



## Databases 1 – Organisation and Creation

Lecture 21 – COMPSCI111/111G SS 2018

1



## 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

2

Show respect for the teacher, yourself and others at all times.



## Today's lecture

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

3



## 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

4



## Examples



5



## 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

6



## 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*

7



## 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**

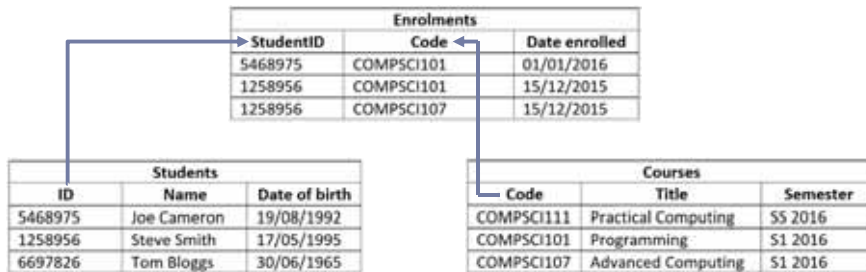
				Field
	StudentId	Name	Address	Phone
	12345	C. Brown	12 Apple St.	555-1234
	67890	L. Van Pelt	34 Pear Ave.	555-5678
Record	22222	P. Patty	56 Grape Blvd.	555-9999

8



## Organising data

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



9



## 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 →

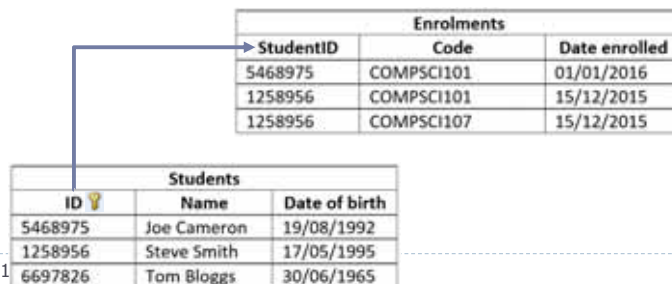
Field Name	Data Type
ID	AutoNumber
First Name	Text
E-mail Address	Text
Level	Text
Room	Text
Date of Birth	Date/Time

10



## 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



11



## 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

12



## Referential integrity

Students			Enrolments		
ID	Name	Date of birth	StudentID	Code	Date enrolled
5468975	Joe Cameron	19/08/1992	5468975	COMPSCI101	01/01/2016
1258956	Steve Smith	17/05/1995	1258956	COMPSCI101	15/12/2015
6697826	Tom Bloggs	30/06/1965	1258956	COMPSCI107	15/12/2015

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

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

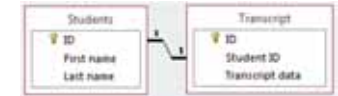


## Types of relationships

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

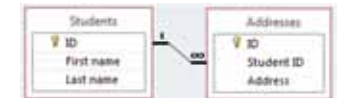
- ▶ **One to one:** one record in PK related to one record in FK

- ▶ 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

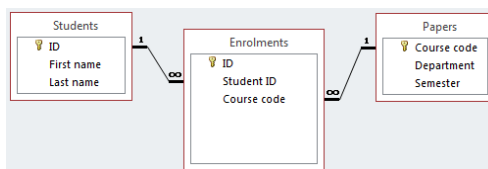
- ▶ **Many to many:** multiple records in PK related to multiple records in FK



- ▶ E.g. An Author can write several Books, and a Book can be written by several Authors

- ▶ E.g. **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)

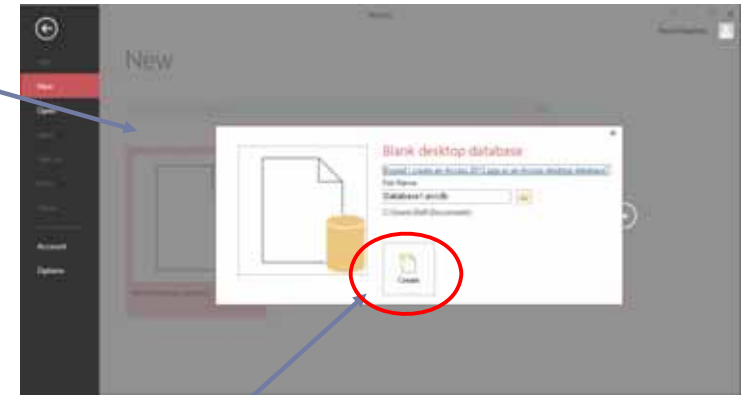
- ▶ Application software that is used to manage databases.
- ▶ Four main functions:
  - ▶ Definition
  - ▶ Update
  - ▶ Querying
  - ▶ Administration
- ▶ Examples:
  - ▶ Microsoft Access
  - ▶ Microsoft SQL Server
  - ▶ MySQL



### Creating a database



Templates

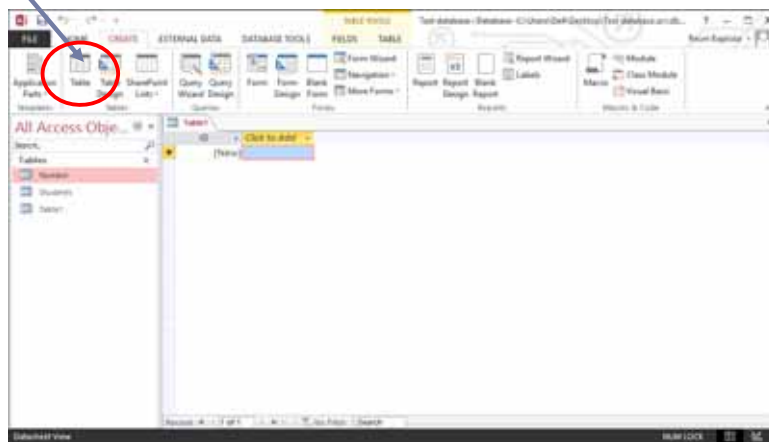


Creating a new database



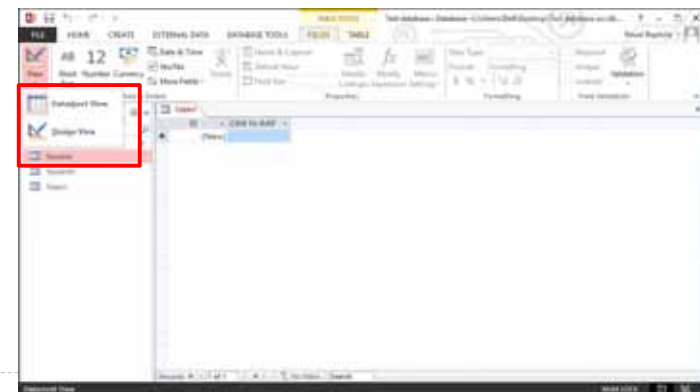
### Creating a table

New table button



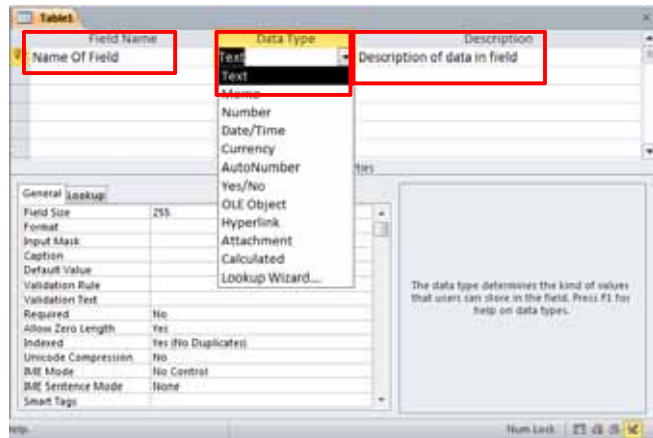
### Creating a table

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





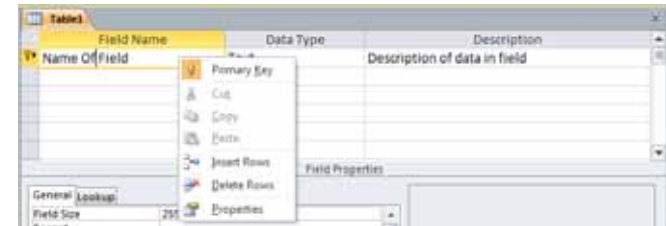
## Design view



21



## Design view

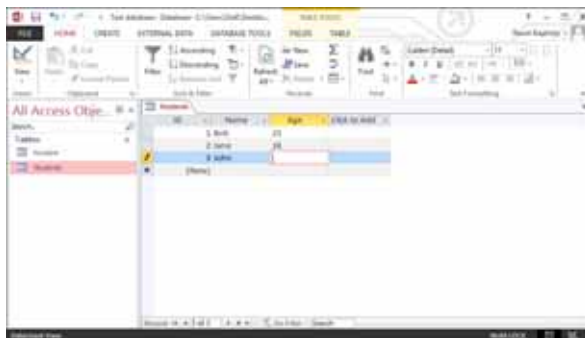


22



## 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)

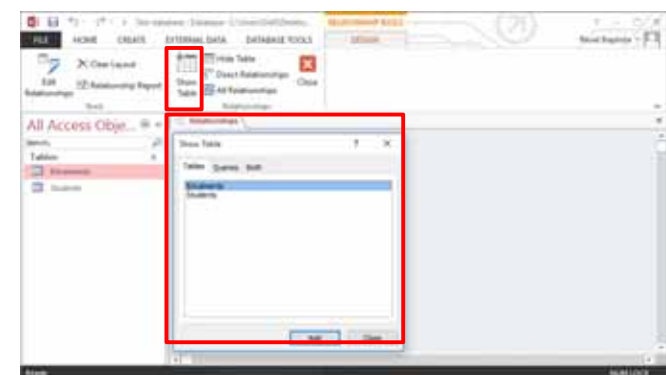


23



## Creating relationships

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



24



## Creating relationships

25

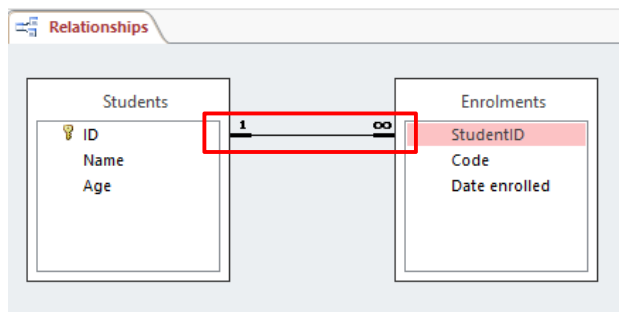


## Creating relationships

26



## Creating relationships



27



## Inserting data

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

StudentID	Code	Date enroll
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?

28



## Inserting data

Microsoft Access

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

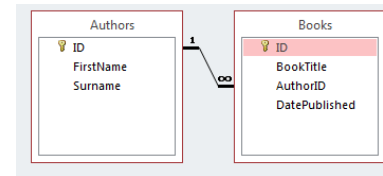
OK Help

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

29



## 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 ...

30



## 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>

31



## 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

32





## 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'