

--1

```
SELECT * FROM Funcionario;
```

--2

```
SELECT Nome, CPF FROM Funcionario;
```

--3

```
SELECT * FROM Funcionario WHERE Sexo = 'Feminino';
```

--4

```
SELECT * FROM Funcionario WHERE Sexo = 'Masculino';
```

--5

```
SELECT * FROM Dependente WHERE Sexo = 'Masculino';
```

--6

```
SELECT * FROM Funcionario WHERE Nome LIKE 'M%';
```

--7

```
SELECT * FROM Dependente WHERE Nome LIKE 'P%';
```

--8

```
SELECT * FROM Funcionario WHERE Nome LIKE '%Cruz%';
```

--9

```
SELECT * FROM Dependente WHERE Sobrenome = 'Cruz';
```

--10

```
SELECT ProjetoID, MIN(QtdeHoras) FROM TrabalhaProjeto GROUP BY ProjetoID;
```

--11

```
SELECT ProjetoID, AVG(QtdeHoras) FROM TrabalhaProjeto GROUP BY ProjetoID;
```

--12

```
SELECT Sexo, COUNT(1) FROM Funcionario GROUP BY Sexo;
```

--13

```
SELECT TOP 2 * FROM Funcionario ORDER BY DataNascimento ASC;
```

--14

```
SELECT TOP 3 * FROM Funcionario ORDER BY DataNascimento DESC;
```

--15

```
SELECT D.Codigo, D.Nome AS NomeDependente, F.Nome AS NomeFuncionario FROM Dependente D INNER JOIN Funcionario F ON D.FuncionarioID = F.FuncionarioID;
```

--16

```
SELECT F.*, TP.ProjetoID FROM Funcionario F INNER JOIN TrabalhaProjeto TP ON F.FuncionarioID = TP.FuncionarioID;
```

--17

```
SELECT P.*, F.Nome AS NomeFuncionario FROM Projeto P INNER JOIN TrabalhaProjeto TP ON P.ProjetoID = TP.ProjetoID INNER JOIN Funcionario F ON TP.FuncionarioID = F.FuncionarioID;
```

--18

```
SELECT * FROM Funcionario LEFT JOIN Dependente ON Funcionario.FuncionarioID = Dependente.FuncionarioID WHERE Dependente.DependenteID IS NULL;
```

--19

```
SELECT F.Nome, COUNT(D.DependenteID) AS QtdeDependente
```

```
FROM Funcionario F
```

```
LEFT JOIN Dependente D ON F.FuncionarioID = D.FuncionarioID
```

```
GROUP BY F.Nome;
```

--20

```
SELECT F.Nome, COALESCE(SUM(TP.QtdeHoras), 0) AS QtdeHoras
```

```
FROM Funcionario F
```

```
LEFT JOIN TrabalhaProjeto TP ON F.FuncionarioID = TP.FuncionarioID
```

```
GROUP BY F.Nome;
```

--21

```
SELECT F.Nome, COALESCE(COUNT(D.DependenteID), 0) AS QtdeDependente
```

```
FROM Funcionario F
```

```
LEFT JOIN Dependente D ON F.FuncionarioID = D.FuncionarioID
```

```
GROUP BY F.Nome;
```

--22

```

SELECT D.Nome, COALESCE(SUM(TP.QtdeHoras), 0) AS QtdeHoras
FROM Funcionario F
LEFT JOIN TrabalhaProjeto TP ON F.FuncionarioID = TP.FuncionarioID
LEFT JOIN Projeto P ON TP.ProjetoID = P.ProjetoID
LEFT JOIN Departamento D ON P.DepartamentoID = D.DepartamentoID
GROUP BY D.Nome;

```

--23

```

SELECT D.Nome
FROM Departamento D
LEFT JOIN Projeto P ON D.DepartamentoID = P.DepartamentoID
WHERE P.ProjetoID IS NULL;

```

--24

```

SELECT F.Nome
FROM Funcionario F
LEFT JOIN Dependente D ON F.FuncionarioID = D.FuncionarioID
WHERE D.DependenteID IS NULL;

```

--25

```

SELECT MONTH(DataNascimento) AS Mes, COUNT(*) AS Qtde
FROM Funcionario
GROUP BY MONTH(DataNascimento);

```

--26

```

SELECT YEAR(DataNascimento) AS Ano, MONTH(DataNascimento) AS Mes, COUNT(*) AS Qtde
FROM Funcionario
GROUP BY YEAR(DataNascimento), MONTH(DataNascimento);

```

--27

```

SELECT Nome, DATEDIFF(YEAR, DataNascimento, GETDATE()) AS Idade
FROM Funcionario;

```

--28

```
CREATE VIEW vw_Funcionarios AS SELECT Nome, FuncionarioID, DataNascimento FROM Funcionario;
```

```
--29
```

```
CREATE VIEW vw_ProjetosPorIdade AS
```

```
SELECT DATEDIFF(YEAR, F.DataNascimento, GETDATE()) AS Idade, COUNT(TP.ProjetoID) AS  
QtdeProjetos
```

```
FROM Funcionario F
```

```
LEFT JOIN TrabalhaProjeto TP ON F.FuncionarioID = TP.FuncionarioID
```

```
GROUP BY DATEDIFF(YEAR, F.DataNascimento, GETDATE());
```

```
--30
```

```
CREATE PROCEDURE sp_ProjetosPorFuncionario @FuncionarioID INT
```

```
AS
```

```
BEGIN
```

```
    SELECT P.*
```

```
    FROM Projeto P
```

```
    INNER JOIN TrabalhaProjeto TP ON P.ProjetoID = TP.ProjetoID
```

```
    WHERE TP.FuncionarioID = @FuncionarioID;
```

```
END;
```

```
--31
```

```
CREATE PROCEDURE sp_FuncionariosPorProjeto @ProjetoID INT
```

```
AS
```

```
BEGIN
```

```
    SELECT F.*
```

```
    FROM Funcionario F
```

```
    INNER JOIN TrabalhaProjeto TP ON F.FuncionarioID = TP.FuncionarioID
```

```
    WHERE TP.ProjetoID = @ProjetoID;
```

```
END;
```

```
--32
```

```
CREATE PROCEDURE sp_FuncionariosAposData @Data date
```

```
AS
```

```

BEGIN

    SELECT *

    FROM Funcionario

    WHERE DataNascimento > @Data;

END;

--33

CREATE PROCEDURE sp_AtualizarDataNascimento @FuncionarioID INT, @NovaData date

AS

BEGIN

    UPDATE Funcionario

    SET DataNascimento = @NovaData

    WHERE FuncionarioID = @FuncionarioID;

END;

--34

CREATE FUNCTION fn_TotalHorasProjeto (@ProjetoID INT)

RETURNS NUMERIC(12, 2)

AS

BEGIN

    DECLARE @TotalHoras NUMERIC(12, 2);

    SELECT @TotalHoras = SUM(QtdeHoras)

    FROM TrabalhaProjeto

    WHERE ProjetoID = @ProjetoID;

    RETURN @TotalHoras;

END;

--35

CREATE FUNCTION fn_ProjetosPorHoras (@QtdeHoras NUMERIC(12, 2))

RETURNS TABLE

AS

```

RETURN

```
SELECT P.*  
  
FROM Projeto P  
  
INNER JOIN TrabalhaProjeto TP ON P.ProjetoID = TP.ProjetoID  
  
GROUP BY P.ProjetoID  
  
HAVING SUM(TP.QtdeHoras) >= @QtdeHoras;
```

--36

Funções retornam valores, enquanto as procedures não retornam valores diretamente. Funções podem ser usadas em consultas SQL, enquanto procedures são chamadas explicitamente.

--37, 38 e 39

```
CREATE FUNCTION fn_AniversariantesNoIntervalo (@Inicio date, @Fim date)  
  
RETURNS TABLE  
  
AS  
  
RETURN
```

```
SELECT F.Nome  
  
FROM Funcionario F  
  
WHERE F.DataNascimento BETWEEN @Inicio AND @Fim;
```

```
CREATE PROCEDURE sp_PessoasPorDepartamento @DepartamentoNome varchar(100) = NULL  
  
AS  
  
BEGIN
```

```
IF @DepartamentoNome IS NULL  
  
    SELECT D.Nome AS Departamento, F.Nome AS Funcionario  
  
    FROM Funcionario F  
  
    LEFT JOIN TrabalhaProjeto TP ON F.FuncionarioID = TP.FuncionarioID  
  
    LEFT JOIN Projeto P ON TP.ProjetoID = P.ProjetoID  
  
    LEFT JOIN Departamento D ON P.DepartamentoID = D.DepartamentoID;  
  
ELSE  
  
    SELECT D.Nome AS Departamento, F.Nome AS Funcionario
```

```
FROM Funcionario F

LEFT JOIN TrabalhaProjeto TP ON F.FuncionarioID = TP.FuncionarioID

LEFT JOIN Projeto P ON TP.ProjetoID = P.ProjetoID

LEFT JOIN Departamento D ON P.DepartamentoID = D.DepartamentoID

WHERE D.Nome = @DepartamentoNome;

END;
```