UFHealth
INFORMATION TECHNOLOGY TRAINING

Microsoft® Office Training

Microsoft Access 2010

Understanding Relationships

Email: training@health.ufl.edu
Web Page: http://training.health.ufl.edu

Microsoft Access: Understanding Relationships

2.5 hours

Proper relationships are essential for "good" data entry. The relationship between two tables allows for specific rules to be applied. Relationships can be set so data has to be entered in the correct order – a student has to exist in the database before they can register. Relationships can be set to prevent orphan data – patients cannot be deleted if they have attended appointments. Topics for this workshop include a discussion of one-to-one and one-to-many relationships; creating junction tables; working with primary and foreign keys; setting the cascade update and cascade delete options. The hands on portion of this workshop will be an individual assignment.

Worksheets:

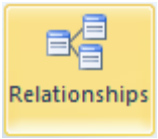
UF UNIVERSITY of FLORIDA

Pandora Rose Cowart
Education/Training Specialist
UF Health IT Training

C3-013 Communicore
PO Box 100152
Gainesville, FL 32610-0152

(352) 273-5051
prcowart@ufl.edu
http://training.health.ufl.edu

Relationships



You can view the relationships of your database by clicking on the **Relationships** button on the **Database Tools** tab.

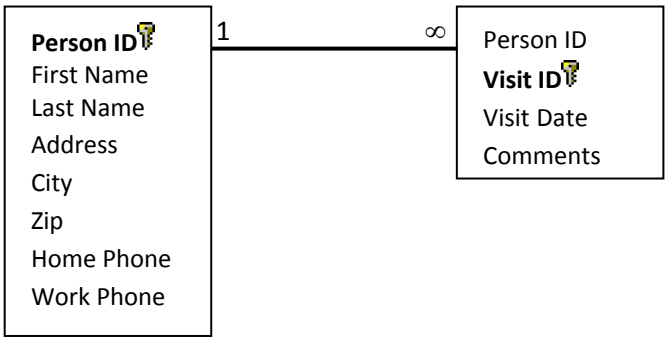
To create *true* relationships, ones that will protect your data integrity, you will need to create **Primary Key** fields in your tables. A **Primary Key** field is set in the design view of the table; it ensures that this field (column) will have no duplicates and no blanks (null values).

You also need to have one field in both tables that is the same. Access is not particular about the spelling of the fieldnames, but the data type and field size must be the same. To create a true relationship one field in the relationship must be the Primary Key or indexed, no duplicates field; this ensures at least one side of the relationship is unique.

One-to-many

The most common relationship you will probably create will be a ***one-to-many***. This is a link between a Primary Key field and a non-Primary Key field.

Example: One person can visit many times. (i.e. Julie Jacobs visited on 01/23, 01/31, and 03/10.)



PERSON TABLE

Person ID	*First Name*	*Last Name*	*Address*
P1	Jack	Johnson	PO Box 123
P2	Julie	Jacobs	PO Box 456
P3	Jeffery	Jackson	PO Box 789

VISIT TABLE

Person ID	***VisitID***	***VisitDate***	***Comments***
P1	**V1**	**1/23/03**	**First Contact**
P2	**V2**	**1/23/03**	**First Contact**
P1	**V3**	**1/31/03**	**Gave Card**
P2	**V4**	**3/10/03**	**Likes hot tea**
P2	**V5**	**2/15/03**	**Spouse John**

The Person table contains each individual that may visit. The Visit table contains each unique visit.

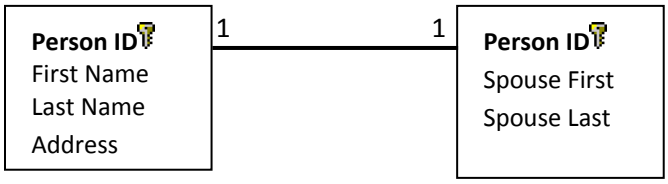
One Person	1-------> ∞	Many Visits
One Person	1<------ 1	One Visit

This is a **One-To-Many** relationship.

One-to-one

The other *true* relationship you can create in Access is a ***one-to-one relationship***. This is a link between a Primary Key field and a matching Primary Key field in the second table.

Example: One person can have one spouse. (i.e. Jack Johnson is married to Mary Johnson.)



PERSON TABLE

Person ID	First Name	Last Name	Address
P1	Jack	Johnson	PO Box 123
P2	Julie	Jacobs	PO Box 456
P3	Jeffery	Jackson	PO Box 789

SPOUSE TABLE

Person ID	Spouse First	Spouse Last
P1	Mary	Johnson
P3	Mark	Jacobs

One-to-one relationships can be merged into one table. You may wish to create new tables if the tables become too large and you want to keep your data set more managable, or if the data you want to track does not apply to a majority of your records. **One-to-one** relationships must have the ***SAME PRIMARY KEY*** in both tables.

In all relationships there must be a primary table, sometimes called a master or parent table, and a secondary table, sometimes called a slave or child table. In a one-to-many relationship, the "One" table is automatically the primary table. In a one-to-one relationship, access determines the primary table by the beginning side. For this example, we would drag the key from the person table to the key in the spouse table; this would set the person table as the primary in the one-to-one relationship.

Many-to-Many

Often you will find that you have a ***many-to-many relationship***.

Example: One student can have many tutors

(Jack Johnson is tutored by Mary McGuire, Misty Martin)

STUDENT TABLE

Student ID	First Name	Last Name	Address
S1	Jack	Johnson	PO Box 123
S2	Julie	Jacobs	PO Box 456
S3	Jeffery	Jackson	PO Box 789
S4	John	Jenson	PO Box 021

TUTOR TABLE

Tutor ID	First Name	Last Name	Office#
T1	Mary	McGuire	3-045
T2	Matt	Madden	3-041
T3	Misty	Martin	3-048
T4	Melanie	Masters	3-053

but each tutor can support many students

(Matt Madden tutors Jack Johnson, Julie Jacobs, and John Jenson)

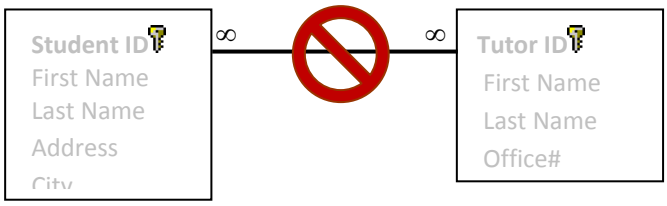
STUDENT TABLE

Student ID	First Name	Last Name	Address
S1	Jack	Johnson	PO Box 123
S2	Julie	Jacobs	PO Box 456
S3	Jeffery	Jackson	PO Box 789
S4	John	Jenson	PO Box 021

TUTOR TABLE

Tutor ID	First Name	Last Name	Office#
T1	Mary	McGuire	3-045
T2	Matt	Madden	3-041
T3	Misty	Martin	3-048
T4	Melanie	Masters	3-053

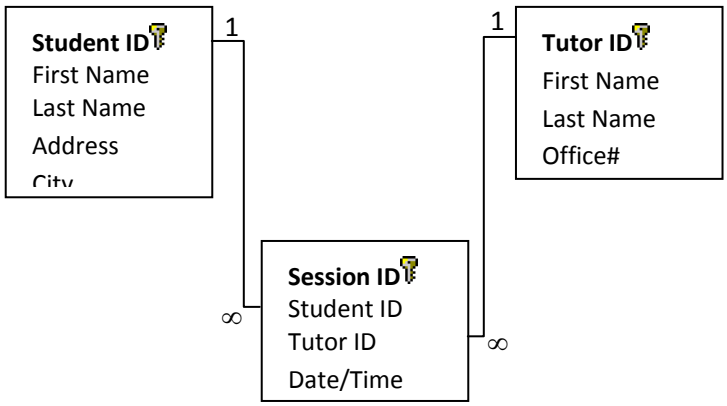
This kind of relationship is too complicated. It requires the intervention of a **Junction Table** to keep track of individual occurrences.



A Junction Table, sometimes called an Instance Table, keeps track of where the two sets of data interact.

- Patients and Medications → Perscriptions
- Doctors and Patients → Appointments
- Students and Teachers → Classes
- Customers and Inventory → Sales
- Students and Tutors → Study Session

The session table pulls the students and the tutors into unique records by keeping track of the date and time of each session.



1 Student, Many Tutors	**1 Tutor, Many Students**
Jack Johnson - Mary, 2/14, 2pm - Mark, 2/14, 5pm - Misty, 2/15, 5pm	Mary McGuire - Jack, 2/14, 2pm - Julie, 2/14, 4pm - John, 2/14, 6pm

Student ID	First Name
S1	Jack
S2	Julie
S3	Jeffery
S4	John

Stud ID	Sess ID	Date/Time	Tutor ID
S1	1	2/14, 2pm	T1
S2	2	2/14, 4pm	T1
S1	3	2/14, 5pm	T5
S4	4	2/14, 6pm	T1
S1	5	2/15, 5pm	T3

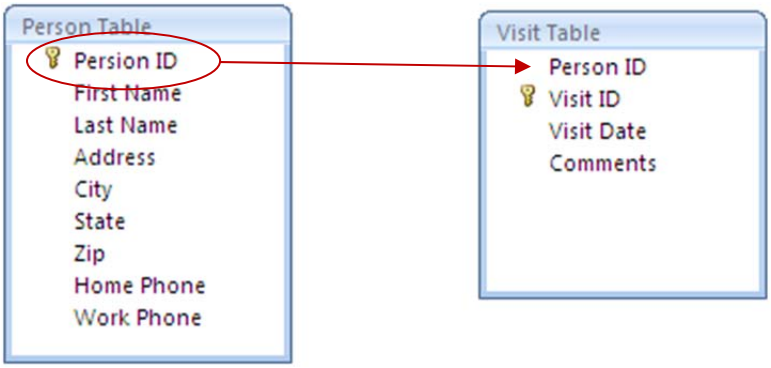
Tutor ID	First Name
T1	Mary
T2	Matt
T3	Misty
T4	Melanie
T5	Mark

Junction Tables have a minimum of three fields.

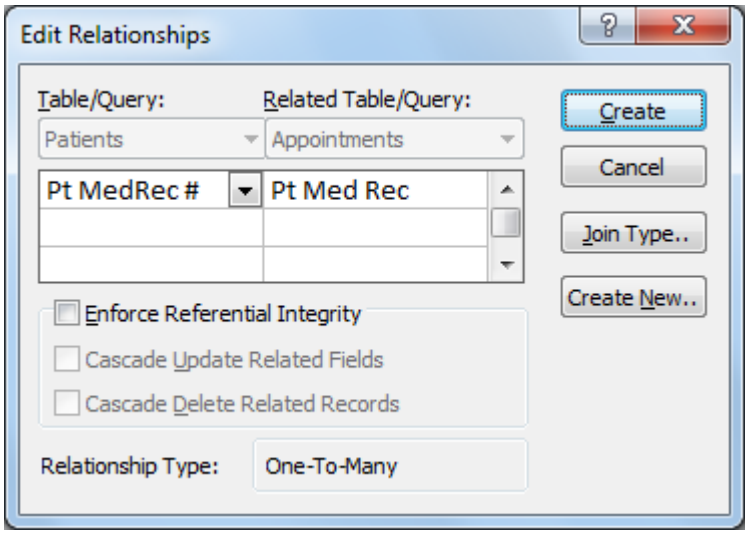
- Its own primary key
- Primary key from first joining table
- Primary key from the second joining table

Create/Edit Relationships

When you create your relationship, one of tables will be a *Primary Table.* The *Primary Table* data will exist before the *Secondary Table* data. To create your relationship, take the **primary key** field from the *Primary Table* and drag it on top of the matching field in the *Secondary Table*.



When you first create your relationship, or if you double-click on the slope of the line to edit an existing relationship, you will see the **Edit Relationships** window.



Here you see the two tables and fields that you are joining. If you accidentally picked the wrong field, you can change it with the drop down menu. If you click **Create** at this point, you will get a simple relationship showing the two fields are connected, but Access will not protect your data across the tables. To do this you will need to **Enforce Referential Integrity**. (See the next page).

The **Relationship Type** (at the bottom of the window) will depend on the status of the original fields.

- If you drag a primary key field to a non-key field, you will see **One-To-Many**.
- If you drag a primary key field to another primary key field, you will get a **One-To-One**.
- If you drag a non-primary key field to another non-primary key field you will see **Indeterminate**. You cannot Enforce Referential Integrity with an indeterminate (many-to-many) relationship type.

Enforce Referential Integrity

Enforcing Referential Integrity applies a set of rules to your tables that will ensure your data reliability. This ensures you will not be able to sell anything to Mr. Smith, unless he exists in your customer table, and that you cannot delete Mr. Smith if he has purchased anything.

From Access 2010 Help File

> The purpose of using referential integrity is to prevent orphan records and to keep references synchronized so that you don't have any records that reference other records that no longer exist. You enforce referential integrity by enabling it for a table relationship. Once enforced, Access rejects any operation that would violate referential integrity for that table relationship. Access rejects updates that change the target of a reference, and also deletions that remove the target of a reference.

Once you have enforced the referential integrity, you have two new choices, Cascade Update Related Fields and Cascade Delete Related Records.

Cascade Update Related Fields

With the referential integrity enforced, Access won't allow you to change the linked field in either table. If you check the "Cascade Update Related Records", you will be able to change the data in the primary table.

One-To-Many relationship, with enforced referential integrity but no cascade update.
The primary key in the Person Table is "Last Name". We need to change Johnson to Johnston.

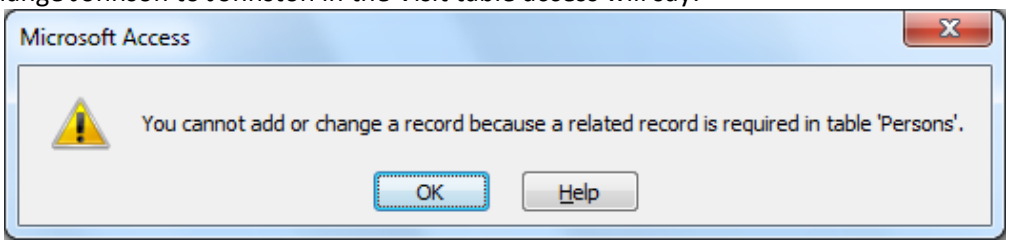
VISIT TABLE

Last Name	Visit ID	Visit Date	Comments
Johnson	V1	1/23/03	First Contact
Jacobs	V2	1/23/03	First Contact
Johnson	V3	1/31/03	Gave Card
Johnson	V4	3/10/03	Likes hot tea
Jackson	V5	2/15/03	Spouse Joan

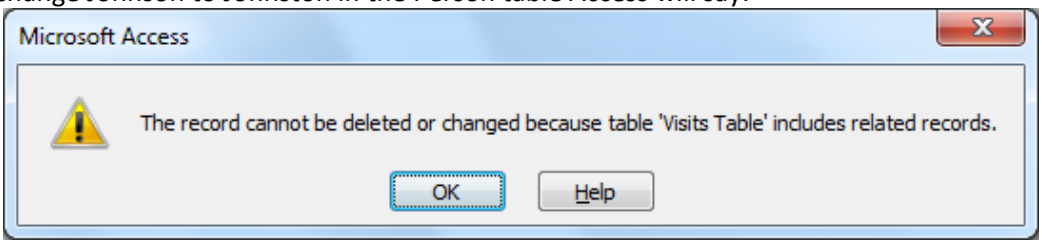
PERSON TABLE

First Name	Last Name	Address
Jack	Johnson	PO Box 123
Julie	Jacobs	PO Box 456
Jeffery	Jackson	PO Box 789

If you try to change Johnson to Johnston in the Visit table access will say:

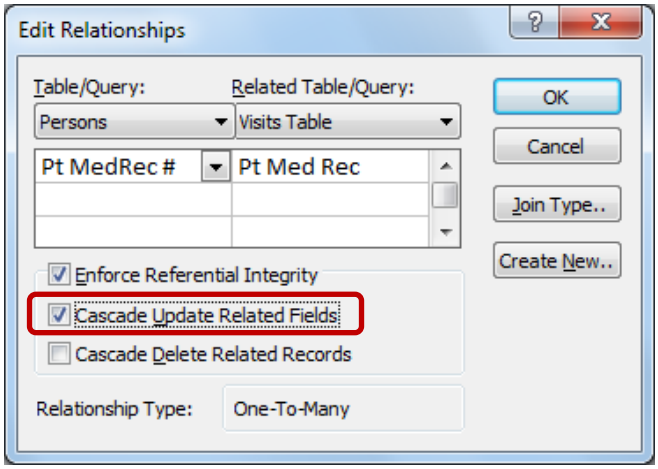


If you try to change Johnson to Johnston in the Person table Access will say:



Solution: Create a new record in the Person table for Mr. Johnston, then change all of the Johnsons to Johnston in the Visit table, and then remove the Johnson record from the Person table.

Or turn on the **Cascade Update Related Fields**.



If you try to change Johnson to Johnston in the Visits table access will give you the same error as above, however if you try to change Johnson to Johnston in the Person table Access will change the data and it will automatically change all the Johnsons to Johnstons in the Visits table.

Cascade Delete Related Records

Once you enforce referential integrity, Access won't allow you to delete the primary key field in the primary table. If you check the "Cascade Delete Related Records", you will be allowed to delete the data in the primary table, and it will (cascade) delete all the related records in the secondary table.

Example: We need to Delete Johnson

One-To-Many relationship, with enforced referential integrity but no cascade delete.

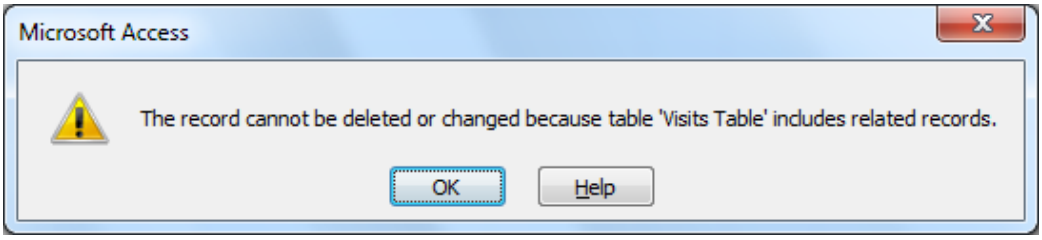
VISIT TABLE

Last Name	VisitID	VisitDate	Comments
Johnson	V1	1/23/03	First Contact
Jacobs	V2	1/23/03	First Contact
Johnson	V3	1/31/03	Gave Card
Johnson	V4	3/10/03	Likes hot tea
Jackson	V5	2/15/03	Spouse Joan

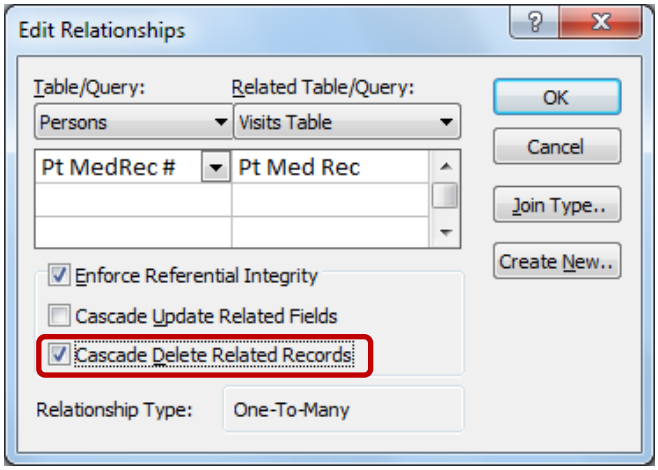
PERSON TABLE

First Name	Last Name	Address
Jack	Johnson	PO Box 123
Julie	Jacobs	PO Box 456
Jeffery	Jackson	PO Box 789

You can delete the records in the Visits table with no problem. There doesn't have to be matching records in the Visits Table for each person in the Person Table. However, if you try to delete Johnson from the Person table, you will get this message:



If you turn on the Cascade Delete Related Records, you will be able to delete Johnson from the Person table, and it will delete all the related records in the Visits table.



VISIT TABLE

Last Name	VisitID	VisitDate	Comments
~~Johnson~~	~~V1~~	~~1/23/03~~	~~First Contact~~
Jacobs	V2	1/23/03	First Contact
~~Johnson~~	~~V3~~	~~1/31/03~~	~~Gave Card~~
~~Johnson~~	~~V4~~	~~3/10/03~~	~~Likes hot tea~~
Jackson	V5	2/15/03	Spouse Joan

PERSON TABLE

First Name	Last Name	Address
~~Jack~~	~~Johnson~~	~~PO Box 123~~
Julie	Jacobs	PO Box 456
Jeffery	Jackson	PO Box 789

This can be dangerous. Make sure you can really delete the related information before letting Access do this for you. Remember this can be changed at any time, so it could be turned temporarily on, and then back off when you are done editing your data.

The cascade options are set for EACH relationship. You can have one or the other or both depending on how you want each set of tables to work together.

Questions to ask yourself

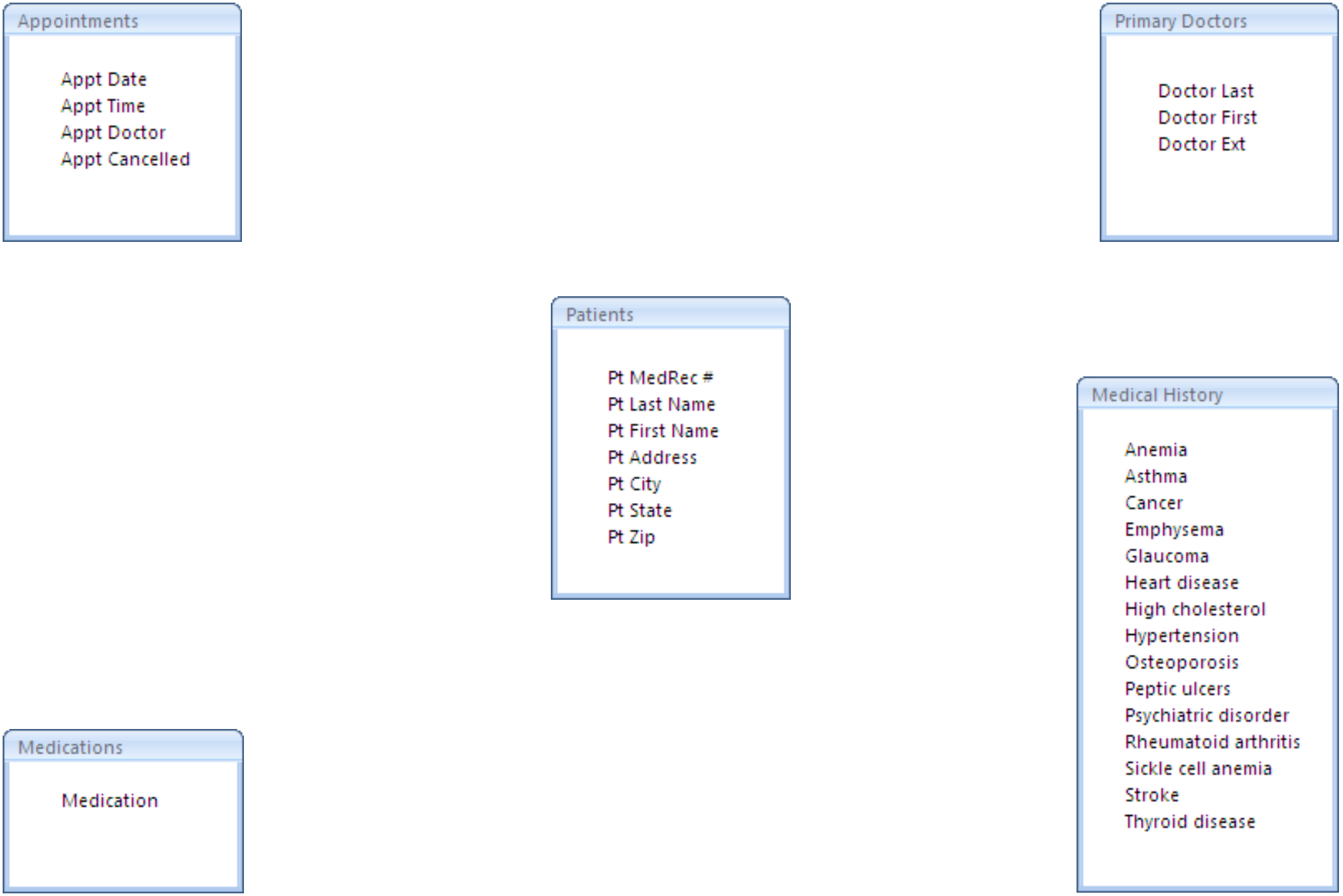
*Answers are **One** or **Many***

One Patient can have _____ Appointment(s)
One Appointment is for _____ Patient(s)

One Patient can have _____ Primary Doctor(s)
One Primary Doctor treats _____ Patient(s)

One Patient can have _____ Medical History
One Medical History belongs to _____ Patient

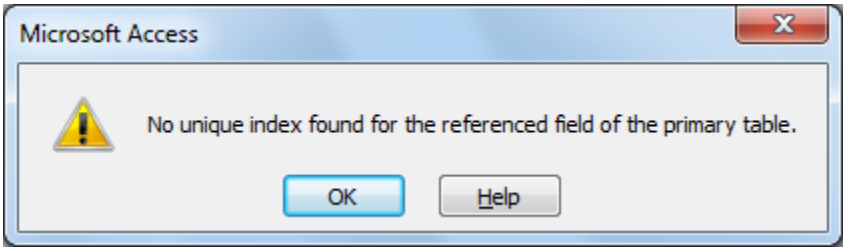
One Patient can take_____ Medication(s)
One medication can be taken by _____ Patient(s)


Appointments
Appt Date
Appt Time
Appt Doctor
Appt Cancelled
Primary Doctors
Doctor Last
Doctor First
Doctor Ext
Patients
Pt MedRec #
Pt Last Name
Pt First Name
Pt Address
Pt City
Pt State
Pt Zip
Medical History
Anemia
Asthma
Cancer
Emphysema
Glaucoma
Heart disease
High cholesterol
Hypertension
Osteoporosis
Peptic ulcers
Psychiatric disorder
Rheumatoid arthritis
Sickle cell anemia
Stroke
Thyroid disease
Medications
Medication

Relationship Error Messages

(1) No Unique Index Found for the referenced field…

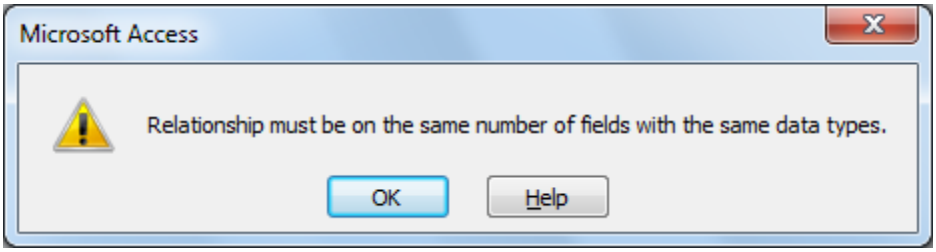
In order to enforce the referential integrity, one of the fields must be a unique index, a Primary Key. Check your tables and set a primary key.



(2) Relationships must be on the same number of fields…

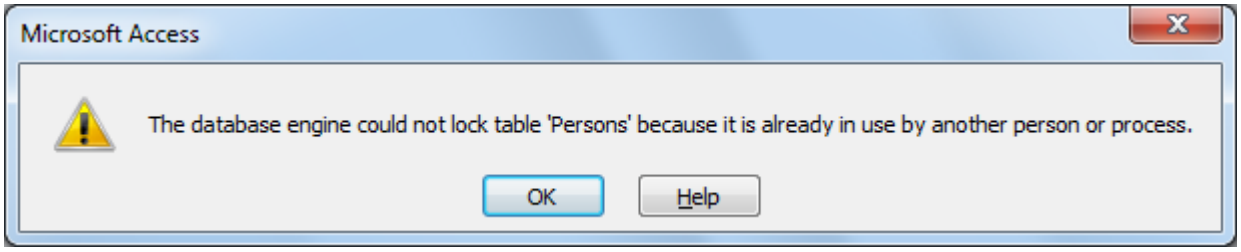
Your fields don't match. It's possible you are linking a text field to a number field, or a date to text, or possibly even a number (integer) to a number (double). You need to make sure the data in both fields is the same type with the same field size.

A note about AutoNumbers: The key side of the relationship will be an AutoNumber; the other side of the relationship will be a number. Both fields should have the same Data Type (usually long integer).



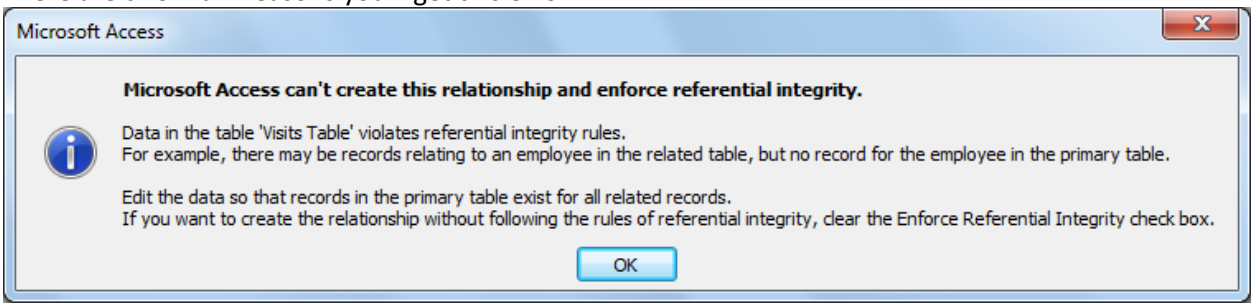
(3) The database engine could not lock table...

Most likely this is because you have one of the tables (or related objects) open.



(4) Microsoft Office Access can't create this relationship…

There are two main reasons you'll get this error.



1. You are going the wrong direction in a One-to-One relationship

    a. Cancel the relationship window, if necessary delete the link between the tables and try again. From the table where the data will always exist to the table where the data *may* exist. (Which data set comes first?)

2. You have created data in the secondary table that doesn't exist in the primary.

    a. This happens if you set both tables to have "AutoNumbers". Remember if one side is an AutoNumber; the other side must be a Number field of the same data type (usually long integer). You'll need to delete the relationship between the two tables, delete (don't just change, delete) the AutoNumber from the secondary table, and recreate the field as a number.

    b. If it's not the AutoNumber, and you cannot see which data is in the secondary table that's missing from the primary table, you can run a FIND UNMATCHED query. The results of the query should show you what is in the secondary table that was not in the primary.

(5) You can't delete the field.../You can't change the data…

If you try to modify a field in the table, that is part of a relationship you might get this error message. You need to delete the relationship and try this again. Remember to close the table before you try to delete the relationship.