

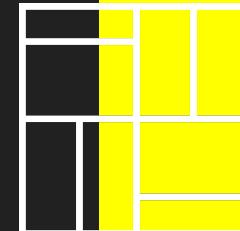
E | D | I | T.

DATABASES FUNDAMENTALS

TUTORA

Natacha Cabral

Senior Data Analyst
@Erakulis



E | D | I | T.

DATABASES FUNDAMENTALS

Apresentações Alunos

Nome

Profissão

Descrição

Motivação

Expectativas



O que não esperar

Depois deste módulo, não te vais tornar automaticamente num(a) Data Engineer ou Database Admin.

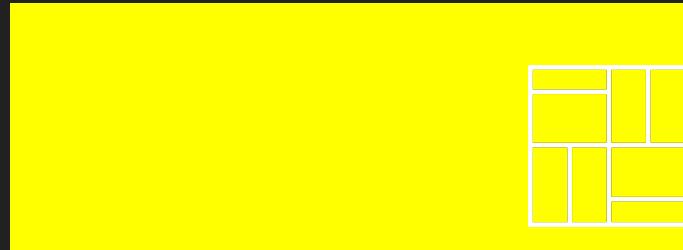
Não te vais tornar automaticamente num(a) guru especialista de SQL ou de bases de dados.



Com a experiência e o tempo adequado para consolidares os conteúdos abordados aqui, terás tudo para poderes chegar lá!

O que esperar

- ✓ Vais aprender como interpretar bases de dados já existentes e como obter a informação que procuras.
- ✓ Vais aprender como as bases de dados funcionam e como podem assumir diferentes propósitos numa organização.
- ✓ Vais aprender a entender a pipeline de dados entre diferentes bases de dados.
- ✓ Vais compreender a arquitetura tradicional utilizada para data storage e as suas limitações.



Critérios de Avaliação

Assiduidade e Pontualidade 15%

Motivação e participação 20%

Domínio dos conteúdos 15%

Aplicação Técnica 20%

Trabalho Final 30%

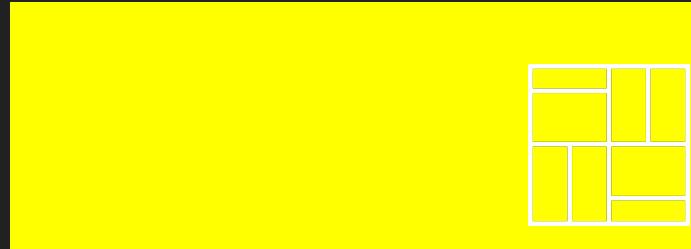
Assiduidade e Pontualidade – Presença nas aulas e cumprimento do horário de formação

Motivação e Participação – Intervenção nos debates de forma pertinente, participação nas aulas

Domínio dos Conteúdos – Revela dominar conteúdos abordados nas aulas quando assim é solicitado

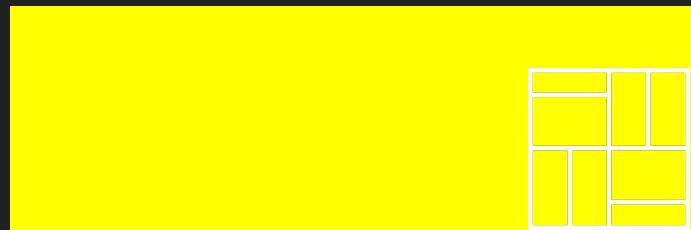
Aplicação Técnica – Demonstração dos conhecimentos adquiridos

Trabalho Final - Exercício final individual



1. Bases de dados - Conceitos Fundamentais

- 1.1.O que são dados?
- 1.2.O que é uma base de dados?
- 1.3.O que é SQL?
- 1.4.O que é uma query?
- 1.5.SQL vs Mysql, Sql Server; Oracle e PostgreSQL
- 1.6.Porque aprender SQL
- 1.7.Base de dados relacionais e não relacionais



E | D | I | T.

1. Base de Dados - Conceitos Fundamentais



Dado



Informação

Tabela

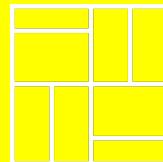


Várias
Informações
(organizadas)

Base de dados



Várias tabelas (armazenadas
num sistema computacional)



Base de dados

Folha de Excel - sheet 1- Dados dos clientes

Nome	Apelido	Email	Idade	Morada	Localidade	Código_Postal
João	Sousa	joao22@gmail.com	45	Av.da Liberdade, 55	Lisboa	1000-100
Maria	Lima	mary@lima.com	26	Rua Serpa Pinto, 45	Porto	4050-050
Francisco	Xavier	patodonald@sapo.pt	33	Rua Cidade da Horta,10	Ponta Delgada	9580-175

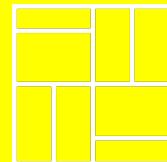
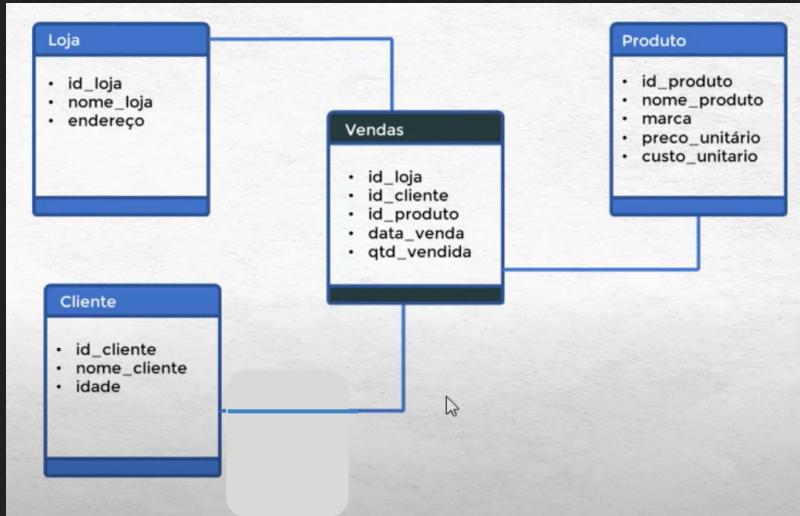


DATABASES FUNDAMENTALS

Base de dados - Conceitos Fundamentais

E | D | I | T.

Desenho esquemático de um banco de dados:



DATABASES FUNDAMENTALS

Base de dados - Conceitos Fundamentais

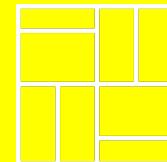
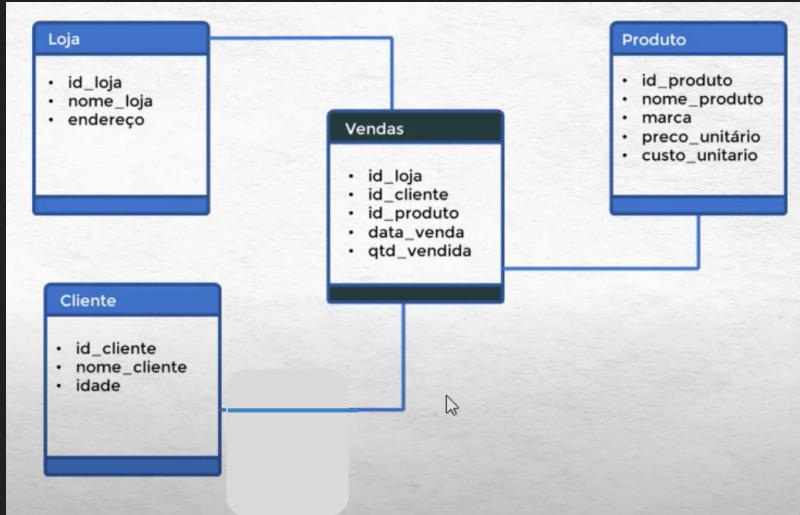
E | D | I | T.

Desenho esquemático de um banco de dados:

Diversas tabelas, com diferentes informações sobre o negócio, e que possuem algum tipo de relação.

A esta base de dados damos o nome de
RELACIONAL

Bancos de dados relacionais serão o foco do curso, até por serem o tipo de bancos de dados mais comumente encontrados no mercado



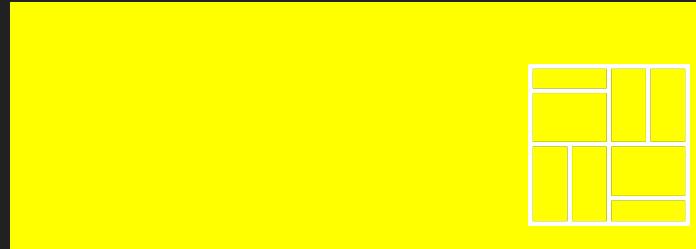
É aqui que entra o DBMS (Database Management Systems)

Um DBMS (Database Management System) é o sistema de software que permite aos utilizadores **criar, manter e consultar** a informação guardada numa base de dados.



Exemplos de sistemas de banco dados: Oracle, SQL Server, DB2, PostgreSQL, MySQL, o próprio Access, entre outros.

Também utilizamos no mercado DBMS na nuvem, como por exemplo o bigquery e o snowflake.

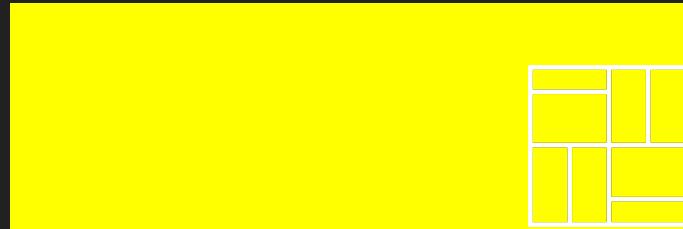


É aqui que entra o
DBMS (Database Management Systems) e o **SQL (Structured Query Language)**



SQL: Linguagem de programação que permite comunicar com bases de dados. Permite armazenar, consultar, adicionar e excluir informações de um banco de dados
SQL em português significa **Linguagem de Consulta Estruturada**.

Query: É uma consulta ao banco de dados

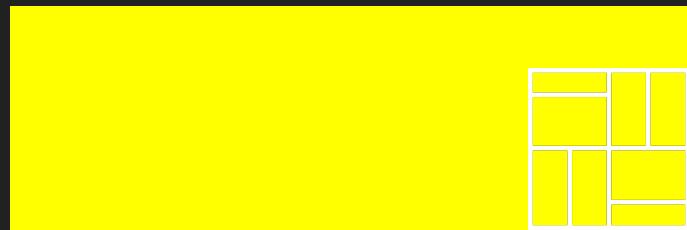


SQL vs Mysql, Sql Server; Oracle e PostgreSQL

SQL: Linguagem de programação, linguagem de consulta aos dados:

Constituída por três subgrupos:

- DML: Data Manipulation Language - Manipulação de dados- seleção, inserção e eliminação (select or insert)
- DDL: Data Definition Language - Manipulação da estrutura de uma base de dados (create, use)
- DCL: Data Control Language - Manipulação de utilizadores e controle de acessos (create login, use db1, create user etc)



SQL vs Mysql, Sql Server; Oracle e PostgreSQL

Mysql, SQL Server, Oracle e PostgreSQL: São programas para gerenciar as bases dados

Não interessa o programa que vou usar, a linguagem é sempre a mesma, SQL – Pode haver algumas particularidades de língua, mas SQL é o que se tem que aprender.

A escolha dos programas depende da escolha das empresas.

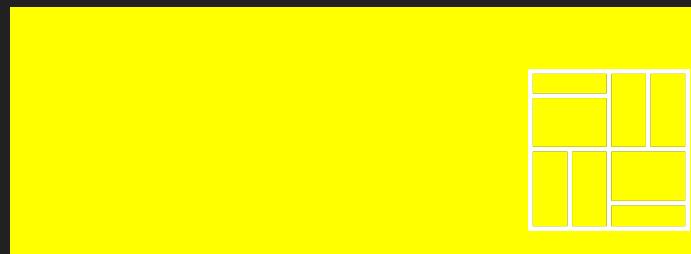
Porque aprender SQL?

Porque é uma linguagem universal para manipular bancos de dados e garante a versatilidade de conhecimento.

Existe cada vez mais um alto volume de dados – é preciso saber trabalhar estes dados

Segurança da informação – Um sistema de bancos de dados garante muita segurança.

Linguagem atemporal – sql surgiu na década de 70 e ainda hoje é utilizado.



Base de Dados relacionais e Não relacionais

Base de dados relacionais

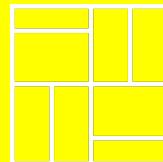
- Dados são armazenados em tabelas
- As tabelas podem ser relacionadas entre si
- Apresentam uma estrutura pré-definida (schema)
- Requerem hardware especializado

Exemplos: MySql, Oracle, Postgres

Base de dados Não relacionais

- Sem linguagem de consulta padrão
- Não relacionais
- Podem ser baseadas em documentos - dados não estruturados ou semiestruturados
- Requerem apenas commodity hardware (componentes previamente disponíveis e baratas)

Exemplos: Mongo DB, BigTable, Redis etc



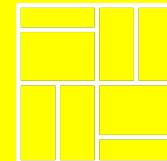
Principais fornecedores

Rank	Mar 2024	Feb 2024	Mar 2023	DBMS	Database Model	Score		
						Mar 2024	Feb 2024	Mar 2023
1.	1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1221.06	-20.39	-40.23
2.	2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1101.50	-5.17	-81.29
3.	3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	845.81	-7.76	-76.20
4.	4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	634.91	+5.50	+21.08
5.	5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	424.53	+4.18	-34.25
6.	6.	6.	6.	Redis +	Key-value, Multi-model ⓘ	157.00	-3.71	-15.45
7.	7.	↑ 8.	Elasticsearch		Search engine, Multi-model ⓘ	134.79	-0.95	-4.28
8.	8.	↓ 7.	IBM Db2		Relational, Multi-model ⓘ	127.75	-4.47	-15.17
9.	9.	↑ 11.	Snowflake +		Relational	125.38	-2.07	+10.98
10.	10.	↓ 9.	SQLite +		Relational	118.16	+0.88	-15.66
11.	11.	↓ 10.	Microsoft Access		Relational	107.93	-5.24	-24.13
12.	12.	12.	Cassandra +		Wide column, Multi-model ⓘ	104.59	-4.69	-9.20
13.	13.	13.	MariaDB +		Relational, Multi-model ⓘ	95.03	-2.20	-1.81
14.	14.	14.	Splunk		Search engine	89.68	-1.97	+1.71
15.	↑ 16.	↑ 16.	Microsoft Azure SQL Database		Relational, Multi-model ⓘ	78.51	-1.06	+1.06
16.	↓ 15.	↓ 15.	Amazon DynamoDB +		Multi-model ⓘ	77.72	-5.18	-3.05
17.	17.	↑ 19.	Databricks +		Multi-model ⓘ	74.34	-2.57	+13.48
18.	18.	↓ 17.	Hive		Relational	64.82	-0.99	-6.09
19.	19.	↑ 21.	Google BigQuery +		Relational	62.67	-0.96	+9.23
20.	20.	↓ 18.	Teradata		Relational, Multi-model ⓘ	48.95	-2.29	-14.79
21.	21.	↑ 22.	FileMaker		Relational	48.81	-1.67	-2.33
22.	↑ 23.	↑ 23.	SAP HANA +		Relational, Multi-model ⓘ	45.49	+0.27	-5.36

Source: <https://db-engines.com/en/ranking>

Legenda

- (Principalmente) SQL
- (Principalmente) NoSQL



E | D | I | T.

2. Modelos de Dados

- 2.1. Tabelas
- 2.2. Views
- 2.3. Materialized View
- 2.4. Chave primária e Chave estrangeira ou de referência

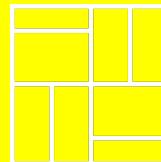


Modelos de Dados

Os objetos mais elementares que constituem uma BD relacional são:

1. Tabelas
2. Views
3. Materialized Views

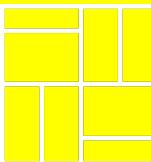
ID	Title	ISBN	Author	Publishing...
1	1984	2343454895456	George Orwell	04/12/1979
2	Anna Karenina	1234548485843	Leo Tolstoy	07/11/1998
4	The Adventures of I	3450345345443	Mark Twain	08/11/1999
5	Ulysses	9944933003232	James Joyce	06/05/2010
8	War and Peace	0944344903312	Leo Tolstoy	08/11/2001
11	The Brothers Karan	9003940397271	Doso	04/07/2012
12	On the Road	0459450444310	Jack Kerouac	30/12/2005
15	The Metamorphosi	2003948930545	Franz Kafka	09/03/1976
16	The Illiad	9449039333923	Homer	05/07/1998
17	The Odyssey	8409404850139	Homer	06/08/1999



Tabela

- Dataset com dados sobre o mesmo domínio/assunto.
- São compostas por linhas e colunas.
- Numa BD relacional, cada registo é unicamente definido por uma chave primária (PK – Primary Key).
- Cada coluna armazena um tipo específico de dados (ex: string, numeric, datetime, boolean, etc.).

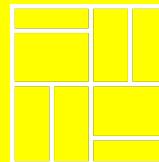
ID	Title	ISBN	Author	Publishing...
1	1984	2343454895456	George Orwell	04/12/1979
2	Anna Karenina	1234548485843	Leo Tolstoy	07/11/1998
4	The Adventures of I	3450345345443	Mark Twain	08/11/1999
5	Ulysses	9944933003232	James Joyce	06/05/2010
8	War and Peace	0944344903312	Leo Tolstoy	08/11/2001
11	The Brothers Karan	9003940397271	Doso	04/07/2012
12	On the Road	0459450444310	Jack Kerouac	30/12/2005
15	The Metamorphosi	2003948930545	Franz Kafka	09/03/1976
16	The Illiad	9449039333923	Homer	05/07/1998
17	The Odyssey	8409404850139	Homer	06/08/1999



View

- Parece uma tabela, mas é apenas uma tabela virtual – só existe quando é utilizada.
- São compostas por linhas e colunas.
- Alterações aplicadas nos dados da tabela correspondente são refletidas na View.
- É tipicamente um subset de uma tabela ou o resultado da junção de duas ou mais tabelas.

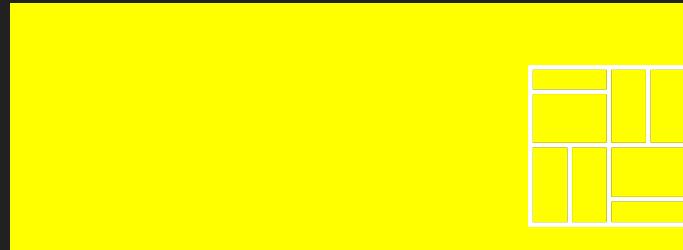
ID	Title	ISBN	Author	Publishing...
1	1984	2343454895456	George Orwell	04/12/1979
2	Anna Karenina	1234548485843	Leo Tolstoy	07/11/1998
4	The Adventures of I	3450345345443	Mark Twain	08/11/1999
5	Ulysses	9944933003232	James Joyce	06/05/2010
8	War and Peace	0944344903312	Leo Tolstoy	08/11/2001
11	The Brothers Karan	9003940397271	Doso	04/07/2012
12	On the Road	0459450444310	Jack Kerouac	30/12/2005
15	The Metamorphosi	2003948930545	Franz Kafka	09/03/1976
16	The Illiad	9449039333923	Homer	05/07/1998
17	The Odyssey	8409404850139	Homer	06/08/1999



Materialized View

- Ao contrário das views, as Views Materializadas são armazenadas fisicamente, o que se traduz em melhor performance.
- Podem ser indexadas, como as tabelas.
- São compostas por linhas e colunas.
- É possível definir a hora e frequência de refresh de dados de uma Materialized View
- Alterações aplicadas nos dados da tabela correspondente são refletidas na Materialized View após o refrescamento de dados.
- É tipicamente um subset de uma tabela ou o resultado da junção de duas ou mais tabelas

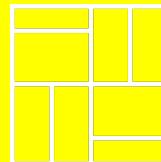
ID	Title	ISBN
1	1984	2343454895456
2	Anna Karenina	1234548485843
4	The Adventures of I	3450345345443
5	Ulysses	9944933003232
8	War and Peace	0944344903312
11	The Brothers Karan	9003940397271
12	On the Road	0459450444310
15	The Metamorphosi	2003948930545
16	The Illiad	9449039333923
17	The Odyssey	8409404850139



Primary Key (PK)

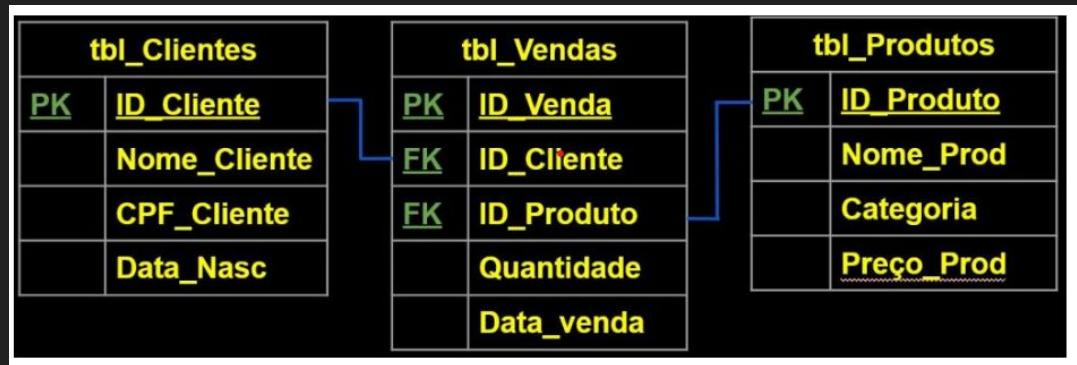
- Uma coluna (ou combinação de colunas) que identificam univocamente cada linha/registo.
- Apresenta um valor único que identifica exatamente cada registo.
- Nunca pode ser NULL.
- Também pode ser uma coluna que é especificamente gerada pela base de dados seguindo uma sequência (ex: 1, 2, 3, ...).

ID
1
2
4
5
8
11
12
15
16
17

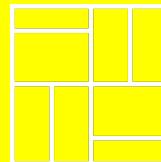


Foreign Key (FK)

- Também conhecida como referencing key.
- Chave utilizada para ligar duas tabelas.
- Coluna ou combinação de colunas que corresponde à PK de outra tabela.



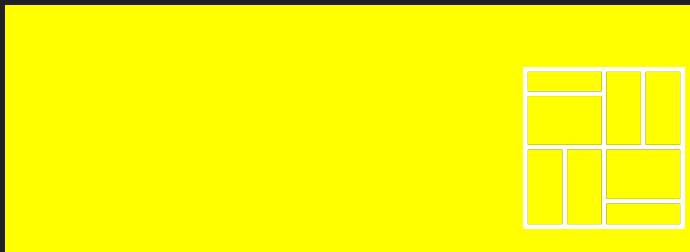
NOTA: A relação entre duas tabelas liga a PK de uma tabela à FK de outra tabela.



Índice

- Índices são utilizados melhorar a velocidade das consultas, permitindo encontrar linhas rapidamente e, como tal, melhorar a performance desta tarefa;
- Funcionam como um “mapa” que facilita encontrar registros específicos;
- Sem um índice, a pesquisa começa pela primeira linha e vai iterando pela tabela inteira
- Mais comuns: PRIMARY KEY, UNIQUE, INDEX

Tipo de índice	Unicidade	Nulo permitido	Quantidade na tabela	Objetivo principal
PRIMARY KEY	Sim	Não	Apenas um	Identificar unicamente registros
UNIQUE	Sim	Sim (apenas um)	Vários possíveis	Garantir unicidade em colunas
INDEX	Não	Sim	Vários possíveis	Melhorar a performance de consultas



Função

- Permite o encapsulamento de operações que podem ser constituídas por vários passos numa só função dentro da base de dados.
- Permite a reutilização de blocos lógicos de código, desde que abstratos o suficiente.
- Podem ser criadas em várias linguagens (ex: SQL, C#, Python, etc.)
- Por exemplo: Remover as linhas com salários negativos na tabela

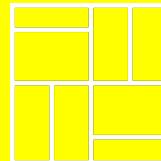
```
CREATE [OR REPLACE] FUNCTION function_name (arguments)
RETURNS return_datatype AS $variable_name$
DECLARE
declaration;
[...]
BEGIN
<function_body>
[...]
RETURN { variable_name | value }
END; LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION increment(i integer)
RETURNS integer AS $$
BEGIN
RETURN i + 1;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE FUNCTION clean_emp() RETURNS void AS '
DELETE FROM emp
WHERE salary < 0;
' LANGUAGE SQL;

SELECT clean_emp();

clean_emp
-----
(1 linha)
```

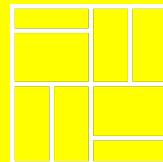


Sequência

- É um tipo especial de dados, criado para gerar identificadores numéricos únicos.
- Tipicamente usada para criar PKs artificiais – porquê?

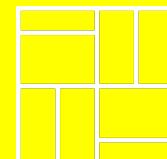
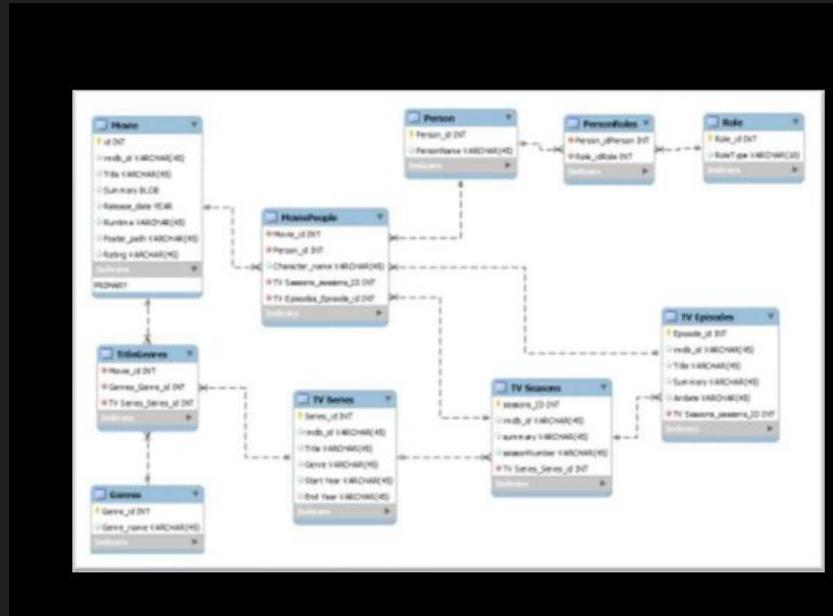
Primary Key (PK)

- Uma coluna (ou combinação de colunas) que identificam univocamente cada linha/registo
- **Apresenta um valor único que identifica claramente cada registro**
- Nunca pode ser Null
- Também pode ser uma coluna que é especificamente gerada pela base de dados seguindo uma sequência (ex: 1, 2, 3, ...).



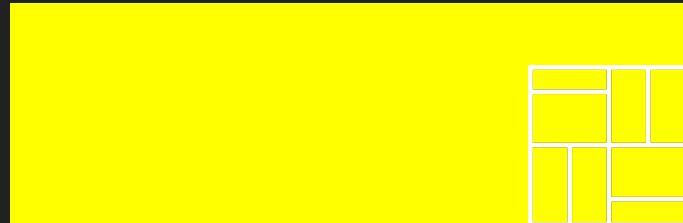
Modelo de dados

- Consiste na definição da base de dados.
- É o desenho por trás da BD, que sustenta os dados, independentemente da tecnologia de armazenamento.
- Fornece uma compreensão conceptual de como a base de dados está estruturada.



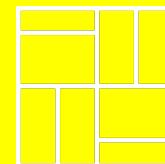
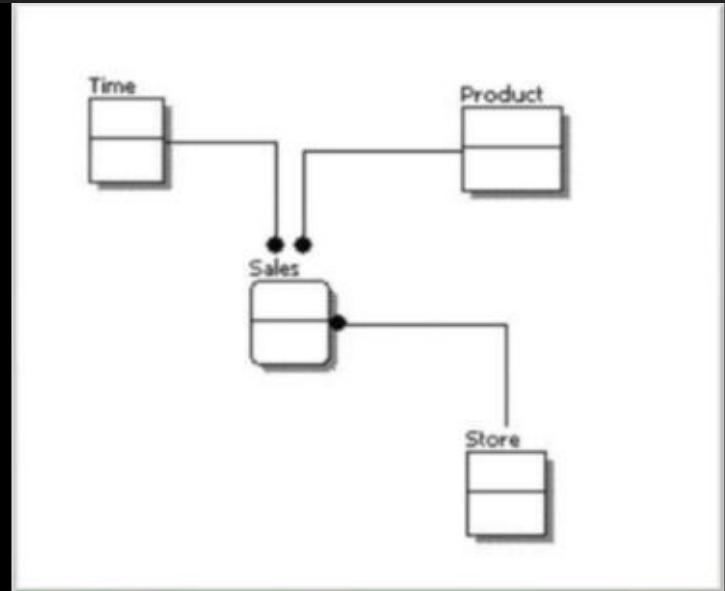
Modelo de dados - Diagrama de Entidades e Relações

- 3 conceitos importantes – Entidade, Atributo e Relação.
 - **Entidade**: uma entidade do mundo real (ex: um Cliente)
 - **Atributo**: uma característica ou propriedades da entidade (ex: a idade)
 - **Relação**: Dependência ou associação entre duas entidades (ex: tabela Loja relacionada com tabela Localização)
- Os modelos de dados podem assumir 3 tipos diferentes:
 - modelo conceptual
 - modelo lógico
 - modelo físico



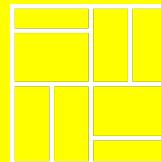
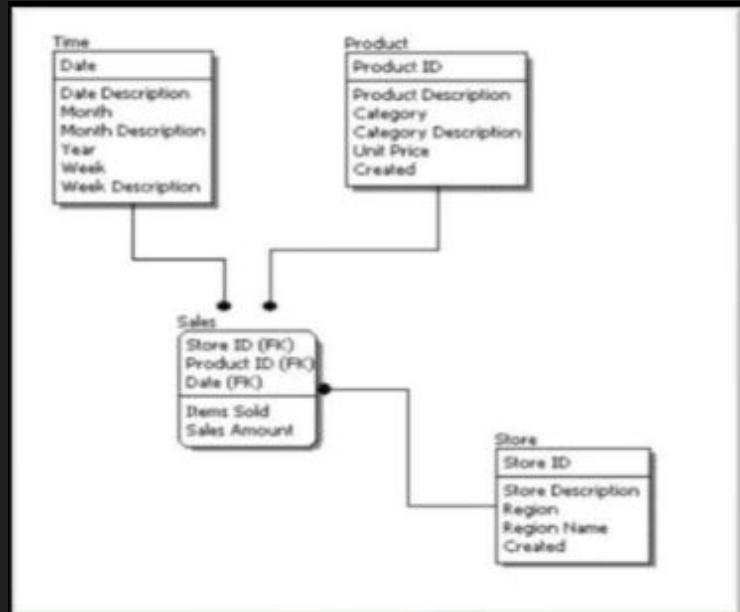
Modelo de dados Conceptual

- Alto nível (= menor detalhe)
- Audiência de negócio / funcional
- Entidades definidas, alguns atributos, relações
- Desenvolvido independentemente das especificações de hardware



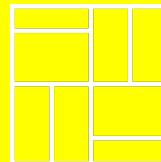
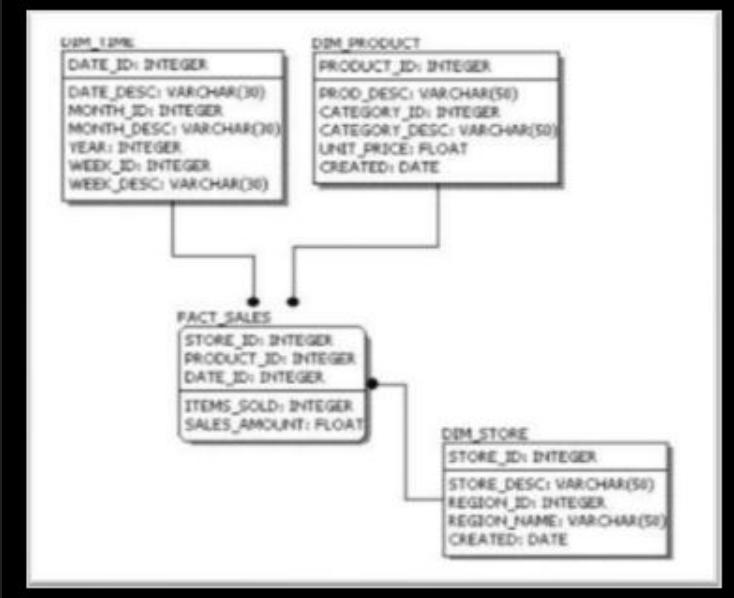
Modelo de dados Lógico

- Inclui mais detalhe de negócio
- Audiência híbrida (técnica / funcional)
- Mais tipos de entidade, atributos e relações
- Descreve requisitos de dados do ponto de vista de negócio



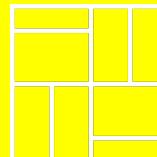
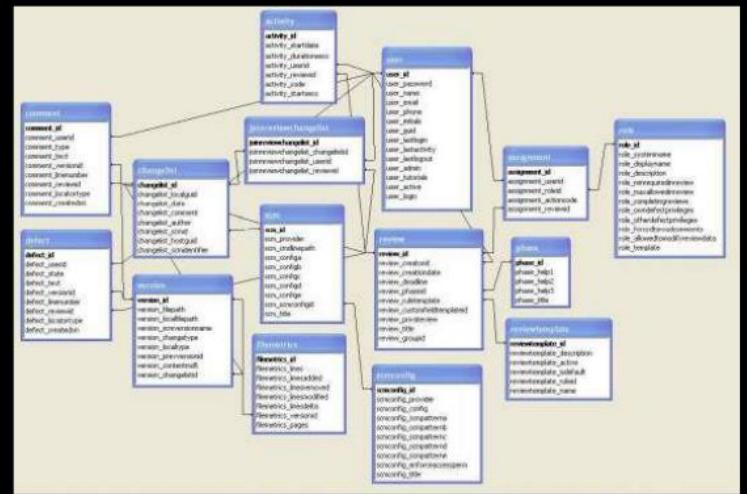
Modelo de dados Físico

- O desenho do schema interno da base de dados
- Capaz de reproduzir o schema de uma base de dados
- Inclui objetos físicos como views, constraints de PK e de FK, índices



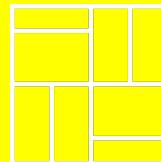
Schema

- O schema é o esqueleto da base de dados
- Representa como os dados estão organizados em tabelas e como as tabelas se relacionam entre elas
- Abstração lógica para agrupar objetos como tabelas, joins e keys.
- É a estrutura dos dados que assentam numa tecnologia de BD específica.
- Revela detalhe acerca da implementação da base de dados.



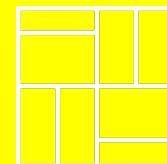
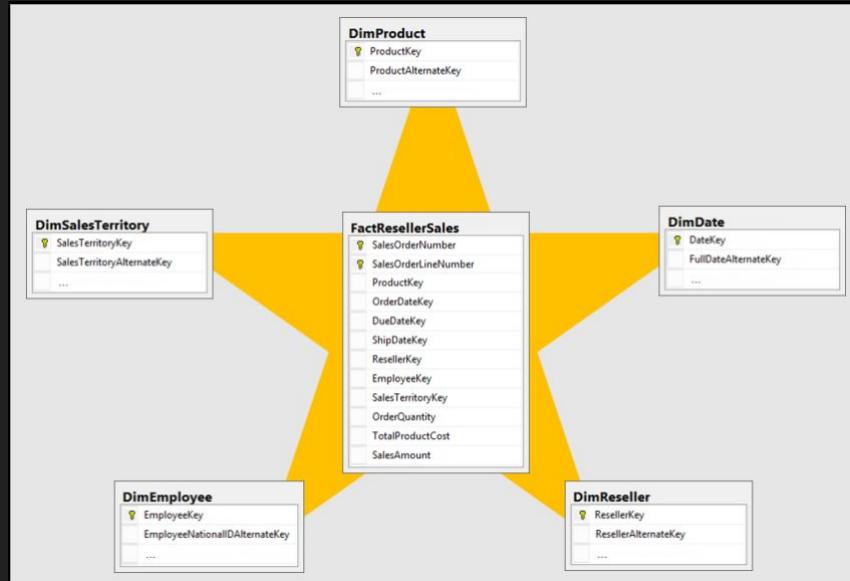
Diferença entre Modelo de Dados e Esquema de Dados

Aspecto	Modelo de Dados	Esquema de Dados
Nível	Conceitual, abstrato	Implementação específica no banco de dados
Objetivo	Descrever a estrutura e relação dos dados	Estruturar os dados fisicamente no banco
Tipo	Relacional, hierárquico, NoSQL, etc.	Tabelas, colunas, chaves, índices, etc.
Alterações	Requer mudanças de paradigma	Afeta a estrutura e requer migração



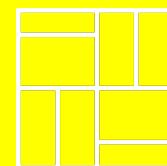
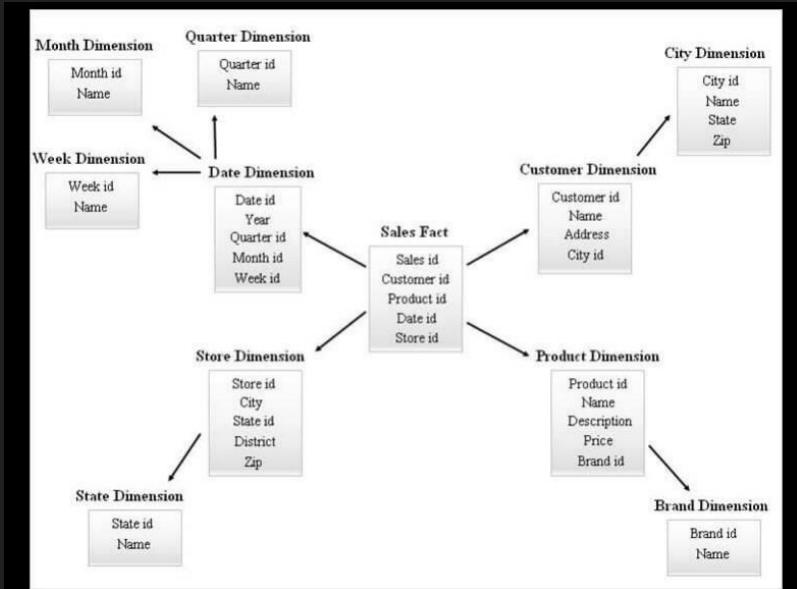
Star Schema

- Pode ser desenhado como uma estrela, uma tabela central com os factos, múltiplas tabelas de dimensão a radiar da mesma
- As dimensões não são normalizadas

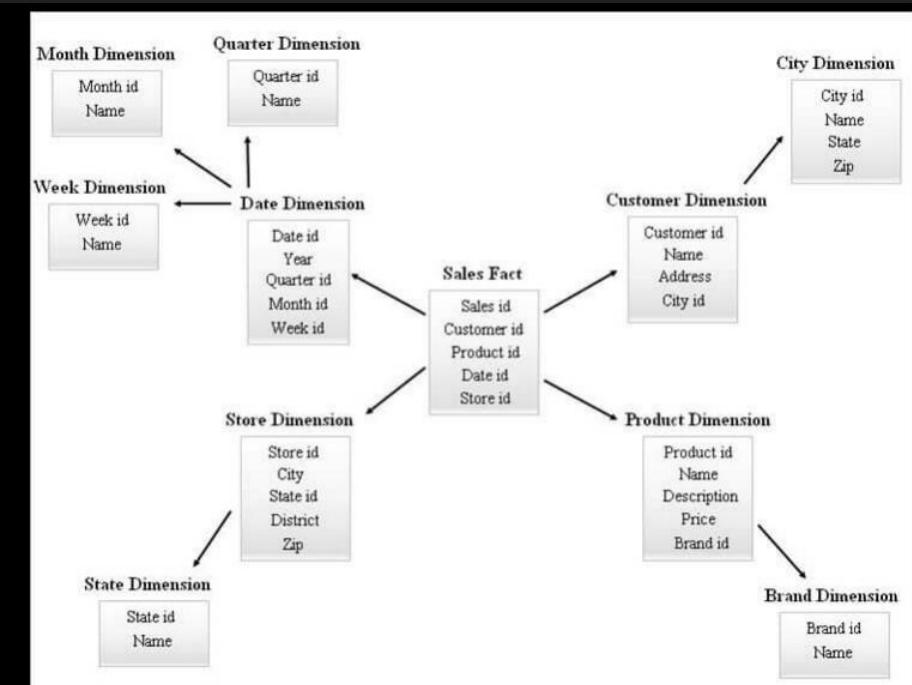
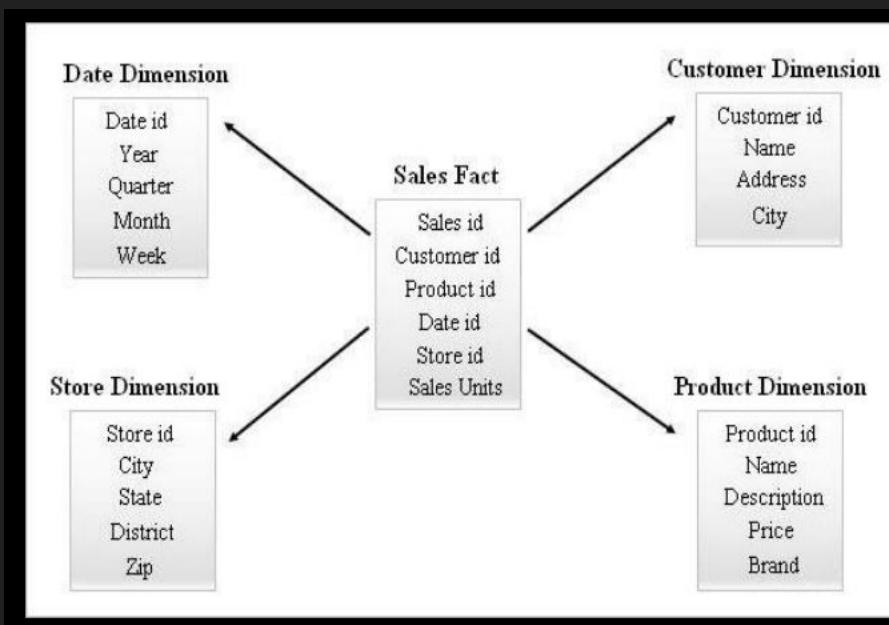


Snow Flake Schema

- Menor redundância de dados
- Menor espaço de armazenamento utilizado pelas dimensões
- Manutenção mais fácil (novos registos e updates)

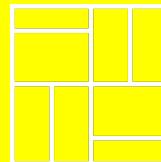


Star & Snow Flake Schema



Vantagens e Desvantagens

Star Schema	Snow Flake Schema
Maior redundância nos dados.	Menor redundância nos dados.
Necessário mais espaço de armazenamento para as tabelas de dimensão.	Espaço de armazenamento para as tabelas de dimensão significativamente menor.
Apresenta tabelas de dimensão não-normalizadas.	Apresenta tabelas de dimensão normalizadas.
Uma única tabela de factos é rodeada de múltiplas tabelas de dimensão.	Uma única tabela de factos é rodeada de múltiplas hierarquias de tabelas de dimensão.
As queries usam <i>joins</i> diretos entre os factos e as dimensões para obter os dados.	As queries usam <i>joins</i> complexos entre os factos e as dimensões para obter os dados.
O tempo de execução das queries é menor.	O tempo de execução das queries é maior.
Qualquer um consegue entender e desenhar o schema facilmente.	É mais difícil entender e desenhar o schema.
Abordagem <i>top-down</i> .	Abordagem <i>bottom-up</i> .



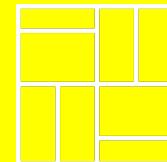
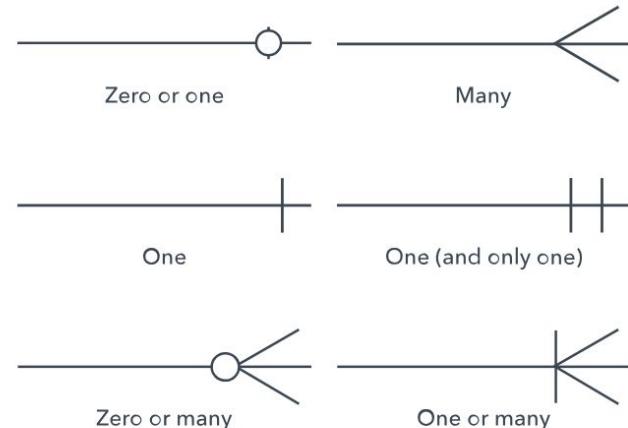
<https://www.lucidchart.com/pages/pt/simbolos-de-diagramas-entidade-relacionamento>

Cardinalidade e ordinalidade

Cardinalidade refere-se ao número máximo de vezes que a instância em uma entidade pode ser relacionada a instâncias de outra entidade. Ordinalidade, por sua vez, é o número mínimo de vezes que uma instância em uma entidade pode ser associada a uma instância em uma entidade relacionada.

A cardinalidade e a ordinalidade são representadas pelo estilo de uma linha e sua extremidade, de acordo com o estilo de notação escolhido.

+	One
<	Many
++	One (and only one)
O+	Zero or one
↖	One or many
○↖	Zero or many



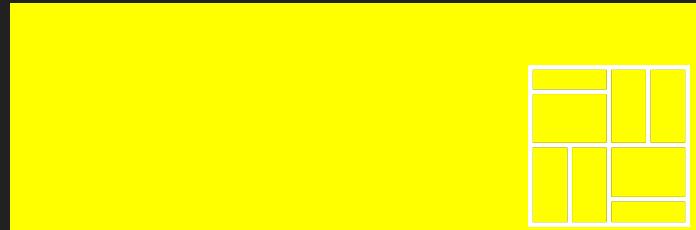
E | D | I | T.

Exercício Prático



Exemplo de um modelo relacional Ginásio

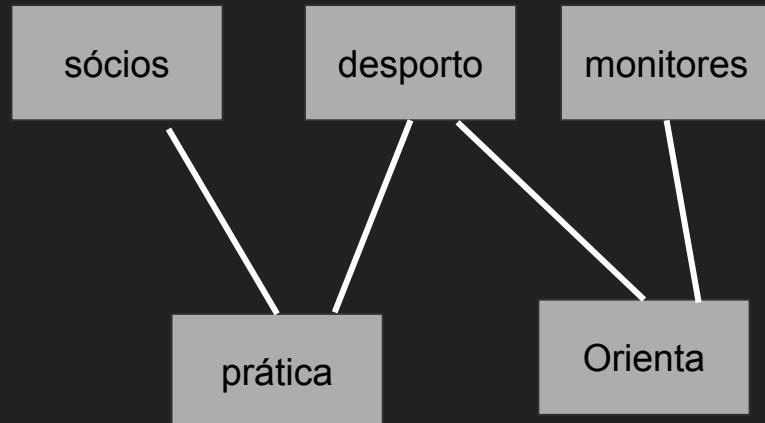
- 1º Criar diagrama conceptual - Definir o que o sistema contém - organizar e definir conceitos e regras comerciais**
- 2º Criar diagrama lógico - Distinguir relações entre entidades e atributos**
- 3º Criar diagrama físico - Definir o tipo de dados**



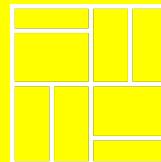
Exemplo de um diagrama conceptual

Ginásio

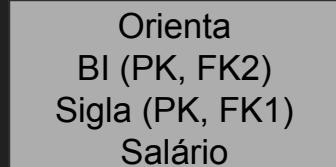
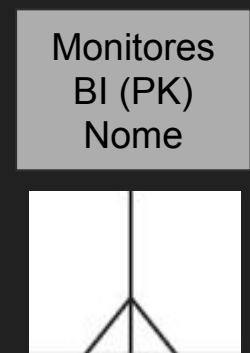
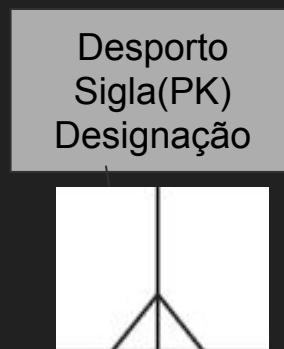
- Sócios
- Monitores
- Desporto
- Sócios praticam desporto e pagam um valor
- Monitores que orientam os sócios e recebem um salário



O foco da modelagem conceitual é detalhar e discutir o funcionamento do negócio do cliente, e não o uso de determinada tecnologia.

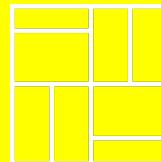


Exemplo de um Diagrama Lógico Ginásio

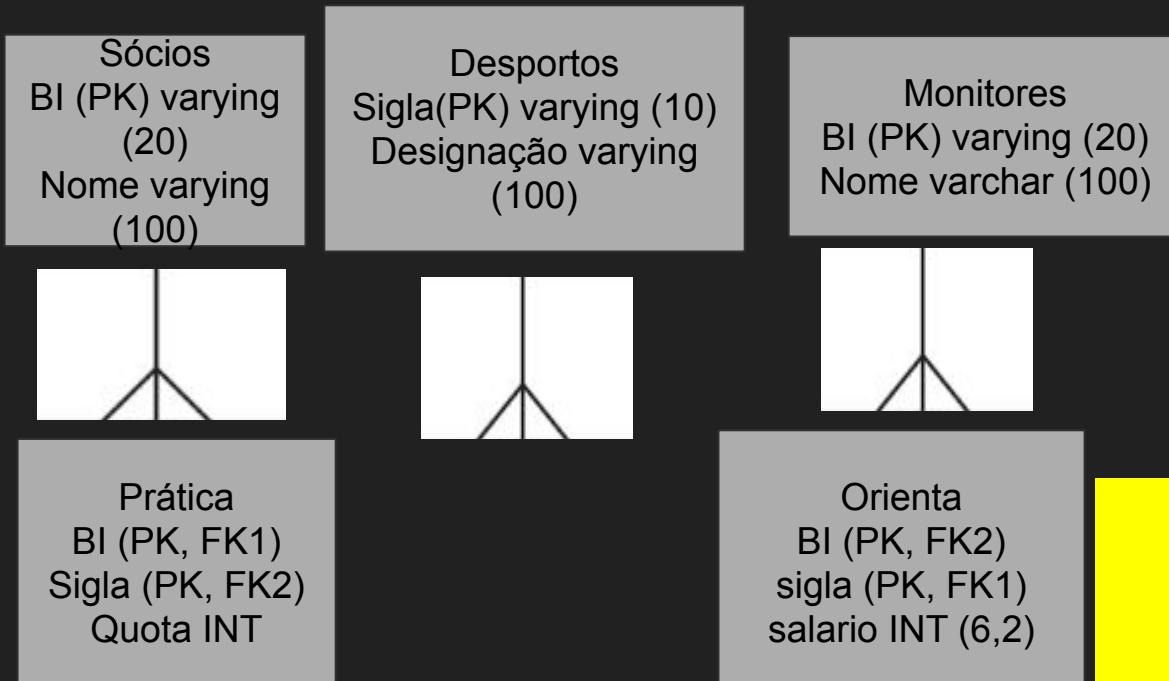


Para criar um modelo lógico do banco de dados mais coeso, são necessárias revisões na descrição do modelo conceitual:

- *Tabelas*
- *Campos*
- *Relacionamentos*
- *Cardinalidades*

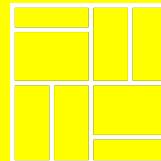


Exemplo de um Diagrama Físico Ginásio



Na criação do modelo físico (na BD) os acentos deverão ser evitados

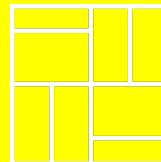
boa prática: criar tabelas e colunas com minúsculas



Exemplo de um modelo relacional Ginásio

- Gym example:

Socios		Monitores		Desportos		Pratica			Orienta		
BI	Nome	BI	Nome	Sigla	Designação	BI	Sigla	Quota	BI	Sigla	Salario
6078	Ana	9876	Luís	KB	Kickbox	6078	AE	25	1234	KB	40
5819	Rui	1234	Joana	NT	Natação	5819	KB	30	1234	NT	30
4526	Nuno			AE	Aeróbica	4526	KB	30	9876	NT	30
3955	Rita					4526	NT	20	9876	AE	35
9999	José					3955	KB	30			
						3955	NT	20			
						3955	AE	25			
						9876	KB	0			





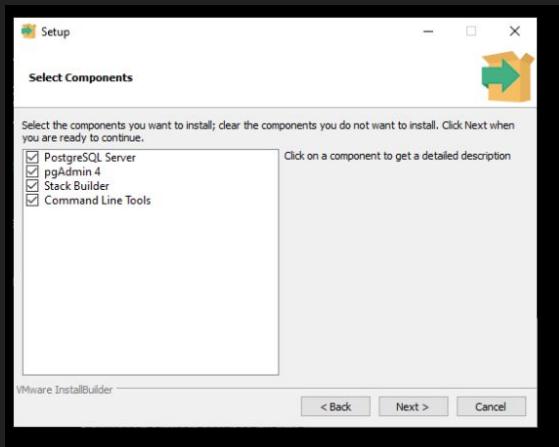
3. SQL Topics

- 3.1. PostgreSQL Instalação
- 3.2. Conceitos básicos
- 3.3. Exercises

PostgreSQL - Instalação

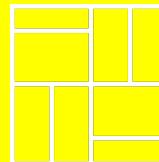
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

- <https://www.postgresql.org/>
- <https://www.postgresql.org/download/>
- Versão 14.5
- Grava a tua password!
- Port: 5432
- Grava o sumário de pré-instalação:



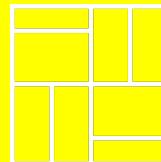
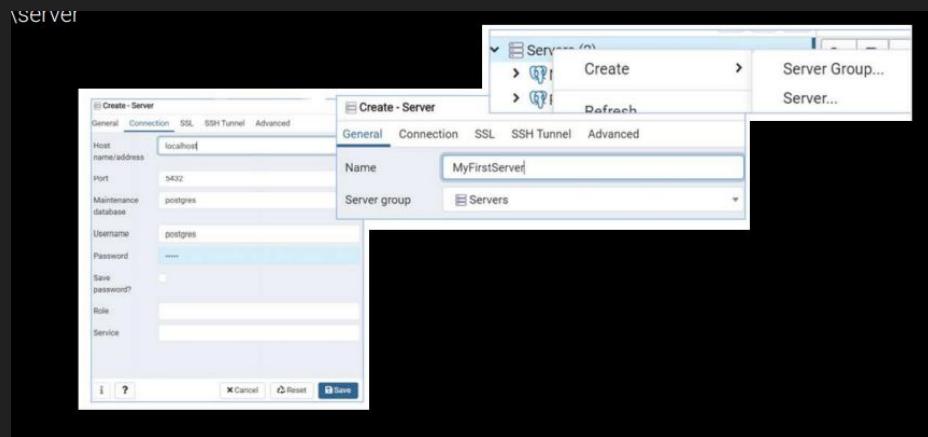
The following settings will be used for the installation::

```
Installation Directory: C:\Program Files\PostgreSQL\14
Server Installation Directory: C:\Program Files\PostgreSQL\14
Data Directory: C:\Program Files\PostgreSQL\14\data
Database Port: 5432
Database Superuser: postgres
Operating System Account: NT AUTHORITY\NetworkService
Database Service: postgresql-x64-14
Command Line Tools Installation Directory: C:\Program Files\PostgreSQL\14
pgAdmin4 Installation Directory: C:\Program Files\PostgreSQL\14\pgAdmin 4
Stack Builder Installation Directory: C:\Program Files\PostgreSQL\14
Installation Log: C:\Users\Administrator\AppData\Local\Temp\install-postgresql.log
```



PostgreSQL - validar a instalação (1/2)

- Abre o pgAdmin – C:\Program Files\PostgreSQL\14\pgAdmin 4\bin
- Se não existir nenhum server, cria um:
- Right-click em servers\create\server
- Preenche



PostgreSQL - validar a instalação (2/2)

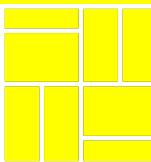
- Right-click em Database (1)\query tool
- Executa a query: SELECT version();



The screenshot shows the pgAdmin 4 interface. On the left, the 'Databases' tree is expanded, showing 'Databases (1)' and 'postgres'. Under 'postgres', several items are listed: Casts, Catalogs, Event Triggers, Extensions, and Foreign Data Wrappers. On the right, a query tool window is open. The title bar says 'postgres/postgres@PostgreSQL 14'. Below the title bar are icons for file operations and a toolbar with 'No limit' set. The main area has tabs for 'Query' and 'Query History', with 'Query' selected. A single line of code is entered: '1 SELECT version();'. Below the code, the results are displayed in a table:

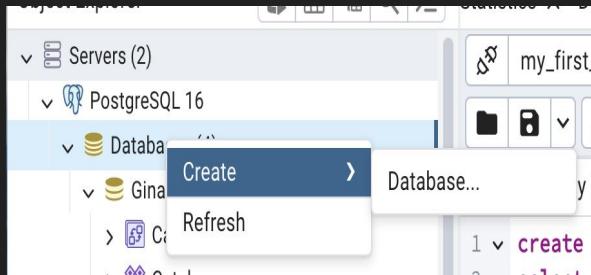
	version
1	PostgreSQL 14.5, compiled by Visual C++ build 1914, 64-bit

Below the table are buttons for 'Data Output', 'Explain', 'Messages', and 'Notifications'.

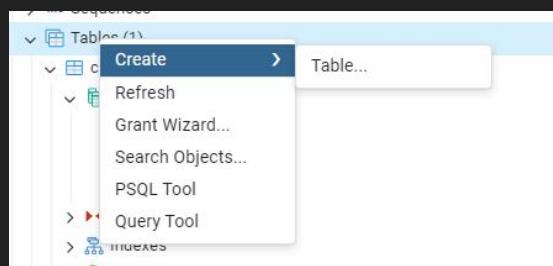


Criar database

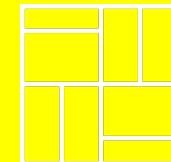
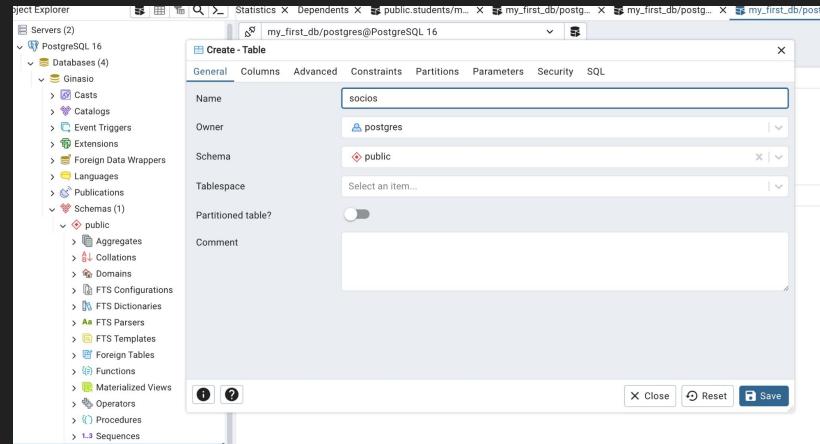
1º Criar Database



2º Criar tabela

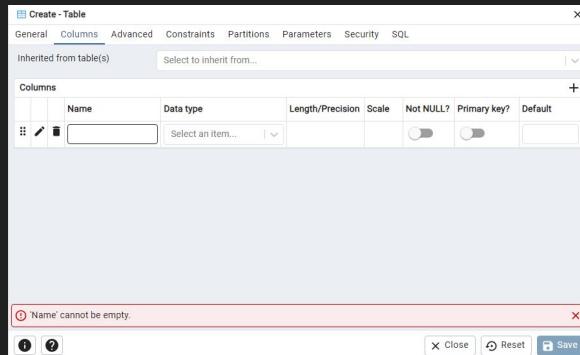


3º Nome da tabela



Criar tabelas

4º Criar colunas



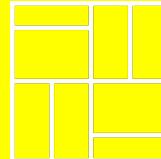
5º ver colunas

Data Output		Messages		Notifications	

	id [PK] bigint	first_name character varying	last_name character varying

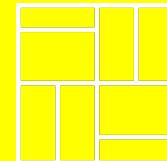
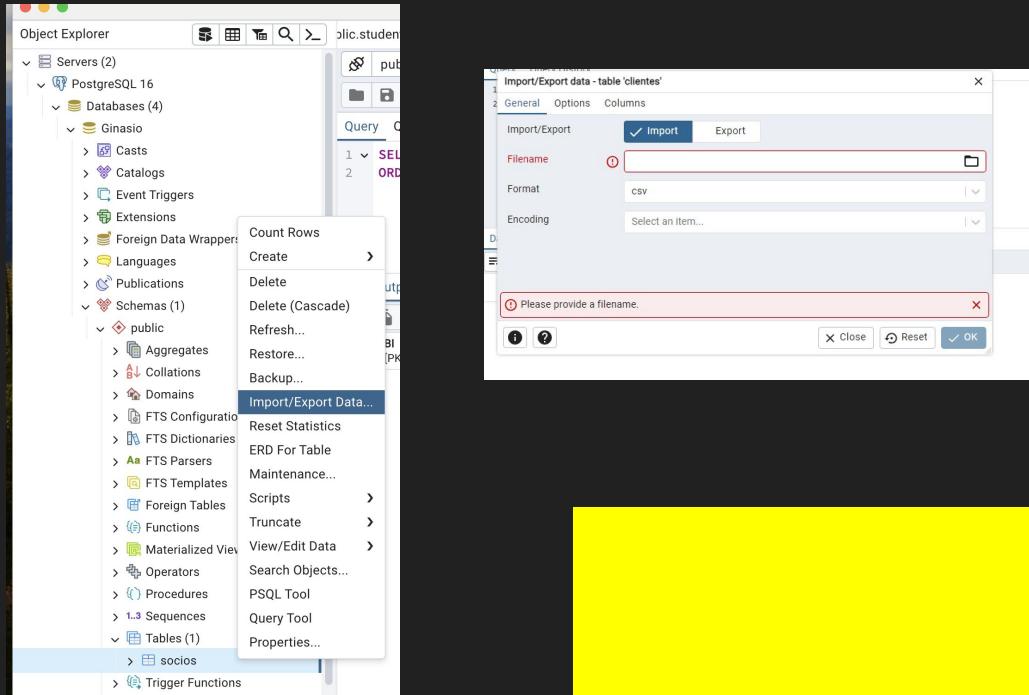
Ou podemos criar directo no sql:

```
CREATE TABLE public.socios ( bi character varying(20)  
PRIMARY KEY, nome character varying(100) NOT NULL  
);
```



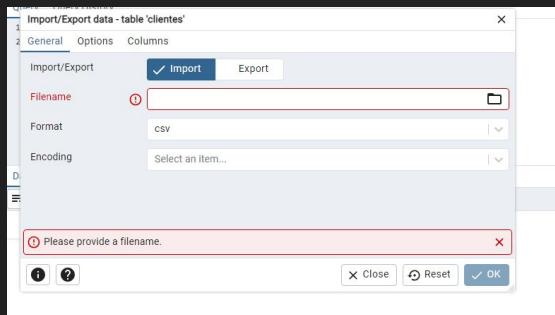
Adicionar dados

1º Importar excel ou outro ficheiro



Adicionar dados

1º Importar excel ou outro ficheiro



2º Manualmente

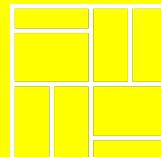
Data Output				Messages	Notifications
		id [PK] bigint	first_name character varying	last_name character varying	
1+	[default]	Maria	Maria	Manuel	

3º Usando a linguagem SQL

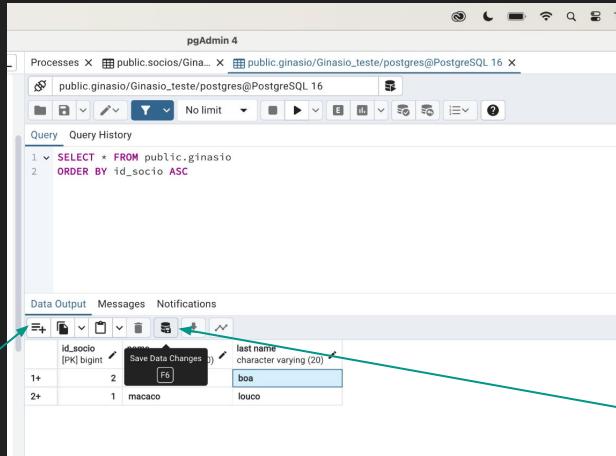
INSERT INTO public.socios
(bi, nome)
VALUES

('6078', 'Ana'),
(‘5819’, ‘Rui’),
(‘4526’, ‘Nuno’),
(‘3955’, ‘Rita’),
(‘9999’, ‘José’);

Data Output			Messages	Notifications
INSERT 0 1				
Query returned successfully in 37 msec.				

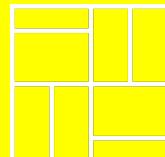


Adicionar dados manualmente

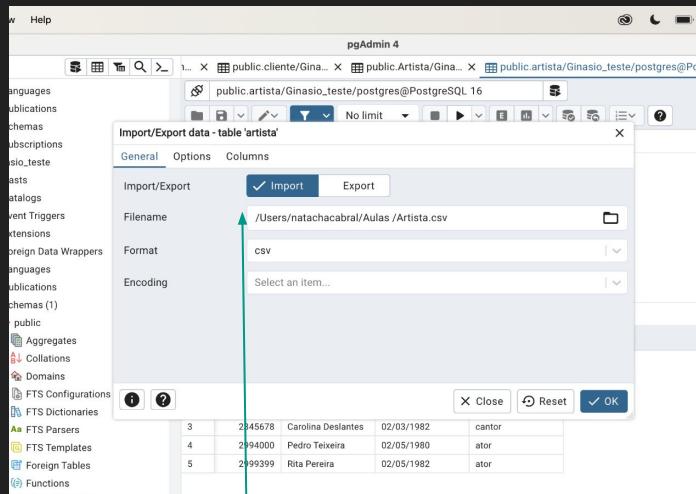


adicionar dados em mais

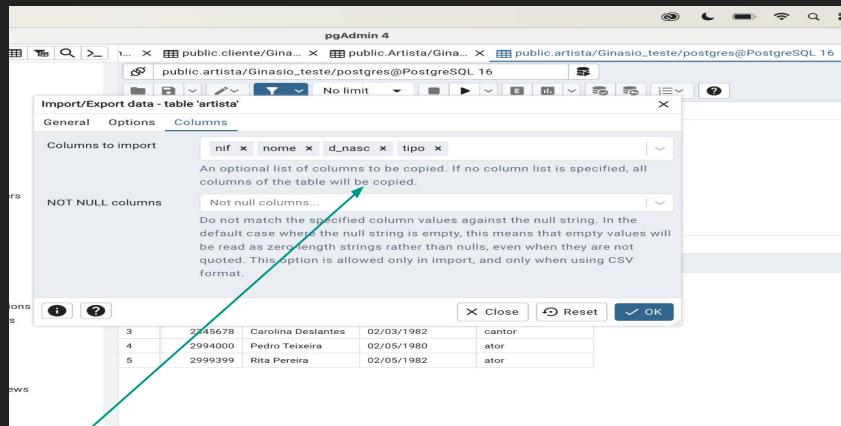
Salvar dados



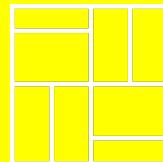
Importar excel ou outro ficheiro



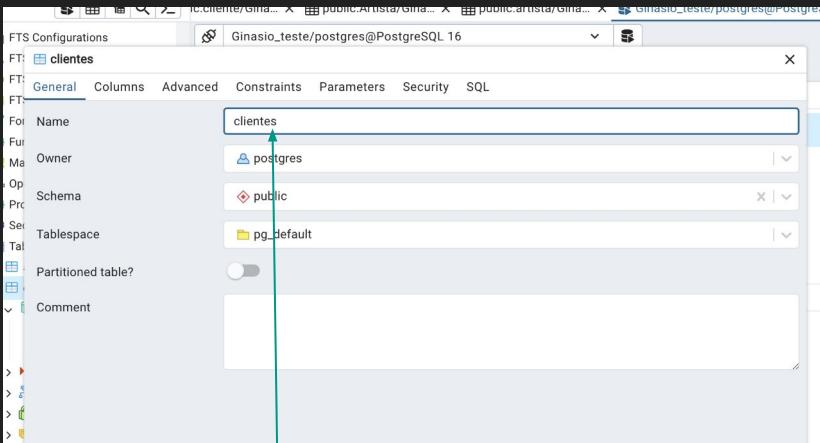
Criar tabela e Importar dados
em mais



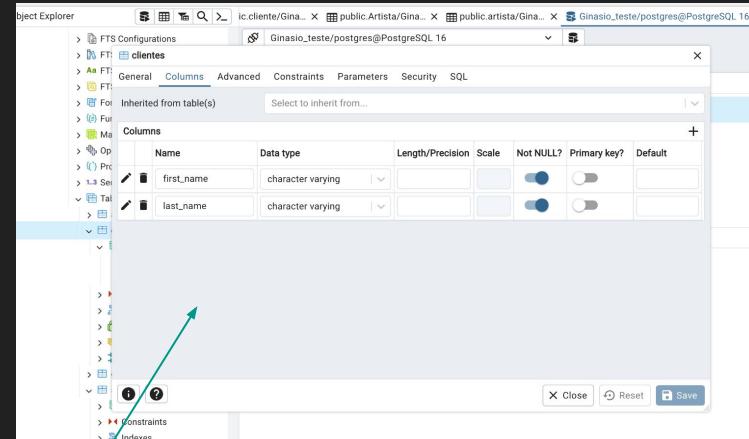
Garantir que colocamos
na tabela exactamente
o mesmo número de
coluna que existe no
ficheiro



Usando a linguagem sql

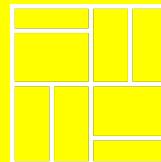


Criar tabela e colunas



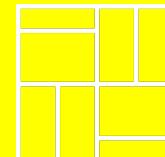
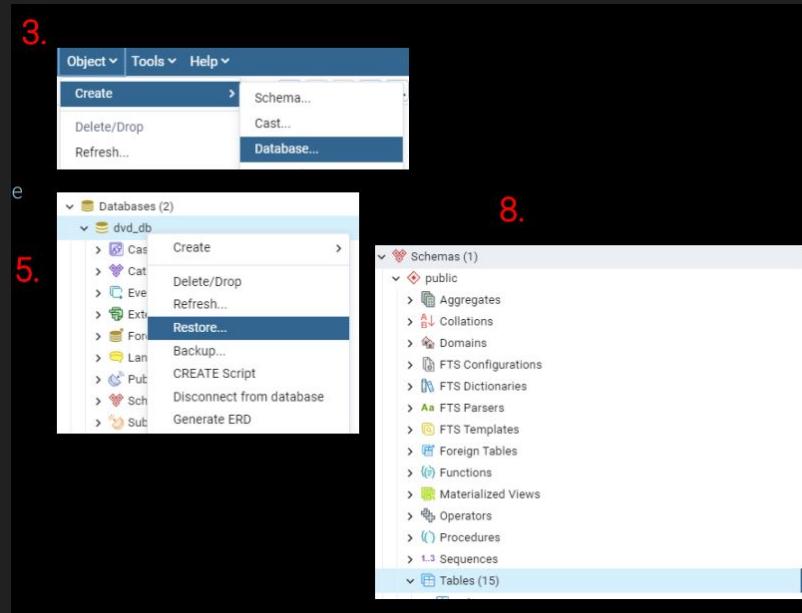
Garantir que colocamos na tabela exactamente o mesmo número de coluna que existe no ficheiro

```
INSERT INTO
public."clientes"("first_name",
"last_name")
VALUES ('Paulo','Coelho')
```



PostgreSQL - Criar BD

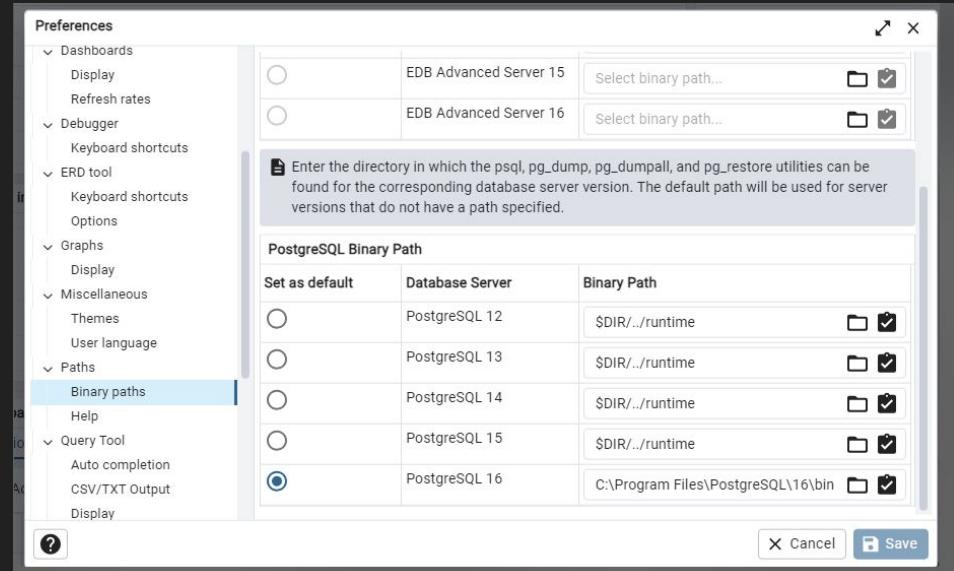
1. Faz download da BD: DVD Rental Sample Database
2. Extrai a BD (.tar) do ZIP fornecido
3. Clica em Object\Create\Database
4. Dá-lhe um nome, como “dvd_db” e guarda
5. Right-click em dvd_db\restore
6. Em file name, coloca o caminho para o .tar da BD
7. Carrega em Restore
8. Valida que consegues ver as 15 tabelas
9. Explora 😊



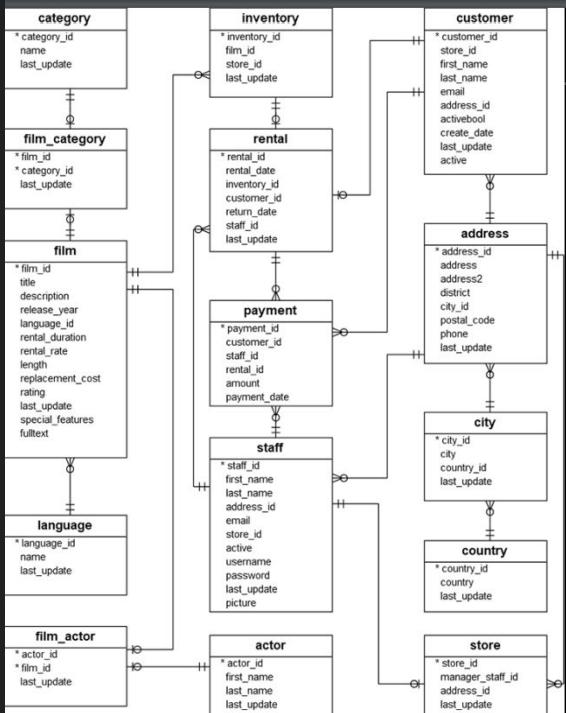
PostgreSQL - Criar BD

Caso não funcione ir a:

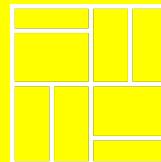
File\Preferences\Paths\Binary paths
e na parte da janela “PostgreSQL Binary Path” ir à versão instalada e no símbolo “pasta” colocar o caminho para o bin.
Depois fazer refresh em tudo e a seguir o restore



PostgreSQL - Criar BD



- DVD Rental Sample Database
- Modelo de dados... Lógico



E | D | I | T.

DATABASES FUNDAMENTALS