



**Corndel  
Digital.**

in association with

**Softwire**

## Workshop: Module 09

Data & Security 1

# Agenda

**1000**      **Welcome and Register**

**Part 1: Relational Databases**

- Recap (10 mins)
- New Stuff (20 mins)
- Exercise – Messing about with SQL (60 mins)

**1200**      **Lunch Break (1 hour)**

**1300**      **Part 2: Data Security & Non Relational Databases**

- Recap (30 mins)
- Exercise – Avoiding Prosecution (90 mins)
- Discussion – Choosing Databases
  - In small groups (45 mins)
  - As a larger group (15 mins)

# Part 1

## Relational Databases

# Recap

## Relational Databases

- Relational Databases are the most common form of data storage today.
- They are made up of **tables** that define the structure of the data
- Each table consists of **columns**
- Data is stored as **rows** in the table
- They are usually interacted with using a form of SQL (Structured Query Language)

User ID	Name	Hire Date	Leaving Date	Is Contractor
1	John Smith	2018-02-23	2020-04-12	true
2	Jane Doe	2020-03-04		false

# New Stuff

## Common SQL Statements

The primary statements used for interacting with an SQL database are the DML (Data Manipulation language) operations:

- SELECT
  - Read data from the database
- INSERT
  - Add new row(s) to a table
- UPDATE
  - Update a value on an existing record
- DELETE
  - Remove existing records

```
SELECT *  
FROM Employee;
```

```
INSERT INTO Sale (ProductId, UserName, Amount)  
VALUES (9, 'user-name', 140);
```

```
UPDATE Stock  
SET Price = 13  
WHERE Id = 4;
```

```
DELETE FROM Employee  
WHERE FirstName = 'Lainey'
```

# New Stuff

## Types of SQL

Many relational databases are interacted with using their own extension of SQL, and often these names are shared between the language and the database

Examples:

- MSSQL (Microsoft SQL) uses T-SQL
- MySQL
- PostgreSQL

Each flavour of SQL will support the core SQL commands, but adds their own syntax on top for more powerful capabilities.

PostgreSQL:

```
select extract(dow from date '2016-12-18'); /* sunday */
```

T-SQL:

```
SELECT DATEPART (WEEKDAY, '2021-01-01'); /* 6 */
```

# Exercise

Messing About With SQL

## Exercise – Messing About With SQL

In this exercise we're going to set up a database running in docker on your machine, and explore the data within it

[GitHub Repo Link](#)

Follow the instructions for Workshop 9, Part 1



## Part 2

Non Relational Databases & Data Security

# Recap

## Non-Relational Databases

Various alternatives to relational databases exist for storing data – these are sometimes referred to as NoSQL (even if some of them still support SQL!)

These can include:

- Document Stores
  - Store data as collections of documents in a standard encoding such as JSON
  - Documents don't have a fixed structure
- Key-Value Stores
  - Like a large dictionary or lookup table
  - Lightweight and fast
- Graph Databases
  - Store relationships between data entries

# Recap

## Non-Relational Databases II

### Pros

- Often much easier to scale horizontally (over multiple machines)
- Because of this, often better suited for handling “Big Data”
- **Can** be faster than traditional relational databases

### Cons

- May not offer ACID guarantees – data integrity is less assured
- May put greater burden for validity checks on the application

# Recap

## Sensitive Data

Databases can store a lot of information that users are trusting you to store appropriately and privately, particularly personal or financial data.

Various standards & legislations exist for ensuring that you have adequately considered the implications of data you are storing

- PCI DSS
  - Industry standard for storing payment details
- Data Protection Act (2018)
  - Legal requirements for storing sensitive clinical data
- GDPR (Adopted 2016, in force from 2018)
  - Legal restrictions on how personal data can be processed

**If you can – don't store it!**

# Recap

## Personally Identifiable Information

GDPR discusses personally identifiable information – this can be a subtle topic!

GDPR Article 4 says:

*‘Personal data’ means any information relating to an identified or identifiable natural person (‘data subject’); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.*

Given this, is a date of birth identifiable?

# Exercise

Avoiding Prosecution

# Exercise – Avoiding Prosecution

In this exercise we're going to explore the data security concerns we might have with the database we have set up

[GitHub Repo Link](#)

Part 2

# Discussion

Choosing The Correct Database



# Discussion – Choosing The Correct Database

In this exercise we're going to discuss how we might choose between the many options we have available for our database for a range of scenarios. We'll split into groups to discuss first, and then reconvene to compare our thoughts

[GitHub Repo Link](#)

Part 3

**Thank You!**