SQL

SELECT column-list FROM tablename
[join-expression]
[WHERE condition]
[ORDER BY column-list]
[GROUP BY column-name]
[HAVING condition];
Join-expression =
table1 [left / right] JOIN table2 ON
condition | USING (column-list)
Conditions : =,>,<,>=,<=,<>, BETWEEN .. AND.., IN (list), IS NULL, LIKE, EXISTS
Logical operators: AND, OR, NOT
Set operations: UNION, INTERSECT, EXCEPT

INSERT INTO tablename [{column-name,}] VALUES (data-value-list)
UPDATE tablename
[SET column-name= <data-value>] [WHERE condition]

PLPGSQL FUNCTION

CREATE [OR REPLACE] FUNCTION function-name (parameter-list) RETURNS <return-type> as $$
[DECLARE
[constant/variable declarations]]
BEGIN
Executable statements
RETURN Return value
[EXCEPTION
exception handlers]
END;$$

PLPGSQL TRIGGER

CREATE TRIGGER triggername [BEFORE / AFTER] operation ON tablename FOR EACH ROW EXECUTE FUNCTION function-name;

PLPGSQL PROCEDURE

CREATE [OR REPLACE] PROCEDURE procedure-name as $$
[DECLARE
[constant/variable declarations]]
BEGIN
Executable statements
[EXCEPTION
exception handlers]
END; $$
Parameters must have a name and a data type but may be optional (DEFAULT NULL).

MONGODB EXAMPLES

Create a products collection:

```
db.createCollection("contacts",
{ validator:{ $or:[
  {phone:{$type:"string"}},
  {email: {$regex: /@mytudublinproduct\.ie$/}},
  {status:{$in:["Unknown","Incomplete"]}}
 ] }}}
```

Insert an order into the productOrders collection ordering 3 items:

```
db.productOrders.insertOne({
OrderNo:1,
OrderDate: new ISODate("2022-04-21"),
items:
    [
    { item: "pencil", qty: 50, type: "no.2" },
    { item: "pen", qty: 20 },
    { item: "eraser", qty: 25 }
  ]})
```

Attributes may be embedded docs or arrays.