Control Flow Functions

- CASE..WHEN

- IF()

- IFNULL()

User-defined variables

- User defined variables are defined per session, i.e., they are specific to the SQL editor window that is open.

```
SET @name = 'John';
SET @age = 25;

SELECT @min_price:=MIN(price),
       @max_price:=MAX(price) FROM shop;
SELECT * FROM shop
WHERE price=@min_price OR price=@max_price;
```

Stored Programs (Stored Routines)

Stored programs allow SQL code to be saved in re-usable blocks in the database.

There are two types:

- Procedures (aka Stored Procedures)

- Functions (aka User-Defined Functions, UDFs)
  - Distinct from in-built MySQL functions like count(), max(), etc.)

Stored Programs (Stored Routines)

MySQL documentation:

https://dev.mysql.com/doc/refman/8.0/en/stored-programs-defining.html

https://dev.mysql.com/doc/refman/8.0/en/user-variables.html

Stored Programs

- Procedures: Use procedures when you need to modify data in the database.

- Functions: Use functions when you need to perform calculations on data and return a result without modifying the underlying data.

Stored Programs differences

- Procedures
  - Can have both input and output parameters.
  - Returns values via one or more output parameters.

- Functions
  - Can only have input parameters.
  - Can return only one value.

Changing the delimiter in code

Notice when writing stored programs that the default delimiter ; (semicolon) is changed to **//**

This is to allow the entire block of code that creates the program to be executed at once, otherwise each statement after the semicolon would be attempted, which will result in a compile time error.

After the program definition, the delimiter is changed back to it's default **;**

Changing the delimiter in code

```
delimiter //

create function calculate_years_of_independence(country_code char(3))
returns smallint
        reads sql data
begin
        declare v_years smallint;

        select year(now()) – IndepYear into v_years from country
    where Code = country_code;

    return v_years;
end //

delimiter ;
```