

PARA
SER,
UTP.

utp.edu.br

DDL

Data definition language

PARA SER, UTP.



Universidade
Tuiuti do
Paraná

Criar um banco de dados

Criar banco de dados

```
CREATE DATABASE Aviacao
```

Selecionar o banco de dados

```
USE Aviacao
```

PARA SER, UTP.



Universidade
Tuiuti do
Paraná

Normalização

- **1ª Forma Normal**

- A tabela não deve conter grupos repetidos e nem atributos com mais de um valor.

- De:

- $\text{PESSOAS} = \{ \underline{\text{ID}} + \text{NOME} + \text{ENDERECO} + \text{TELEFONES} \}$

- Para:

- $\text{PESSOAS} = \{ \text{ID} + \text{NOME} + \text{ENDERECO} \}$

- $\text{TELEFONES} = \{ \text{PESSOA_ID} + \text{TELEFONE} \}$

PARA SER, UTP.



Universidade
Tuiuti do
Paraná

Normalização

- **2ª Forma Normal**

- Atributos não chaves da tabela devem depender unicamente da chave primária (não podendo depender apenas de parte dela).

- De:

- $ALUNOS_CURSOS = \{ \underline{ID_ALUNO} + \underline{ID_CURSO} + NOTA + DESCRICAO_CURSO \}$

- Para:

- $ALUNOS_CURSOS = \{ ID_ALUNO + ID_CURSO + NOTA \}$
- $CURSOS = \{ ID_CURSO + DESCRICAO \}$

PARA SER, UTP.



Universidade
Tuiuti do
Paraná

Normalização

- **3ª Forma Normal**

- Atributos não chave de uma tabela devem ser mutuamente independentes e dependentes unicamente e exclusivamente da chave primária.

- De:

- $\text{FUNCIONARIOS} = \{ \text{ID} + \text{NOME} + \text{ID_CARGO} + \text{DESCRICAO_CARGO} \}$

- Para:

- $\text{FUNCIONARIOS} = \{ \text{ID} + \text{NOME} + \text{ID_CARGO} \}$
- $\text{CARGOS} = \{ \text{ID_CARGO} + \text{DESCRICAO} \}$

PARA SER, UTP.



Universidade
Tuiuti do
Paraná

Schema

```
CREATE TABLE Pilots (  
    ID int PRIMARY KEY,  
    First_name nvarchar(100) NOT  
NULL,  
    Last_name nvarchar(100) NOT  
NULL,  
    Title nvarchar(100) NOT NULL,  
    Birth_date date NOT NULL  
);
```

```
CREATE TABLE Planes (  
    ID int PRIMARY KEY,  
    Model nvarchar(100) NOT NULL,  
    Built_date date NOT NULL,  
    Number_of_seats int NOT NULL  
);
```

```
CREATE TABLE Flights (  
    ID int PRIMARY KEY,  
    [From] nvarchar(100) NOT NULL,  
    [To] nvarchar(100) NOT NULL,  
    Departure_date date NOT NULL,  
    Planes_ID int NOT NULL,  
    constraint Planes_fk foreign key (Planes_ID) references  
Planes(ID)  
);
```

Schema

```
CREATE TABLE Pilots_on_flights  (  
    Flights_ID int NOT NULL,  
    Pilots_ID int NOT NULL,  
    constraint Pilots_on_flights_pk primary key (Flights_ID,  
    Pilots_ID),  
    constraint Flightsfk foreign key (Flights_ID) references  
    Flights(ID),  
    constraint Pilots_fk foreign key (Pilots_ID) references  
    Pilots(ID)  
);
```

Schema Browser

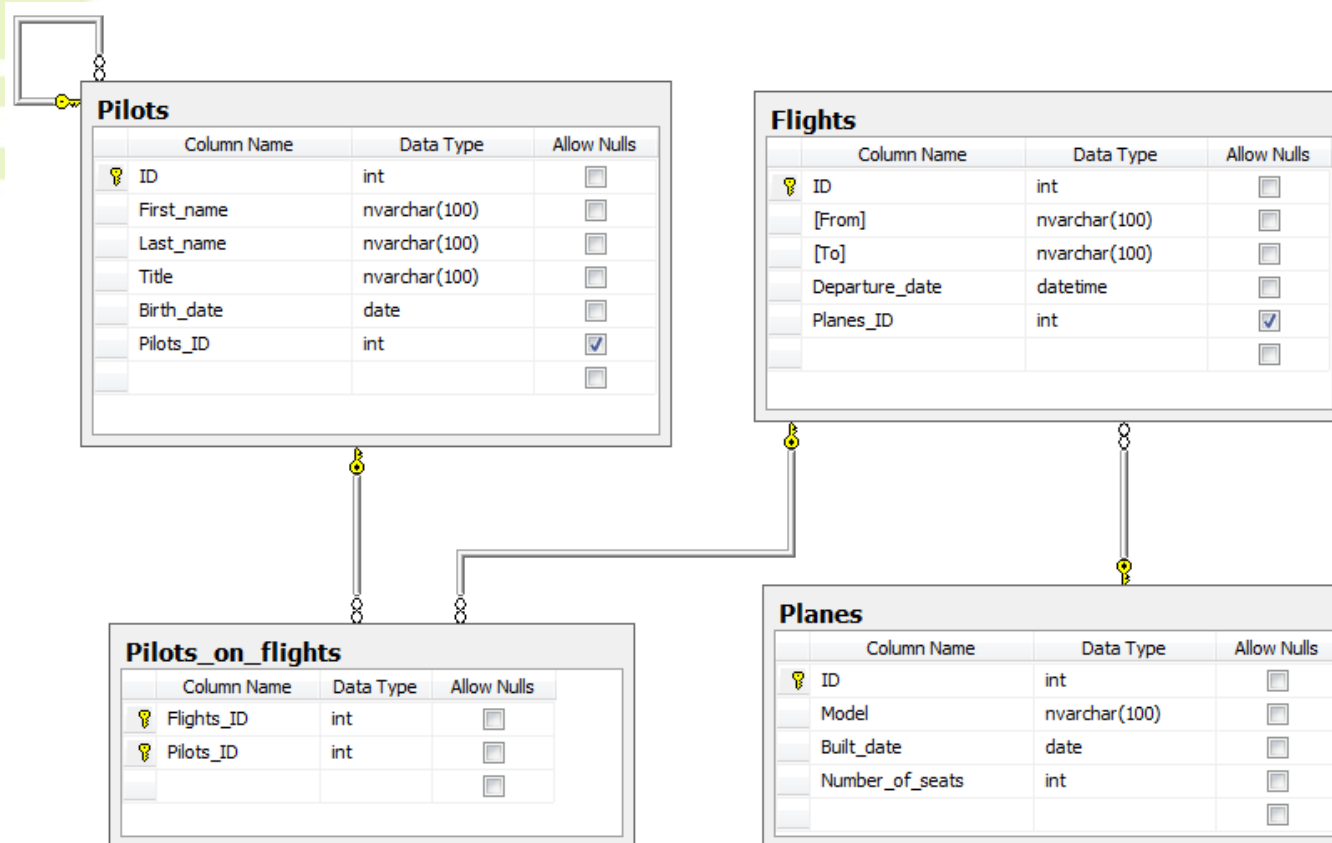
- + [Flights](#) (TABLE)
- + [Pilots](#) (TABLE)
- + [Pilots_on_flights](#) (TABLE)
- + [Planes](#) (TABLE)

PARA SER, UTP.



**Universidade
Tuiuti do
Paraná**

Esquema banco de dados



PARA SER, UTP.



Universidade
Tuiuti do
Paraná

Chave Primária e Estrangeira

- PessoaID int NOT NULL PRIMARY KEY
- PessoaID int FOREIGN KEY
REFERENCES Pessoas(PessoaID)

PARA SER, UTP.



Universidade
Tuiuti do
Paraná

Chave Primária e Estrangeira

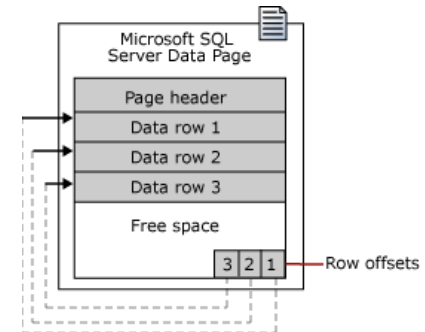
- ALTER TABLE Pedidos ADD FOREIGN KEY (PessoaID) REFERENCES Pessoas(PessoaID);
- ALTER TABLE Pedidos DROP CONSTRAINT FK_PersonOrder;

PARA SER, UTP.

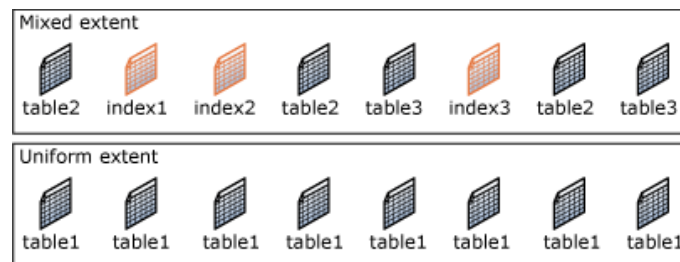


Universidade
Tuiuti do
Paraná

Pages e Extents



- Page
 - É a unidade fundamental de armazenamento de dados no Sql Server.
 - O tamanho da página é 8 Kb, e possuem um Header.
 - Todas as páginas são armazenadas em Extents.
- Extents
 - Unidade básica pela qual o espaço é gerenciado.
 - Um extents corresponde a 8 páginas contiguas ou 64Kb.



PARA SER, UTP.



Universidade
Tuiuti do
Paraná