

Learning Aims

- Explain the concept of a relational database
- Define the terms: flat file, entity, attribute, primary key, foreign key, secondary key, entity relationship modelling, referential integrity
- Produce an entity relationship model for a simple scenario involving multiple entities



A **flat file database** is one that stores all data in a single table

It is simple and easy to understand but causes data redundancy, inefficient storage and is harder to maintain

ID	first_name	last_name	Personal tutor	FormRoom
1	sam	smith	Roger Hinds	6b
2	fred	lynch	Jess Little	8j
3	depak	noor	Roger Hinds	6b
4	archie	henns	Mary Kent	8k
5	helga	jordan	Mary Kent	8k
6	lizzy	bell	Mary Kent	8k
7	xavier	horten	Jack Berry	3m

- This table has redundant data - the tutor and form room information repeats
- This is inefficient
- If a tutor changed their name we would need to find all instances of that name and change them all
- Missing any would mean the **table** had inconsistent data



What is a relational database

- The key difference between a relational database and a flat-file database is that in a relational database the data is **grouped into entities** and stored in **multiple linked tables**.
- It uses keys to connect related data which reduces data **redundancy**, makes efficient use of storage and is easier to maintain



Relational database

- A new **table** could be created to store the tutor information and the tutor information in the student table could be moved to the new **table**. Then a **foreign key** in the student **table** (TutorID) could link a student to their tutor

ID	first_name	last_name	TutorID
1	sam	smith	1
2	fred	lynch	2
3	depak	noor	1
4	archie	henns	3
5	helga	jordan	3
6	lizzy	bell	3
7	xavier	horten	4

- Now the name of each tutor and their form room is stored only once
- This means if they change only one piece of data, the data is updated in the entire **database** and **Inconsistency** is avoided

TutorID	Tutor Name	FormRoom
1	Roger Hinds	6b
2	Jess Little	8j
3	Mary Kent	8k
4	Jack Berry	3m



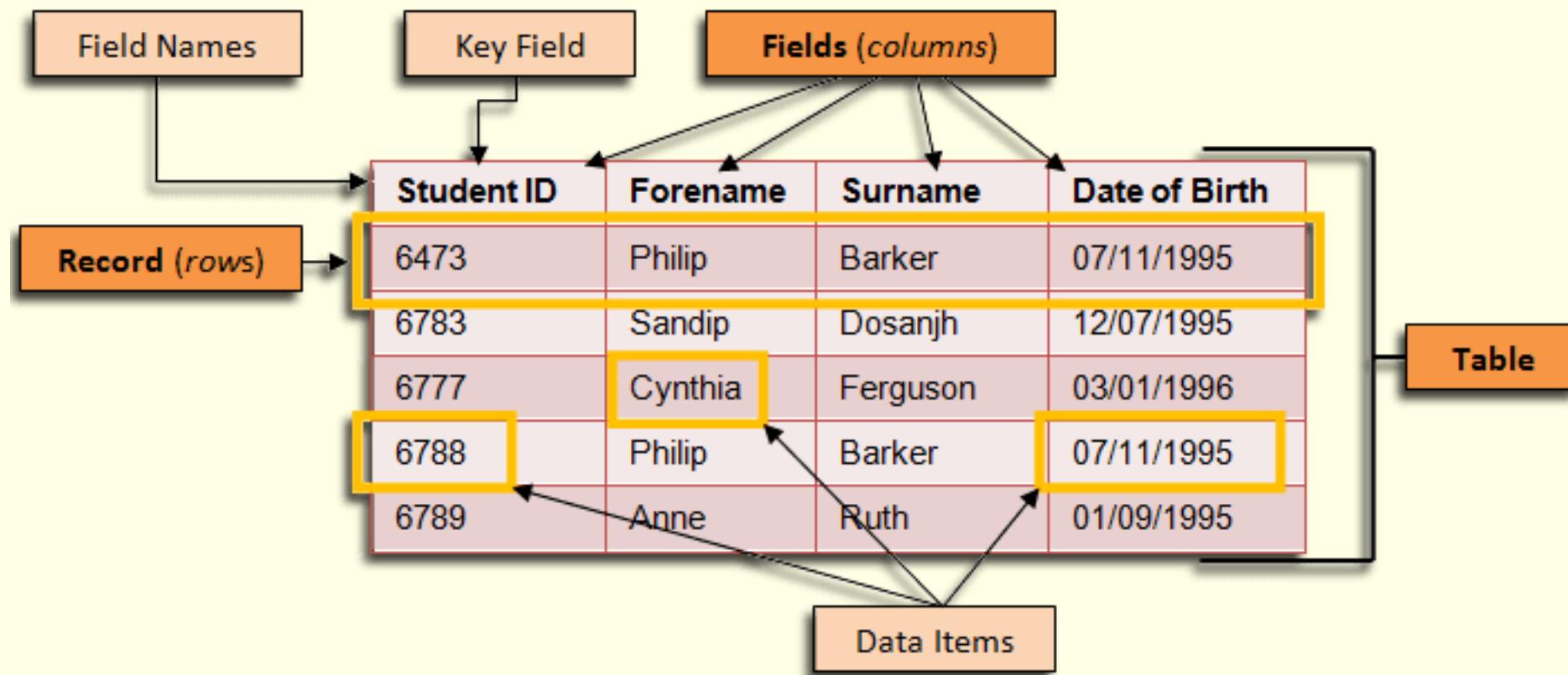
Key terms

- **Entity** - A real-world thing that is modelled in a database. For example, a customer, stock item, sale.
- **Attributes** - Each entity in a database system has attributes, for example: Title, Firstname, Surname



What is a database table?

- In a relational database, **each table represents the attributes of one entity**. Database tables are made up of **record** and **attributes (fields)**.



Key terms

Term	Definition
Field	A single piece of data in a record
Record	A group of related fields, representing one data entry
Table	A collection of records with a similar structure
Primary key	A unique identifier for each record in a table. Usually an ID number
Compound primary key (sometimes called composite)	A combination of (2 or more) fields that is unique for all records
Foreign key	A field in a table that refers to the primary key in another table. Used to link tables and create relationships
Secondary key	A field or fields that are indexed for faster searching
Database Management System (DBMS)	Software used to manage databases. Examples include MySQL, Oracle, Microsoft SQL Server, PostgreSQL



What is an entity and how are they related to database tables?

- An entity is a “real-world thing” about which data can be held in a database.
- In a relational database, each entity corresponds to a separate table in the database.

Examples of entities include:

- Customers
- Products
- Pupils
- Suppliers
- Loans
- Videos/DVD's
- Flights
- Employees
- Treatments
- Contracts
- Library books
- Cars
- Orders
- Zoo animals
- Rentals



What is an attribute and how are they related to database tables?

- Attributes are the facts, aspects, properties, or details about an entity.

Examples of attributes include:

Entity	Library books	Flights	employee
Attributes	ISBN number author category	Flight No aircraft type departure Arrival date/time destination	Name Gender address DOB qualifications job title

In a database, each attribute corresponds to a separate field in the database.



Entity descriptions

An entity description is normally written using the format

Entity1 (Attribute1, Attribute2...)

The entity description for Dentist is therefore written:

Dentist (Title, Firstname, Surname, Qualification)



Entity identifier and primary key

Each entity needs to have an **identifier** which uniquely identifies the entity. In a relational database, the entity identifier is known as the **primary key** and it will be referred to as such in this section.

Clearly none of the attributes so far identified for **Dentist** and **Patient** is suitable as a primary key.

A numeric or string ID such as D13649 could be used.

In the entity description, the primary key is underlined:

Dentist (DentistID, Title, Firstname, Surname, Qualification)

Is National Insurance Number a suitable primary key for Patient? If not, why not?

NI number is not a suitable primary key because many patients may not know their NI number and some patients may not have an NI number, e.g. if not British.



What is a primary key and how are they used to link database tables?

- Primary key is used to store an attribute that makes that particular entity entry in the database unique.

For example:

- NHS number
- passport number
- vehicle registration
- booking reference
- flight number



Secondary key

A database needs to be set up so that it can be searched quickly. An **index** of all the primary keys in the database, and where the record is held, is automatically maintained by the database software.

However, more than one index may be needed.

If for example a patient rings up to make an appointment with the dentist, they are unlikely to know their patient ID, **A secondary index** on surname is likely to be held.



Rules

Rules:

- Each field in a table has a **unique name**.
- Each field stores a **single item of data** – For example, a field called Date of Birth would store no more than one date of birth value.
- Each field has a particular **data type** – for example, text, Boolean, integer, date/time, etc.
- Each field can have its own **validation rules** – these ensure that data recorded meets certain rules.

Example

Here is part of a data table. It is designed to store details of hotel-room bookings. It shows three rows and four columns.

room_number	date	room_type	customer_ref
101	21/03/2015	double	26335
310	22/03/2015	single	45335
250	23/03/2015	double	36587

Note that a combination of room number and date is sufficient to make a primary key field. Many tables make use of a special reference such as student_number to produce a key field.



Relationships

- The tables of a relational database are linked through relationships.
- **The tables in a relational database are linked by the Primary key in one table being added as a foreign key in another table to form a relationship between the entities.**

Example

Here, the field customer_ref forms the primary key in **TblCustomer**, but is a foreign key in **TblRoom**. It allows a relationship to link the tables.

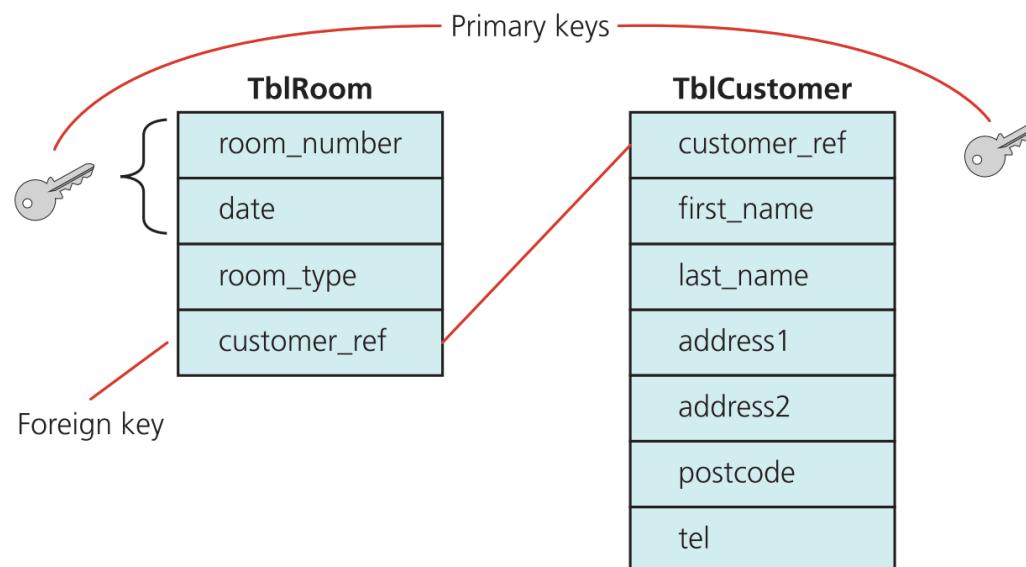


Figure 15.2 Relational database



Relationships between entities

The different entities in a system may be linked in some way, and the two entities are said to be related.

There are only three different 'degrees' of relationship between two entities.

A relationship may be

One-to-one Examples of such a relationship include the relationship between Husband and Wife, Country and Prime Minister.

One-to-many Examples include the relationship between Mother and Child, Customer and Order, Borrower and Library Book.

Many-to-many Examples include the relationship between Student and Course, Stock Item and Supplier, Film and Actor.



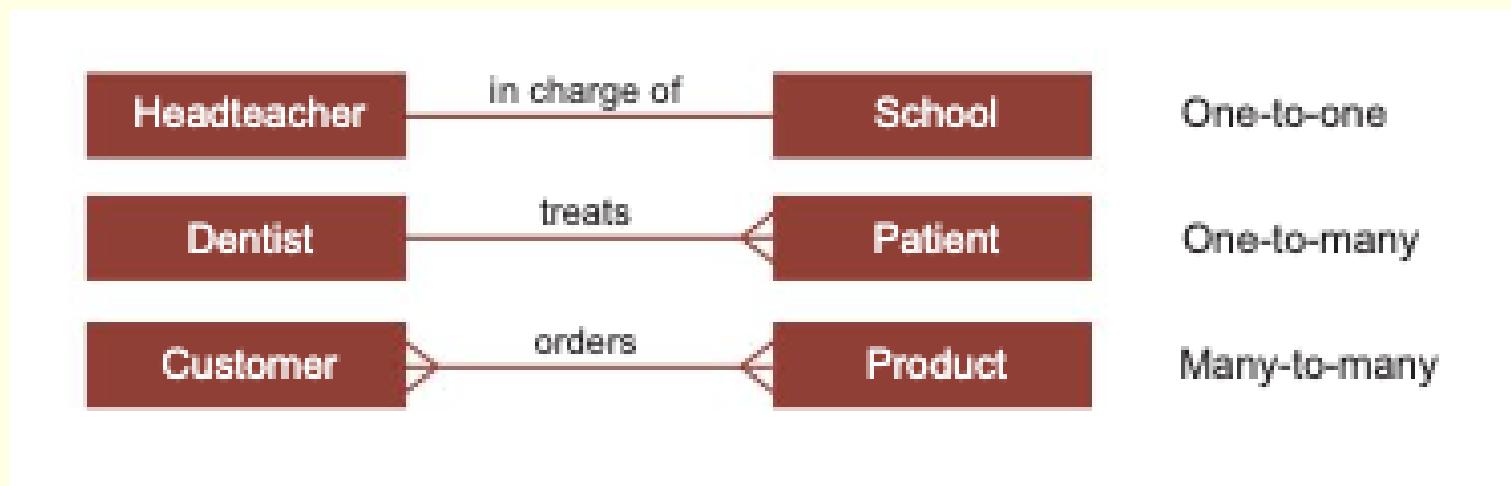
Entity relationship modelling

The relationship between entities can be modelled graphically.

An entity relationship diagram is a diagrammatic way of representing the relationships between the entities in a database.

To show the relationship between two entities, both the degree and the name of the relationship need to be specified.

In the first relationship shown below, the degree is one-to-one, the name of the relationship is in charge of.



The concept of a relational database

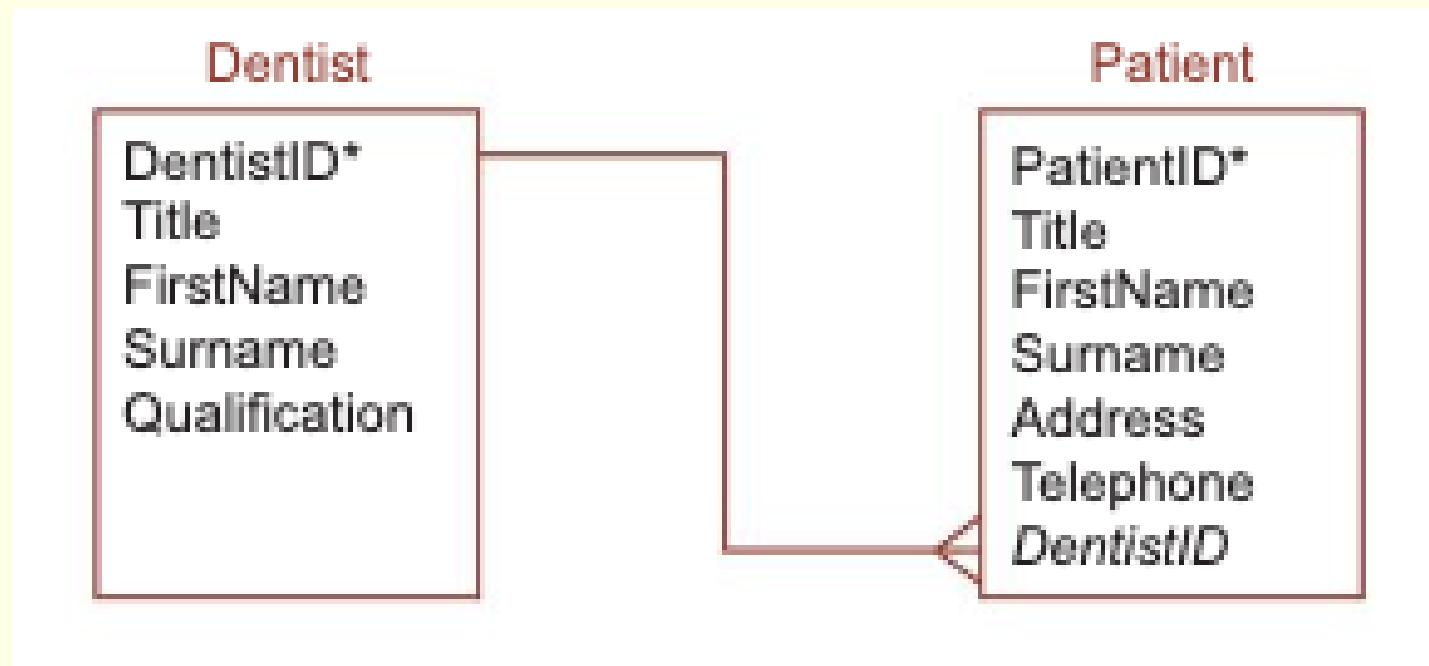
- In a relational database, a separate table is created for each entity identified in the system.
- Where a relationship exists between entities, an extra field called a **foreign key** links the two tables.



Foreign key

A foreign key is an attribute that creates a join between two tables. It is the attribute that is common to both tables, and the primary key in one table is the foreign key in the table to which it is linked.

Example: In the one-to-many relationship between Dentist and Patient, the entity on the 'many' side of the relationship will have **DentistID** as an extra attribute. This is the foreign key.

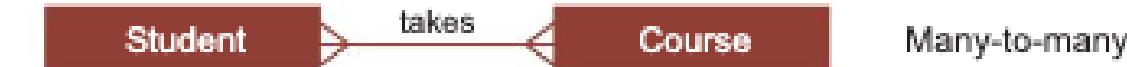


Linking tables in a many-to-many relationship

When there is a many-to-many relationship between two entities, tables **cannot** be directly linked in this way.

For example, consider the relationship between Student and Course.

A student takes many courses, and the same course is taken by many students.



In this case, an extra table is needed to link the Student and Course tables. We could call this StudentCourse, or Enrolment, for example.



The three tables will now have attributes something like those shown below:

Student (StudentID, Name, Address)

Enrolment (StudentID, CourseID)

Course (CourseID, Subject, Level)

Composite key

In this data model, the table linking **Student** and **Course** has two foreign keys, each linking to one of the two main tables. The two foreign keys also act as the primary key of this table.

A primary key which consists of more than one attribute is called a composite primary key.

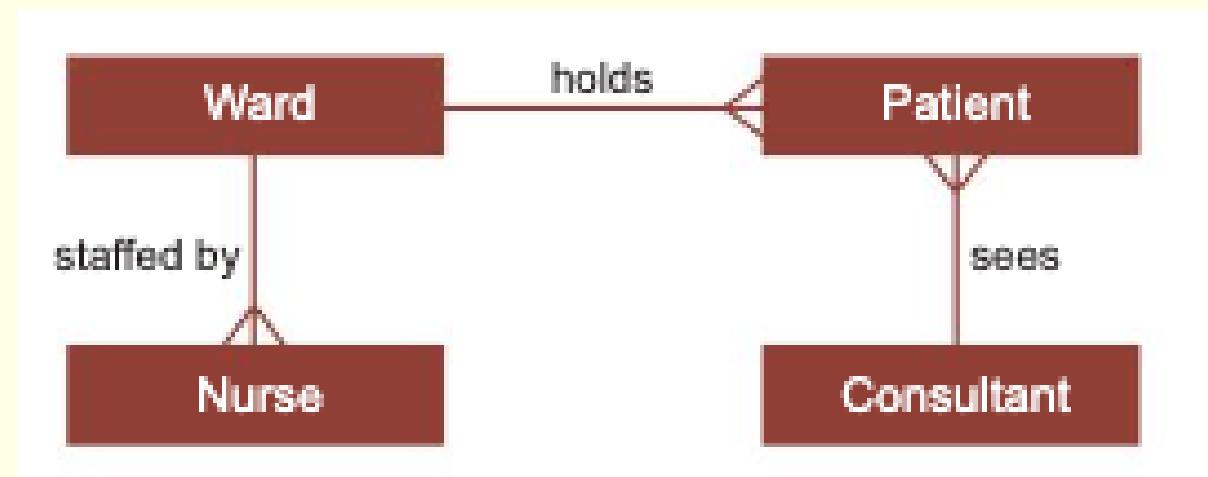


Drawing an entity relationship diagram

A database system will frequently involve many different entities linked to each other, and an entity relationship diagram can be drawn to show all the relationships.

A hospital inpatient system may involve entities Ward, Nurse, Patient and Consultant. A ward is staffed by many nurses, but each nurse works on only one ward.

A patient is in a ward and has many nurses looking after them, as well as a consultant, who sees many patients on different wards.



What are the advantages of relational databases?

- **No data redundancy** – in a well-designed relational database there should be no duplicated data (other than the key field).
- **No data inconsistency** – as data is not duplicated, there is no risk of the same data item being stored differently in another record.
- **Flexibility** – A relational database can be queried with greater flexibility than a flat-file system. Relationships mean that data can be combined in a variety of ways to produce the views that different areas of an organisation require.



Explorer task

What is the suitability of flat files and relational databases for use by a family at home and for use in a large mail order company.

Flat files

Limited amount of data

Limited technical expertise
available in family

Data format difficult to change

Security not a major issue for
family compared with company

Relational database

Large volume of data for company

Requires technical knowledge

Easy to add data and search for data

Easy to link to other applications / e.g. address labels

Saves space / reduces data duplication / redundant
data

Improves data consistency / integrity

Easy to change data format

Improves security / easy to control access to data



SUMMARY:

- The use of **foreign keys** enables tables to be **linked** to form relationships. A foreign key is a field in a table that is **also** a **key field** in another table.
- The relationships remove **data redundancy**, i.e. the need for data to be duplicated (as happens in the case in flat-file databases).
- To create a relationship between two tables, we use the **key field** (primary key) from one of the tables and place it as a **foreign key** in the other table.
- A **key field** is a field that is guaranteed to have a **unique** value for each record in the table.
- **Foreign key** relationships are used to ensure the **referential integrity** of data in a database.



Referential integrity

What is Referential Integrity?

- Ensures **consistency** between related tables in a relational database
- Maintains valid **relationships** between **primary** and **foreign** keys
- There should not be foreign keys for which a matching primary key in the linked table does not exist

Foreign key constraints

- Value in a foreign key field must either:
 - Match a primary key value in the related table, or
 - Be **null** (if allowed)
- Enforce **referential integrity**

Maintaining referential integrity

- Use database management systems (DBMS) with **built-in support**
- Implement triggers to **enforce** custom referential integrity rules
- Regularly **validate** and clean up data to ensure **consistency**

Benefits and Drawbacks of Referential Integrity

Benefits	Drawbacks
Ensures data consistency and accuracy	Can impact performance due to additional checks
Prevents orphaned records	May require additional planning and design