

# 9 Databases

Name: \_\_\_\_\_ Class: \_\_\_\_\_ Date: \_\_\_\_\_

Total: 27 marks

KEY WORDS record • field • primary key • database • query

## Objective

Build the skills to answer exam questions on a **single-table database** : records, fields, data types, a primary key, validation, and SQL queries using SELECT, FROM, WHERE, ORDER BY, AND, OR, SUM and COUNT.

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

## 1 Key knowledge and worked examples

### ■ The table

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

field (a column)  
↓

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

record (a row) →

primary key  
(unique)  
↑

*One row is one record; one column is one field*

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

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

*Choose the type that matches the data*

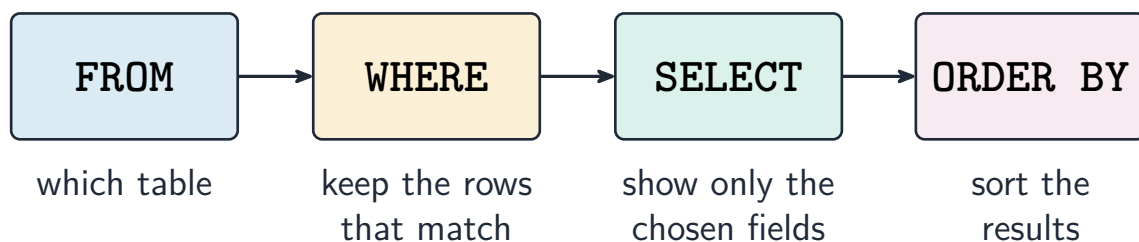
| Data type                     | Used for                      | Example       |
|-------------------------------|-------------------------------|---------------|
| text / alphanumeric character | letters, digits and symbols   | 10A           |
| Boolean                       | one character                 | 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    |

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.

## ■ SQL

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



*FROM chooses the table, WHERE keeps rows, SELECT chooses fields, ORDER BY sorts*

| 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                       |

**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
```

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.

## 2 Practice

Use this Book table.

| BookID | Title       | Author  | Price | InStock |
|--------|-------------|---------|-------|---------|
| B1     | Dune        | Herbert | 12.50 | TRUE    |
| B2     | 1984        | Orwell  | 8.00  | TRUE    |
| B3     | Emma        | Austen  | 9.50  | FALSE   |
| B4     | Animal Farm | Orwell  | 7.00  | TRUE    |

**2.1** State the difference between a record and a field. Use the table. [2]

---



---

**2.2** Give a suitable data type for Title, Price, InStock and the number of copies of a book. [4]

---



---



---



---

**2.3** Explain why **BookID** is a suitable primary key and **Author** is not. [2]

---

---

**2.4** Complete the query so it shows the title of every book by Austen. [3]

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

---

---

---

**2.5** State the difference between **ORDER BY Price ASC** and **ORDER BY Price DESC**. [2]

---

---

**2.6** State the difference between **COUNT(BookID)** and **SUM(Price)**. [2]

---

---

### 3 Exam-style questions

**3.1** The best primary key for the **Book** table is: [1]

- ☐ **A** Author  
☐ **B** InStock  
☐ **C** BookID  
☐ **D** Price

**3.2** Write an SQL query that shows the title and price of every book by Orwell that is in stock, with the cheapest book first. [4]

---

---

---

---

---

**3.3** State the titles returned by this query, in the order shown. [3]

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

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

(a) Write the SQL query. [3]

---

---

---

(b) State the number it returns for the table above. [1]

---

## Solutions

---

Each answer shows the steps, then the **result**.

### 2.1

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

### 2.2

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

### 2.3

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

### 2.4

- `SELECT Title`
- `FROM Book`
- `WHERE Author = 'Austen'`

### 2.5

- `ASC` puts the lowest price first
- `DESC` puts the highest price first

### 2.6

- `COUNT` counts the matching rows
- `SUM` adds the values in `Price`

### 3.1 C

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

### 3.2

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

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

### 3.3

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

### 3.4

(a)

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

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

(b)

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