POWERING POTENTIAL

Data Manipulation

Module 4L: Databases

Databases

Module 4: Contents

- CRUD operations.
- Inserting, updating, and deleting.

Objectives

Data manipulation

Describe CRUD operations in MySQL:

- What does the syntax look like?

Insert data into a table:

- How can we start filling our table with data?

Update a record from a table:

- How can we easily edit records?

The current game database

games

PRODUCT ID	TITLE	QUANTITY	PRICE	AGE RATING
1	SHOOT THE COOL GUN 9	8965	£79.99	18
2	GUNBLADERS XXII	546	£64.99	15
3	PAINT DRYING SIMULATOR	435	£37.99	3
4	SITAR HERO	456	£45.99	12

customers

CUSTOMER ID	NAME	ADDRESS	EMAIL	PASSWORD
1	SIMON	256 BYTE STREET	SI@MAIL.CO.UK	**********
2	MARKUS	47 RED TIE ROAD	MARKUS47@POST.COM	**********
3	EMMA	63 NUMBER LANE	EM@LETTER.BOX	*******

orders

ORDER #	CUSTOMER ID	PLACED	TOTAL
1	1	2019-08-06	45.99
1	2	2019-08-14	37.99

orderline

ORDER ID	CUSTOMER ID	QTY_ORDERED
1	1	1
2	2	1

CRUD operations

Databases

Module 4: Data Manipulation

QA CRUD operations

You may see applications being referred to as CRUD applications:

- **C**reate.
- **R**ead.
- **U**pdate.
- **D**elete.

In MySQL syntax, there are a few ways that we can use CRUD functionality:

- Create: **CREATE**, **INSERT INTO**.
- Read: **SELECT**, **DESCRIBE**, **SHOW**.
- Update: **ALTER**, **UPDATE**.
- Delete: **DROP**, **DELETE**.

Data Manipulation Language (DML)

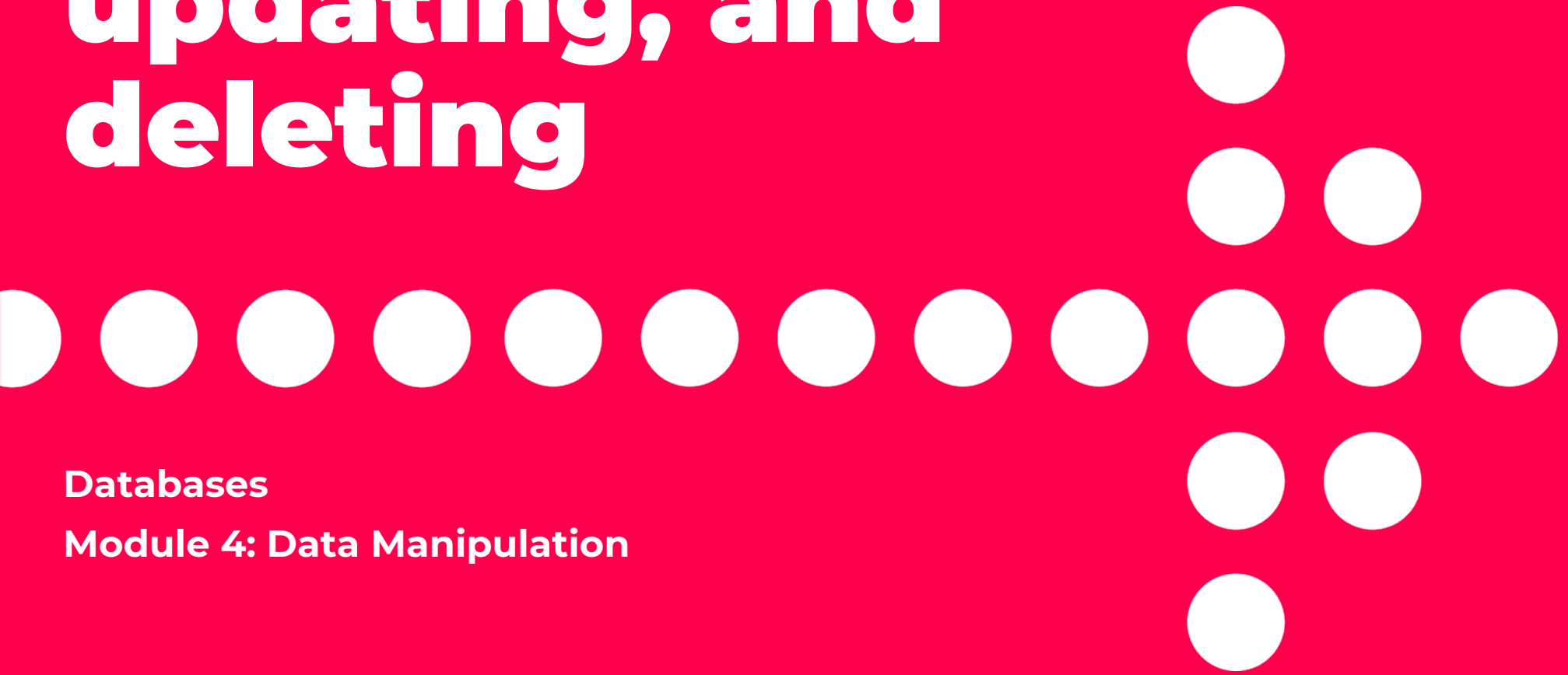
Data Manipulation Language (DML) is used to manipulate the *content* of the database:

- Arguably the most widely-used subtype of MySQL.

The most well-used CRUD operations we'd expect to use in DML are:

- Inserting data to, and deleting data from, tables.
- Reading data from tables based on various criteria.
- Updating the existing records in a table.

Inserting, updating, and deleting



Databases

Module 4: Data Manipulation

DML: Inserting into tables

The syntax for inserting records into a table breaks down into the following:

- Outline the table to insert into.
- Outline the fields that are being inserted into.
- Outline the values that are being inserted.

If we are inserting into all fields in a table, we can omit the column names from the statement:

```
INSERT INTO table_name (column_1, column_4, column_5)
VALUES (value_1, value_2, value_3);
```

```
INSERT INTO table_name
VALUES (value_1, value_2, value_3);
```

Example: Inserting a record

If we take the customer's table we're using for the GAME database:

customers

CUSTOMER ID	NAME	ADDRESS	EMAIL	PASSWORD
1	SIMON	256 BYTE STREET	SI@MAIL.CO.UK	**********
2	MARKUS	47 RED TIE ROAD	MARKUS47@POST.COM	***************
3	EMMA	63 NUMBER LANE	EM@LETTER.BOX	*******

We can insert into it by doing something like this:

```
INSERT INTO customers (name, address, email, password)
VALUES ('Jeremy', '132 Islington Row', 'jez@islington.co','1SL1n8t0n');
```

Why might we not have included the customer ID in our insert statement?

Creating a record for our GAME database

Outcome:

- Get used to adding records to a table.

Steps:

10 minutes, solo

- Using the table below as an example, insert a few records into our **customers'** table:

customers

CUSTOMER ID	NAME	ADDRESS	EMAIL	PASSWORD
1	SIMON	256 BYTE STREET	SI@MAIL.CO.UK	**********
2	MARKUS	47 RED TIE ROAD	MARKUS47@POST.COM	**********
3	EMMA	63 NUMBER LANE	EM@LETTER.BOX	*******

Creating records

customers

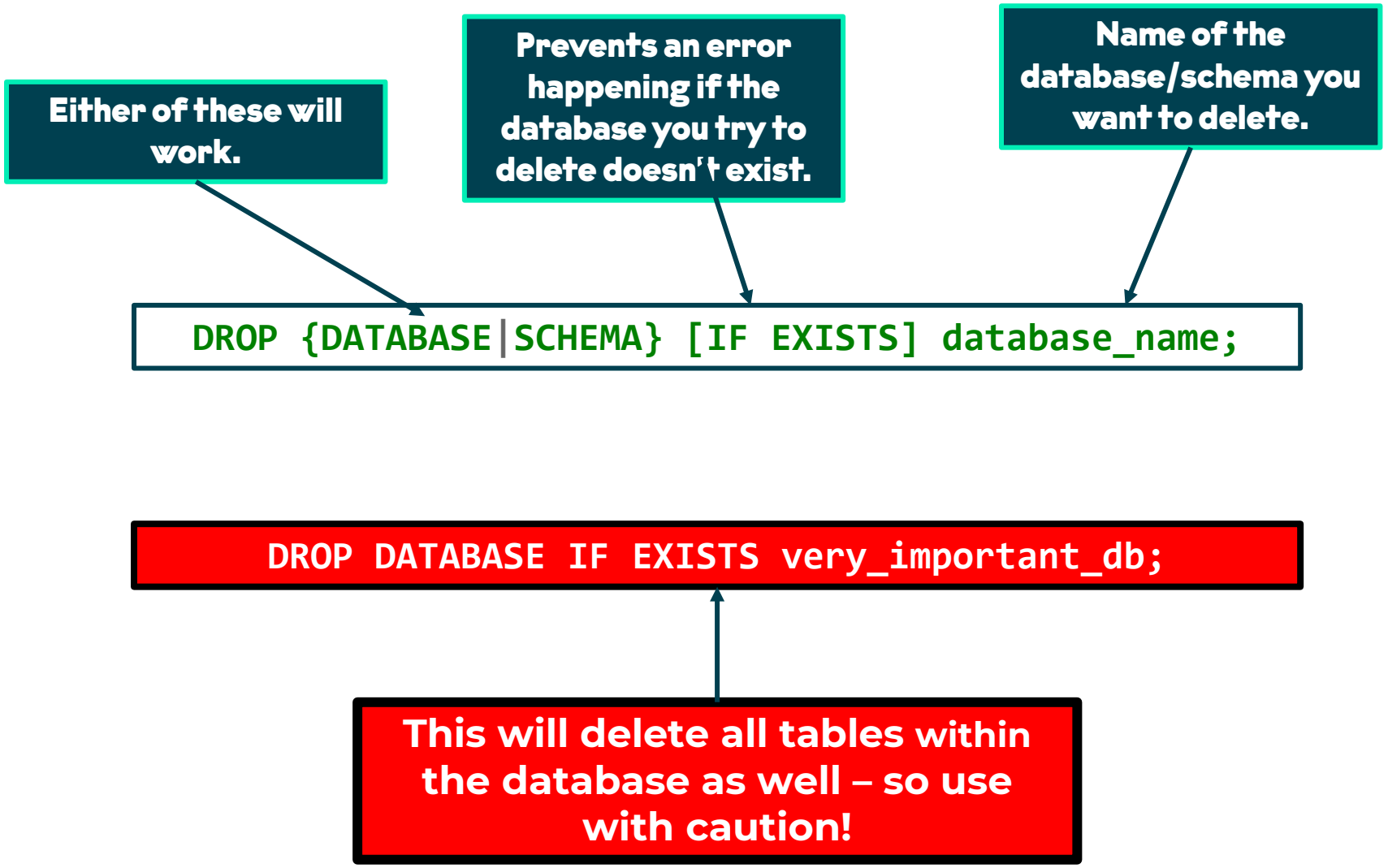
CUSTOMER ID	NAME	ADDRESS	EMAIL	PASSWORD
1	SIMON	256 BYTE STREET	SI@MAIL.CO.UK	**********
2	MARKUS	47 RED TIE ROAD	MARKUS47@POST.COM	**********
3	EMMA	63 NUMBER LANE	EM@LETTER.BOX	*******

```
INSERT INTO customer (customer_name, address, email, password)
VALUES ('Simon', '256 Byte Street', 'si@mail.co.uk', 'interestingpassword');

INSERT INTO customer (customer_name, address, email, password)
VALUES ('Markus', '47 Red Tie Road', 'markus47@post.com', 'boringpassword');

INSERT INTO customer (customer_name, address, email, password)
VALUES ('Emma', '63 Number Lane', 'em@letter.box', 'imafish');
```

Deleting a database



Deleting a table

Deleting a table is easy:

- You can list multiple tables if more than one needs to be dropped.
- However, table data and definitions are all removed, so use with caution.
- If not all the listed tables exist, MySQL will still drop all tables that do exist.

```
DROP TABLE table_that_shouldnt_be_dropped;
```

```
DROP TABLE [if exists] table_that_shouldnt_be_dropped;
```

Deleting a record

Deleting rows from a table uses the DELETE keyword

- To delete a row, specify exactly **WHERE** the row you want to delete is.
- If you don't do this, MySQL will simply delete everything with no mercy.
- These delete commands can become much more complex, by using some additional keywords that we can look at later.

```
DELETE FROM customers;
```

```
DELETE FROM customers WHERE name='Simon';
```

Updating a record

The syntax for updating records in a table breaks down into the following:

- Outline the table that the record exists in.
- Specify the value for the changed field.
- Outline any conditions.

```
UPDATE table_name
SET column1=value1, column2=value2
WHERE field=value;
```

Updating records in our GAME database

Outcome:

- Get used to editing tables in a database.

Steps:

10 minutes, solo

- Edit your customers table so that there are **postcode** and **age** fields associated with it.
- Edit your games table with a **release_date** field.
- Add any other bits you think the GAME database might need, but don't update the orders field yet.

Updating records

```
ALTER TABLE <tName> add age int;
ALTER TABLE <tName> add postcode varchar(8);

UPDATE customers SET age=25, postcode='NG1 1AA' WHERE name='Simon’;
UPDATE customers SET age=31, postcode='ETC ETC' WHERE name=‘Markus’;
UPDATE customers SET age=21, postcode='ETC ETC' WHERE name=‘Emma';
```

To view the content of the table, use the command:

```
mysql> SELECT * from customers;

+-------------+--------+-----------------+-------------------+----------------------+----------+-----+
|customer_id  |   name |         address |             email |             password | postcode | age |
+-------------+--------+-----------------+-------------------+----------------------+----------+-----|
|           1 |  Simon | 256 Byte Street |   simon@nomis.com | interestingpassword |  NG1 1AA |  25 |
|           2 | Markus | 47 Red Tie Road | markus47@post.com |      boringpassword |  ETC ETC |  31 |
|           3 |   Emma |  63 Number Lane |     em@letter.box |             iamafish |  ETC ETC |  21 |
+-------------+--------+-----------------+-------------------+----------------------+----------+-----+
3 rows in set (0.01 sec)
```

Updating records

```
ALTER TABLE games ADD release_date date;
UPDATE games SET release_date='2014-09-12' WHERE title='Sitar Hero';
```

To view the content of the table, use the command:

```
mysql> SELECT * from games;

+------------+--------------------------+----------+-------+------------+--------------+
| product_id |                    title | quantity | price | age_rating | release_date |
+------------+--------------------------+----------+-------+------------+--------------+
|          1 |     Shoot The Cool Gun 9 |     8965 | 79.99 |         18 |   2012-10-10 |
|          2 |          Gunbladers XXII |      546 | 64.99 |         15 |   2009-09-12 |
|          3 |   Paint Drying Simulator |      435 | 37.99 |          3 |   1994-05-05 |
|          4 |                Sitar Hero |      456 | 45.99 |         12 |   2014-09-12 |
+------------+--------------------------+----------+-------+------------+--------------+
4 rows in set (0.03 sec)
```

Summary

DATABASES: MODULE 4

Describe CRUD operations in MySQL:

- Create, Read, Update, Delete can be used with various DDL and DML commands in MySQL.

Insert data into a table:

- We inserted records into the customer and games tables of the GAME database.

Update a record from a table:

- We updated an existing record and altered the schema of our GAME database.

Thank you for listening

Any questions?