Databases

Fields, records and primary keys in a single-table database, the data types, and the SQL commands SELECT, FROM, WHERE, ORDER BY, SUM and COUNT.

Single-table database

A database is a structured collection of data that allows people to extract information in a way that meets their needs. A **single-table database** contains only one table.

Fields

Fields represent the individual **columns** in the table, each storing a specific type of information. For example:

1. **StudentID**
   - Data type: integer
   - Description: a unique identifier for each student
   - Validation: must be unique and not null

Records

Records represent the individual **rows** in the table, each containing data for a specific student.

StudentID	Name	Age
1	Hugh Jackman	7
2	Bruce Wayne	8
3	Peter Parker	7

In this table, `Name` is a **field** (a column) and the row for StudentID 2 is a **record**.

Validation

Validation has already been discussed in Chapter 7.

Note Academy

Data Types

Data types are categories of data that tell the computer how to interpret the values stored in variables and how much space to allocate for them in memory. The table below shows the data types included in your syllabus.

Data type	Description
Text / alphanumeric	A number of characters
Character	A single character
Boolean	One of two values: either True or False, 1 or 0, Yes or No
Integer	Whole number
Real	Decimal number
Date / time	Date and time

Primary Keys

A **primary key** is a unique identifier for each record in a database table. It ensures that each record can be uniquely identified and accessed, which is crucial for maintaining data integrity and enabling efficient data retrieval.

For example, in the student table above, `StudentID` will be used as the primary key for that table because it is unique, cannot be null (empty) and cannot be changed.

Structured Query Language (SQL)

Structured Query Language (SQL) is the standard query language for writing scripts to obtain useful information from a database. The single-table database below is what we will be using in the examples ahead.

Table name: `ItemList`

ItemID	ItemName	Price
254	Apple	20
255	Coffee	100
256	Chocolate	50
257	Bread	20

Note Academy

SELECT and FROM

The SELECT and FROM commands are mandatory to use, while the rest are optional.

```sql
SELECT ItemID, ItemName
FROM ItemList
```

Result:

ItemID	ItemName
254	Apple
255	Coffee
256	Chocolate
257	Bread

From the example above, we can deduce that:

- the SELECT command is used to select the columns.
- the FROM command specifies the table to be used within the database.

To select all the columns in the table, use this command:

```sql
SELECT *
FROM ItemList
```

Note Academy

WHERE

The WHERE command is used to specify a condition.

```sql
SELECT ItemName, Price
FROM ItemList
WHERE ItemID = 255
```

Result:

ItemName	Price
Coffee	100

ORDER BY

The ORDER BY command organises the data in ascending or descending order. The example below orders the table in ascending order of price.

```sql
SELECT *
FROM ItemList
ORDER BY Price
```

Result:

ItemID	ItemName	Price
254	Apple	20
257	Bread	20
256	Chocolate	50
255	Coffee	100

To order it in descending order, use this command:

```sql
SELECT *
FROM ItemList
ORDER BY Price DESC
```

SUM

The SUM statement is used to sum all integer or real values in a field.

```sql
SELECT SUM(Price)
FROM ItemList
```

Result:

Note Academy

SUM(Price)
190

COUNT

The COUNT statement counts all the records in a field.

SQL

```
SELECT COUNT(ItemID)
FROM ItemList
WHERE Price = 20
```

Result:

COUNT(ItemID)
2

Note Academy