|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The Club ID, Club Name and Stadium contain information that is repeated more than once. To remove this, we can split the table into two separate tables: **Player** and **Club**.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Player Table** | | | | | | **Player ID** | **First Name** | **Surname** | **Squad No.** | **Club ID** | | P101 | James | McGinn | 2 | C100 | | P201 | Graeme | Rooney | 7 | C100 | | P103 | Darren | May | 2 | C201 | | P102 | Simon | Shinnie | 10 | C201 | | P001 | Sean | Ball | 18 | C201 | | P200 | Billy | Logan | 7 | C102 |  |  |  |  | | --- | --- | --- | | **Club Table** | | | | **Club ID** | **Club Name** | **Stadium** | | C100 | Rovers | Town Park | | C201 | United | ABC Arena | | C102 | Central | Rose Lane |   The primary key for the Player table is **Player ID** because it is unique.  The primary key for the Club table is **Club ID** because it is unique (it is possible for teams from other countries to share the same name; it is possible for two teams to share a stadium).  A foreign key has to be a primary key in another table – in this case **Club ID** is used as the foreign key to link the two tables together (this field appears in both tables).  If you designed a query to search for the first name and surname of all players who played with Rovers it would be:   |  |  |  |  | | --- | --- | --- | --- | | **Field Name** | First Name | Surname | Club Name | | **Table Name** | Player | Player | Club | | **Criteria** |  |  | Rovers | | Manual A manual database is one that is paper-based. Examples include Yellow Pages, Argos catalogue etc. Electronic An electronic database is stored on a computer. It has many advantages over a paper-based database including:   * Takes up less storage space * Quicker to search for and sort information * Easier to change/edit data that is stored * Easier to move data eg send via email or storage device instead of carrying a filing cabinet  Flat File A flat file database stores all data in one table. Field A field is one piece of information within a table. These can be identified by the different **headings** at the top of each column. There are different types of fields:   * Text – stores words * Number – stores digits * Boolean – stores yes/no values * Date – stores date formats e.g. DD/MM/YYYY * Time – stores time formats e.g. HH/MM/SS, 00.00.00 * Attachment – stores a picture or file  Record A record is one completed set of fields within a table. These can be identified by the number of **completed rows** in a table. Searching A **query** is used to find information within a database. It is made up of the **fields** you want to search and the **criteria** (the specific information you want to find). |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sorting Database tables can be sorted by fields in two different orders: ascending or descending.  **Ascending** – lowest to highest, A – Z, 0 – 9.  **Descending** – highest to lowest, Z – A, 9 – 0. Relational A relational database stores data in **two or more** linked tables. The have several advantages over using a flat file database:   * Data only needs to be entered once * Data is easier to edit – one change is automatically updated everywhere else it appears where as a flat file requires you to change it yourself * Saves time  Keys A **primary key** is used to ensure that every record is unique.  A **foreign key** is used to link two tables together. Example 1  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Club ID** | **Club Name** | **Location** | **Price £** | **Indoors?** | **Teacher** | | C011 | Table Tennis | Building | 10 | Yes | A. O’Neil | | C012 | Table Tennis | Hall | 10 | Yes | B. Weston | | C003 | Football | Arena | 15 | No | D. Carson | | C004 | Basketball | Hall | 12 | Yes | A. O’Neil |   In the table above, there are **6** fields and **4** records. | The field types would be as follows:   * Club ID – Text * Club Name – Text * Location – Text * Price £ – Number * Indoors? – Boolean * Teacher – Text   It is sorted by the field **Club Name** in a **descending** order (every other field changes order).  The primary key would be **Club ID** because it is a different value (unique) for every row. Example 2  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Player ID** | **First Name** | **Surname** | **Squad No.** | **Club ID** | **Club Name** | **Stadium** | | P101 | James | McGinn | 2 | C100 | Rovers | Town Park | | P201 | Graeme | Rooney | 7 | C100 | Rovers | Town Park | | P103 | Darren | May | 2 | C201 | United | ABC Arena | | P102 | Simon | Shinnie | 10 | C201 | United | ABC Arena | | P001 | Sean | Ball | 18 | C201 | United | ABC Arena | | P200 | Billy | Logan | 7 | C102 | Central | Rose Lane |   In the table above there are **7** fields and **6** records. |