

Complementos de Bases de Dados

2022/2023

Licenciatura em Engenharia Informática



Laboratório 1 – Introdução ao SQL Server 2019

Objetivos:

- Revisão SQL
- Familiarização com o ambiente Management Studio
- Common Table Expressions - CTE

O enunciado está dividido em 2 partes:

I. Síntese da revisão da matéria SQL com exemplos de aplicação

II. Exercícios para prática.

Os exemplos e exercícios pressupõem a instalação prévia do MS SQL Server + Management Studio e o carregamento da base de dados Adventure Works LT 2019.

Preparação

1. Criar o diagrama da base de dados
2. Selecionar a base de dados: *AdventureWorksLT2019*
3. Escolher a opção *Database Diagrams -> New Database Diagram*
4. Escolher todas as tabelas
5. Guardar o diagrama

I. Resumo da Matéria SQL

Acompanhando a revisão do docente execute os exemplos representativos de vários aspetos da linguagem SQL. Estes exemplos permitiram a revisão da sintaxe da linguagem, paradigma do Modelo Relacional e familiarização com a base de dados de suporte aos exercícios. Observe os resultados produzidos.

SELECT: Projeções, Alias e Order By

- `select c.FirstName as 'Nome', c.LastName as 'Apelido', c.EmailAddress
from salesLT.Customer c;`
- `select c.FirstName as "Nome", c.LastName as "Apelido", c.EmailAddress
from salesLT.Customer c
order by Nome ASC, Apelido DESC`

WHERE, Datatypes e Operadores

- `select c.CustomerID, c.FirstName as 'Nome', c.LastName as 'Apelido',
c.EmailAddress
from salesLT.Customer c
Where c.CustomerID > 20;`
- `select c.CustomerID, c.Title, c.FirstName as 'Nome', c.LastName as 'Apelido',
c.EmailAddress
from salesLT.Customer c
Where c.CustomerID > 20 AND c.Title = 'Mr.';`
- `select c.CustomerID, c.FirstName as "Nome", c.LastName as "Apelido",
c.EmailAddress
from salesLT.Customer c
Where c.CustomerID BETWEEN 20 AND 30;`
- `select distinct c.CustomerID, c.FirstName as 'Nome', c.LastName as 'Apelido',
c.EmailAddress
from salesLT.Customer c
Where c.FirstName like 'a%';`
- `select distinct c.CustomerID, c.FirstName as 'Nome', c.LastName as 'Apelido',
c.EmailAddress
from salesLT.Customer c
Where c.FirstName like '_[il]%';`
- `select distinct c.CustomerID, c.FirstName as 'Nome', c.LastName as 'Apelido',
c.EmailAddress
from salesLT.Customer c
Where c.FirstName like '[a-c]%';`
- `select c.CustomerID, c.EmailAddress
from salesLT.Customer c
Where c.FirstName in ('Orlando', 'Rosmarie');`
- `select c.CustomerID, c.FirstName, c.MiddleName, c.LastName
from salesLT.Customer c
Where c.MiddleName is null`
- `select c.CustomerID, c.FirstName, c.MiddleName, c.LastName
from salesLT.Customer c
Where c.MiddleName is not null`

Funções Linha e Datatypes

- string

- `select c.FirstName, len(c.FirstName) as 'length'
from SalesLT.Customer c`
- `select upper(c.FirstName) as "FName", lower(c.LastName) as "LName"
from SalesLT.Customer c`
- `select concat(c.FirstName, space(1), c.LastName) as 'Full Name'
from SalesLT.Customer c`

- `select concat_ws(' ',c.FirstName, c.LastName) as 'Full Name'`
`from SalesLT.Customer c`
- `select charindex('@', c.EmailAddress) 'pos of @', c.EmailAddress`
`from SalesLT.Customer c`
- `select LEFT(c.EmailAddress,4) as 'email name'`
`from SalesLT.Customer c;`
- `select LEFT(c.EmailAddress,charindex('@', c.EmailAddress)-1) as 'email name'`
`from SalesLT.Customer c;`
- `select substring(c.firstName,1,1) as 'Initial'`
`from SalesLT.Customer c;`

- datetime

- `select distinct datename(yy,c.ModifiedDate) as 'year',`
`datename(qq,c.ModifiedDate) as 'quarter'`
`from SalesLT.Customer c`
`order by year, quarter;`
- `select`
`datepart(dw,c.ModifiedDate) as 'weekDay',`
`datepart(wk,c.ModifiedDate) as 'semana',`
`datepart(ISO_WEEK,c.ModifiedDate) as 'semanaISO'`
`from SalesLT.Customer c;`
- `select`
`c.ModifiedDate,`
`day(c.ModifiedDate)as 'dia',`
`month(c.ModifiedDate)as 'mes',`
`year(c.ModifiedDate)as 'ano'`
`from SalesLT.Customer c`
- `select`
`format(c.ModifiedDate,'MM-yyyy'),`
`datediff(m,c.ModifiedDate,getdate()) as 'meses'-- return integer`
`from SalesLT.Customer c;`
- `select`
`format(c.ModifiedDate,'dd-MM-yyyy'),`
`dateadd(month,2, c.ModifiedDate) as '+ 2 month result'`
`from SalesLT.Customer c;`

- condicionais

- `select concat_ws(' ',c.FirstName, isnull(MiddleName,''), c.LastName)`
`from SalesLT.Customer c;`
- `select IIF(c.MiddleName is null,`
`concat_ws('.',left(c.firstName,1) , left(c.lastName,1),''),`
`concat_ws('.',left(c.firstName,1) , left(c.middleName,1),`
`left(c.lastName,1),''))`
`from SalesLT.Customer c`

-- Case 1

- `select case c.MiddleName`
`when null then concat_ws('.',left(c.firstName,1) , left(c.lastName,1),'')`
`ELSE concat_ws('.',left(c.firstName,1) , left(c.middleName,1),`
`left(c.lastName,1),'')`
`END`
`from SalesLT.Customer c`

```
-- Case 2
• select case
    when c.MiddleName is null then concat_ws('.',left(c.firstName,1) ,
    left(c.lastName,1),'')
    ELSE concat_ws('.',left(c.firstName,1) , left(c.middleName,1),
    left(c.lastName,1),'')
    END
from SalesLT.Customer c
```

Group By, Funções Grupo e Having

- select distinct a.CountryRegion
from SalesLT.Address a
- select a.CountryRegion
from SalesLT.Address a
group by a.CountryRegion
- select a.CountryRegion, count(a.City) '# de cidades'
from SalesLT.Address a
group by a.CountryRegion
- select ProductCategoryID, avg(p.listPrice) as 'media da categoria'
from SalesLT.Product p
group by ProductCategoryID;
- select ProductCategoryID, max(p.listPrice) as 'max da categoria'
from SalesLT.Product p
group by ProductCategoryID;
- select ProductCategoryID, sum(p.Weight) as 'sum da categoria'
from SalesLT.Product p
where p.Weight is not null
group by ProductCategoryID
having sum(p.Weight) > 1000;

Joins

- select oh.SalesOrderID, c.FirstName, c.LastName
from SalesLT.SalesOrderHeader oh
join SalesLT.Customer c on oh.CustomerID=c.CustomerID;

- "self" join

- select ppc.Name 'Parent', pcc.Name 'child'
from SalesLT.ProductCategory pcc
join SalesLT.ProductCategory ppc on ppc.ProductCategoryID
=ppc.ParentProductCategoryID
- select distinct c.FirstName, oh.SalesOrderID
from SalesLT.Customer c
right join SalesLT.SalesOrderHeader oh on c.CustomerID=oh.CustomerID;
- select distinct c.FirstName, oh.SalesOrderID
from SalesLT.Customer c
left join SalesLT.SalesOrderHeader oh on c.CustomerID=oh.CustomerID;
- select c.FirstName, c.SalesPerson, a.salesPerson
from SalesLT.Customer c
cross join (select SalesPerson from SalesLT.Customer) a ;

Subqueries

- No select/projeções

```
select
    ((oh2.TotalDue)/(select sum(oh.TotalDue) from SalesLT.SalesOrderHeader oh))*100 as
    'perc'
from SalesLT.SalesOrderHeader oh2
```

-- alternativa

```
select
    ((oh2.TotalDue)/a.calc_sum)*100 as 'perc'
from SalesLT.SalesOrderHeader oh2
    cross join (select sum(oh.TotalDue) calc_sum from SalesLT.SalesOrderHeader oh) a;
```

- No where: 1 resultado (1 linha e 1 coluna)

```
select oh.CustomerID
from SalesLT.SalesOrderHeader oh
where oh.TotalDue = (select max(oh2.TotalDue)
                    from salesLT.SalesOrderHeader oh2
                    )
```

- No where: 1 coluna várias linhas (também semelhantemente/aplicável no HAVING)

```
select oh.SalesOrderID
from SalesLT.SalesOrderHeader oh
where oh.CustomerID in (select c.CustomerID
                       from salesLT.Customer c
                       where
                           right(c.SalesPerson,len(c.salesPErson)-charindex('\',c.SalesPerson)) = 'linda3'
                       )
```

-- alternativa com join

```
select oh.SalesOrderID
from SalesLT.SalesOrderHeader oh
    join SalesLT.Customer c on c.CustomerID=oh.CustomerID
where
    right(c.SalesPerson,len(c.salesPErson)-charindex('\',c.SalesPerson)) = 'linda3'
```

- Operador ANY (> que algum dos que estão no conjunto)

```
select oh.SalesOrderID
from SalesLT.SalesOrderHeader oh
where month(oh.OrderDate) = any (select distinct month(c.ModifiedDate)
                                from SalesLT.Customer c
                                where year(c.ModifiedDate) = 2006)
```

- Operador ALL (> que qualquer-todos os que estão no conjunto)

```
select oh.SalesOrderID
from SalesLT.SalesOrderHeader oh
where month(oh.OrderDate) > all (select distinct month(c.ModifiedDate)
                                from SalesLT.Customer c
                                where year(c.ModifiedDate) = 2006)
```

- No from

```
select max(a.CountCustID)
from (select year(c.ModifiedDate) as 'ano', count(distinct c.CustomerID) 'CountCustID'
      from SalesLT.Customer c
      group by year(c.ModifiedDate)) a ;
```

Common Table Expressions

```
-- With CTE
with conta as
(
    select a.CountryRegion, count(*) ct
    from SalesLT.Address a
    group by a.CountryRegion
)
Select a.CountryRegion, count(*)
from SalesLT.Address a
group by a.CountryRegion
Having count(*) = (select max(ct) from conta)
```

Exemplo:

Devolver *region* e respectivas vendas totais, para regiões que totalizam vendas abaixo da média de vendas totais por região.

```
select *
from
    (select a.CountryRegion, sum(oh.TotalDue) sales
     from SalesLT.SalesOrderHeader oh
     join SalesLT.Customer c on c.CustomerID=oh.CustomerID
     join SalesLT.CustomerAddress ca on ca.CustomerID=c.CustomerID
     join SalesLT.Address a on a.AddressID=ca.AddressID
     group by a.CountryRegion) country_sales
join
    (select avg(x.sales) calc_avg
     from
        (select a.CountryRegion, sum(oh.TotalDue) sales
         from SalesLT.SalesOrderHeader oh
         join SalesLT.Customer c on c.CustomerID=oh.CustomerID
         join SalesLT.CustomerAddress ca on ca.CustomerID=c.CustomerID
         join SalesLT.Address a on a.AddressID=ca.AddressID
         group by a.CountryRegion) x) avg_sales
on country_sales.sales<avg_sales.calc_avg
```

```
-- alternativa
with sales as
(select a.CountryRegion, sum(oh.TotalDue) total
 from SalesLT.SalesOrderHeader oh
 join SalesLT.Customer c on c.CustomerID=oh.CustomerID
 join SalesLT.CustomerAddress ca on ca.CustomerID=c.CustomerID
 join SalesLT.Address a on a.AddressID=ca.AddressID
 group by a.CountryRegion)

select *
from sales
join (select avg(sales.total) cal_avg from sales) avg_sales
on sales.total<avg_sales.cal_avg
```

II. Exercícios

1. Listar todos os clientes (Nome Completo: primeiro, meio, último nome e email) ordenados decendentemente pelo último nome – 847 linhas;
2. Listar os clientes que não têm nenhuma ordem de compra (SalesLT.SalesOrderHeader) – 815 linhas;
3. Total monetário de vendas por Produto (somatório de OrderQty*UnitPrice);
4. O Produto com o maior valor monetário de vendas;
5. Listagem de produtos (nome e preço) da categoria “Bikes” – 97 linhas;
6. Listar apenas as categorias com mais de 20 produtos – 5 linhas.
7. Quantidades de produtos por categoria (mostrando o nome da categoria e o número de produtos associados), ordenados por número de produtos – 37 linhas;
8. A percentagem que o valor monetário de vendas representa em cada produto face ao valor total de vendas, ordenado pela % decendente (Obs: no cálculo do valor retirar o valor de desconto ao preço do produto);

-- Fim do Enunciado --