SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20031

Computing Technology Design Project

Week 07

Adding and Manipulating the Data

Database Development Lifecycle

1. Planning
2. Requirement gathering
3. Conceptual design
4. Logical design
5. **Physical design**
6. Construction
7. Implementation and rollout
8. Ongoing support

(A) Database Manipulation Language

Inserting Data Statement

The MySQL INSERT INTO Statement

The INSERT INTO statement is used to insert new records in a table.

INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

2. If you are adding values for all the columns of the table you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

Inserting Data Example

The following SQL statement inserts a new record in the "Customers" table:

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

The selection from the "Customers" table will now look like this:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway

Updating Data Statement

The MySQL UPDATE Statement

The UPDATE statement is used to modify the existing records in a table,

UPDATE Syntax

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Note: Be careful when updating records in a table! Notice the `WHERE` clause in the `UPDATE` statement. The `WHERE` clause specifies which record(s) that should be updated. If you omit the `WHERE` clause, all records in the table will be updated!

Updating Data Example

Demo Database:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y	Ana Trujillo	Avda. de la Constitución	México	05021	Mexico

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City = 'Frankfurt'
WHERE CustomerID = 1;
```

The selection from the "Customers" table will now look like this:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

Deleting Data Statement

The MySQL DELETE Statement

The DELETE statement is used to delete existing records in a table ,

DELETE Syntax

```
DELETE FROM table_name WHERE condition;
```

Note: Be careful when deleting records in a table! Notice the `WHERE` clause in the `DELETE` statement. The `WHERE` clause specifies which record(s) should be deleted. If you omit the `WHERE` clause, all records in the table will be deleted!

Deleting Data Example

Demo Database:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

The "Customers" table will now look like this:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

(B) The special concept of NULL

MySQL NULL Values

What is a NULL Value? → A field with a NULL value is a field with no value.

If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value

- **IS NULL Syntax** and **IS NOT NULL Syntax**

```
SELECT column_names
FROM table_name
WHERE column_name IS NULL;
```

```
SELECT column_names
FROM table_name
WHERE column_name IS NOT NULL;
```

Note: A NULL value is different from a zero value or a field that contains spaces. A field with a NULL value is one that has been left blank during record creation!

(C) Bulk Uploading Data

Bulk Uploading Data in MySQL

Bulk loading is the fastest way to insert large numbers of rows (Import Data) into a MySQL table. Using this facility instead of regular SQL insert statements, you can insert rows more rapidly.

- Create Database
- Add Tables
- Import Data

https://www.youtube.com/watch?v=OnXB3ZRrOW0

(D) Creating Dummy Data

Creating Dummy Data

Dummy data is *mock data generated at random* as a *substitute for live data* in *testing environments*. In other words, dummy data acts as a placeholder for live data, the latter of which testers only introduce once it's determined that the trail program does not have any unintended, negative impact on the underlying data

Take your pick from a number of online options

- https://www.mockaroo.com/
- https://analystanswers.com/dummy-data-definition-example-how-to-generate-it/

Tutorial & Workshop

1. See **Canvas -> Modules -> Week 7**