Introduction to Database

Day 1 – Session 1

Module 7

TECH TALENT
ACADEMY

WOMEN IN DATA
ACADEMY

Session Content

- Big Data
- What are Databases
- The Different Types of Databases
- Entity Relationship Modelling
- Normalisation
- Referential Integrity
- Transaction Processing
- ACID
- Record Locking and Data Redundancy

Big Data

“extremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions.”

Big Data

A large amount of data

- No real clear cut definition, more of a buzz word.

However, usually needs to meet 6 criteria

- Volume – Amount of data
- Variety – types of data
- Velocity – the speed at which the data is generated
- Veracity – the degree to which data can be trusted
- Value – commercial value of the data collected
- Variability – the ways in which the data can be used and formatted

Big Data

There are 3 main ways the data can be presented:

- Structured – **data** that adheres to a pre-defined **data** model
- Unstructured – **Unstructured data** is information that either does not have a pre-defined **data** model or is not organized in a pre-defined manner.
- Semi-Structured – **Semi-structured data** is a form of **structured data** that does not obey the tabular structure of **data** models associated with relational databases or other forms of **data** tables

Big Data

Big Data

Why do companies need big data strategies?

- Data is money! – it’s the new oil!
- Allows the company to understand their customers
- Therefore allows the companies to improve process and services
- Which overall improves the experience for customers

Big Data

Question:

1. What examples of Big Data can you think of?
2. Why is this big data?
3. Why do you think the organisations need all that data?
4. How do we process all this data?

WOMEN IN DATA
ACADEMY

Big Data and Machine Learning

Why does machine learning need big data?

- The more data the better
- Data reliance
- Reduce bias
- Learning from experience

Challenges:

- Quality vs quantity
- Privacy

What is Databases?

A database is a way for computers to store information or data

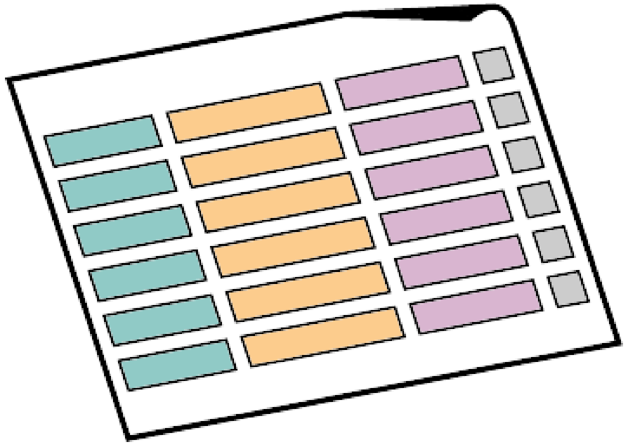
A database stores information in a meaningful way so that it is easy to extract information quickly.

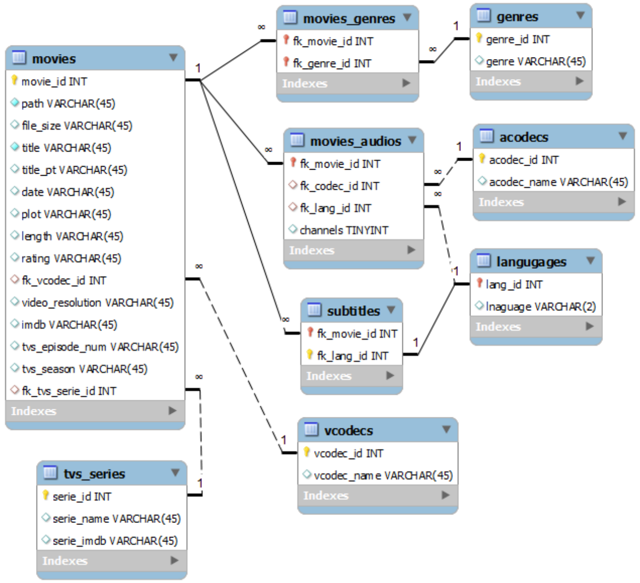
The key here is that the data MUST be organised in a meaningful way – why?

[This Photo](#) by Unknown Author is licensed under [CC BY-NC](#)

Different Types of Databases

- Flat File
- Relational
- NoSQL





movies
movie_id INT
path VARCHAR(45)
file_size VARCHAR(45)
title VARCHAR(45)
title_pt VARCHAR(45)
date VARCHAR(45)
plot VARCHAR(45)
length VARCHAR(45)
rating VARCHAR(45)
fk_vcodec_id INT
video_resolution VARCHAR(45)
imdb VARCHAR(45)
tvs_episode_num VARCHAR(45)
tvs_season VARCHAR(45)
fk_tvs_serie_id INT
Indexes

movies_genres
fk_movie_id INT
fk_genre_id INT
Indexes

genres
genre_id INT
genre VARCHAR(45)
Indexes

movies_audios
fk_movie_id INT
fk_codec_id INT
fk_lang_id INT
channels TINYINT
Indexes

acodecs
acodec_id INT
acodec_name VARCHAR(45)
Indexes

languages
lang_id INT
lnaguage VARCHAR(2)
Indexes

subtitles
fk_movie_id INT
fk_lang_id INT
Indexes

vcodecs
vcodec_id INT
vcodec_name VARCHAR(45)
Indexes

tvs_series
serie_id INT
serie_name VARCHAR(45)
serie_imdb VARCHAR(45)
Indexes

1
∞

Redis
HBASE
Cassandra
mongoDB
riak
CouchDB
relax
Tokyo Cabinet
Scalaris
Project Voldemort
Neo4j
the graph database
membase

Flat File

- All data held in one single table
  - Think a CSV file or filing cabinet
- Small/Simple databases
- Problem with redundancy
  - Update issues
  - Potentially even legal issues

Relational Database

Are a way of splitting the data into several tables and then creating a relationship between them.

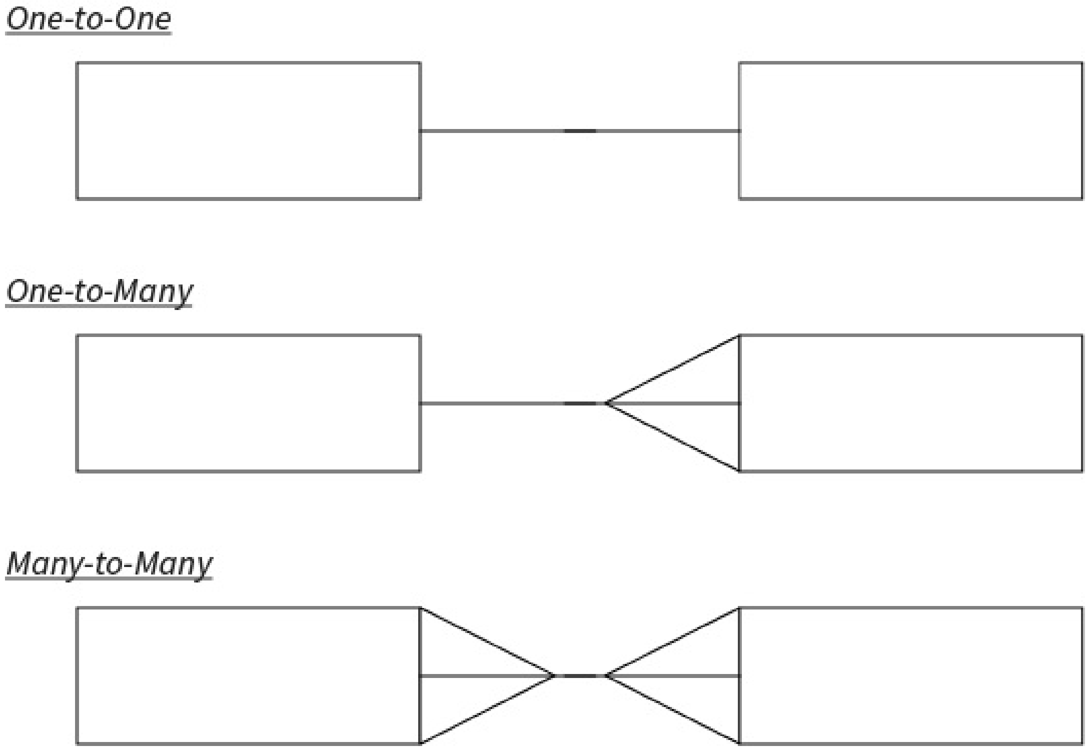
These tables contain primary, foreign and secondary keys.

- Primary Keys (PK): A unique identifier for its own table
- Foreign Key (FK): The primary key of another table in the table (can have multiple)
- Secondary key: any other field within a database that is not a PK or FK

The relationships are created through using PK and FK. By using PK and FK it allows us to also index our day quickly and efficiently and also reduce the repeatability within the database which removes the issue of data redundancy (largely)

However, relational databases are more difficult to create than a flat file and harder to maintain.

Entity Relationship Modelling (ERM)



Normalisation

Normalisation is the process of converting a flat file database (with a single table) to a relational database (with many tables).

There are various levels of normalisation that remove repetition to a greater or lesser extent.

1st Normal Form

A table is in First Normal Form (1NF) if there are **no repeating groups**

In other words, each column must contain only a single value and each row must have an item in every column.

This can usually be done by putting the data into two tables ... separating the repeated data into a separate group.

2nd Normal Form

To move to 2NF, any partial dependencies must be removed

This basically means each record should not have a composite primary key

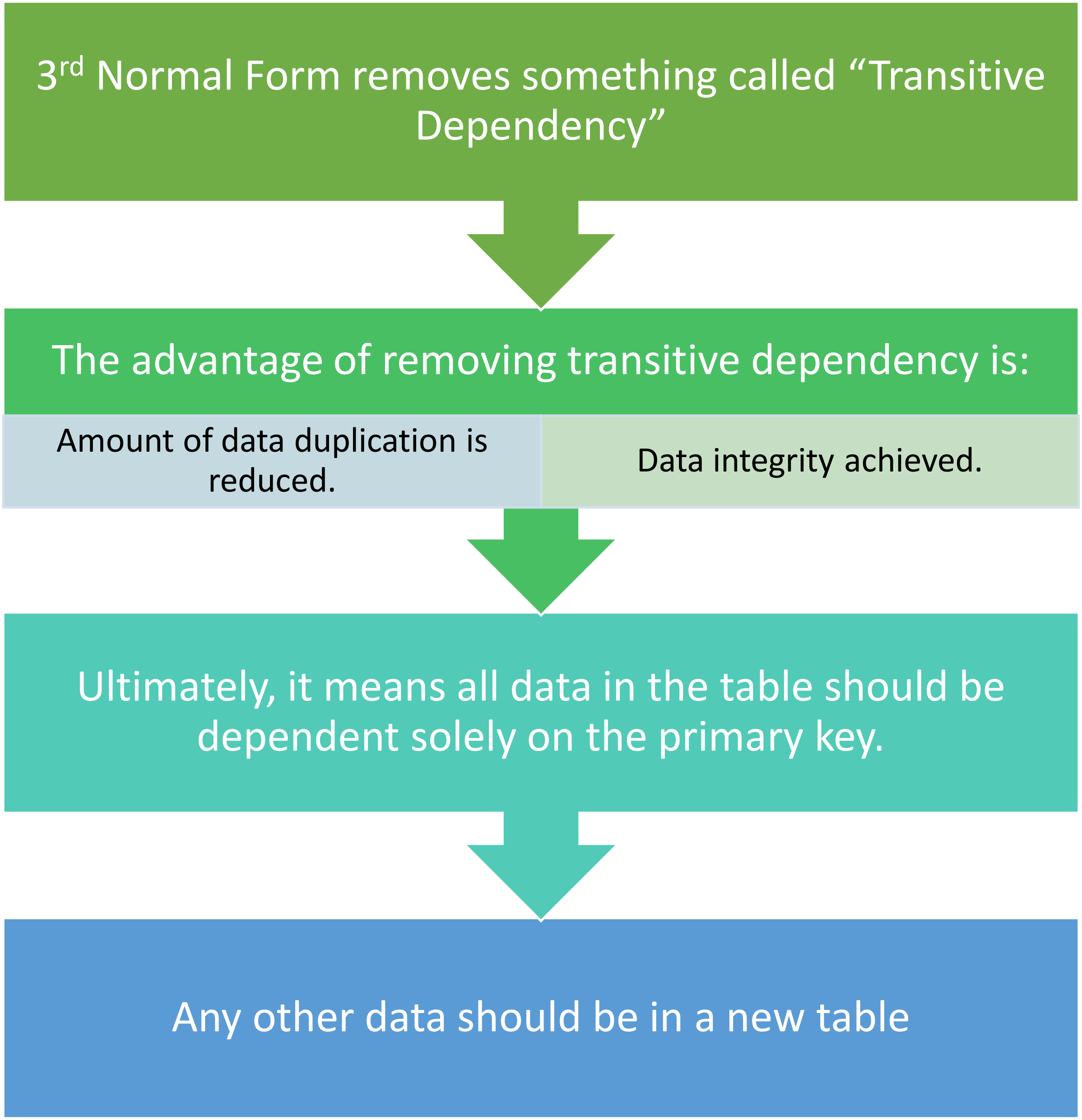
This removes:

Many to many relationships

Repeated Data

WOMEN IN DATA
ACADEMY

3rd Normal Form



3rd Normal Form removes something called “Transitive Dependency”

The advantage of removing transitive dependency is:

Amount of data duplication is reduced.

Data integrity achieved.

Ultimately, it means all data in the table should be dependent solely on the primary key.

Any other data should be in a new table

Referential Integrity

It is always important to maintain the integrity of your data. Therefore there are a few rules your database must follow:

- Transactions should maintain referential integrity –
  - this means keeping a database in a consistent state so changes to data in one table must take into account data in linked tables
  - e.g. you cannot delete data that is linked to existing data in another table.
- Referential Integrity is often enforced by DBMS

Transaction Processing

Transaction processing are changes in the state of a database. These states can be:

- Addition (Create)
- Read
- Alteration (Update)
- Deletion (Delete)
- CRUD

ACID (Atomicity, Consistency, Isolation, Durability)

Transactions must conform to the ACID rules:

- Atomicity: They should either succeed or fail but never partially succeed
- Consistency: The transaction should only change the database according to the rules of the database
- Isolation: Each transaction shouldn’t affect or overwrite other transactions concurrently being processed
- Durability: Once a transaction has been started it must remain no matter what happens

Record Locking and Data Redundancy

Record locking

- Is the technique of preventing simultaneous access to objects in a database in order to prevent updates from being lost
- or inconsistencies in the data arising. A record is locked whenever a user retrieves it for editing or updating. Anyone
- else attempting to retrieve the same record is denied access until the transaction is completed or cancelled, e.g. if one
- transaction is amending a record, no other transaction can until the first transaction is complete.

Record Locking and Data Redundancy

Data redundancy

- Is the unnecessary repetition of data that leads to inconsistencies.
- Data should have redundancy set up, so if part of a database is lost it should be recoverable from elsewhere.
- Redundancy can be provided by RAID setup or mirroring servers.

Reflection

- What is big data?
- What are the different types of databases?
- What are relational databases and how do they differ from flat file databases?
- What are the different ERM?
- What are the stages of Normalisation?
- What is ACID?
- What is record locking and data redundancy?

WOMEN IN DATA
ACADEMY

TECH TALENT
ACADEMY