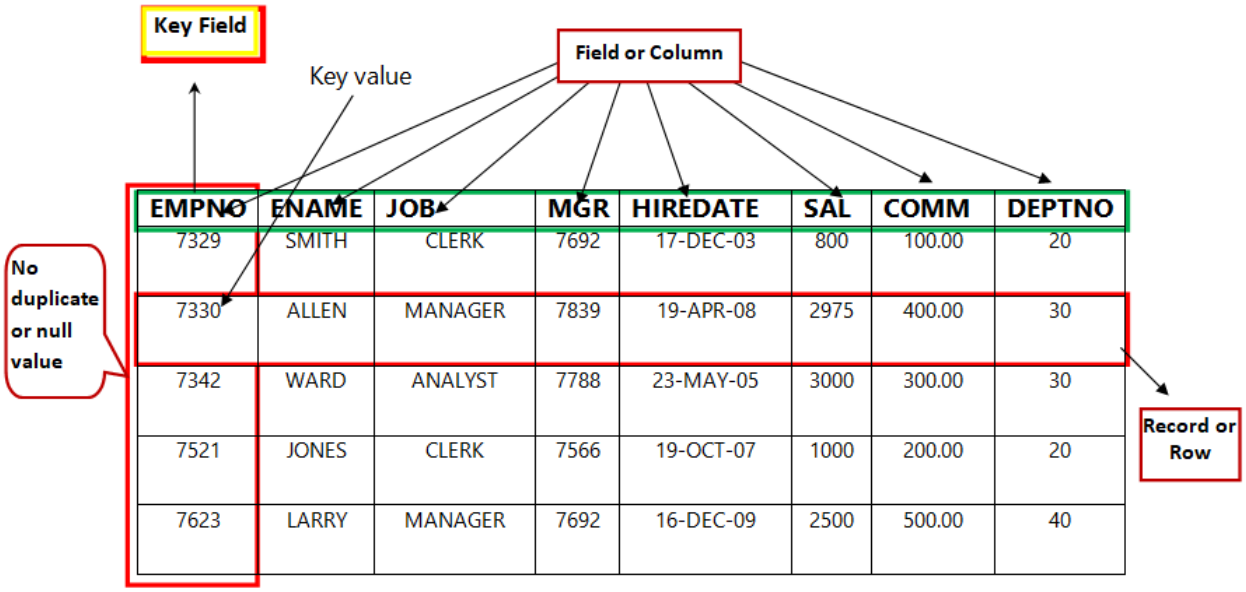
Creating Arrays of Records

```
class CustomerType
{
        String name;
        int id;
}

public class ExampleRecords
{
    public static void main()

    {
        CustomerType[] customerList = new CustomerType[5];
        quickCreate(customerList);

        printList(customerList);

    }
    //END OF MAIN

    //=========================================================
    public static void printList(CustomerType[] arr)
    {
        for (int i=0;i<=arr.length-1;i++)
        {
           System.out.print(arr[i].name+", " + arr[i].id+", ");

        }
        System.out.println();
    }

    //=========================================================
    public static void quickCreate(CustomerType[] arr)
    {
        for(int i=0;i<arr.length;i++)  arr[i] = new CustomerType();

        arr[0].name="a";
        arr[0].id=1;
        arr[1].name="b";
        arr[1].id=2;
        arr[2].name="c";
        arr[2].id=3;
        arr[3].name="d";
        arr[3].id=4;
        arr[4].name="e";
        arr[4].id=5;

    }

}
```



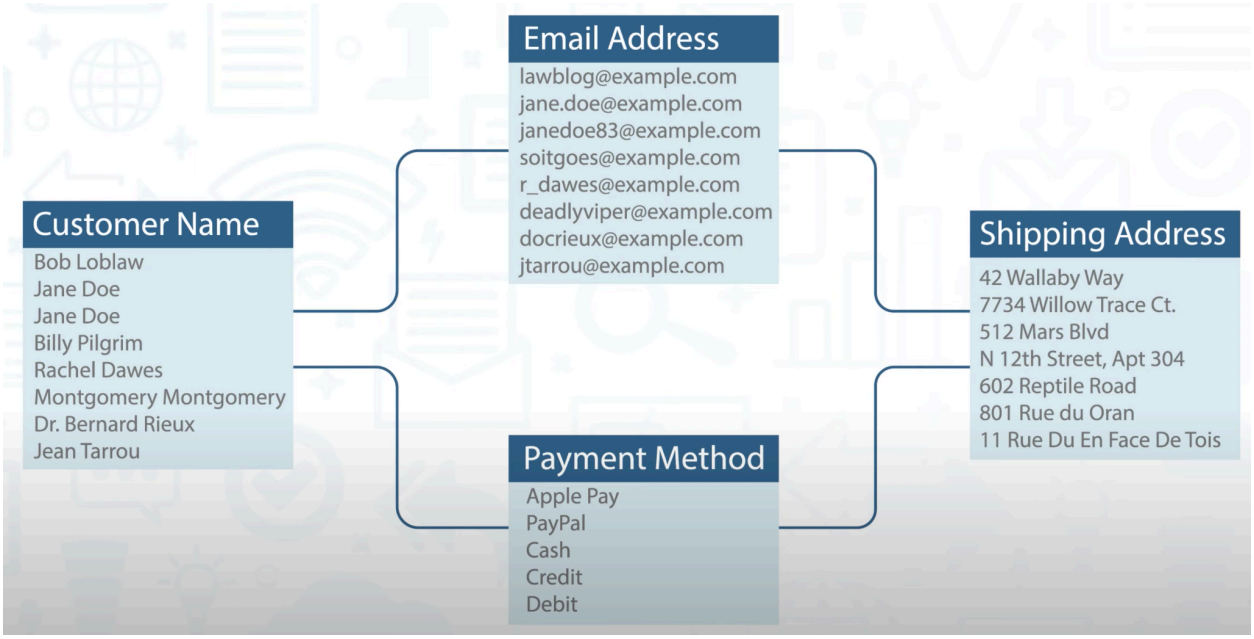
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7329	SMITH	CLERK	7692	17-DEC-03	800	100.00	20
7330	ALLEN	MANAGER	7839	19-APR-08	2975	400.00	30
7342	WARD	ANALYST	7788	23-MAY-05	3000	300.00	30
7521	JONES	CLERK	7566	19-OCT-07	1000	200.00	20
7623	LARRY	MANAGER	7692	16-DEC-09	2500	500.00	40

Related Content: Relational Databases

Notice: For extension only as it is relevant to organizing data. This is not directly on our syllabus content as we study IB Option D

Video tutorial: [What is a relational database and why we need it](#)?

Keywords: Relational Database, Spreadsheet, MS Access, data redundancy, data integrity.



Data integrity defines a data set that is both **valid** and **accurate**. All data entering a system has to be checked for validity and accuracy.

Flat-File Model

Flat-File (one table)

Patient Id	Name	D.o.B	Gender	Phone	Doctor Id	Doctor	Room
134	Jeff	4-Jul-1993	Male	7876453	01	Dr Hyde	03
178	David	8-Feb-1987	Male	8635467	02	Dr Jekyll	06
198	Lisa	18-Dec-1979	Female	7498735	01	Dr Hyde	03
210	Frank	29-Apr-1983	Male	7943521	01	Dr Hyde	03
258	Rachel	8-Feb-1987	Female	8367242	02	Dr Jekyll	06

The advantages and disadvantages of a flat file:

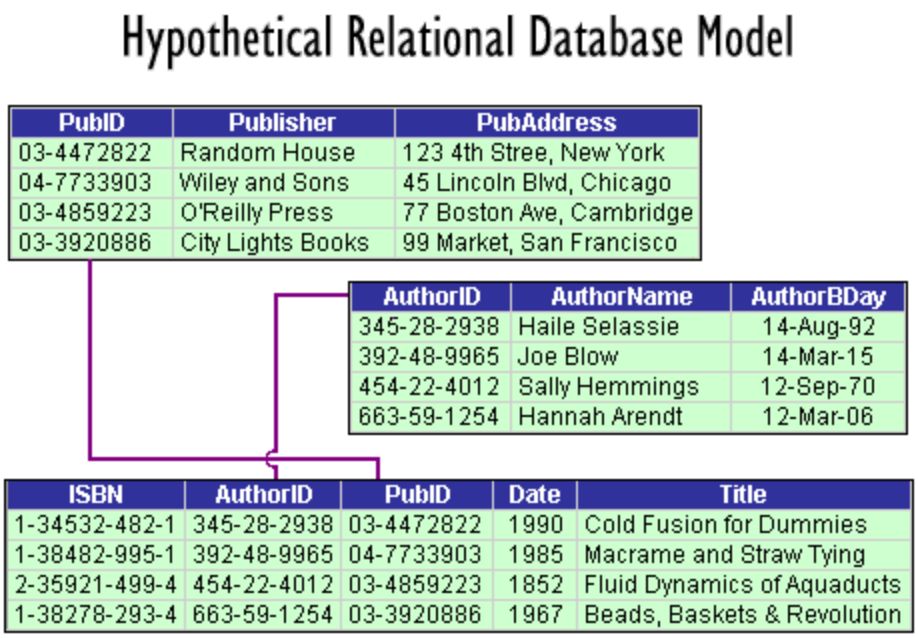
1. All records are stored in one place
2. Easy to understand
3. Simple sorting and filtering can be carried out
4. Can use a standard spreadsheet to store it
5. you can use a single table to store it

The disadvantages of flat file:

1. They can suffer from data **redundancy** – duplication of data. In the example the fields Doctor and Room are duplicated for each record.
2. Any update (add, edit, delete) processes would be unnecessarily complex and would require more steps which may be more processor intensive or hardware intensive (disks / file storage).
3. There is a problem with data **integrity** – if a patient changes doctor you may have errors in your database regarding correct values for doctor room numbers. This will impact upon the integrity of your database. Redundant data can also impact upon data integrity. For example. adding a new patient may result in a lack of **data consistency** as it may not be clear what room the doctor is in when entering patient data. The doctors room number may change and it is not related to patient personal info.

4. Slow for huge database.
5. Searching process is time consuming.

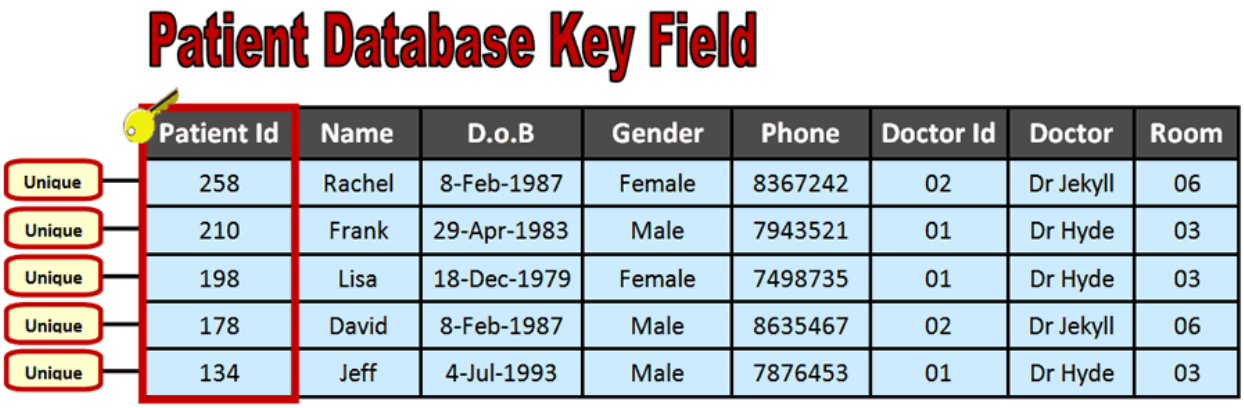
Relational Database Model (structured)



1. **Data Integrity**: You only need to change the data in one of the tables when updating the database.
2. **Data Redundancy**: A relational database ensures that no attributes are repeated.
3. **Data Consistency**: There is no chance of the same attribute being stored in a different format in a different file
4. **Data Flexibility**: Processing is faster. When dealing with queries, it is much easier to complete a data search with a relational database.
5. **Greater Efficiency**: As you only have to input the data only once into a relational database it saves time and human resources.

Definition of Key fields

A **field** in a record that holds unique data which identifies that record from all the other records in the file or database. Account number, product code and customer name are typical **key fields**. As an identifier, each **key** value must be unique in each record.



A **relational database** has more than one table and the tables are linked using **key fields**. For example, a library database could have three tables:

1. **customer** - when a customer joins the library a record is created. It stores their details such as their first name and surname and includes a unique Customer ID.
2. **book** - each book in the library has a record. It stores details about the book, such as the author and title and includes a unique book ID.
3. **lending** - when a customer borrows a book, the lending table stores the customer's unique ID and the book's unique ID in a record. The record could also include additional information such as when the book was borrowed and when it's due back.

The customer and book ID are both examples of key fields.

Advantages

- The book's details and the customer's details need only be entered into the database once.
- Because of this, mistakes are less likely to happen and if there were a mistake in a customer's record, for example, correcting it will correct the mistake database-wide.
- Duplication is avoided - this keeps the database's file size down.
- Details about books and customers are easily accessible using their unique IDs.
- Queries can be performed and reports generated, eg a list of books a customer has borrowed since joining the library.

Why use a database?

- Databases can store very large numbers of records efficiently (they take up little space).
- It is very quick and easy to find information.
- It is easy to add new data and to edit or delete old data.
- Data can be searched easily, eg 'find all Ford cars'.
- Data can be sorted easily, for example into 'date first registered' order.
- Data can be imported into other applications, for example a mail-merge letter to a customer saying that an MOT test is due.
- More than one person can access the same database at the same time - multi-access.
- Security may be better than in paper files.

Database uses

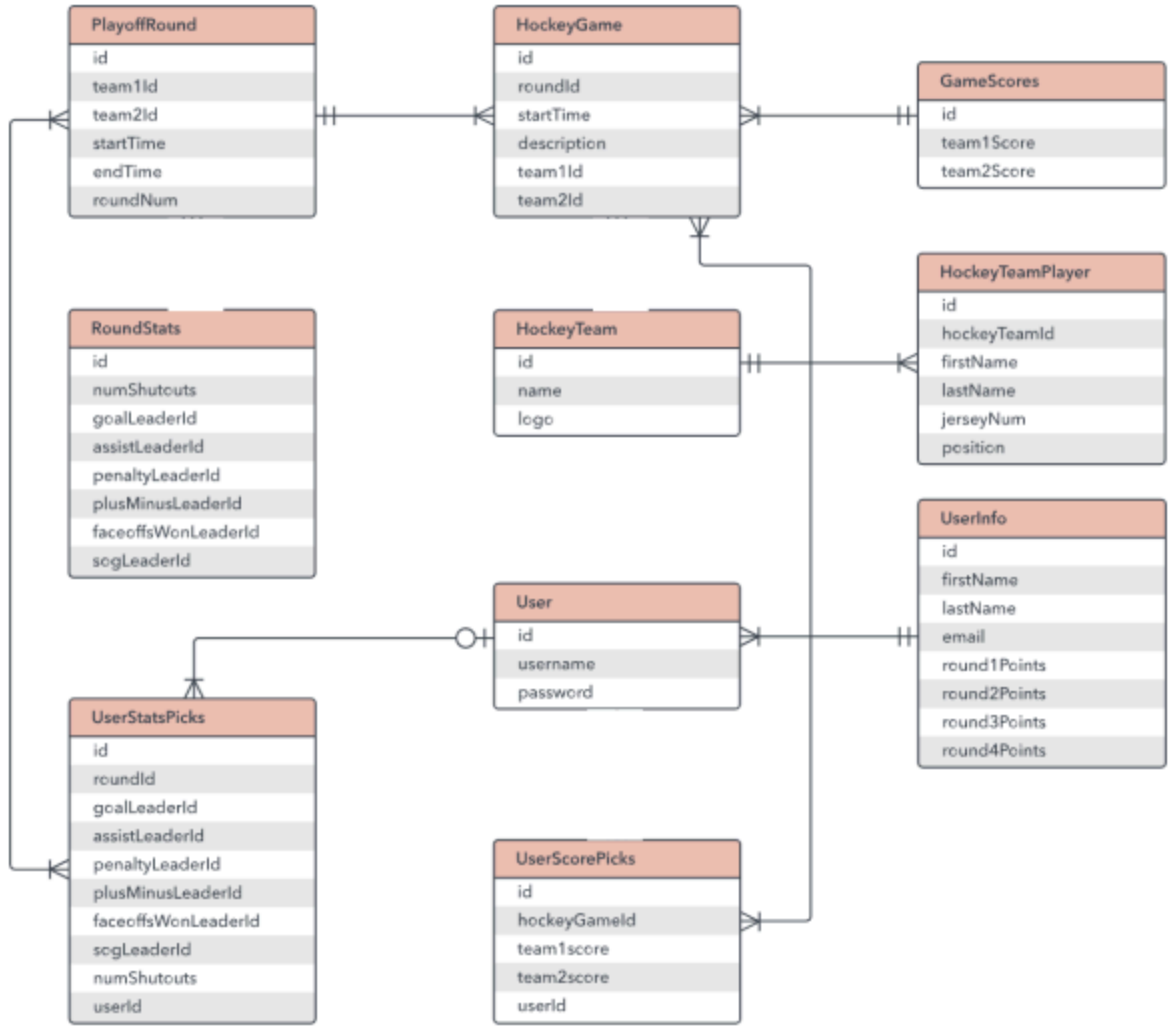
When a database holds details about people, it's likely to include their first name, surname and their date of birth. In addition to this, specialist information is stored depending on the database's intended use.

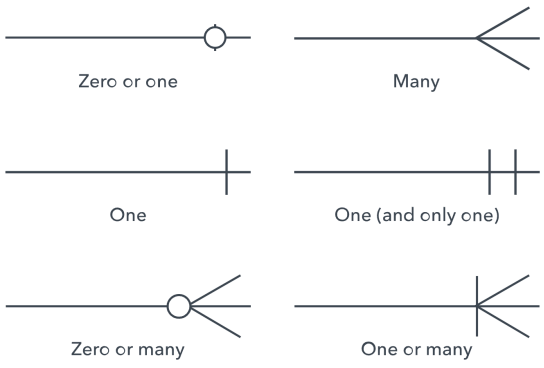
- The police have details of all known criminals in a database, eg crimes they've committed.
- Schools use a database to store details about their pupils, eg how many days they've been off school sick.
- A hospital will store details of all its patients in a database, eg a history of their health issues.
- The Government uses a database to store records of people's income tax payments.
- A database is used to keep track of all the drivers in central London who have (or haven't) paid the Congestion Charge.

What is an ER diagram?

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships

and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.





What would the ER relationship styles for our previous example?

PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Stree, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sep-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1852	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution