A very short introduction to databases

What is a database?

What is a database?

A database is a shared collection of logically related data and its description.

The database represents the entities (real-world things), the attributes (their relevant properties), and the logical relationships between the entities.

Why would you use one?

Exercise 1:

Why use a library when you already have a bookcase?

Exercise 2:

Why use a database when you have a spreadsheet?

Databases help solve these issues:

1. Size of data
2. Ease of updating data
3. Accuracy
4. Security
5. Redundancy

How does it fit together?

Key database concepts

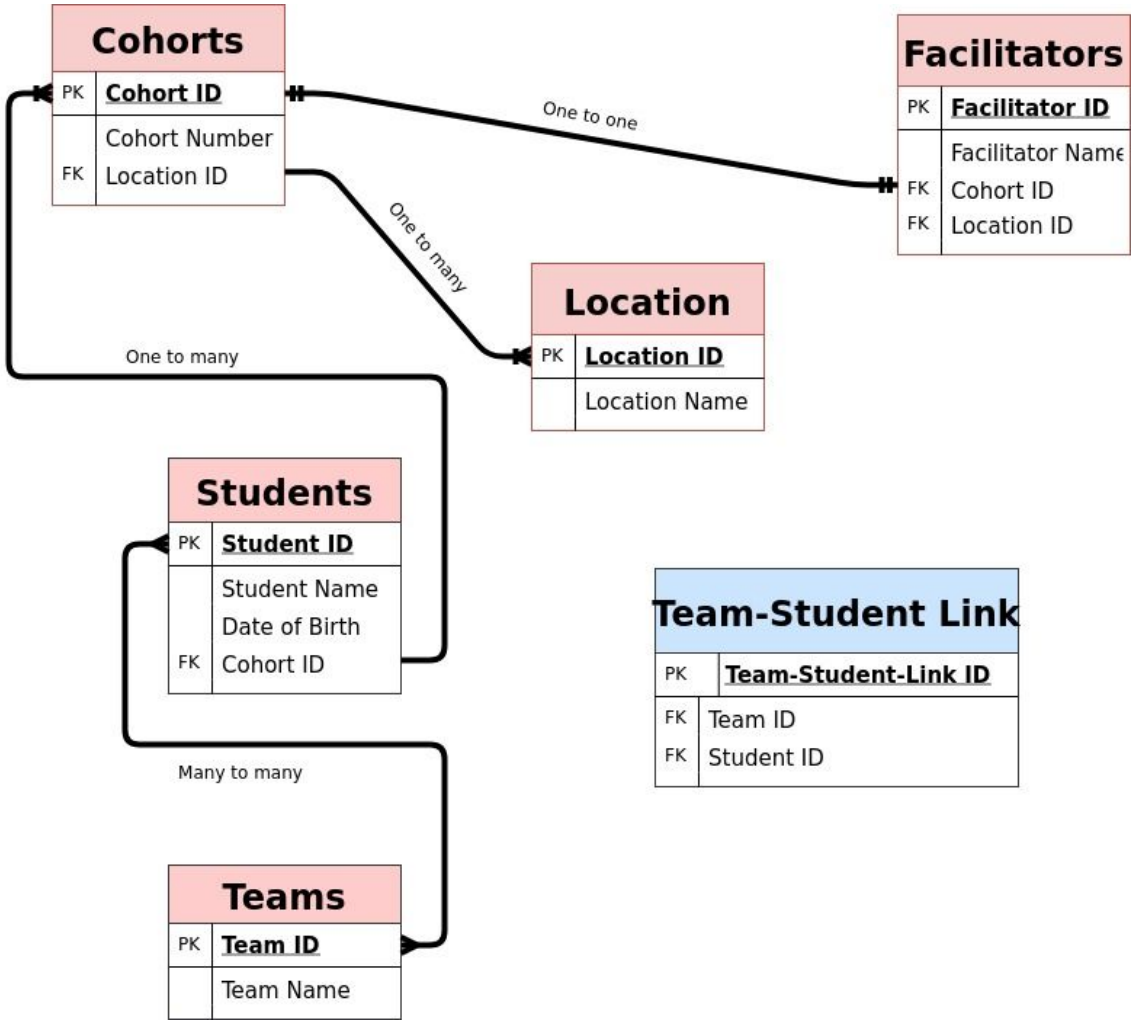
	Real world	**Databases**
Language	English (UK vs US)	SQL (syntax differences)
Database Management System	Library	PostgreSQL, MySQL
Database	Collection of books	Database
Schema	Classification system	Schema
Client	Library terminal, librarian	PSQL, PGCLI

Exercise 3:

What kind of relationships exist between people? What type of relationships are they?

Relationships

Relationship	Type


Cohorts
PK
Cohort ID
Cohort Number
FK
Location ID
One to one
Facilitators
PK
Facilitator ID
Facilitator Name
FK
Cohort ID
FK
Location ID
One to many
Location
PK
Location ID
Location Name
One to many
Students
PK
Student ID
Student Name
Date of Birth
FK
Cohort ID
Team-Student Link
PK
Team-Student-Link ID
FK
Team ID
FK
Student ID
Many to many
Teams
PK
Team ID
Team Name

Primary Key - Foreign Key

In SQL these relationships are defined using **Primary Key-Foreign Key** constraints.

Primary Key

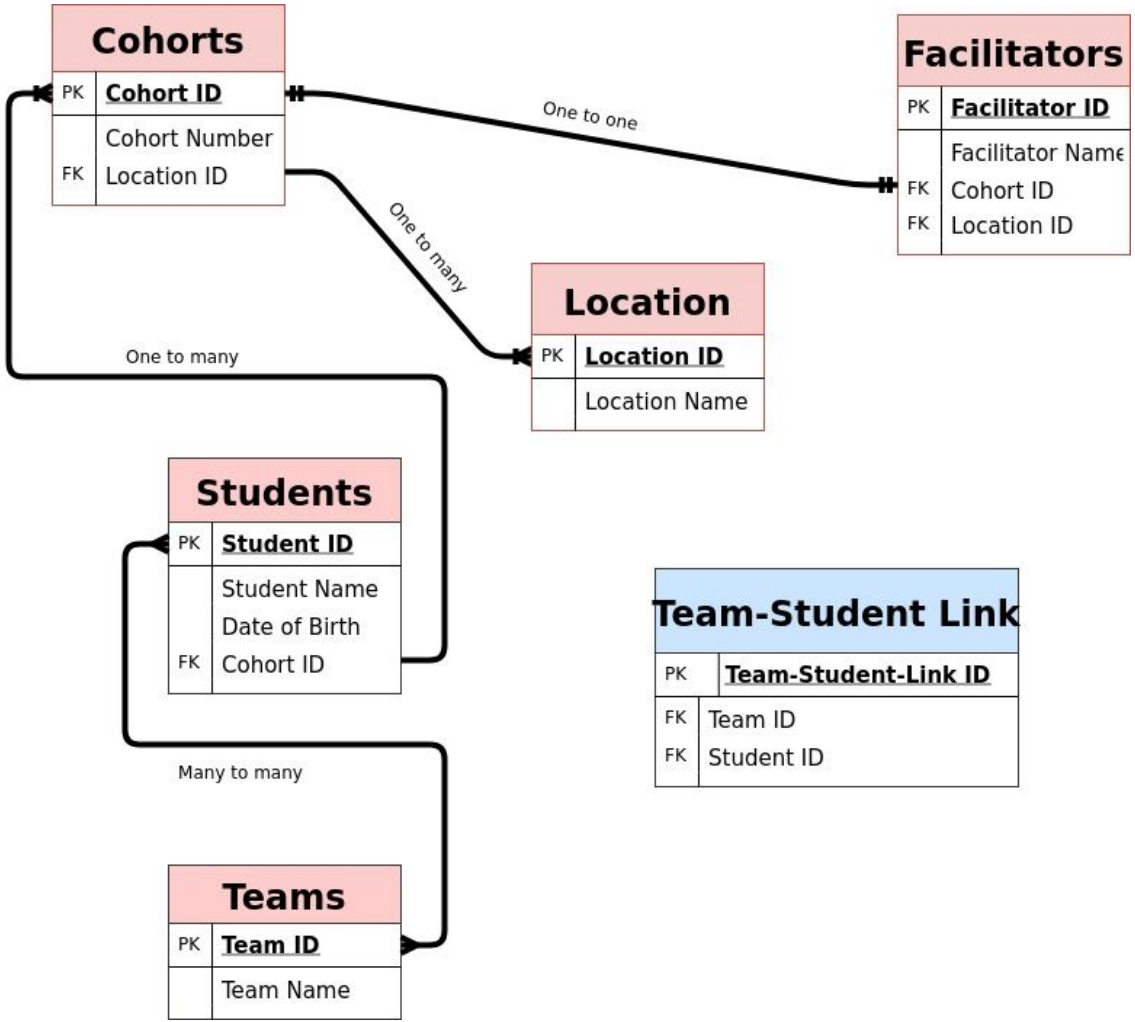
Id	Name	Surname
1	Charles	Dickens
2	Virginia	Woolf

- A primary key is a unique attribute which the database uses to identify a row in a table.
- It is a unique, auto-incrementing ID which is filled in by the database - in other words it is NEVER NULL.
- A primary ID number will only ever be issued once

Foreign Key

Id	Book	Author ID
1	Orlando	2
2	David Copperfield	1

- When we need to refer to a record in a separate table we reference its ID as a foreign key.
- A foreign key is defined in a second table, but it refers to the primary key or a unique key in the first table.


Cohorts
PK
Cohort ID
Cohort Number
FK
Location ID
One to one
Facilitators
PK
Facilitator ID
Facilitator Name
FK
Cohort ID
FK
Location ID
One to many
Location
PK
Location ID
Location Name
One to many
Students
PK
Student ID
Student Name
Date of Birth
FK
Cohort ID
Team-Student Link
PK
Team-Student-Link ID
FK
Team ID
FK
Student ID
Many to many
Teams
PK
Team ID
Team Name

1. One-One Relationship (1-1 Relationship)

One-to-One (1-1) relationship is defined as the relationship between two tables where both the tables should be associated with each other based on only one matching row.

```
CREATE TABLE cohorts
(
Cohort_Id INT PRIMARY KEY,
Cohort_Name VARCHAR(255),
);
```

```
CREATE TABLE facilitators
(
Facilitator_Id INT PRIMARY KEY,
Facilitator_Name VARCHAR(255),
Cohort_Id INT UNIQUE FOREIGN KEY
REFERENCES cohorts(Cohort_Id)
);
```

2. One-Many Relationship (1-M Relationship)

The One-to-Many relationship is defined as a relationship between two tables where a row from one table can have multiple matching rows in another table.

```
CREATE TABLE cohorts
(
Cohort_Id INT PRIMARY KEY,
Cohort_Name VARCHAR(255),
);
```

```
CREATE TABLE students
(
Student_Id INT PRIMARY KEY,
FullName     VARCHAR(255),
Cohort_Id   INT FOREIGN KEY
REFERENCES cohorts(Cohort_Id)
);
```

3. Many-Many Relationship (M-M Relationship)

M-M relationship is defined as a relationship when one or more rows in a table are associated with one or more rows in another table.

```
CREATE TABLE students
(
Student_Id INT PRIMARY KEY,
Cohort_Name VARCHAR(255),
);
```

```
CREATE TABLE teams
(
Team_Id INT PRIMARY KEY,
Team_Name    VARCHAR(255),
);
```

```
CREATE TABLE connections
(
Student_Id INT FOREIGN KEY REFERENCES students(Student_Id),
Team_Id INT FOREIGN KEY REFERENCES teams(Team_Id),
);
```