

# Consultas com PL/SQL

Jaisson Duarte

# Quem sou eu?

- Jaisson Duarte
- Técnico em Informática - IFRS Ibirubá
- Aluno 9º semestre de Ciência da Computação - IFRS Ibirubá
- Programador de Sistemas Pleno - Cotribá



jaisson



jaisson-duarte



jaisson.duarte@ibiruba.ifrs.edu.br



# Agenda

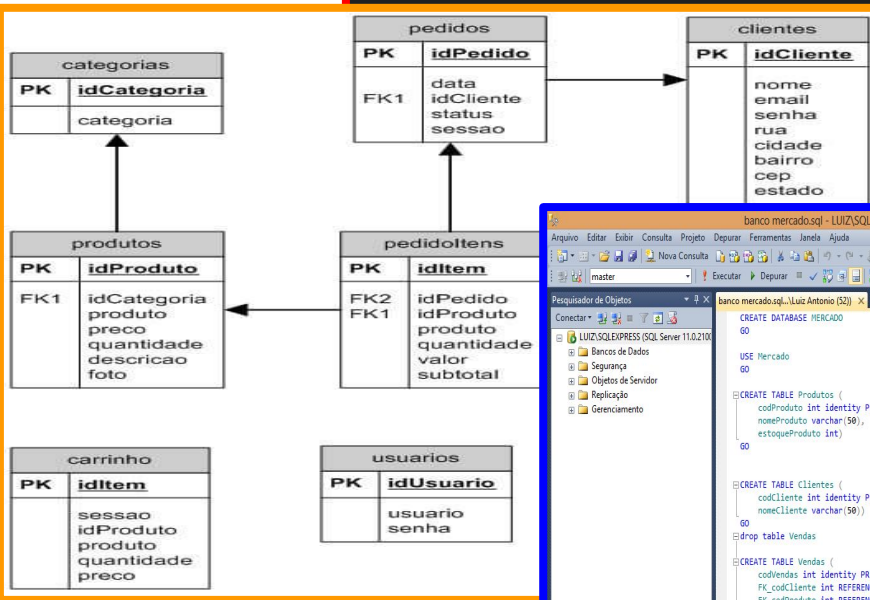
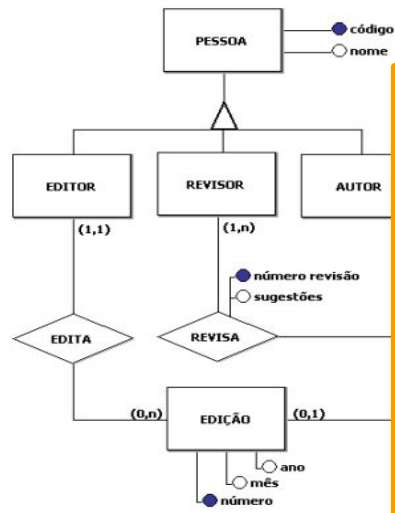
- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL



# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# Introdução



```
CREATE DATABASE MERCADO
GO

USE Mercado
GO

CREATE TABLE Produtos (
    codProduto int identity PRIMARY KEY,
    nomeProduto varchar(50),
    estoqueProduto int
)
GO

CREATE TABLE Clientes (
    codCliente int identity PRIMARY KEY,
    nomeCliente varchar(50)
)
GO

DROP TABLE Vendas

CREATE TABLE Vendas (
    codVendas int identity PRIMARY KEY,
    fk_codCliente int REFERENCES Clientes (codCliente),
    fk_codProduto int REFERENCES Produtos (codProduto),
    quantidadeVendas int,
    Data datetime DEFAULT getdate()
)
GO

INSERT Produtos Values ('Maça',500), ('Banana',300), ('Melancia',450)
INSERT Clientes Values ('Sergio'),('Carlos'),('Carine')
INSERT Vendas Values (1,1,10,default), (2,3,5,default),
(3,2,4,default), (1,3,10,default)
GO
```

Introdução

SQL

Structured Query Language

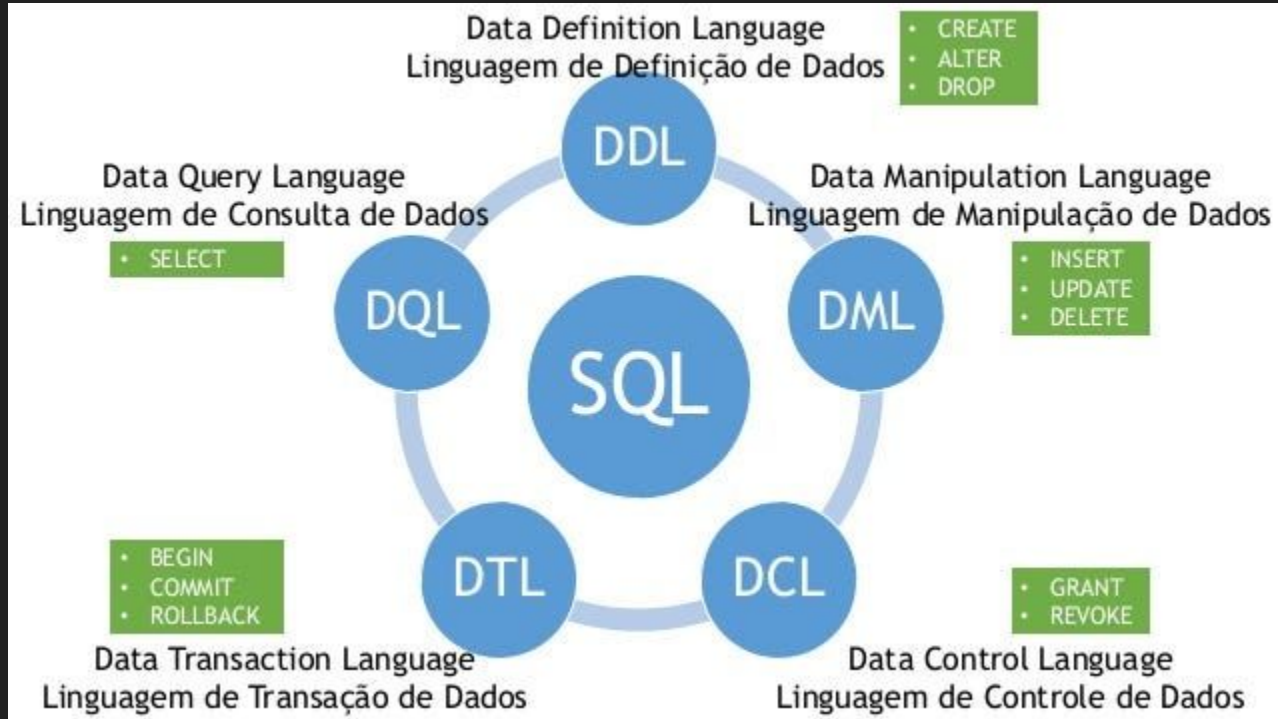
Linguagem de consulta estruturada



# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# Revisão SQL



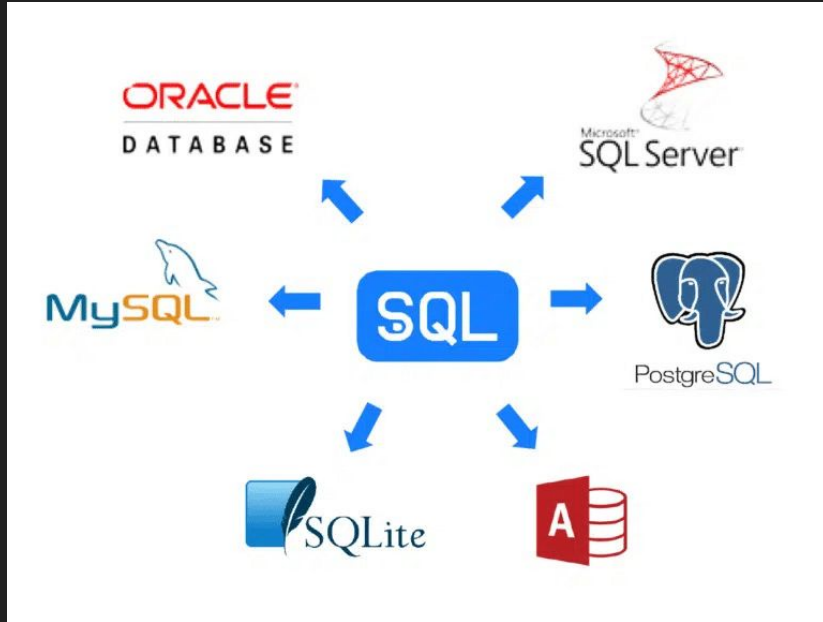




# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# SGBDs



## Dialetos

- SQL Server: Transact-SQL/ T-SQL
- PostgreSQL: PL/pgSQL
- Oracle: PL/SQL
- Firebird: 1,2,3...
- Access: JET SQL
- .....

# SGBDs

O padrão SQL ANSI é uma tentativa de padronização da linguagem SQL por organizações internacionais como o American National Standards Institute (traduz-se como "Instituto Nacional Americano de Padrões") ou ANSI.

Essa padronização ocorreu e foi necessária devido ao fato de terem surgidos vários dialetos de bancos que foram criados por diversos fabricantes.

Segue abaixo a listagem de como o padrão foi se desenvolvendo e melhorando com o tempo.

- **SQL-86** - Primeira versão da linguagem, lançada em 1986, consiste basicamente na versão inicial da linguagem criada pela IBM.
- **SQL-92** - Lançada em 1992, inclui novos recursos tais como tabelas temporárias, novas funções, expressões nomeadas, valores únicos, instrução CASE etc.
- **SQL:1999 (SQL3)** - Lançada em 1999, foi a versão que teve mais recursos novos significativos, entre eles: a implementação de expressões regulares, recursos de orientação a objetos, queries recursivas, triggers, novos tipos de dados (boolean, LOB, array e outros), novos predicados etc.
- **SQL:2003** - Lançada em 2003, inclui suporte básico ao padrão XML, sequências padronizadas, instrução MERGE, colunas com valores auto-incrementais etc.
- **SQL:2006** - Lançada em 2006, não inclui mudanças significativas para as funções e comandos SQL. Contempla basicamente a interação entre SQL e XML.

Fonte : [www.fabioprado.net](http://www.fabioprado.net)

# SGBDs

iso.org/standard/76583.html



Standards

About us

News

Taking part

Store

Applications

OBP

English

Search



← ICS ← 35 ← 35.060

## ISO/IEC 9075-1:2023

Information technology — Database languages SQL — Part 1: Framework (SQL/Framework)

### General information

 Preview

Status : Published

Publication date : 2023-06

Edition : 6

Number of pages : 74

Technical Committee : [ISO/IEC JTC 1/SC 32](#) Data management and interchange

ICS : [35.060](#) Languages used in information technology

### Buy this standard

Format

Language

✓ PDF

English

CHF 187

 Buy

# SGBDs

The screenshot shows the Oracle SQL Language Reference page for Release 19. The page has a dark header with the Oracle logo, a hamburger menu, and the text "Help Center". A search bar contains "SQL Language Reference" and a "Search" button. Below the header, a breadcrumb trail reads "Database / Oracle / Oracle Database / Release 19". The main title "SQL Language Reference" is prominently displayed. On the left, a sidebar contains a tree view of the document structure. The main content area is titled "Oracle Compliance to Core SQL" and includes an introductory paragraph, a list of supported products, and a detailed list of support levels for various SQL features.

Database / Oracle / Oracle Database / Release 19

## SQL Language Reference

- ▶ 13 SQL Statements: COMMIT to CREATE JAVA
- ▶ 14 SQL Statements: CREATE LIBRARY to CREATE SCHEMA
- ▶ 15 SQL Statements: CREATE SEQUENCE to DROP CLUSTER
- ▶ 16 SQL Statements: DROP CONTEXT to DROP JAVA
- ▶ 17 SQL Statements: DROP LIBRARY to DROP SYNONYM
- ▶ 18 SQL Statements: DROP TABLE to LOCK TABLE
- ▶ 19 SQL Statements: MERGE to UPDATE
- ▶ A How to Read Syntax Diagrams
- ▶ B Automatic and Manual Locking Mechanisms During SQL Operations
- ▼ C Oracle and Standard SQL
  - ANSI Standards
  - ISO Standards
  - Oracle Compliance to Core SQL**
  - Oracle Support for Optional Features of SQL/Foundation
  - Oracle Compliance with SQL/CLI

### Oracle Compliance to Core SQL

The ANSI and ISO SQL standards require conformance claims to state the type of conformance and the implemented facilities. The minimum claim of conformance is called Core SQL and is defined in Part 2, SQL/Foundation, and Part 11, SQL/Schemata, of the standard. The following products provide full or partial conformance with Core SQL as described in the tables that follow:

- Oracle Database server, release 12.2
- OTT (Oracle Type Translator), release 12.2
- Pro\*C/C++, release 12.2
- Pro\*COBOL, release 12.2

The SQL standards conformance features can be used either as a guide to portability, or as a guide to functionality. From the standpoint of portability, the user is interested in conformance to both the precise syntax and semantics of the standard feature. From the standpoint of functionality, the user is less concerned about the precise syntax and more concerned with issues of semantics. The tables in this appendix use the following terms regarding support for standard syntax and semantics:

- Full Support: The feature is supported with standard syntax and semantics.
- Partial Support: Some, but not all, of the standard syntax is supported; whatever is supported has standard semantics.
- Enhanced Support: The standard semantics is supported, as well as additional functionality.
- Equivalent Support: The standard semantics is supported using non-standard syntax.
- Similar Support: Neither the standard's syntax nor semantics are supported precisely, but similar functionality is provided.

Oracle's support for the features of Core SQL is listed in [Table C-1](#):

# SGBDs

≡ exemplo3.sql ●

≡ exemplo3.sql > {} SELECT

```
1
2  PL/PgSQL
3  select * from pessoa where nome ~* 'Jaiisson'
4
5  SQL ANSI
6  SELECT * FROM PESSOA WHERE NOME LIKE '%JAISSON%'
7
```



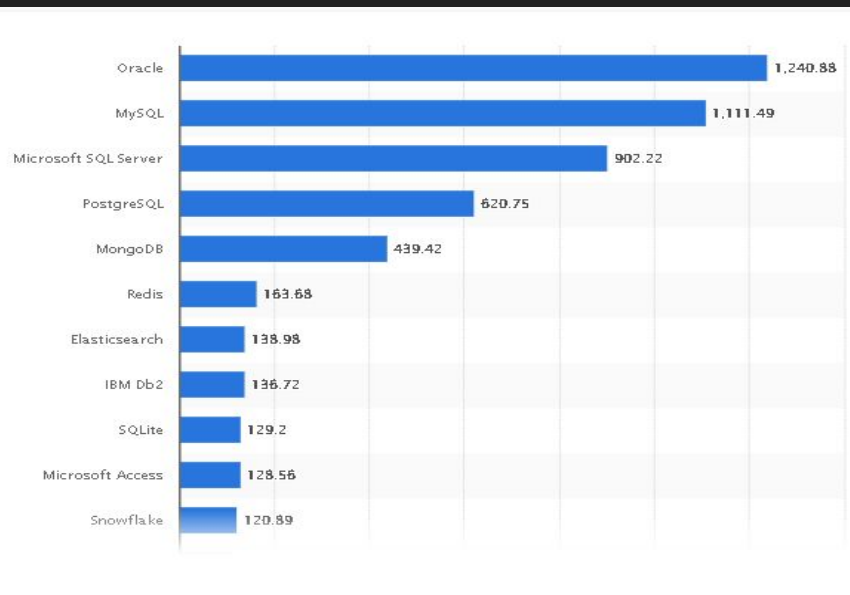
# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# Ranking popularidade database

Classificação			SGBD	Tipos de Banco de Dados	Pontuação		
maio 2023	abril 2023	maio 2022			maio 2023	abril 2023	maio 2022
1.	1.	1.	Oracle	Relacional	1232,64	4,36	-30,18
2.	2.	2.	MySQL	Relacional	1172,46	14,68	-29,64
3.	3.	3.	Microsoft SQL Server	Relacional	920,09	1,57	-21,11
4.	4.	4.	PostgreSQL	Relacional	617,9	9,49	2,61
5.	5.	5.	MongoDB	Orientado a Documento	436,61	-5,29	-41,63
6.	6.	6.	Redis	Não é SQL	168,13	-5,42	-10,89
7.	7.	7.	IBM Db2	Relacional	143,02	-2,48	-17,31
8.	8.	8.	Elasticsearch	Não é SQL	141,63	0,56	-16,06
9.	9.	↑10.	SQLite	Relacional	133,86	-0,68	-0,87
10.	10.	↓9.	Microsoft Access	Relacional	131,17	-0,2	-12,27

Fonte: Austrian IT Consulting, disponível em: <http://db-engines.com/en/>. Acesso em: 02/05/2023.



Fonte:  
<https://www.statista.com/statistics/809750/worldwide-popularity-ranking-data-base-management-systems/>





# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Live SQL
- Consultas

Oracle


**ORACLE®**


---

**D A T A B A S E**

# Oracle



 Oracle University





Treinamento


Certificação


Solutions

Buy



 Português (Brasil)





My Subscriptions

### Oracle Cloud Infrastructure and Technology Exams

Purchase a single-use exam for Technology or Oracle Cloud Infrastructure (OCI) exams delivered by Oracle.

**R\$ 1228**

Valid for 6 months

Add to Cart

Pay with your Learning Credit account

View details

### Oracle Cloud Applications DELTA Exams & OCI Multicloud Architect Associate Exam

Purchase a single-use exam for Oracle Cloud Applications Delta recertification exams delivered by Oracle.

Valid only for:  
Delta Exams where an exam credential has not expired

Or

Purchase a single-use exam attempt to take the OCI Multicloud Architect Associate exam delivered by Oracle.

**R\$ 626**

Valid for 6 months

Add to Cart

Pay with your Learning Credit account

View details

### Oracle Cloud Applications Exams

Purchase a single-use exam for Oracle Cloud Applications exams delivered by Oracle.

**R\$ 1228**

Valid for 6 months

Add to Cart

Pay with your Learning Credit account

View details

### Oracle Foundations Exams for Students

Purchase a single-use exam for an Oracle Foundations exam delivered by Oracle.

Valid only for:  
Java Foundations | 1Z0-811  
Oracle Database Foundations | 1Z0-006

**R\$ 476**

Valid for 6 months

Add to Cart

Pay with your Learning Credit account

View details



# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# Mercado de Trabalho

A média salarial de Desenvolvedor PL/SQL é de R\$ 12.000 por mês nessa localidade (Brasil). A remuneração variável de Desenvolvedor PL/SQL em Brasil é de R\$ 6.500, variando entre R\$ 810 e R\$ 20.521. As estimativas de salários têm como base 45 salários enviados de forma sigilosa ao Glassdoor por pessoas com o cargo de Desenvolvedor PL/SQL nessa localidade (Brasil).

Fonte: Glassdoor

## Salários em Brasil

Localização

Cidade ou área metropolitana

Buscar

ou

Encontre uma empresa específica

Nome da empresa

Buscar

Ordenar:

Popularidade ▼

Empresa

Faixa salarial base em (BRL)

Intervalo



### CWI Software

Desenvolvedor PL/SQL: prestador de serviços mensal

Cerca de R\$ 5 mil - R\$ 11 mil

4,6 ★

2 salários Ver 4 salários de todas as localizações

R\$ 5 mil

R\$ 11 mil



### CWI Software

Desenvolvedor PL/SQL: mensal

Cerca de R\$ 5 mil - R\$ 7 mil

4,6 ★

2 salários Ver 4 salários de todas as localizações

R\$ 5 mil

R\$ 7 mil



### Reply

Desenvolvedor PL/SQL: mensal

Cerca de R\$ 4 mil - R\$ 8 mil

4,2 ★

2 salários Ver 2 salários de todas as localizações

R\$ 4 mil

R\$ 8 mil



### Art IT

Desenvolvedor PL/SQL: mensal

Cerca de R\$ 4 mil - R\$ 7 mil

4,5 ★

2 salários Ver 2 salários de todas as localizações

R\$ 4 mil

R\$ 7 mil



### Deloitte

Desenvolvedor PL/SQL: mensal

Cerca de R\$ 11 mil - R\$ 12 mil

4 ★

1 salários Ver 1 salários de todas as localizações

R\$ 11 mil

R\$ 12 mil



### Robert Bosch

Desenvolvedor PL/SQL: mensal

Cerca de R\$ 7 mil - R\$ 8 mil

4,4 ★

1 salários Ver 1 salários de todas as localizações

R\$ 7 mil

R\$ 8 mil



### Linx

Desenvolvedor PL/SQL: mensal

Cerca de R\$ 11 mil - R\$ 11 mil

4,1 ★

1 salários Ver 1 salários de todas as localizações

R\$ 11 mil

R\$ 11 mil

Vagas ▾

Data do anúncio ▾

Nível de experiência ▾

Empresa ▾

Presencial/remota ▾

Candidatura simplificada

Todos os filtros

Desenvolvedor Plsql em: Brasil

75 resultados

Configurar alerta

**Integration - BTP Dats LAC**

SAP

São Leopoldo, Rio Grande do Sul, Brasil (Híbrido)



28 ex-alunos trabalham aqui

Promovida

**Desenvolvedor sql**

Netvagas

São Paulo, São Paulo, Brasil (Híbrido)

Há 3 dias

**DESENVOLVEDOR - ORACLE PLSQL**

Tata Consultancy Services

São Paulo, São Paulo, Brasil (Híbrido)



Recrutando agora

Promovida • Candidatura simplificada

**Oracle PL/SQL Developer**

Bunge

Gaspar, Santa Catarina, Brasil (Presencial)



8 ex-alunos trabalham aqui

Promovida

**ESPECIALISTA PHP**

Cast group

Brasil (Remoto)



1 ex-aluno trabalha aqui

Promovida • Candidatura simplificada

**Pessoa Engenheira/Analista de Desenvolvimento de Sistemas (Liderança Técnica)**

Siemens

São Paulo, Brasil (Híbrido)



Recrutando agora

Promovida • 9 candidaturas

**Oracle PL/SQL Developer**

Bunge · Gaspar, Santa Catarina, Brasil · há 1 semana · 54 candidaturas



Presencial · Tempo integral · Júnior



+ de 10.001 funcionários · Produtos alimentícios



8 ex-alunos trabalham aqui

Veja como você se compara a 40 candidatos. [Experimente o Premium grátis](#)

Competências: PL/SQL, SQL, e mais 8

Ver verificações relacionadas a este anúncio de vaga. [Exibir tudo](#)

Aplicar

Salvar

**Sobre a vaga**

Location : BAL - GASPAR

City : Gaspar

State : Santa Catarina (BR-SC)

Country : Brazil (BR)

Requisition Number : 31537

Bunge has an exciting opportunity available for a Oracle PL/SQL Developer. In this role you will be part of a global

team working on challenging, meaningful projects impacting core business activities.

Since 1818, Bunge has been connecting farmers to consumers to deliver essential food, feed, and fuel to the world. Looking to the future, our ambition is to continuously reinvent ourselves, leveraging data to be at the forefront of analytics, technology and talent to accomplish our purpose in a better, faster and simpler way. Bunge is committed to operating and thriving in the digital world – creating world class agile teams where teammates are empowered and encouraged to collaborate and test and learn to succeed.

At Bunge, people don't just come here to work, they come here to grow – solving challenges that directly impact the world with a diverse team of thinkers and doers.



[Candidatos](#) > [Buscar vagas](#) > [Programador PL SQL](#) > [RS](#)

Cargo ou palavra-chave

Cidade, estado ou região

[Rio Grande do Sul \(estado\)](#) X

▼ Mais filtros

Buscar

CATHO

▶ **AUMENTE A VISIBILIDADE DO SEU CV EM ATÉ 18X PARA MAIS DE 5 MIL VAGAS ANUNCIADAS NA CATHO TODOS OS DIAS**

[CONHECER PLANOS](#)

## 5 vagas de emprego de Programador PL SQL em Rio Grande do Sul

Total de anúncios: 5

[Mais relevantes](#)

## Desenvolvedor PL/SQL Pleno - Opportunity to Make a Difference

GRABS

A Combinar

**1 vaga:** Gramado - RS (1)

Atualizada em 11/09

We are desiring to recruit a versatile Desenvolvedor PL/SQL Pleno to join our stellar team at Huntz in Gramado, RS. Growing your career as a Full Time Desenvolvedor PL/SQL Pleno... [continuar lendo](#)

[Quero me candidatar](#)

## Pessoa Desenvolvedora PL/SQL Porto Alegre - RS and Remote Full-time employee

EMPREGARA

A Combinar

**1 vaga:** Porto Alegre - RS (1)

Atualizada em 31/08

Buscamos uma Pessoa Desenvolvedora PL/SQL, para trabalhar em nosso parceiro. Nossa área de Gestão de Outsourcing estará próxima a você, realizando reuniões 1:1 para garantir o... [continuar lendo](#)

[Quero me candidatar](#)

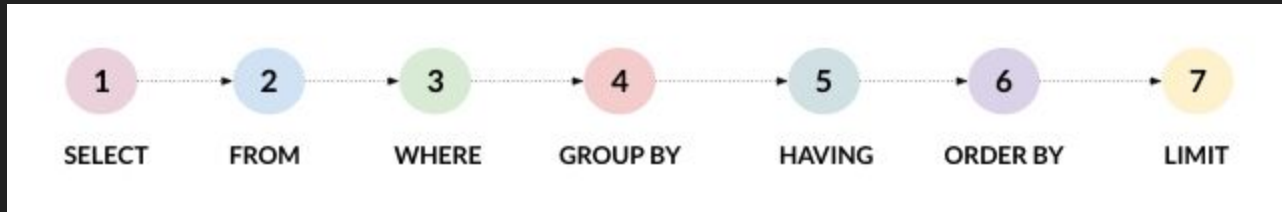
## Senior PL/SQL Developer - Remote Work - Exciting Challenge



# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# Estrutura de uma Consulta



# Estrutura de uma Consulta

≡ exemplo1.sql ●

≡ exemplo1.sql > {} SELECT

```
1  SELECT
2    (apresentar)
3  FROM
4    (tabela principal)
5  JOIN
6    (tabelas complementares)
7  WHERE
8    (filtro)
9  GROUP BY
10   (agrupar)
11  HAVING
12   (filtro agrupamento)
13  ORDER BY
14   (ordenação)
15  LIMIT
16   (limitar registros)
17
```

≡ exemplo2.sql ●

≡ exemplo2.sql > {} SELECT

```
1  ✓ SELECT c.nmCidade AS "Cidade"
2    COUNT(1) "Total Pessoas"
3  FROM cidade c
4  JOIN ENDereco e on (e.cdCidade = c.idCidade)
5  WHERE c.dsUF = 'RS'
6  GROUP BY c.nmCidade
7  HAVING COUNT(1) > 5000
8  ORDER BY 1
9  LIMIT 10
10
```



# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# Consulta

Cláusulas:

- FROM
- JOIN
- SUBCONSULTAS
- SUBSELECT
- TABELA DERIVADA
- CTE
- FUNÇÕES SIMPLES
- FUNÇÕES ANALÍTICAS
- SUPER AGRUPAMENTOS
- CROSS TAB

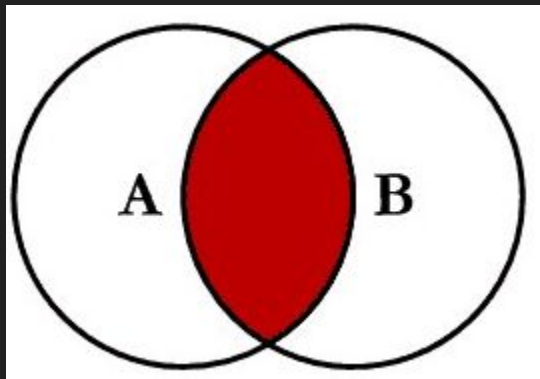
# FROM

```
Script000.sql > {} SELECT  
1  SELECT SYSDATE  
2  FROM DUAL
```

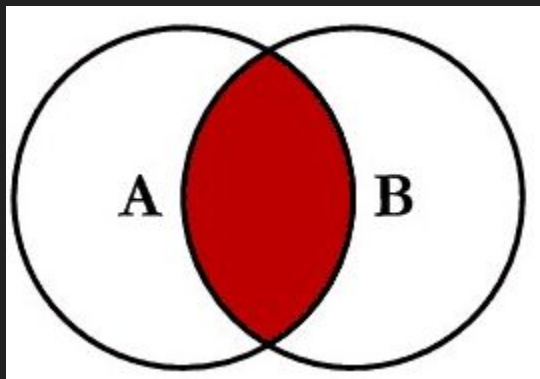
```
Script000.sql > {} SELECT  
1  SELECT idPessoa "Cód Pessoa",  
2  nmPessoa as nome  
3  FROM PESSOA
```

```
Script000.sql > {} SELECT  
1  SELECT SALDO.SALDODIA  
2  FROM TABLE (SALDO_CAIXA(SYSDATE)) SALDO  
3
```

# JOIN (SIMPLES / USING)



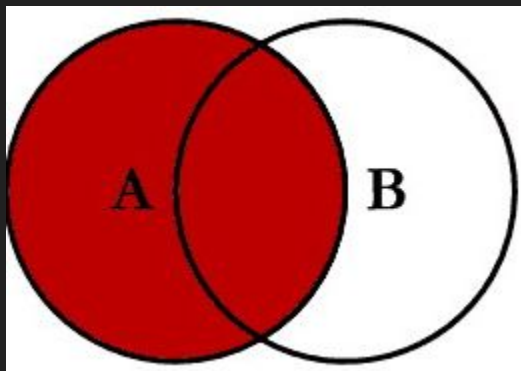
```
Script000.sql > {} SELECT
1  SELECT P.nmPessoa "Pessoa",
2         D.nmDependente "Dependente"
3  FROM PESSOA P
4  JOIN DEPENDENTE D ON (D.cdPessoa = P.cdPessoa)
5  WHERE D.cdPessoa IS NULL
```



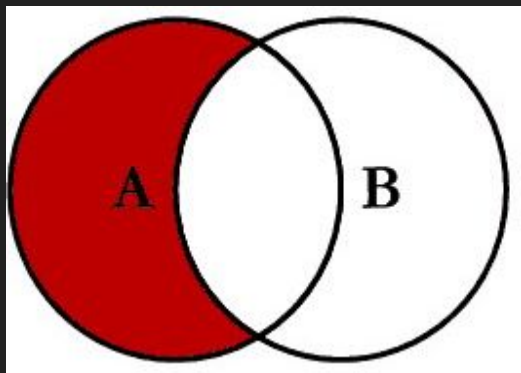
```
Script000.sql > {} SELECT
1  SELECT P.nmPessoa "Pessoa",
2         D.nmDependente "Dependente"
3  FROM PESSOA P
4  JOIN DEPENDENTE D USING (cdPessoa)
5
```



# JOIN - (LEFT JOIN)

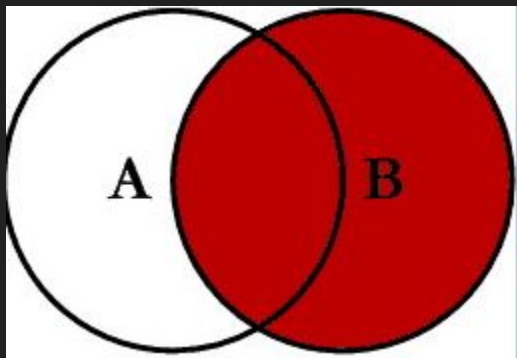


```
Script000.sql > {} SELECT
1  SELECT P.nmPessoa "Pessoa",
2  D.nmDependente "Dependente"
3  FROM PESSOA P
4  LEFT JOIN DEPENDENTE D ON (D.cdPessoa = P.cdPessoa)
5
```

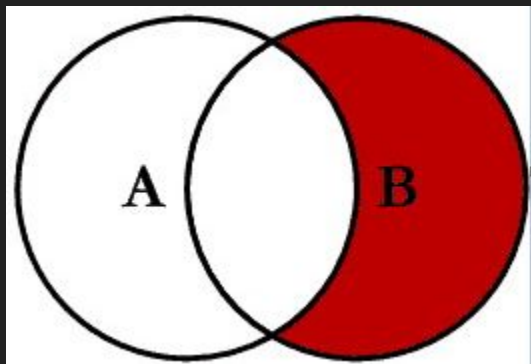


```
Script000.sql > {} SELECT
1  SELECT P.nmPessoa "Pessoa",
2  D.nmDependente "Dependente"
3  FROM PESSOA P
4  LEFT JOIN DEPENDENTE D ON (D.cdPessoa = P.cdPessoa)
5  WHERE D.cdPessoa IS NULL
6
```

# JOIN (RIGHT JOIN)

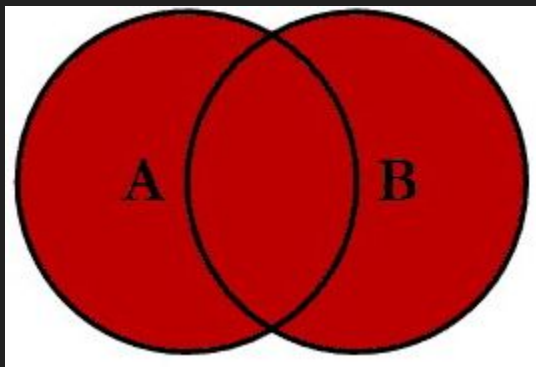


```
Script000.sql > {} SELECT
1  SELECT d.department_id, e.last_name
2  FROM departamento d
3  RIGHT JOIN funcionario f ON f.cdDepartamento= d.idDepartamento
4
```

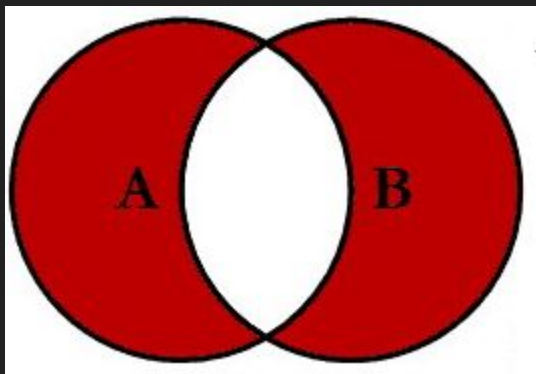


```
Script000.sql > {} SELECT
1  SELECT d.department_id, e.last_name
2  FROM departamento d
3  RIGHT JOIN funcionario f ON f.cdDepartamento= d.idDepartamento
4  WHERE d.idDepartamento IS NULL
5
```

# JOIN (FULL JOIN)

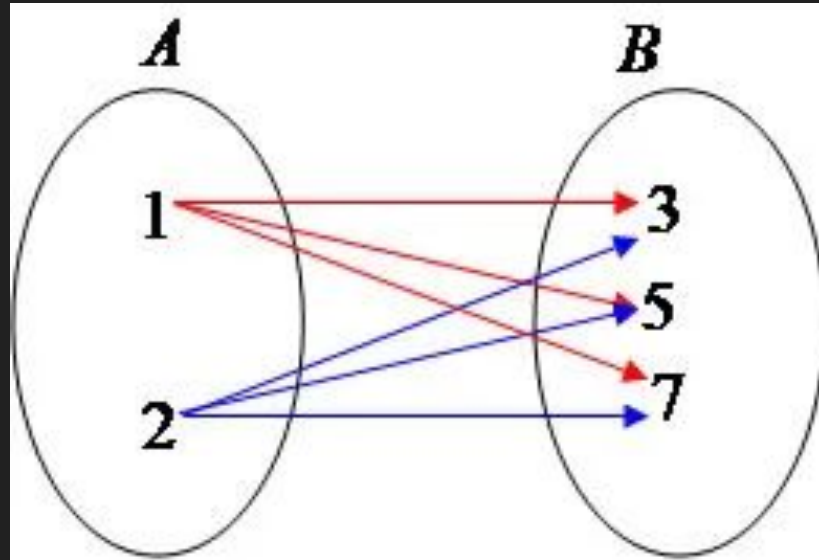


```
Script000.sql > {} SELECT
1  SELECT COALESCE(v.cdPessoa, f.cdPessoa) cdPessoa,
2          SUM(COALESCE(v.vlrVendas, 0)) vlrVendas
3          SUM(COALESCE(f.vlrTotal, 0)) vlrTotal
4  FROM VENDAS v
5  FULL JOIN FINANCEIRO f ON f.cdPessoa = v.cdPessoa
6  GROUP BY COALESCE(v.cdPessoa, f.cdPessoa)
7
```

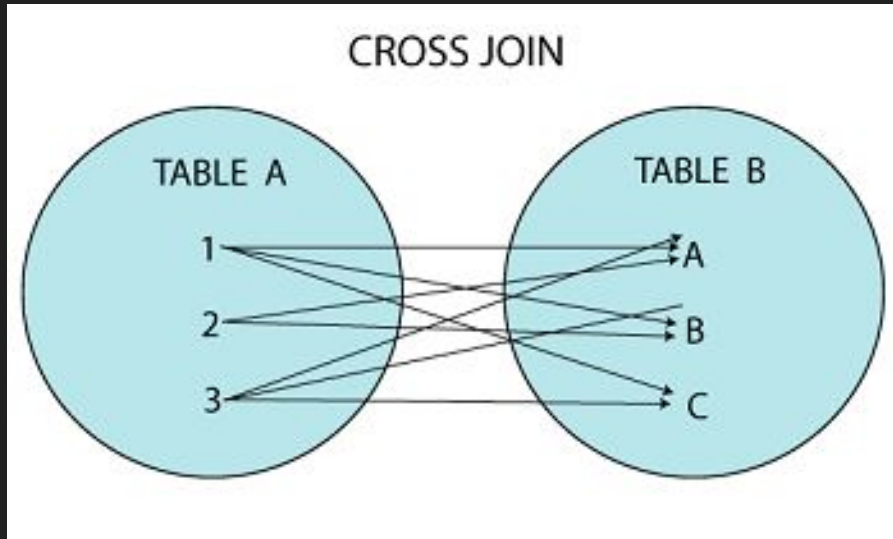


```
Script000.sql > {} SELECT
1  SELECT COALESCE(v.cdPessoa, f.cdPessoa) cdPessoa
2  FROM VENDAS v
3  FULL JOIN FINANCEIRO f ON f.cdPessoa = v.cdPessoa
4  WHERE COALESCE(v.cdPessoa, f.cdPessoa) IS NULL
5
```

# JOIN (PRODUTO CARTESIANO)



# JOIN (CROSS JOIN)



Meals		MealName	DrinkName
Omlet		Omlet	Orange Juice
Fried Egg		Fried Egg	Orange Juice
Sausage		Sausage	Orange Juice
		Omlet	Tea
		Fried Egg	Tea
		Sausage	Tea
		Omlet	Coffee
		Fried Egg	Coffee
		Sausage	Coffee

CROSS JOIN

Drinks	
Orange Juice	
Tea	
Coffee	

# JOIN (CROSS JOIN)

Script000.sql > {} SELECT

```
1  with
2  v dia as (
3      select distinct
4          last_day(dia('01/01/2020') + rownum) posicao
5          from all_objects
6          where dia('01/01/2020') + rownum < hoje
7          order by posicao
8  )
9
10 v select to_char(dia.posicao, 'MM/YYYY') ano,
11         count(1) total
12         from dia
13         cross join cliente c
14         where associado(c.idCliente, dia.posicao) = 'S'
15         group by to_char(dia.posicao, 'MM/YYYY')
16
17     Custo: 50538541
18
```

Script000.sql > {} SELECT

```
1  with
2  dia as (
3      select distinct
4          last_day(dia('01/01/2020') + rownum) posicao
5          from all_objects
6          where dia('01/01/2020') + rownum < hoje
7          order by posicao
8  )
9
10 select to_char(dia.posicao, 'MM/YYYY') ano,
11        count(1) total
12        from dia, cliente c
13        where associado(c.idCliente, dia.posicao) = 'S'
14        group by to_char(dia.posicao, 'MM/YYYY')
15
16     Custo: 50534904
17
```

0.0071964879239391%



# JOIN (NATURAL JOIN)

Script002.sql > {} SELECT

```
1  with
2  clientes as (
3      select 123 idPessoa, 'Jaisson' nmCliente, to_date('20/03/1990', 'DD/MM/YYYY') dtnascimento from dual
4      union all
5      select 321, 'Frederico', to_date('15/08/1995', 'DD/MM/YYYY') dtnascimento from dual
6  ),
7
8  fornecedor as (
9      select 123 idPessoa, 'jaisson@empresa.com.br' dsEmail from dual
10     union all
11     select 321 idFornecedor, 'frederico@empresa.com.br' dsEmail from dual
12 )
13
14 select *
15     from clientes
16 natural join fornecedor
```

Resultado da Consulta x				
Todas as Linhas Extraídas: 2 em 0,015 segundos				
	IDPESSOA	NMCLIENTE	DTNASCIMENTO	DSEMAIL
1	123	Jaisson	20/03/90	jaisson@empresa.com.br
2	321	Frederico	15/08/95	frederico@empresa.com.br

# JOIN (CROSS versus NATURAL)

Planilha Query Builder

```
1 with
2 clientes as (
3   select 123 idPessoa, 'Jaisson' nmCliente, to_date('20/03/1990', 'DD/MM/YYYY') dtnascimento from dual
4   union all
5   select 321, 'Frederico', to_date('15/08/1995', 'DD/MM/YYYY') dtnascimento from dual
6 ),
7
8 fornecedor as (
9   select 123 idPessoa, 'jaisson@empresa.com.br' dsEmail from dual
10  union all
11  select 321 idFornecedor, 'frederico@empresa.com.br' dsEmail from dual
12 )
13
14 select *
15 from clientes
16 natural join fornecedor;
```

Resultado da Consulta x

Todas as Linhas Extraídas: 2 em 0,015 segundos

	IDPESSOA	NMCLIENTE	DTNASCIMENTO	DSEMAIL
1	123 Jaisson	20/03/90	jaisson@empresa.com.br	
2	321 Frederico	15/08/95	frederico@empresa.com.br	

Planilha Query Builder

```
2 clientes as (
3   select 123 idPessoa, 'Jaisson' nmCliente, to_date('20/03/1990', 'DD/MM/YYYY') dtnascimento from dual
4   union all
5   select 321, 'Frederico', to_date('15/08/1995', 'DD/MM/YYYY') dtnascimento from dual
6 ),
7
8 fornecedor as (
9   select 123 idPessoa, 'jaisson@empresa.com.br' dsEmail from dual
10  union all
11  select 321 idFornecedor, 'frederico@empresa.com.br' dsEmail from dual
12 )
13
14 select *
15 from clientes
16 cross join fornecedor;
```

Resultado da Consulta x

Todas as Linhas Extraídas: 4 em 0,016 segundos

	IDPESSOA	NMCLIENTE	DTNASCIMENTO	IDPESSOA_1	DSEMAIL
1	123 Jaisson	20/03/90		123 jaisson@empresa.com.br	
2	123 Jaisson	20/03/90		321 frederico@empresa.com.br	
3	321 Frederico	15/08/95		123 jaisson@empresa.com.br	
4	321 Frederico	15/08/95		321 frederico@empresa.com.br	



# Subconsulta

Consulta dentro de uma consulta

- Subconsultas de uma única linha
- Subconsultas de várias linhas
- Subconsultas de várias colunas
- Subconsultas correlacionadas
- Subconsultas aninhadas

## Subconsulta de uma única linha

Script003.sql > {} SELECT

```
1  select p.dtemissao,  
2      c.nmCliente "Cliente",  
3      (select sum(pi.quantidade * pi.vlrUnitario) total  
4          from pedido_item pi  
5          where pi.cdPedido = p.idPedido) total_itens  
6  from pedido p  
7  join cliente c on (c.idCliente = p.cdCliente)  
8  where (select count(1)  
9          from pedido_item pi  
10         where pi.cdPedido = p.idPedido) > 5  
11
```

# Subconsulta de de várias linhas

IN

Lista de opções fixas.

ANY/SOME <, >, =, <=, >=

Compara a consulta "principal" com a subconsulta, retorna se uma das linhas do conjunto satisfaz a condição.

ALL <, >, =, <=, >=

Compara a consulta "principal" com a subconsulta, retorna se todas as linhas do conjunto satisfazem a condição.

EXISTS

Compara a consulta "principal" com a subconsulta, retornar somente se existir.

## Subconsultas de várias linhas

```
12 select p.dtemissao,  
13         c.nmCliente "Cliente",  
14         sum((pi.quantidade * pi.vlrUnitario)) total_itens  
15 from pedido p  
16 join cliente c on (c.idCliente = p.cdCliente)  
17 join pedido_item pi on (pi.cdPedido = p.idPedido)  
18 where p.dtemissao >= sysdate -7  
19        and c.cdMunicipio in (select m.idMunicipio  
20                               from municipio m  
21                               where m.dsUF = 'RS')  
22 group by p.dtemissao, c.nmCliente
```

23

## Subconsultas de várias linhas

```
25 select ename, sal, job, deptno  
26     from emp  
27    where sal > ANY (select distinct  
28                      |   |   |  
29                      |   |   | sal  
30                      |   |   | from emp  
                        |   |   | where deptno = 30);
```

31

## Subconsultas de várias linhas

```
32 select ename, sal, job, deptno
33     from emp
34     where sal > ALL (select distinct
35                       sal
36                       from emp
37                       where deptno = 30);
```

## Subconsultas de várias linhas

```
39  select m.empno,m.ename,m.job,m.deptno
40      from emp m
41  where exists (select e.empno
42                from emp e
43                where e.mgr=m.empno)
```

## Subconsultas de várias colunas



# Subconsultas correlacionadas

Possuem uma relação com a consulta principal

```
12 select p.dtemissao,  
13        c.nmCliente "Cliente",  
14        sum((pi.quantidade * pi.vlrUnitario)) total_itens  
15 from pedido p  
16 join cliente c on (c.idCliente = p.cdCliente)  
17 join pedido_item pi on (pi.cdPedido = p.idPedido)  
18 where p.dtemissao >= sysdate -7  
19        and c.cdMunicipio in (select m.idMunicipio  
20                               from municipio m  
21                               where m.dsUF = 'RS')  
22 group by p.dtemissao, c.nmCliente  
23
```

```
39 select m.empno,m.ename,m.job,m.deptno  
40        from emp m  
41 where exists (select e.empno  
42               from emp e  
43               where e.mgr=m.empno)
```

# Subconsultas aninhadas

```
53 select roma.numerocm, extract(year from roma.dtemissao) ano,
54      sum((((case
55              when exists (select *
56                           from romatransf tsi
57                           where (tsi.estab      = roma.estab)
58                                and (tsi.romaneio  = roma.romaneio)
59                                and (tsi.entradasaída = roma.entradasaída)
60                                and (tsi.numerocm   = roma.numerocm))
61              then (select coalesce(sum(tsi.quantidade) ,0)
62                       from romatransf tsi
63                       where (tsi.estab      = roma.estab)
64                            and (tsi.romaneio  = roma.romaneio)
65                            and (tsi.entradasaída = roma.entradasaída)
66                            and (tsi.numerocm   = roma.numerocm)
67                            and not exists (select *
68                                           from u_roma_invalido
69                                           where  estab = tsi.estab
70                                                and  entradaída = tsi.entradasaída
71                                                and  numerocm = tsi.numerocmtransf))
72              else roma.pesoliquido
73              end)))) pesoliquido
74 from roma
75 join itemagro on (itemagro.item = roma.item)
76 join pessoas on (pessoas.numerocm = roma.numerocm)
77 where roma.romaneioconfig in (select romaneioconfig
78                               from romacfg
79                               where movprod = 'S')
80 and (itemagro.grupo in (1088, 1090))
81 and roma.dtemissao >= to_date('01/01/2023', 'DD/MM/YYYY')
82 group by roma.numerocm, extract(year from roma.dtemissao);
```

# SUBSELECT

```
86  SELECT FILIAL.REDUZIDO ORIGEM,
87         NFCAB.DTEMISSAO, NFCAB.NOTA,
88         DESTINO.REDUZIDO DESTINO,
89         ITEMAGRO.ITEM, ITEMAGRO.DESCRICAO,
90         NFITEM.QUANTIDADE QTDE_SAIDA,
91         NFITEM.VALORTOTAL VLR_SAIDA,
92         NFPE.DTENTRADA , NFPE.ROMANEIO ROMANEIO_ENTRADA,
93         NFPE.QUANTIDADE QTDE_ENTRADA, NFPE.VALORTOTAL VLR_ENTRADA
94  FROM NFCAB
95  JOIN FILIAL ON (FILIAL.ESTAB = NFCAB.ESTAB)
96  JOIN FILIAL DESTINO ON (DESTINO.ESTAB = NFCAB.NUMEROCM)
97  JOIN NFITEM ON (NFITEM.ESTAB = NFCAB.ESTAB) AND (NFITEM.SEQNOTA = NFCAB.SEQNOTA)
98  JOIN ITEMAGRO ON (ITEMAGRO.ITEM = NFITEM.ITEM)
99  LEFT JOIN (
100         SELECT N.ESTAB, P.CHAVEACESSONFP CHAVEACESSONFE,
101                N.DTEMISSAO DTENTRADA,
102                NFI.ROMANEIO, NFI.QUANTIDADE, NFI.VALORTOTAL
103         FROM NFCAB N
104         JOIN NFITEM NFI ON (NFI.ESTAB = N.ESTAB) AND (NFI.SEQNOTA = N.SEQNOTA)
105         JOIN NFCABPRODUTOR P ON (P.ESTAB = N.ESTAB) AND (P.SEQNOTA = N.SEQNOTA)
106         WHERE N.NOTACONF = 1101
107     ) NFPE ON (NFPE.CHAVEACESSONFE = NFCAB.CHAVEACESSONFE)
108  WHERE NFCAB.NOTACONF = 1100
109         AND NFCAB.STATUS = 'N'
110         AND NFCAB.DTEMISSAO BETWEEN :DTINICIO AND :DTFIM
111  ORDER BY NFCAB.DTEMISSAO DESC
```

# TABELA DERIVADA

```
114 SELECT ....
115     NFPRODUTOR.NFPRODUTOR || '/' || NFPRODUTOR.SERIENFPRODUTOR NFP,
116     SUM(NFITEM.QUANTIDADE) QUANTIDADE,
117     COALESCE(IMPOSTOS.VLRIMPOSTO, 0) IMPOSTOS,
118     COALESCE((SELECT ROMADESC.REFTABELA
119              FROM ROMADESC
120              WHERE (ROMADESC.ESTAB = NFCAB.ESTAB)
121                   AND (ROMADESC.ROMANEIO = ROMANEIO.ROMANEIO)
122                   AND (ROMADESC.ENTRADASAIDA = 'E')
123                   AND (ROMADESC = 2)
124                   AND (ROMADESC.NUMEROCM = NFCAB.NUMEROCM)), 0) IMPUREZA,
125     ....
126 FROM   NFCAB
127     ....
128 LEFT JOIN LATERAL (SELECT SUM(I.VALORIMPOSTO) VLRIMPOSTO
129                    FROM   NFCABIMPOSTO I
130                    JOIN   IMPOSTO_U U ON (U.IMPOSTO = I.IMPOSTO)
131                    WHERE  I.ESTAB = NFCAB.ESTAB
132                           AND I.SEQNOTA = NFCAB.SEQNOTA) IMPOSTOS ON (1=1)
133 LEFT JOIN LATERAL (SELECT NFPRODUTOR, SERIENFPRODUTOR
134                    FROM   NFCABPRODUTOR NFP
135                    WHERE  NFP.ESTAB = NFCAB.ESTAB
136                           AND NFP.SEQNOTA = NFCAB.SEQNOTA
137                           AND ROWID = (SELECT MAX(ROWID)
138                                       FROM   NFCABPRODUTOR NFPI
139                                       WHERE  NFPI.ESTAB = NFP.ESTAB
140                                       AND NFPI.SEQNOTA = NFP.SEQNOTA)) NFPRODUTOR ON (1=1)
141 LEFT JOIN LATERAL (SELECT MAX(ROMANEIO) ROMANEIO
142                    FROM   NFCABROMA
143                    WHERE  NFCABROMA.ESTAB = NFCAB.ESTAB
144                           AND NFCABROMA.SEQNOTA = NFCAB.SEQNOTA
145                           AND NFCABROMA.NUMEROCM = NFCAB.NUMEROCM) ROMANEIO ON (1=1)
146 WHERE  ....
147 GROUP BY ....
148 ORDER BY ....
149
```

# CTE - COMMON TABLE EXPRESSION



# CTE

Script004.sql U ●

Script004.sql

```
1  WITH
2  nmTblTMP ([colunas]) AS (
3  |   consulta sql
4  )
5  ...
6
7  ou
8
9  WITH
10 nmTblTMP AS (
11 |   consulta sql
12 )
13 ...
14
15
```

# CTE

```
14
15 with
16 cliente (idpessoa, nmpessoa) as (
17 | select 123 idPessoa, 'Jaisson Duarte' from dual
18 | )
19
20 select * from cliente;
21
22 -----
23 -----
24 -----
25
26 with
27 cliente as (
28 | select 123 idPessoa, 'Jaisson Duarte' from dual
29 | )
30
31 select * from cliente;
32
```

# FUNÇÕES SIMPLES

<https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/index.html#Oracle%C2%AE-Database>

Funções Simples

Funções de Agregação



# FUNÇÕES ANALÍTICAS

# FUNÇÕES ANALÍTICAS

≡ exemplo3.sql ●

≡ exemplo4.sql ✕

≡ exemplo4.sql

```
1  SELECT NOTA.NUMERO, NOTA.VALOR,  
2         SUM(NOTA.VALOR) OVER (PARTITION BY NULL  
3                                ORDER BY NOTA.NUMERO  
4                                ROWS BETWEEN UNBOUNDED PRECEDING AND 0 PRECEDING) AS SALDOATUAL  
5  FROM  NOTA_FISCAL NOTA  
6
```

# FUNÇÕES ANALÍTICAS

```
13
14 with
15 fonte as (
16     select cidade.nmcidade cidade,
17            sum(nota.valor) valor
18     from nota
19     join pessoa on (pessoa.idpessoa = nota.cd pessoa)
20     join cidade on (cidade.idcidade = pessoa.cdcidade)
21     where nota.dtemissao >= to_date('01/01/2023', 'DD/MM/YYYY')
22           and cidade.nmcidade in ('CRUZ ALTA', 'IBIRUBA', 'QUINZE DE NOVEMBRO')
23     group by cidade.nome
24 )
25
26 select fonte.cidade, fonte.valor,
27        round(RATIO_TO_REPORT(fonte.valor) OVER () * 100, 2) representa
28     from fonte
29     order by 3;
30
```

# FUNÇÕES ANALÍTICAS

<https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/Analytic-Functions.html#GUID-527832F7-63C0-4445-8C16-307FA5084056>

SUPER AGRUPAMENTO

# SUPER AGRUPAMENTOS

≡ exemplo3.sql ●

≡ exemplo4.sql

≡ exemplo5.sql ✕

≡ exemplo5.sql > {} SELECT

```
1  select extract(month from nota.dtemissao) mes,  
2         cidade.nmcidade cidade,  
3         sum(nfcab.valor) total  
4  from nota  
5  join pessoa on (pessoa.idpessoa = nota.cd pessoa)  
6  join cidade on (cidade.idcidade = pessoa.cdcidade)  
7  where nota.dtemissao >= to_date('01/01/2023', 'DD/MM/YYYY')  
8         and cidade.nmcidade in ('CRUZ ALTA', 'IBIRUBA', 'QUINZE DE NOVEMBRO')  
9  group by extract(month from nota.dtemissao),  
10         cidade.nome  
11  order by mes, cidade  
12
```

# SUPER AGRUPAMENTOS

≡ exemplo3.sql ●

≡ exemplo4.sql

≡ exemplo5.sql ●

≡ exemplo5.sql > {} SELECT

```
1  select extract(month from nota.dtemissao) mes,
2      cidade.nmcidade cidade,
3      sum(nfcab.valor) total
4  from nota
5  join pessoa on (pessoa.idpessoa = nota.cd pessoa)
6  join cidade on (cidade.idcidade = pessoa.cdcidade)
7  where nota.dtemissao >= to_date('01/01/2023', 'DD/MM/YYYY')
8      and cidade.nmcidade in ('CRUZ ALTA', 'IBIRUBA', 'QUINZE DE NOVENBRO')
9  GROUP BY GROUPING SETS ((extract(month from nota.dtemissao), cidade.nmcidade),
10      (extract(month from nota.dtemissao)))
11  order by mes, cidade
12
```

# SUPER AGRUPAMENTOS

<https://oracle-base.com/articles/misc/rollup-cube-grouping-functions-and-grouping-sets#rollup>



# CROSS TABLE

# CROSS TABLE

Input			Output						
Key	Attribute	Value	Color	Diameter	Height	Length	Weight	Width	
Ball	Color	Red	Red	4	-	-	-	-	-
Ball	Diameter	4	-	4	-	-	-	-	-
Box	Height	2	-	-	2	6	1	-	4
Box	Length	6	-	-	-	-	-	-	-
Box	Weight	1	-	-	-	-	-	-	-
Box	Width	4	-	-	-	-	10	-	-
Dog	Color	Black	Black	-	-	-	-	-	-
Dog	Weight	10	-	-	-	-	-	-	-
Rod	Diameter	1	-	-	-	-	-	-	-
Rod	Length	10	-	-	-	-	-	-	-

Output			Input						
Product	Month	Sales	Jan 2014	Feb 2014	Mar 2014	Apr 2014	May 2014	Jun 2014	
A	Jan 2014	100	100	98	103	83	103	82	
A	Feb 2014	98	-	98	-	-	-	-	
A	Mar 2014	103	-	-	103	-	-	-	
A	Apr 2014	83	-	-	-	83	-	-	
A	May 2014	103	-	-	-	-	103	-	
A	Jun 2014	82	-	-	-	-	-	82	
B	Jan 2014	284	284	279	297	305	294	292	
B	Feb 2014	279	-	279	-	-	-	-	
B	Mar 2014	297	-	-	297	-	-	-	
B	Apr 2014	305	-	-	-	305	-	-	

# CROSS TABLE

Planilha Query Builder

```
1 select extract(month from nfcab.dtemissao) mes,
2 cidade.nome_cidade,
3 sum(nfcab.valor) valor
4 from nfcab
5 join contamov on (contamov.numerocm = nfcab.numerocm)
6 join cidade on (cidade.cidade = contamov.cidade)
7 where nfcab.dtemissao >= to_date('01/06/2023', 'DD/MM/YYYY')
8 and nfcab.notaconf = 1
9 and cidade.cidade in ('02119', '01474', '04079')
10 group by extract(month from nfcab.dtemissao), cidade.nome
11 order by 1, 2
12
```

Saída do Script x Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 15 em 14,835 segundos

	MES	CIDADE	VALOR
1	6	CRUZ ALTA	11998569,44
2	6	IBIRUBÁ	2087571,35
3	6	QUINZE DE NOVENBRO	915881,59
4	7	CRUZ ALTA	10485799,83
5	7	IBIRUBÁ	1873116,53
6	7	QUINZE DE NOVENBRO	772981,22
7	8	CRUZ ALTA	10840167,87
8	8	IBIRUBÁ	2549649,07
9	8	QUINZE DE NOVENBRO	899084,45
10	9	CRUZ ALTA	11184496,42
11	9	IBIRUBÁ	2246598,18
12	9	QUINZE DE NOVENBRO	658928,11
13	10	CRUZ ALTA	172581,76
14	10	IBIRUBÁ	303787,38
15	10	QUINZE DE NOVENBRO	183010,06

# CROSS TABLE - PIVOT

Planilha Query Builder

```
2 with
3 fonte as (
4 select extract(month from nfcab.dtemissao) mes,
5         cidade.nome cidade,
6         sum(nfcab.valor) valor
7   from nfcab
8   join contamov on (contamov.numeroocm = nfcab.numeroocm)
9   join cidade on (cidade.cidade = contamov.cidade)
10  where nfcab.dtemissao >= to_date('01/06/2023', 'DD/MM/YYYY')
11        and nfcab.notaconf = 1
12        and cidade.cidade in ('02119', '01474', '04079')
13  group by extract(month from nfcab.dtemissao), cidade.nome
14 )
15
16 select *
17   from fonte
18  pivot (sum(valor) for mes in (6 as "Junho", 7 "Julho", 8 "Agosto", 9 "Setembro", 10 "Outubro"))
19
```

Saída do Script x Resultado da Consulta x

Todas as Linhas Extraídas: 3 em 24,181 segundos

	CIDADE	Junho	Julho	Agosto	Setembro	Outubro
1	IBIRUBÁ	2087571,35	1873116,53	2549649,07	2246598,18	303677,19
2	QUINZE DE NOVENBRO	915881,59	772981,22	899084,45	658928,11	181353,35
3	CRUZ ALTA	11998569,44	10485799,83	10840167,87	11184496,42	172581,76

# CROSS TABLE - UNPIVOT

The screenshot shows a SQL Query Builder window with a query editor and a results pane. The query editor contains the following SQL code:

```
1 with
2 fonte as (
3     select 'a' v1, 'e' v2, 'i' v3, 'o' v4, 'u' v5
4     from dual
5 )
6
7 select novo_coluna
8     from fonte
9 unpivot (novo_coluna for value_type in (v1,v2,v3,v4,v5))
10
```

The results pane shows the output of the query, which is a table with one column named 'NOVO\_COLUNA' and five rows of data:

	NOVO_COLUNA
1	a
2	e
3	i
4	o
5	u

# CROSS TABLE - PIVOT



# Agenda

- Introdução
- Revisão SQL
- SGBDs
- Ranking Popularidade
- Oracle
- Mercado de Trabalho
- Estrutura de uma Consulta
- Consultas
- Live SQL

# Live SQL

<https://livesql.oracle.com/>

<https://github.com/jaisson/sql>



Obrigado