A3 DATABASES · A3.1

Database fundamentals

Why the **relational** model wins over a flat file: **entities**, **primary** and **foreign keys**, relationships, and the benefits and limitations.

01 Key terms

Entity	A table (one type of thing).
Field	A column (an attribute).
Record	A row (a tuple).
Primary key	Unique ID; never empty.
Foreign key	Holds another table's primary key.
DBMS	Software that manages the database.

02 Flat file vs relational

- **Flat file**
  One table, data repeated. Causes redundancy and risks inconsistency.
- **Relational**
  Several linked tables; each fact stored once and joined by keys.

Redundancy Repeating the same data unnecessarily.

Integrity Data stays accurate and consistent.

03 Relationships and cardinality

1:1

One-to-one

One record relates to exactly one other. Example: a person and their passport.

1:N

One-to-many

One record relates to many. Example: one owner with many pets. The most common type.

M:N

Many-to-many

Many relate to many. Example: students and courses. Resolved with a linking table.

04 Benefits

Redundancy	Much less repeated data.
Integrity	Each fact stored once, so consistent.
Scalable	Copes well as data grows.
Security	Access can be controlled per table.
Queries	Powerful joins across linked tables.

05 Limitations

Complex	Harder to design than a flat file.
Speed	Can be slower for very large workloads.
Overkill	A flat file may suit a tiny dataset.
Planning	Needs careful design up front.

06 Worked example · flat file to relational

Owner table	OwnerID (primary key), Name, Phone. Each owner is stored exactly once.	TABLE 1
Pet table	PetID (primary key), Name, Species, and OwnerID as a foreign key.	TABLE 2
The link	OwnerID joins each pet to its owner (one-to-many). Update a phone number in one place, with no inconsistency.	RESULT

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Eight things that lose marks in Paper 1 if you slip on them. Skim before you walk in.

01

A **primary key** is unique and never empty; a **foreign key** is another table's primary key.

02

Entity = table, **field** = column, **record** = row. Use the right word.

03

A **flat file** is one table (repeated data); **relational** uses several linked tables.

04

Redundancy wastes space and risks **inconsistency** when copies disagree.

05

Cardinality: **1:1**, **1:N** (most common), **M:N** (needs a linking table).

06

Benefits: less redundancy, better **integrity** and consistency, scalability.

07

Give **limitations** too: complexity and design effort. Relational is not always best.

08

For full marks, name the keys and state the **cardinality** when describing a relationship.