

Databases 1 – Organisation and Creation

Lecture 21 – COMPSCI111/111G SS 2018



Resources



- ▶ Pre-lecture reading:
 - ▶ Read material/lecture-handout prior to your lecture
- ▶ Post-lecture Quizzes: a set of multiple choice questions after each lecture
 - ▶ Go to <https://coderunner2.auckland.ac.nz>
 - ▶ Allow students to revise material after each lecture
- ▶ In-class Exercise
 - ▶ One question/exercise during lecture (discuss or work in groups)
- ▶ Kahoot
 - ▶ On their personal devices, players can then join by going to *kahoot.it* in their web browser (on install the kahoot app to your own device) , and entering the pin displayed on the screen at the front of the room
 - ▶ They then enter their nickname, seeing it displayed at the front
 - ▶ They then use their device to answer each question, with the aim to get as many points as possible and get to the top of the leaderboard



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: 10036
 LastName: Lancaster
 First Name: Jessica
 Called:
 Class: 0
 Line: 18
 Birth Date: 4/15/1982
 Sex: M
 Note:

Current Grade Averages:

Reading	Spelling	English	Bonus
81	89	85	0

Scores

Note	Gr.	Score	LetterC	A/T/D	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 definition	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/

Property Slides Details

Property Name: Long Island
 Property Unit:
 Address: Cincinnati
 Town: OH
 County: US
 Size: 340
 Status: Closed
 Start Value: \$34,000

Last Bid Details

Property ID: 41
 Bid Time: 31-Jan-16 6:21:09 PM
 Property ID: 5
 Highest Bid: \$140,000.00
 Unit: 1
 Total: \$140,000.00

Corporate Sales Overview

Select View: Corporate Regional City

Corporate Performance (YTD)

Corporate Revenue Regional Performance Y/Y

Revenue in millions of dollars: \$27,490,580

Regional Performance (YTD)

Line chart showing regional performance from Jun 2006 to Dec 2006. Legend includes Central, Mid-Atlantic, Northeast, Northwest, South, Southeast, Southwest, and Web.

Category Analysis (YTD)

Units Sold: Books 22.04%, Movies 6.87%, Music 6.88%, Electronics 17.46%

Subcategory Analysis (YTD)

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**

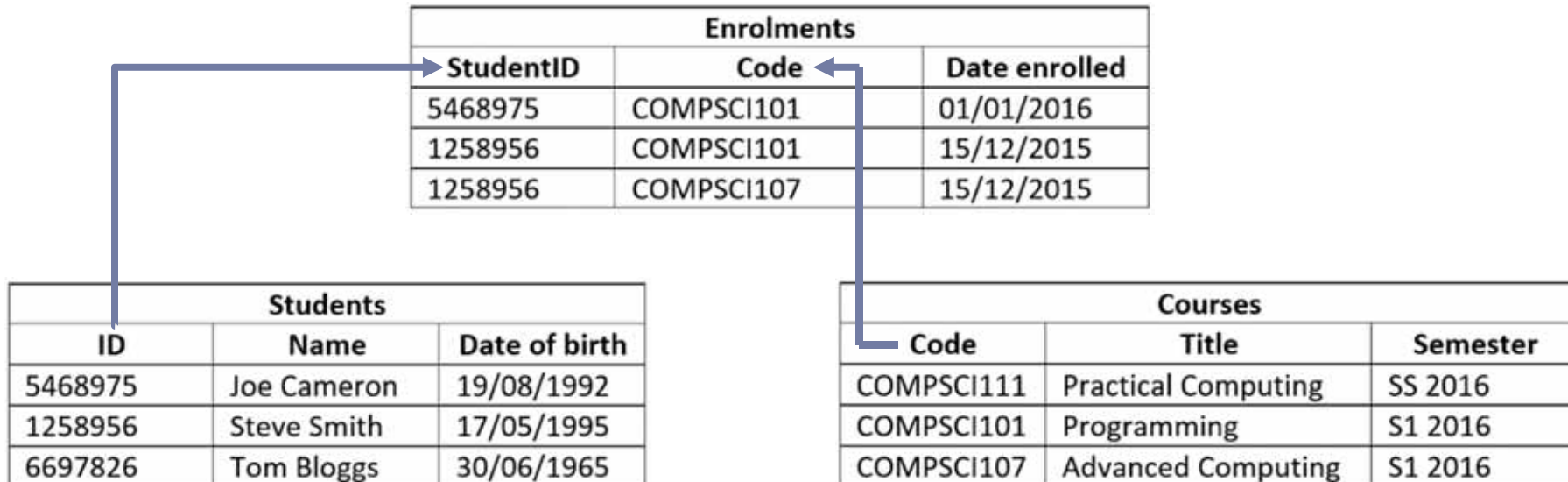
			Field
StudentId	Name	Address	Phone
12345	C. Brown	12 Apple St.	555-1234
67890	L. Van Pelt	34 Pear Ave.	555-5678
22222	P. Patty	56 Grape Blvd.	555-9999

Record



Organising data

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





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

Primary key →

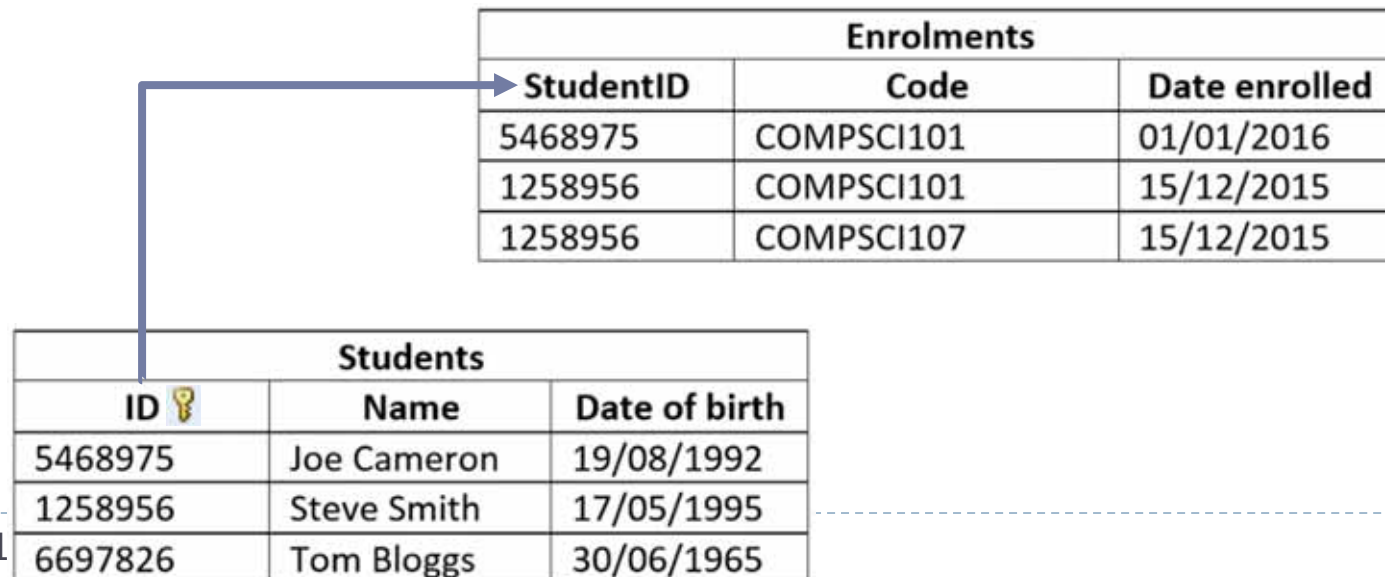
	Field Name	Data Type
🔑	ID	AutoNumber
	First Name	Text
	E-mail Address	Text
	Level	Text
	Room	Text
	Date of Birth	Date/Time



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



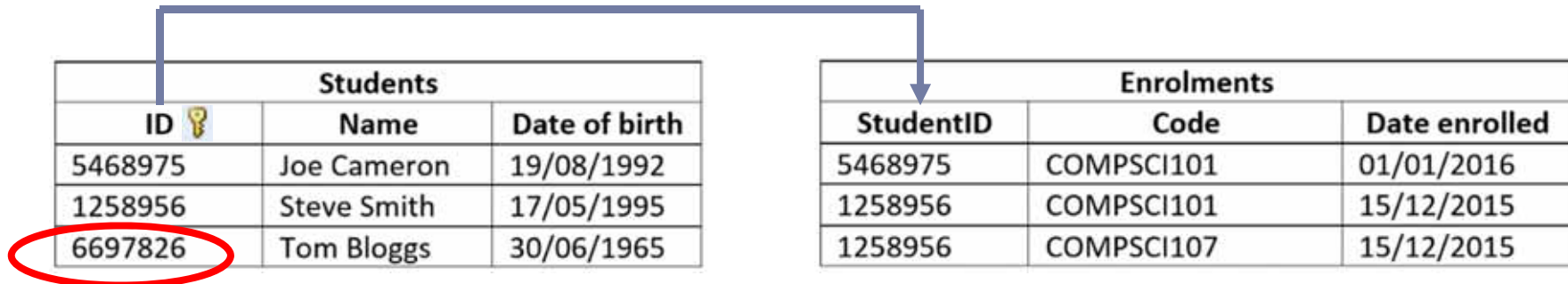


2. Referential integrity

- ▶ 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



Insert 9998881, COMPSCI111, 22/12/2016 into Enrolments ✘

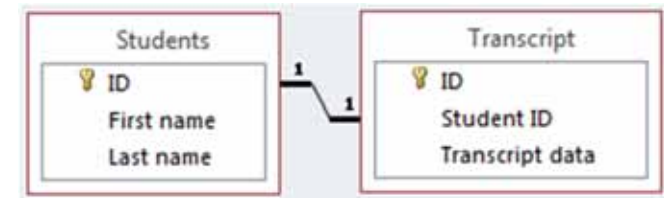
Insert 6697826, COMPSCI105, 16/12/2016 into Enrolments ✔



Types of relationships

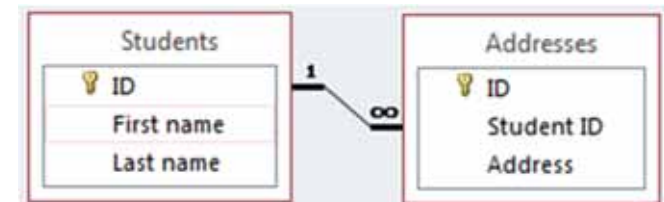
- ▶ There are three kinds of relationship that can exist between tables
- ▶ **One to one:** one record in PK related to one record in FK

- ▶ Eg. student can only have one transcript



- ▶ **One to many:** one record in PK related to multiple records in FK

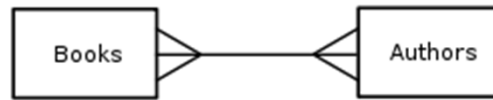
- ▶ Eg. student can have multiple addresses



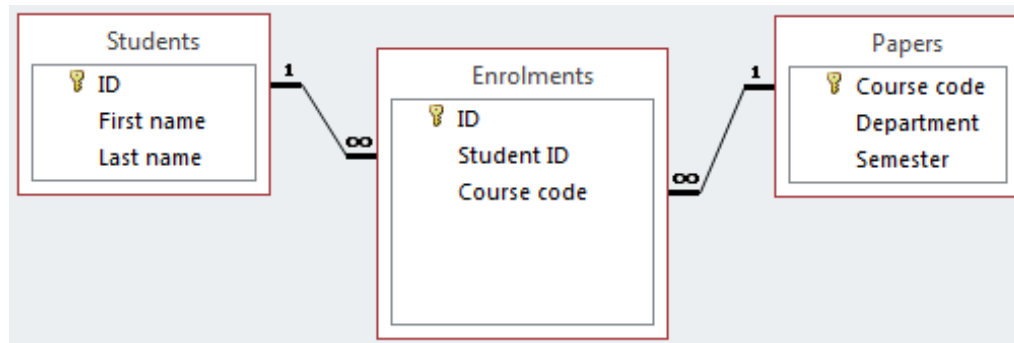


Types of relationships

- ▶ **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. **many** students can be enrolled in **many** papers
- ▶ The many-to-many relationships are usually implemented by a pair of one-to-many relationships





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

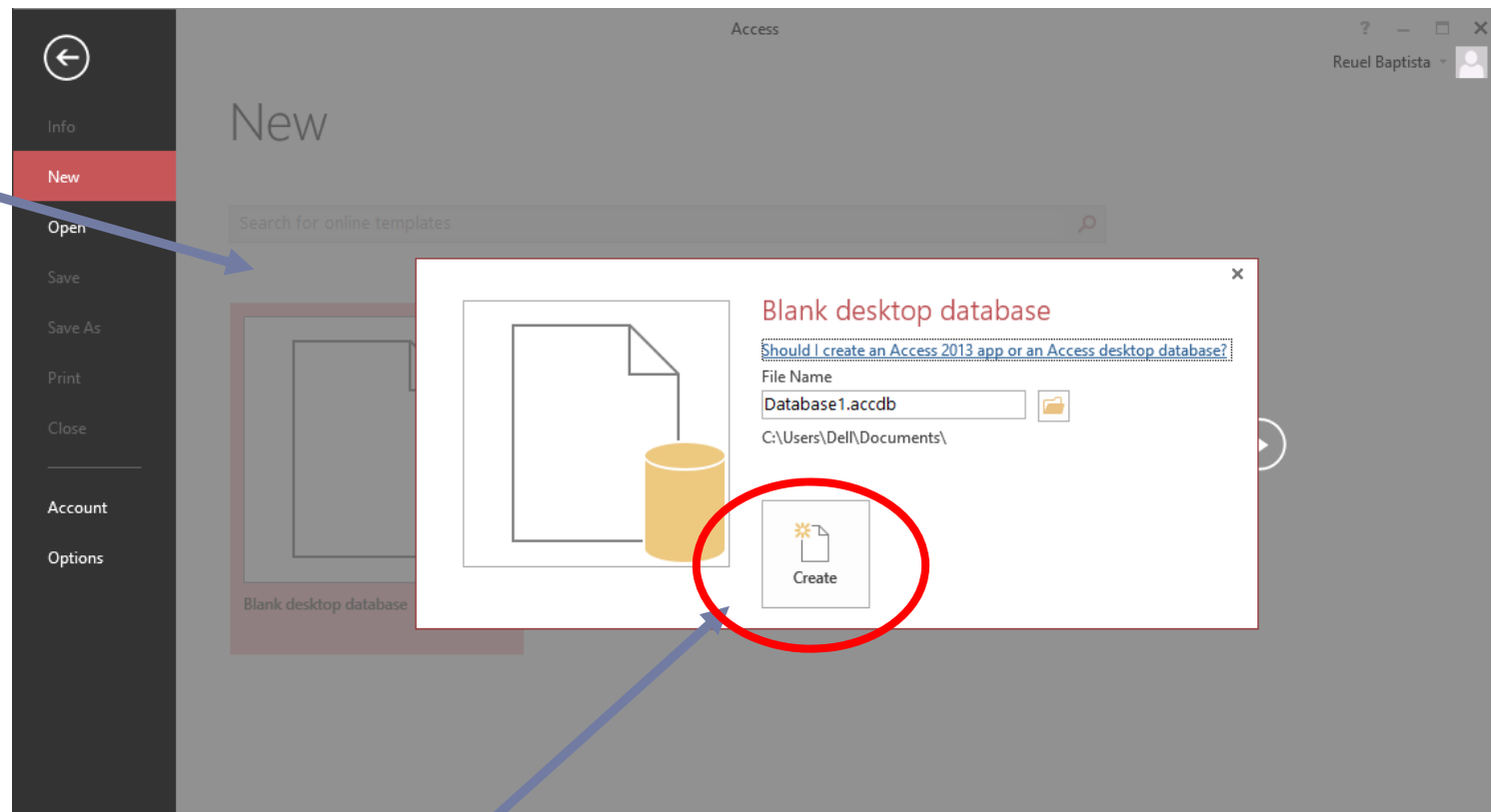




Creating a database



Templates

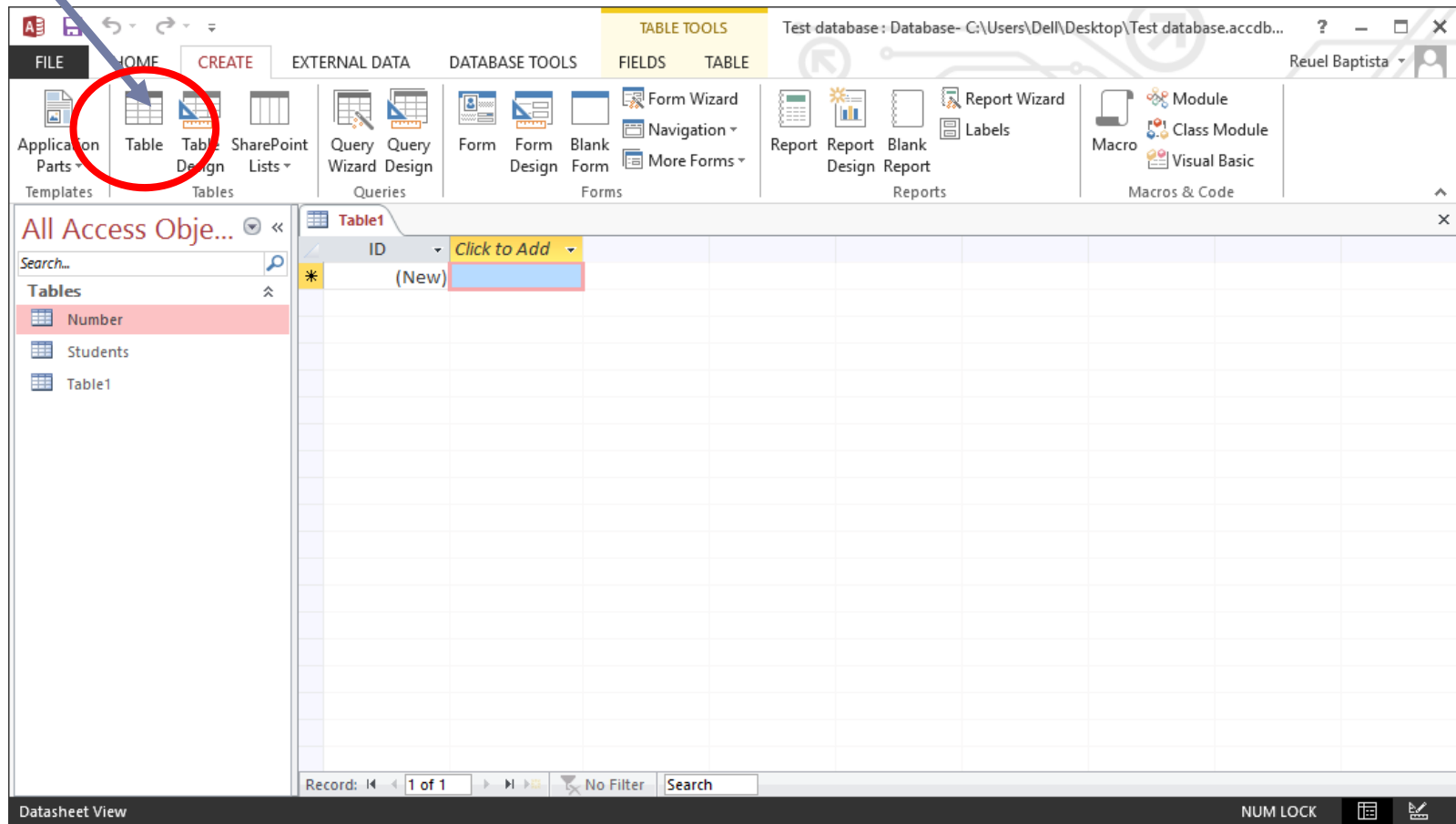


Creating a new database



Creating a table

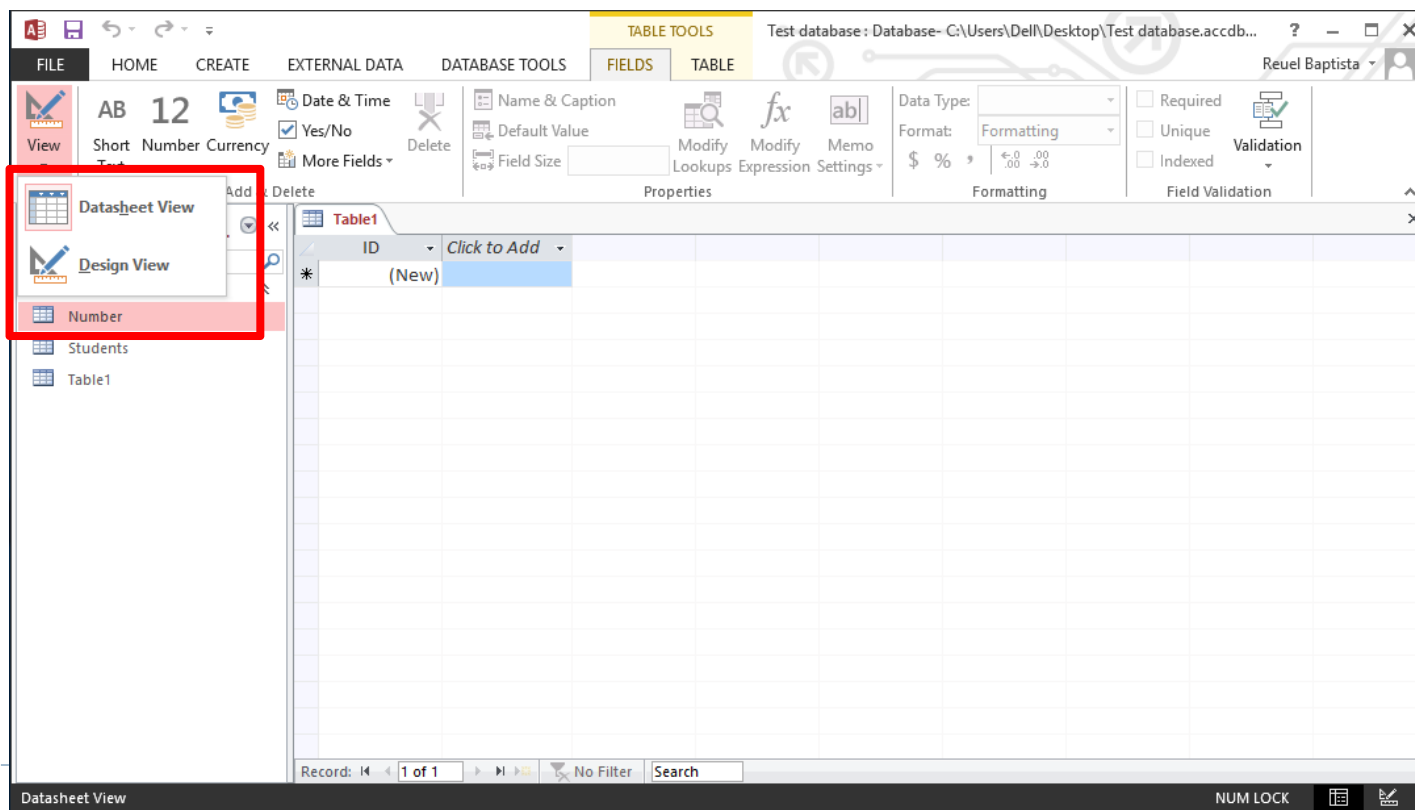
New table button





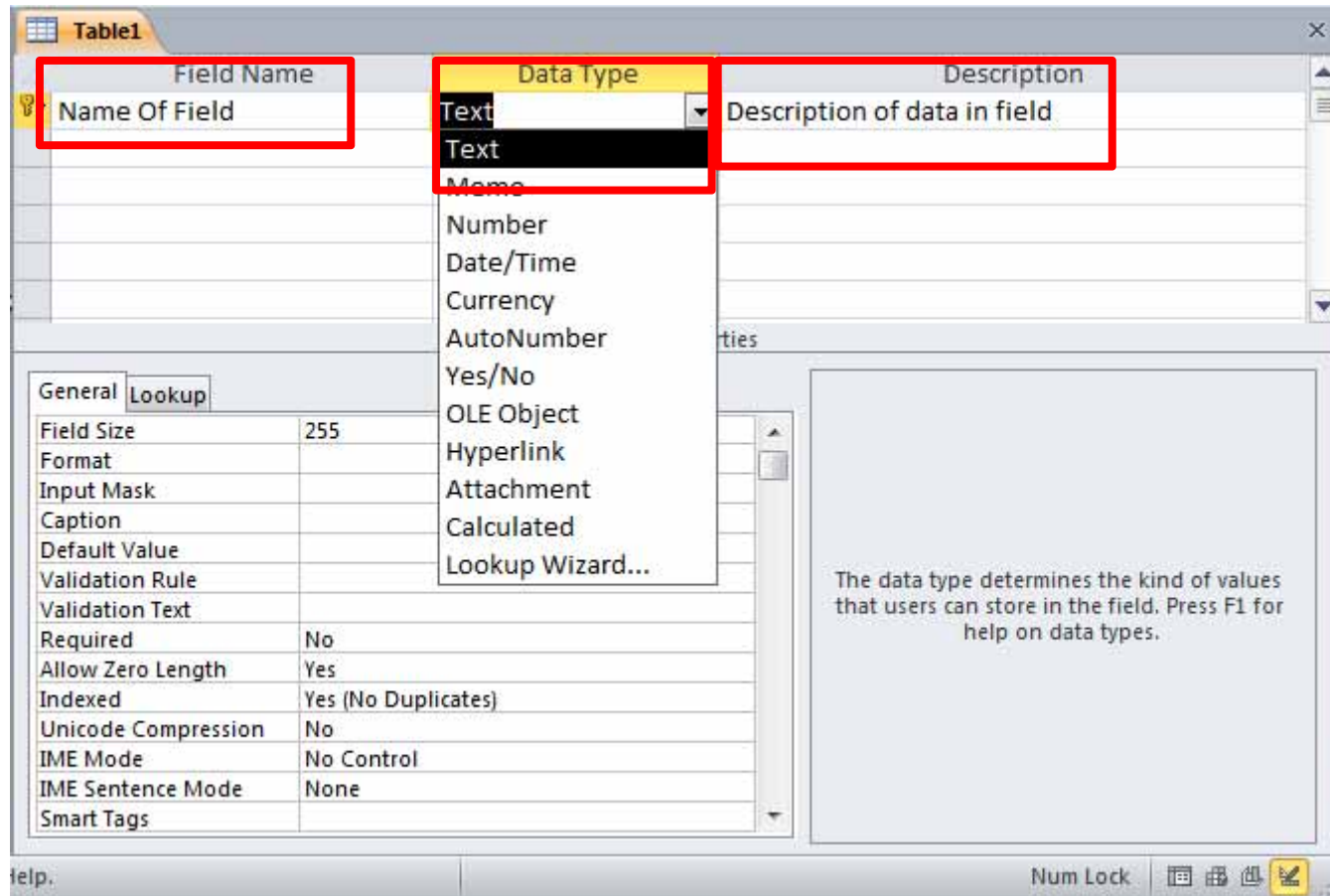
Creating a table

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



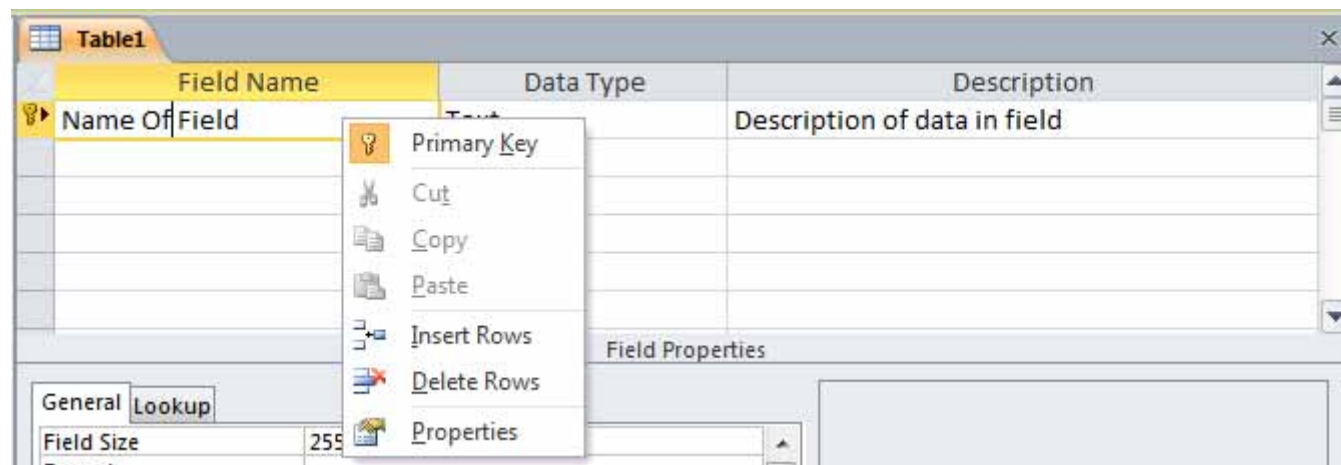


Design view





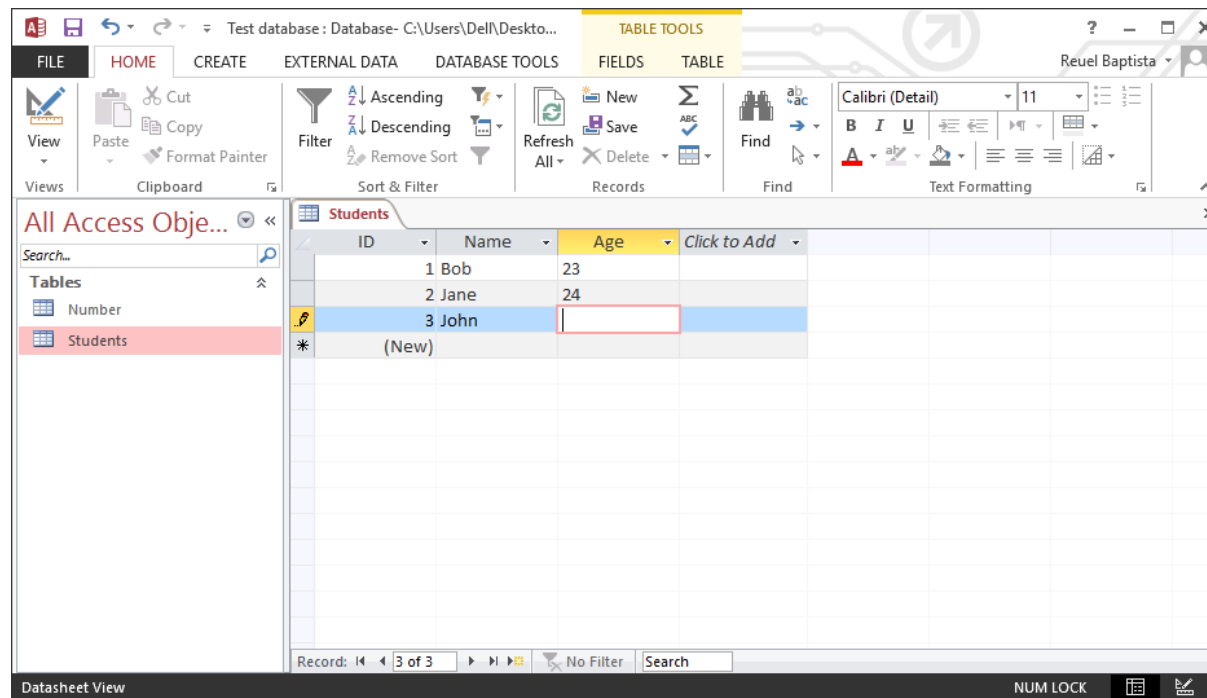
Design view





Datasheet view

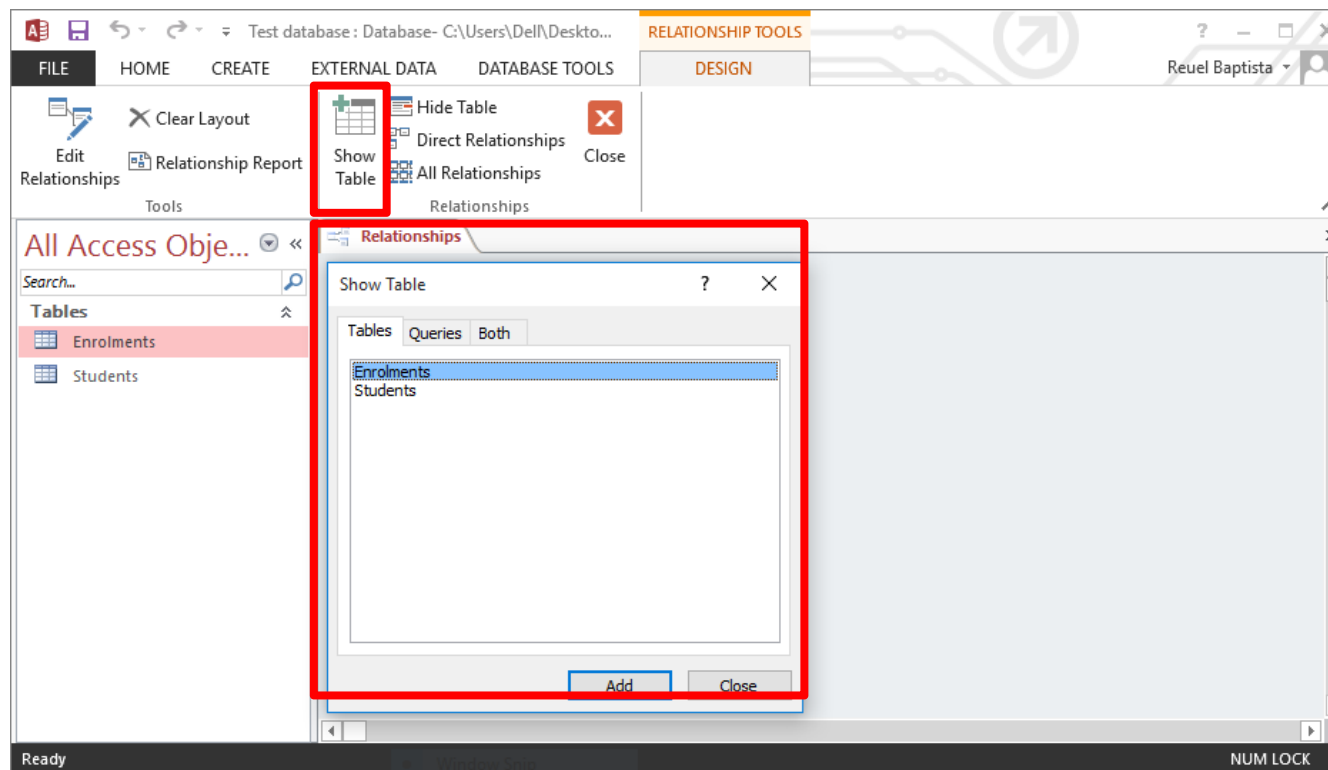
- ▶ Allows us to enter data into our table
- ▶ Need to ensure that we enter the correct type of data in each field (eg. no text in a number field)





Creating relationships

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





Creating relationships

The screenshot shows the Microsoft Access interface. The main window displays the Relationships window with two tables: 'Students' and 'Enrolments'. The 'Students' table has fields 'ID', 'Name', and 'Age'. The 'Enrolments' table has fields 'StudentID', 'Code', and 'Date enrolled'. A blue arrow points from the 'ID' field in 'Students' to the 'StudentID' field in 'Enrolments'. The 'Edit Relationships' dialog box is open, showing the relationship between 'Students' and 'Enrolments' with 'ID' as the primary key and 'StudentID' as the foreign key. The relationship type is set to 'One-To-Many'. The dialog box includes options for 'Enforce Referential Integrity', 'Cascade Update Related Fields', and 'Cascade Delete Related Records'. The 'Create' button is highlighted.

Ready NUM LOCK



Creating relationships

The screenshot displays the Microsoft Access interface in the Relationships view. The main window shows two tables: 'Students' with fields 'ID' (primary key), 'Name', and 'Age'; and 'Enrolments' with fields 'StudentID', 'Code', and 'Date enrolled'. An 'Edit Relationships' dialog box is open, showing a relationship between 'Students' and 'Enrolments'. The 'ID' field from 'Students' is linked to the 'StudentID' field in 'Enrolments'. The 'Enforce Referential Integrity' checkbox is checked, and the 'Relationship Type' is set to 'One-To-Many'. The 'Create' button in the dialog is highlighted with a red box.

Microsoft Access - Test database: Database- C:\Users\Dell\Desktop...

RELATIONSHIP TOOLS

DESIGN

Reuel Baptista

FILE HOME CREATE EXTERNAL DATA DATABASE TOOLS

Clear Layout Hide Table Direct Relationships Close

Show Table All Relationships

Relationships

All Access Objects

Search...

Tables

- Enrolments
- Students

Students

- ID
- Name
- Age

Enrolments

- StudentID
- Code
- Date enrolled

Edit Relationships

Table/Query: Students Related Table/Query: Enrolments

ID StudentID

Enforce Referential Integrity

Cascade Update Related Fields

Cascade Delete Related Records

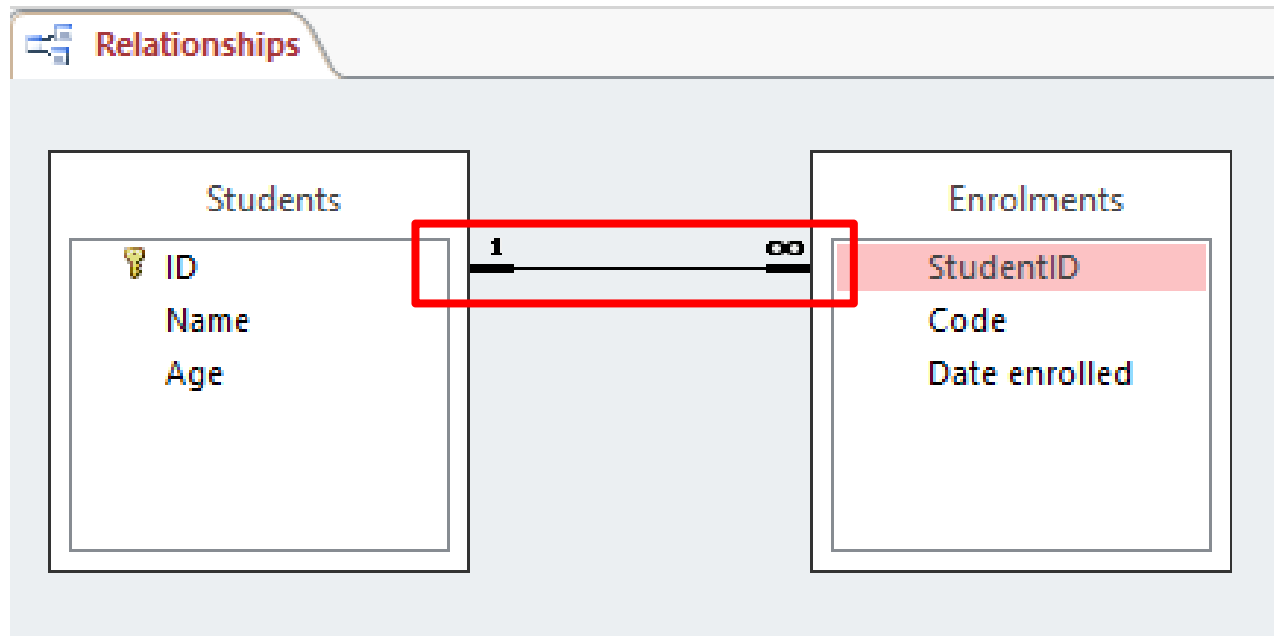
Relationship Type: One-To-Many

Create Cancel Join Type.. Create New..

Ready Window Snip NUM LOCK



Creating relationships

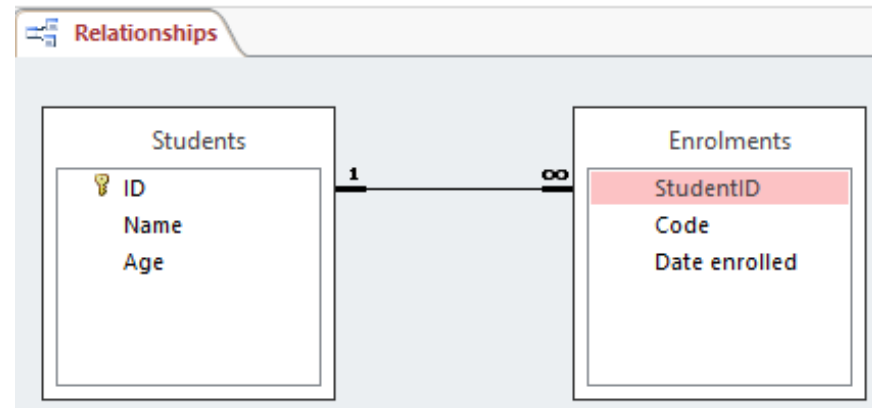




Inserting data

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

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



- ▶ Can we insert this record in the Enrolments table?



Inserting data

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

Microsoft Access

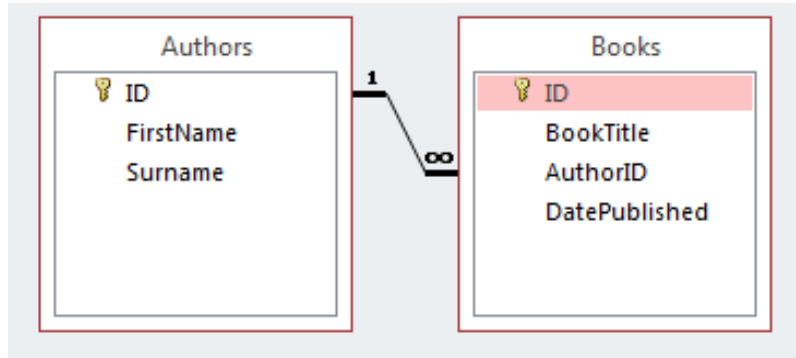
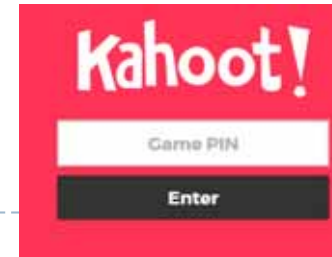
You cannot add or change a record because a related record is required in table 'Students'.

OK Help

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



Exercises



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 Administrator	2	
3	How about Never?	1	

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



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

- ▶ Post-Lecture-Quiz: PLQ_21
 - ▶ <https://coderunner2.auckland.ac.nz/moodle/mod/quiz/view.php?id=627>



Practical in-class Exercise

- ▶ Create a blank database
 - ▶ Start **Access 2016**
 - ▶ Choose the “**Blank desktop database**”
 - ▶ File Name: **Employees.accdb**
 - ▶ Click the **Create** button
 - ▶ Note: The blank database will be presented. The “Tables” tab will already be selected for you.
 - ▶ Choose View->Design View. You will be prompted to save the table. Call it **Departments** and click OK
 - ▶ Add the following fields: DEPT_CODE, DEPARTMENT_NAME
 - ▶ Set the **primary** key
 - ▶ Choose Create->Table. Repeat the above steps.
 - ▶ Table name: **Employees**
 - ▶ Add the following fields:
 - EMPLOYEE_ID, FIRST_NAME, LAST_NAME, DEPT_CODE, HIRE_DATE , CREDIT_LIMIT, PHONE_NUMBER, MANAGER_ID
 - Set the **primary** key, **foreign key** and the **relationship** between them



Practical in-class Exercise

- ▶ Enter the following values:
- ▶ Departments:
 - ▶ 'ACT', 'ACCOUNTING'
 - ▶ 'EXE', 'EXECUTIVE'
 - ▶ 'MKT', 'MARKETING'
 - ▶ 'PER', 'PERSONNEL'
 - ▶ 'SAL', 'SALES'
 - ▶ 'SHP', 'SHIPPING'
- ▶ Employees:
 - ▶ 201, 'SUSAN', 'BROWN', 'EXE', '01-JUN-1998', 30, '3484'
 - ▶ 202, 'JIM', 'KERN', 'SAL', '16-AUG-1999', 25, '8722', 201
 - ▶ 203, 'MARTHA', 'WOODS', 'SHP', '02-FEB-2009', 25, '7591', 201
 - ▶ 204, 'ELLEN', 'OWENS', 'SAL', '01-JUL-2008', 15, '6830', 202
 - ▶ 205, 'HENRY', 'PERKINS', 'SAL', '01-MAR-2006', 25, '5286', 202
 - ▶ 206, 'CAROL', 'ROSE', 'ACT'