the Master Course

{CODENATION}

Backend Development

SQL & Aggregate Functions

{CODENATION}

{CODENATION}

Learning Objectives

To apply constraints to table columns.

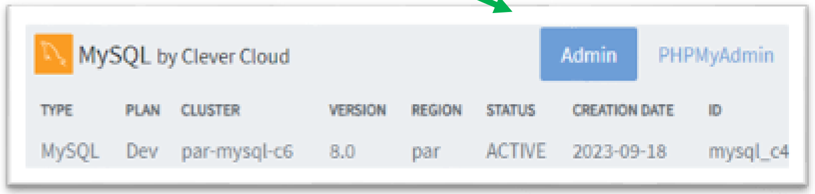
To use a range of operators, keywords and clauses to create declarative queries

To demonstrate and apply basic aggregate SQL functions.

Let's reset...

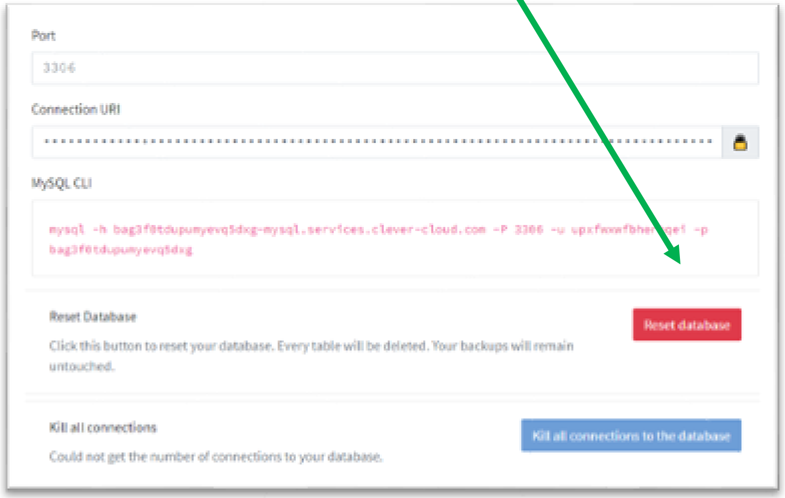
We will reset our Clever-Cloud database...

Click on Admin



TYPE	PLAN	CLUSTER	VERSION	REGION	STATUS	CREATION DATE	ID
MySQL	Dev	par-mysql-c6	8.0	par	ACTIVE	2023-09-18	mysql_c4

Scroll further down the page to see the reset database button.



Port

3306

Connection URI

```
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

MySQL CLI

```
mysql -h bag3f0tdupumyevq5dxg-mysql.services.clever-cloud.com -P 3306 -u upxfwxwfbher [illegible] qe1 -p
bag3f0tdupumyevq5dxg
```

Reset Database

Click this button to reset your database. Every table will be deleted. Your backups will remain untouched.

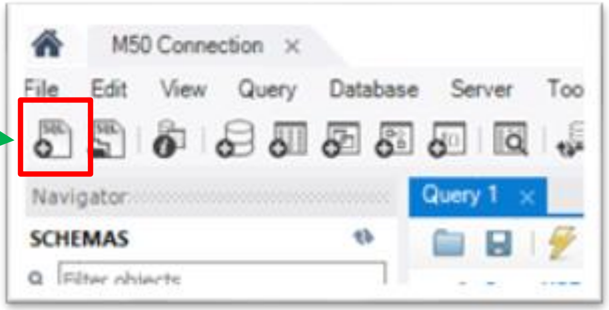
Kill all connections

Could not get the number of connections to your database.

We are now ready to create a new table...

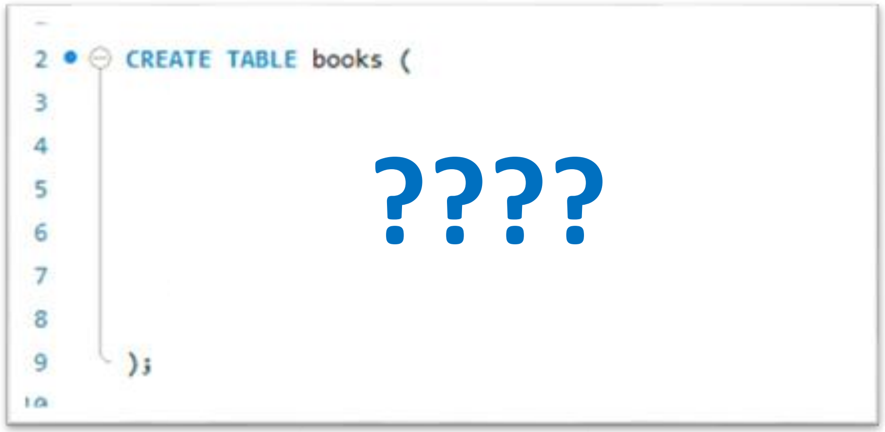
A new Book table...

Add a New Query window for our new table.



This help to keep us organised!

What might a Book table contain?



```
CREATE TABLE books (

);
```

????

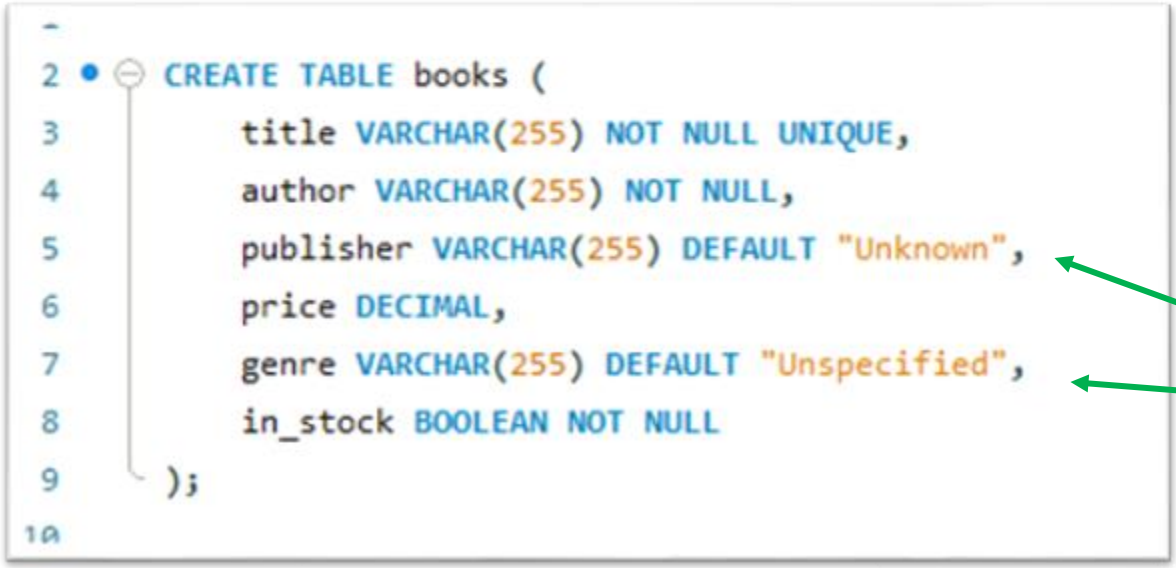
Column names?

Data types?

Default values?

For replication or unique?

A new Book table...



```
CREATE TABLE books (
    title VARCHAR(255) NOT NULL UNIQUE,
    author VARCHAR(255) NOT NULL,
    publisher VARCHAR(255) DEFAULT "Unknown",
    price DECIMAL,
    genre VARCHAR(255) DEFAULT "Unspecified",
    in_stock BOOLEAN NOT NULL
);
```

Why do these have a 'default' value?

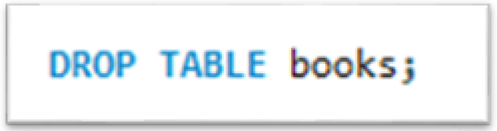
snake_case is commonly used by developers for SQL field names.

booleans can be represented as 0 or 1

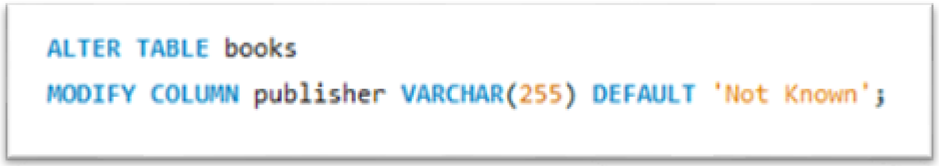
Run the query to see your new table with its columns.

Changed your mind...?

The opposite of creating a table is to DROP a table. This will delete the whole table from the database so handle with care!


```sql
DROP TABLE books;
```

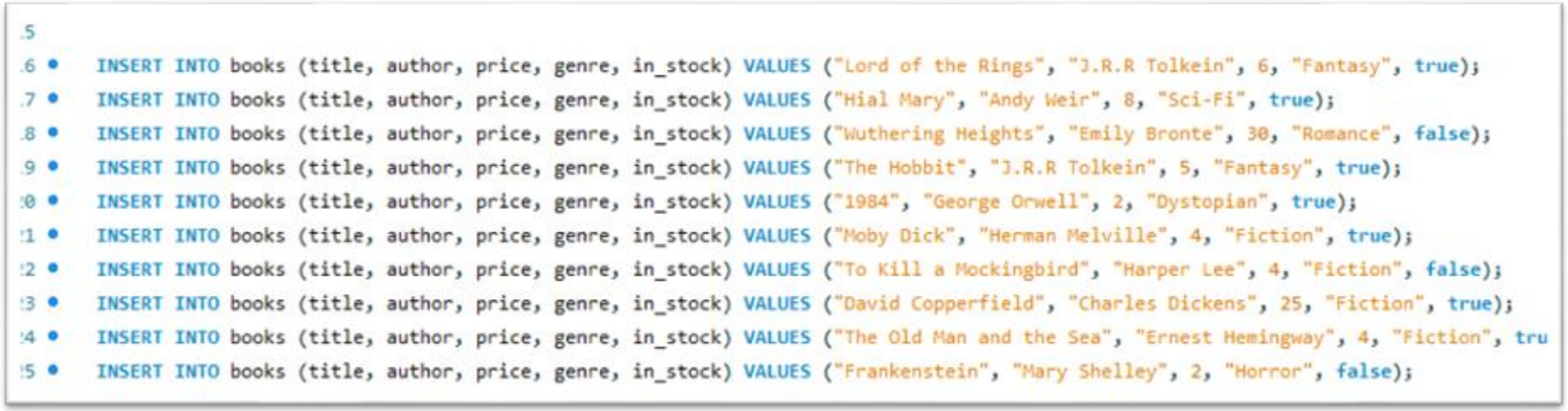
To change one of the fields in the table then we can ALTER the table and MODIFY the column details.


```sql
ALTER TABLE books
MODIFY COLUMN publisher VARCHAR(255) DEFAULT 'Not Known';
```

Further Reading: https://www.w3schools.com/sql/sql_alter.asp

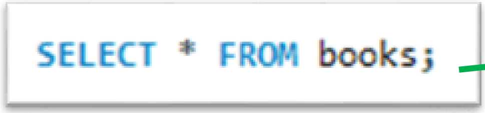
More data ...

We need some data to work with and so we can use INSERT as before:

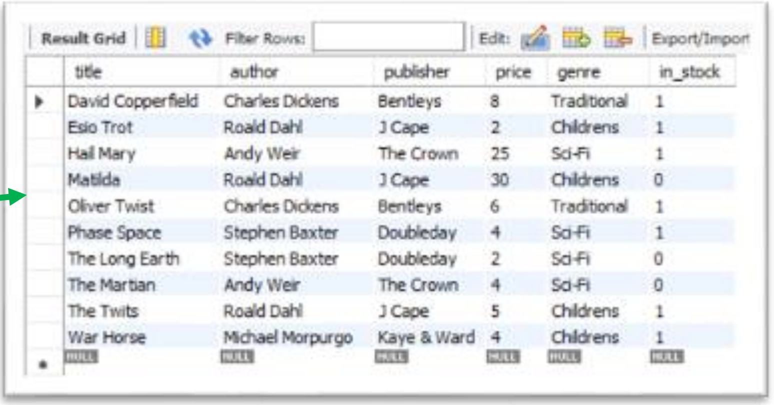


```
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("Lord of the Rings", "J.R.R Tolkein", 6, "Fantasy", true);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("Hial Mary", "Andy Weir", 8, "Sci-Fi", true);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("Wuthering Heights", "Emily Bronte", 30, "Romance", false);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("The Hobbit", "J.R.R Tolkein", 5, "Fantasy", true);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("1984", "George Orwell", 2, "Dystopian", true);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("Moby Dick", "Herman Melville", 4, "Fiction", true);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("To Kill a Mockingbird", "Harper Lee", 4, "Fiction", false);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("David Copperfield", "Charles Dickens", 25, "Fiction", true);
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("The Old Man and the Sea", "Ernest Hemingway", 4, "Fiction", tru
INSERT INTO books (title, author, price, genre, in_stock) VALUES ("Frankenstein", "Mary Shelley", 2, "Horror", false);
```

Check that the data is in the table ...



```
SELECT * FROM books;
```



title	author	publisher	price	genre	in_stock
David Copperfield	Charles Dickens	Bentleys	8	Traditional	1
Esio Trot	Roald Dahl	J Cape	2	Childrens	1
Hail Mary	Andy Weir	The Crown	25	Sci-Fi	1
Matilda	Roald Dahl	J Cape	30	Childrens	0
Oliver Twist	Charles Dickens	Bentleys	6	Traditional	1
Phase Space	Stephen Baxter	Doubleday	4	Sci-Fi	1
The Long Earth	Stephen Baxter	Doubleday	2	Sci-Fi	0
The Martian	Andy Weir	The Crown	4	Sci-Fi	0
The Twits	Roald Dahl	J Cape	5	Childrens	1
War Horse	Michael Morpurgo	Kaye & Ward	4	Childrens	1
NULL	NULL	NULL	NULL	NULL	NULL

What would happen if we tried to insert the same data into the table?

Activity

Create a new table called 'Authors' to hold the relevant information for book authors. Use appropriate data types, and other conditions, etc...

Populate this new table with the information for 10 unique authors – include the ones in your book table.

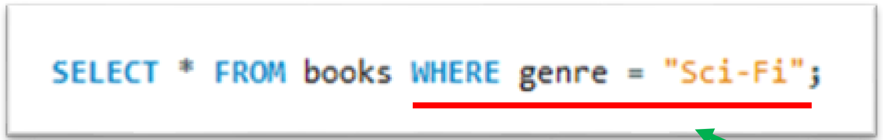
Stretch

Research the use of Primary and Foreign keys used in tables.

Filtering our results ...

Now, when it comes to READING our data, we might want to filter our returned results in some way.

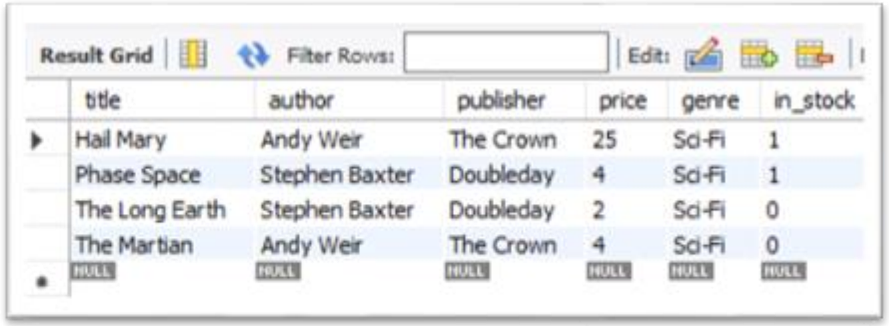
For instance, in our table we might want to see the books which are in the fiction genre.



```
SELECT * FROM books WHERE genre = "Sci-Fi";
```

We can add a WHERE clause to it to select only the books with that genre.

Run the query to see the filtered results.



title	author	publisher	price	genre	in_stock
Hail Mary	Andy Weir	The Crown	25	Sci-Fi	1
Phase Space	Stephen Baxter	Doubleday	4	Sci-Fi	1
The Long Earth	Stephen Baxter	Doubleday	2	Sci-Fi	0
The Martian	Andy Weir	The Crown	4	Sci-Fi	0
NULL	NULL	NULL	NULL	NULL	NULL

Activity

10 mins

A. Write a SELECT statement to return the books that are in stock.

B. Write a SELECT statement to return the books that are over £3 in price.

C. Write a SELECT statement to return the books that are between £4 and £10, inclusive.

D. Write a SELECT statement to return the fantasy books that are in stock.

Stretch

Create a similar question for someone else in the group to answer.

Primary Key

The SQL Primary Key is a column that uniquely identifies each record in a table. The Primary Key speeds up data access and is used to establish a relationship between tables.

- ✓ It contains a unique value.
- ✓ It cannot be null.
- ✓ A table can only have one Primary Key.

Further Reading: https://www.w3schools.com/sql/sql_primarykey.asp

Primary Key

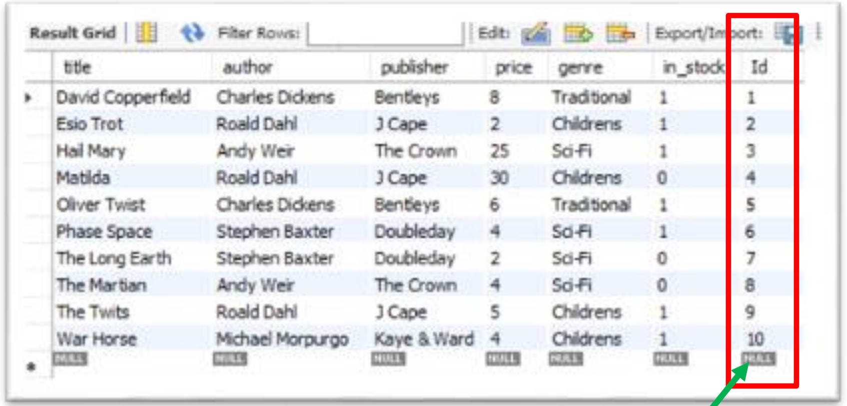
Let's add a primary key to our books table...

```
ALTER TABLE books ADD Id INT UNIQUE PRIMARY KEY AUTO_INCREMENT;
```

Adds 1 each time a new row is added.

Can never have the same id for different rows.

```
SELECT * FROM books;
```



title	author	publisher	price	genre	in_stock	Id
David Copperfield	Charles Dickens	Bentleys	8	Traditional	1	1
Esio Trot	Roald Dahl	J Cape	2	Childrens	1	2
Hail Mary	Andy Weir	The Crown	25	Sci-Fi	1	3
Matilda	Roald Dahl	J Cape	30	Childrens	0	4
Oliver Twist	Charles Dickens	Bentleys	6	Traditional	1	5
Phase Space	Stephen Baxter	Doubleday	4	Sci-Fi	1	6
The Long Earth	Stephen Baxter	Doubleday	2	Sci-Fi	0	7
The Martian	Andy Weir	The Crown	4	Sci-Fi	0	8
The Twits	Roald Dahl	J Cape	5	Childrens	1	9
War Horse	Michael Morpurgo	Kaye & Ward	4	Childrens	1	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL

We should now see that each row has been given a unique id.

CODE ALONG

Activity

5 mins

Add a Primary Key to your authors table called 'id'.

Check that it has added the values for each row.
INSERT a new row in the author table with another author's details.

What happened to the Id?

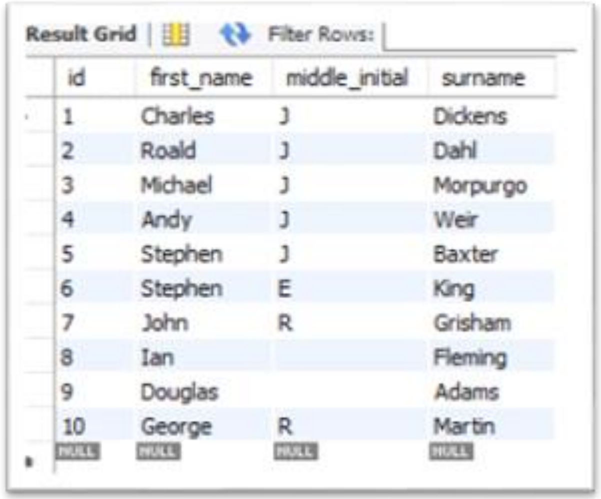
Stretch

Research the SQL command for DELETING data rows from a table.

Possible SQL for authors table ...

```
CREATE TABLE authors (
        id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(255) DEFAULT 'Not specified',
    middle_initial VARCHAR(15) DEFAULT 'Not Specified',
    surname VARCHAR(255) NOT NULL
);
```

```
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Charles', 'J', 'Dickens');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Roald', 'J', 'Dahl');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Michael', 'J', 'Morpurgo');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Andy', 'J', 'Weir');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Stephen', 'J', 'Baxter');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Stephen', 'E', 'King');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('John', 'R', 'Grisham');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Ian', '', 'Fleming');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('Douglas', '', 'Adams');
INSERT INTO authors (first_name, middle_initial, surname) VALUES ('George', 'R', 'Martin');
```



id	first_name	middle_initial	surname
1	Charles	J	Dickens
2	Roald	J	Dahl
3	Michael	J	Morpurgo
4	Andy	J	Weir
5	Stephen	J	Baxter
6	Stephen	E	King
7	John	R	Grisham
8	Ian		Fleming
9	Douglas		Adams
10	George	R	Martin
NULL	NULL	NULL	NULL

The Power of Relationships?

One of the great uses of the relational aspect of tables is that we can use details from one table with another table...

How can we link our two tables together?

{CODENATION}

Normalising a Database

Normalization is the process of organizing data in a database.

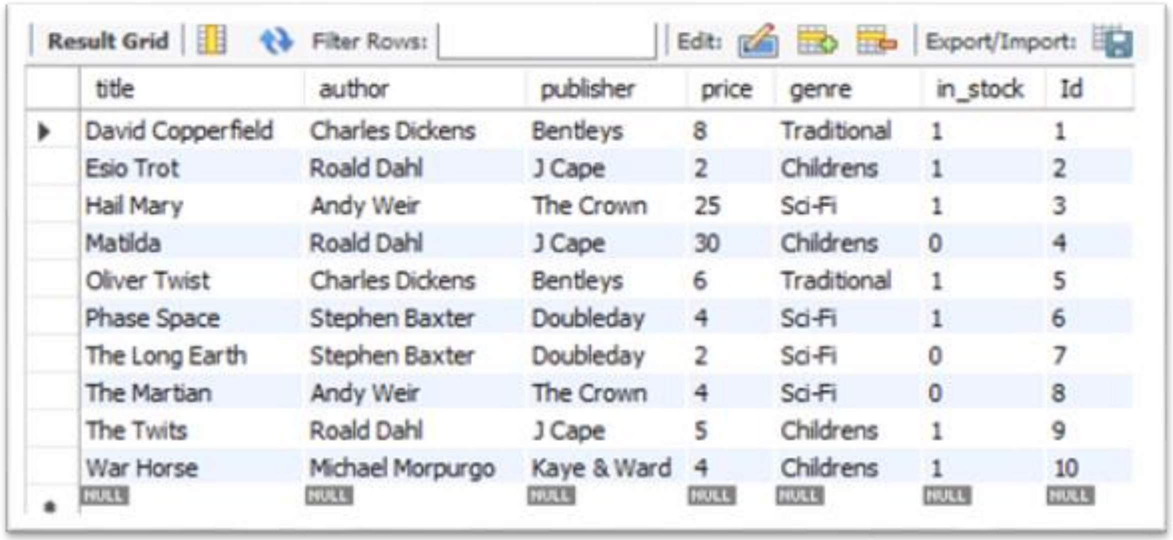
It includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

Normalising a Database

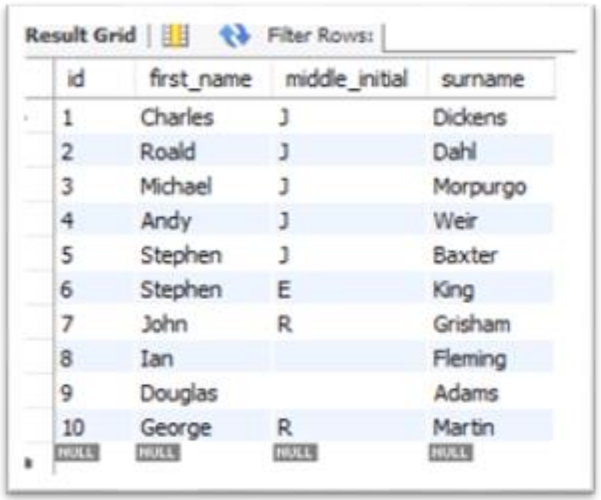
Three guidelines for normalising a database:

- ❑ Every table should have a Primary Key
- ❑ Eliminate redundant data (no repeated data)
- ❑ Column names should be unique

How can we normalise this data?

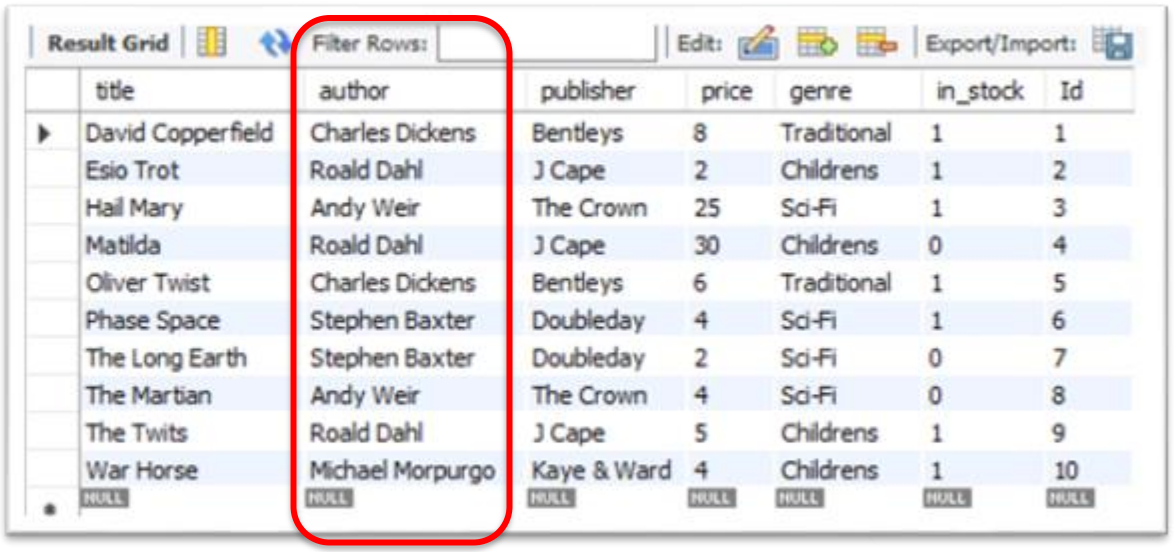


	title	author	publisher	price	genre	in_stock	Id
▶	David Copperfield	Charles Dickens	Bentleys	8	Traditional	1	1
	Esio Trot	Roald Dahl	J Cape	2	Childrens	1	2
	Hail Mary	Andy Weir	The Crown	25	Sci-Fi	1	3
	Matilda	Roald Dahl	J Cape	30	Childrens	0	4
	Oliver Twist	Charles Dickens	Bentleys	6	Traditional	1	5
	Phase Space	Stephen Baxter	Doubleday	4	Sci-Fi	1	6
	The Long Earth	Stephen Baxter	Doubleday	2	Sci-Fi	0	7
	The Martian	Andy Weir	The Crown	4	Sci-Fi	0	8
	The Twits	Roald Dahl	J Cape	5	Childrens	1	9
	War Horse	Michael Morpurgo	Kaye & Ward	4	Childrens	1	10
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

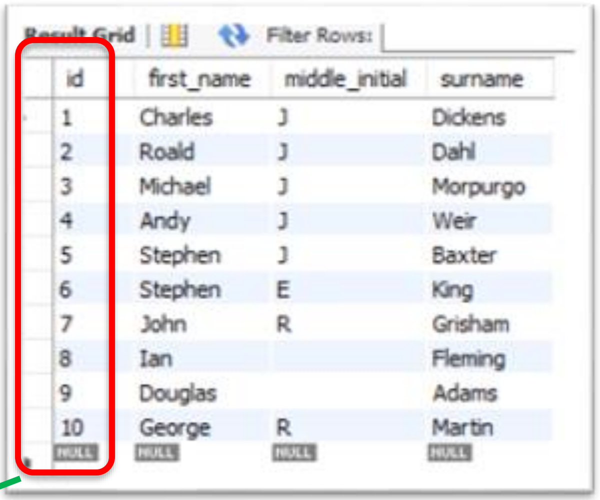


	id	first_name	middle_initial	surname
	1	Charles	J	Dickens
	2	Roald	J	Dahl
	3	Michael	J	Morpurgo
	4	Andy	J	Weir
	5	Stephen	J	Baxter
	6	Stephen	E	King
	7	John	R	Grisham
	8	Ian		Fleming
	9	Douglas		Adams
	10	George	R	Martin
*	NULL	NULL	NULL	NULL

How can we normalise this data?



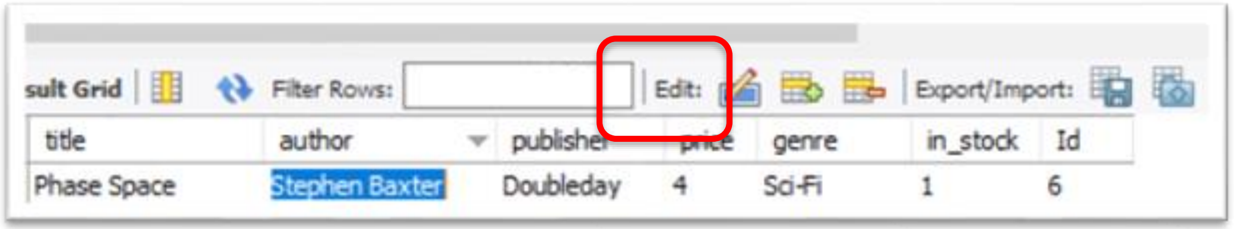
title	author	publisher	price	genre	in_stock	Id
David Copperfield	Charles Dickens	Bentleys	8	Traditional	1	1
Esio Trot	Roald Dahl	J Cape	2	Childrens	1	2
Hail Mary	Andy Weir	The Crown	25	Sci-Fi	1	3
Matilda	Roald Dahl	J Cape	30	Childrens	0	4
Oliver Twist	Charles Dickens	Bentleys	6	Traditional	1	5
Phase Space	Stephen Baxter	Doubleday	4	Sci-Fi	1	6
The Long Earth	Stephen Baxter	Doubleday	2	Sci-Fi	0	7
The Martian	Andy Weir	The Crown	4	Sci-Fi	0	8
The Twits	Roald Dahl	J Cape	5	Childrens	1	9
War Horse	Michael Morpurgo	Kaye & Ward	4	Childrens	1	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL



id	first_name	middle_initial	surname
1	Charles	J	Dickens
2	Roald	J	Dahl
3	Michael	J	Morpurgo
4	Andy	J	Weir
5	Stephen	J	Baxter
6	Stephen	E	King
7	John	R	Grisham
8	Ian		Fleming
9	Douglas		Adams
10	George	R	Martin
NULL	NULL	NULL	NULL

Repeated data can be simplified

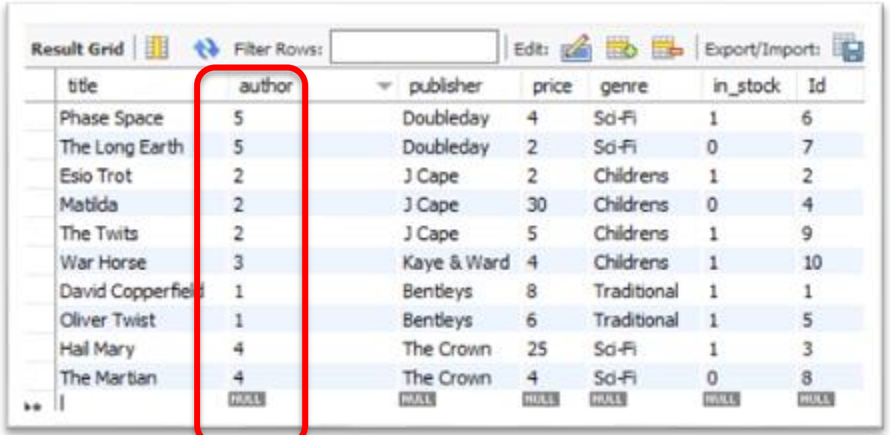
Let's normalise this data?



sult Grid | Filter Rows: | Edit: | Export/Import:

title	author	publisher	price	genre	in_stock	Id
Phase Space	Stephen Baxter	Doubleday	4	Sci-Fi	1	6

Edit the data in the books table to show an authorId rather than their name

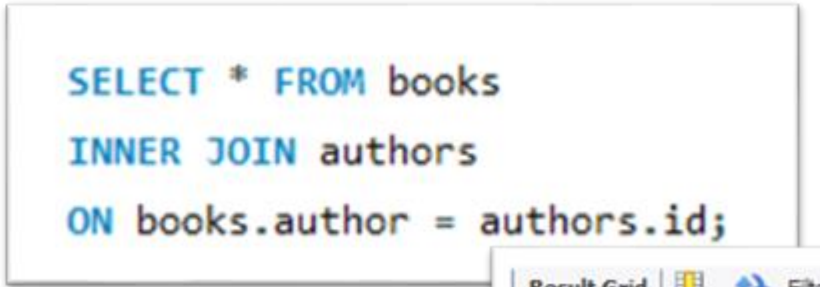


Result Grid | Filter Rows: | Edit: | Export/Import:

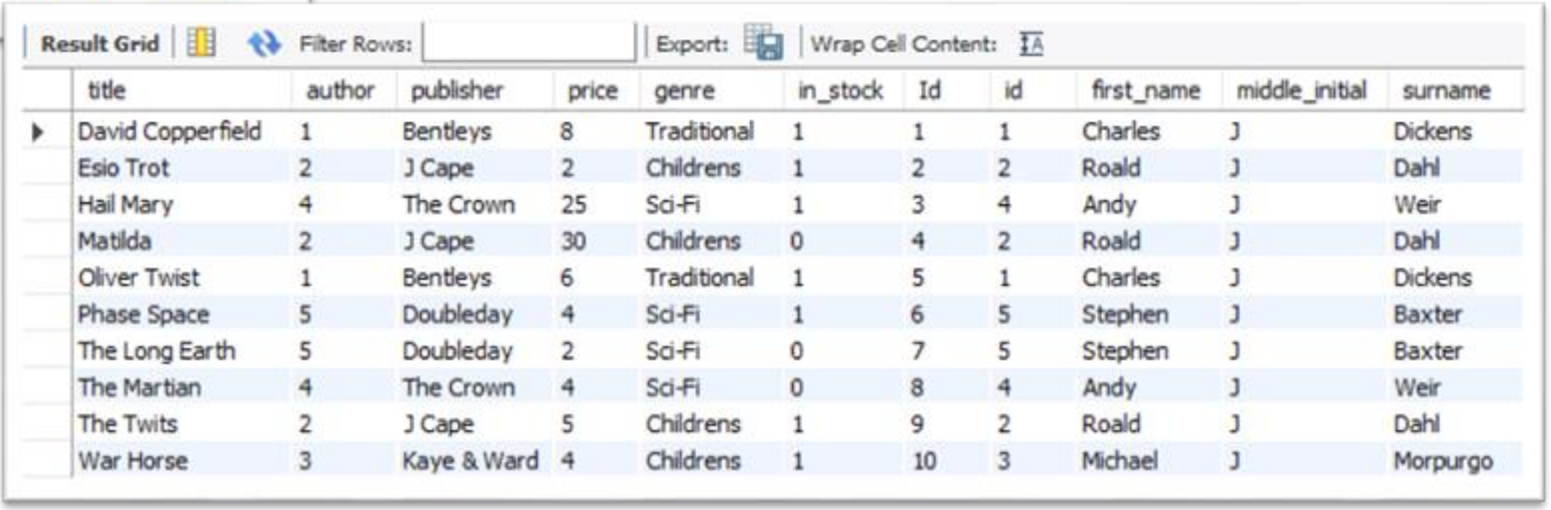
title	author	publisher	price	genre	in_stock	Id
Phase Space	5	Doubleday	4	Sci-Fi	1	6
The Long Earth	5	Doubleday	2	Sci-Fi	0	7
Esio Trot	2	J Cape	2	Childrens	1	2
Matilda	2	J Cape	30	Childrens	0	4
The Twits	2	J Cape	5	Childrens	1	9
War Horse	3	Kaye & Ward	4	Childrens	1	10
David Copperfield	1	Bentleys	8	Traditional	1	1
Oliver Twist	1	Bentleys	6	Traditional	1	5
Hail Mary	4	The Crown	25	Sci-Fi	1	3
The Martian	4	The Crown	4	Sci-Fi	0	8
	NULL	NULL	NULL	NULL	NULL	NULL

This is much better as is what we call, '**Normalising**' a database.

We can now improve and refactor our SQL statements...

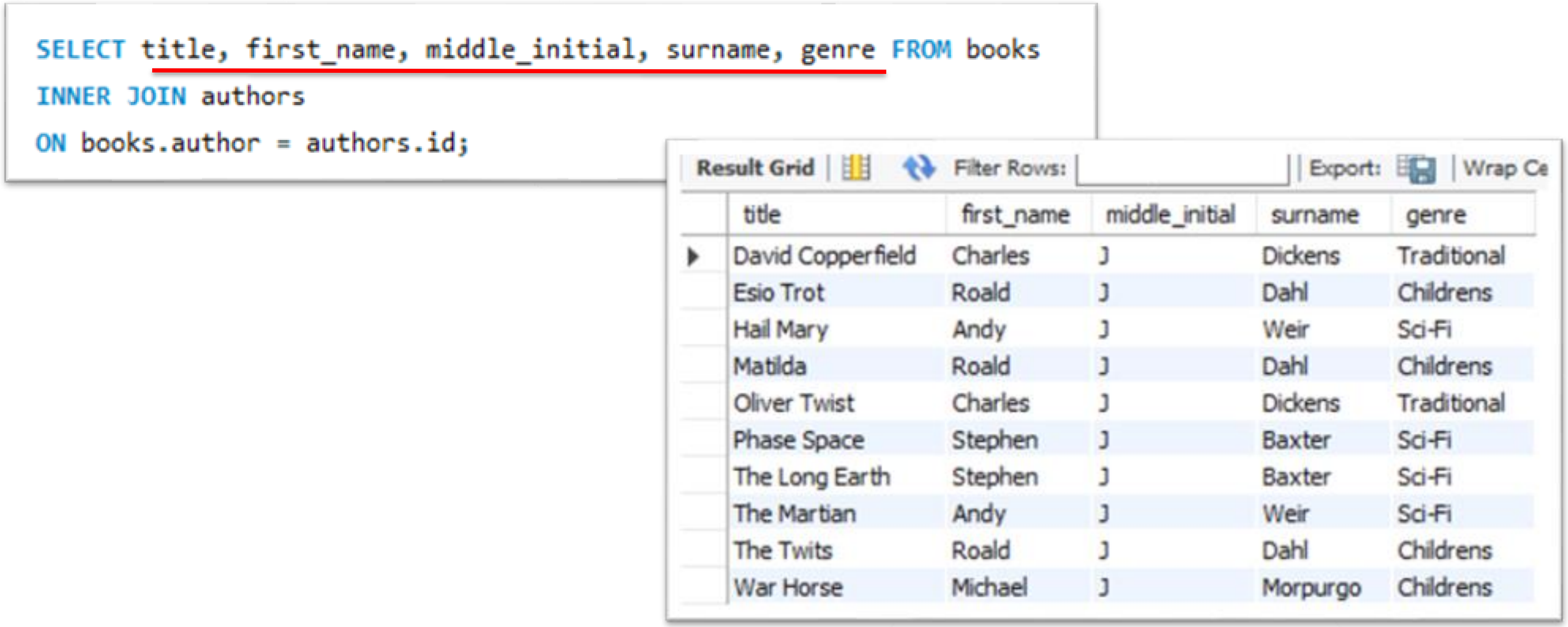


```
SELECT * FROM books
INNER JOIN authors
ON books.author = authors.id;
```



title	author	publisher	price	genre	in_stock	Id	id	first_name	middle_initial	surname
David Copperfield	1	Bentleys	8	Traditional	1	1	1	Charles	J	Dickens
Esio Trot	2	J Cape	2	Childrens	1	2	2	Roald	J	Dahl
Hail Mary	4	The Crown	25	Sci-Fi	1	3	4	Andy	J	Weir
Matilda	2	J Cape	30	Childrens	0	4	2	Roald	J	Dahl
Oliver Twist	1	Bentleys	6	Traditional	1	5	1	Charles	J	Dickens
Phase Space	5	Doubleday	4	Sci-Fi	1	6	5	Stephen	J	Baxter
The Long Earth	5	Doubleday	2	Sci-Fi	0	7	5	Stephen	J	Baxter
The Martian	4	The Crown	4	Sci-Fi	0	8	4	Andy	J	Weir
The Twits	2	J Cape	5	Childrens	1	9	2	Roald	J	Dahl
War Horse	3	Kaye & Ward	4	Childrens	1	10	3	Michael	J	Morpurgo

We can improve this further...



```
SELECT title, first_name, middle_initial, surname, genre FROM books
INNER JOIN authors
ON books.author = authors.id;
```

title	first_name	middle_initial	surname	genre
David Copperfield	Charles	J	Dickens	Traditional
Esio Trot	Roald	J	Dahl	Childrens
Hail Mary	Andy	J	Weir	Sci-Fi
Matilda	Roald	J	Dahl	Childrens
Oliver Twist	Charles	J	Dickens	Traditional
Phase Space	Stephen	J	Baxter	Sci-Fi
The Long Earth	Stephen	J	Baxter	Sci-Fi
The Martian	Andy	J	Weir	Sci-Fi
The Twits	Roald	J	Dahl	Childrens
War Horse	Michael	J	Morpurgo	Childrens

https://www.w3schools.com/sql/sql_join.asp for more information.

All, okay?

CRUD

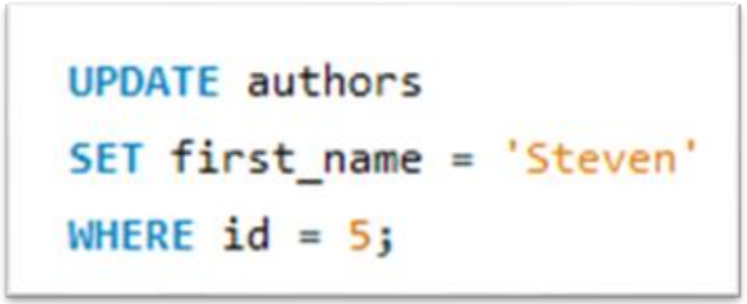
Create	✓
Read	✓
Update	☐
Delete	✓

We can create, read and delete data.

But what about **Updating** data?

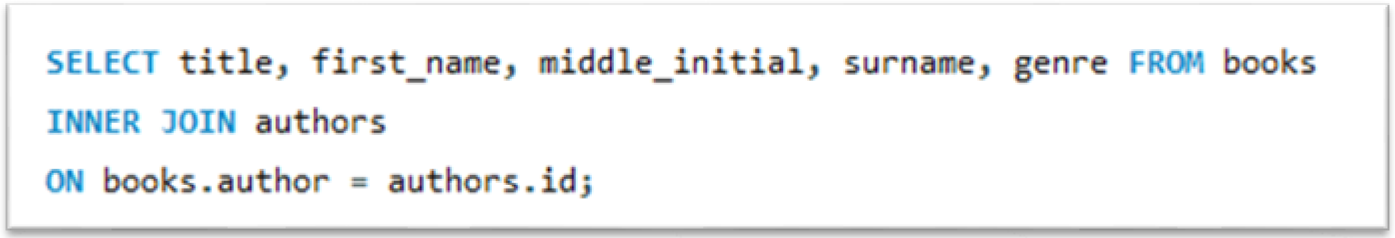
We can improve this further...

What if I wanted to correct a mistake on the authors table?
For example, ‘Charles’ to ‘Charlie...



```sql
UPDATE authors
SET first_name = 'Steven'
WHERE id = 5;
```

Then run this again...



```sql
SELECT title, first_name, middle_initial, surname, genre FROM books
INNER JOIN authors
ON books.author = authors.id;
```

Activity

5 mins

Add a SQL statement to update the surname of one of your authors. Check it with a suitable READ statement.

???? ADD ANOTHER ACTIVITY HERE

Stretch

????? PLEASE ADD

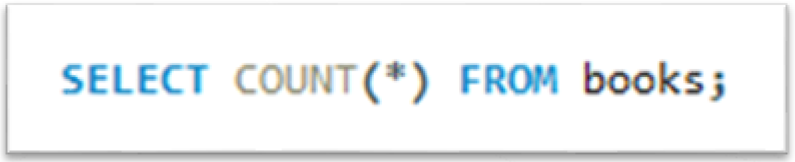
Aggregate Functions

They allow us to perform mathematical functions on our data, e.g., counting items, finding average prices of books, etc

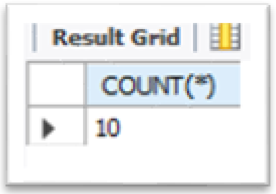
Function: Count

This function returns the number of items in the table.

All books:

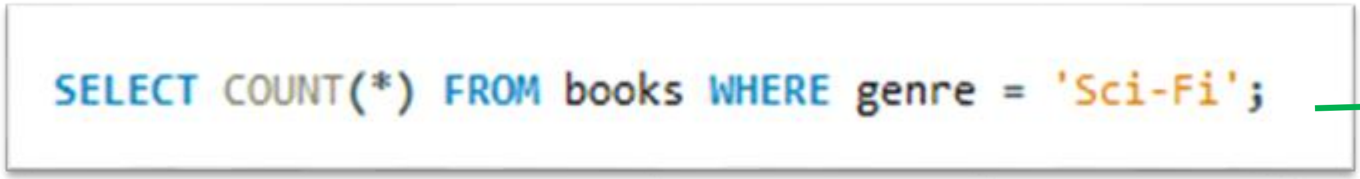


```sql
SELECT COUNT(*) FROM books;
```

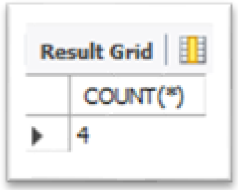


```
Result Grid
COUNT(*)
10
```

Specific books:

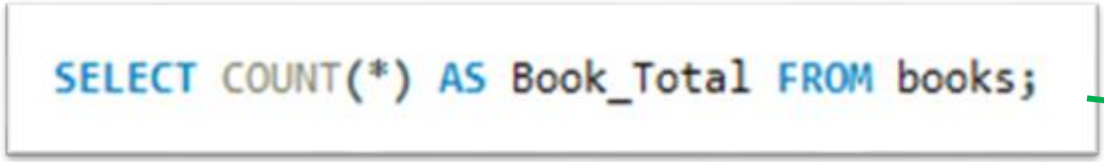


```sql
SELECT COUNT(*) FROM books WHERE genre = 'Sci-Fi';
```

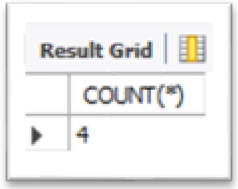


```
Result Grid
COUNT(*)
4
```

All books using 'AS':



```sql
SELECT COUNT(*) AS Book_Total FROM books;
```

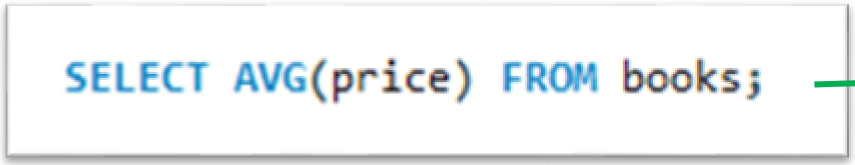


```
Result Grid
COUNT(*)
4
```

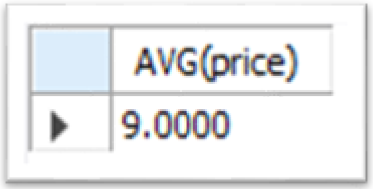
Function: AVG

This function returns the mathematical average of given values.

All books:

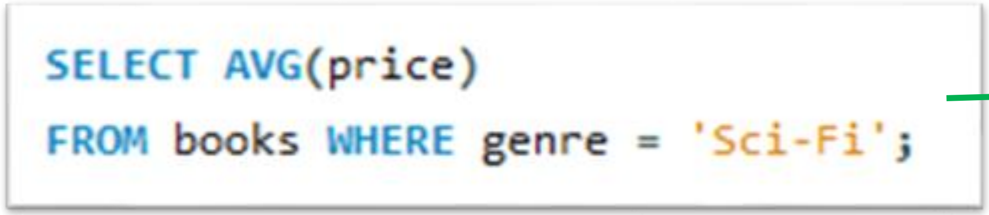


```
SELECT AVG(price) FROM books;
```

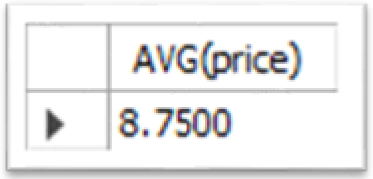


```
AVG(price)
9.0000
```

Specific books:

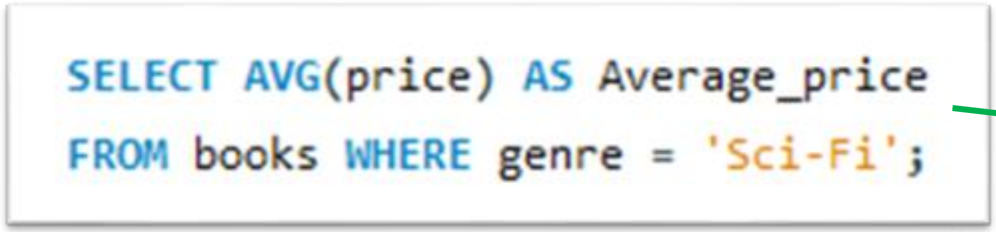


```
SELECT AVG(price)
FROM books WHERE genre = 'Sci-Fi';
```

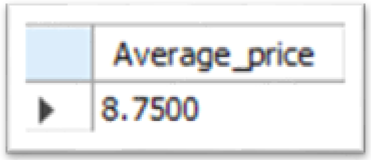


```
AVG(price)
8.7500
```

All books using 'AS':



```
SELECT AVG(price) AS Average_price
FROM books WHERE genre = 'Sci-Fi';
```

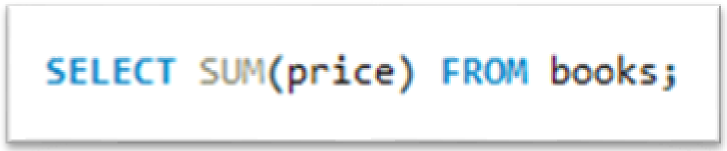


```
Average_price
8.7500
```

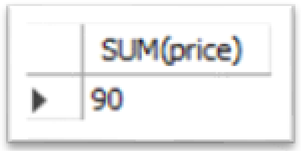
Function: SUM

This function returns the sum of values in the table.

All books:

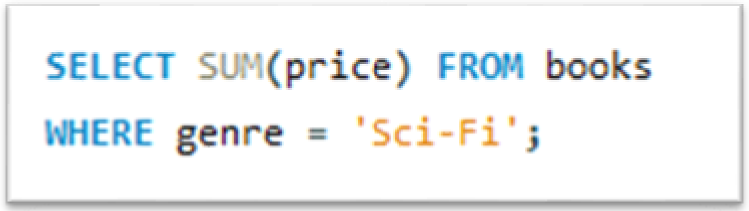


```sql
SELECT SUM(price) FROM books;
```

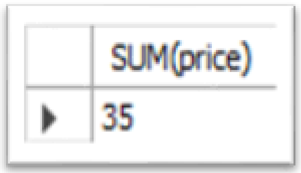


```
  SUM(price)
▶ 90
```

Specific books:



```sql
SELECT SUM(price) FROM books
WHERE genre = 'Sci-Fi';
```

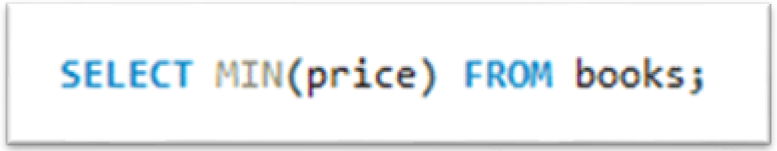


```
  SUM(price)
▶ 35
```

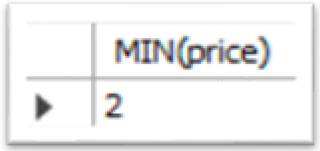
Function: Min and Max

These functions return the smallest or largest value of the data

MIN

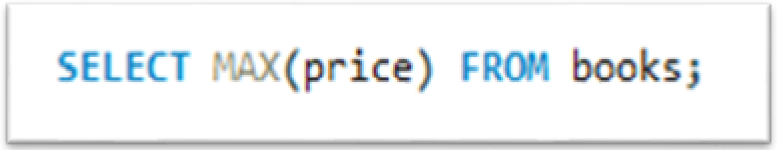


```
SELECT MIN(price) FROM books;
```

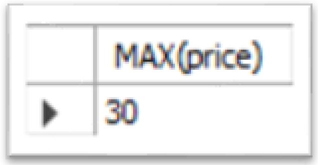


	MIN(price)
▶	2

MAX



```
SELECT MAX(price) FROM books;
```



	MAX(price)
▶	30

Activity

Practice your SQL statements with these questions:

1. Update the price of the book by Andy Weir to £11.
2. Find the average price of a Fiction book.
3. Find the total price of the books that are in stock.
4. How many books are out of stock?
5. What is the highest priced book in stock?
6. What is the lowest priced Sci-Fi book?

Stretch

A. List the authors whose first name begins with the letter S and D.

????? MORE QUESTIONS

Learning Objectives

To apply constraints to table columns.

To use a range of operators, keywords and clauses to create declarative queries

To demonstrate and apply basic aggregate SQL functions.