

# 8 Databases – Part 1

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

## Vocabulary

*Write each word three times.*

| English                   | Write it 3 times (English) |       |       |
|---------------------------|----------------------------|-------|-------|
| relational database table | _____                      | _____ | _____ |
| record                    | _____                      | _____ | _____ |
| field                     | _____                      | _____ | _____ |
| primary key               | _____                      | _____ | _____ |
| foreign key               | _____                      | _____ | _____ |
| referential integrity     | _____                      | _____ | _____ |
| normalisation             | _____                      | _____ | _____ |
| query                     | _____                      | _____ | _____ |
| DBMS                      | _____                      | _____ | _____ |

## Recall

At IGCSE you stored data in a single table of rows and columns. Bring this with you:

- A table holds data in rows (one per item) and columns (one per piece of information).

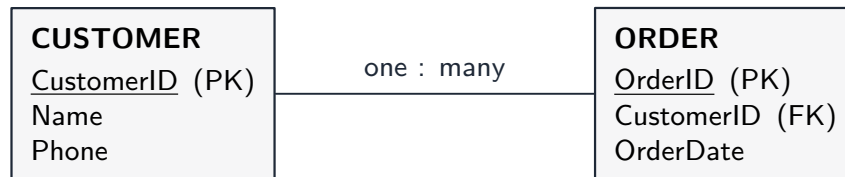
At AS we link several tables together (a relational database) and use a language to ask it questions.

## New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

### ■ 1 Why a relational database

Putting everything in one big flat file repeats the same data many times (wasteful and easy to get wrong). A **relational database** splits the data into linked **table** units. In a table, each row is a **record** and each column is a **field**.

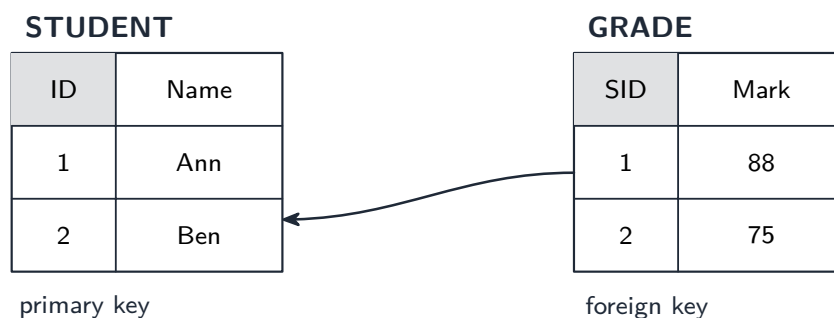


The **foreign key** CustomerID in ORDER refers to the **primary key** CustomerID in CUSTOMER, linking each order to its customer.

*Worked example.* If every borrowing record stored a member's full address, changing one address would mean editing dozens of rows; keeping members in their own table means you change it once.

### ■ 2 Keys

- the **primary key** is the field that gives each record a unique label;
- a **foreign key** is a primary key from another table, used to link the two tables;
- referential integrity** means every foreign key must point to a record that really exists.



Splitting the data neatly so nothing is repeated is called **normalisation**.

*Worked example.* In a **Loans** table, the **MemberID** foreign key matches the **MemberID** primary key in the **Members** table, so each loan links to exactly one member.

### ■ 3 Asking questions with SQL

A **query** asks the database for chosen data. The language is SQL, and a **DBMS** (database management system) runs it. For example:

```
SELECT name, grade
FROM   students
WHERE  grade > 80;
```



*Worked example.* That query returns the **name** of only those students whose **grade** is above 80 – the **WHERE** line filters out the rest.

**Watch out:** `SELECT *` returns *every* column. List just the fields you need (e.g. `SELECT name, grade`) when you don't want them all.

## Practice

8.1 State one problem with storing all data in a single flat file. [1]

---

8.2 Explain what a primary key is. [1]

---

8.3 State the difference between a record and a field. [2]

---

---

8.4 Explain what a foreign key is used for. [2]

---

---

**8.5** Write a SQL query that selects **all** fields from a table called **Books**. [2]

**8.6** Explain *why* splitting data into linked tables makes updating it safer than one big flat file. [2]

**8.7 Challenge.** Write a SQL query that selects the **title** and **author** of books from a **Books** table where the **year** is after 2000. [2]