



You must be able to

- choose a field, a data type and a primary key for a table
- write and read a single-table SQL query
- list the rows a query returns



Method — The table

A **database** is an organised store of data. At IGCSE the data is in one table.

- A **record** is one row: all the data about one thing.
- A **field** is one column: one fact that every record has.

Method — The table

Data type	Used for	Example
text / alphanumeric character	letters, digits and symbols	10A
Boolean	one character	M
integer	one of two values	TRUE or FALSE
real	a whole number	42
date/time	a number with a decimal point	12.50
	a date or a time	14/03/2009

Method — The table

A **primary key** is a field whose value is **unique** in every record. No two rows share it. BookID can be a primary key. Author cannot, because one author can have many books.

Validation checks that a value is sensible before it is stored, for example a range check on a price.

Method — The table

A database is an organised store of data.

field (a column)

record (a row)

StudentID	FirstName	DateOfBirth	FormClass	FeesPaid
1	Amy	14/03/2009	10A	TRUE
2	Ben	02/11/2008	10B	FALSE

primary key
(unique)

Method — The table

A database is an organised store of data.

text	a name
number	an age
date/time	a birthday
boolean	yes / no

Field types: text, number, date/time and Boolean

Method — SQL

SQL (Structured Query Language) picks rows out of the table.
Read a query in this order: FROM, WHERE, SELECT, ORDER BY.

Method — SQL

Keyword	Job
SELECT	which fields to show; * means every field
FROM	which table
WHERE	which rows; join tests with AND or OR
ORDER BY ... ASC	sort smallest or A to Z first
ORDER BY ... DESC	sort largest or Z to A first
COUNT	how many rows match
SUM	add a numeric field

Method — SQL

Worked example. Show the title and price of every Orwell book that is in stock, cheapest first.

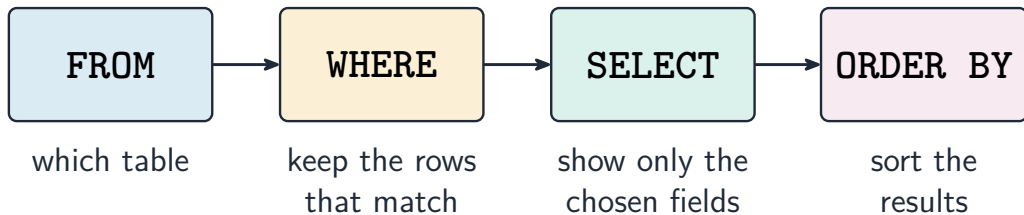
```
SELECT Title, Price  
FROM Book  
WHERE Author = 'Orwell' AND InStock = TRUE  
ORDER BY Price ASC
```

Method — SQL

FROM names the table. WHERE keeps only rows that pass both tests. SELECT shows two fields. ORDER BY Price ASC puts the lowest price first.

Method — SQL

Show the title and price of every Orwell book that is in stock, cheapest first.



Four steps: FROM, WHERE, SELECT, ORDER BY



Practice 2.1 ■ 2 marks

State the difference between a record and a field. Use the table.

Solution 2.1

Method: The table — p.??

Worked steps

- a record is one row, such as all the data about Dune **B1**
- a field is one column, such as Author **B1**

Practice 2.2 ■ 4 marks

Give a suitable data type for Title, Price, InStock and the number of copies of a book.

Solution 2.2

Worked steps

- Title: **text** (B1)
- Price: **real** (B1)
- InStock: **Boolean** (B1)
- number of copies: **integer** (B1)

Practice 2.3 ■ 2 marks

Explain why BookID is a suitable primary key and Author is not.

Solution 2.3

Method: The table — p.??

Worked steps

- every BookID is different, so it identifies one record **B1**
- Author repeats, so it cannot identify one book **B1**

Practice 2.4 ■ 3 marks

Complete the query so it shows the title of every book by Austen.

```
SELECT .....  
FROM .....  
WHERE .....
```

Solution 2.4

Method: SQL — p.??

Worked steps

- SELECT Title **B1**
- FROM Book **B1**
- WHERE Author = 'Austen' **B1**

Practice 2.5 ■ 2 marks

State the difference between `ORDER BY Price ASC` and `ORDER BY Price DESC`.

Solution 2.5

Method: SQL — p.??

Worked steps

- ASC puts the lowest price first **B1**
- DESC puts the highest price first **B1**

Practice 2.6 ■ 2 marks

State the difference between `COUNT(BookID)` and `SUM(Price)`.

Solution 2.6

Worked steps

- COUNT counts the matching rows **B1**
- SUM adds the values in Price **B1**



Exam-style 3.1 ■ 1 mark

The best primary key for the Book table is:

A Author

B InStock

C BookID

D Price

Solution 3.1

A Author

B InStock

C BookID



D Price

Worked steps

C

- BookID is unique for every book **B1**
- Author, InStock and Price can repeat

Exam-style 3.2 ■ 4 marks

Write an SQL query that shows the title and price of every book by Orwell that is in stock, with the cheapest book first.

Solution 3.2

Method: SQL — p.??

Worked steps

- `SELECT Title, Price` **B1**
- `FROM Book` **B1**
- `WHERE Author = 'Orwell' AND InStock = TRUE` **B1**
- `ORDER BY Price ASC` **B1**

Solution 3.2

```
SELECT Title, Price  
FROM Book  
WHERE Author = 'Orwell' AND InStock = TRUE  
ORDER BY Price ASC
```

Exam-style 3.3 ■ 3 marks

State the titles returned by this query, in the order shown.

```
SELECT Title  
FROM Book  
WHERE Price < 10 AND InStock = TRUE  
ORDER BY Title ASC
```

Solution 3.3

Method: SQL — p.??

Worked steps

- WHERE keeps 1984 (price 8.00, in stock) and Animal Farm (price 7.00, in stock) **M1**
- Emma is under 10 but InStock is FALSE, so it is rejected; Dune is 12.50, so it is rejected **M1**
- ORDER BY Title ASC gives **1984**, then **Animal Farm** **A1**

Exam-style 3.4 ■ 4 marks

A shop wants the number of books that are not in stock.

Exam-style 3.4 (a) ■ 3 marks

Write the SQL query.

Solution 3.4 (a)

Worked steps

- `SELECT COUNT(BookID)` **B1**
- `FROM Book` **B1**
- `WHERE InStock = FALSE` **B1**

Solution 3.4 (a)

```
SELECT COUNT(BookID)
FROM Book
WHERE InStock = FALSE
```

Exam-style 3.4 (b) ■ 1 mark

State the number it returns for the table above.

Solution 3.4 (b)

Worked steps

- only Emma is not in stock, so the result is **1** A1