

FIAP GRADUAÇÃO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

BUILDING RELATIONAL DATABASE & SQL

Prof. Alan Barros dos Reis



profalan.reis@fiap.com.br

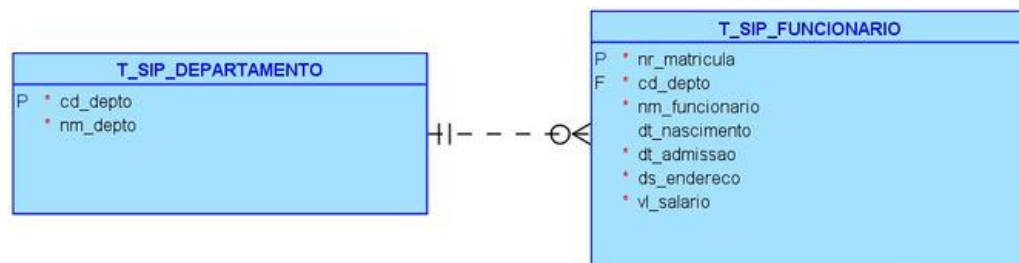
Introdução a Banco de Dados

EXEMPLOS DE DADOS

CD_DEPTO	NM_DEPTO
1	FINANCEIRO
2	TECNOLOGIA DA INFORMAÇÃO
3	CONTAS A PAGAR
4	FATURAMENTO
5	RECURSOS HUMANOS

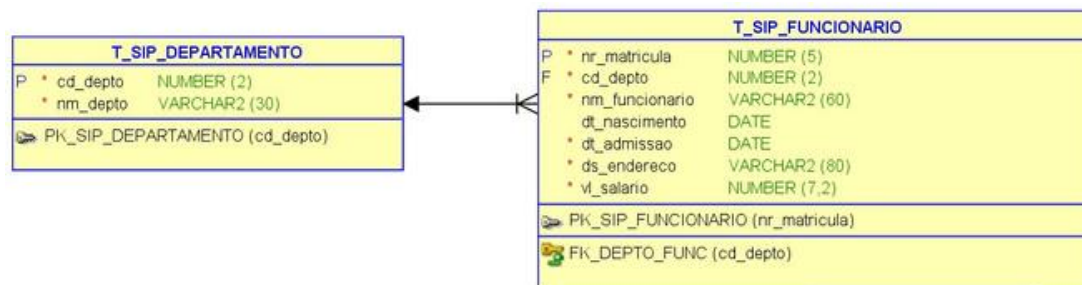
NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDEREÇO	VL_SALARIO
12345	1	JOAO DA SILVA	10/05/85	15/09/12	RUA X, 49	5684,66
12346	1	MANUEL DA SILVA	05/10/98	10/11/15	RUA X, 31	3542,11
12347	1	JANDIRA DA SILVA	10/12/00	15/09/18	RUA X, 25	1875,96
12348	2	KATIA REGINA SOUZA	15/01/95	03/10/15	RUA Y, 49	3894,63
12349	5	MARIA DAS DORES SOUZA	18/08/83	23/10/17	RUA Y, 35	1542,55
12350	2	ALFREDO DE SOUZA	04/05/99	03/10/15	RUA Y, 27	5874,52
12351	3	GISELE DE JESUS	15/04/99	20/03/17	RUA Z, 49	1020,66
12352	3	RAFAEL DE JESUS	10/08/98	10/08/12	RUA Z, 55	2563,44
12353	3	ROSANA DE JESUS	14/03/87	15/08/19	RUA Z, 79	4879,55
12354	4	JOSEFINA DE ALMEIDA	16/10/97	25/03/13	RUA Y, 33	4561,88
12355	4	LUCIANA DE ALMEIDA	10/02/84	28/09/11	RUA Y, 44	2345,52
12356	5	THIAGO DE ALMEIDA	10/03/98	24/10/18	RUA Y, 55	1254,22
12357	5	LARISSA DE CAMARGO	14/02/97	04/08/15	RUA V, 22	1245,55
12358	5	ANTONIO DE CAMARGO	25/01/85	12/08/16	RUA V, 44	2451,33
12359	5	JOSE DE CAMARGO	23/10/98	20/04/17	RUA V, 88	6541,22

Dados são representados, através de estruturas de armazenamento.



Visão lógica da estrutura de armazenamento.

Visão física da estrutura de armazenamento.



EXEMPLOS DE INFORMAÇÃO Apresentamos a informação referente **a quantidade de funcionários por departamento e a respectiva média salarial mensal**. A informação é obtida com base nos dados armazenados, que foram extraídos, conforme a necessidade do usuário.

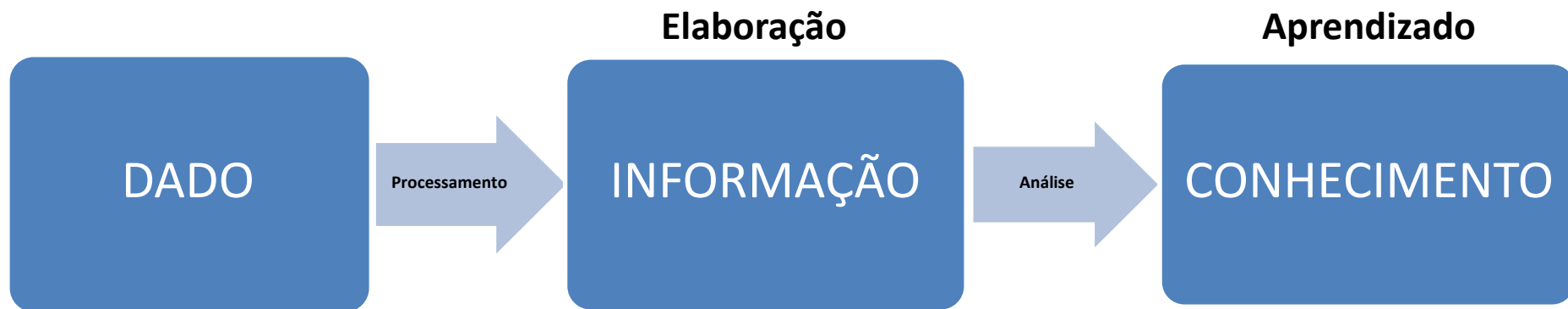
CD_DEPTO	NM_DEPTO	QTDE. FUNCIONARIOS	MEDIA SALARIAL
3	CONTAS A PAGAR	3	R\$2821.22
4	FATURAMENTO	2	R\$3453.70
1	FINANCEIRO	3	R\$3700.91
5	RECURSOS HUMANOS	5	R\$2606.97
2	TECNOLOGIA DA INFORMAÇÃO	2	R\$4884.58

Exemplo do comando
para extrair a
informação necessária.

```
SELECT D.CD_DEPTO,
       D.NM_DEPTO,
       COUNT(F.NR_MATRICULA) AS "QTDE. FUNCIONARIOS" ,
       TO_CHAR(ROUND(AVG(F.VL_SALARIO),2), 'L99999.99') AS "MEDIA SALARIAL"
FROM T_SIP_DEPTO D INNER JOIN T_SIP_FUNCIONARIO F
     ON (D.CD_DEPTO = F.CD_DEPTO)
GROUP BY D.CD_DEPTO, D.NM_DEPTO
ORDER BY D.NM_DEPTO;
```

DADO => INFORMAÇÃO => CONHECIMENTO

No exemplo apresentamos a quantidade de funcionários por departamento e a respectiva média salarial mensal, estamos agregando o conhecimento. O conhecimento adquirido atrelado a inteligência, nos permite resolver problemas.



A informação é considerada como sendo um dos **ativos estratégicos de maior importância** dentro de uma empresa. Sendo assim, um dos recursos mais importantes para qualquer empresa é a sua coleção de dados. Um banco de dados pode ajudar

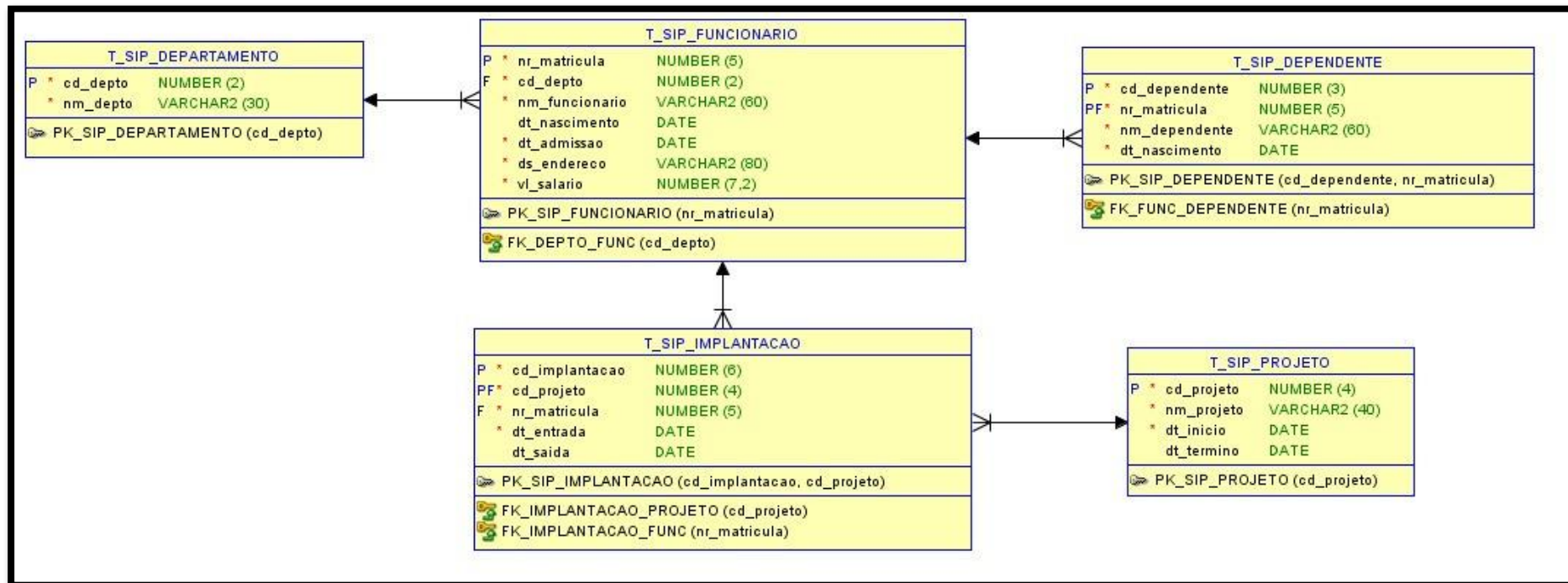
- Na organização de uma quantidade crescente de informações;
- Na geração de informações que contribuam aos negócios da empresa



- **coleção lógica e coerente de dados** (dados dispostos de forma desordenada não pode ser referenciado como banco de dados);
- **é projetado, construído e populado** com dados para um propósito específico;
- um banco de dados possui um **conjunto pré-definido de usuários** e aplicações;
- **representa algum aspecto do mundo real**, porção da realidade, o qual é chamado de “**mini-mundo**” ou de “**universo de discurso**”; qualquer alteração efetuada no mini-mundo é automaticamente refletida no banco de dados.

Propriedades de um Banco de Dados

PROJETADO: Exemplo do projeto físico do banco de dados



Propriedades de um Banco de Dados

CONSTRUÍDO: Exemplo do conjunto de códigos, através da linguagem SQL, para implementar o banco de dados.

```
-- SCRIPT IMPLANTAÇÃO DE PROJETOS

-- Exclusão das tabelas e respectivos relacionamentos
DROP TABLE T_SIP_DEPARTAMENTO CASCADE CONSTRAINTS ;
DROP TABLE T_SIP_DEPENDENTE CASCADE CONSTRAINTS ;
DROP TABLE T_SIP_FUNCIONARIO CASCADE CONSTRAINTS ;
DROP TABLE T_SIP_IMPLANTACAO CASCADE CONSTRAINTS ;
DROP TABLE T_SIP_PROJETO CASCADE CONSTRAINTS ;

-- Criação das tabelas

-- Tabela DEPARTAMENTO
CREATE TABLE T_SIP_DEPARTAMENTO
(
    cd_depto NUMBER (2) NOT NULL ,
    nm_depto VARCHAR2 (30) NOT NULL
) ;

-- Criação de constraints
ALTER TABLE T_SIP_DEPARTAMENTO ADD CONSTRAINT PK_SIP_DEPARTAMENTO PRIMARY KEY ( cd_depto ) ;
ALTER TABLE T_SIP_DEPARTAMENTO ADD CONSTRAINT UN_SIP_DEPTO_NOME UNIQUE ( nm_depto ) ;

-- Tabela DEPENDENTE
CREATE TABLE T_SIP_DEPENDENTE
(
    cd_dependente NUMBER (3) NOT NULL ,
    nr_matricula NUMBER (5) NOT NULL ,
```

Propriedades de um Banco de Dados

POPULADO: Exemplo de registros (linhas), introduzidos na estrutura implementada, visando avaliar a estrutura e extrair as informações, conforme necessidades do usuário.

```
-- SCRIPT PARA POPULAR O BD IMPLANTACAO DE PROJETOS

-- ELIMINA O CONTEUDO DAS TABELAS
/*
DELETE FROM T_SIP_IMPLANTACAO;
DELETE FROM T_SIP_PROJETO;
DELETE FROM T_SIP_DEPENDENTE;
DELETE FROM T_SIP_FUNCIONARIO;
DELETE FROM T_SIP_DEPTO;
*/

-- POPULAR A TABELA DEPTO COM 5 DEPTOS
INSERT INTO T_SIP_DEPTO VALUES (1, 'FINANCEIRO');
INSERT INTO T_SIP_DEPTO VALUES (2, 'TECNOLOGIA DA INFORMAÇÃO');
INSERT INTO T_SIP_DEPTO VALUES (3, 'CONTAS A PAGAR');
INSERT INTO T_SIP_DEPTO VALUES (4, 'FATURAMENTO');
INSERT INTO T_SIP_DEPTO VALUES (5, 'RECURSOS HUMANOS');

-- POPULAR A TABELA FUNCIONARIO COM 3 FUNCIONARIOS PARA CADA DEPTO
-- DEPTO = 1
INSERT INTO T_SIP_FUNCIONARIO VALUES
(12345, 1, 'JOAO DA SILVA', TO_DATE('10/05/1985','DD/MM/YYYY'),
TO_DATE('15/09/2012','DD/MM/YYYY'),'RUA X, 49', 5684.66);

INSERT INTO T_SIP_FUNCIONARIO VALUES
(12346, 1, 'MANUEL DA SILVA', TO_DATE('05/10/1998','DD/MM/YYYY'),
TO_DATE('10/11/2015','DD/MM/YYYY'),'RUA X, 31', 3542.11);
```

NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDERECO	VL_SALARIO
12345	1	JOAO DA SILVA	10/05/85	15/09/12	RUA X, 49	5684,66
12346	1	MANUEL DA SILVA	05/10/98	10/11/15	RUA X, 31	3542,11
12347	1	JANDIRA DA SILVA	10/12/00	15/09/18	RUA X, 25	1875,96
12348	2	KATIA REGINA SOUZA	15/01/95	03/10/15	RUA Y, 49	3894,63
12349	5	MARIA DAS DORES SOUZA	18/08/83	23/10/17	RUA Y, 35	1542,55
12350	2	ALFREDO DE SOUZA	04/05/99	03/10/15	RUA Y, 27	5874,52
12351	3	GISELE DE JESUS	15/04/99	20/03/17	RUA Z, 49	1020,66
12352	3	RAFAEL DE JESUS	10/08/98	10/08/12	RUA Z, 55	2563,44
12353	3	ROSANA DE JESUS	14/03/87	15/08/19	RUA Z, 79	4879,55
12354	4	JOSEFINA DE ALMEIDA	16/10/97	25/03/13	RUA Y, 33	4561,88
12355	4	LUCIANA DE ALMEIDA	10/02/84	28/09/11	RUA Y, 44	2345,52
12356	5	THIAGO DE ALMEIDA	10/03/98	24/10/18	RUA Y, 55	1254,22
12357	5	LARISSA DE CAMARGO	14/02/97	04/08/15	RUA V, 22	1245,55
12358	5	ANTONIO DE CAMARGO	25/01/85	12/08/16	RUA V, 44	2451,33
12359	5	JOSE DE CAMARGO	23/10/98	20/04/17	RUA V, 88	6541,22

CD_DEPTO	NM_DEPTO
1	FINANCEIRO
2	TECNOLOGIA DA INFORMAÇÃO
3	CONTAS A PAGAR
4	FATURAMENTO
5	RECURSOS HUMANOS

Exemplo da estrutura de armazenamento populada

SQLx NoSQL

De forma simples a principal diferença entre banco de dados relacional e não relacional está relacionada, com o modo como as informações são inseridas e organizadas.

- ☐ O banco de dados relacional oferece maior consistência e confiabilidade, mas exige o relacionamento entre várias tabelas para o acesso à informação.
- ☐ O não relacional tem como vantagem uma escalabilidade (capacidade de crescimento) maior, com a informação agrupada e armazenada no mesmo registro.

Sistema de Banco de Dados: Requisitos SGBD

Os dados podem ser compartilhados: Várias aplicações podem acessar os mesmos dados de um banco de dados. E oferecer visões diferentes, conforme a necessidade do usuário.

Abaixo temos um banco de dados de "ALUNOS", contendo: Alunos, Disciplinas, Turmas, Histórico e Pré-Requisitos. Essa estrutura pode ser acessada/visualizada por diferentes usuários e para cada usuário é compartilhado os dados conforme as necessidades e regras de negócio.

ALUNO

Nome	Numero_aluno	Tipo_aluno	Curso
Silva	17	1	CC
Braga	8	2	CC

DISCIPLINA

Nome_disciplina	Numero_disciplina	Creditos	Departamento
Introd. à ciência da computação	CC1310	4	CC
Estruturas de dados	CC3320	4	CC
Matemática discreta	MAT2410	3	MAT
Banco de dados	CC3380	3	CC

TURMA

Identificacao_turma	Numero_disciplina	Semestre	Ano	Professor
85	MAT2410	Segundo	07	Kleber
92	CC1310	Segundo	07	Anderson
102	CC3320	Primeiro	08	Carlos
112	MAT2410	Segundo	08	Chang
119	CC1310	Segundo	08	Anderson
135	CC3380	Segundo	08	Santos

HISTORICO_ESCOLAR

Numero_aluno	Identificacao_turma	Nota
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PRE_REQUISITO

Numero_disciplina	Numero_pre_requisito
CC3380	CC3320
CC3380	MAT2410
CC3320	CC1310

DADO_ESCOLAR					
Nome_aluno	Historico_escolar_aluno				
	Numero_disciplina	Nota	Semestre	Ano	Identificacao_turma
Silvah	CC1310	C	Segundo	08	119
	MAT2410	B	Segundo	08	112
Braga	MAT2410	A	Segundo	07	85
	CC1310	A	Segundo	07	92
	CC3320	B	Primeiro	08	102
	CC3380	A	Segundo	08	135

A figura representa a necessidade de um usuário, em apenas acessar e imprimir o histórico escolar de cada aluno.

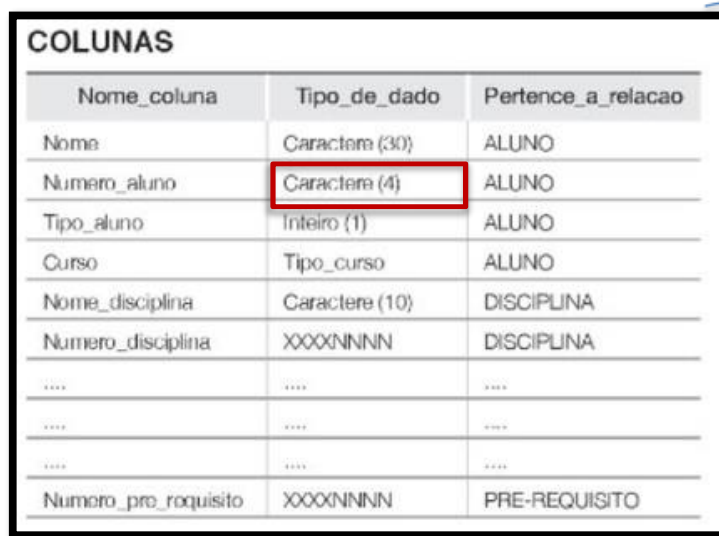
A figura representa a necessidade de um usuário, em apenas verificar se os alunos possuem todos os pré-requisitos para cada disciplina em que se inscreveram.

PRE_REQUISITO_DISCIPLINA

Nome_disciplina	Numero_disciplina	Pre_requisitos
Banco de dados	CC3380	CC3320
		MAT2410
Estrutura de dados	CC3320	CC1310

Sistema de Banco de Dados: Requisitos SGBD

Restrições de Integridade É assegurar que os dados de um banco de dados estão corretos. O tipo mais simples de **restrição de integridade**, é a capacidade de especificar o tipo de dado e tamanho para cada item de dado.



A diagram showing a table with columns and data types. A red box highlights the entry 'Caractere (4)' for the column 'Numero_aluno'. A blue arrow points from the red box to the text 'Restrições de integridade estão associadas a regras de negócio'.

Nome_coluna	Tipo_de_dado	Pertence_a_relacao
Nome	Caractere (30)	ALUNO
Numero_aluno	Caractere (4)	ALUNO
Tipo_aluno	Inteiro (1)	ALUNO
Curso	Tipo_curso	ALUNO
Nome_disciplina	Caractere (10)	DISCIPLINA
Numero_disciplina	XXXXNNNN	DISCIPLINA
....
....
....
Numero_pre_requisito	XXXXNNNN	PRE-REQUISITO

Restrições de integridade estão associadas a regras de negócio. Esse conjunto de necessidades são identificadas, e é de responsabilidade do projetista do banco de dados implementar essas regras. **Regras de negócio:** São premissas e restrições que precisam ser aplicadas, para que uma operação (negócio) possa funcionar da maneira esperada.

Restrições de Integridade Podemos restringir o conteúdo que um determinado item de dado terá, dentro da estrutura de armazenamento, através de **restrições adicionais**, conhecidas como **CONSTRAINTS**. Podemos fazer uma checagem, por exemplo:

T_ALUNO

P * nr_aluno NUMBER (5)
* nm_aluno VARCHAR2 (50)
* dt_nascimento DATE
ds_email VARCHAR2 (40)
* st_aluno NUMBER

PK_ALUNO (nr_aluno)

```
1 CREATE TABLE t_aluno (  
2   nr_aluno      NUMBER(5) NOT NULL,  
3   nm_aluno      VARCHAR2(50) NOT NULL,  
4   dt_nascimento DATE NOT NULL,  
5   ds_email      VARCHAR2(40),  
6   st_aluno      NUMBER NOT NULL,  
7 )  
8  
9  
10 COMMENT ON COLUMN t_aluno.st_aluno IS  
11   'status do aluno'  
12 1- Cancelado  
13 2-Trancado  
14 3-Ativo  
15 4-Inativo';  
16  
17 ALTER TABLE t_aluno  
18   ADD CONSTRAINT ck_aluno_status CHECK ( st_aluno IN (  
19     1,  
20     2,  
21     3,  
22     4  
23   ) );  
24  
25 ALTER TABLE t_aluno ADD CONSTRAINT pk_aluno PRIMARY KEY ( nr_aluno );
```

Propriedades da Tabela - T_ALUNO

Restrições do Nível da Tabela

Nome	Regra de Validação	Gerar em DDL
1 OK_ALUNO_STATUS	st_aluno in (1,2,3,4)	<input checked="" type="checkbox"/>

Regra de Validação
st_aluno in (1,2,3,4)

Comentários Observações

OK Aplicar Regras de Nomeação Cancelar Ajuda

Utilizando ferramentas para modelagem, é possível gerar o código, inclusive com a restrição criada para atender uma regra de negócio.

Restrições de Integridade

Em banco de dados relacional, cada registro de aluno, por exemplo, deve ser isto é identificado de forma única, ou seja, deve possuir um valor exclusivo, chamado de **restrição de chave** (chave primária) ou **singularidade**.

Constraint Primary Key



Restrição de Acesso (Privacidade dos Dados)



Cria contas e especifica as restrições de acesso

Restrições

- ✓ Apenas Leitura
- ✓ Leitura e Atualização



SISTEMAS DE INFORMAÇÃO (Aplicações)



Gerente Financeiro

Recebe relatórios financeiros confidenciais, contendo receitas e despesas para tomada de decisão.



Depto. Expedição

Recebe o que foi vendido e tem a responsabilidade de cuidar de todos os aspectos necessários para o envio de mercadorias.



PDV: Ponto de Venda

Realiza a venda e registra todos os produtos comercializados.



TIMELINE EM BDR

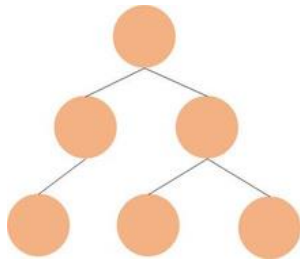


1960

Os Primeiros Modelos de Dados

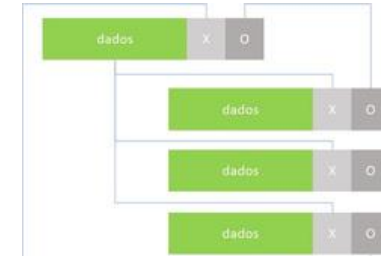
- Os fundamentos de bancos de dados surgiram na empresa IBM na década de 1960 através de pesquisas de funções de automação de escritório. As empresas descobriram que estava muito custoso empregar um número grande de pessoas para fazer trabalhos como armazenar e indexar (organizar) arquivos. Diante disso, surgiram dois modelos:

MODELO HIERÁRQUICO (IMS – INFORMATION MANAGEMENT SYSTEM).



- Também se utilizava de registros para representar os dados e links para os relacionamentos; São organizados na forma de uma árvore com raiz;

MODELO EM REDE (CODASYL – COMMITTEE FOR DATA SYSTEMS LANGUAGE)



- Os primeiros trabalhos foram realizados em 1964 por Charles Bachman; Dados são representados por uma coleção de registros e os relacionamentos por meio de links; É representado por um diagrama constituído por caixas e linhas;



1970

Modelo Relacional

- Estes modelo se baseia no princípio de que todos os dados estão guardados em tabelas e, que podem ser relacionados com outras tabelas.
- Esse conceito foi criado por Edgar Frank Codd, e descrito no artigo "**Relational Model of Data for Large Shared Data Banks**", sendo o primeiro modelo de dados descrito teoricamente.
- O modelo relacional se tornou um sucessor do modelo hierarquico e de rede e, permitiu que inúmeros trabalhos ou modelos fossem baseados a partir dele.
- Esse modelo tem como características:
 - Escalonamento Vertical;
 - Esquema Predefinido (Tabela e Colunas);
 - Prioriza a Consistência e a Qualidade do Dado;
 - Tabela se relacionam com outras Tabelas;



DR. EDGAR FRANK CODD (1923 -2003)
O PAI DO MODELO RELACIONAL

Information Retrieval

P. BAKENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users of terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information. Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on many relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, relational language, predicate calculus, security, data integrity

CR CATEGORIES: 3.70, 3.72, 3.75, 4.30, 4.32, 4.39

1. Relational Model and Normal Form

1.1. INTRODUCTION

This paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childe [1], the principal application of relations to data systems has been to deductive question-answering systems. Levin and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of data independence—the independence of application programs and terminal activities from growth in data type and changes in data representation—and certain kinds of data inconsistency which are expected to become troublesome even in nondeductive systems.

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for noninferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap"). Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. DATA DEPENDENCIES IN PRESENT SYSTEMS

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [3, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed without logically impairing some application programs is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependencies, indexing dependencies, and access path dependencies. In some systems these dependencies are not clearly separable from one another.

1.2.1. *Ordering Dependencies.* Elements of data in a data bank may be stored in a variety of ways, some involving no concern for ordering, some permitting each element to participate in one ordering only, others permitting each element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the

Volume 12 / Number 6 / June, 1970

Communications of the ACM 377

<https://www.seas.upenn.edu/~zives/03/cis550/codd.pdf>

1976

Modelo Entidade-Relacionamento (MER)

- Peter Chen, desenvolveu o modelo capaz de descrever um modelo de dados e ilustrar o aspectos de informação de um domínio de negócios ou seus requisitos.
- Denominado de Modelo Entidade-Relacionamento (MER), o mesmo é utilizado para representar uma implementação de um banco de dados relacional.
- Os principais componentes desse modelo são as Entidades, os Relacionamentos e as Cardinalidades.

DR. PETER CHEN
criador do modelo ER.

The Entity-Relationship Model—Toward a Unified View of Data

PETER PIN-SHAN CHEN
Massachusetts Institute of Technology

A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of database design and description using the model and the diagrammatic technique is given. Some implications for data integrity, information retrieval, and data manipulation are discussed.

The entity-relationship model can be used as a basis for unification of different views of data: the network model, the relational model, and the entity set model. Semantic ambiguities in these models are analyzed. Possible ways to derive their views of data from the entity-relationship model are presented.

Key Words and Phrases: database design, logical view of data, semantics of data, data models, entity-relationship model, relational model, data Base Task Group, network model, entity set model, data definition and manipulation, data integrity and consistency
CIT Categories: 3.50, 3.70, 4.30, 4.34

1. INTRODUCTION

The logical view of data has been an important issue in recent years. Three major data models have been proposed: the network model [2, 3, 7], the relational model [8], and the entity set model [25]. These models have their own strengths and weaknesses. The network model provides a more natural view of data by separating entities and relationships (to a certain extent), but its capability to achieve data independence has been challenged [8]. The relational model is based on relational theory and can achieve a high degree of data independence, but it may lose some important semantic information about the real world [12, 15, 23]. The entity set model, which is based on set theory, also achieves a high degree of data independence, but its viewing of values such as "3" or "red" may not be natural to some people [25].

This paper presents the entity-relationship model, which has most of the advantages of the above three models. The entity-relationship model adopts the more natural view that the real world consists of entities and relationships. It

Copyright © 1976, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery. A version of this paper was presented at the International Conference on Very Large Data Bases, Framingham, Mass., Sept. 22-24, 1975.

Author's address: Center for Information System Research, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139.

ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, Pages 9-36.

<https://dl.acm.org/doi/10.1145/320434.320440>

1974

IBM System R

- Desenvolvido pela IBM San Jose e serviu de base para o IBM SQL/DS, IBM DB2, Oracle e todos os SGBDS existentes; Foi o primeiro banco de dados a utilizar SQL, entretanto a IBM chamava de SEQUEL



COMPUTING PRACTICES

A History and Evaluation of System R

Donald D. Chamberlin
Morton M. Astrahan
Michael W. Blasgen
James N. Gray
W. Frank King
Bruce G. Lindsay
Raymond Lorie
James W. Mehli
Thomas G. Price
Franco Putzolu
Patricia Griffiths Selinger
Mario Schödl
Donald R. Slutz
Irving L. Traiger
Bradford W. Wade
Robert A. Yost

IBM Research Laboratory
San Jose, California

1. Introduction

Throughout the history of information storage in computers, one of the most readily observable trends has been the focus on data independence. C.J. Date [27] defined data independence as "immunity of applications to change in storage structure and access strategy." Modern database systems offer data independence by providing a high-level user interface through which users deal with the information contents of their data, rather than the various lists, pointers, arrays, lists, etc. which are used to represent that information. The system assumes responsibility for choosing an appropriate internal

representation to cope without the aid of the user. This is the main reason that the user is not made aware of the change.

The relational data model was proposed by E.F. Codd [22] in 1970 as the next logical step in the trend toward data independence. Codd observed that conventional database systems store information in two ways: (1) by the contents of records stored in the database, and (2) by the ways in which these records are connected together. Different systems use various means for the connections among records, such as links, sets, chains, parents, etc. For example, in Figure 1(a), the fact that supplier Acme supplies bolts is represented by connections between the relevant part and supplier records. In such a system, a user frames a question, such as "What is the lowest price for bolts?", by writing a program which "navigates" through the maze of connections until it arrives at the answer to the question. The user of a "navigational" system has the burden (or opportunity) to specify exactly how the query is to be processed; the user's algorithm is thus embodied in a program which is dependent on the data structure that existed at the time the program was written.

Relational database systems, as proposed by Codd, have two important properties: (1) all information is

represented by the information; indeed, the representation of a given fact may change over time without users being aware of the change.

The relational data model was proposed by E.F. Codd [22] in 1970 as the next logical step in the trend toward data independence. Codd observed that conventional database systems store information in two ways: (1) by the contents of records stored in the database, and (2) by the ways in which these records are connected together. Different systems use various means for the connections among records, such as links, sets, chains, parents, etc. For example, in Figure 1(a), the fact that supplier Acme supplies bolts is represented by connections between the relevant part and supplier records. In such a system, a user frames a question, such as "What is the lowest price for bolts?", by writing a program which "navigates" through the maze of connections until it arrives at the answer to the question. The user of a "navigational" system has the burden (or opportunity) to specify exactly how the query is to be processed; the user's algorithm is thus embodied in a program which is dependent on the data structure that existed at the time the program was written.

Relational database systems, as proposed by Codd, have two important properties: (1) all information is

represented by the information; indeed, the representation of a given fact may change over time without users being aware of the change.

The relational data model was proposed by E.F. Codd [22] in 1970 as the next logical step in the trend toward data independence. Codd observed that conventional database systems store information in two ways: (1) by the contents of records stored in the database, and (2) by the ways in which these records are connected together. Different systems use various means for the connections among records, such as links, sets, chains, parents, etc. For example, in Figure 1(a), the fact that supplier Acme supplies bolts is represented by connections between the relevant part and supplier records. In such a system, a user frames a question, such as "What is the lowest price for bolts?", by writing a program which "navigates" through the maze of connections until it arrives at the answer to the question. The user of a "navigational" system has the burden (or opportunity) to specify exactly how the query is to be processed; the user's algorithm is thus embodied in a program which is dependent on the data structure that existed at the time the program was written.

Um documento que conta a história e avaliação do System R

1974

INGRES

- Desenvolvido pelos professores [Michael Stonebraker](#) e [Eugene Wong](#) UCB.
- O INGRES foi um concorrente direto do Sytem R da IBM. O mesmo só foi comercializado a partir da década de 80. Serviu como base para Sybase, PostgreSQL e MSSQL Server.
- Este sistema utilizava o QUEL como linguagem de consulta.
- Atualmente o INGRES pertence à ACTIAN e sua versão é a 11.0 (desde 2017).
-

INGRES

Disponível em:

<https://www.actian.com/data-management/actian-x-hybrid-rdbms/>

1978

ORACLE v2

- É o Primeiro Sistema Gerenciador de Banco de Dados da História Desenvolvido pela ORACLE CORPORATION;
- Sua primeira versão foi lançada em 1978 (44 anos) - É considerado o SGBDR com maior tempo de atuação.
- Utilizava uma linguagem baseada na IBM (SEQUEL)
-



1979 - 1983



Imagem do Oracle V4

Disponível em:
<https://www.oracle.com/>



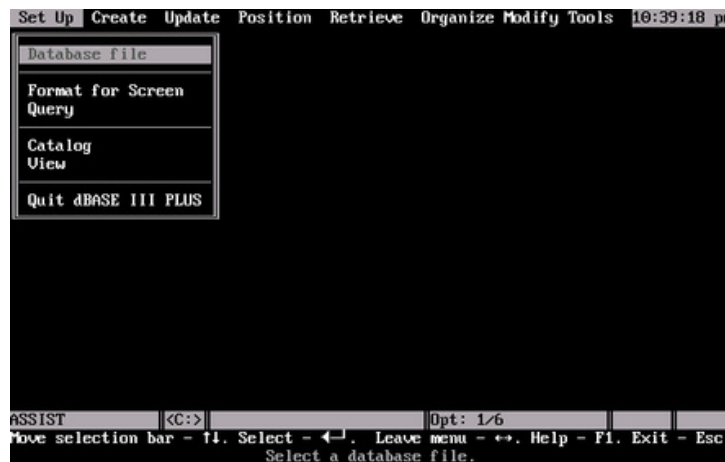
1979

dBASE I

- É um Banco de dados que implementa nativamente (embutido)
- Desenvolvido pela [Wayne Ratliff](#); É considerado o
- segundo SGBD com maior tempo de atuação (43 anos).
- Utilizava uma linguagem baseada pela IBM (SEQUEL)



Disponível em:
<https://www.dbase.com/>



70-79

Alguns Pontos Chave sobre a DEC. 70

- *A criação do modelo relacional*
- *A documentação em banco de dados*
- *Surgimento dos primeiros Bancos de Dados voltados à Indústria*
- *Criação do QUEL e SEQUEL, que serviram como base ao SQL*



1980

SISTEMA GERENCIADOR DE BANCO DE DADOS RELACIONAL (SGBDR)

- **Relational Data Base Management System (RDBMS)** ou Sistema Gerenciador de Banco de Dados Relacional (SGBDR) tem o princípio em Gerenciar N Banco de Dados dentro uma infraestrutura/negócio.
- O SGBDR tem como características:
 - Manipulação de dados; Definição de estruturas de dados (Tabelas e Colunas); Segurança contra falhas e acessos não autorizados;
 - Controle de uso compartilhado dos dados por diversos usuários;
 - Manter a integridade dos dados (Atender as características do ACID)
 -

Antes do SGBDR



Problema

- Equipamento ^S exclusivo para execução do **software** e armazenamento de dados;
- Apenas uma pessoa de cada vez poderia acessar;
- Se o equipamento danificasse, a base de dados era comprometida;
- Dados Isolados;

1986

Linguagem Estruturada de Consulta (SQL)

- *Structured Query Language (SQL) ou Linguagem Estruturada de Consulta se torna um padrão mundial.*
- *Os modelos em rede e hierárquico passam a ficar em segundo plano praticamente sem desenvolvimentos utilizando seus conceitos, porém vários sistemas legados continuam em uso;*
- *Embora o SQL tenha sido originalmente criado pela IBM, houve-se a necessidade de ser criado e adaptado um padrão para a linguagem.*
- *Esta tarefa foi realizada pela American National Standards Institute (ANSI) em 1986 e International Organization for Standardization (ISO) em 1987. Portanto, para cada versão lançada é se utilizado sql[ano]. exemplo: sql-92 (de 1992), sql-3 (de 2003), sql-16 (de 2016)*



1988

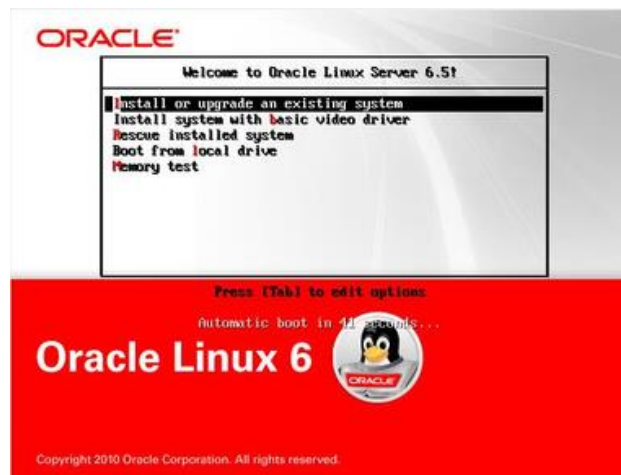
ORACLE v6

- É o Primeiro Sistema Gerenciador de Banco de Dados da História a utilizar oficialmente SQL definido pela ANSI e ISO; Foi a primeira versão a oferecer suporte escalabilidade, backup e recuperação online. Foi a primeira versão a
- oferecer PL/SQL (Procedures, Triggers, Functions em SQL). Atualmente se encontra na versão ORACLE 19c (2019)
-
-

ORACLE
ORACLE CORPORATION

1983 -1995

Disponível em:
<https://www.oracle.com/>





80-89

Alguns Pontos Chave sobre a DEC. 80

- *O Surgimento dos primeiros SGBDs*
- *A criação do SQL pela IBM*
- *A Padronização do SQL pela ANSI e a ISO (SQL-86 e SQL-89)*
- *A evolução da computação na industria e a necessidade de armazenar informação centralizada*



1990

Crescimento da Internet

- Através do protocolo World Wide Web (WWW) permitiu a distribuição de documentos hipertexto (HTTP) acessíveis a navegador web à Internet.
- Diante desse novo cenário, começou a surgir novos nichos/categorias de comércio. Denominado de Comércio Eletrônico ou Digital, onde usuários poderiam comprar e usufruir produtos à distância;
- Em razão dessa alta demanda, houve-se a necessidade de aperfeiçoar os Servidores de Banco de Dados, assim como o surgimento de novos **softwares** para SGBDR, que permiti-se o funcionamento de uma aplicação WEB por 24x7 (24h por dia, 7 dias da semana)



Netscape (Navegador popular na época)



Internet Explorer da Microsoft (Navegador popular na época)

1990

Microsoft SQL Server

- SGDBR; Desenvolvido pela Microsoft e Sybase (inicialmente);
- Multiplataforma (Linux, Windows etc);
- Sua primeira versão foi lançada em 1989 no OS/2 e apenas em 1993 no Windows
-
- A cada 2 anos é lançado uma versão do SQL Server. No momento se encontra na versão 16.0, sendo a 15.0 como a versão estável e 10.25 como a versão mínima

The logo for Microsoft SQL Server from 1990 to 2004. It features the word "Microsoft" in a small, sans-serif font above the words "SQL Server" in a larger, bold, sans-serif font.

1990 - 2004

The logo for Microsoft SQL Server from 2005 to 2008. It features the word "Microsoft" in a small, sans-serif font above the words "SQL Server" in a larger, bold, sans-serif font. The "S" in "SQL" is stylized with a small registered trademark symbol.

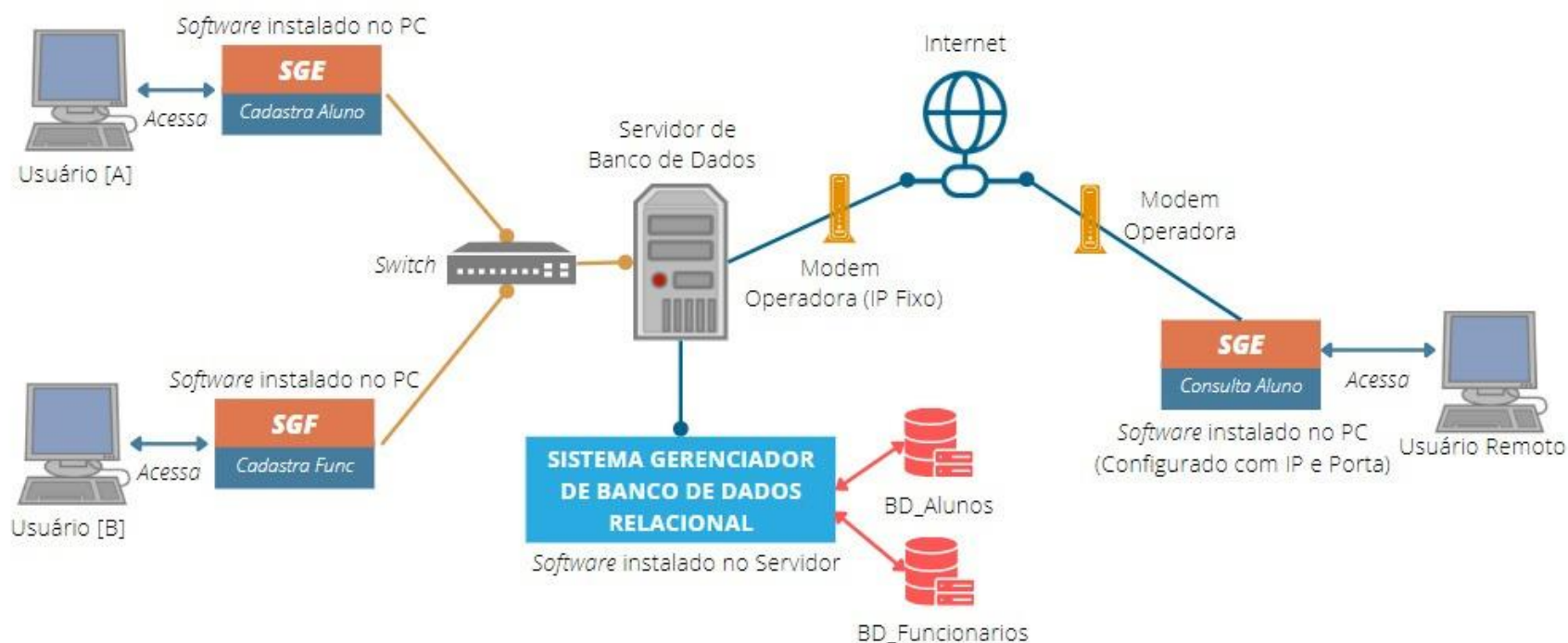
2005 - 2008

The logo for Microsoft SQL Server from 2009 to the present. It features a stylized red and grey icon on the left, followed by the word "Microsoft" in a small, sans-serif font above the words "SQL Server" in a larger, bold, sans-serif font. The "S" in "SQL" is stylized with a small registered trademark symbol.

2009 - Atualmente

Disponível em: <https://www.microsoft.com/pt-br/sql-server/sql-server-2019>

SGBDR depois da Internet Comercial



Esse diagrama é baseado sobre os softwares em desktop (com ou sem interface gráfica) que eram muito populares na década de 90

1993

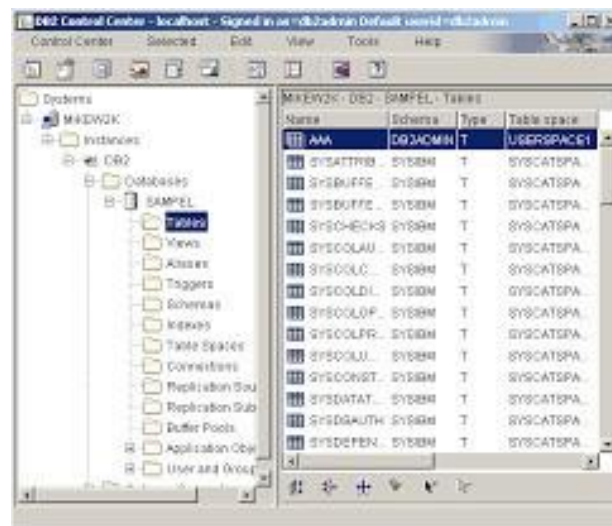
IBM Db2

- SGDBR; Lançado em 1983, mas apenas em versão 3.4 foi possível utilizar SQL. Desenvolvido pela IBM; Foi o
- Sucessor do **System R**. Atualmente o Db2 se encontra em versão 11.5 e possui um
- grande suporte no serviço de nível da IBM
-
-



Disponível em:

<https://www.ibm.com/products/db2>



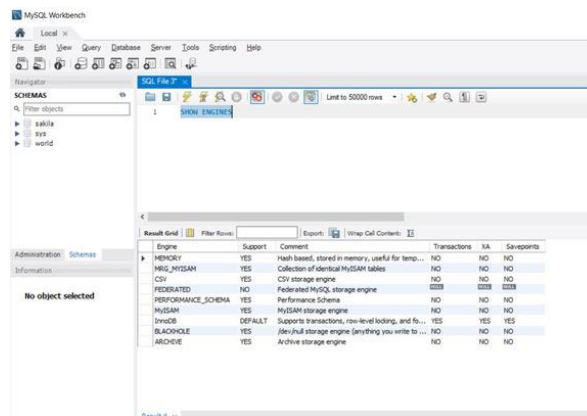
1995

MySQL

- SGDBR; Desenvolvido pela Oracle;
- Multiplataforma (Linux, macOS, Windows etc);
- Sua primeira versão foi lançada em 23 de maio de 1995 (27 anos)
- Atualmente se encontram a versão 8.0 com suporte a 28 idiomas



1995 -Atualmente



Interface do Workbench do MySQL

1996

PostgreSQL

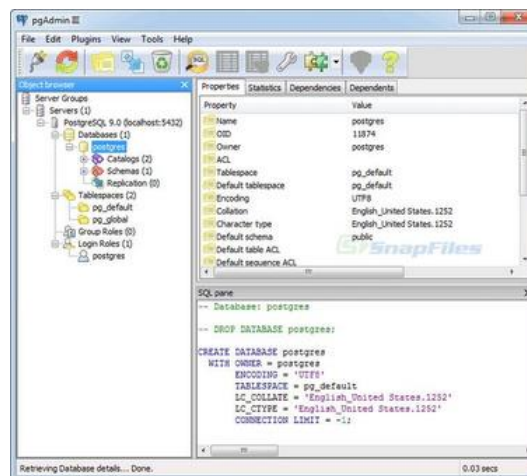
SGDBR de Código Aberto; Desenvolvido pela Postgres Multiplataforma;

Sua primeira versão foi lançada em 8 de julho de 1996 (26 anos)

Atualmente se encontra na versão 15.1 com suporte a 28 idiomas



1996 - Atualmente



Interface do pgAdmin do PostgreSQL

90-99

Alguns Pontos Chave sobre a DEC. 90

- *Surgimento de novos SGBDs, inclusive com Interface Gráfica ao Usuário*
- *O nascimento da Internet comercial e as aplicações WEB com a necessidade de armazenar informações - Os primeiros SGBD à Internet*
- *A Maturidade sobre o conceito Modelo Relacional*
- *Surgimento de novos conceitos (Data Warehouse, WIIS, GIS, Data Mining)*
- *A evolução da computação na indústria e a necessidade de armazenar informação de maneira centralizada e normalizada (pensando na integridade e a qualidade dos dados)*
- *O Surgimento dos Bancos de Dados Não Relacional (No-SQL)*



2000

SQLite**e**

- É uma Biblioteca que implementa um Banco de dados SQL nativamente (embutido);
- Desenvolvido por Richard Hipp; Voltado para aplicações em
- em dispositivos de sistemas embarcados;
- Sua primeira versão foi lançada em 17 de agosto de 2000 (17 anos) e encontra-se na versão 3.40
-



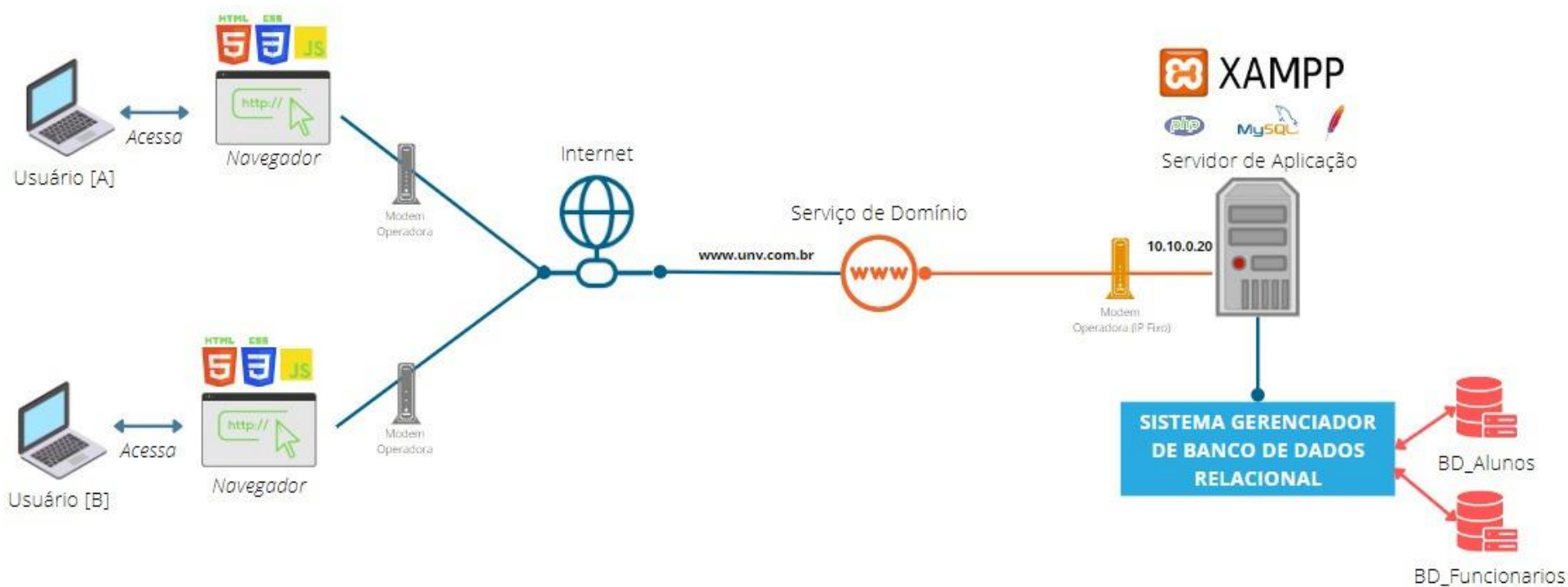
2000 - Atualmente

Disponível em:

<https://www.sqlite.org/index.html>



SGBDR depois dos Aplicativos Web



Esse diagrama é baseado sobre os aplicativos web (dinâmicos) que utilizavam o Navegador do Usuário (CSS, JavaScript e HTML) que recebiam informações de um Servidor de Aplicativos (XAMPP com Apache, MySQL e PHP)

2006

ARMAZENAMENTO EM NÚVEM

- *O Armazenamento em nuvem é a tecnologia que permite que governos, indústrias, Universidade ou usuários comúensem armazenar, mantereacessardado sem servidoresdealtadisponibilidade via internet.*
- *Em 2006, a amazon foi primeira empresa a lançar o serviço a núvem para empresas, denominado de Amazon WebServices(AWS)*



2006 -2010



Atualmente

2006

BANCO DE DADOS EM NÚVEM

- Banco de dados em nuvem é um [base de dados](#) que normalmente é executado em um [computação em nuvem](#) plataforma e o acesso ao banco de dados é fornecido como um serviço, que é denominado Banco de dados como serviço (DBaaS).



Existe desde 2006

Oferece [Amazon RDS](#) como DBaaS



Google Cloud

Existe desde 2008

Oferece [Firebase](#) como DBaaS



Existe desde 2010

Oferece [Azure SQL Database](#) como DBaaS



IBM Cloud

Existe desde 2011

Oferece [Db2](#) como DBaaS



Existe desde 2016

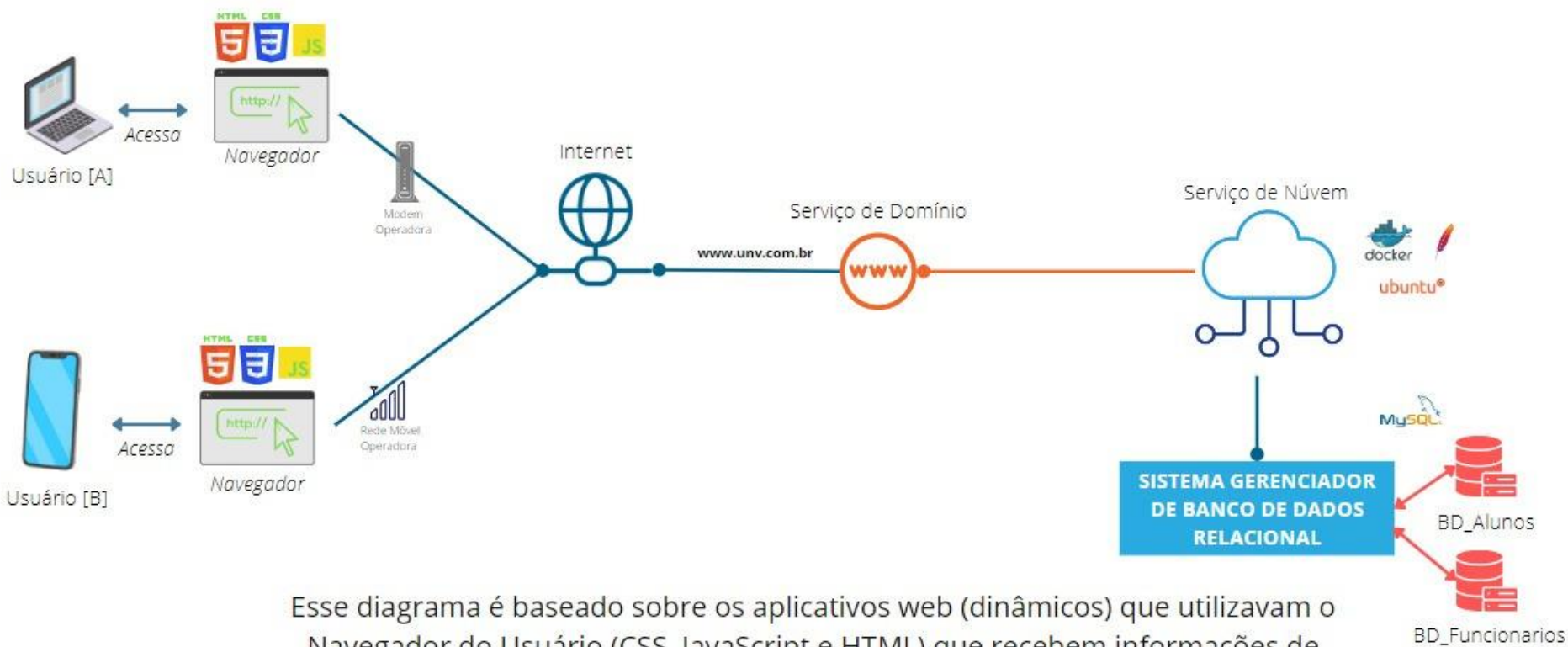
Oferece [ORACLE](#) como DBaaS

00-10

Alguns Pontos Chave sobre a DEC. 00

- *Surgimento dos primeiros SGBDnR*
- *O nascimento dos serviços de nuvem*
- *A Maturidade sobre o conceito Modelo Relacional e Não Relacional*
- *Surgimento do DBaaS*
- *A evolução da computação na indústria e a necessidade de armazenar informação de maneira centralizada e desnormalizada (pensando no empenho)*

SGBDR depois do Cloud



Esse diagrama é baseado sobre os aplicativos web (dinâmicos) que utilizavam o Navegador do Usuário (CSS, JavaScript e HTML) que recebem informações de um Servidor de Aplicativos em um Cloud

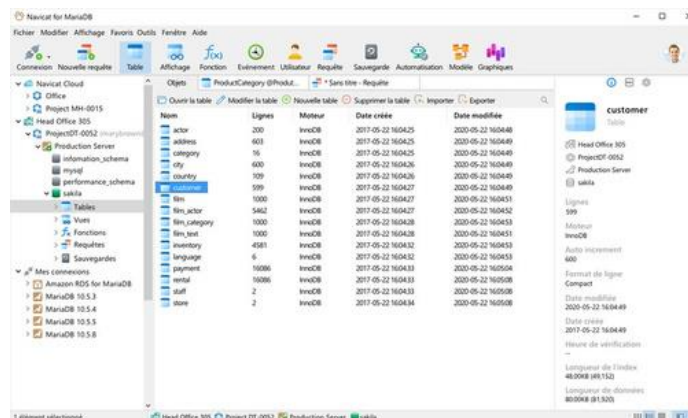
2009

MariaDB

- SGDBR; Desenvolvido pelo fundador do MySQL Multiplataforma (Linux, macOS, Windows e etc); Sua
- primeira versão foi lançada em 22 de Janeiro de 2009 (13 anos) MariaDB é considerado um **fork** do
- MySQL, tendo as mesmas funcionalidades e características. O MariaDB é considerado um substituto
- natural (ou sucessor) do MySQL
-
-



2009- Atualmente



In t
erface





ORACLE 19c

19^c ORACLE[®] Database

ORACLE Live SQL

Feedback Help glebot@hotmail.com

SQL Worksheet

Clear Actions Save Run

```
1 select * from v$version;
```

BANNER	BANNER_FULL	BANNER_LEGACY	CON_ID
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production	Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.2.0.0.0	Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production	0

[Download CSV](#)

© 2019 Oracle Corporation - [Privacy](#) - [Terms of Use](#)

ORACLE Integrated Cloud Applications & Platform Services

[Oracle Learning Library](#) - [Oracle Database Documentation 19c, 18c, 12c](#) - [Follow on Twitter](#)

Live SQL 19.1.2, running Oracle Database 19c Enterprise Edition - 19.2.0.0.0

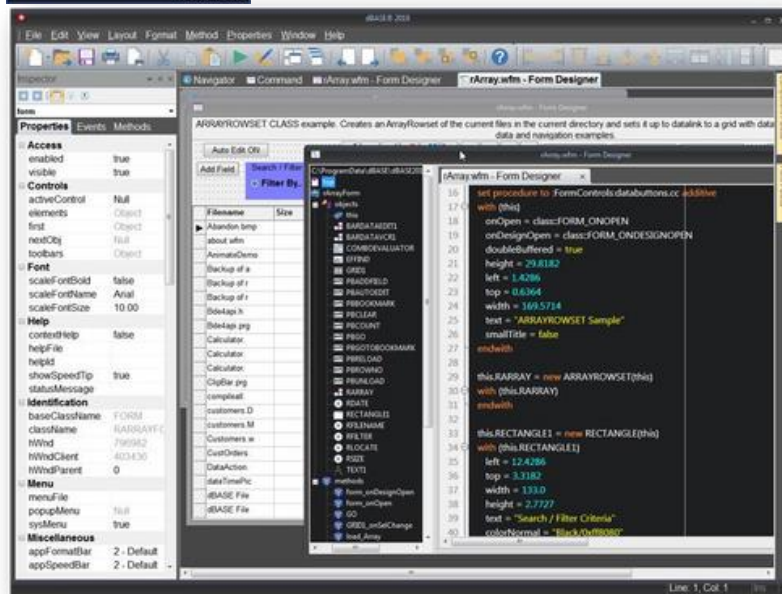
Built with ❤ using Oracle APEX

Disponível em:
<https://www.oracle.com/>



2019

dBASE 2019

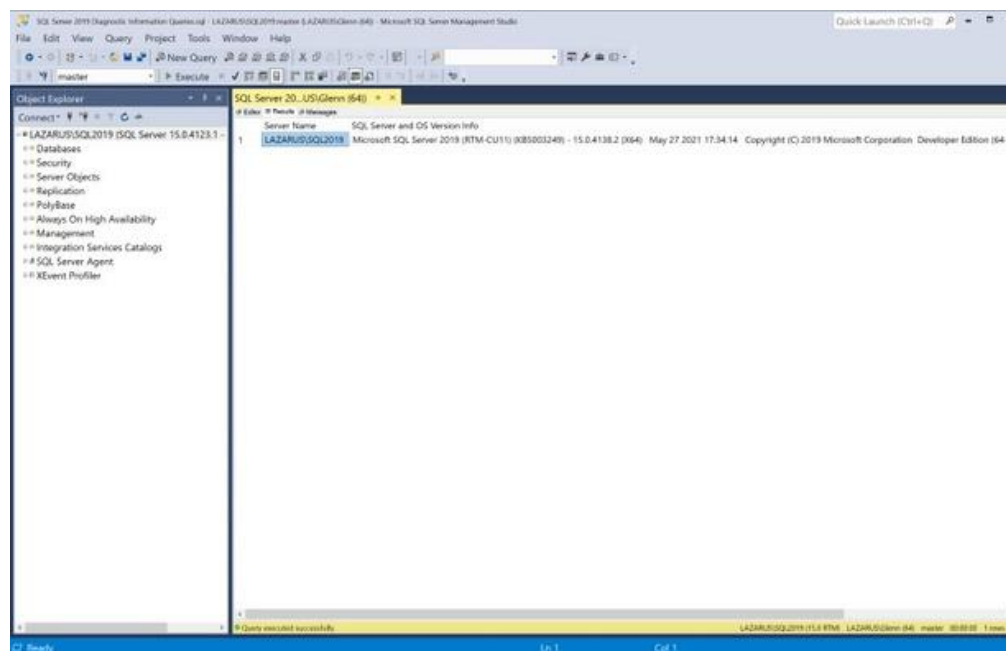
**dBASE® 2019
Optimized
Edition**


Disponível em:
<https://www.dbase.com/>



2019

Microsoft SQL Server 2019



Disponível em: <https://www.microsoft.com/pt-br/sql-server/sql-server-2019>

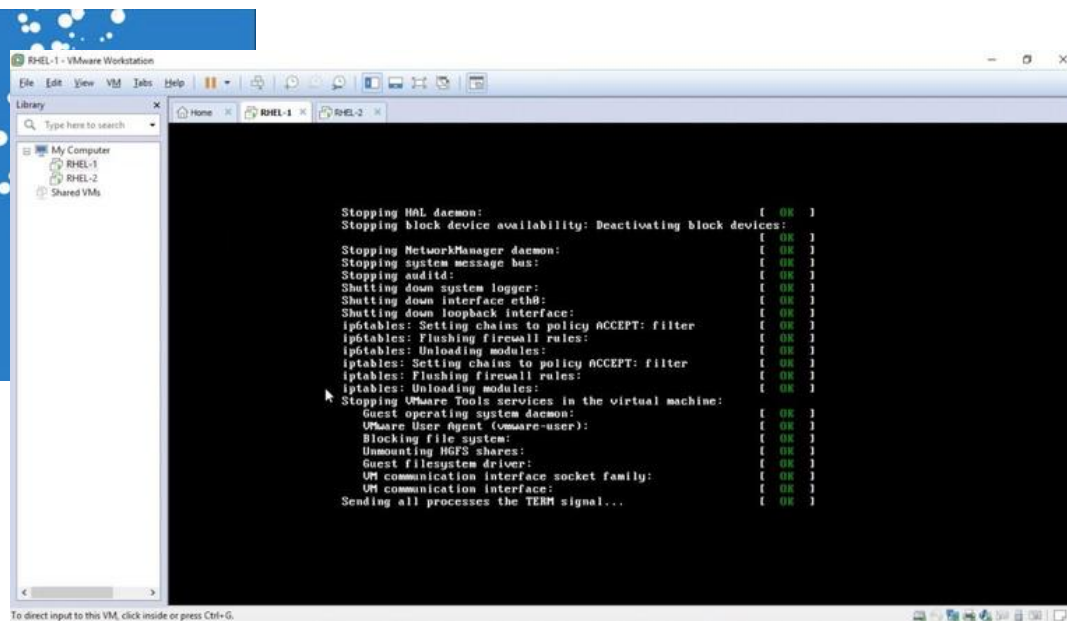


2020

IBM Db2 v11.5

IBM Db2 V11.5.4 Edition updates

Roger Bitar
Offering Manager, Db2

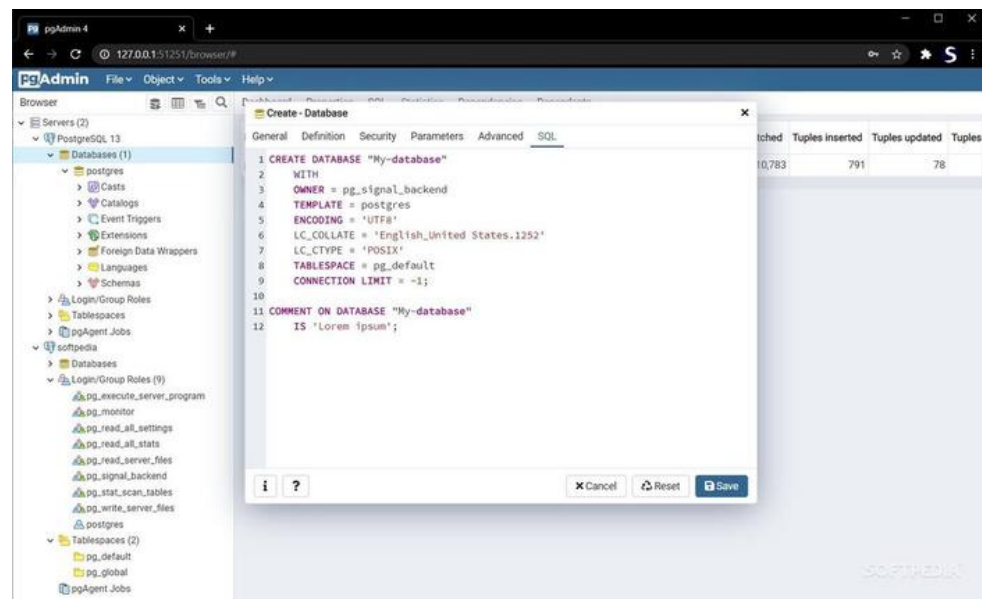


Disponível em:

<https://www.ibm.com/products/db2>

2022

PostgreSQL 15.1

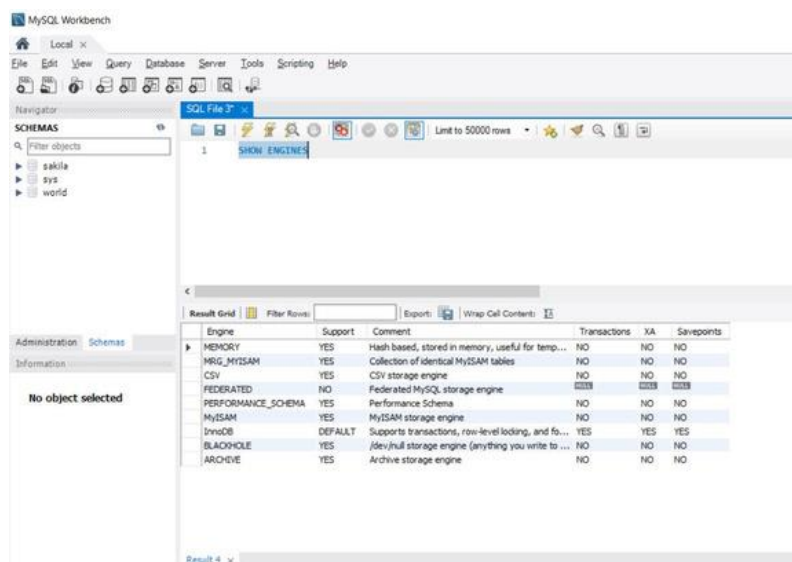


Disponível em:
<https://www.postgresql.org/>



2022

MySQL 8.0



Disponível em: <https://www.mysql.com/>

2023

Atualmente [...]

- *Atualmente existem vários modelos de bancos de dados tais como, orientado a objetos, orientado a documentos, orientado a grafos, ETC . Porém, o mais comum ainda é o banco de dados relacional.*
- *A decisão entre qual modelo de banco de dados utilizar baseia-se no tipo de dados que você pretende armazenar.*
- *Data mining (Mineração de dados), data warehouse (Armazém de dados), data analysis (Análise de dados) são técnicas utilizadas atualmente e no futuro serão utilizados cada vez mais.*



ELMASRI, R.; NAVATHE, S.B. Sistemas de Banco de Dados: Fundamentos e Aplicações. 6ª. Edição -Pearson, 2011.

SILBERSCHATZ, A; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. 6qa. Edição -Campus, 2012.

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibida sem o consentimento formal, por escrito, do Professor (autor).