



Name:

Index:

Reg. No.:

Class:

Date:

Chapter 9 Databases

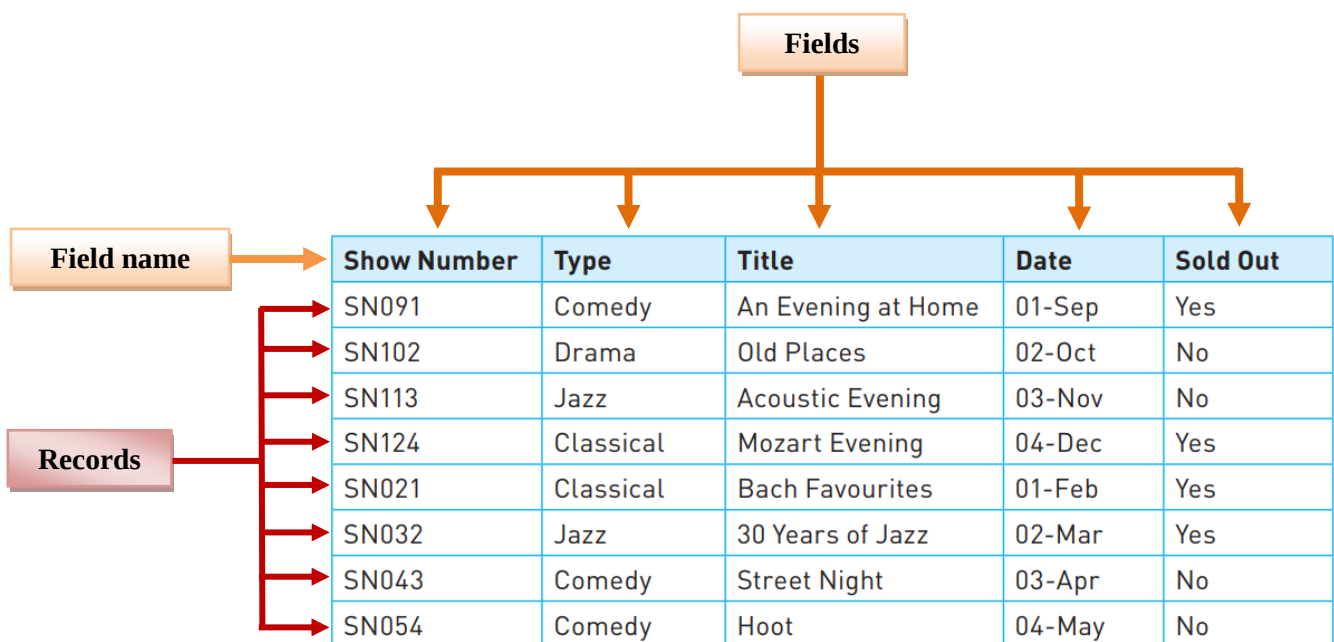
Single Table Databases

What is a database?

- A **Database** is an organised collection of data
- A database is made up of either **one or multiple tables** which are made up of **fields** and **records** to organise how it stores data
- A database is often stored on remote servers so **multiple users** can access it at the same time, useful for online systems

Fields & Records

- A field is represented in a database by a **column**
- A record is represented in a database by a **row**



Validation

- When a table is created, validation rules can be assigned to the different fields
- A validation rule controls what data can be entered into that field

Type	Description
Length Check	This type of validation checks the number of characters that have been entered into a field. For example, you might make phone numbers entered have to be eleven characters long
Format Check	This type of validation checks data entered meets an exact format. For example, a product code might have to be two letters followed by five numbers
Range Check	A range check will check the number entered is within a specific range. For example, the age of a dog would be between 0 - 40. Any other number would be disallowed
Presence Check	A presence check can be added to fields which cannot be left blank
Type Check	A type check will allow data with a specific data type to be entered into a field. For example, if text was entered into a field which is supposed to contain the date of birth of a person it would not be allowed
Check Digits	Check digit validation is a process used to verify the accuracy of numbers such as credit card numbers. A check digit is a single digit added to the end of the number, which is calculated based on a specific algorithm applied to the other digits in the number. When the data is re-entered the same algorithm can be applied, and if it produces a different result the code is incorrect

Data Types

- **Integer** - whole number. Eg: 1, 20, -50
- **Real** - decimal number. Eg: 2.05, -20.80
- **Text/alphanumeric** – Data that contains text. Eg: “Ahmed”, “ABC123”
- **Character** – single character. Eg: “a”, “F”, “@”
- **Date/Time** – date and time. Eg: ‘12/05/2024’, ‘12:13’
- **Boolean** - true or false values. Eg: True, False, Yes, No

SQL

SQL (Structured Query Language) **statements** are used to query data in the database and extract useful information.

- **SELECT**- Select the fields you want to display
- **FROM** - the table containing the data
- **WHERE** - the search criteria/condition
 - **SELECT** Name, Rating **FROM** Movie **WHERE** Rating > 8.4;
- **Operators**

Logical Operators	
Operators	Description
=	Equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<>	Not equal to

Boolean Operators	
Operators	Description
AND	specify multiple conditions that must all be true
OR	specify multiple conditions where one or more conditions must be true

- **SELECT** Name, Country **FROM** Movie **WHERE** Country <> “USA” **AND** Genre = “Family”;
- **SELECT** Name, Country **FROM** Movie **WHERE** Country = “USA” **OR** Country = “UK”;
- **ORDER BY**
 - Order the result according to ascending or descending. Keyword for ascending is **ASC** and keyword for descending is **DESC**.
 - **SELECT** Name, Genre **FROM** Movie **ORDER BY** Name **ASC**;
 - **SELECT** Name, Genre **FROM** Movie **ORDER BY** Name **DESC**;
- **SUM** (add numerical data) and **COUNT** (number of items)
 - **SELECT SUM**(Quantity) **FROM** Table;
 - **SELECT COUNT**(*) **FROM** Table **WHERE** Price > 2;

Worked Example

Customers

ID	Name	Age	City	Country
1	John Doe	30	New York	USA
2	Jane Doe	25	London	UK
3	Peter Lee	40	Paris	France

Write a SQL query to display the ID, name & age of customers who are older than 25.

Answer

```
SELECT ID, Name, Age
FROM Customers
WHERE Age > 25;
```

Output

ID	Name	Age
1	John Doe	30
3	Peter Lee	40