



## Ficha de trabalho Nº4

<b>Domínio:</b> Programação em SQL - Iniciação	<b>FEC.SQI.C.D.61</b>
<b>Formador:</b> Manuel Carlos Abreu Loureiro	<b>Data:</b> 2023/02/27

Caso ainda não tenha criado a sua base de dados sistema, importe o seguinte código SQL para a sua base sistema qualquer outra base de dados utilizada durante formação.

```
DROP TABLE IF EXISTS utilizador;  
CREATE TABLE utilizador (  
  idUtilizador INT(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  utilizador VARCHAR(30) NOT NULL,  
  dataReg TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
DROP TABLE IF EXISTS estado;  
CREATE TABLE estado(  
  idEstado INT(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  estado VARCHAR(20) NOT NULL,  
  dataReg TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
DROP TABLE IF EXISTS senha;  
CREATE TABLE senha (  
  idSenha INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  idUtilizador INT(11),  
  senha VARCHAR(50),  
  dataReg TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
DROP TABLE IF EXISTS token;  
CREATE TABLE token(  
  idToken INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  idUtilizador INT(11),  
  token VARCHAR(50),  
  dataReg TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
DROP TABLE IF EXISTS estado_utilizador;  
CREATE TABLE estado_utilizador(  
  idEstadoUtilizador INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  idEstado INT(11),  
  idUtilizador INT(11),  
  dataReg TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```



-- 1 Criar procedimento para registar estado

```
DROP PROCEDURE IF EXISTS p_registar_estado ;  
DELIMITER $$
```

```
CREATE PROCEDURE p_registar_estado  
(IN a VARCHAR(50))  
BEGIN  
INSERT INTO estado VALUES (NULL, a, CURRENT_TIMESTAMP);  
END
```

```
$$ DELIMITER ;
```

```
CALL p_registar_estado('inválido');  
CALL p_registar_estado('ativo');
```

-- 2 Criar procedimento para registar utilizador

```
DROP PROCEDURE IF EXISTS p_registar_utilizador;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE p_registar_utilizador  
(IN a VARCHAR(50))  
BEGIN  
INSERT INTO utilizador VALUES (NULL, a, CURRENT_TIMESTAMP);  
SET @idUtilizador = LAST_INSERT_ID();  
SET @token = substring(sha1(rand()),1,40);  
INSERT INTO token VALUES (NULL, @idUtilizador, @token, CURRENT_TIMESTAMP);  
INSERT INTO estado_utilizador VALUES (NULL, 1, @idUtilizador, CURRENT_TIMESTAMP);  
END
```

```
$$ DELIMITER ;
```

```
CALL p_registar_utilizador('manuel,@mail.pt');
```

-- 2 Criar procedimento para validar utilizador

```
DROP PROCEDURE IF EXISTS p_validar_utilizador;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE p_validar_utilizador  
(IN a INT(11), b VARCHAR(50), c VARCHAR(50))  
BEGIN  
  -- a > idUtilizador  
  -- b > senha  
  -- c > token  
  INSERT INTO senha VALUES (NULL, a, sha1(b), CURRENT_TIMESTAMP);  
  DELETE FROM token WHERE token = c LIMIT 1;  
  INSERT INTO estado_utilizador VALUES (NULL, 1, a, CURRENT_TIMESTAMP);  
END
```

```
$$ DELIMITER ;
```

```
CREATE TABLE eventos(  
  idEvento INT(11) AUTO_INCREMENT PRIMARY KEY,  
  evento VARCHAR(100),  
  tabela VARCHAR(100),  
  idTabela INT(11),  
  dataEvento TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```

## Exercícios

### 1. Vistas

- a. Crie uma vista que apresente todos os estados.
- b. Crie uma vista que apresente todos os utilizadores e todos os seu estados.  
(idUtilizador, utilizador, estado)
- c. Crie uma vista que apresente todos os utilizadores não validados.  
(idUtilizador, utilizador, estado)
- d. Crie uma vista que apresente todos os utilizadores ativos.  
(idUtilizador, utilizador, estado)
- e. Crie uma vista que apresente os utilizadores e todas as suas senhas.  
(idUtilizador, utilizador, senha)
- f. Crie uma vista que apresente os utilizadores e a sua senha atual.  
(idUtilizador, utilizador, estado)
- g. Crie uma vista que apresente os utilizadores e o seu estado atual.  
(idUtilizador, utilizador, estado)
- h. Crie uma vista que mostre todos os tokens e seus utilizadores.  
(idUtilizador, utilizador, token)



2. Procedimentos
  - a. p\_recuperar\_senha
  - b. p\_registar\_senha
  - c. Crie um procedimento que remova um token que tenha mais de 24 horas
  - d. Crie um procedimento que remova um utilizador da base de dados e de todas as tabelas a si associadas
3. Triggers
  - a. Crie um trigger para cada evento ocorrido em cada tabela de senha quando ocorrer uma nova inserção
  - b. Crie um trigger para cada evento ocorrido em cada tabela de estado\_utilizador quando ocorrer uma nova inserção
  - c. Crie um trigger para cada evento ocorrido em cada tabela de token para quando ocorrer uma nova inserção ou um registo for removido.
4. Que alterações faria na base de dados se fosse permitido mais do que um método de autenticação, como por exemplo o telemóvel ou outro email para o mesmo utilizador? Apresente a sua solução em SQL



```
DROP TRIGGER IF EXISTS insert_utilizador;  
CREATE TRIGGER insert_utilizador AFTER INSERT ON `utilizador` FOR EACH ROW INSERT  
INTO eventos VALUES (NULL, 'registo', 'utilizador', NEW.idUtilizador, CURRENT_TIMESTAMP);
```

```
DROP TRIGGER IF EXISTS after_delete_utilizador;  
CREATE TRIGGER after_delete_utilizador BEFORE DELETE ON utilizador FOR EACH ROW  
INSERT INTO eventos VALUES (NULL, 'Apagado', 'utilizador', OLD.idUtilizador,  
CURRENT_TIMESTAMP);
```

```
DROP TRIGGER IF EXISTS after_update_utilizador;  
CREATE TRIGGER after_update_utilizador AFTER UPDATE ON utilizador FOR EACH ROW  
INSERT INTO eventos VALUES (NULL, 'Atualizado', 'utilizador', OLD.idUtilizador,  
CURRENT_TIMESTAMP);
```



```
CREATE TABLE `barco` (  
  `idBarco` int(11) NOT NULL,  
  `nome` varchar(100) NOT NULL,  
  `pesoMaximo` int(11) NOT NULL,  
  `tripulacao` int(11) NOT NULL,  
  `comprimento` decimal(3,1) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
INSERT INTO `barco` (`idBarco`, `nome`, `pesoMaximo`, `tripulacao`, `comprimento`) VALUES  
(1, 'Aurélio', 100000, 2, '10.0'),  
(2, 'Simone', 20000, 3, '6.0'),  
(3, 'Batista', 50000, 3, '8.0'),  
(4, 'Fortunato', 40000, 2, '10.0'),  
(5, 'Evaristo', 45000, 2, '6.0'),  
(6, 'Afonso', 30000, 3, '8.0');
```

```
CREATE TABLE `peixe` (  
  `idPeixe` int(11) NOT NULL,  
  `nome` varchar(100) NOT NULL,  
  `pesoMedio` decimal(4,1) NOT NULL,  
  `preco` decimal(4,1) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
INSERT INTO `peixe` (`idPeixe`, `nome`, `pesoMedio`, `preco`) VALUES  
(1, 'Faneca', '0.2', '5.0'),  
(2, 'Carapau', '18.5', '6.0'),  
(3, 'Dourada', '3.5', '12.5'),  
(4, 'Pescada', '1.2', '7.5'),  
(5, 'Cavala', '2.0', '8.0'),  
(6, 'Atum', '12.0', '15.0'),  
(7, 'Peixe espada', '100.0', '12.0'),  
(8, 'Peixe gato', '200.0', '14.5');
```

```
CREATE TABLE `pescado` (  
  `idPescado` int(11) NOT NULL,  
  `idPescaria` int(11) NOT NULL,  
  `idPeixe` int(11) NOT NULL,  
  `peso` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
INSERT INTO `pescado` (`idPescado`, `idPescaria`, `idPeixe`, `peso`) VALUES
(1, 1, 1, 10000),
(2, 1, 5, 20000),
(3, 2, 6, 15000),
(4, 2, 4, 25000),
(5, 3, 2, 8000),
(6, 4, 8, 7000),
(7, 5, 7, 12000),
(8, 5, 8, 1000),
(9, 6, 4, 3000),
(10, 6, 2, 5000),
(11, 7, 1, 8500),
(12, 8, 6, 9000),
(13, 9, 6, 13000),
(14, 10, 4, 4000),
(15, 11, 7, 5000),
(16, 12, 8, 20000);
```

```
CREATE TABLE `pescador` (
  `idPescador` int(11) NOT NULL,
  `nome` varchar(100) NOT NULL,
  `cidade` varchar(100) NOT NULL,
  `idade` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Extraindo dados da tabela `pescador`
--
```

```
INSERT INTO `pescador` (`idPescador`, `nome`, `cidade`, `idade`) VALUES
(1, 'Marco Rocha', 'Vila do Conde', 52),
(2, 'Miguel Abrantes', 'Aveiro', 44),
(3, 'Paulo Trindade', 'Afurada', 46),
(4, 'Jorge Vieira', 'Afurada', 45),
(5, 'Carlos Santana', 'Mira', 45),
(6, 'Daniel Oliveira', 'Ilhavo', 66),
(7, 'Ricardo Abreu', 'Ilhavo', 68),
(8, 'Luis Ralha', 'Vila do Conde', 35);
```

```
CREATE TABLE `pescador_pescaria` (
  `idPescadorPescaria` int(11) NOT NULL,
  `idPescador` int(11) NOT NULL,
  `idPescaria` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
INSERT INTO `pescador_pescaria` (`idPescadorPescaria`, `idPescador`, `idPescaria`) VALUES
(1, 1, 1),
(2, 2, 1),
(3, 3, 2),
(4, 4, 2),
(5, 5, 3),
(6, 8, 3),
(7, 6, 4),
(8, 5, 5),
(9, 8, 5),
(10, 1, 5),
(11, 2, 6),
(12, 4, 6),
(13, 3, 7),
(14, 8, 8),
(15, 6, 8),
(16, 5, 9),
(17, 1, 9),
(18, 1, 10),
(19, 3, 10),
(20, 4, 11),
(21, 8, 11),
(22, 5, 12);
```

```
CREATE TABLE `pescaria` (
  `idPescaria` int(11) NOT NULL,
  `dataReg` text NOT NULL,
  `hora` time NOT NULL,
  `duracao` int(11) NOT NULL,
  `idBarco` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
INSERT INTO `pescaria` (`idPescaria`, `dataReg`, `hora`, `duracao`, `idBarco`) VALUES
(1, '2017-1-1', '06:00:00', 2, 1),
(2, '2017-1-1', '07:30:00', 1, 2),
(3, '2017-1-1', '08:00:00', 1, 4),
(4, '2017-1-2', '05:00:00', 4, 3),
(5, '2017-1-2', '05:30:00', 6, 4),
(6, '2017-1-2', '06:00:00', 2, 2),
(7, '2017-1-2', '06:30:00', 3, 6),
(8, '2017-1-3', '07:00:00', 4, 2),
(9, '2017-1-3', '07:30:00', 5, 4),
(10, '2017-1-4', '08:00:00', 1, 2),
(11, '2017-1-4', '09:30:00', 1, 3),
(12, '2017-1-5', '06:00:00', 3, 6);
```





```
ALTER TABLE `barco`  
  ADD PRIMARY KEY (`idBarco`);
```

```
ALTER TABLE `peixe`  
  ADD PRIMARY KEY (`idPeixe`);
```

```
ALTER TABLE `pescado`  
  ADD PRIMARY KEY (`idPescado`),  
  ADD KEY `idPescaria` (`idPescaria`),  
  ADD KEY `idPeixe` (`idPeixe`);
```

```
ALTER TABLE `pescador`  
  ADD PRIMARY KEY (`idPescador`);
```

```
ALTER TABLE `pescador_pescaria`  
  ADD PRIMARY KEY (`idPescadorPescaria`),  
  ADD KEY `idPescador` (`idPescador`),  
  ADD KEY `idPescaria` (`idPescaria`);
```

```
ALTER TABLE `pescaria`  
  ADD PRIMARY KEY (`idPescaria`),  
  ADD KEY `idBarco` (`idBarco`);
```

```
ALTER TABLE `barco`  
  MODIFY `idBarco` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
```

```
ALTER TABLE `peixe`  
  MODIFY `idPeixe` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
```

```
ALTER TABLE `pescado`  
  MODIFY `idPescado` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;
```

```
ALTER TABLE `pescador`  
  MODIFY `idPescador` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
```

```
ALTER TABLE `pescador_pescaria`  
  MODIFY `idPescadorPescaria` int(11) NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=23;
```

```
ALTER TABLE `pescaria`  
  MODIFY `idPescaria` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;
```

```
ALTER TABLE `pescado`  
  ADD CONSTRAINT `pescado_ibfk_1` FOREIGN KEY (`idPescaria`) REFERENCES `pescaria`  
  (`idPescaria`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `pescado_ibfk_2` FOREIGN KEY (`idPeixe`) REFERENCES `peixe`  
  (`idPeixe`) ON DELETE CASCADE ON UPDATE CASCADE;
```



```
ALTER TABLE `pescador_pescaria`  
  ADD CONSTRAINT `pescador_pescaria_ibfk_1` FOREIGN KEY (`idPescador`) REFERENCES  
`pescador` (`idPescador`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `pescador_pescaria_ibfk_2` FOREIGN KEY (`idPescaria`) REFERENCES  
`pescaria` (`idPescaria`) ON DELETE CASCADE ON UPDATE CASCADE;  
  
ALTER TABLE `pescaria`  
  ADD CONSTRAINT `pescaria_ibfk_1` FOREIGN KEY (`idBarco`) REFERENCES `barco`  
(`idBarco`) ON DELETE CASCADE ON UPDATE CASCADE;
```



-- 1 Apresente os pescadores que saíram para pescar na data 2017-1-3

idPescador	nome
1	Marco Rocha
5	Carlos Santana
6	Daniel Oliveira
8	Luis Ralha

-- 2 Apresente as espécies de peixe pescadas na data 2017-1-3

idPeixe	nome
6	Atum

-- 3 Projete o número de pescarias realizadas no dia 2 de janeiro de 2017.

numero
4

-- 4 Apresente o número total de barcos que suportem um peso máximo superior a 35.000 kg

numero
4



-- 5 Apresente o número de pescadores que saíram por viagem.

idPescaria	dataReg	numero
1	2017-1-1	2
2	2017-1-1	2
3	2017-1-1	2
4	2017-1-2	1
5	2017-1-2	3
6	2017-1-2	2
7	2017-1-2	1
8	2017-1-3	2
9	2017-1-3	2
10	2017-1-4	2
11	2017-1-4	2
12	2017-1-5	1

-- 6 Projete o número de espécies de peixe diferente capturados em cada pescaria.

idPescaria	dataReg	numero
1	2017-1-1	2
2	2017-1-1	2
3	2017-1-1	1
4	2017-1-2	1
5	2017-1-2	2
6	2017-1-2	2
7	2017-1-2	1
8	2017-1-3	1
9	2017-1-3	1
10	2017-1-4	1
11	2017-1-4	1
12	2017-1-5	1



-- 7 Projete as pescarias cujo o peso total de peixes capturados excedeu os 10.000 kg

idPescaria	dataReg
1	2017-1-1
2	2017-1-1
5	2017-1-2
9	2017-1-3
12	2017-1-5

-- 8 Projete as pescarias nas quais a idade média dos pescadores foi superior a 50 anos de idade?

idPescaria	dataReg
4	2017-1-2
8	2017-1-3

-- 9 Projete os barcos que nunca capturaram Atum. (A solução pode estar errada)

idBarco	nome
1	Aurélio
3	Batista
5	Evaristo
6	Afonso

-- 10 Projete os barcos nos quais 'Paulo Trindade' nunca pescou? (A solução pode estar errada)

idBarco	nome
1	Aurélio
3	Batista
4	Fortunato
5	Evaristo
6	Afonso



-- 11 Projete o número de pescarias feitas por cada pescador? Considere que alguns pescadores podem nunca ter feito uma viagem

idPescador	nome	numero
7	Ricardo Abreu	0
6	Daniel Oliveira	2
2	Miguel Abrantes	2
4	Jorge Vieira	3
3	Paulo Trindade	3
5	Carlos Santana	4
8	Luis Ralha	4
1	Marco Rocha	4

-- 12 Quantas vezes cada espécie de peixe foi capturado? Considere que algumas espécies de peixe podem nunca ter sido capturados.

idPeixe	nome	numero
6	Atum	3
8	Peixe gato	3
4	Pescada	3
2	Carapau	2
1	Faneca	2
7	Peixe espada	2
5	Cavala	1
3	Dourada	0

-- 13 Em qual pescaria, ou pescarias, se verificou a tripulação mais nova?

idPescaria	dataReg	media
3	2017-1-1	40.0000
11	2017-1-4	40.0000



-- 14 Em qual ou quais pescarias foi pescada a menor quantidade de peixe?

idPescaria	dataReg	peso
10	2017-1-4	4000

-- 15 Quais pescadores fizeram pescarias em que o limite de peso do barco foi ultrapassado?

-- 16 Quais peixes foram capturados em pescarias em que o limite de peso do barco foi ultrapassado?