WHEN I STARTED PROGRAMMING, WE DIDN'T HAVE ANY OF THESE SISSY "ICONS" AND "WINDOWS."
ALL WE HAD WERE ZEROS AND ONES -- AND SOMETIMES WE DIDN'T EVEN HAVE ONES.
I WROTE AN ENTIRE DATABASE PROGRAM USING ONLY ZEROS.
YOU HAD ZEROS? WE HAD TO USE THE LETTER "O."
S.Adams
© 1992 United Feature Syndicate, Inc.
4-8

Databases 1 – Organisation and Creation

Lecture 14 – COMPSCI111/111G SS 2020

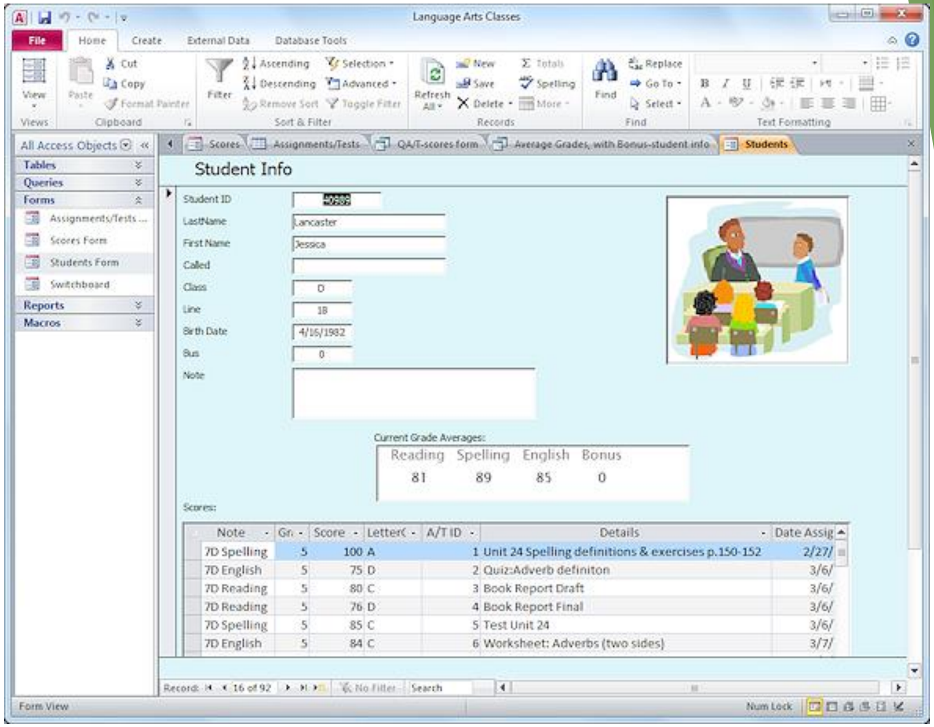
Today’s lecture

- What is a database?
- Understanding how data is organised in a database
- Creating a database in Microsoft Access

What is a database?

- A (typically large) collection of data about a particular topic, organized systematically
- Examples:
  - Catalogue of library books
  - Patients' files in a clinic
  - Entries in an address book
  - Students in a class
- Computers allow us to store and manage databases that contain very large amounts of information

Examples



Student Info

Student ID	40989
LastName	Lancaster
First Name	Jessica
Called	
Class	0
Line	18
Birth Date	4/16/1982
Bus	0
Note	

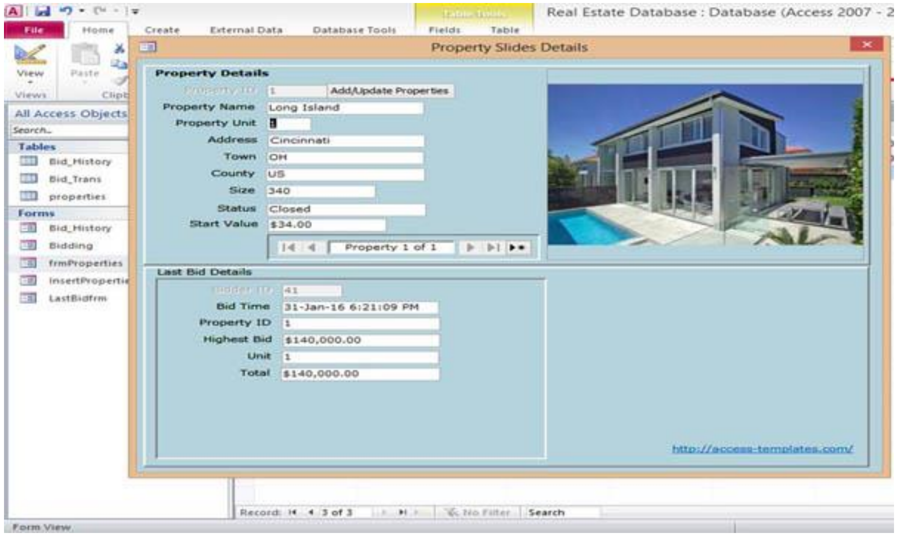
Current Grade Averages:

Reading	Spelling	English	Bonus
81	89	85	0

Scores:

Note	Gr.	Score	Letter(	A/T ID	Details	Date Assig
7D Spelling	5	100	A	1	Unit 24 Spelling definitions & exercises p.150-152	2/27/
7D English	5	75	D	2	Quiz:Adverb definiton	3/6/
7D Reading	5	80	C	3	Book Report Draft	3/6/
7D Reading	5	76	D	4	Book Report Final	3/6/
7D Spelling	5	85	C	5	Test Unit 24	3/6/
7D English	5	84	C	6	Worksheet: Adverbs (two sides)	3/7/

Record: 16 of 92



Real Estate Database : Database (Access 2007 - 2

Property Slides Details

Property Details

Property ID	1
Property Name	Long Island
Property Unit	1
Address	Cincinnati
Town	OH
County	US
Size	340
Status	Closed
Start Value	$34.00

Add/Update Properties

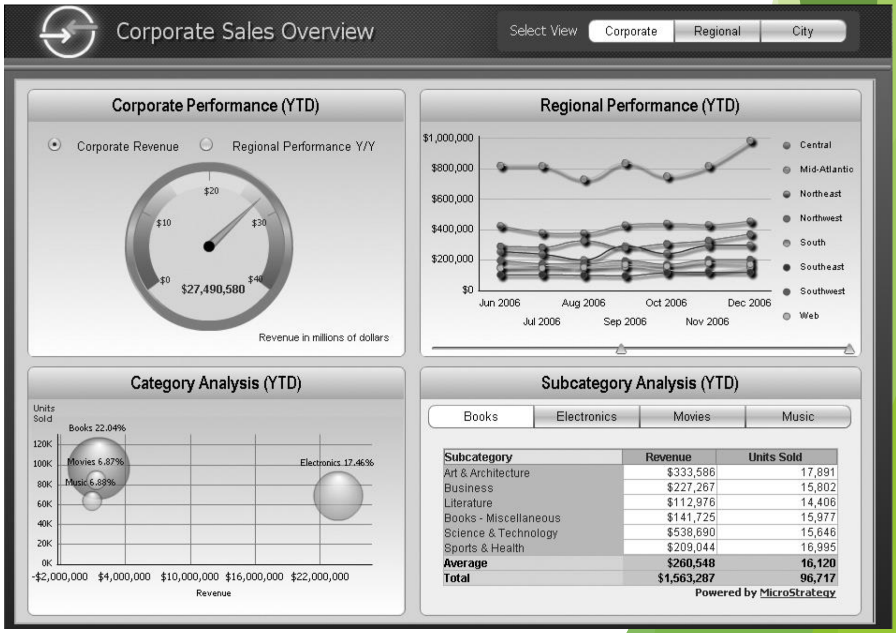
Property 1 of 1

Last Bid Details

Bidder ID	41
Bid Time	31-Jan-16 6:21:09 PM
Property ID	1
Highest Bid	$140,000.00
Unit	1
Total	$140,000.00

http://access-templates.com/

Record: 3 of 3



Subcategory	Revenue	Units Sold
Art & Architecture	$333,586	17,891
Business	$227,267	15,802
Literature	$112,976	14,406
Books - Miscellaneous	$141,725	15,977
Science & Technology	$538,690	15,646
Sports & Health	$209,044	16,995
Average	**$260,548**	**16,120**
Total	**$1,563,287**	**96,717**

Powered by MicroStrategy

Aspects of a database

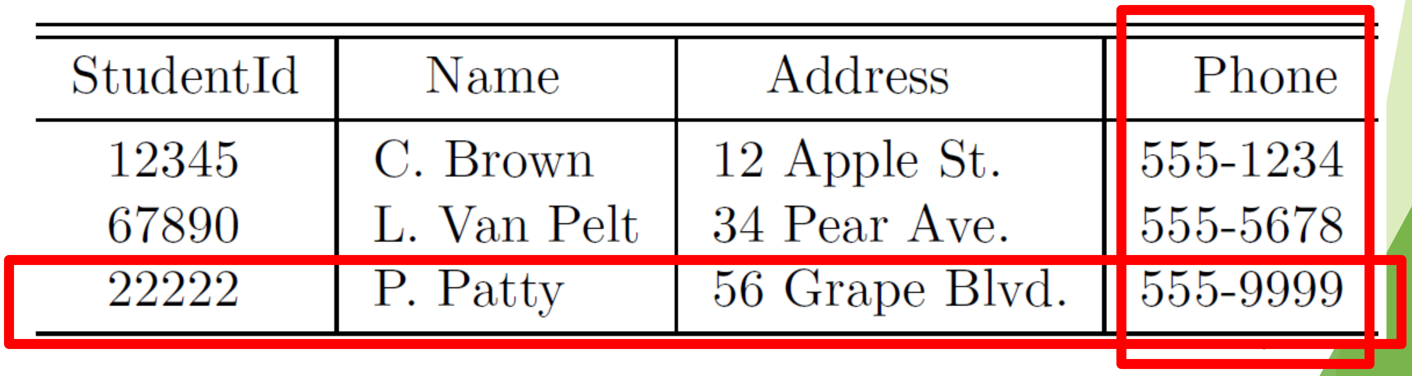
- Before we can create our database, we need to decide how to:
  1. **Organise** data in our database
  2. **Enter** data in our database
  3. **Retrieve** data from our database
  4. **Present** the retrieved data to the user

1. Organising data - models

- A **model** defines how data is organised and structured within the database
  - We're going to look at the relational model in this course
- When deciding what data to store in a database, we need to think about:
  - **Entities:** things about which we store information
    - Eg. students in uni, courses in uni
  - **Relationships:** specific connections among entities
    - Eg. *students* enrolled in *CompSci111/111G*

Organising data - tables

- The **relational model** was developed by Edgar Codd in 1970
- Data is stored and organized in tables
  - A table’s columns are called **fields**; an entity’s attributes
  - A table’s rows are called **records**; one instance of an entity
- A collection of tables form a **database**



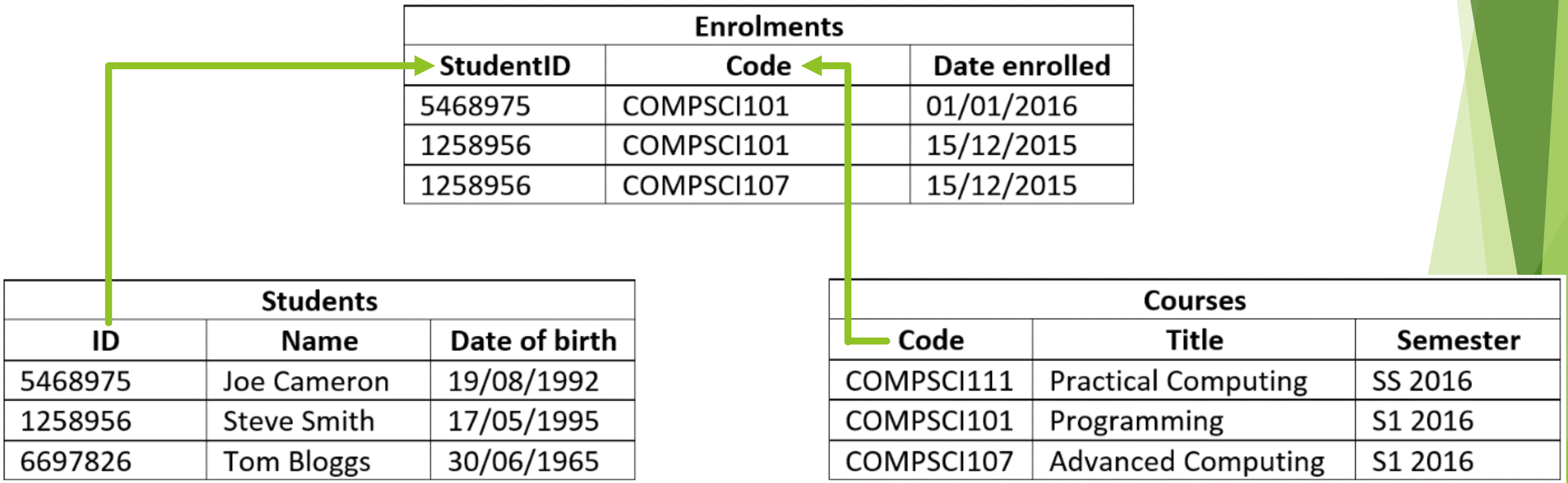
Field

StudentId	Name	Address	Phone
12345	C. Brown	12 Apple St.	555-1234
67890	L. Van Pelt	34 Pear Ave.	555-5678
22222	P. Patty	56 Grape Blvd.	555-9999

Record

Organising data

- Tables are connected together using **relationships**, thereby creating links between different entities



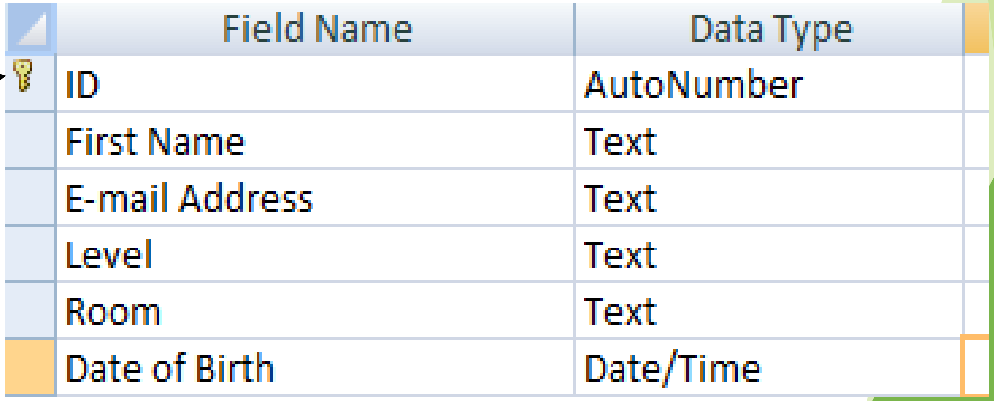
Enrolments		
StudentID	**Code**	**Date enrolled**
5468975	COMPSCI101	01/01/2016
1258956	COMPSCI101	15/12/2015
1258956	COMPSCI107	15/12/2015

Students		
ID	**Name**	**Date of birth**
5468975	Joe Cameron	19/08/1992
1258956	Steve Smith	17/05/1995
6697826	Tom Bloggs	30/06/1965

Courses		
Code	**Title**	**Semester**
COMPSCI111	Practical Computing	SS 2016
COMPSCI101	Programming	S1 2016
COMPSCI107	Advanced Computing	S1 2016

Organising data

- There are two parts to a relationship; **primary key** and **foreign key**
- 1. Primary key:
  - Generally, all tables must have a primary key field
  - All records must have a value in the primary key field
  - The primary key's values must be unique

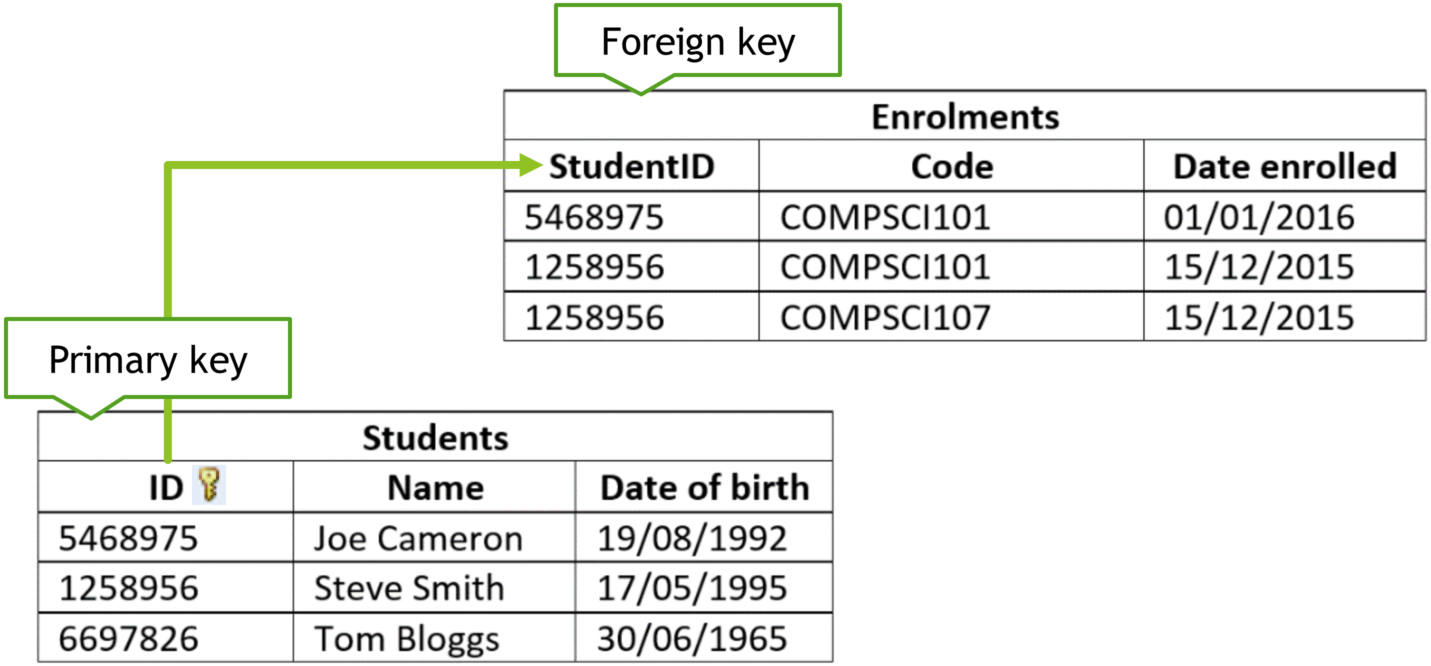


Field Name	Data Type
ID	AutoNumber
First Name	Text
E-mail Address	Text
Level	Text
Room	Text
Date of Birth	Date/Time

Primary key

Organising data

- 2. Foreign key
  - A field in one table that is related to a primary key field in another table
  - Creates a **connection** between the two fields
  - Can take blank values and/or repeated value depending on the relationship type



Foreign key

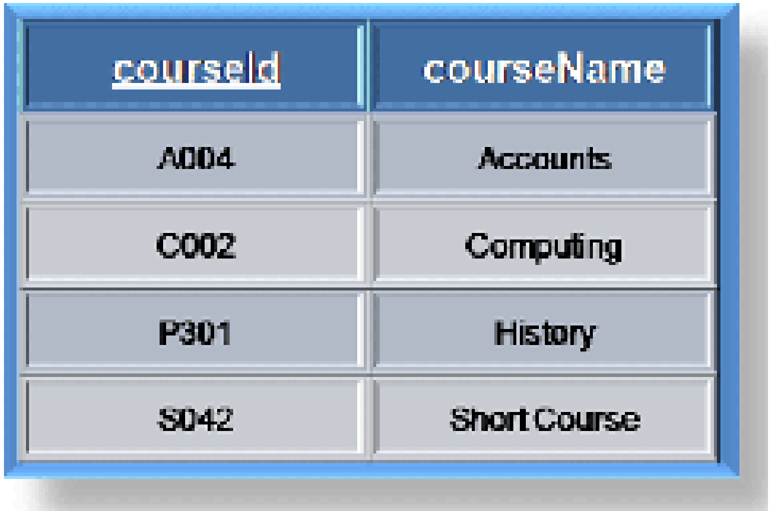
Enrolments		
StudentID	**Code**	**Date enrolled**
5468975	COMPSCI101	01/01/2016
1258956	COMPSCI101	15/12/2015
1258956	COMPSCI107	15/12/2015

Primary key

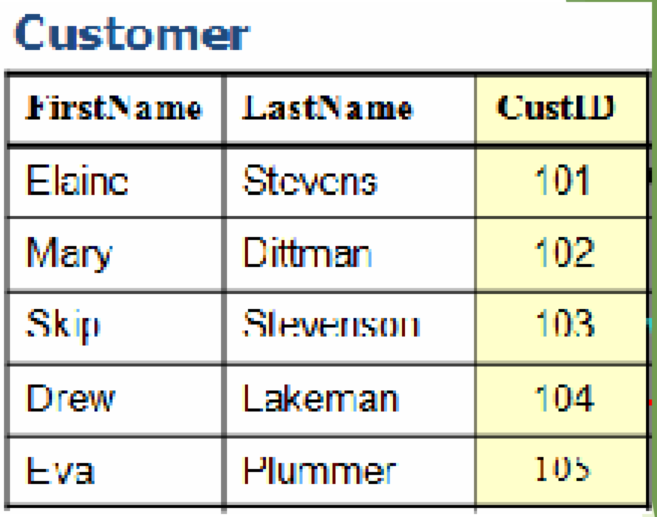
Students		
ID 🔑	**Name**	**Date of birth**
5468975	Joe Cameron	19/08/1992
1258956	Steve Smith	17/05/1995
6697826	Tom Bloggs	30/06/1965

Exercise 1

- Consider the following tables:
  - Identify the primary key:



courseId	courseName
A004	Accounts
C002	Computing
P301	History
S042	Short Course



Customer

FirstName	LastName	CustID
Elaine	Stevens	101
Mary	Dittman	102
Skip	Stevenson	103
Drew	Lakeman	104
Eva	Plummer	105

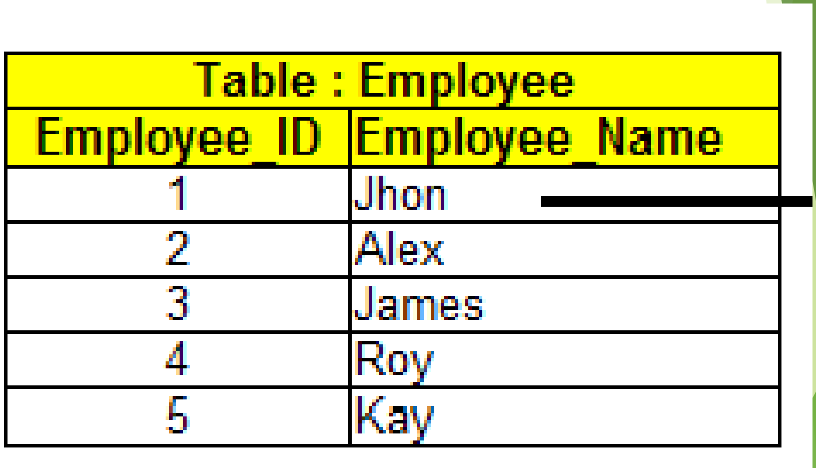
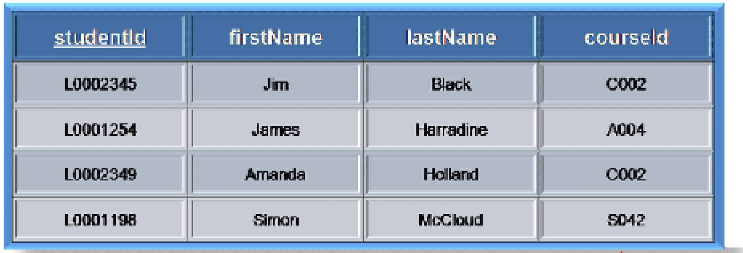


Table : Employee

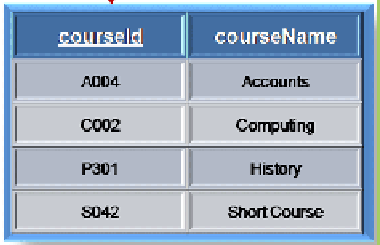
Employee_ID	Employee_Name
1	Jhon
2	Alex
3	James
4	Roy
5	Kay

Exercise 2:



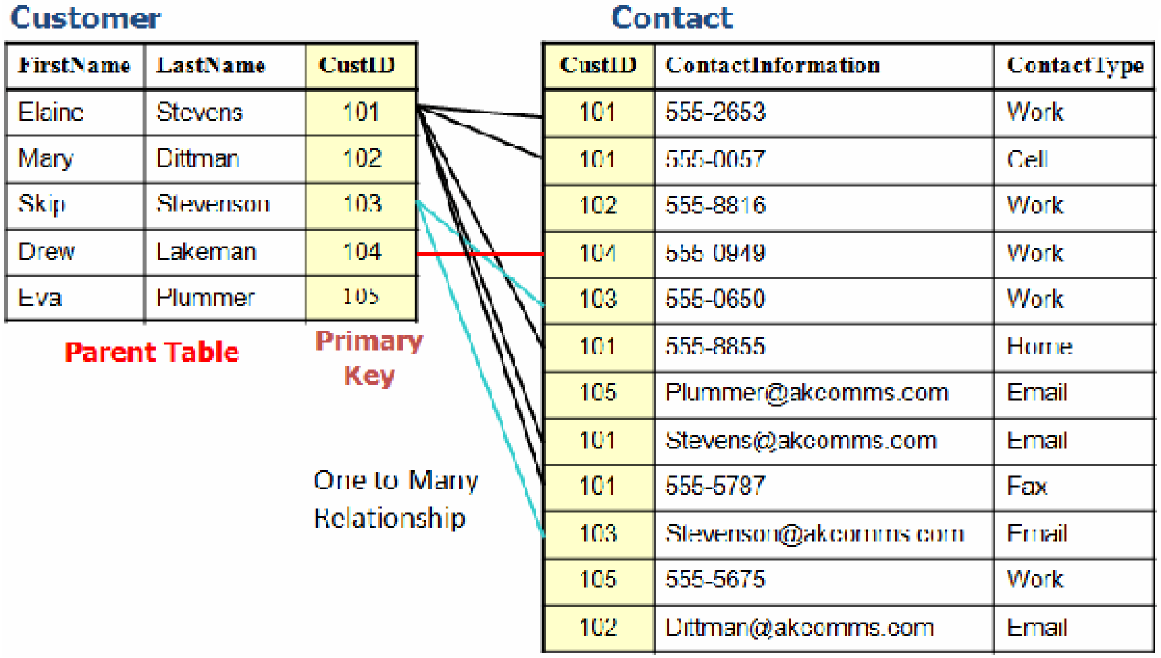
studentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042

Relationship



courseId	courseName
A004	Accounts
C002	Computing
P301	History
S042	Short Course

- Identify the foreign key:



Customer

FirstName	LastName	CustID
Elaine	Stevens	101
Mary	Dittman	102
Skip	Stevenson	103
Drew	Lakeman	104
Eva	Plummer	105

Parent Table

Primary Key

One to Many Relationship

Contact

CustID	ContactInformation	ContactType
101	555-2653	Work
101	555-0057	Cell
102	555-8816	Work
104	555-0949	Work
103	555-0650	Work
101	555-8855	Home
105	Plummer@akcomms.com	Email
101	Stevens@akcomms.com	Email
101	555-5787	Fax
103	Stevenson@akcomms.com	Email
105	555-5675	Work
102	Dittman@akcomms.com	Email

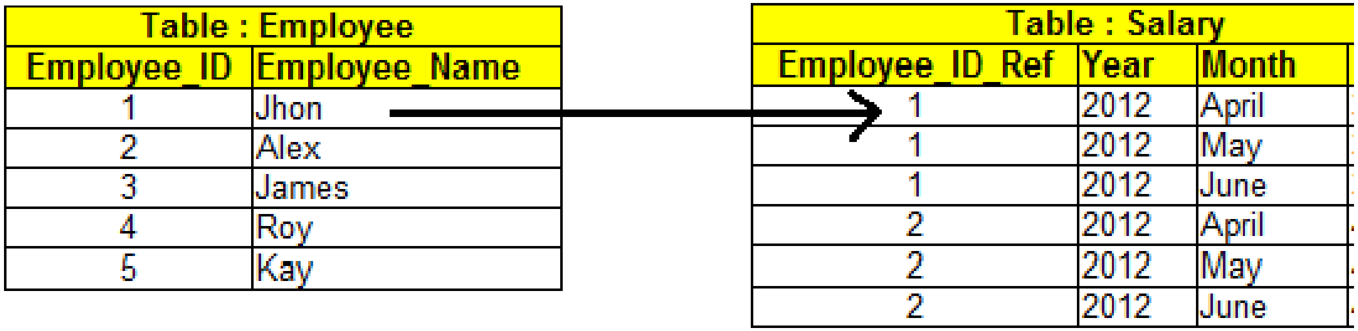


Table : Employee

Employee_ID	Employee_Name
1	Jhon
2	Alex
3	James
4	Roy
5	Kay

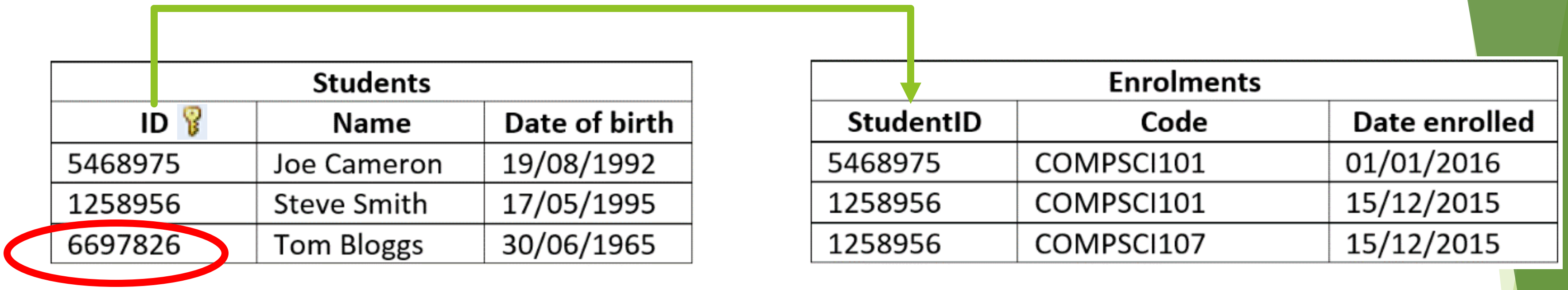
Table : Salary

Employee_ID_Ref	Year	Month
1	2012	April
1	2012	May
1	2012	June
2	2012	April
2	2012	May
2	2012	June

2. Referential integrity

- An important concept underlying relationships between tables
- Referential integrity requires all values of a foreign key field to be:
  - Present in the related primary key field, OR
  - Null (ie. blank)
- Helps to ensure the data in the primary key and foreign key is valid and consistent

Referential integrity



Students		
ID 🔑	Name	Date of birth
5468975	Joe Cameron	19/08/1992
1258956	Steve Smith	17/05/1995
6697826	Tom Bloggs	30/06/1965

Enrolments		
StudentID	Code	Date enrolled
5468975	COMPSCI101	01/01/2016
1258956	COMPSCI101	15/12/2015
1258956	COMPSCI107	15/12/2015

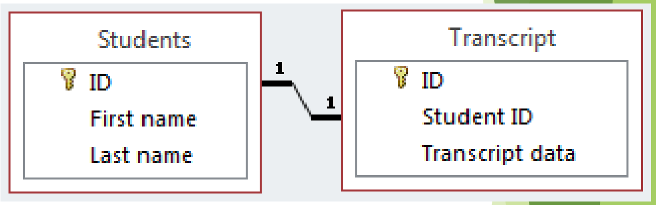
Insert 9998881, COMPSCI111, 22/12/2016 *into* Enrolments ✗

Insert 6697826, COMPSCI105, 16/12/2016 *into* Enrolments ✓

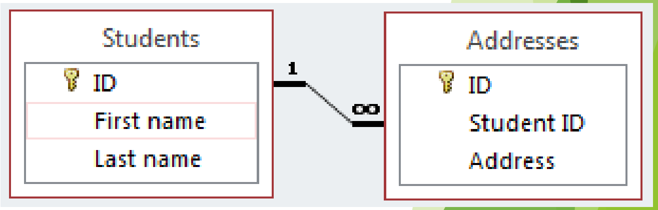
Types of relationships

- There are three kinds of relationship that can exist between tables

- **One to one:** one record in PK related to one record in FK
  - Eg. student can only have one transcript

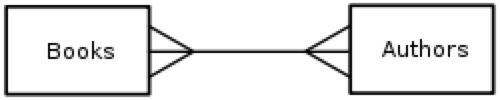


- **One to many:** one record in PK related to multiple records in FK
  - Eg. student can have multiple addresses

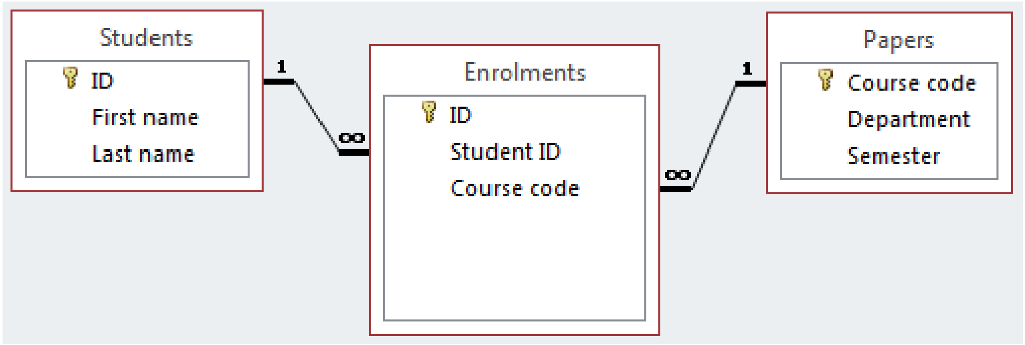


Types of relationships

- **Many to many:** multiple records in PK related to multiple records in FK



  - E.g. An Author can write several Books, and a Book can be written by several Authors
  - E.g. <u>**many**</u> students can be enrolled in <u>**many**</u> papers
- The many-to-many relationships are usually implemented by a pair of one-to-many relationships


Students
- ID
- First name
- Last name

Enrolments
- ID
- Student ID
- Course code

Papers
- Course code
- Department
- Semester

Aspects of a database

- Before we can create our database, we need to decide how to:
  1. **Organize** data in our database
     - Models, tables, relationships
  2. **Enter** data in our database
  3. **Retrieve** data from our database
  4. **Present** the retrieved data to the user

3. Database Management System (DBMS)

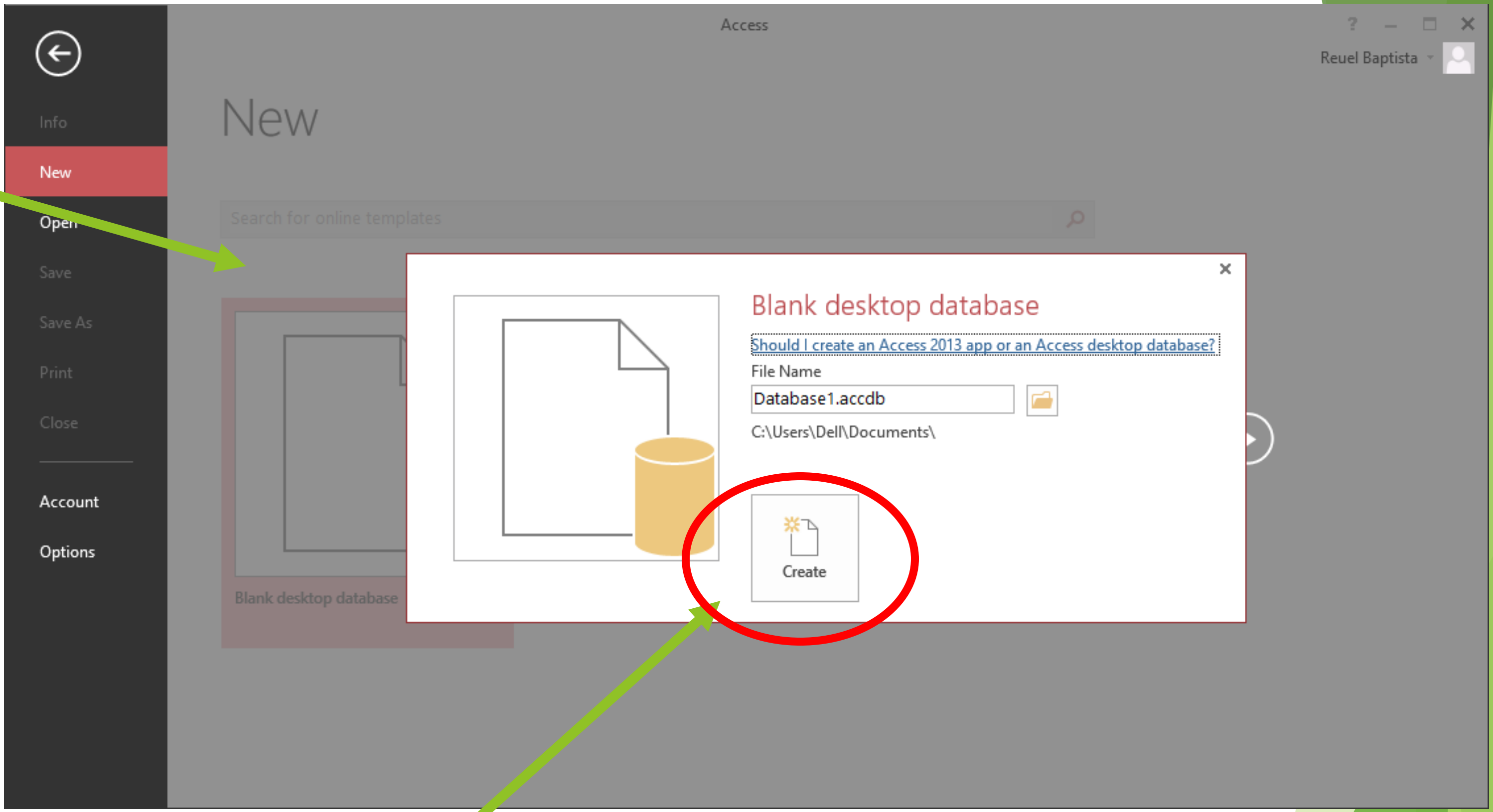
- Application software that is used to manage databases.

- Four main functions:
  - Definition
  - Update
  - Querying
  - Administration

- Examples:
  - Microsoft Access
  - Microsoft SQL Server
  - MySQL

Creating a database

Templates



Blank desktop database

Should I create an Access 2013 app or an Access desktop database?

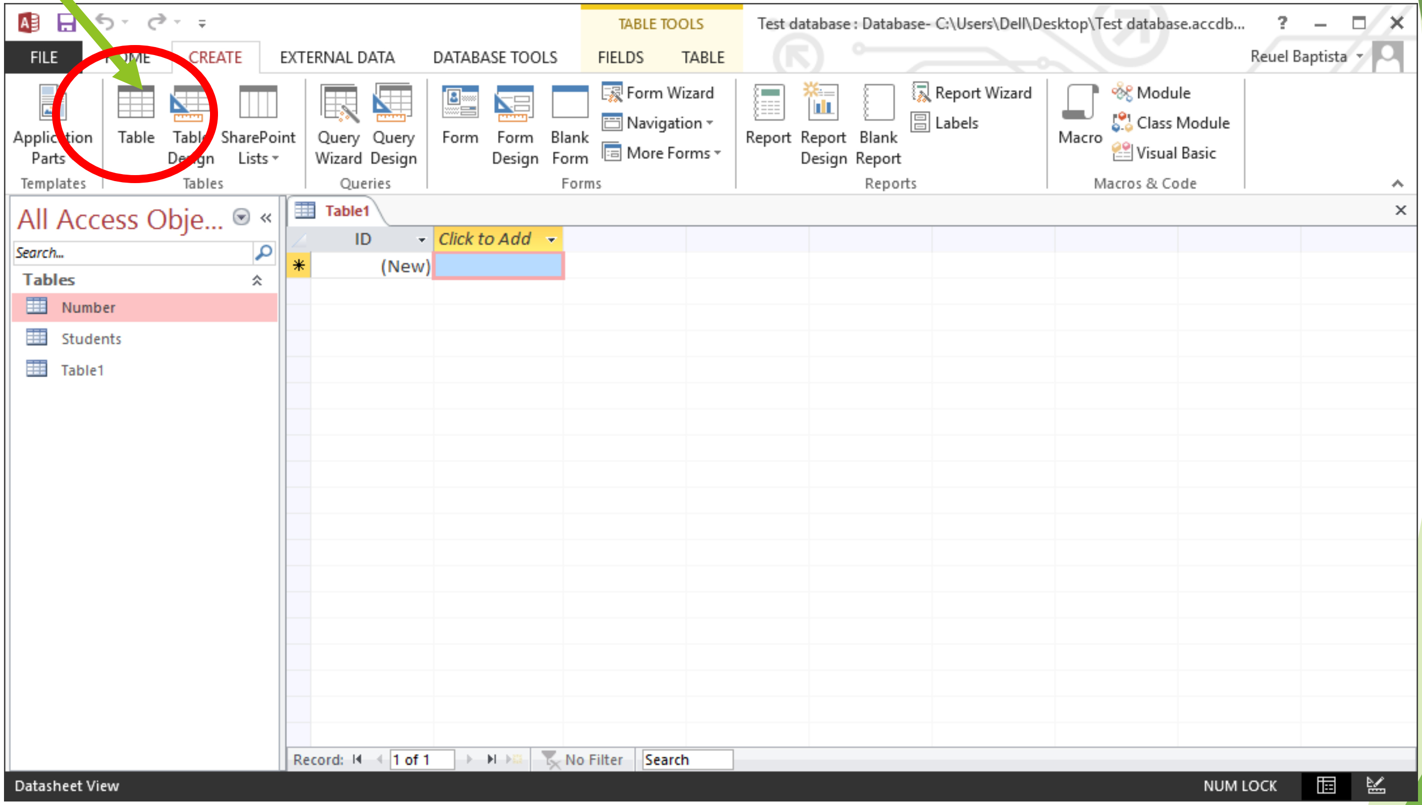
File Name: Database1.accdb

C:\Users\Dell\Documents\

Creating a new database

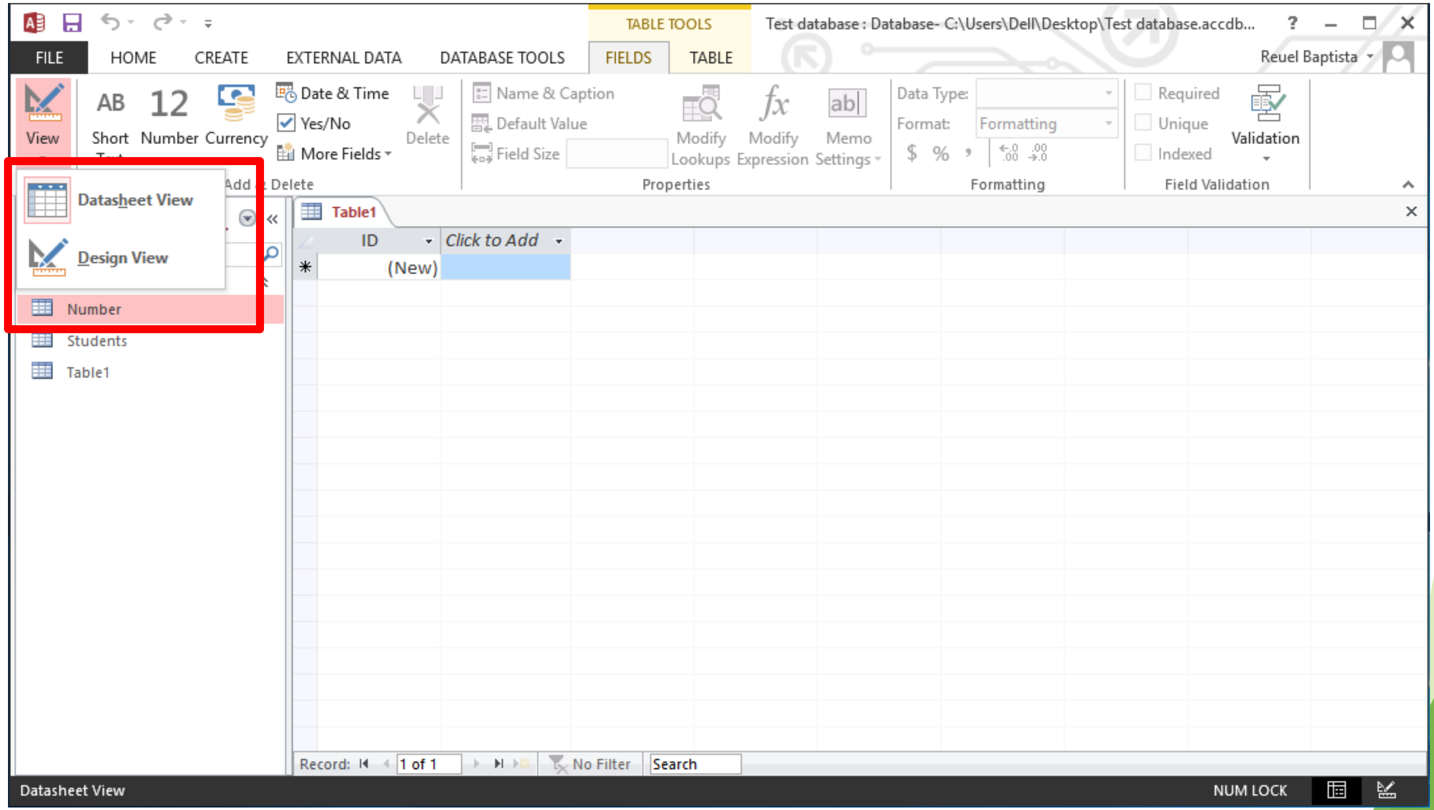
Creating a table

New table button

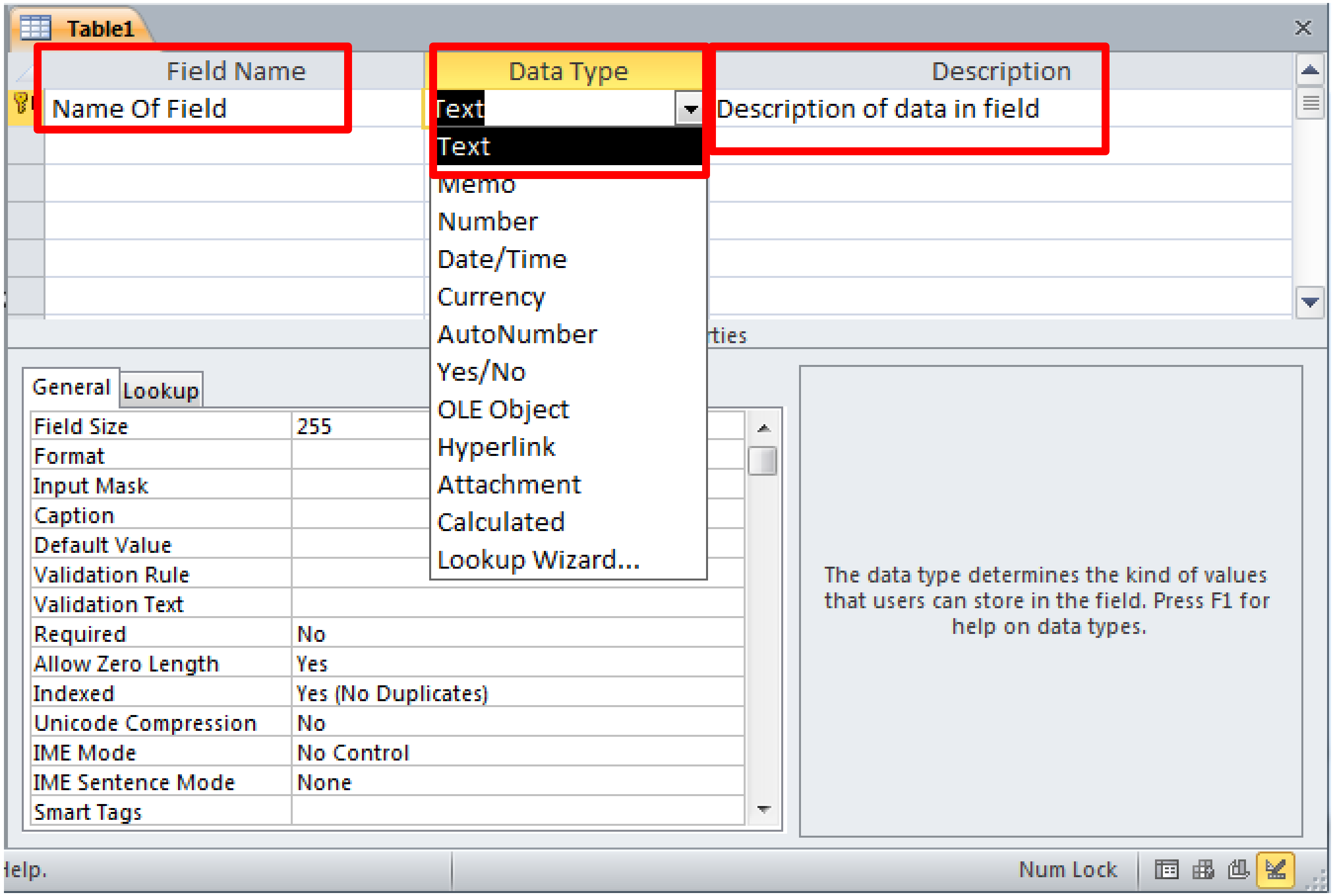


Creating a table

- **Design view:** create/view the fields in the table
- **Datasheet view:** create/view data in the table



Design view



Field Name	Data Type	Description
Name Of Field	Text	Description of data in field

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

The data type determines the kind of values that users can store in the field. Press F1 for help on data types.

Design view

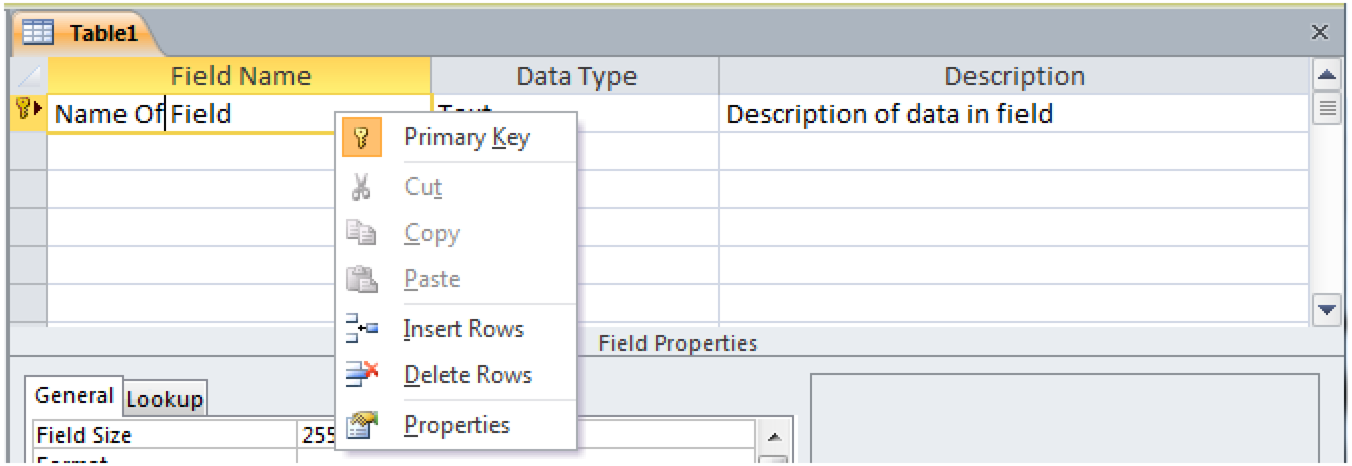


Table1

Field Name	Data Type	Description
Name Of Field	Text	Description of data in field

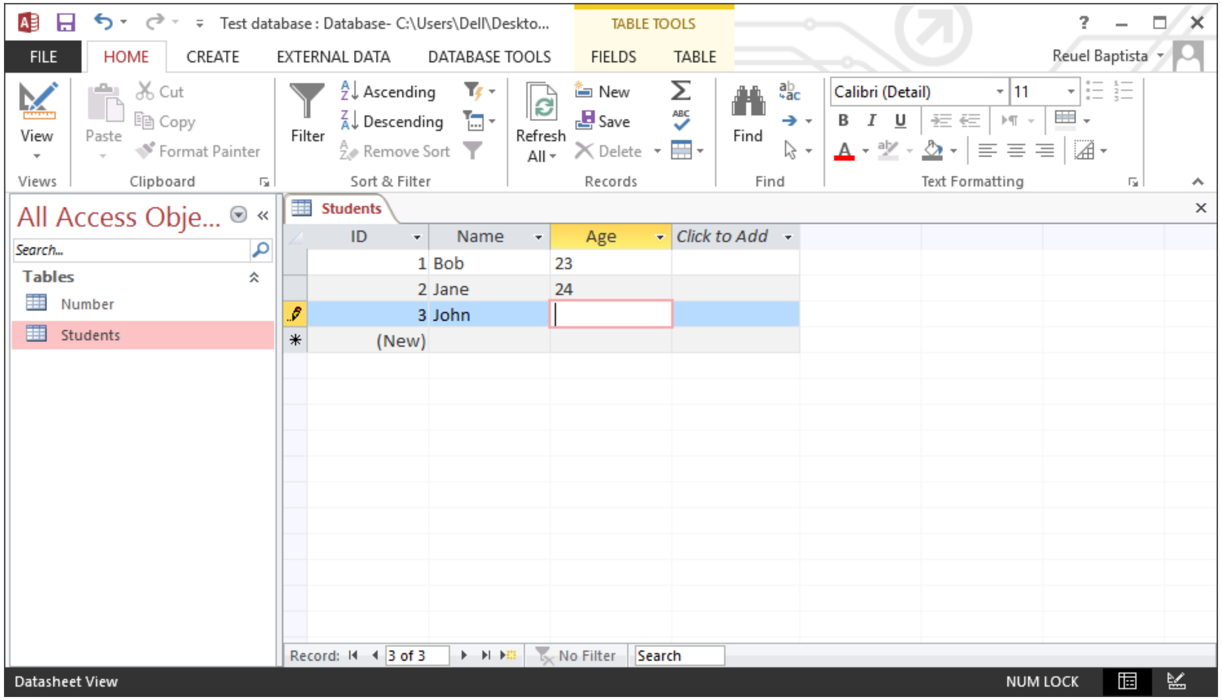
Field Properties

General | Lookup

Field Size 255

Datasheet view

- Allows us to enter data into our table
- Need to ensure that we enter the correct type of data in each field (eg. no text in a number field)

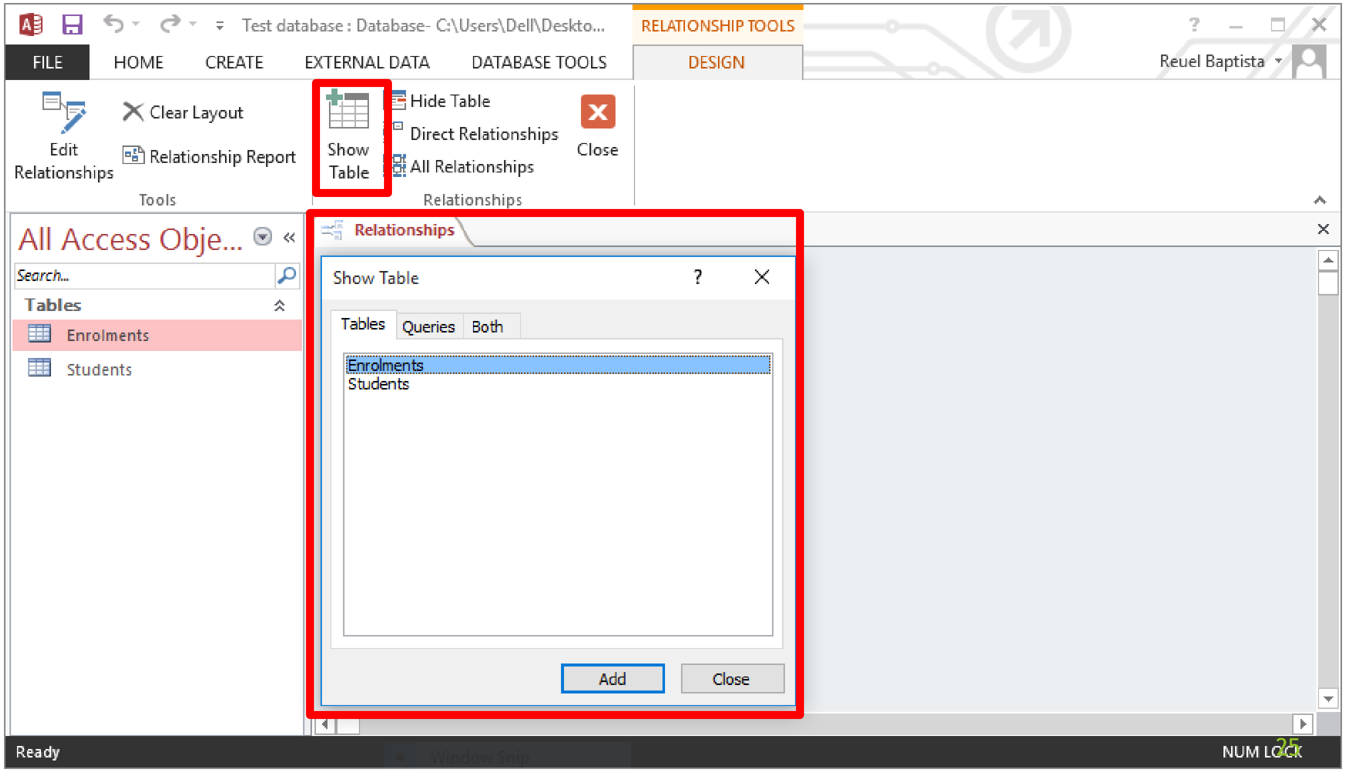


ID	Name	Age	Click to Add
1	Bob	23	
2	Jane	24	
3	John		
(New)			

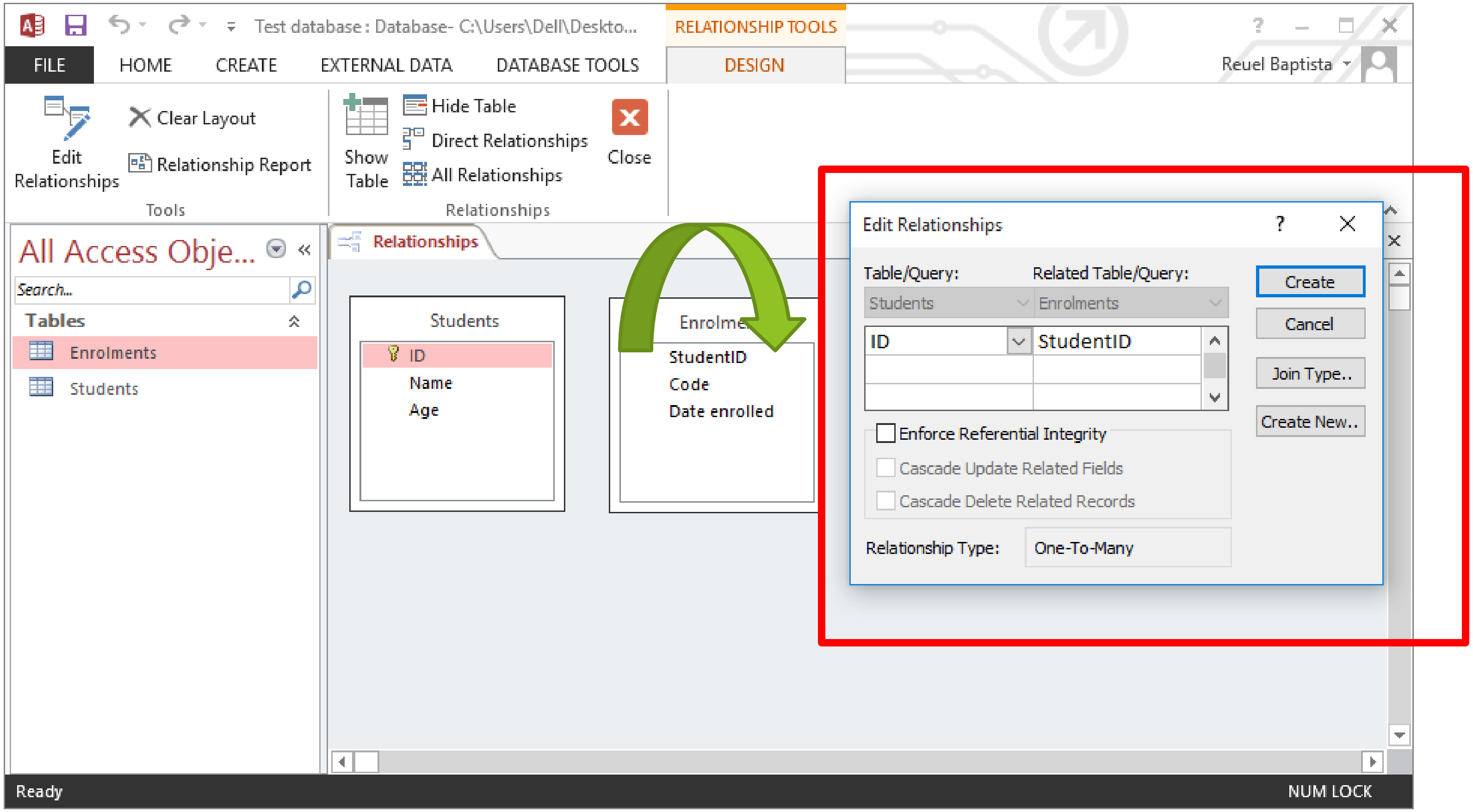
Record: 3 of 3 No Filter Search

Creating relationships

- Relationships view allows us to create relationships between fields in different tables
- Database Tools tab → Relationships button



Creating relationships



Edit Relationships

Table/Query:	Related Table/Query:
Students	Enrolments
ID	StudentID

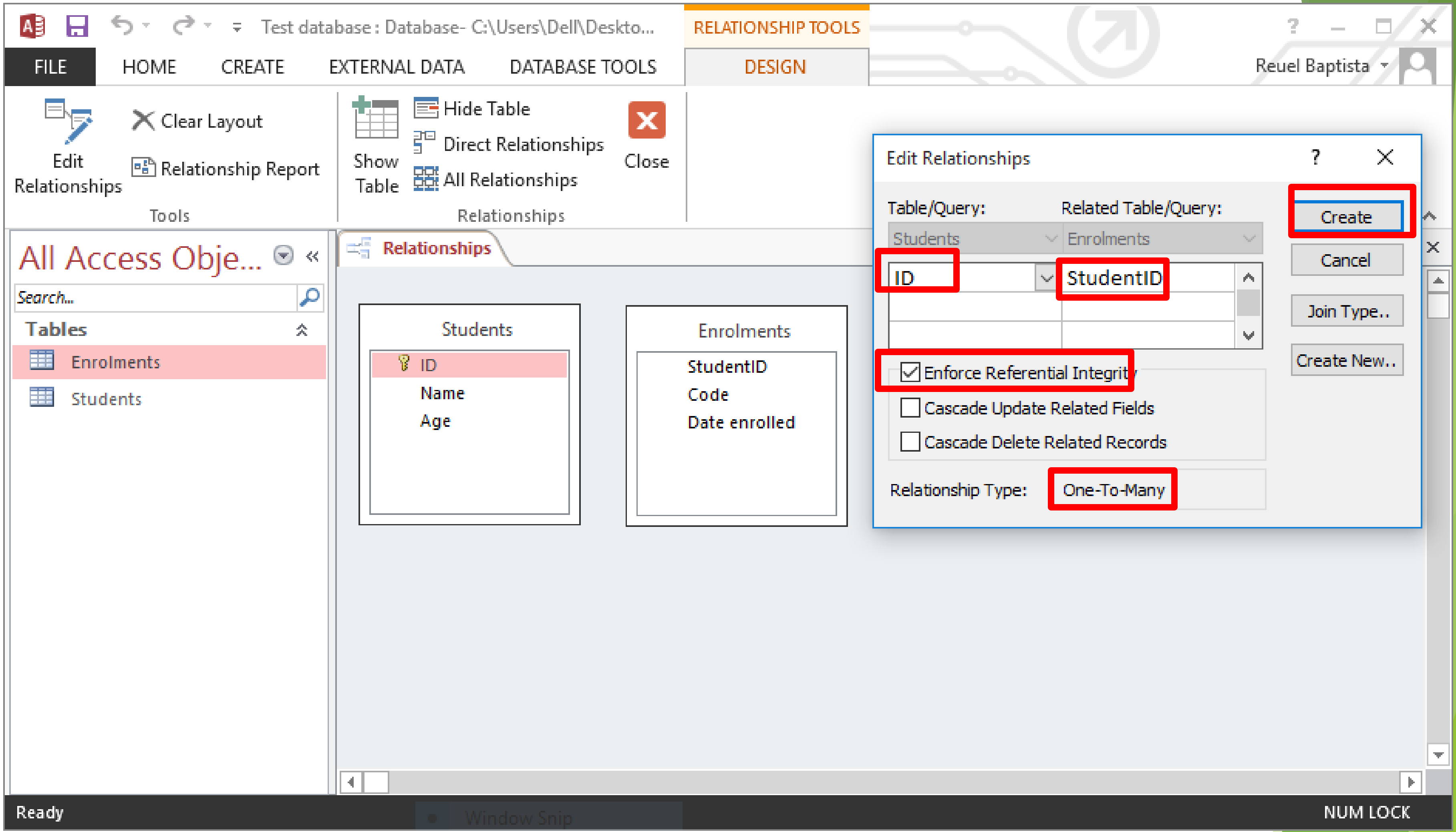
□ Enforce Referential Integrity

□ Cascade Update Related Fields

□ Cascade Delete Related Records

Relationship Type: One-To-Many

Creating relationships



Edit Relationships

Table/Query:	Related Table/Query:
Students	Enrolments
ID	StudentID

☑ Enforce Referential Integrity

□ Cascade Update Related Fields

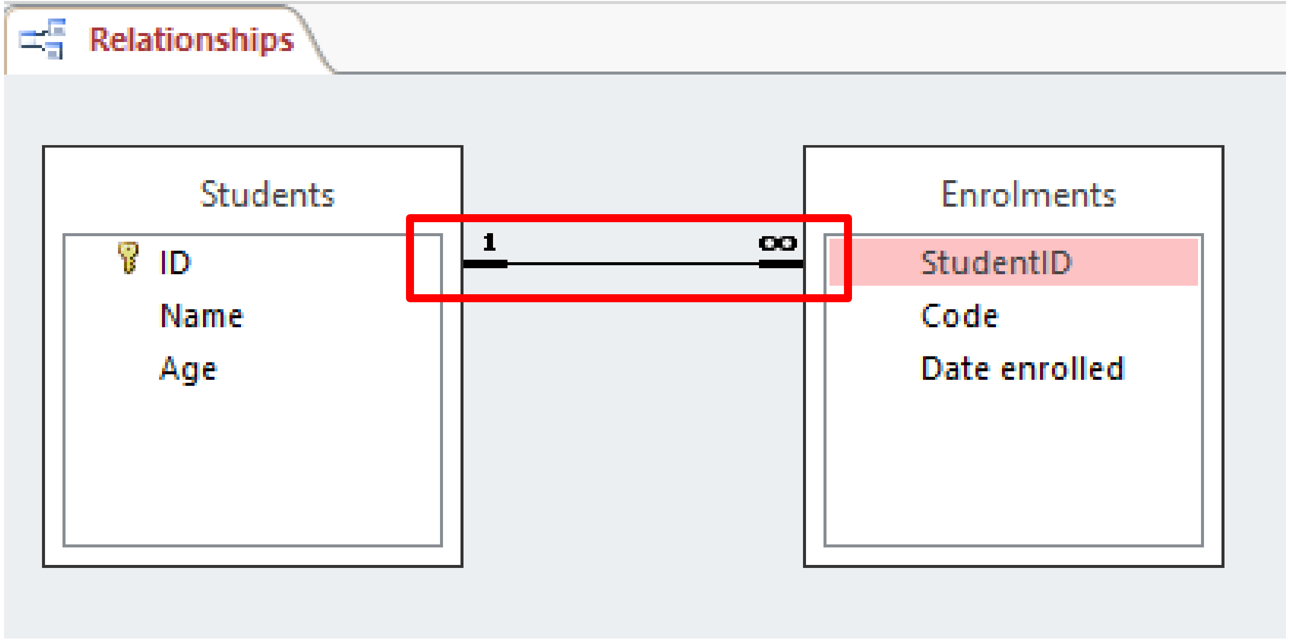
□ Cascade Delete Related Records

Relationship Type: One-To-Many

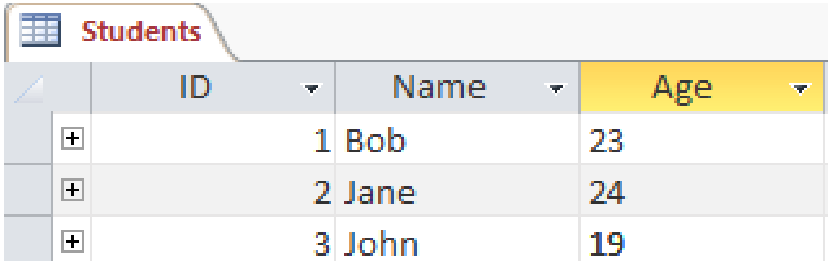
Students: ID, Name, Age

Enrolments: StudentID, Code, Date enrolled

Creating relationships

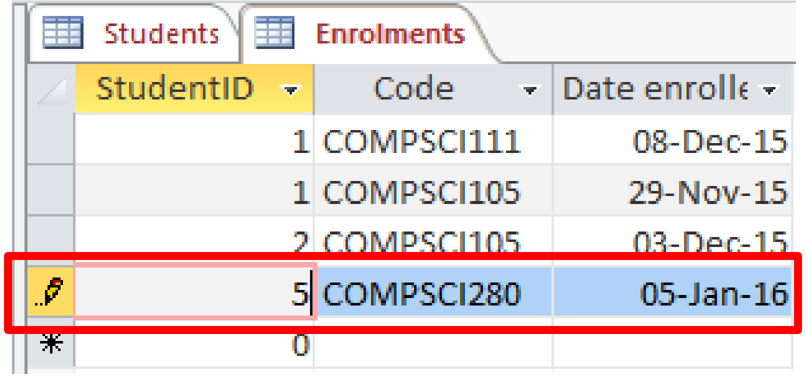


Inserting data



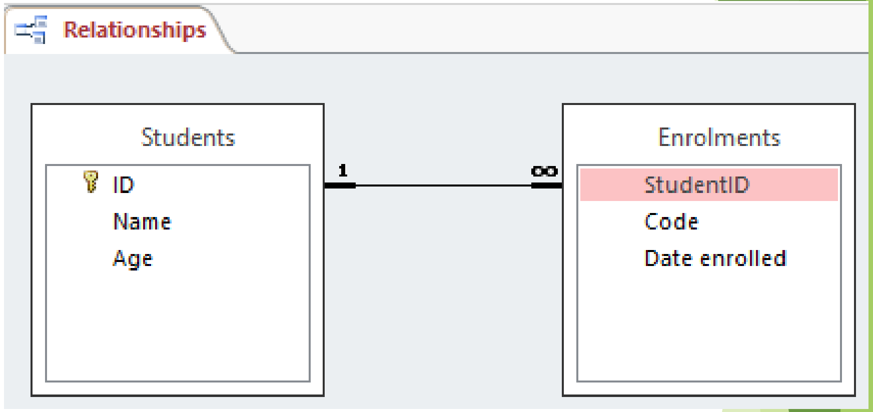
Students

ID	Name	Age
1	Bob	23
2	Jane	24
3	John	19



Students | Enrolments

StudentID	Code	Date enrolle
1	COMPSCI111	08-Dec-15
1	COMPSCI105	29-Nov-15
2	COMPSCI105	03-Dec-15
5	COMPSCI280	05-Jan-16
0		



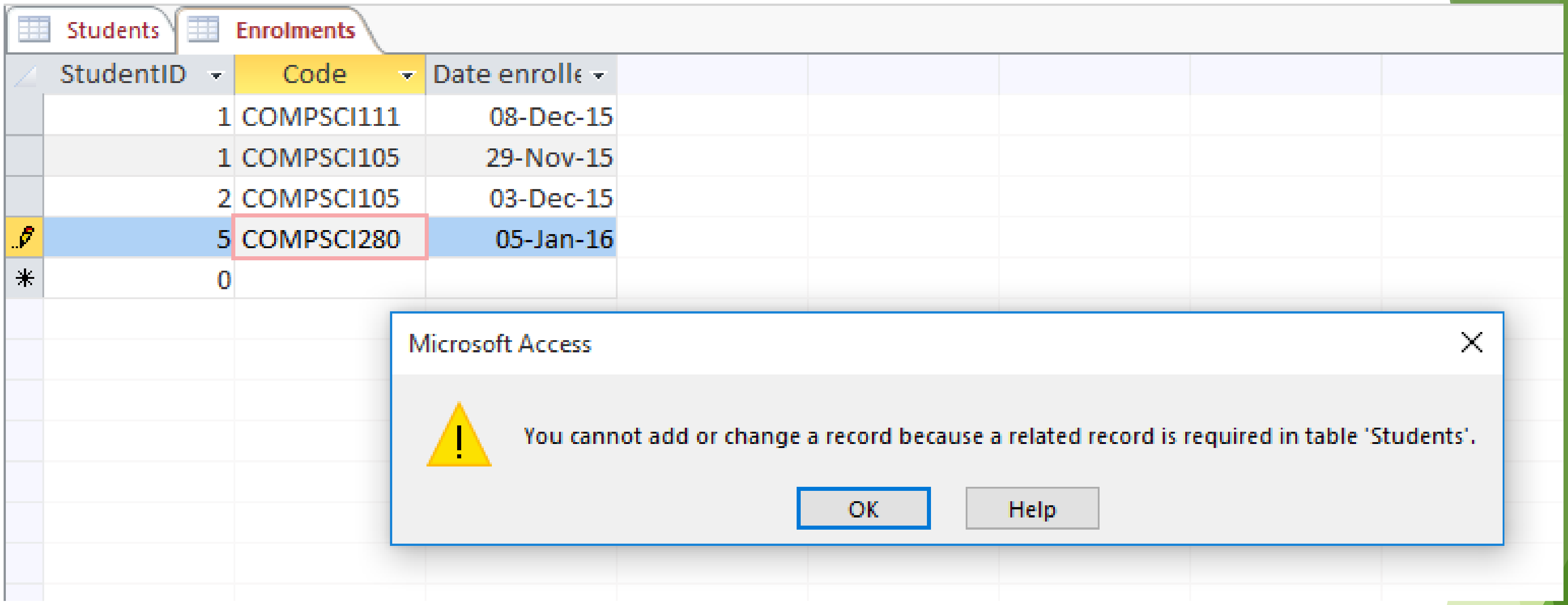
Relationships

Students: ID, Name, Age

Enrolments: StudentID, Code, Date enrolled

- Can we insert this record in the Enrolments table?

Inserting data

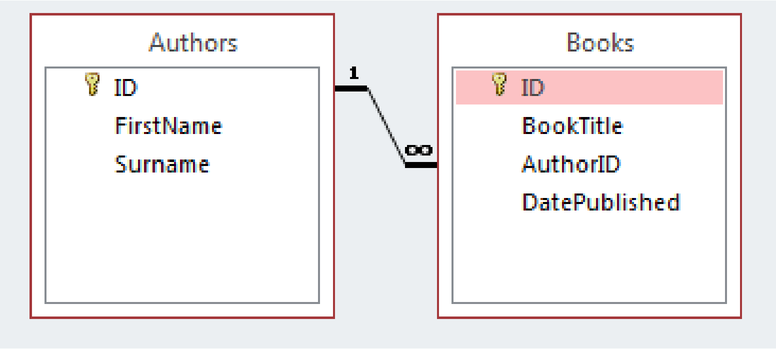


StudentID	Code	Date enrolle
1	COMPSCI111	08-Dec-15
1	COMPSCI105	29-Nov-15
2	COMPSCI105	03-Dec-15
5	COMPSCI280	05-Jan-16
0		

You cannot add or change a record because a related record is required in table 'Students'.

- This won't work; `StudentID`'s value ('5') doesn't exist in the primary key `ID`

Exercise 3



Authors

ID	FirstName	Surname
1	Sarah	Buchman
2	Wendy	Heydemark
3	Hallie	Hull

Books

ID	BookTitle	AuthorID
1	200 Years of German Humor	1
2	Ask Your System Administator	2
3	How about Never?	1

1. What is the primary key … ?
2. What is the primary key and foreign key … ?

Summary

- A database is used to store information in a systematic and orderly manner
- The relational model uses tables to store information about entities and relationships to connect tables together
- Relationships require tables, primary keys, foreign keys
- Referential integrity helps to maintain consistency in our database
- Microsoft Access is a popular DBMS that we can use to insert and manage data in our database