerseySize | Number | | N | | | Size |
| intGamePants | Number | | N | | | Size |
| intPracticePants | Number | | N | | | Size |
| intHelmetSize | Number | | N | | | Size |
| intGameShoeSize | Number | | N | | | Size |

### 2. TBLPERSONSTYPE – Entity

| Attribute | Type | Format / Example | Req. | PK or FK | Ref. Table | Comments |
|---|---|---|---|---|---|---|
| chrPersonTypeID | Text(1) | S | Y | PK | | Group Type ID |
| strPersonTypeDesc | Text(50) | Staff | Y | | | Group Type Desc. |

| Relationship / Entities | Connectivity | Attributes |
|---|---|---|
| PERSON TYPE → PERSONS | 1 : M | Group Type ID |

### 3. TBLDEPARTMENT – Entity

| Attribute | Type | Format / Example | Req. | PK or FK | Ref. Table | Comments |
|---|---|---|---|---|---|---|
| chrDepartmentCodeID | Text(3) | FOP | Y | PK | | Department ID |
| strDepartmentDesc | Text(50) | Football Operations | Y | | | Department Desc. |

| Relationship / Entities | Connectivity | Attributes |
|---|---|---|
| DEPARTMENT → PERSONS | 1 : M | Department ID |

## 4. TBLVENDOR – Entity

| Attribute | Type | Format / Example | Req. | PK or FK | Ref. Table | Comments |
|---|---|---|---|---|---|---|
| guidVendorID | AutoNumber | 1 | Y | PK | | Vendor ID |
| strVendorName | Text(50) | Schutt Equipment Inc. | Y | | | Vendor Name |
| strVenAddStreet1 | Text(50) | 123 Chain Bridge Rd. | Y | | | Street 1 |
| strVenAddStreet2 | Text(50) | | N | | | Street 2 |
| strVenAddCity | Text(50) | Richmond | Y | | | City |
| strVenAddState | Text(2) | VA | Y | | | State |
| strVenAddZip | Text(50)-Case | 22180 | Y | | | Zip |
| strVenPhone | Text(50)-Case | 804-741-5768 | Y | | | Phone |
| strVenWebSite | Text(50) | http://www.Schutt.com | N | | | Web Address |

| Relationship / Entities | Connectivity | Attributes |
|---|---|---|
| VENDOR → ITEMS | 1 : M | Vendor ID |

## 5. TBLITEMS – Entity

| Attribute | Type | Format / Example | Req. | PK or FK | Ref. Table | Comments |
|---|---|---|---|---|---|---|
| guidItemID | AutoNumber | 1 | Y | PK | | Item ID |
| guidVendorID | Number | 12 | Y | FK | tblVendor | Vendor ID |
| chrItemTypeID | Text(3) | PPS | Y | FK | tblItemCategoryType | Category Typ |
| chrItemSizeTypeID | Text(4) | SM | Y | FK | tblItemSizeTypeID | Item Size |
| strItemDescription | Text(50) | Schutt Practice Shells | Y | | | Description |
| intNumOnHand | Number | 52 | Y | | | Quantity |
| strItemNotes | Text(255) | New Shipment | N | | | Notes |

## 6. TBLITEMSIZETYPES – Entity

| Attribute | Type | Format / Example | Req. | PK or FK | Ref. Table | Comments |
|---|---|---|---|---|---|---|
| chrItemSizeTypeID | Text(4) | SM | Y | PK | | Item Size ID |
| strSizeTypeDesc | Text(50) | Small | Y | | | Description |
| intSortOrder | Number | 0 | Y | | | Sort Order |

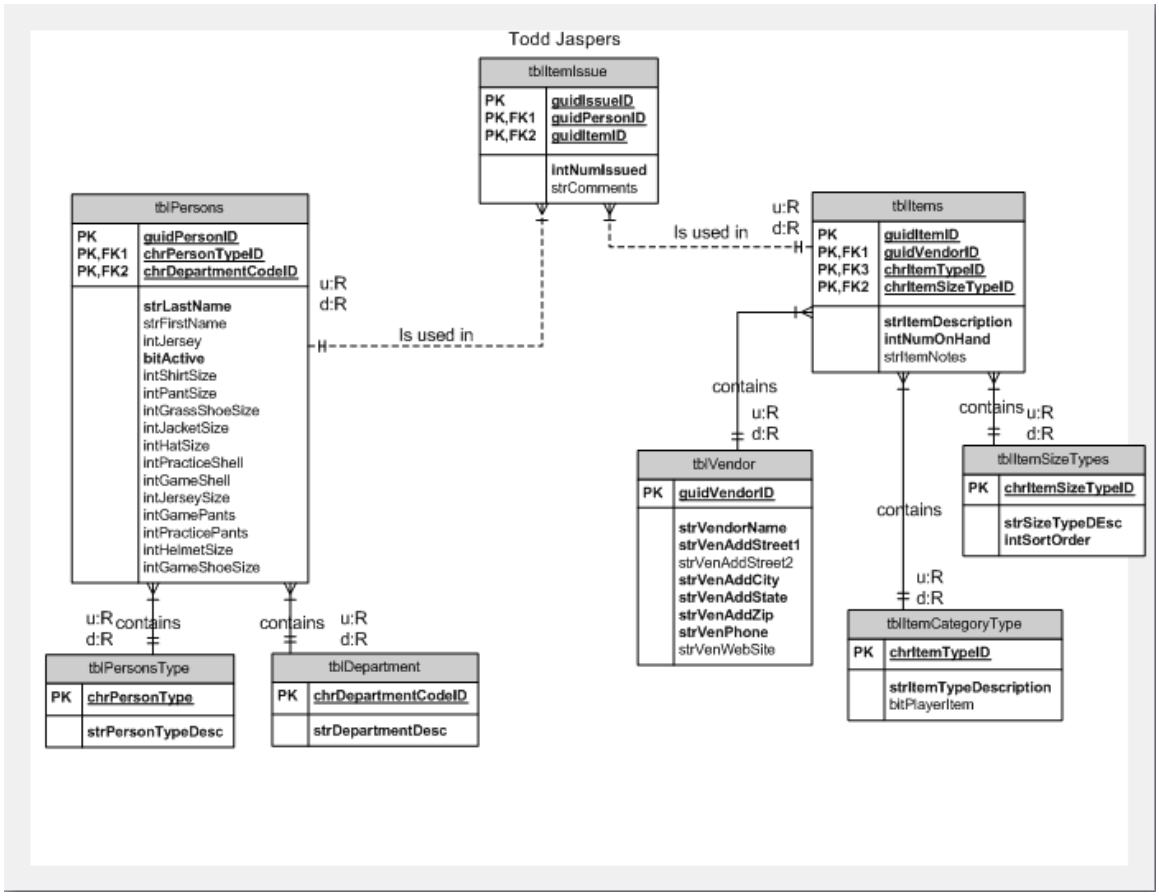
| Relationship / Entities | Connectivity | Attributes |
|---|---|---|
| SIZE TYPES → ITEMS | 1 : M | Item Size Type ID |

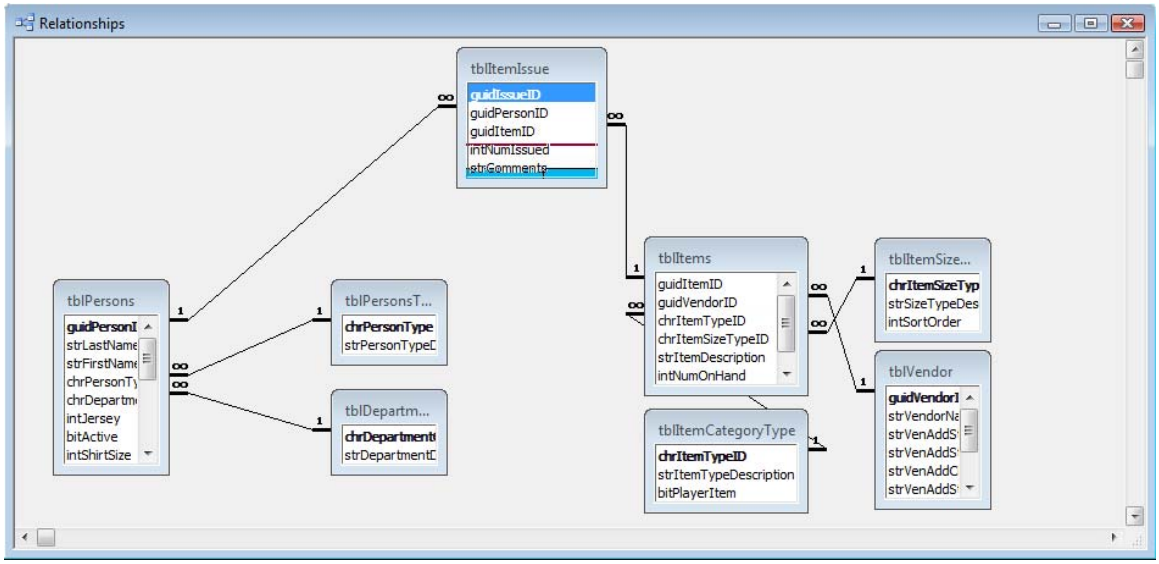
## 7. TBLITEMCATEGORYTYPE - Entity

| Attribute | Type | Format / Example | Req. | PK or FK | Ref. Table | Comments |
|---|---|---|---|---|---|---|
| chrItemTypeID | Text(3) | PPS | Y | PK | | Item Type ID |
| strItemTypeDescription | Text(50) | Player Practice Shells | Y | | | Description |
| bitPlayerItem | Yes/No | True | Y | | | Boolean |

| Relationship / Entities | Connectivity | Attributes |
|---|---|---|
| ITEM TYPES → ITEMS | 1 : M | Item Type ID |

## VI – Equipment Entity Relational Diagram

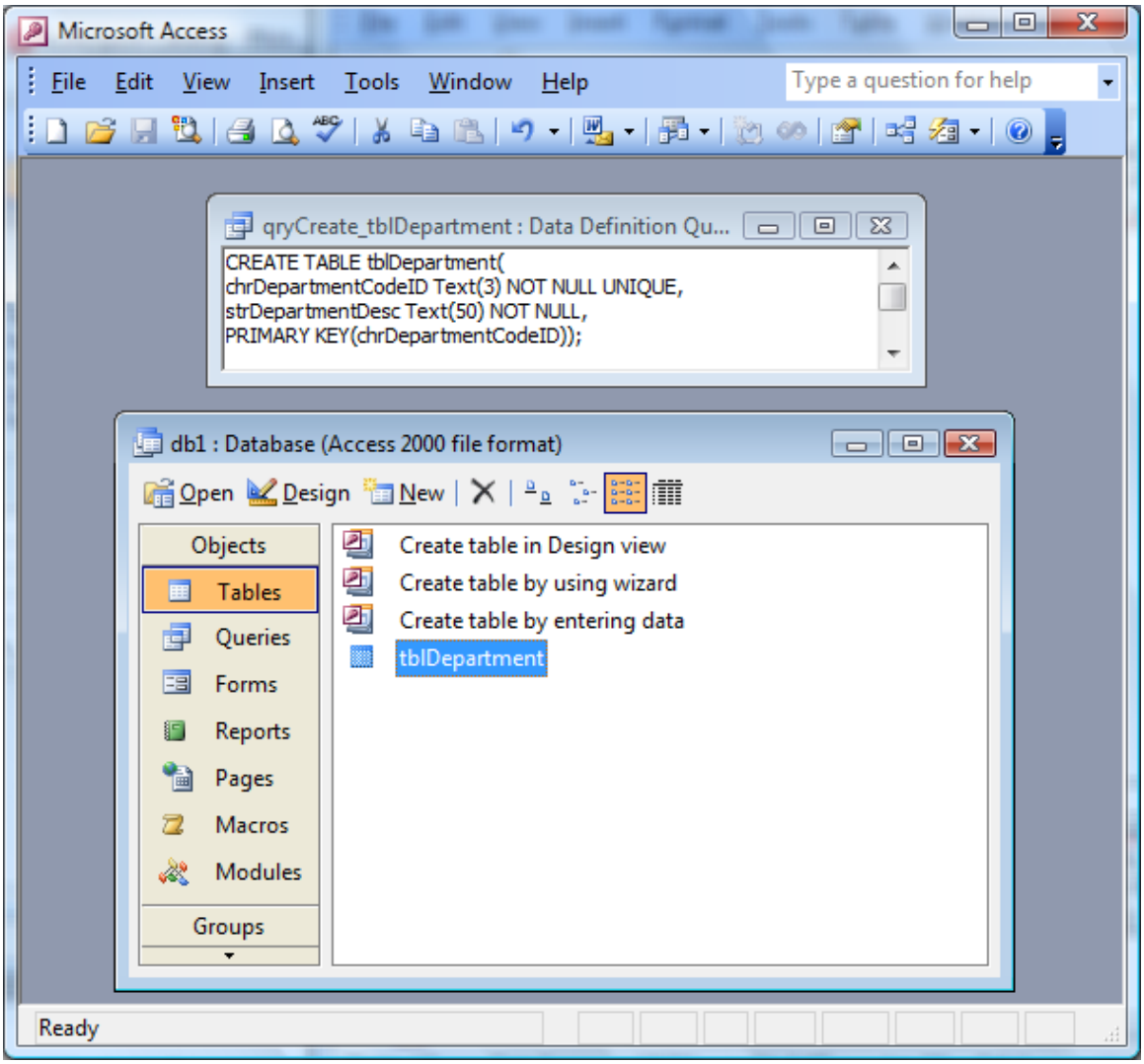


## VII – Implementation of Relational Schema – Creation of Tables by Using Access SQL Data Definition Queries.

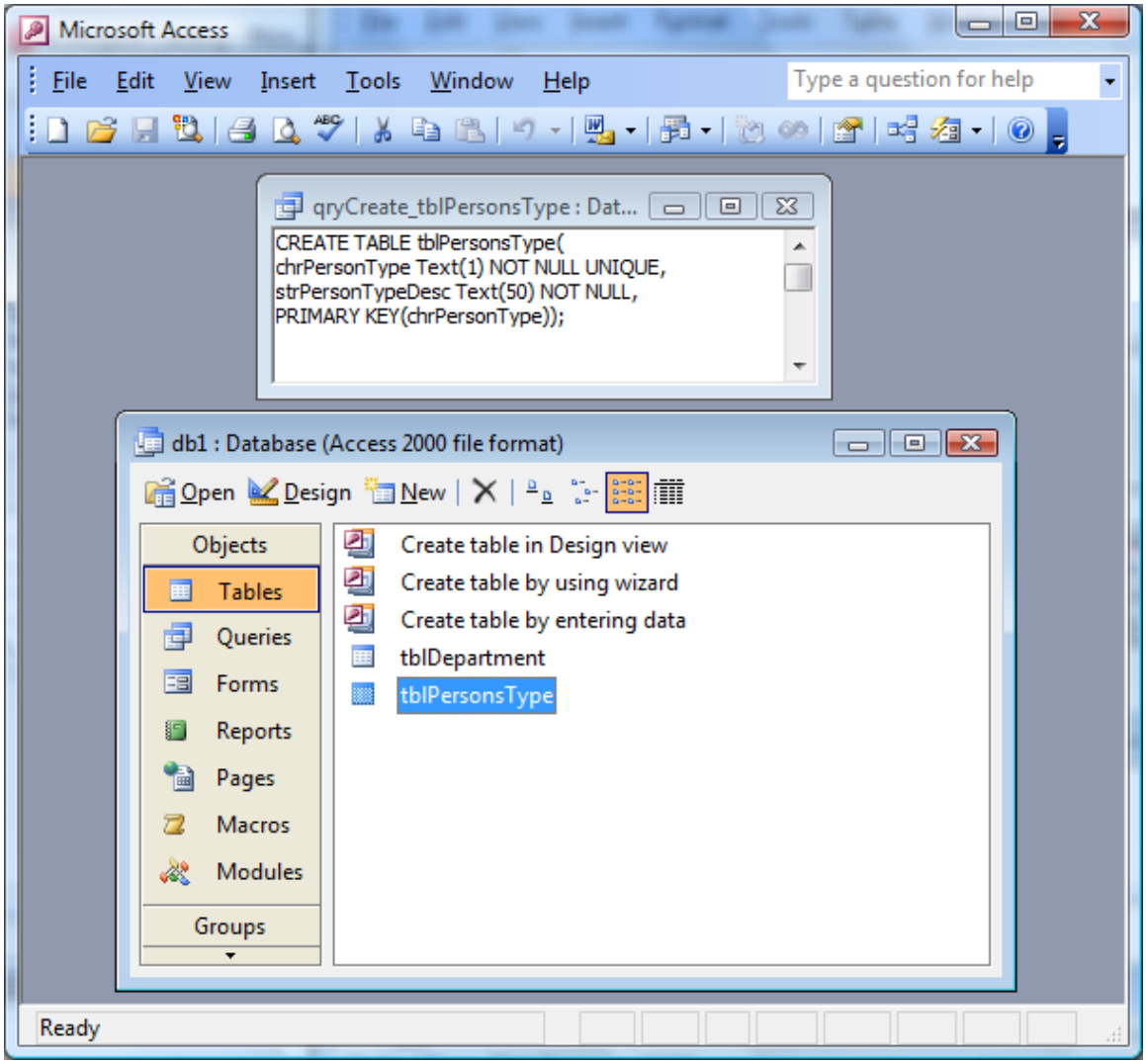
1) **tblDepartment** TABLE

| DDL Query Name: | qryCreate_tblDepartment |
|---|---|
| | CREATE TABLE tblDepartment(<br>chrDepartmentCodeID Text(3) NOT NULL UNIQUE,<br>strDepartmentDesc Text(50) NOT NULL,<br>PRIMARY KEY(chrDepartmentCodeID)); |



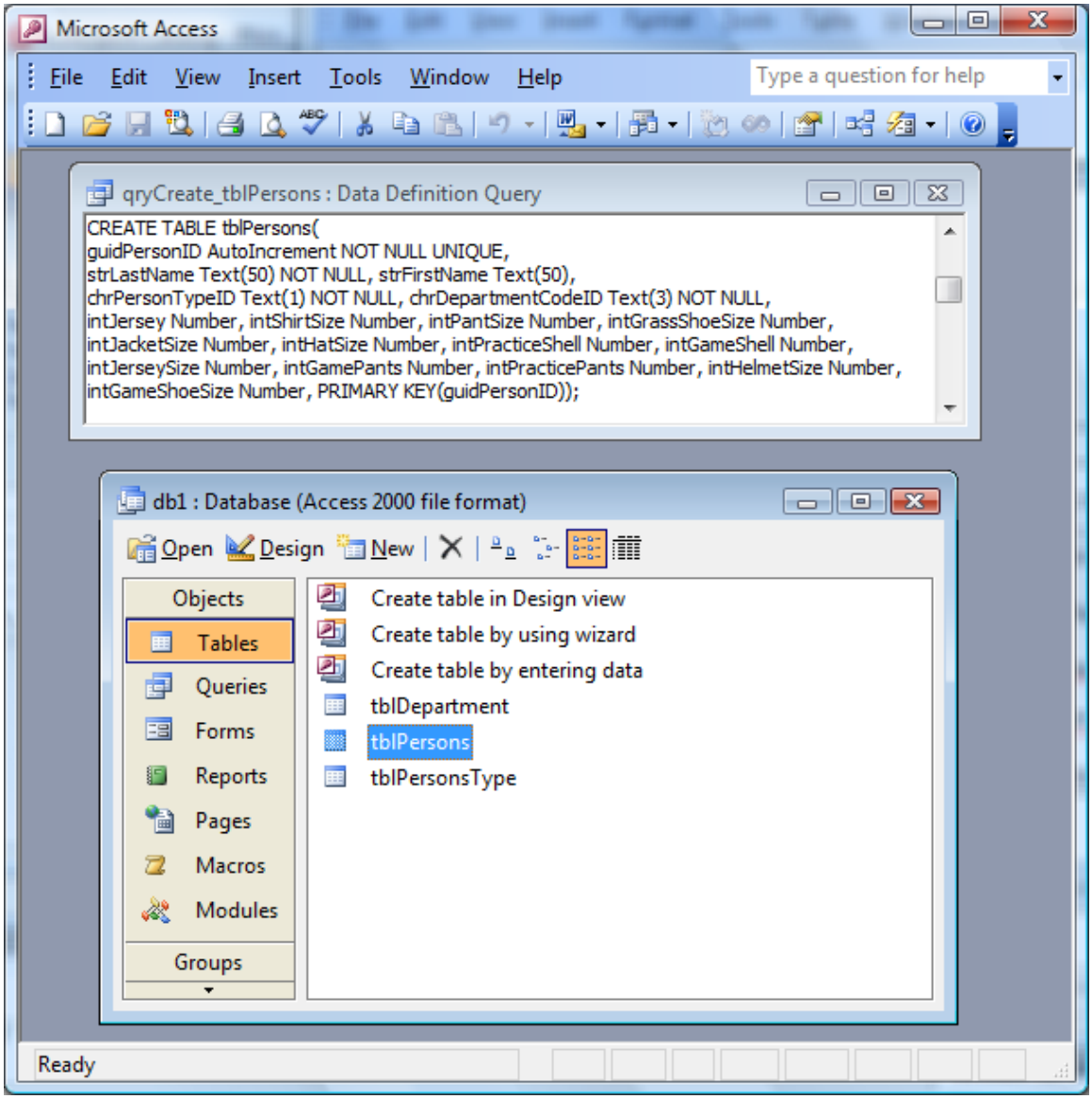
2) **tblPersonsType** TABLE

| DDL Query Name: | qryCreate_tblPersonsType |
|---|---|
| | CREATE TABLE tblPersonsType(<br>chrPersonType Text(1) NOT NULL UNIQUE,<br>strPersonTypeDesc Text(50) NOT NULL,<br>PRIMARY KEY(chrPersonType)); |



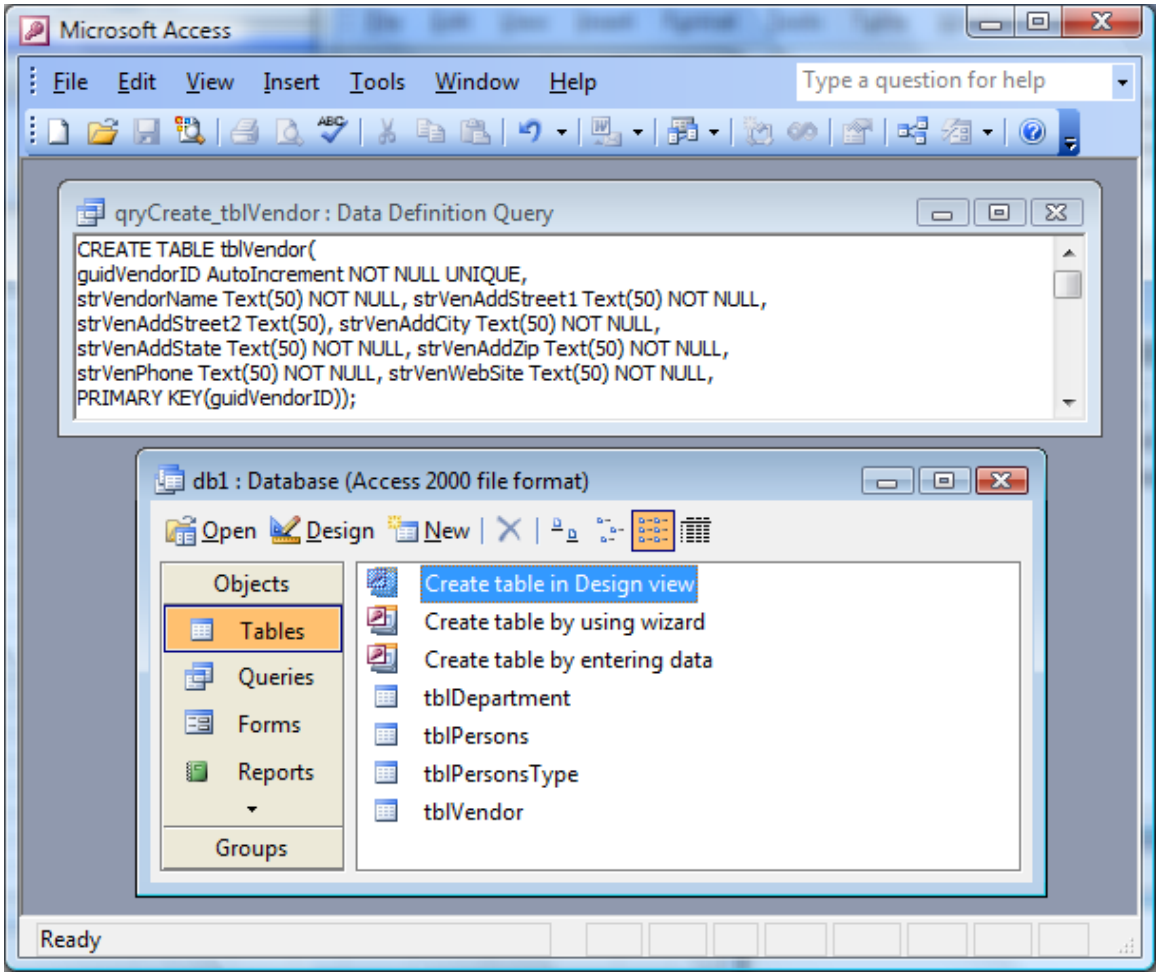
3) **tblPersons** TABLE

| DDL Query Name: | qryCreate_tblPersons |
|---|---|
| | CREATE TABLE tblPersons(<br>guidPersonID AutoIncrement NOT NULL UNIQUE,<br>strLastName Text(50) NOT NULL, strFirstName Text(50),<br>chrPersonTypeID Text(1) NOT NULL, chrDepartmentCodeID Text(3) NOT NULL,<br>intJersey Number, intShirtSize Number, intPantSize Number, intGrassShoeSize Number,<br>intJacketSize Number, intHatSize Number, intPracticeShell Number, intGameShell Number,<br>intJerseySize Number, intGamePants Number, intPracticePants Number, intHelmetSize Number,<br>intGameShoeSize Number, PRIMARY KEY(guidPersonID),<br>FOREIGN KEY (chrPersonTypeID) REFERENCES tblPersonsType,<br>FOREIGN KEY (chrDepartmentCodeID) REFERENCES tblDepartment); |



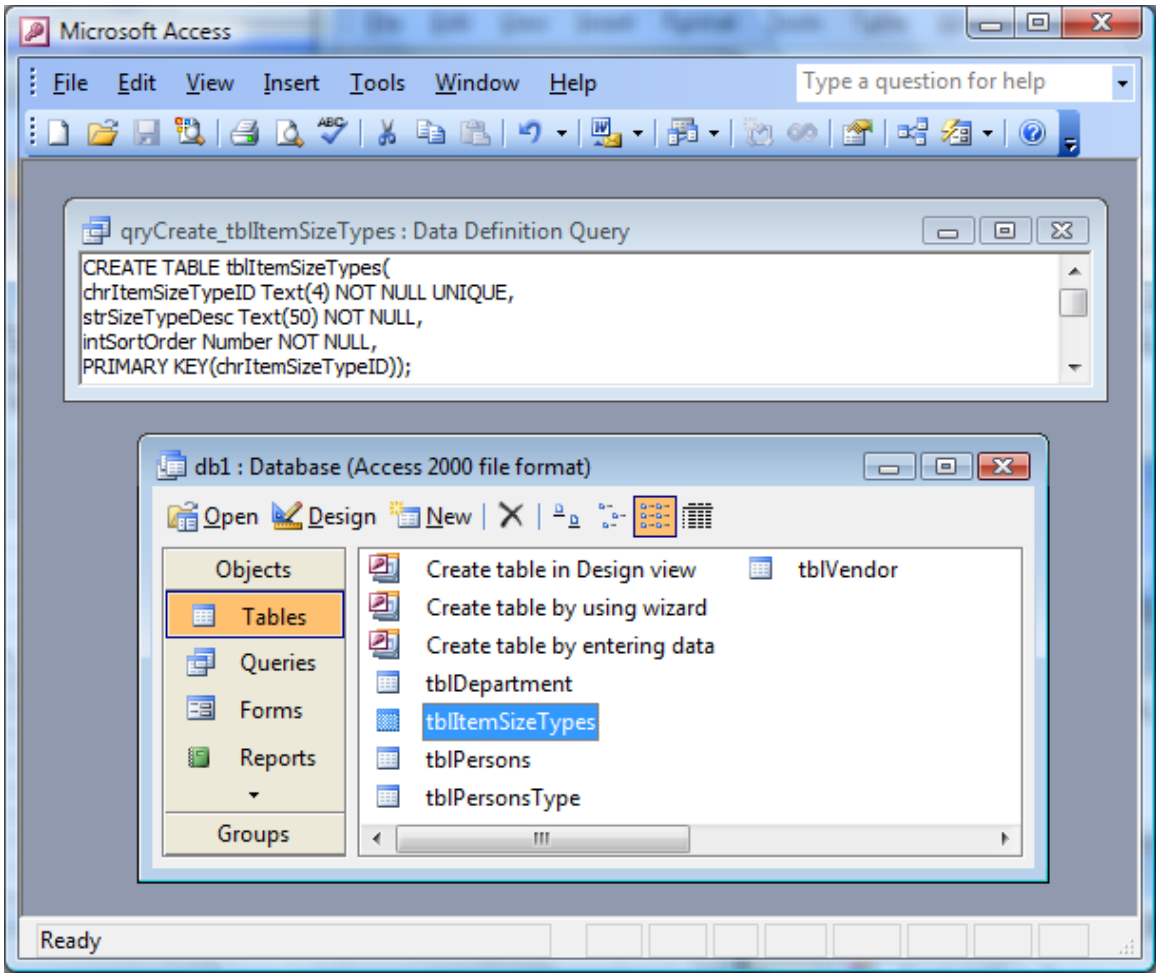
4) **tblVendor** TABLE

| DDL Query Name: | qryCreate_tblVendor |
|---|---|
| | CREATE TABLE tblVendor(<br>guidVendorID AutoIncrement NOT NULL UNIQUE,<br>strVendorName Text(50) NOT NULL, strVenAddStreet1 Text(50) NOT NULL,<br>strVenAddStreet2 Text(50), strVenAddCity Text(50) NOT NULL,<br>strVenAddState Text(50) NOT NULL, strVenAddZip Text(50) NOT NULL,<br>strVenPhone Text(50) NOT NULL, strVenWebSite Text(50) NOT NULL,<br>PRIMARY KEY(guidVendorID)); |



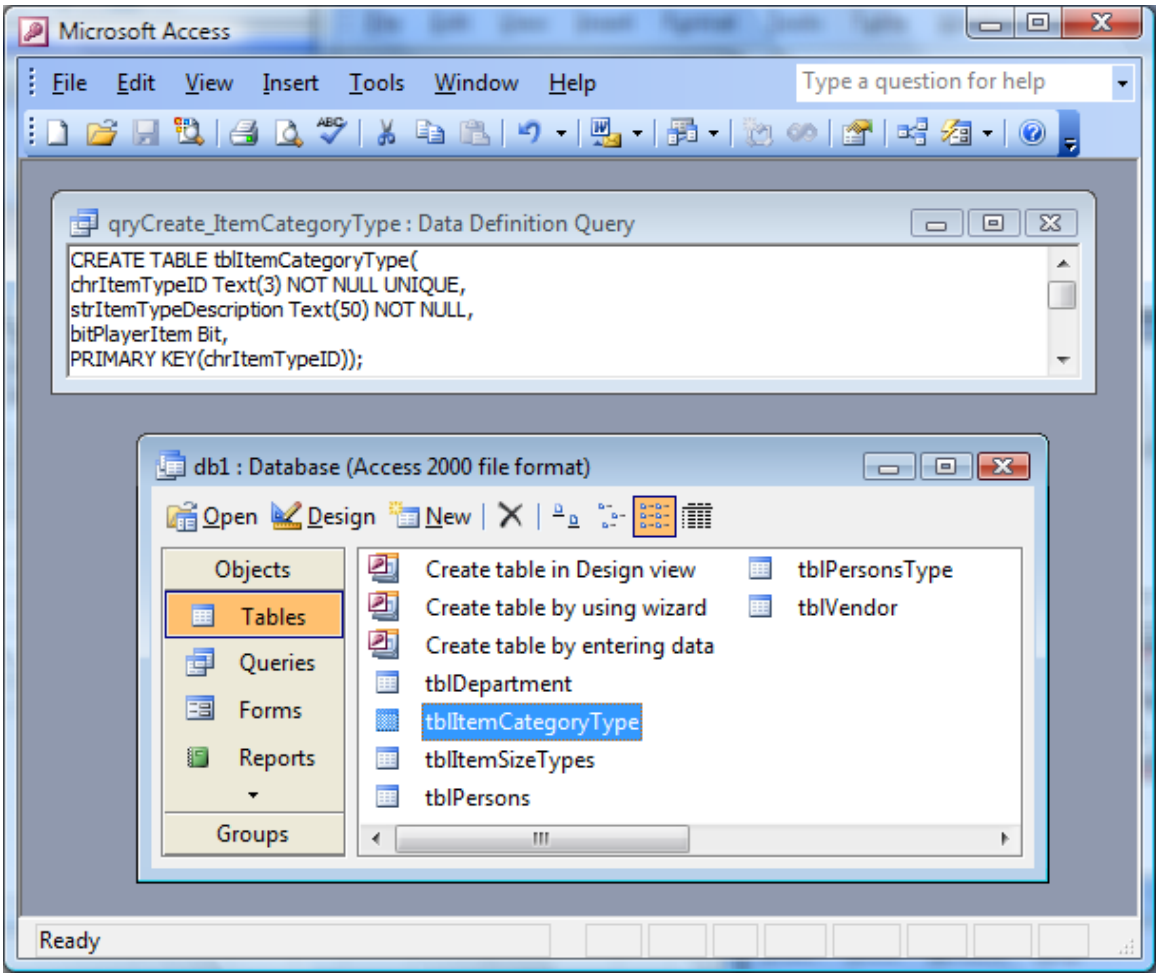
5) **tblItemSizeTypes** TABLE

| DDL Query Name: | qryCreate_tblItemSizeTypes |
|---|---|
| | CREATE TABLE tblItemSizeTypes(<br>chrItemSizeTypeID Text(4) NOT NULL UNIQUE,<br>strSizeTypeDesc Text(50) NOT NULL,<br>intSortOrder Number NOT NULL,<br>PRIMARY KEY(chrItemSizeTypeID)); |



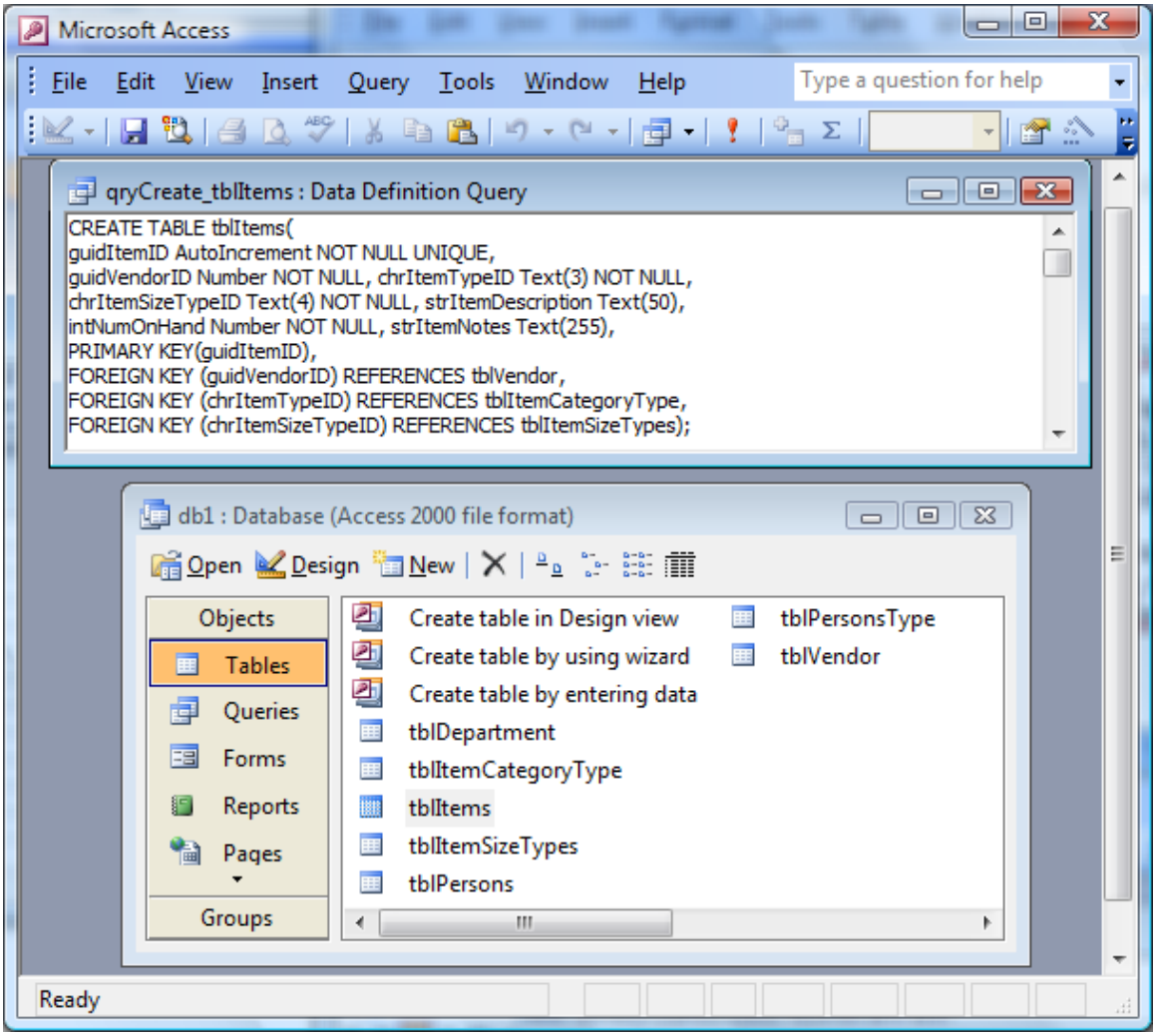
6) **tblItemCategoryType** TABLE

| DDL Query Name: | qryCreate_tblItemCategoryType |
|---|---|
| | CREATE TABLE tblItemCategoryType(<br>chrItemTypeID Text(3) NOT NULL UNIQUE,<br>strItemTypeDescription Text(50) NOT NULL,<br>bitPlayerItem Bit,<br>PRIMARY KEY(chrItemTypeID)); |



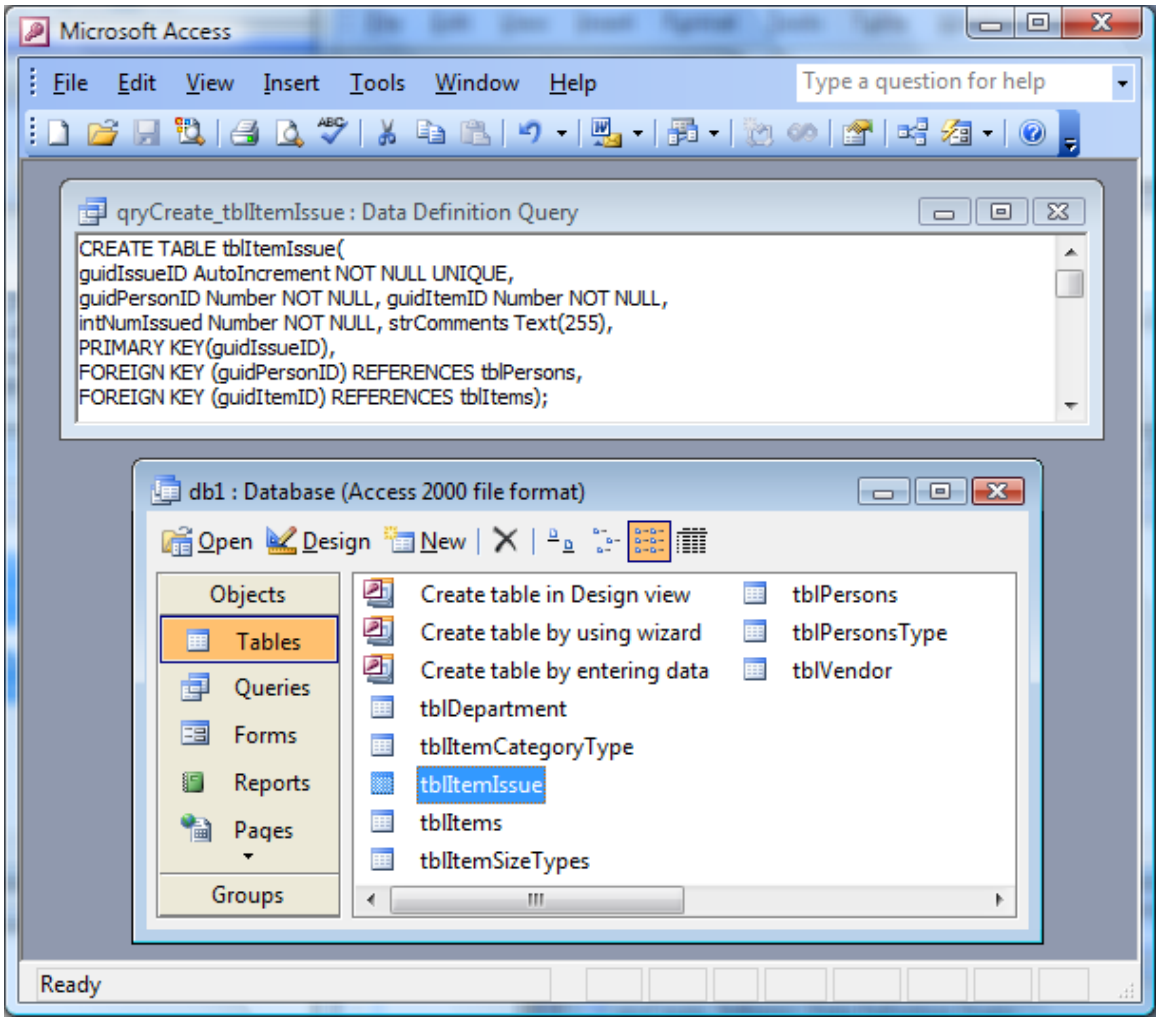
7) **tblItems** TABLE

| DDL Query Name: | qryCreate_tblItems |
|---|---|
| | CREATE TABLE tblItems(<br>guidItemID AutoIncrement NOT NULL UNIQUE,<br>guidVendorID Number NOT NULL, chrItemTypeID Text(3) NOT NULL,<br>chrItemSizeTypeID Text(4) NOT NULL, strItemDescription Text(50),<br>intNumOnHand Number NOT NULL, strItemNotes Text(255),<br>PRIMARY KEY(guidItemID),<br>FOREIGN KEY (guidVendorID) REFERENCES tblVendor,<br>FOREIGN KEY (chrItemTypeID) REFERENCES tblItemCategoryType,<br>FOREIGN KEY (chrItemSizeTypeID) REFERENCES tblItemSizeTypes); |



8) **tblItemIssue** TABLE

| DDL Query Name: | qryCreate_tblItemIssue |
|---|---|
| | CREATE TABLE tblItemIssue(<br>guidIssueID AutoIncrement NOT NULL UNIQUE,<br>guidPersonID Number NOT NULL, guidItemID Number NOT NULL,<br>intNumIssued Number NOT NULL, strComments Text(255),<br>PRIMARY KEY(guidIssueID),<br>FOREIGN KEY (guidPersonID) REFERENCES tblPersons,<br>FOREIGN KEY (guidItemID) REFERENCES tblItems); |

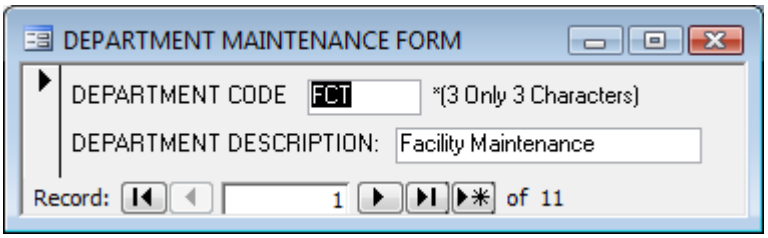


## VIII – Population of Tables / Form Creation

In order to properly load the tables with usable data, forms were created to facilitate the proper data entry. This is more desirable than importing excel spreadsheets since the data can be properly authorized when input through the form. In some cases, because of the complexity of the Persons table layout, the form necessitated additional programming using Visual Basic. Examples of this will be shown below where appropriate.

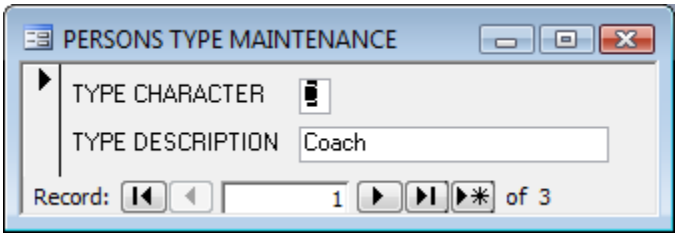
**1)** tblDepartment TABLE

A very simple form was all that was needed for the DEPARTMENT table. This table is used far less frequently. As such, it has been labeled the "Department Maintenance Form". The data in this form is unlikely to change after the initial import. Still, for ease of use, this form was created to facilitate growth.




**2)** tblPersonsType TABLE

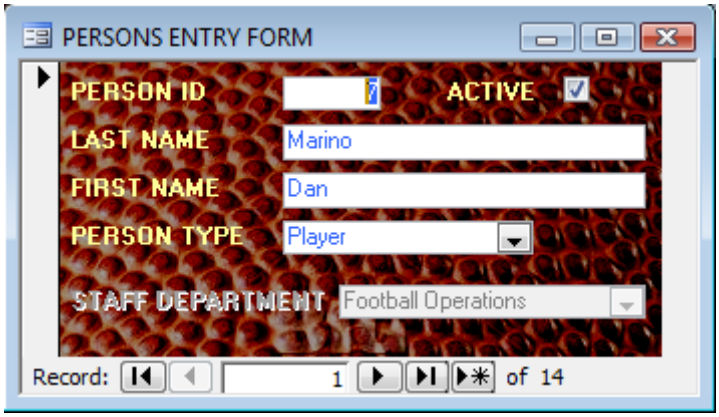
As with the Department table, the PersonsType table will also unlikely be used after the initial loading of the data. However, due to the unique and specific nature of a football team, it's imperitive that this application have the diversity to be able to conform to changes.



**3)** tblPersons TABLE

The Persons table is unique in that it holds a different purpose depending upon the user that's operating the system. Because of this, TWO forms were created to facilitate the proper data entry into the system.

The **FIRST** form is the PERSONS ENTRY FORM. This form is used exclusively for creating the person, and assigning the respective department. It's worth noting that under current NFL operational schemas, Coaches and Players all exist under a department of "Football Operations". In the past, it was possible for coaches to be listed under departments other than Football Operations. This is no longer the case for most franchises. With those requirements in place, it was necessary to modify the form programming to automatically assign "Football Operations" to the Coaches and Players.

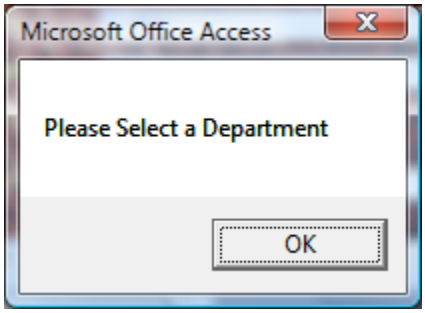



The "Staff Department" field and label are disabled unless a Person Type of "Staff" is selected. If a Person Type of Coach or Player is selected, the department type of "Football Operations" is automatically assigned.

Below is the programming for the On-Click Event for the PersonsType ComboBox:

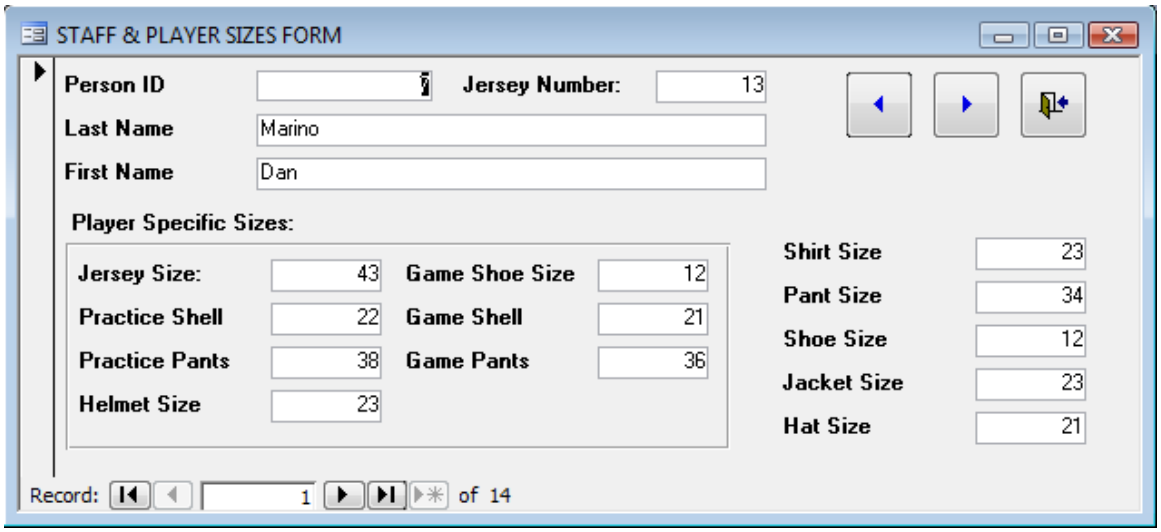
```
Private Sub Combo10_Change()
     If (Combo10.Text = "Staff") Then
         cBxDepartment.Enabled = True
         MsgBox ("Please Select a
Department")
     Else
         cBxDepartment.Enabled = False
         cBxDepartment.Value = "FOP"
     End If
End Sub
```

When “Staff” is selected, the user is prompted with this notification box:




At this point, the user is granted access to the department field and may select the appropriate department.

The **SECOND** form related to the tblPersons table is specifically for entering player / coach / staff size information. This data is not immediately used for anything other than as a reference for the equipment department. With the exception of items like shirts and pants, most football equipment is ordered to a generic size, and the “fitted” to the player based on their specific measurements. This form will be used in entering the data for the player sizes. It’s worth noting that the ONLY persons who will display on this screen are pre-existing staff / players that were created in the aforementioned form. A user cannot create a new person from within this form as it will not allow it. Take note of the additional navigational buttons in the upper right-hand corner.

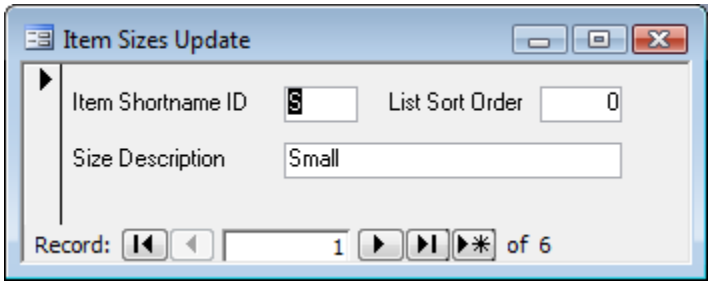


**4)** tblVendor TABLE

The form tblVendor is used to store all the information related to the product vendors. This information is crucial to have stored in the database as it makes it easily accessible when ordering new equipment or submitting shells / equipment for repair. Since this table will be frequently modified in the beginning and likely the end of the season, it's more prominently displayed and has been dressed up.


**5)** tblItemSizeTypes TABLE

As with some of the earlier forms, the Item Size Types are unlikely to change, if at all. Although at the risk of drafting a historically massive linebacker that's completely off the charts, it might still be necessary to update these tables. Take note that the "Sort Order" exists so that the equipment manager can define how he wants the sizes to be ordered on the forms (or report). This gives him / her the utmost flexibility.

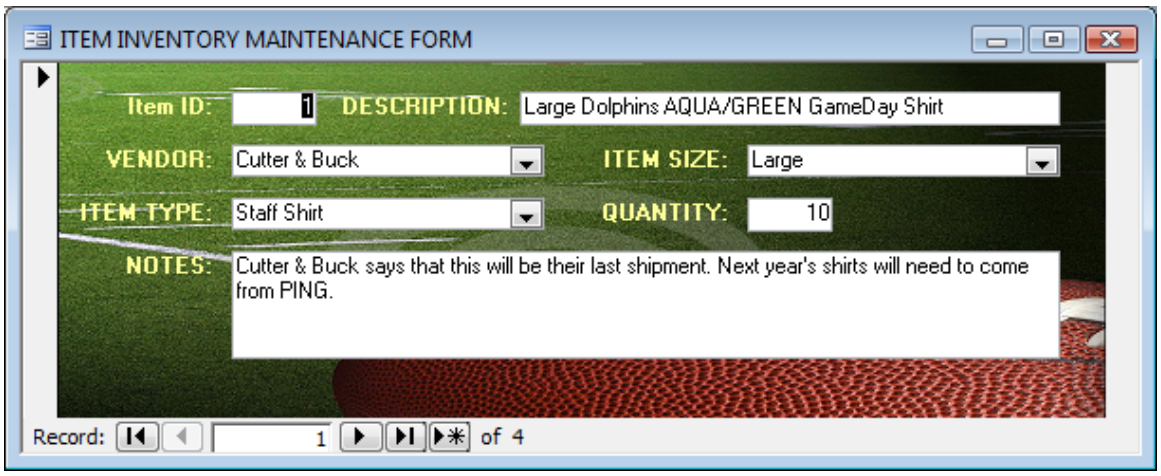


**6)** tblItemCategoryType TABLE

The Item Category Type table, while infrequently changed, will likely need to be used more often than some of the more static tables. Therefore it was necessary to create a more visible form for updating the item type categories. The form was dressed up slightly as it will be used more frequently than some of the other less frequently used forms.

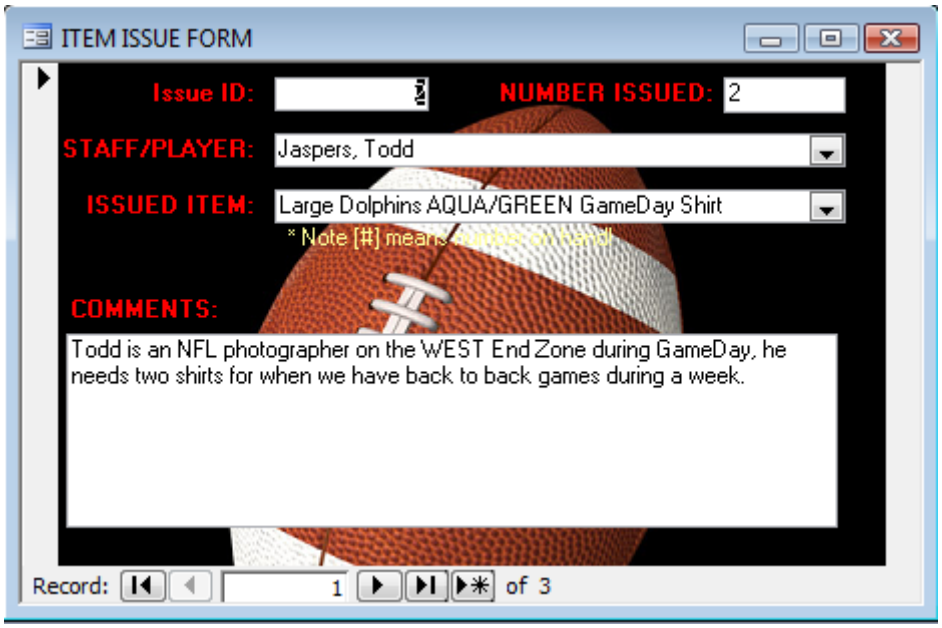

**7)** tblItems TABLE

The Items table is one of, if not the most commonly used table. It therefore would necessitate a form that is frequently used as this is where the equipment manager would spend the majority of his / her time. This form is rather diverse, but uncomplicated. The Vendor, Item Type, and Item Size pull from their respective sources and the ID KEY is stored in the tblItems table in their respective fields.



**8)** tblItemIssue TABLE

The Item Issue Table will be the most commonly used table throughout the entire system. This is the form that will allow the equipment manager to issue items to the staff and the players. This form needs to be diverse and well organized.




The pull-down for the item is a multi-layered column. It provides the equipment manager a display which shows the full description as well as the number "on hand".

| ITEM DESCRIPTION | ON HAND |
|---|---|
| Large Dolphins AQUA/GREEN GameDay Shirt | [10] |
| Nike Cleats | [52] |
| Small Dolphins AQUA/GREEN GameDay Shirt | [6] |
| XL Dolphins AQUA/GREEN GameDay Shirt | [4] |

This was performed with the following SQL Code in the ComboBox's DATA field / Control Field.

```
SELECT tblItems.guidItemID,
tblItems.strItemDescription AS [ITEM DESCRIPTION],
"["+Cstr(tblItems.intNumOnHand)+"]" AS [ON HAND] FROM
tblItems ORDER BY tblItems.strItemDescription;
```

Note that "Convert To String / Cstr" function had to be used, as well as the Concatenate Feature.

## IX – Queries, Reports & Quick View

To help facilitate the use of the system, queries were created not only for report usage, but also for the user to make changes. This section will be broken up into TWO sections, REPORTS and QUICK VIEW.

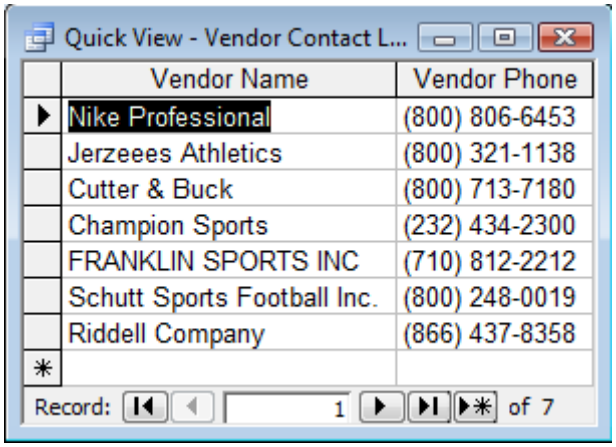
QUICK VIEW:

**QuickView – Vendor Contact List**

SQL –

```
SELECT T1.strVendorName AS [Vendor Name], T1.strVenPhone AS
[Vendor Phone]
FROM tblVendor AS T1;
```

POP-UP –




| Vendor Name | Vendor Phone |
|---|---|
| Nike Professional | (800) 806-6453 |
| Jerzeees Athletics | (800) 321-1138 |
| Cutter & Buck | (800) 713-7180 |
| Champion Sports | (232) 434-2300 |
| FRANKLIN SPORTS INC | (710) 812-2212 |
| Schutt Sports Football Inc. | (800) 248-0019 |
| Riddell Company | (866) 437-8358 |

**QuickView –Currently Issued Items (Player, Coach, Staff are identical)**

SQL –

```
SELECT tblPersons.strLastName + ", " + tblPersons.strFirstName as
Name, tblItemIssue.intNumIssued as [# Issued],
tblItemCategoryType.strItemTypeDescription as [Item Type],
tblItemSizeTypes.strSizeTypeDesc as [Size Desc],
tblItems.strItemDescription as Description
FROM tblPersons INNER JOIN ((tblItemSizeTypes INNER JOIN
(tblItemCategoryType INNER JOIN tblItems ON
tblItemCategoryType.chrItemTypeID = tblItems.chrItemTypeID) ON
tblItemSizeTypes.chrItemSizeTypeID = tblItems.chrItemSizeTypeID)
INNER JOIN tblItemIssue ON tblItems.guidItemID =
tblItemIssue.guidItemID) ON tblPersons.guidPersonID =
tblItemIssue.guidPersonID
WHERE (((tblPersons.chrPersonTypeID)="S"));
```

POP-UP –

Quick View Coaches - Currently Issued Items : Select Query

| Name | # Issued | Item Type | Size Desc | Description |
|---|---|---|---|---|
| Jaspers, Todd | 2 | Staff Shirt | Large | Large Dolphins AQUA/GREEN GameDay Shirt |
| Abella, Lilliana | 1 | Staff Shirt | Large | Large Dolphins AQUA/GREEN GameDay Shirt |
| Farendorf, Jessica | 1 | Staff Shirt | Small | Small Dolphins AQUA/GREEN GameDay Shirt |

Record: 1 of 3

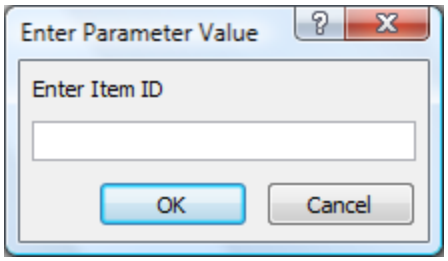
## Quick Delete - Delete Item by Item ID

This form allows the equipment manager to delete an item by it's item ID.

SQL –

```
DELETE *
FROM tblItems
WHERE [Enter Item ID] = guidItemID;
```

POP-UP –



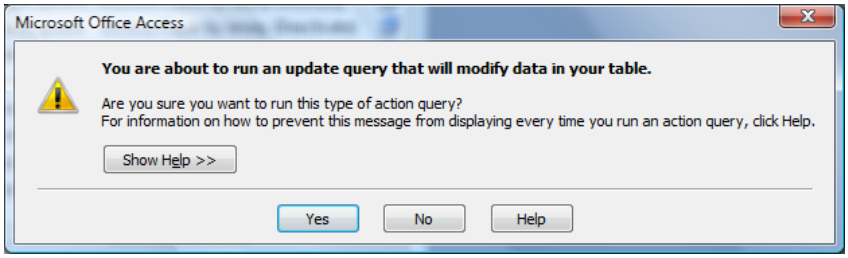

## Quick Update - Activate Player by Jersey (Activate)

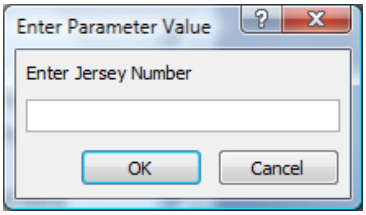
This form allows the equipment manager to re-activate a player by Jersey Number.

SQL –

```
UPDATE tblPersons SET bitActive = true
WHERE [Enter Jersey Number]=intJersey;
```

POP-UP –



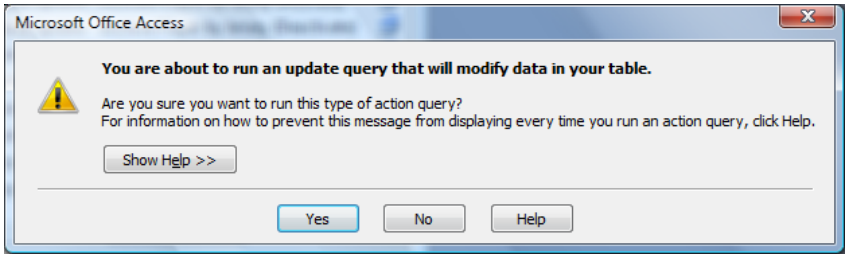
## Quick Update - Archive Player by Jersey (Deactivate)

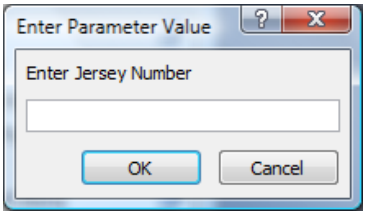
This form allows the equipment manager to archive a player by Jersey Number.

SQL –

```
UPDATE tblPersons SET bitActive = false
WHERE [Enter Jersey Number] =  intJersey;
```

POP-UP –



REPORTS:

## Report - CURRENT ISSUANCE REPORT

This report is one of the more complex reports. Although fairly simply visually, it includes a lot of the features that SQL offers. This report lists all the items in the system, sorted by type. It includes the description, the total number on hand, the total number issued, and the "avalaible" number is mathematically derived from the two. Please note this query includes: ALIASES, INNER JOINS, a LEFT JOIN, GROUPING, ORDERING, and aggregate functions like SUM.

**"Available"** Field Function **–**

```
=IIf(IsNull([Issued]),[Total],[Total]-[Issued])
```

SQL Query – **xqryReportIssuance**

```
SELECT tblItemCategoryType.strItemTypeDescription AS Type,
tblItemSizeTypes.strSizeTypeDesc AS [Size],
tblItems.strItemDescription AS Description, tblItems.intNumOnHand
AS Total, Sum(tblItemIssue.intNumIssued) AS Issued,
tblItems.intNumOnHand-Sum(tblItemIssue.intNumIssued) AS
Available, tblItemSizeTypes.intSortOrder
FROM (tblItemSizeTypes INNER JOIN (tblItemCategoryType INNER JOIN
tblItems ON tblItemCategoryType.chrItemTypeID =
tblItems.chrItemTypeID) ON tblItemSizeTypes.chrItemSizeTypeID =
tblItems.chrItemSizeTypeID) LEFT JOIN tblItemIssue ON
tblItems.guidItemID = tblItemIssue.guidItemID
GROUP BY tblItemCategoryType.strItemTypeDescription,
tblItemSizeTypes.strSizeTypeDesc, tblItems.strItemDescription,
tblItems.intNumOnHand, tblItemSizeTypes.intSortOrder
ORDER BY tblItemCategoryType.strItemTypeDescription,
tblItemSizeTypes.intSortOrder;
```

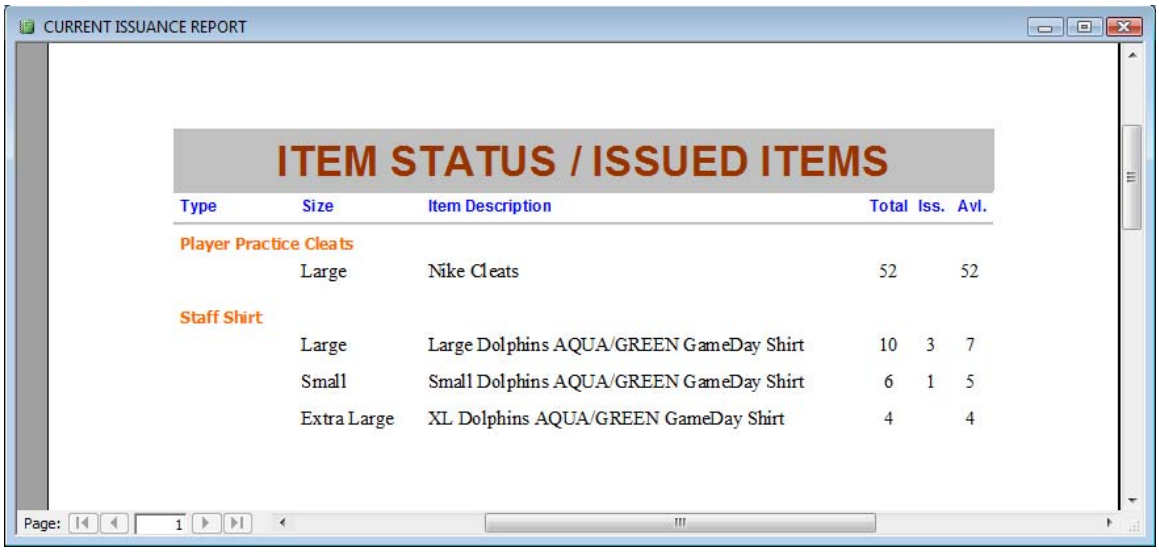
Query Output:

xqryReportIssuance : Select Query

| Type | Size | Description | Total | Issued | Available | intSortOrder |
|---|---|---|---|---|---|---|
| Player Practice Cleats | Large | Nike Cleats | 52 | | | 2 |
| Staff Shirt | Small | Small Dolphins AQU | 6 | 1 | 5 | 0 |
| Staff Shirt | Large | Large Dolphins AQU | 10 | 3 | 7 | 2 |
| Staff Shirt | Extra Large | XL Dolphins AQUA/ | 4 | | | 3 |

Record: 1 of 4

Report Output:



CURRENT ISSUANCE REPORT

ITEM STATUS / ISSUED ITEMS

| Type | Size | Item Description | Total | Iss. | Avl. |
|---|---|---|---|---|---|
| Player Practice Cleats | | | | | |
| | Large | Nike Cleats | 52 | | 52 |
| Staff Shirt | | | | | |
| | Large | Large Dolphins AQUA/GREEN GameDay Shirt | 10 | 3 | 7 |
| | Small | Small Dolphins AQUA/GREEN GameDay Shirt | 6 | 1 | 5 |
| | Extra Large | XL Dolphins AQUA/GREEN GameDay Shirt | 4 | | 4 |

Page: 1

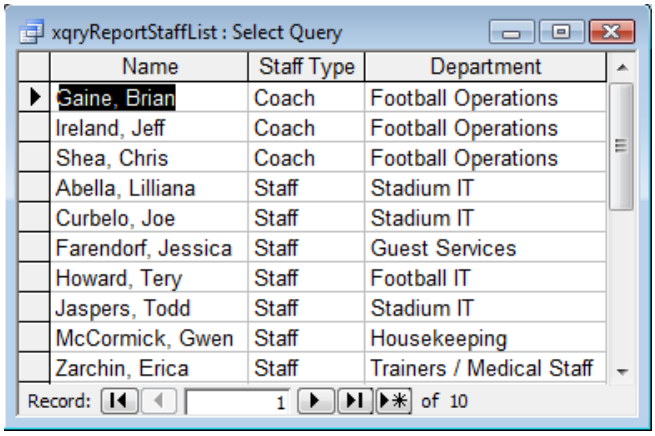
## Report - CURRENT STAFF LIST

This report is one of the less complicated reports. It simply lists all staff, organized by type (non Players). Note, this query includes Inner Joins, and conditionals including NOT EQUAL to “<>”.

SQL Query – **xqryReportStaffList**

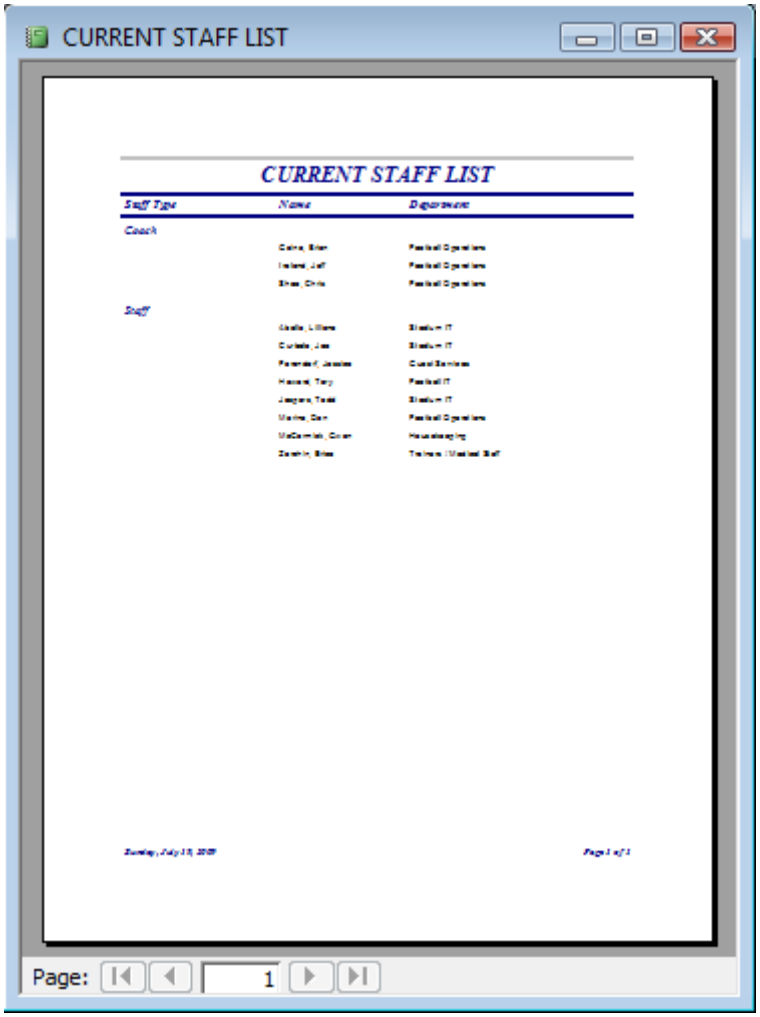
```
SELECT tblPersons.strLastName+", "+tblPersons.strFirstName AS Name,
tblPersonsType.strPersonTypeDesc AS [Staff Type],
tblDepartment.strDepartmentDesc AS Department
FROM tblPersonsType INNER JOIN (tblDepartment INNER JOIN tblPersons ON
tblDepartment.chrDepartmentCodeID=tblPersons.chrDepartmentCodeID) ON
tblPersonsType.chrPersonType=tblPersons.chrPersonTypeID
WHERE (((tblPersonsType.strPersonTypeDesc)<>"Player") AND
((tblPersons.bitActive)=True))
ORDER BY tblPersonsType.strPersonTypeDesc, tblPersons.strLastName+",
"+tblPersons.strFirstName;
```

Query Output:




| Name | Staff Type | Department |
|---|---|---|
| Gaine, Brian | Coach | Football Operations |
| Ireland, Jeff | Coach | Football Operations |
| Shea, Chris | Coach | Football Operations |
| Abella, Lilliana | Staff | Stadium IT |
| Curbelo, Joe | Staff | Stadium IT |
| Farendorf, Jessica | Staff | Guest Services |
| Howard, Tery | Staff | Football IT |
| Jaspers, Todd | Staff | Stadium IT |
| McCormick, Gwen | Staff | Housekeeping |
| Zarchin, Erica | Staff | Trainers / Medical Staff |

Report Output:



## Report – VENDOR CONTACT LIST

This report is simply a means for the equipment manager to be able to see a complete list of all of his vendors.

SQL Query – **xqryReportVendorList**

```
SELECT tblVendor.strVendorName, tblVendor.strVenAddStreet1,
tblVendor.strVenAddStreet2, tblVendor.strVenAddCity,
tblVendor.strVenAddState, tblVendor.strVenAddZip, tblVendor.strVenPhone,
tblVendor.strVenWebSite
FROM tblVendor
ORDER BY tblVendor.strVendorName;
```

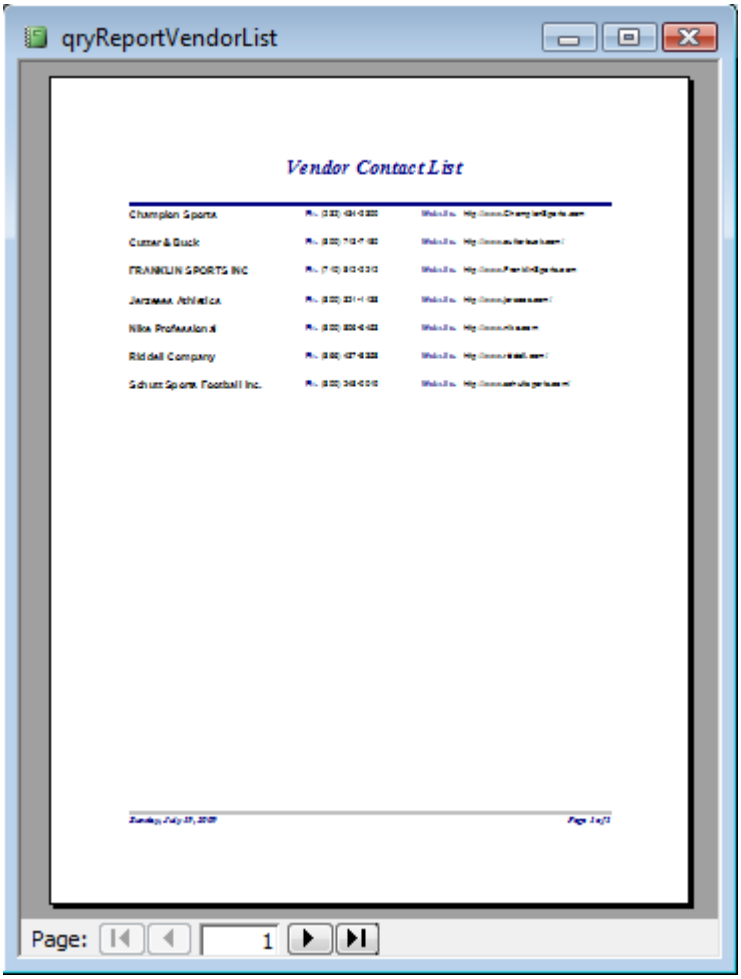
Query Output:

xqryReportVendorList : Select Query

| strVendorName | strVenAddStreet1 | strVenAddStreet2 | strVenAddCity | strVenAddState | strVenAddZip | strVenPhone | strVenWebSite |
|---|---|---|---|---|---|---|---|
| Champion Sports | 123 Chain Bridge R | | Vienna | VA | 22180- | (232) 434-2300 | http://www.Chan |
| Cutter & Buck | 701 N. 34th Street | Suite 400 | Seattle | WA | 98130- | (800) 713-7180 | http://www.cutte |
| FRANKLIN SPORTS INC | North 23rd Jefferso | | Deming | NM | 72320- | (710) 812-2212 | http://www.Frank |
| Jerzeees Athletics | 755 Lee Street | | Alexander City | AL | 35011-0012 | (800) 321-1138 | http://www.jerze |
| Nike Professional | One Bowerman Dri | | Beaverton | OR | 97005- | (800) 806-6453 | http://www.nike. |
| Riddell Company | 9801 W. Higgins R | Suite 800 | Rosemont | IL | 60018- | (866) 437-8358 | http://www.riddel |
| Schutt Sports Football Inc. | 8 McFadden Road | | Easton | PA | 18045- | (800) 248-0019 | http://www.schut |

Record: 1 of 7

Report Output:



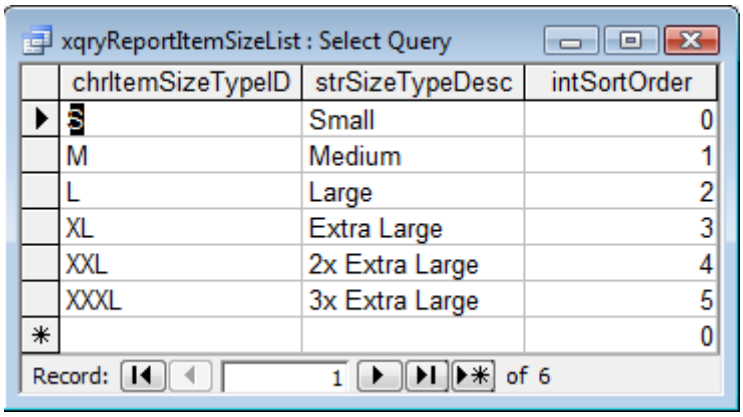
**Report – ITEM SIZE DEFINITIONS REPORT**

This report lists the item size definitions, this is probably the least complicated query / report in the system.

SQL Query – **xqryReportItemSizeList**

```
SELECT tblItemSizeTypes.*
FROM tblItemSizeTypes
ORDER BY tblItemSizeTypes.intSortOrder;
```

Query Output:

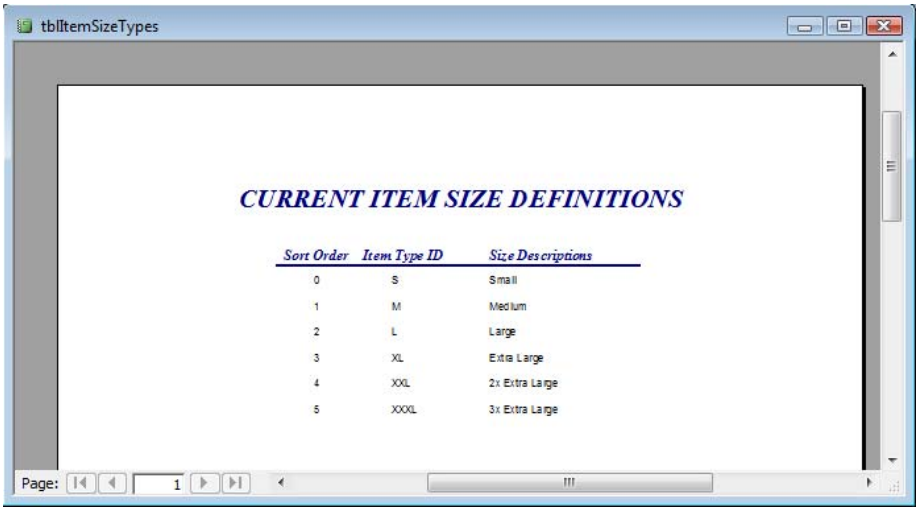


xqryReportItemSizeList : Select Query

| chrItemSizeTypeID | strSizeTypeDesc | intSortOrder |
|---|---|---|
| S | Small | 0 |
| M | Medium | 1 |
| L | Large | 2 |
| XL | Extra Large | 3 |
| XXL | 2x Extra Large | 4 |
| XXXL | 3x Extra Large | 5 |
| | | 0 |

Record: 1 of 6

Report Output:



tblItemSizeTypes

CURRENT ITEM SIZE DEFINITIONS

| Sort Order | Item Type ID | Size Descriptions |
|---|---|---|
| 0 | S | Small |
| 1 | M | Medium |
| 2 | L | Large |
| 3 | XL | Extra Large |
| 4 | XXL | 2x Extra Large |
| 5 | XXXL | 3x Extra Large |

Page: 1

**Report – ITEM TYPE DEFINITIONS REPORT**

This report lists the item type definitions, sorted by whether or not the item is for Players Only, or not.

SQL Query – **xqryReportItemTypeList**

```
SELECT tblItemCategoryType.chrItemTypeID,
tblItemCategoryType.strItemTypeDescription,
tblItemCategoryType.bitPlayerItem
FROM tblItemCategoryType
ORDER BY tblItemCategoryType.bitPlayerItem;
```

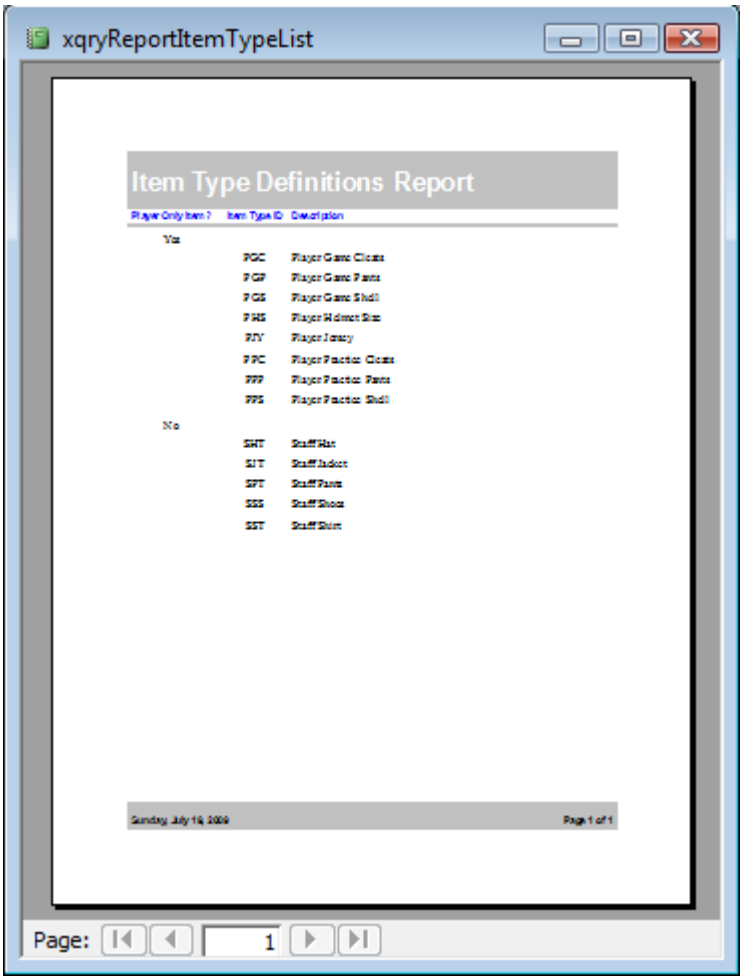
Query Output:

xqryReportItemTypeList : Select Query

| chrItemTypeID | strItemTypeDescription | bitPlayerItem |
|---|---|---|
| PPS | Player Practice Shell | ☑ |
| PPP | Player Practice Pants | ☑ |
| PPC | Player Practice Cleats | ☑ |
| PJY | Player Jersey | ☑ |
| PHS | Player Helmet Size | ☑ |
| PGS | Player Game Shell | ☑ |
| PGP | Player Game Pants | ☑ |
| PGC | Player Game Cleats | ☑ |
| SST | Staff Shirt | ☐ |
| SSS | Staff Shoes | ☐ |
| SPT | Staff Pants | ☐ |
| SJT | Staff Jacket | ☐ |
| SHT | Staff Hat | ☐ |
| ✳ | | ☐ |

Record: 1 of 13

Report Output:



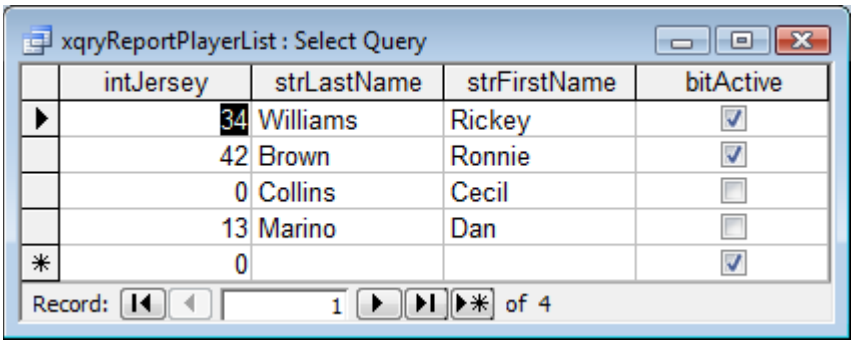
**Report – PLAYER LIST by Jersey Number**

This report lists all Players by Jersey Number, sorted by active status.

SQL Query – **xqryReportPlayerList**

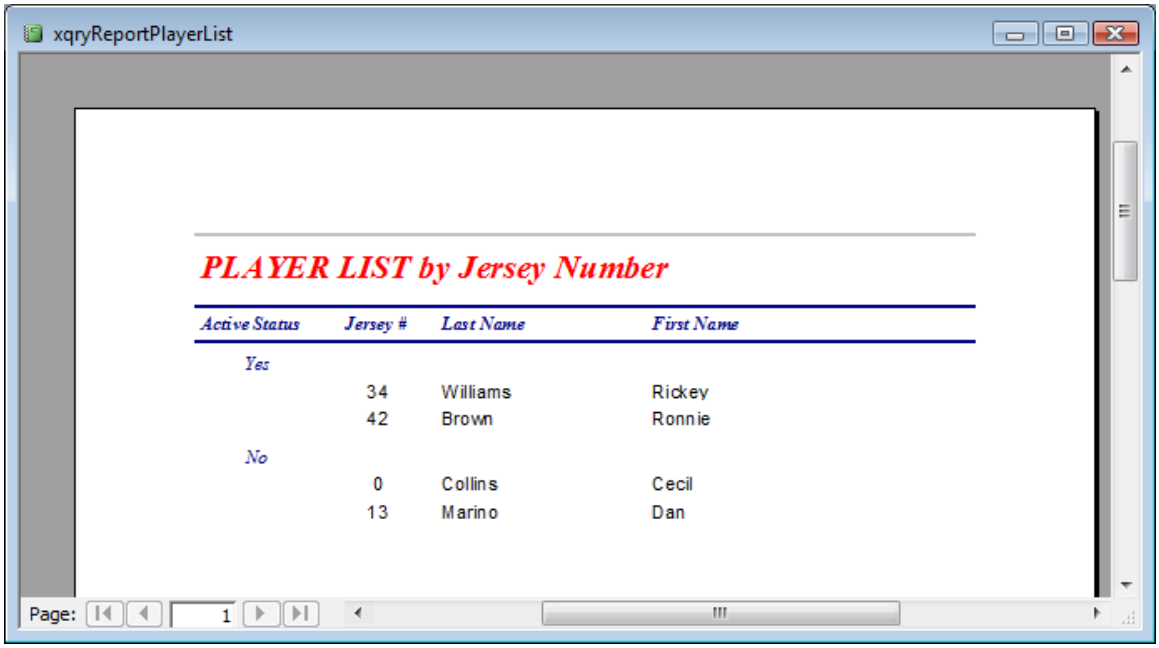
```
SELECT tblPersons.intJersey, tblPersons.strLastName,
tblPersons.strFirstName, tblPersons.bitActive
FROM tblPersons
WHERE (((tblPersons.chrPersonTypeID)="P"))
ORDER BY tblPersons.bitActive, tblPersons.intJersey;
```

Query Output:




| intJersey | strLastName | strFirstName | bitActive |
|---|---|---|---|
| 34 | Williams | Rickey | ☑ |
| 42 | Brown | Ronnie | ☑ |
| 0 | Collins | Cecil | ☐ |
| 13 | Marino | Dan | ☐ |
| 0 | | | ☑ |

Report Output:



## Report – STAFF & PLAYER SIZES LIST

This report lists the clothing sizes of all players and staff.

SQL Query – **xqryReportPlayerSizes**

```
SELECT tblPersons.guidPersonID, tblPersons.strLastName,
tblPersons.strFirstName, tblPersons.chrPersonTypeID,
tblPersons.intJersey, tblPersons.bitActive, tblPersons.intShirtSize,
tblPersons.intPantSize, tblPersons.intGrassShoeSize,
tblPersons.intJacketSize, tblPersons.intHatSize,
tblPersons.intPracticeShell, tblPersons.intGameShell,
tblPersons.intJerseySize, tblPersons.intGamePants,
tblPersons.intPracticePants, tblPersons.intHelmetSize,
tblPersons.intGameShoeSize
FROM tblPersons;
```

Query Output:

xqryFormPlayerSizes : Select Query

| guidPersonID | strLastName | strFirstName | chrPersonTypeI | intJersey | bitActive | intShirtSize | intPantSize | intGrassShoeSi | intJac |
|---|---|---|---|---|---|---|---|---|---|
| 7 | Marino | Dan | P | 13 | ☐ | 23 | 34 | 12 | |
| 8 | Jaspers | Todd | S | 0 | ☑ | 43 | 36 | 12 | |
| 9 | Howard | Tery | S | 0 | ☑ | 33 | 28 | 8 | |
| 10 | Curbelo | Joe | S | 0 | ☑ | 42 | 34 | 10 | |
| 12 | Gaine | Brian | C | 0 | ☑ | 0 | 0 | 0 | |
| 13 | Ireland | Jeff | C | 0 | ☑ | 0 | 0 | 0 | |
| 14 | Shea | Chris | C | 0 | ☑ | 0 | 0 | 0 | |
| 15 | Brown | Ronnie | P | 42 | ☑ | 0 | 0 | 0 | |
| 16 | Williams | Rickey | P | 34 | ☑ | 0 | 0 | 0 | |
| 17 | Zarchin | Erica | S | 0 | ☑ | 0 | 0 | 0 | |
| 18 | Farendorf | Jessica | S | 0 | ☑ | 0 | 0 | 0 | |
| 19 | McCormick | Gwen | S | 0 | ☑ | 0 | 0 | 0 | |
| 20 | Collins | Cecil | P | 0 | ☐ | 0 | 0 | 0 | |
| 21 | Abella | Lilliana | S | 0 | ☑ | 0 | 0 | 0 | |
| (AutoNumber) | | | | 0 | ☑ | 0 | 0 | 0 | |

Record: 3 of 14

Report Output:

xqryReportPlayerSizes

### STAFF PLAYER SIZE REFERENCE SHEET

| Type ID | | | Shirt | Pants | Shoe | Jacket | Hat | PShell | GShell | Jersey | GPants | PPants | Helm | GShoe |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | | | | | | | | | | | | | | |
| | Brian | Gaine | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Jeff | Ireland | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Chris | Shea | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | | | | | | | | | | | | | | |
| | Ronnie | Brown | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Cecil | Collins | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Dan | Marino | 23 | 34 | 12 | 23 | 21 | 22 | 21 | 43 | 36 | 38 | 23 | 12 |
| | Rickey | Williams | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | | | | | | | | | | | | | | |
| | Lilliana | Abella | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Joe | Curbelo | 42 | 34 | 10 | 6 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Jessica | Farendorf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Tery | Howard | 33 | 28 | 8 | 33 | 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Todd | Jaspers | 43 | 36 | 12 | 42 | 22 | 0 | 0 | 34 | 0 | 0 | 0 | 0 |
| | Gwen | McCormick | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Page: 1

## MACRO – Print Issuance Report (Current)

This macro produces a report and automatically prints it.

Print Issuance Report (Current)

| | Action | Comment |
|---|---|---|
| ▶ | OpenQuery | Open Query - xqryIssuanceReport |
| | OpenReport | Print Report - Current Issuance Report |
| | | |

SQL Query – **xqryIssuanceReport (MAKE TABLE QUERY)**

```
SELECT tblItems.strItemDescription AS [Item Description],
tblItems.intNumOnHand AS Total, Sum(tblItemIssue.intNumIssued) AS Issued,
tblItems.intNumOnHand-Sum(tblItemIssue.intNumIssued) AS Available,
tblItemSizeTypes.strSizeTypeDesc AS [Size],
tblItemCategoryType.strItemTypeDescription AS Type INTO
tblItemIssuanceReport_TEMP
FROM (tblItemSizeTypes INNER JOIN (tblItemCategoryType INNER JOIN
tblItems ON tblItemCategoryType.chrItemTypeID = tblItems.chrItemTypeID)
ON tblItemSizeTypes.chrItemSizeTypeID = tblItems.chrItemSizeTypeID) INNER
JOIN tblItemIssue ON tblItems.guidItemID = tblItemIssue.guidItemID
GROUP BY tblItems.strItemDescription, tblItems.intNumOnHand,
tblItemSizeTypes.strSizeTypeDesc,
tblItemCategoryType.strItemTypeDescription;
```

Query Output:

xqryReportIssuance : Select Query

| | Type | Size | Description | Total | Issued | Available | intSortOrder |
|---|---|---|---|---|---|---|---|
| ▶ | Player Practice Cleats | Large | Nike Cleats | 52 | | | 2 |
| | Staff Shirt | Small | Small Dolphins AQ | 6 | 1 | 5 | 0 |
| | Staff Shirt | Large | Large Dolphins AQ | 10 | 3 | 7 | 2 |
| | Staff Shirt | Extra Large | XL Dolphins AQUA | 4 | | | 3 |

Record: 1 of 4

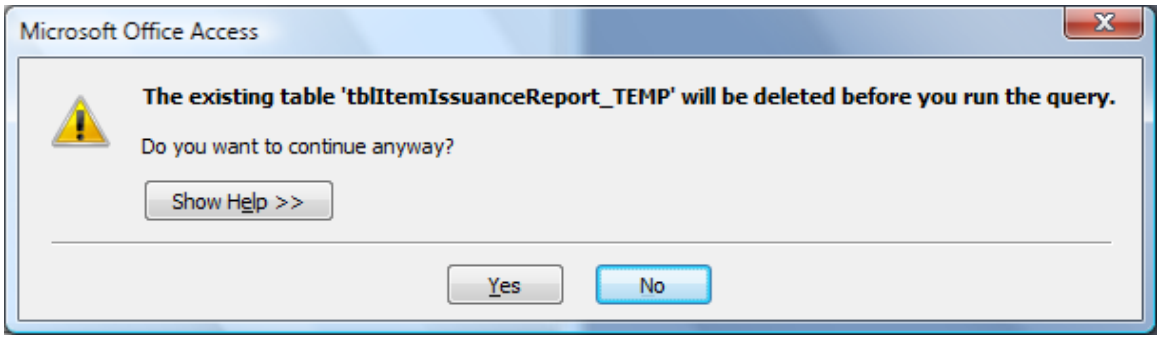
Report Output:

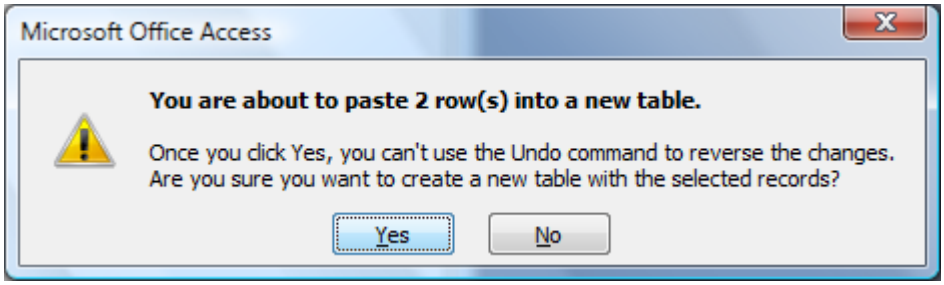
Microsoft Office Access

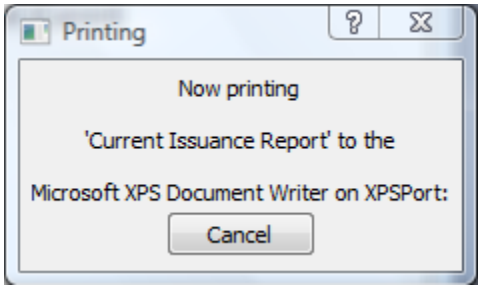
**You are about to run a make-table query that will modify data in your table.**

Are you sure you want to run this type of action query?
For information on how to prevent this message from displaying every time you run an action query, click Help.

Yes No Help


Microsoft Office Access
The existing table 'tblItemIssuanceReport_TEMP' will be deleted before you run the query.
Do you want to continue anyway?
Show Help >>
Yes
No


Microsoft Office Access
You are about to paste 2 row(s) into a new table.
Once you click Yes, you can't use the Undo command to reverse the changes.
Are you sure you want to create a new table with the selected records?
Yes
No


Printing
Now printing
'Current Issuance Report' to the
Microsoft XPS Document Writer on XPSPort:
Cancel