

SQL FOR DATA SCIENCE

INFNET – Bootcamp Data Science, Data Analytics & Machine Learning

SQL (STRUCTURED QUERY LANGUAGE)

- O que é (Overview)
 - Structured Query Language
 - Standard -> RDBMS (Relational Database Management Systems)
 - RDBMS -> gerencia dados como uma coleção de tabelas em que todos os relacionamentos são representados por valores comuns em tabelas relacionadas
- Motivação
 - Um dos requisitos mais exigidos de um Cientista de Dados
- Objetivos
 - Aplicações
 - Benefícios
- SQL (ambiente)
- SQL (Data Types)

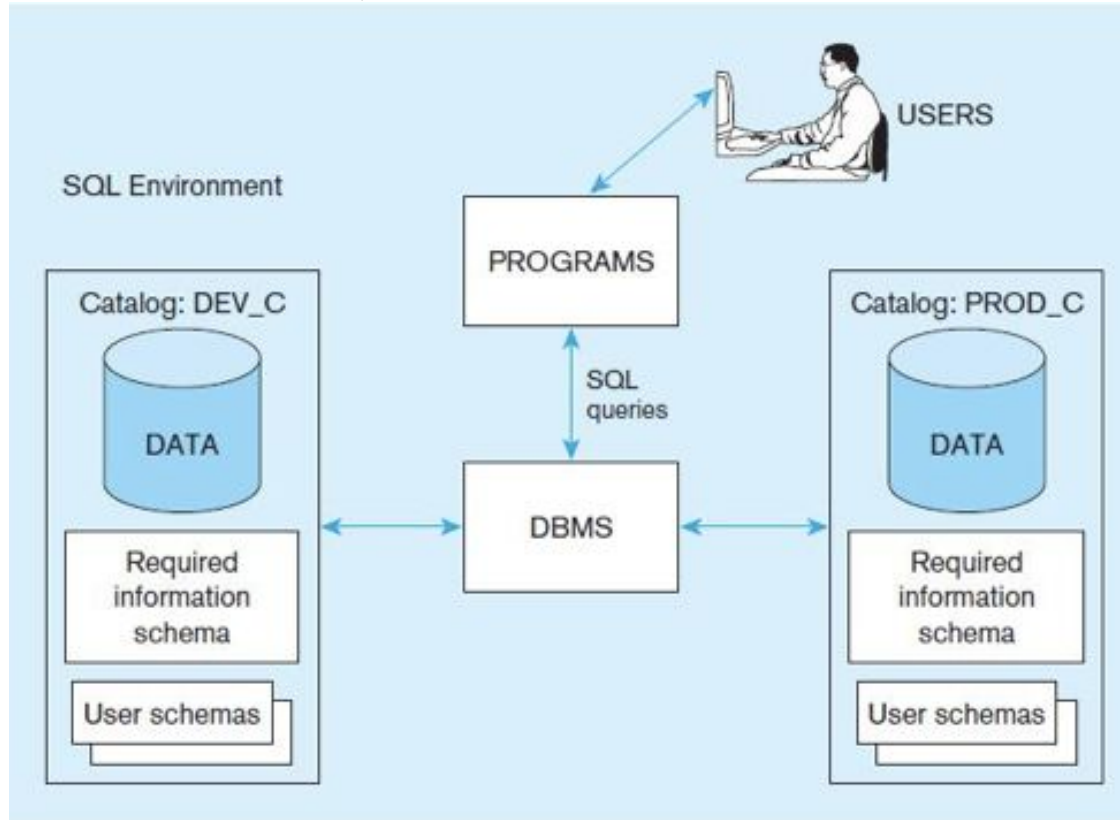
SQL - OBJETIVOS E BENEFÍCIOS

- Útil para gerenciar grande quantidade de dados - Big Data
- Acesso fácil e seguro
- Produtividade
- Portabilidade
- Longevidade

SQL (ENVIRONMENT)

- Catalog
 - A set of schemas that constitute the description of a database
- Schema
 - The structure that contains descriptions of objects created by a user (base tables, views, constraints)
- Data Definition Language (DDL)
 - Commands that define a database, including creating, altering, and dropping tables and establishing constraints
- Data Manipulation Language (DML)
 - Commands that maintain and query a database
- Data Control Language (DCL)
 - Commands that control a database, including administering privileges and committing data

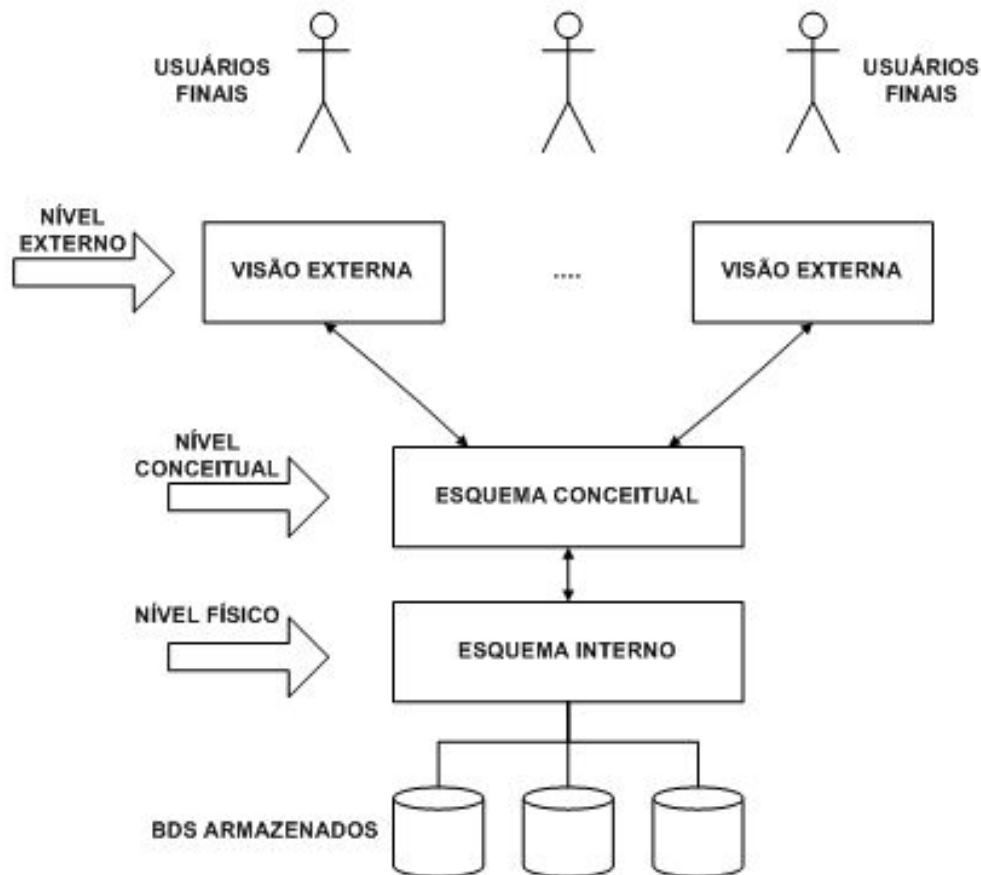
SQL (ENVIRONMENT)



SQL (RELATIONAL MODEL)

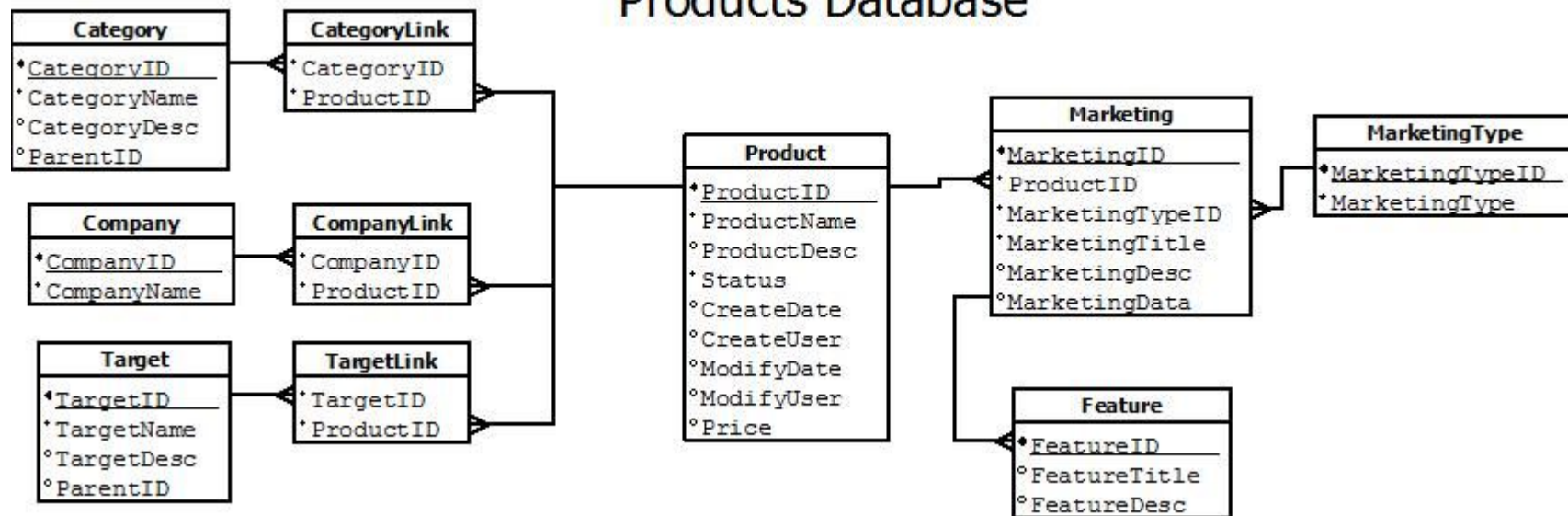
- Modelo amplamente usado
- Dados são representados em estruturas simples
- Tabelas
 - Independência lógica dos dados
 - Independência física dos dados
 - Armazenamento independente dos dados
- ERD(Entity Relationship Diagrams)
 - Tables x Relationships
 - Entities and attributes

SQL



SQL(EXAMPLE)

Products Database



SQL (ENTITIES AND ATTRIBUTES)

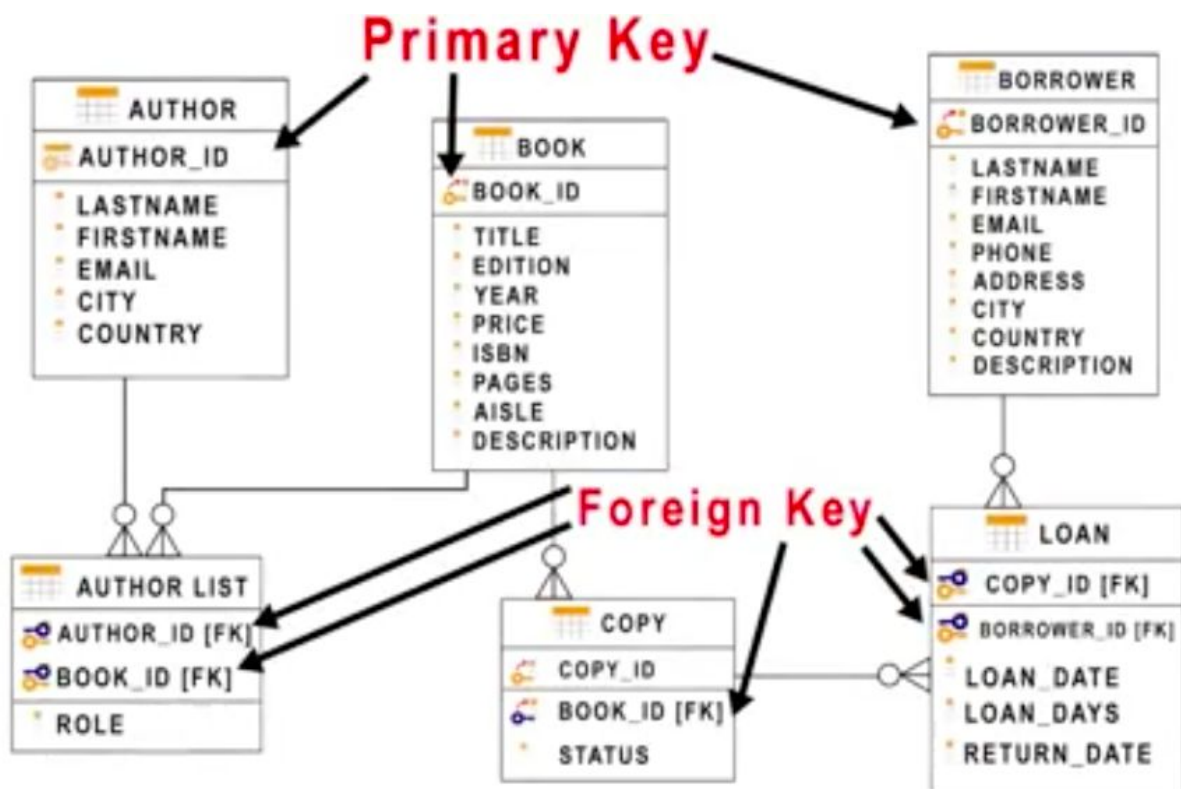


SQL (TABLES (ENTITIES) ; COLUMNS(ATTRIBUTES))

Table: Book

Book_id	Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
0	SQL for Beginners	1	2021	39.99	00-123-457-800	150	DB-A02	Teaches you basics fundamentals
1	Advanced topics in SQL	2	2018	49.99	01-345-567-810	221	DB-A03	

SQL



SQL (DDL - CREATE)

- Identify data types for attributes
- Identify columns that can and cannot be null
- Identify columns that must be unique (candidate keys)
- Identify primary key–foreign key mates
- Determine default value
- Identify constraints on columns (domain specifications)
- Create the table and associated indexes

SQL - CREATE SYNTAX

```
CREATE TABLE tablename
( {column definition [table constraint] } . . .
[ON COMMIT {DELETE | PRESERVE} ROWS] );

where column definition ::=
column_name
    {domain name | datatype [(size)] }
    [column_constraint_clause . . .]
    [default value]
    [collate clause]

and table constraint ::=
    [CONSTRAINT constraint_name]
    Constraint_type [constraint_attributes]
```

SQL- CREATE TABLES

```
create table Book (  
    Book_Id int( 6 ) NOT NULL --PRIMARY KEY---,  
    Title varchar2 ( 100 ) NOT NULL,  
    Edition int ( 2 ),  
    Year year,  
    Price real,  
    ISBN_A varchar2 ( 20 ) UNIQUE,  
    Pages int (4),  
    xxxxxxx  
    xxxxxxxx,  
    *** PRIMARY KEY (Book_Id) ***  
    constraint pk_Book primary key (Book_Id,Title),  
);
```

SQL - INSERT

INSERT INTO **TABLE** VALUES

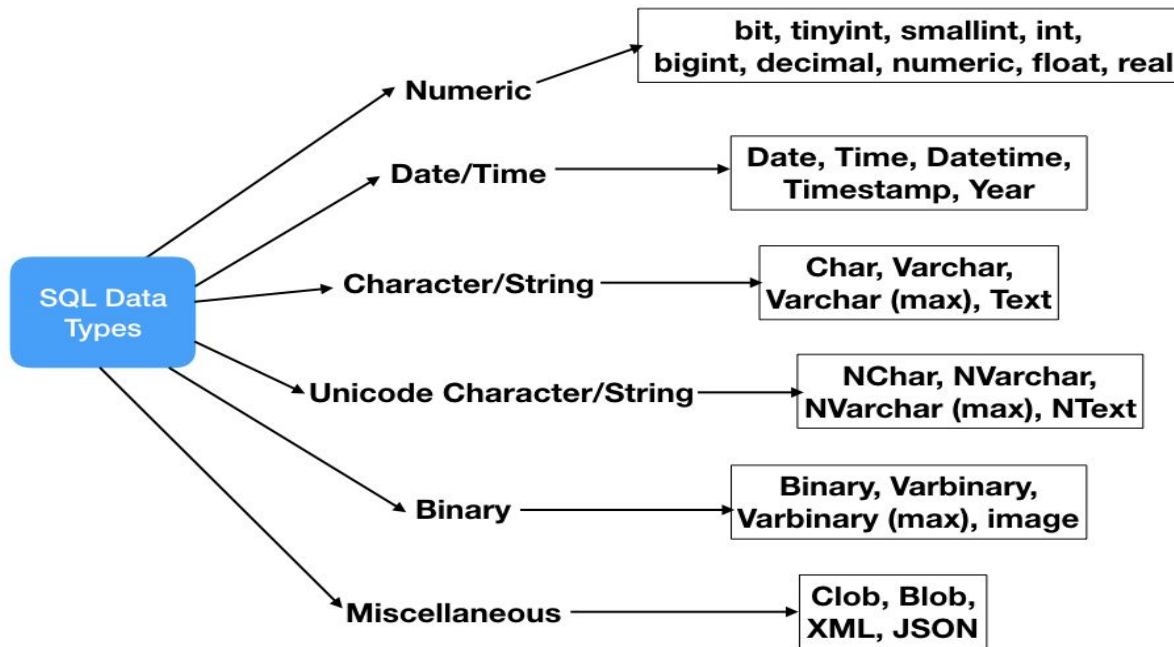
```
INSERT INTO BOOK VALUES (0, 'SQL for  
Beginners',1,2021,39.99,'00-123-457-800',150,'DB-A02','Teach  
es you basics fundamentals')
```

```
INSERT INTO BOOK(Book_Id,Title,ISBN,Pages) VALUES (0,'SQL  
for Beginners','00-123-457-800',150)
```

COMMIT !!!!!

```
(SELECT * FROM TABLE)
```

SQL (DATA TYPES)



SQL (EXAMPLE)

Table: Project




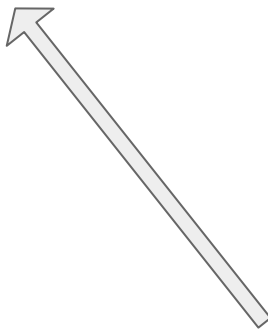
 ID (Primary_Key)
NAME
BEGIN_DATE
END_DATE

Table: Task

ID (Primary_Key) 
NAME
PRIORITY
STATUS_ID
PROJECT_ID (Foreign_Key) 
BEGIN_DATE
END_DATE



SQL

```
CREATE TABLE projects (id integer PRIMARY KEY,  
                           name text NOT NULL,  
                           begin_date text,  
                           end_date text  
                           );
```

ID (Primary_Key)
NAME
BEGIN_DATE
END_DATE

SQL

```
CREATE TABLE IF NOT EXISTS tasks (id integer PRIMARY KEY,
```

```
name text NOT NULL,
```

```
priority integer,
```

```
status_id integer NOT NULL,
```

```
project_id integer NOT NULL REFERENCES projects (id),
```

```
begin_date text NOT NULL,
```

```
end_date text NOT NULL)
```

```
#FOREIGN KEY (project_id) REFERENCES projects (id)
```

ID (Primary_Key)
NAME
PRIORITY
STATUS_ID
PROJECT_ID (Foreign_Key)
BEGIN_DATE
END_DATE

SQL - O QUE TRABALHAREMOS HOJE

- 1) INSERT - a partir de uma lista
- 2) SELECT - Filtrando por atributos
- 3) SELECT WHERE - Subset da Tabela
- 4) SELECT COUNT
- 5) SELECT DISTINCT
- 6) SELECT (CONDIÇÃO LÓGICA)
- 7) SELECT LIMIT
- 8) SELECT PATTERNS
- 9) SELECT ORDER BY

REFERÊNCIA - SQL COOKBOOK, 2nd Edition

SQL - TAREFAS DE HOJE

- 1) Carregar os dados dos arquivos nas respectivas tabelas
- 2) Quantos estudantes obtiveram média acima de 83.5 ?
- 3) Faça uma consulta que retorne os cursos dados ordenados por ordem alfabética.
- 4) Faça uma consulta que retorne o nome de todas as pessoas cujo o último nome termina com "ez".
- 5) Quantos estudantes precisaram de 3 tentativas para serem aprovados
- 6) Usando uma lógica condicional retorne : Id do aluno, nota, aprovado ou reprovado. Aprovado se nota ≥ 70.0

Dica: Fazer conforme o exemplo ### Using Conditional Logic in a SELECT Statement #### do notebook

SQL - PARA AMANHÃ

- 1) Estudar os conceitos dados hoje e realizar as tarefas do Laboratório
- 2) Estudar os capítulos 1,2,3 e 4 do Livro (CookBook)
- 3) Conceitos a serem trabalhados :
 - a) SELECT ORDER BY
 - b) SELECT LIKE
 - c) SELECT USING MULTIPLE JOINS