

Exercise walk-through

Database Concepts ■ 8.1

eXercise

You must be able to

- state the limitations of a **file-based** approach and how a **relational database** 关系数据库 fixes them
- use the terms **table**, **record**, **field**, **primary key** 主键 and **foreign key** 外键
- read an **entity-relationship (E-R) diagram** and explain whether tables are in **3NF**

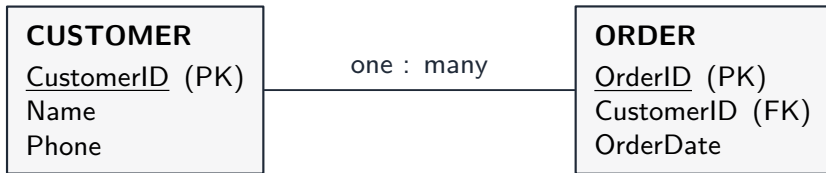
Method — File-based limits → relational database

Storing data in separate **flat files** causes **data redundancy** (the same fact stored in several files), **data inconsistency** (copies updated separately disagree), and **data dependence** (programs tied to the file format). A relational database stores data once in linked **tables** managed by one DBMS, fixing all three.

Method — Relational terms and keys

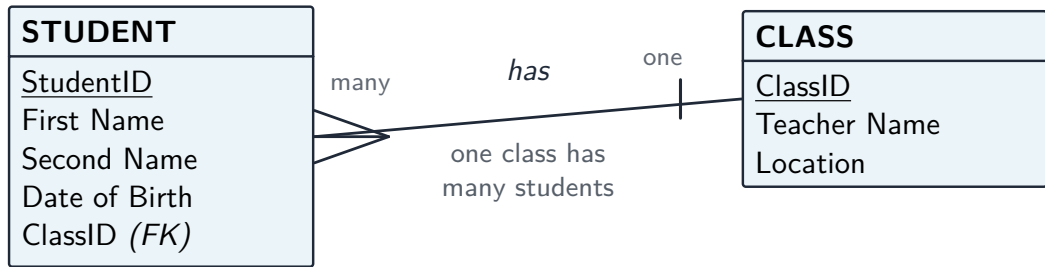
Term	Meaning
table (relation)	a grid for one entity, e.g. CUSTOMER
record (row)	one instance of the entity
field (column)	one piece of data about each record
primary key	field(s) that uniquely identify each record
foreign key	a field that refers to the primary key of another table

Method — Relational terms and keys



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

Method — Relational terms and keys



An entity-relationship diagram models the data

Method — Normalisation (up to 3NF)

Normalisation removes redundancy. A table is in **Third Normal Form (3NF)** when every non-key field depends on **the key, the whole key, and nothing but the key** — i.e. atomic values (1NF), no partial dependency on part of a composite key (2NF), and no field depending on another non-key field (3NF). A **many-to-many** relationship is split using a **link (junction) table** holding two foreign keys.

Method — Relationships and referential integrity

Two tables are linked by a **relationship** — **one-to-one**, **one-to-many** or **many-to-many**. The foreign key is placed on the “**many**” side. An **entity-relationship (E-R) diagram** shows these links.

Referential integrity 参照完整性 means every **foreign key** value must match an existing **primary key** in the other table (or be empty). It stops you recording, for example, an order for a customer who does not exist — the DBMS rejects a foreign key that has no matching record.

Practice 2.1 ■ 2 marks

State **two** limitations of a file-based approach to storing data.

Solution 2.1

Method: File-based limits → relational database

Worked steps

Any two of: data redundancy (same data in several files); data inconsistency (copies disagree); data dependence (programs tied to the file format); weak security/sharing/querying **B1** **B1**.

Practice 2.2 ■ 2 marks

Define **primary key** and **foreign key**.

Solution 2.2

Worked steps

Primary key = field(s) that **uniquely identify** each record; foreign key = a field that refers to the **primary key of another table** **B1** **B1**.

Practice 2.3 ■ 1 mark

State what an **entity** is in a relational database.

Solution 2.3

Worked steps

A **thing** the database stores data about (e.g. a customer, a book) — one table per entity **B1**.

Practice 2.4 ■ 1 mark

State what **First Normal Form (1NF)** requires.

Solution 2.4

Method: Normalisation (up to 3NF)

Worked steps

All values are **atomic** (single values) — no repeating groups in a field **B1**.

Exam-style 3.1 ■ 2 marks

- (a) State **two** problems caused by storing data in separate flat files.
- (b) State how a relational database removes one of them.

Solution 3.1

Method: File-based limits → relational database

Worked steps

- (a) Any two: redundancy; inconsistency; data dependence; poor sharing/security
B1 B1.
- (b) e.g. each fact is stored **once** in a table, so there is no redundancy/inconsistency
B1.

Exam-style 3.2 ■ 3 marks

Define each term: (a) record (b) field (c) primary key.

Solution 3.2

Method: Relational terms and keys

Worked steps

Record = one **row** (one instance of the entity) **B1**; field = one **column** (one piece of data) **B1**; primary key = field(s) that **uniquely identify** a record **B1**.

Exam-style 3.3 ■ 3 marks

Explain what it means for a table to be in **Third Normal Form (3NF)**.

Solution 3.3

Method: Normalisation (up to 3NF)

Worked steps

Every non-key field depends on **the whole primary key** and on **nothing but the key** **B1**; values are atomic (1NF) and there is no partial dependency (2NF) **B1**; no non-key field depends on another non-key field (no transitive dependency) **B1**.

Exam-style 3.4 ■ 3 marks

A school needs to record which students belong to which clubs; a student can join many clubs and a club has many students. Describe how to represent this **many-to-many** relationship using tables.

3.5 (a) State what is meant by **referential integrity**.

(b) A database has a CUSTOMER table and an ORDER table; one customer can place many orders. State the **relationship type** and which table holds the **foreign key**.

Solution 3.4

Method: Relationships and referential integrity

Worked steps

Create a **link (junction) table** (e.g. STUDENT_CLUB) **B1** holding a foreign key to STUDENT and a foreign key to CLUB **B1**; each row records one student-in-one-club, so many-to-many becomes two one-to-many relationships **B1**.