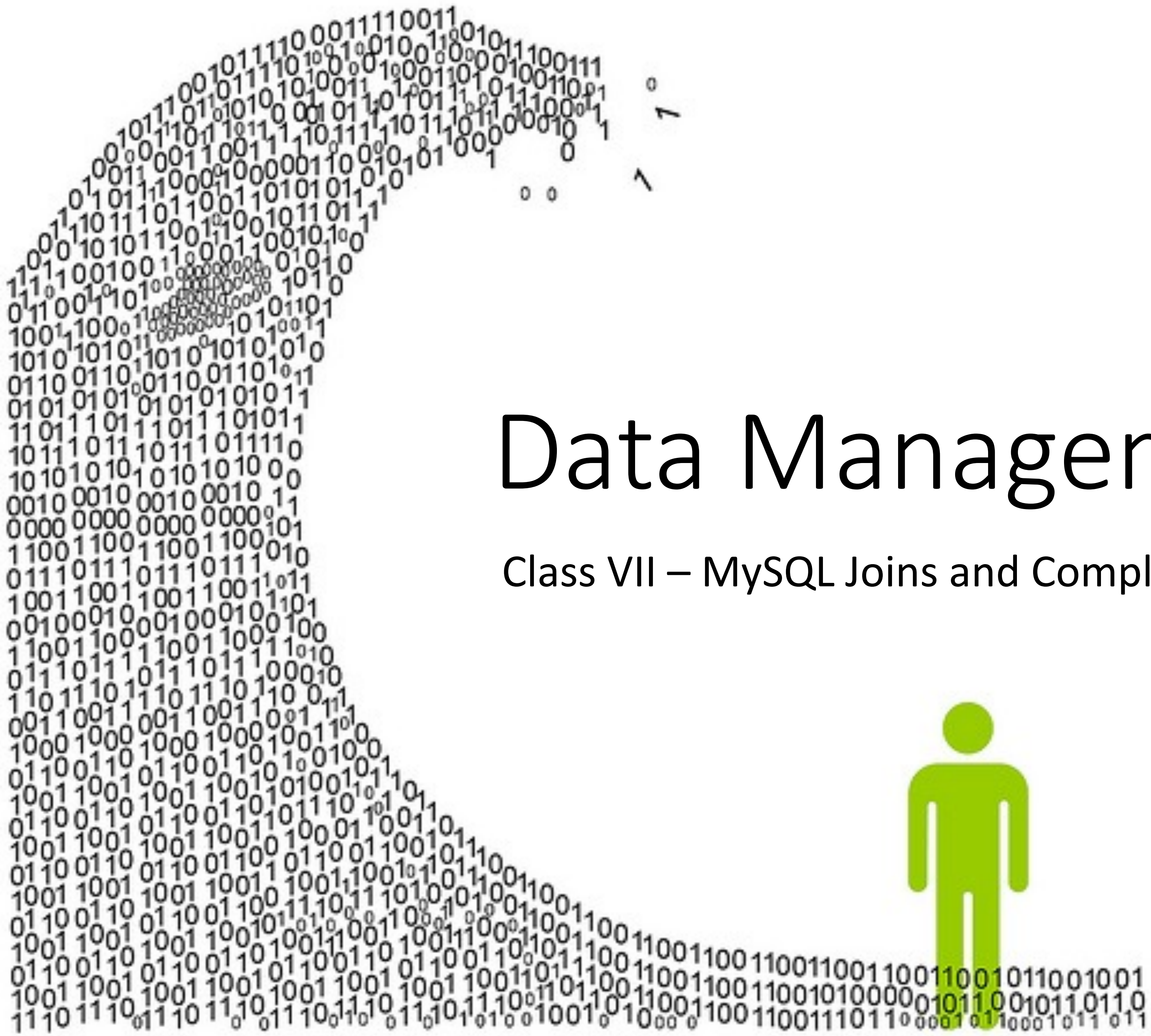


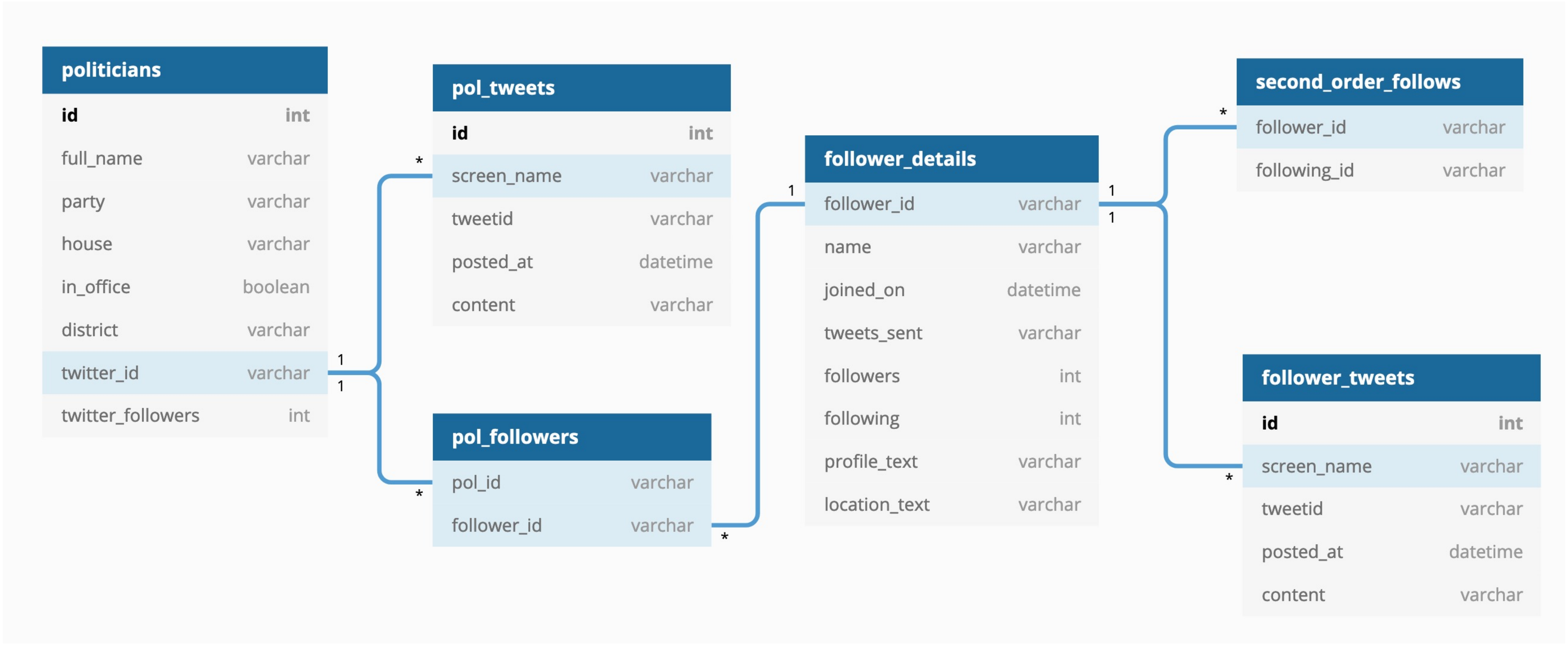
Data Management

Class VII – MySQL Joins and Complex Queries

Querying Multiple Tables

- Perhaps most powerful unique feature of SQL compared to data frames is the **JOIN** command, which lets you search and combine data from multiple different tables.
- There are many cases where this is useful; often, you’ll want to create a database storing different kinds of data (information about a user, information about the people they follow on social media, information about their social media posts, etc.) which is linked together by *keys,* unique identifiers like Twitter IDs that are found in all the different bits of data.

A set of linked tables…



politicians	
id	**int**
full_name	varchar
party	varchar
house	varchar
in_office	boolean
district	varchar
twitter_id	varchar
twitter_followers	int

pol_tweets	
id	**int**
screen_name	varchar
tweetid	varchar
posted_at	datetime
content	varchar

pol_followers	
pol_id	varchar
follower_id	varchar

follower_details	
follower_id	varchar
name	varchar
joined_on	datetime
tweets_sent	varchar
followers	int
following	int
profile_text	varchar
location_text	varchar

second_order_follows	
follower_id	varchar
following_id	varchar

follower_tweets	
id	**int**
screen_name	varchar
tweetid	varchar
posted_at	datetime
content	varchar

Querying Multiple Tables (2)

- To write a query that uses data from multiple tables, we use the **JOIN** command. This literally “joins” the two tables together, so all their columns become part of one ‘super-table’ which we can filter and query using the usual tools in the **SELECT** command.
- Usually, you want to make sure that the correct rows in Table A are joined with the rows in Table B – for example, that a Twitter user in Table A is only joined up with their own tweets from Table B.
- This is accomplished using the **ON** statement, which tells SQL how to choose the rows to join.

Querying Multiple Tables (3)

- This is a very simple example of an SQL JOIN…

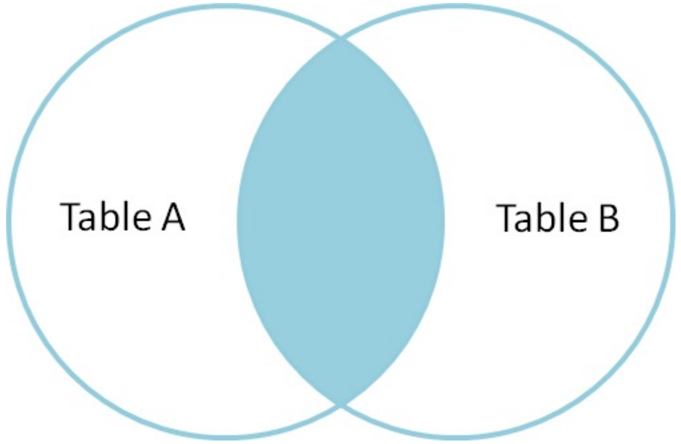
```
SELECT * FROM table1
INNER JOIN table2
ON table1.id = table2.id
```

- This query would return a single row with all of table1’s columns *and* all of table2’s columns, for every single row in either column where the ‘id’ value was the same.

Inner and Outer JOINs

- In the last example, we used an "INNER JOIN". This is the most common kind of JOIN, and it returns only the records in Table A and Table B which have the same ID field.
  - In other words, if there's data in either table which *doesn't* have matching data in the other table – for example, a Twitter user in Table A who doesn't have any tweets stored in Table B – their data won't be returned at all.

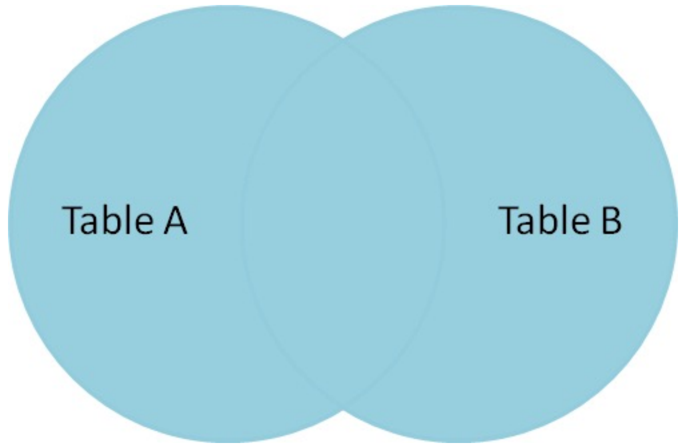
- It's called an "inner" join because like the Venn diagram opposite, only records which *overlap* between the two tables are returned in the query.



Inner and Outer JOINs

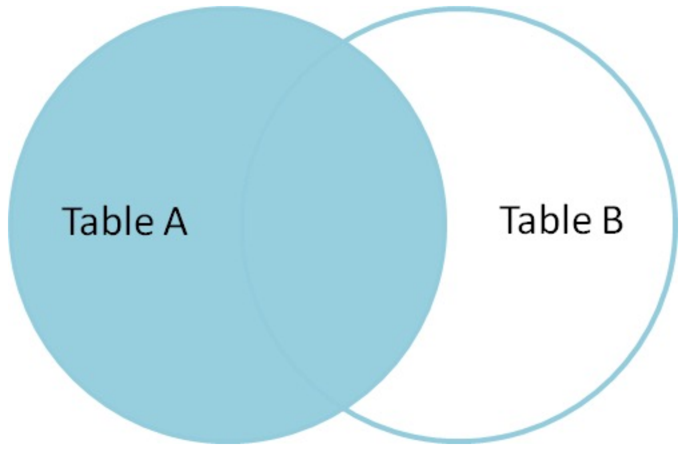
- There are also OUTER joins – the most commonly used of which are LEFT and RIGHT joins.
    - These will return records from tables *even when there isn’t a matching record in the other table* – so the values in the columns from those tables will become “NULL”, showing that there was no match.

- This is a FULL OUTER JOIN between Table A and Table B. Every record from both tables will be returned, with “NULL” values filling in when there wasn’t a match in the other table.

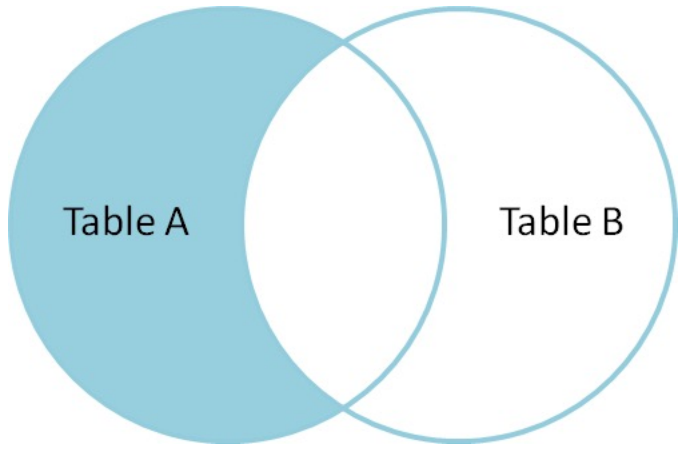


Inner and Outer JOINs (3)

- This is a LEFT JOIN between Table A and Table B. Every record from Table A is returned, with “NULL” values filling in when there wasn’t a match in Table B.



- This is also a LEFT JOIN between Table A and Table B – but here, we check if the Table B columns are NULL (e.g. **WHERE tableB.id IS NULL**) and only return those columns. This way we can find which records in Table A have no matches in Table B.



More on JOINs

- Using the JOIN commands is really powerful, as it allows you to combine data from multiple tables easily. One common usage is to create a new table to store your analysis results – this way, JOIN lets you easily access the original data and analysis results together.
- It definitely takes a while to understand how these commands work. For now, the main thing is to know that they exist and are available to you… When you need to actually use them, you can always find detailed guides and references online.

SQL Best Practices

- Finally for today, I thought we should talk a little bit about how to most effectively use SQL databases.

- There are some things about SQL which are different to how you’re used to doing things, and these can be a bit counter-intuitive.

1) Storage is Cheap

- A lot of people have the instinct to try to minimise the amount of data they store – they avoid using too many tables, for example.
- In reality, **storage is cheap**. Even your smartphone probably has at least 64Gb of storage – enough to store vast amounts of text data. Cloud services only charge a few cents a month for storing gigabytes of data.
- So… Don’t be afraid to store *lots* of tables, and in general, *never delete data*. Instead of deleting data from tables, make a column called “deleted” (or whatever you like) and set it to True when you don’t want to use that data. You never know if it might be useful in future.

2) Processing Takes Time

- Storage is cheap… But processing always takes time, even on the fastest computer.
- Your objective should be to make sure that you never have to recalculate the same analysis result twice.
- So, when you run an analysis – especially one that takes time, like tokenisation – you can often benefit by making a new table and storing your analysis results.
- Next time, you just look up the result in the database, rather than having to run the analysis all over again!

3) Back up, back up, and back up some more.

- SQL databases do a lot of clever stuff to keep your data safe… But ultimately, if they’re running on your laptop, they can’t rescue you if you drop the laptop in a lake, or it gets stolen in a bar.

- MySQL provides a command line program called ‘mysqldump’ which will ‘dump’ your databases and tables out into a backup file – which you can then save on a network drive, an external hard disc etc.

- You can download tools to do this automatically every few days. It’s a really, really good idea to do so!