INF10002
Database Analysis and Design

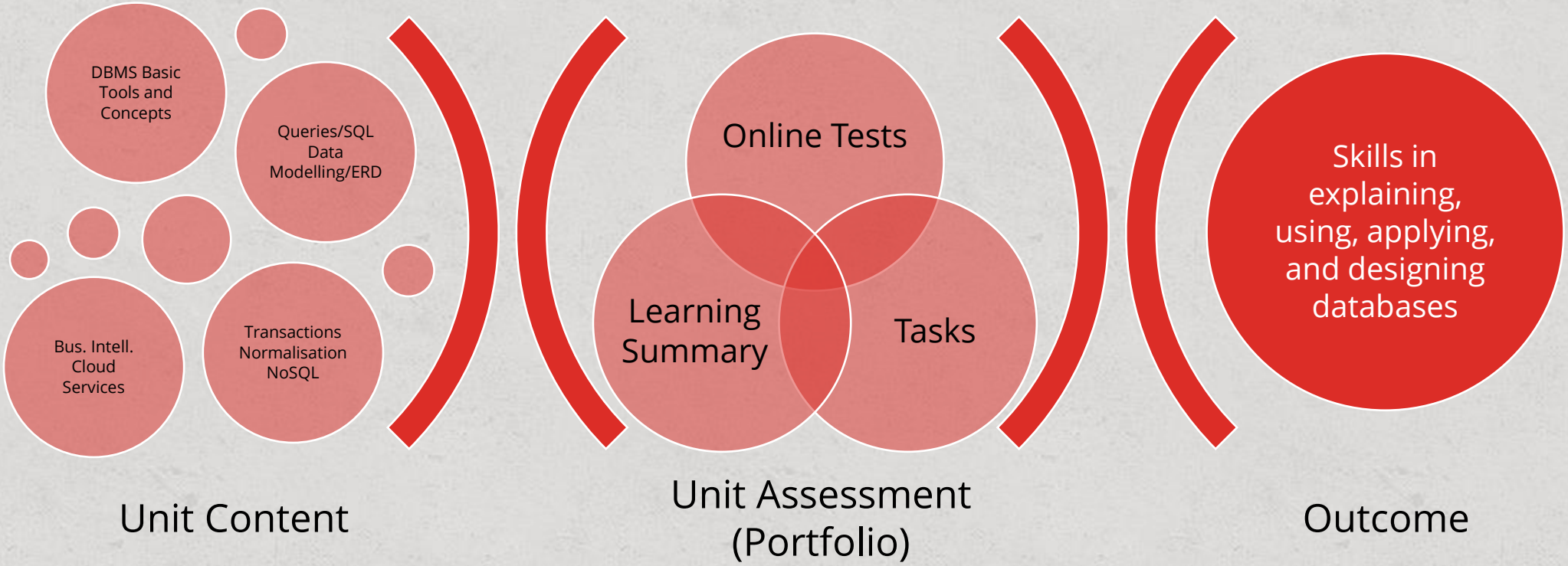
Week 01

Overview, Relational Model,
RDBMS, Database, MS Access, Table, Record, Field
Primary Key, Filter & Query

Unit **Overview**

Aims | Content | Assessment | Schedule | Tools

Unit **Diagram**



Unit **Assessment**

- Portfolio unit
  - No exam

- Requirements for
  - Pass
  - Credit
  - Distinction
  - High Distinction

- See Canvas > Unit Assessment for more specifics on the above

Unit **Tools and Techniques**

- Microsoft Access
- Microsoft Power BI
- SQL and iSQLJr (Oracle)
- ERD Modelling (and Microsoft Visio, Draw i.0)
- Microsoft Azure DocumentDB information

Communications

- Large student body.
- Canvas announcements.
- Questions:
  - **Contact your tutor first.**
  - Best method of communication is by email.
  - Tutor will refer issue to Convenor if required.

Understanding **Data**

Challenges | Unstructured | Semi-Structured | Structured

The Modern **Data Challenge**

- Organisations need to store and retrieve usually large amounts of data
- Data can be divided into three major **categories**
  - **Structured Data**
  - **Semi-Structured Data**
  - **Unstructured Data**
- Structured Data is typically used by Relational Database Management Systems (RDBMSs) such as Access, Oracle, SQL Server, MySQL.

Unstructured Data

- **Unstructured data** is not organized in a pre-defined manner.
- The organization **does not know the format** nor the content of the data in advance.
- Consider data sourced from **social media & email** etc. The contents are unpredictable.
  - The data may contain text, audio, video, links, images. One item may include many data about many organizational functions.
- **How** do organizations store such data so that it can be retrieved, collated, analyzed?
- We will deal with the topic of unstructured data later in **future weeks**.

Semi-Structured Data

- **Semi-structured** data is information that doesn't match the requirements of a relational database.
- The data is organized / arranged that makes it easier to analyze.
- Examples of semi-structured data include **XML** documents and **NoSQL** databases.
- We will briefly deal with the topic of semi-structured data in future weeks

Structured Data

- **Relational Database Management Systems** require data to be stored in a very structured way.
- These systems deal with data that has a **repetitive pattern or format**.
- Consider **Student** data stored in a University. While every student is different, the university want to store data in the same format for every student. Data Types are also specified for each piece of information
  - Student ID – Numeric
  - Student Name – Alpha
  - Gender – Alpha
  - DateOfBirth - Date
  - HomeAddress - Alpha
  - PhoneNo - Numeric
  - NextOfKin - Alpha

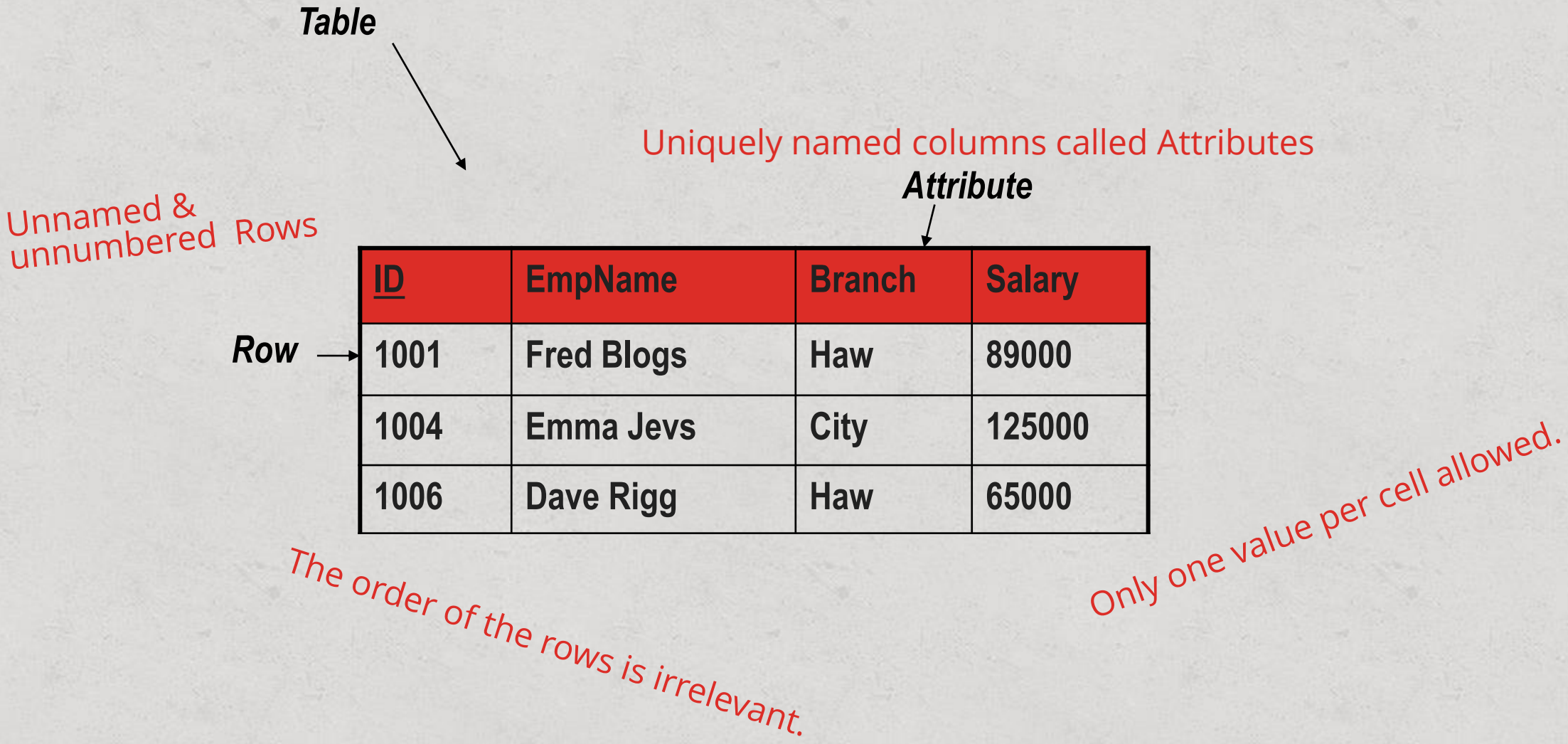
Relational Data Model and **RDBMS**

Tables | Columns | Rows | Advantages |RDBMS | Servers | Interfaces

Relational Data **Model**

- **RDBMSs are based on the Relational Data Model**
  - Developed by Ted Codd in 1970.
  - Data is represented in the form of two-dimensional **tables**.
- Each two-dimensional table has the following properties:
  - A set of uniquely named **Columns / Attributes**
  - A list of unnamed/unnumbered **Rows**
  - The **order** of the rows is **irrelevant**.
- A **Row** consists of a sequence of **Attributes**
  - **One cell** for each Attribute
  - Only **one value per cell** is allowed.
- **All** Relational Database Management Systems are **based on** the Relational Data Model.

Relational Data **Model** (cont.)



Relational **Databases**

- A relational database is simple of collection of related tables

- Example:
  - **Student Table** (stores data about students)
  - **Unit Table** (stores data about university units of study)
  - **Enrolment Table** (stores data about the units that a student has enrolled into)

RDBMS

- A RDBMS is a collection of programs that allow developers / users to store & retrieve data from relational databases

- It allows users to perform CRUD (create, read, update & delete) operations on data in the tables. E.g:
  - **Create** a student
  - **Retrieve** a student's details
  - **Update** a student's details
  - **Delete** a student details from a table

Setting up a RDBMS

- **Tables**
  - Follow a 2 dimensional structure
  - Each row of data has a Primary Key value
  - No duplicates e.g. Student ID

- **Constraints** can be added to validate data
  - Student ID is correct length
  - Student type is PG or UG (post or undergraduate)
  - Student is enrolled in a degree that actually exists

Advantages of RDBMs

- **Constraints** are usually easier to implement compared to a spreadsheet e.g. 'lookups' are easier and 'rules' are easier
- When **searching** for data, only rows that match criteria are retrieved, unlike a s/s where the entire spreadsheet is loaded
- The data only exists in **one location** (e.g. a database server) so we don't have multiple copies of the same(?) spreadsheet
- **Changes** to data only occur in a single place and all users can see that change immediately, unlike s/s where sheets may be consolidated
- **Other benefits** such as concurrency, eliminating data redundancy, relationships with other tables (To be covered over next few weeks)

RDBMS and **Database Servers**

- A database server is a computer that is **networked to other computers**
- The database server **stores databases**
- Users on the network **can access the data** stored in the databases
- There is only **one copy** of the data (excluding backups etc.)
- RDBMSs allow **multiple users on the network to update data** in database tables.
  - Many people can check the price of product 20
  - Many people may choose to enroll in INF10002 simultaneously
  - Many users may purchase tickets for a flight at the same time.

RDBMS and **Databases Servers** (cont.)

- Imagine trying to use a Word Processor or Spreadsheet to do this**!!!**
- Could **Telstra** store details about all of their customers in a Word Processing document or spreadsheet? Loading! Searching! Updating!
- Could **ANZ** store all deposits and withdrawals of all customers in a single spreadsheet? Thousands / Millions / Billions of rows.
- Imagine trying to retrieve all the deposits made by **Cust 123**4 over the past 24 months from spreadsheet data
  - Size of sheets
  - Load time
  - Lack of computer memory
  - Unnecessary loading of other customers banking transactions.

GUI Interfaces

- There are **two major ways** of interacting with data within a Database

- The first is using a **GUI** (graphical user interface).
  - E.g. Microsoft Access
  - Most actions are carried out with mouse clicks, pull down menus, drag and drop and minimal typing

- **Pros:**
  - **Easy** to use (after initial training)

- **Cons:**
  - Complex actions may be **relatively slow** involving lots of menu options and filters etc.
  - Complex actions **cannot easily be saved**. They must be replicated again and again. Can be tedious.

SQL Interfaces

- An alternative method of interacting with a RDBMS is using **SQL (structured query language)**

- Every action is carried out via the SQL language that enforces a **very strict syntax**.

```
SELECT   movieno, title, relyear
FROM     movie
WHERE    (relyear >=2005 And relyear<=
2009)
AND      title IS LIKE "%Harry Potter%"
ORDER BY relyear
```

- **Pros:**
  - Useful for programmers and developers who want to automate common tasks
  - Typing an SQL command can (to some people) be quicker than a series of mouse clicks

- **Cons:**
  - Relatively difficult to learn
  - Requires typing skills
  - Strict syntax

- Many RDBMs offer a **combination** of GUI and SQL interfaces

Data Basics in MS Access

Overview | Tables | Datasheets | Rows/Records | Columns/Fields | Properties | Primary Keys | Autonumber |

Access and LibreOffice Base

- Access is a **complete RDBMS**
- It is mainly used by individuals or small organisations
- It is a good learning tool for students
- Access requires **MS Windows**
  - **Mac Users** can run MS Access using VirtualBox, Apps on Demand -Citrix (see Canvas→Modules→Unit Information).
- Access is installed on all **on-campus PCs**

Multi-users, **Sharing** and Access

- Access users typically **do not create a database server**
- They usually **do not share data with other users** (unlike most business database users).
- Swinburne labs **do not easily allow students to share databases** with other users.
- So you will be creating **single user databases** in labs ☺
- You will experience using data in a multi user environment in other units in your course

Table **Design and Definition**

- The design view of a table shows details about all **fields** in a table
- Each field has a **data type** (defines what type of data is allowed to be stored in this field)
- We will concentrate on **Short Text, Number, Currency & Date** this semester
- The  symbol indicates this field is a **Primary Key**
- In this example, movies are **uniquely identified** by MOVIENO:

MOVIE

Field Name	Data Type	Description (Optional)
MOVIENO	Number	
TITLE	Short Text	
RELYEAR	Number	
RUNTIME	Number	
RATINGCODE	Short Text	
COLOUR_CODE	Short Text	
TMDB_SCORE	Number	
TMDB_VOTES	Number	
IMDB_ID	Short Text	

Datasheet View

- The datasheet view of a table **shows rows in a table**
- Not all rows can be loaded at once. If a table contained millions of rows, this would **not be possible**
- The datasheet view is **dangerous**. If you change the contents of a cell and move to a new row, the data is changed permanently. There is no question asking "Do you want to save changes"!

MOVIE

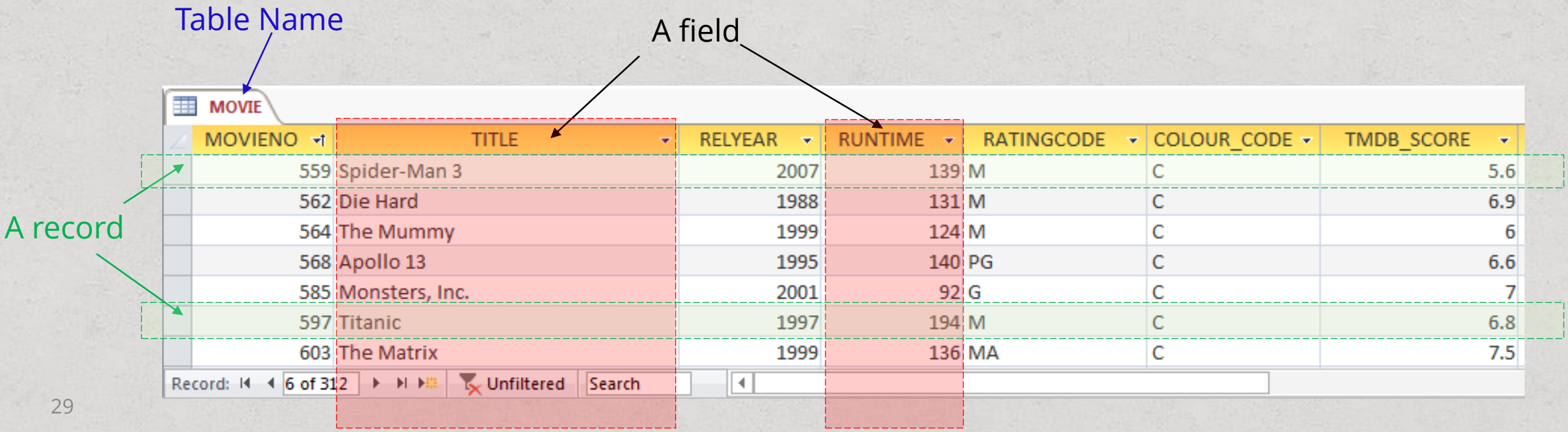
MOVIENO	TITLE	RELYEAR	RUNTIME	RATINGCODE	COLOUR_CODE	TMDB_SCORE	TMDB_VOTES	IMDB_ID
22	Pirates of the Caribbean: The Curse c	2003	143	M	C	6.9	2114	tt0325980
24	Kill Bill: Vol. 1	2003	111	MA	C	6.9	1021	tt0266697
70	Million Dollar Baby	2004	132	M	C	7	371	tt0405159
77	Memento	2000	113	MA	C	7.4	788	tt0209144
78	Blade Runner	1982	117	MA	C	7.5	986	tt0083658
95	Armageddon	1998	151	M	C	6.2	760	tt0120591
114	Pretty Woman	1990	119	MA	C	6.4	303	tt0100405
118	Charlie and the Chocolate Factory	2005	115	PG	C	6	426	tt0367594
120	The Lord of the Rings: The Fellowshi	2001	178	M	C	7.4	3515	tt0120737
121	The Lord of the Rings: The Two Towe	2002	179	M	C	7.4	2877	tt0167261

Records/Fields vs. **Rows/Columns**

- Most databases & DB developers use the term **Row rather than Record**
  - Access uses the term Record.
  - Oracle (which we will use later) uses the term Row
  - We will use them interchangeably during the semester
- Most databases & DB developers use the term **Column rather than Field**
  - Access uses the term Field.
  - Oracle uses the term Column
  - We will use them interchangeably during the semester

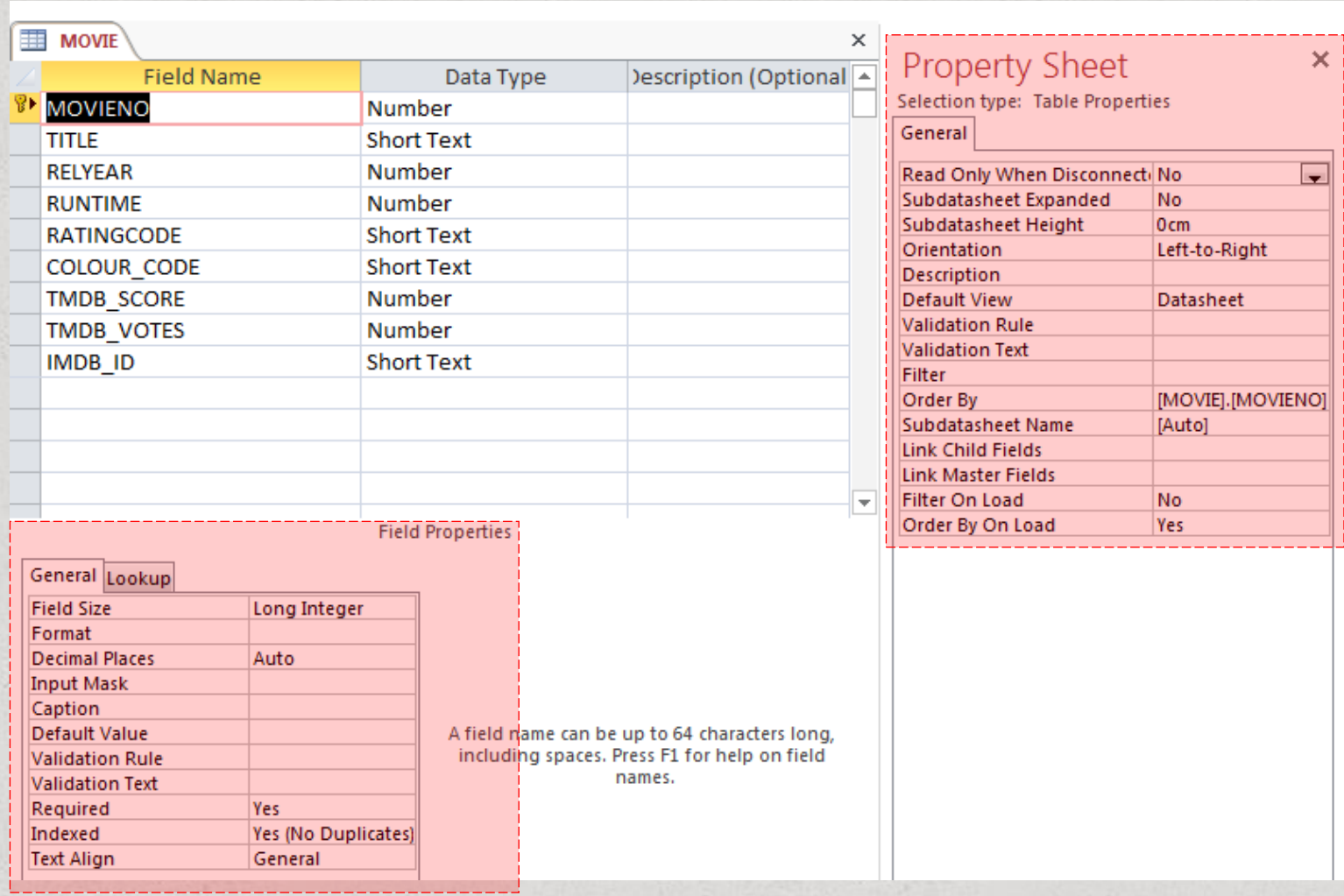
Access **Fields and Records**

- The movie table contains 312 records
- Each record contains data about one movie
- Each record is made up of many fields.



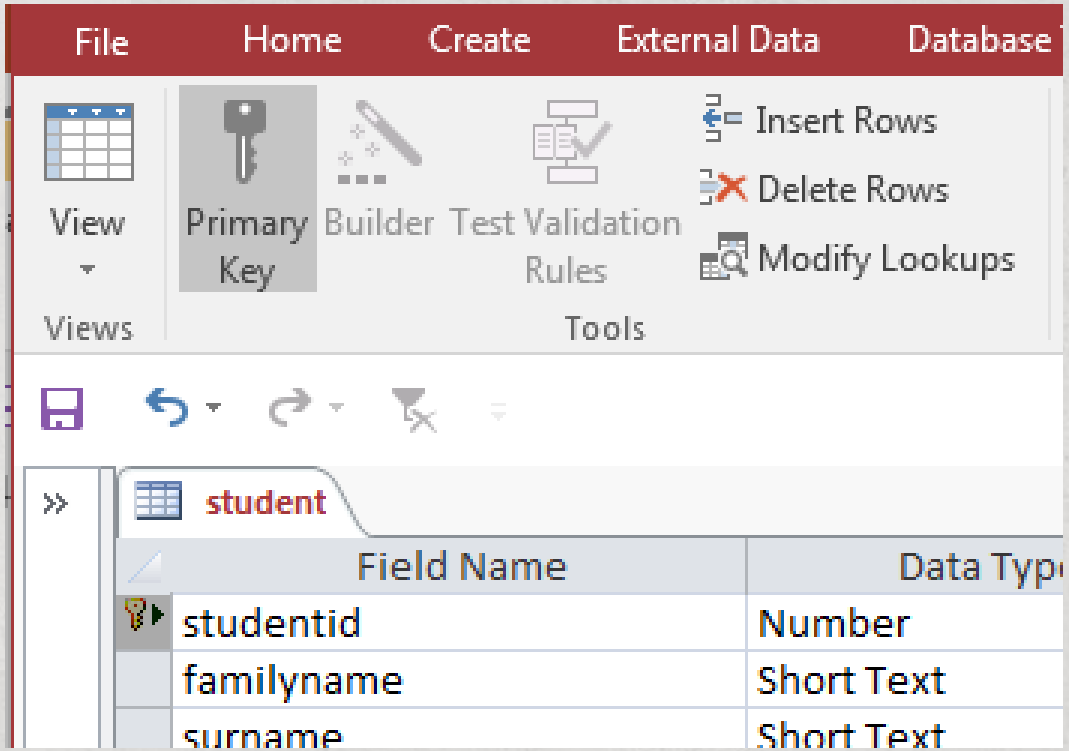
Field **Properties** and **Property Sheet**

- As you use the Design view of a table, you may see
  - The **Property Sheet**
  - **Field Properties**
- Don't be overwhelmed.
- This semester, we will ignore almost all of these settings



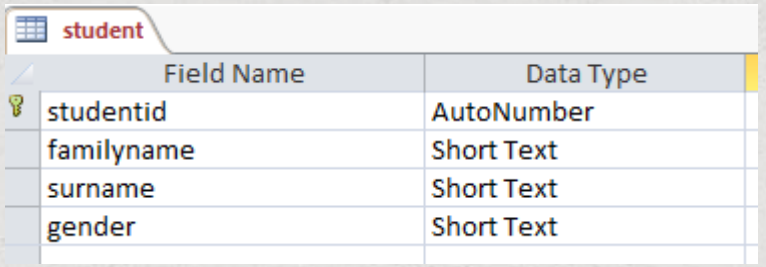
Primary Keys

- **Every** table that you create **must** have a **Primary Key**.
- While this is optional within Access, any table based on Codd's Relational Model must have a Primary Key.
- A Primary Key is a field which will contain a value that is a **unique identifier** for each record
- **No duplicates are** permitted.
- The data should be **static**. It should be a value that should never need to change. Select a field and then Click the  icon



AutoNumber

- If you create a table and forget to nominate a Primary Key, Access will ask if you want it to create one for you
- Such a key will have a **AutoNumber datatype**
- In such cases, the user would never have to enter a new student number for each student.
- Instead Access will simply allocate a **sequential numbered value**
- This is a very useful mechanism
- However, **avoid** using AutoNumber datatypes for the first weeks of the semester.

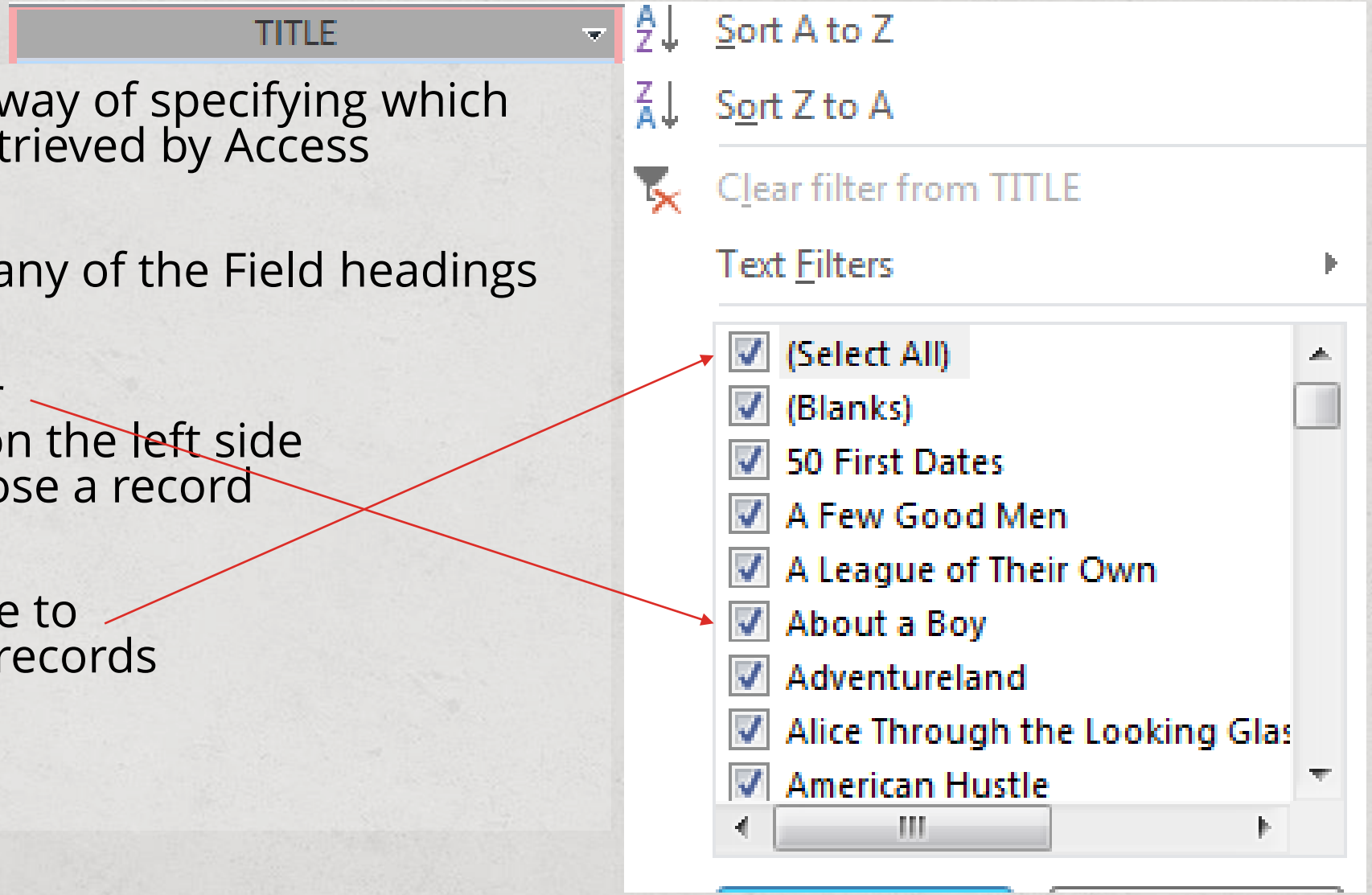


Filters and Queries

Datasheet Filter | Design Grid | Running | Saving | Reusing

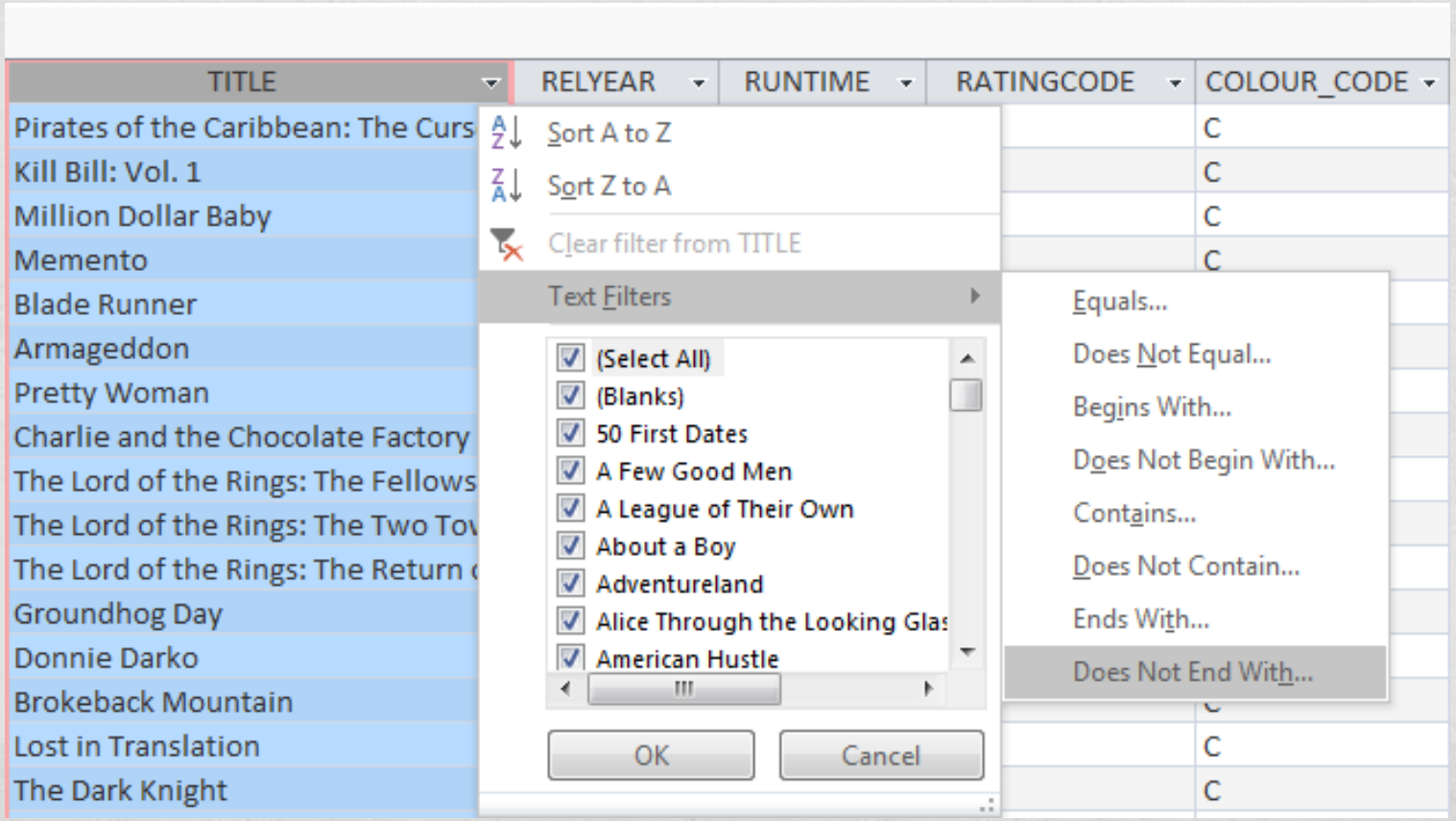
Datasheet **Filters**

- **Filters** are a useful way of specifying which records are to be retrieved by Access
- Click the ▼ icon on any of the Field headings
- Simply check one or more of the boxes on the left side of the menu to choose a record
- Click Select All toggle to select / unselect all records



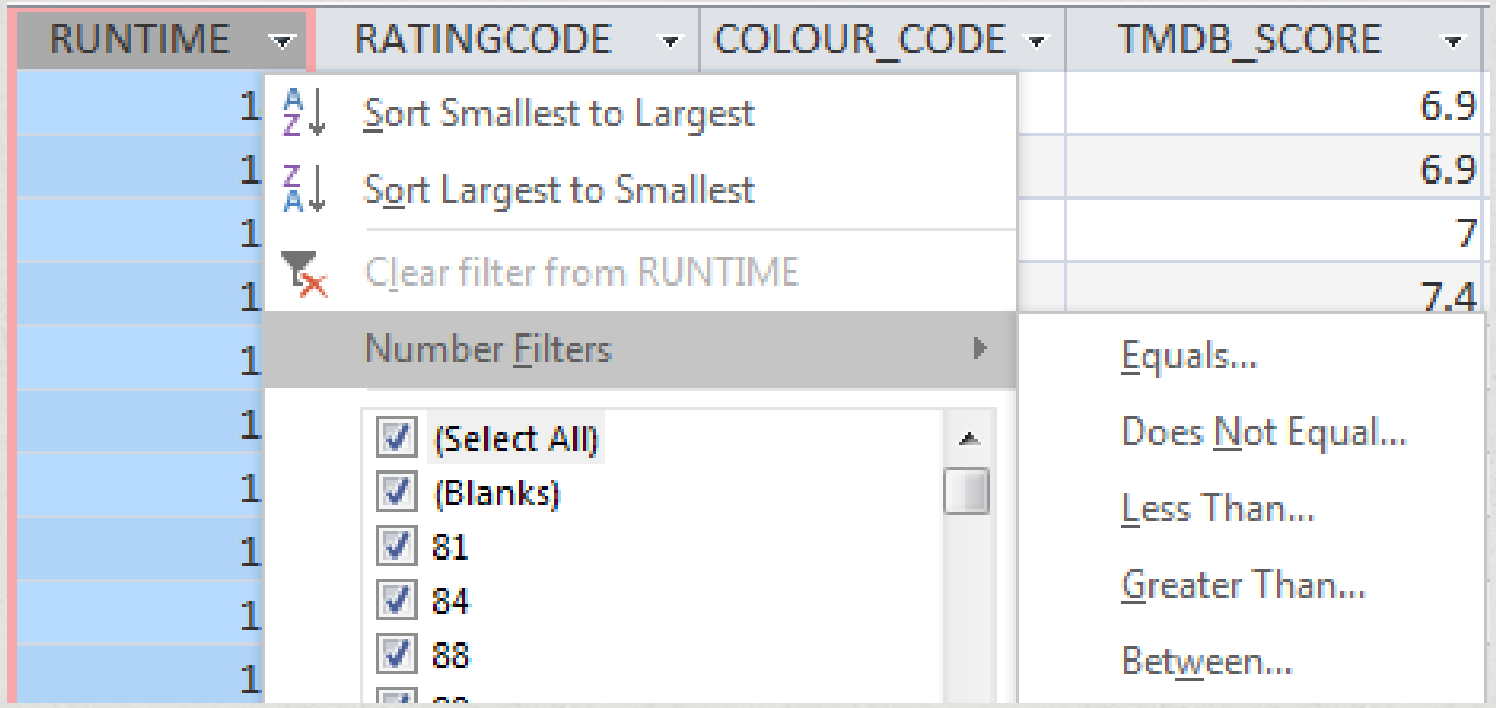
Datasheet **Filters** (cont.)

- Click the **Text** Filters menu option to specify others filter options



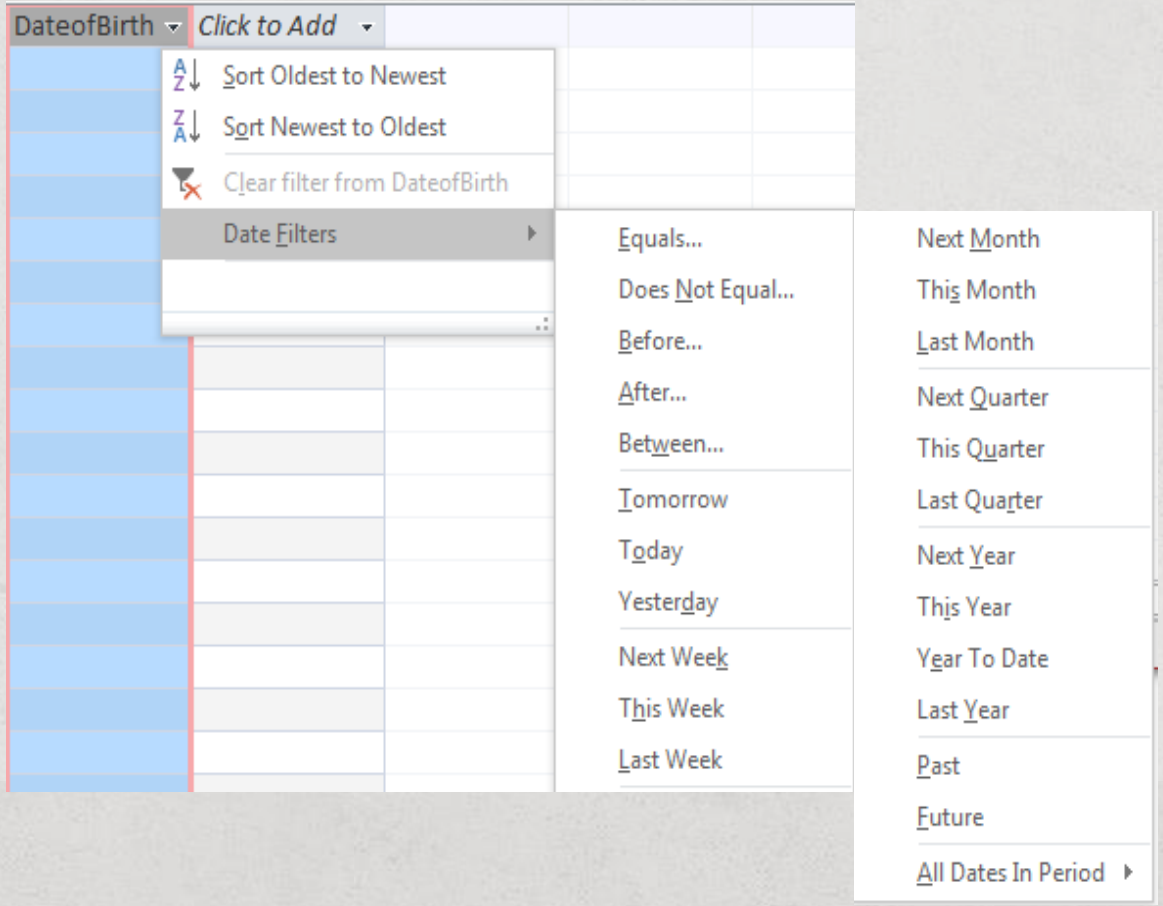
Datasheet **Filters** (cont.)

- **Numeric** columns have a different selection of filter options



Datasheet **Filters** (cont.)

- **Date** columns have huge selection of filter options



Filters **vs.** Queries

- Filters are useful for **'one-off' searches**

- However, Filter settings cannot be saved.
  - The next time you want to repeat the search, you **need to re-create** all of the filter settings again.

- Queries by comparison are most useful when a search **will be repeated**.
  - Filters can be saved as a 'Query'
  - A query can be re-run repeated may times
  - A query can behave just like a table.
  - A new query can use an existing query as its source (instead of a table)

Queries

- An Access query is literally a question that you ask about your data
  - e.g. What is the name and rating code of all movies that were released in the year 2015.

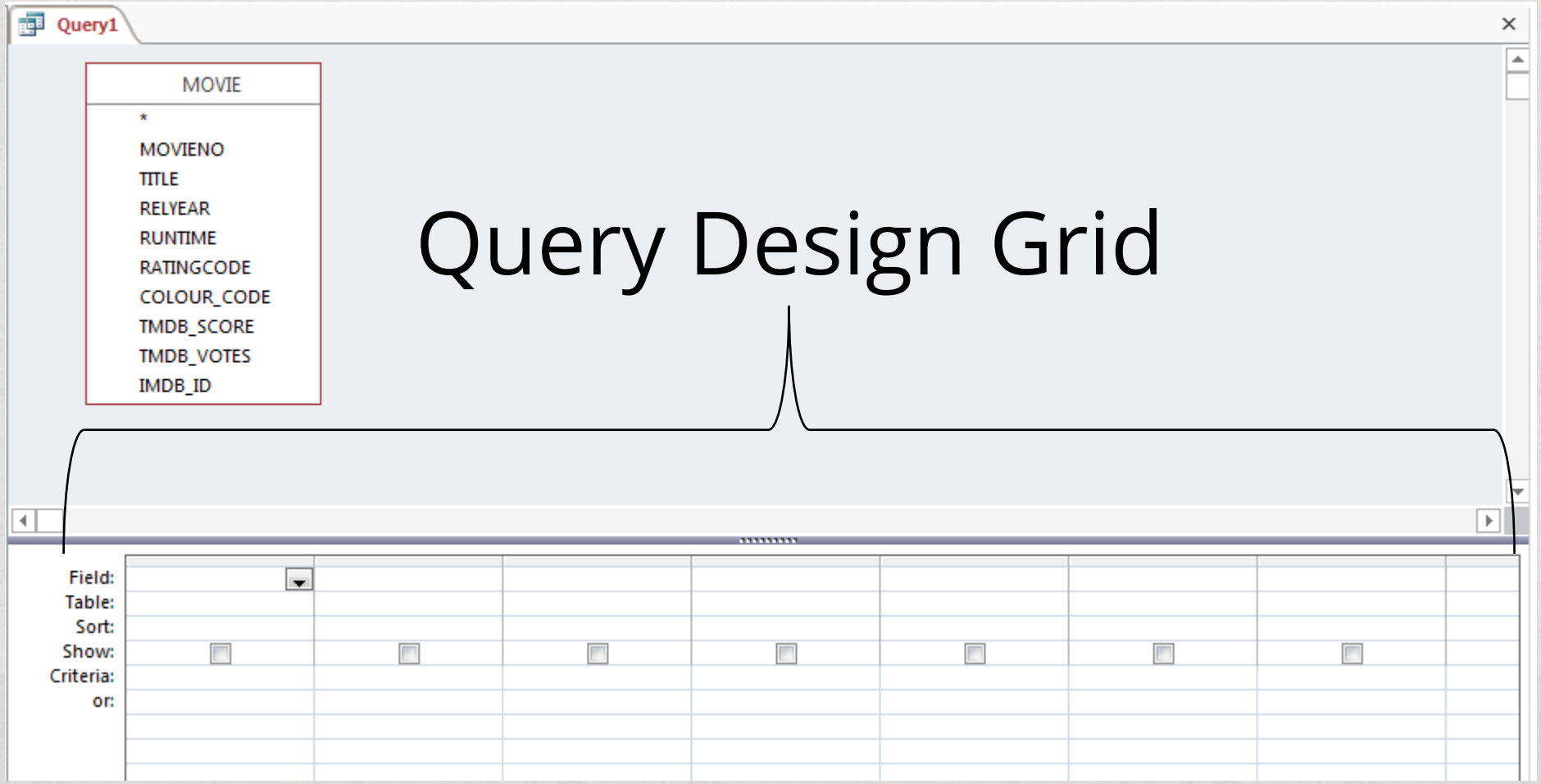
- To Build a query think about:
  - Which **table(s)** store this data
  - Which **fields** do you want displayed
  - What **criteria** does a record have to match
  - What **sequence** do you want to matching data to appear in

Queries (cont.)

- Q: What is the title and rating code of all movies that were released in the year 2015.
  - Which table are necessary?
    - **Movie** table
  - Which fields do you want displayed
    - **Title, RatingCode**
  - What criteria does a row have to match
    - **Relyear** must be **equal** to **2015**
  - What sequence do you want to matching data to appear in
    - **Alphabetical** by **title**

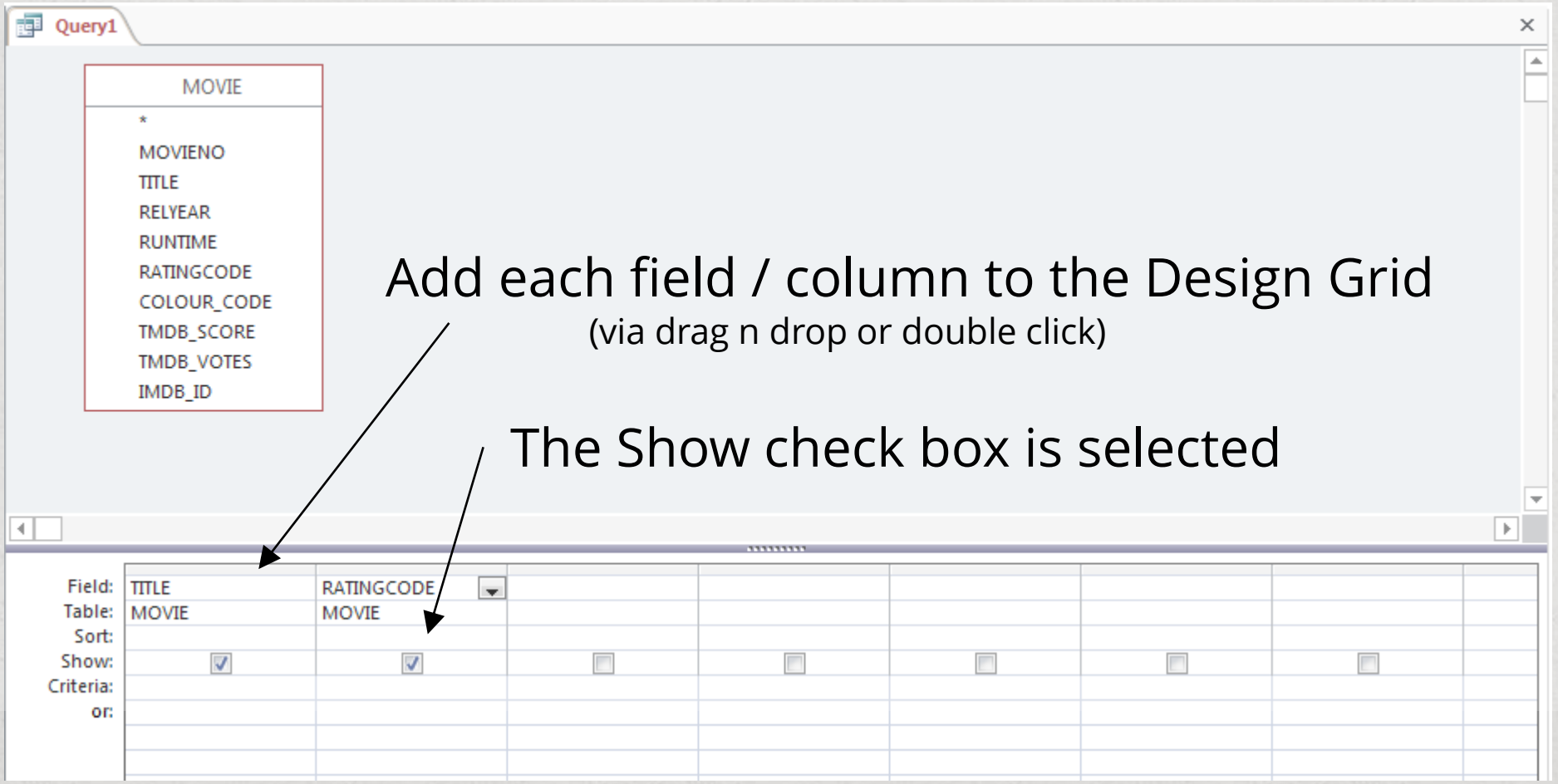
Query **Design Grid**

Q: What is the **title** and **rating code** of all **movies** that were **released in the year 2015**?



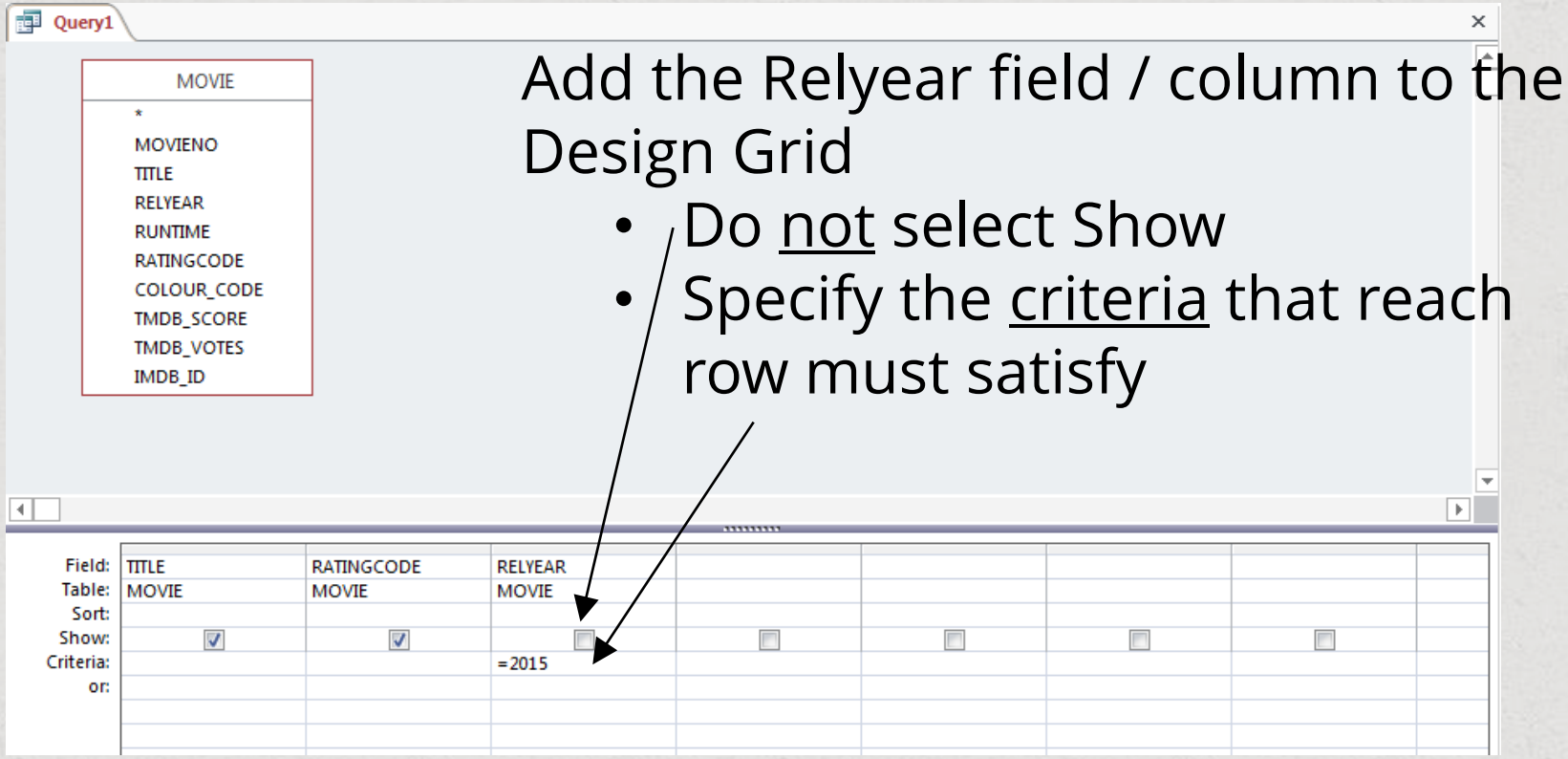
Query **Design Grid** (cont.)

Q: What is the **title** and **rating code** of all **movies** that were **released in the year 2015**?



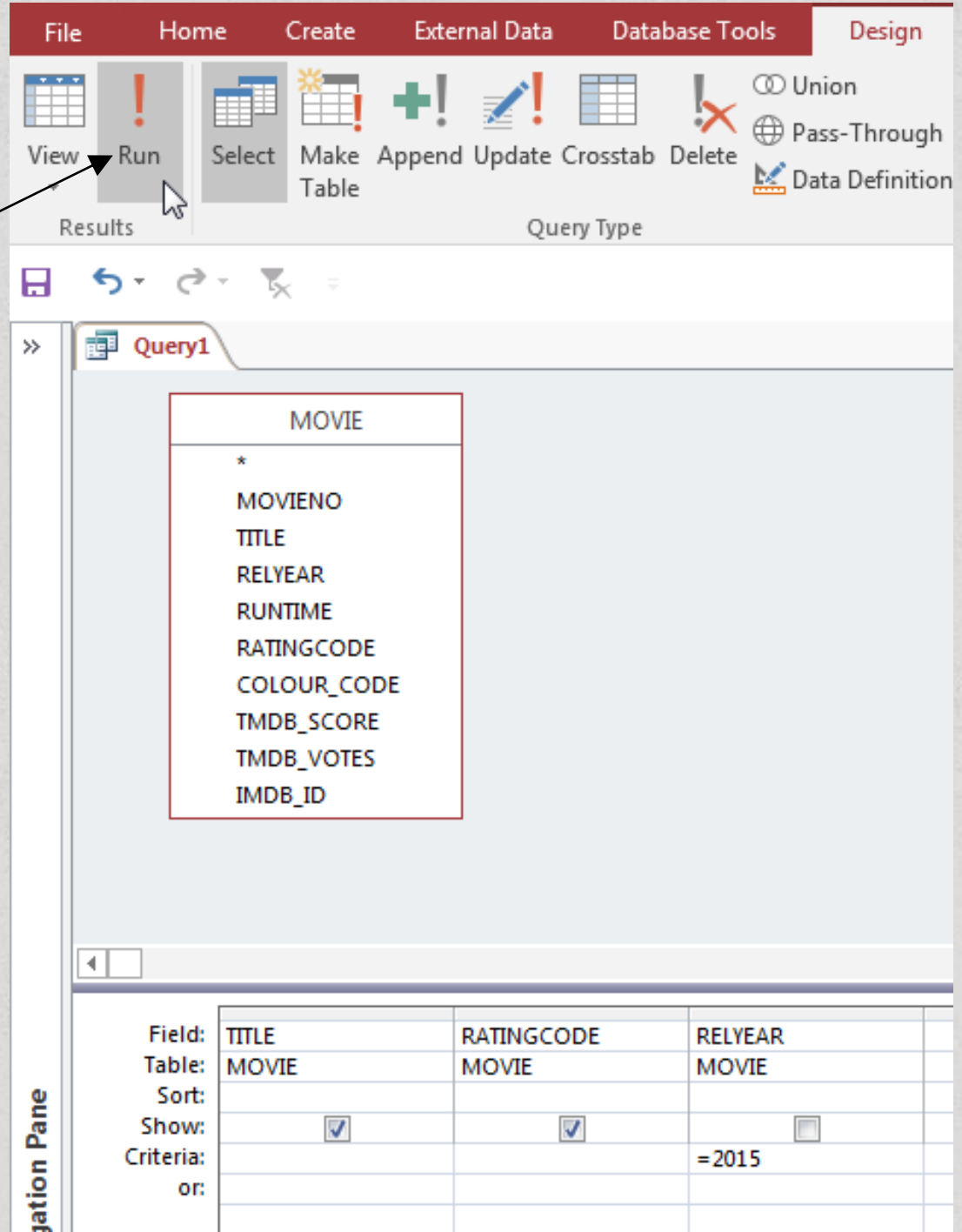
Query **Design Grid** (cont.)

- Q: What is the **title** and **rating code** of all **movies** that were **released in the year 2015**?



Run the Query

- Click the Run ! to execute the query
- A Datasheet view appears with the matching rows



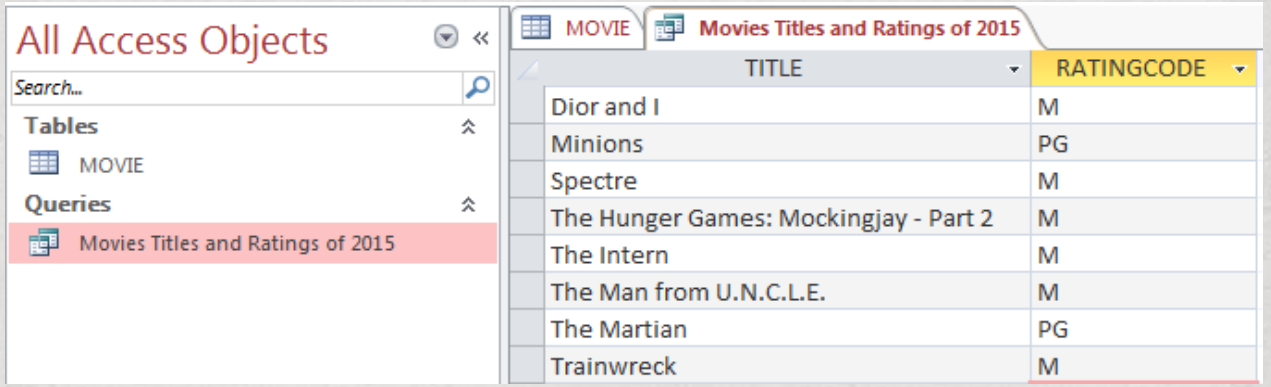
Query1

TITLE	RATINGCODE
Dior and I	M
Spectre	M
The Hunger Games: Mockingjay - Part 2	M
The Intern	M
The Man from U.N.C.L.E.	M
The Martian	PG
Trainwreck	M

Saving Queries

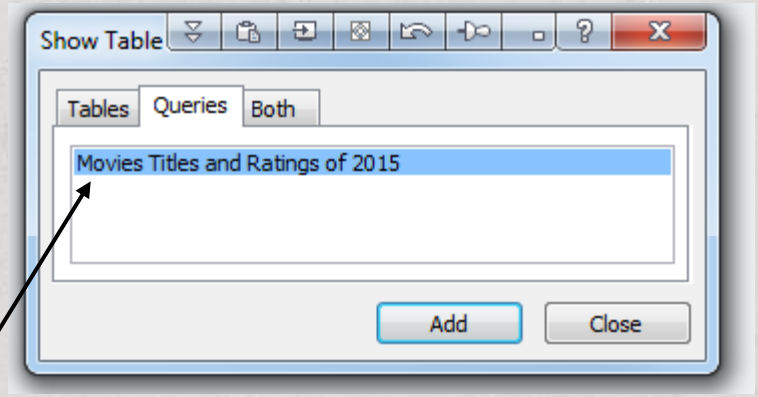
- A Query can be **saved** and **executed again** at a later time.
- The Query is saved **within** the database.
- This database now **contains**
  - A Table named Movie
  - A Query named Movie Title and Ratings of 2015

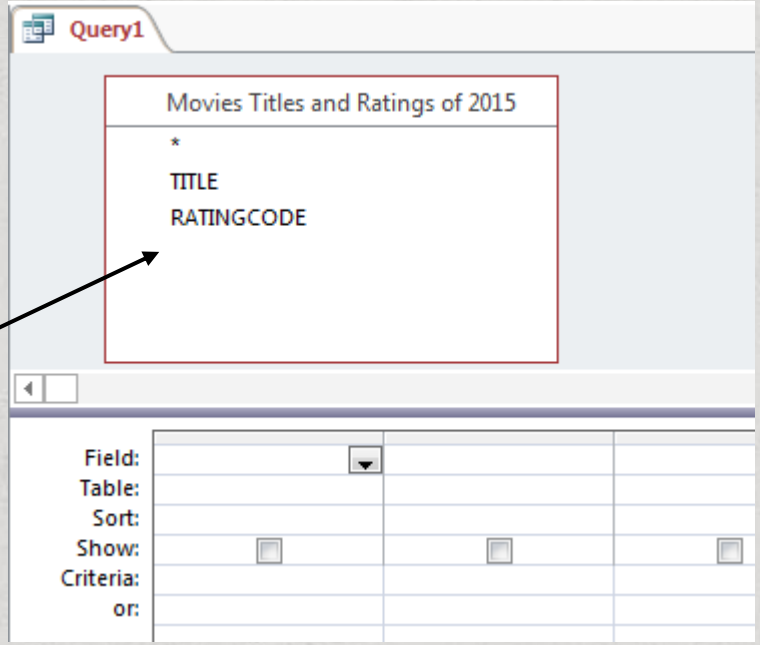
- To run the query again**, simply double click** on the Query named Movies Titles and Ratings of 2015.



Reusing Queries

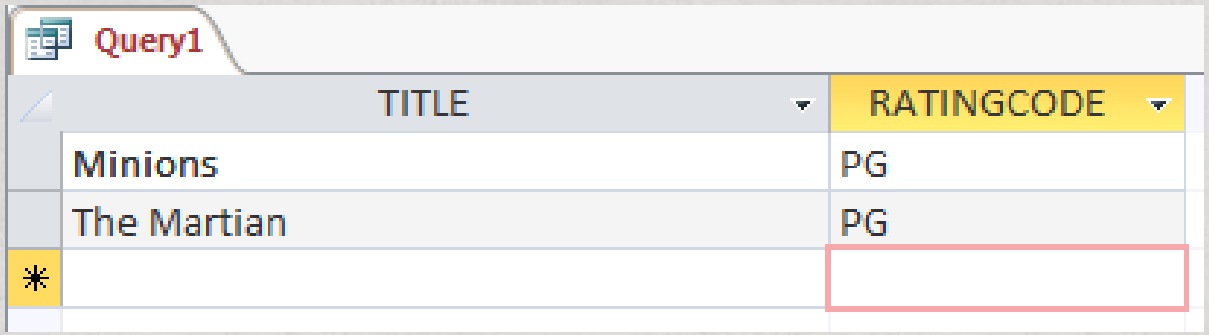
- Queries are often treated in the same way as Tables within Access
- Let us suppose we want to change the criteria
- Now, we only want to display PG movies from the year 2015
- We could edit and modify the previous query.
- Instead… we can create a new query
- Create a New Query via Create / Query Design menu option.
- **Don't** base the data on a table but instead choose ‘Movie Titles and Ratings of 2015’ from the Queries tab.
- The only columns that are available for this query are Title and RatingCode

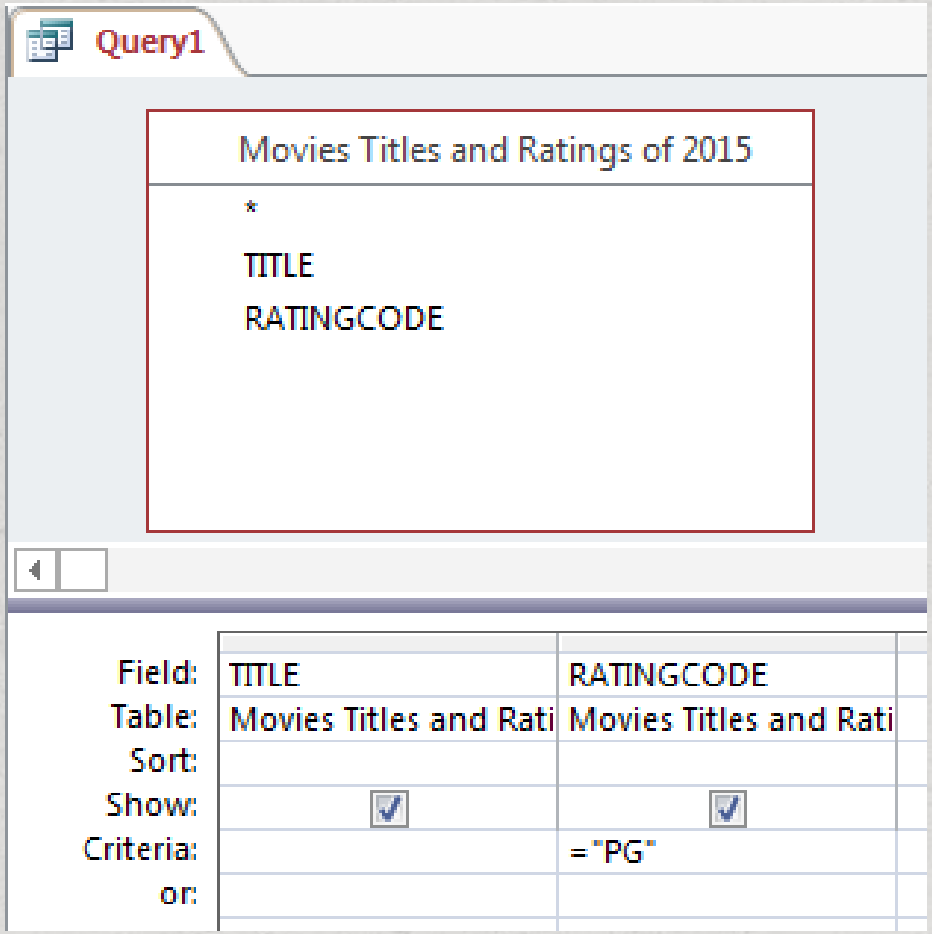




Reusing Queries (cont.)

- **Add** the Title and RatingCode fields / columns to the Design Grid.
- **Add** a criteria to RatingCode so that only PG movies are displayed.
- You could save this as a new query.





References on Queries

- There are many resources available describing how to create Access Queries
- The Access Bible 2016 available on-line via Swinburne Library.
- Also:
  - https://support.office.com/en-us/article/Examples-of-query-criteria-3197228c-8684-4552-ac03-aba746fb29d8?ui=en-US&rs=en-US&ad=US&fromAR=1
  - https://support.office.com/en-us/article/Use-the-OR-criteria-to-query-on-alternate-or-multiple-conditions-d04b3e63-b477-430e-8fdc-7e37189ede88
  - https://support.office.com/en-us/article/Examples-of-Access-query-criteria-0c7e9394-c485-454f-bc00-3bd3ec617805
  - http://www.gcflearnfree.org/access2013/query-criteria-quick-reference-guide/1/
  - http://www.fontstuff.com/access/acctut06.htm

Multiple Criteria, Keywords, and Wildcards

AND | OR | LIKE | Field Names

Multiple criteria in a field

- When a field has **multiple values** on **multiple rows** in the Criteria section, it means that a movie record only needs to satisfy **ONE** of the criteria

- **Meaning:** Evaluate each record. Select the record when

  the relyear is equal to 2013

  OR

  the relyear is equal to 2016

- *Either criteria must be met by a record.*

Field:	TITLE	RELYEAR	RATINGCODE
Table:	MOVIE	MOVIE	MOVIE
Sort:			
Show:	☑	☑	☑
Criteria:		2013	
or:		2016	

TITLE	RELYEAR	RATINGCODE
Alice Through the Lookir	2016	PG
American Hustle	2013	M
Batman v Superman: Dav	2016	M
Captain America: Civil W	2016	M
Dallas Buyers Club	2013	MA
Deadpool	2016	MA
Finding Dory	2016	G
Frozen	2013	G
Ghostbusters	2016	PG
Gravity	2013	PG
Iron Man 3	2013	M
Jason Bourne	2016	M
Man of Steel	2013	M

Multiple criteria in **multiple fields**

- When multiple fields have values the Criteria section, it means that a movie record must satisfy **ALL** of the criteria

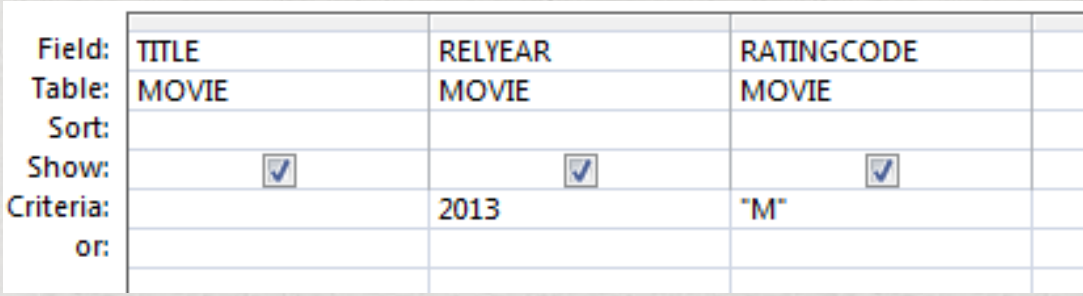
- **Meaning:** Evaluate each record. Select the record when

  the relyear is equal to 2013

  AND

  the ratingcode is equal to "M"

- *Both criteria must be met by a record.*
- *When reading the Criteria across the query it means AND*



Field:	TITLE	RELYEAR	RATINGCODE
Table:	MOVIE	MOVIE	MOVIE
Sort:			
Show:	☑	☑	☑
Criteria:		2013	"M"
or:			

TITLE	RELYEAR	RATINGCODE
American Hustle	2013	M
Iron Man 3	2013	M
Man of Steel	2013	M
Star Trek Into Darkness	2013	M
The Hunger Games: Catc	2013	M
The Lone Ranger	2013	M
Thor: The Dark World	2013	M
World War Z	2013	M

Multiple criteria in **multiple fields** (cont.)

- When **multiple criteria rows** have multiple **values** in the Criteria section, it is tricky.
- Evaluate each record against each Criteria row separately.
- If a record meets all the Criteria of a Criteria Row, then that record is selected.
- **Meaning** Evaluate each record. Select the record when

  All of Criteria A is satisfied

  OR

  All of Criteria B is satisfied
- *Either criteria A or criteria B must be met by a record.*

	Field:	TITLE	RELYEAR	RATINGCODE
	Table:	MOVIE	MOVIE	MOVIE
	Sort:			
	Show:	☑	☑	☑
A	Criteria:		2013	"PG"
B	or:		2016	"M"

TITLE	RELYEAR	RATINGCODE
Batman v Superman: Dawn of Justi	2016	M
Captain America: Civil War	2016	M
Gravity	2013	PG
Jason Bourne	2016	M
Mike & Dave Need Wedding Dates	2016	M
Oz: The Great and Powerful	2013	PG
Special Correspondents	2016	M
The Nice Guys	2016	M
Zoolander 2	2016	M

An alternative way to perform **OR**

- You can use the **OR keyword** instead of using multiple rows in the Criteria area

- Meaning Evaluate each record. Select the record when

  the relyear is equal to 2013

  OR

  the relyear is equal to 2016

Field:	TITLE	RELYEAR	RATINGCODE
Table:	MOVIE	MOVIE	MOVIE
Sort:			
Show:	☑	☑	☑
Criteria:		2013 Or 2016	
or:			

TITLE	RELYEAR	RATINGCODE
Alice Through the Lookir	2016	PG
American Hustle	2013	M
Batman v Superman: Dav	2016	M
Captain America: Civil W	2016	M
Dallas Buyers Club	2013	MA
Deadpool	2016	MA
Finding Dory	2016	G
Frozen	2013	G
Ghostbusters	2016	PG
Gravity	2013	PG
Iron Man 3	2013	M
Jason Bourne	2016	M
Man of Steel	2013	M

An alternative way to perform **OR** (cont.)

- You can use the **OR keyword** instead of using multiple rows in the Criteria area
- Meaning. Evaluate each record. Select the record when

  (the relyear is equal to 2013
  OR the relyear is equal to 2016)

  AND

  (the ratingcode is equal to "PG“ OR the ratingcode is equal to "M")
- This means you can have
- 2013 And PG , 2013 And M
- Or 2016 And PG, 2016 and M

Field:	TITLE	RELYEAR	RATINGCODE
Table:	MOVIE	MOVIE	MOVIE
Sort:			
Show:	☑	☑	☑
Criteria:		2013 Or 2016	"PG" Or "M"
or:			

TITLE	RELYEAR	RATINGCODE
Alice Through the Looking Glass	2016	PG
American Hustle	2013	M
Batman v Superman: Dawn of Justi	2016	M
Captain America: Civil War	2016	M
Ghostbusters	2016	PG
Gravity	2013	PG
Iron Man 3	2013	M
Jason Bourne	2016	M
Man of Steel	2013	M
Mike & Dave Need Wedding Dates	2016	M
Oz: The Great and Powerful	2013	PG
Special Correspondents	2016	M

An alternative way to perform **AND**

- You can use the AND keyword in the Criteria area.
- It is most useful when defining a numeric range.
- **Meaning** Evaluate each record. Select the record when the relyear is between 2011 and 2015

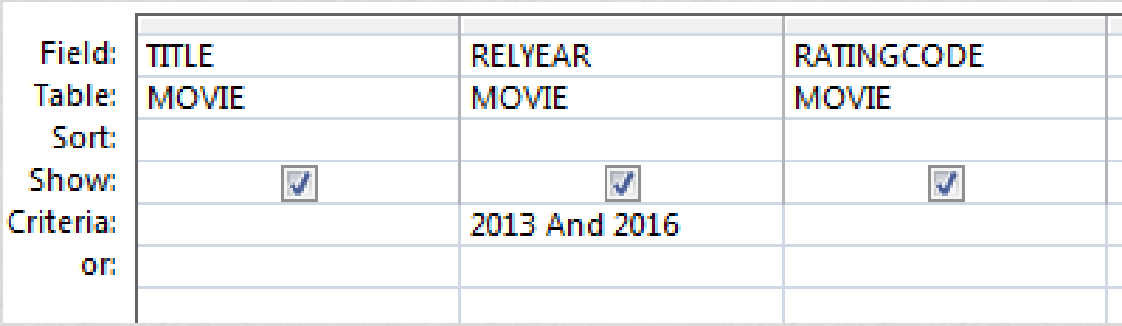
Field:	MOVIENO	TITLE	RELYEAR
Table:	MOVIE	MOVIE	MOVIE
Sort:			
Show:	☑	☑	☑
Criteria:			>=2011 And <=2015
or:			

MOVIENO	TITLE	RELYEAR
168672	American Hustle	2013
68734	Argo	2012
1771	Captain America: The First Avenger	2011
100402	Captain America: The Winter Soldier	2014
212778	Chef	2014
83542	Cloud Atlas	2012

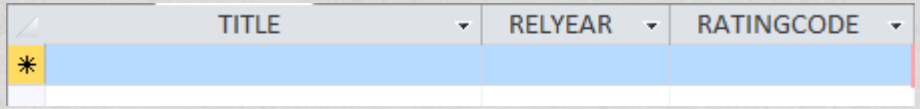
Be carful with AND

- Be careful when using AND
- Try to understand what you are searching for.
- Meaning:
  Evaluate each record. Select the record when
  - the relyear is equal to 2013 AND
  - the relyear is equal to 2016

This is impossible!!!!!



Field:	TITLE	RELYEAR	RATINGCODE
Table:	MOVIE	MOVIE	MOVIE
Sort:			
Show:	☑	☑	☑
Criteria:		2013 And 2016	
or:			



	TITLE	RELYEAR	RATINGCODE
*			

Empty.
No rows match this criteria

Text criteria

- When specifying text values in the Criteria section, the text must be **surrounded by double quotes**
- Access is not case sensitive: **”The” is the same as "THE" and "the"**
- The opposite is true in other databases
- In Oracle, **"THE" is different to "The" and "the"**

Text **Wildcards**

- You **can** use Wildcards in access
- Notice the space after **the**
- If **no space** then you will get an outcome like “Thelma and Louise”
- when you actually want only Titles that start with THE
- These must be used with the **‘Like’ keyword**
- The Like condition allows you to use **wildcards to perform pattern matching**
- The patterns that you can use **are**:

Wildcard	Description	Example	Explanation
*	Allows a match of any **string** of any length	Like "the *"	Title must begin with "the"
		Like "*day*"	Title must contain day
?	Allows a match on a **single character**	Like "the ????"	Title must begin with "the" and be followed by exactly 4 characters

Field **Names**

- When creating field names
  - **Avoid** using spaces
  - **Use** letters A-Z, a-z, and numbers 0-9

- In **queries you will often see examples of field names** surrounded by square brackets
  - e.g. [movieno] [title] [relyear]

- This was introduced because Access allows you create field names that contain spaces! (Please avoid doing this! Leads to headaches).

- Using [ ] is the only way to tell Access the start and end charters of a field name

Tasks based on Week 1 Lecture

- There are tutorial questions and submission tasks associated with this lecture.

- **The tutorial questions…**
  - Should be reviewed by students after this lecture
  - They will be discussed in the tutorial.
  - Tutorial usually discus topics from the previous week's lecture
  - Topics discussed in the above sessions may appear in tasks and in tests

- **The submission tasks…**
  - Can be completed in your own time(remember to check deadlines)
  - You will need MS Access for the first few weeks of the semester
  - The completed tasks must be submitted via Canvas

End.

Thanks for viewing the recording. Good luck this Semester ☺.