

Instituto Federal Goiano - Campus Ceres  
Bacharelado em Sistemas de Informação  
Prof. Me. Ronneesley Moura Teles

Andrey Ribeiro  
Jeferson Rossini  
Wesley Moraes  
Marcos Arriel  
Jonathan Sousa

*Banco de Dados de anúncios*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>DER</b>	<b>2</b>
<b>3</b>	<b>Como funciona?</b>	<b>2</b>
3.1	anuncio.sql . . . . .	3
3.2	inserts.sql . . . . .	3
3.3	view.sql . . . . .	4

# Banco de Dados de anúncios

## 1 Introdução

Este artigo contém uma modelagem e uma estrutura de armazenamento de dados para um sistema de controle de resultados de uma campanha de publicidade. Com a construção de um sistema que gerência e controla todos os resultados de uma campanha de publicidade, tais como região em que deseja anunciar, público, quantidade de visualizações do anúncio ou cliques neles, formas de pagamento, crédito prévio para o anúncio, podendo ser adaptado para diversos outros projetos que necessitem de um controle e medição de resultados publicitários.

## 2 DER

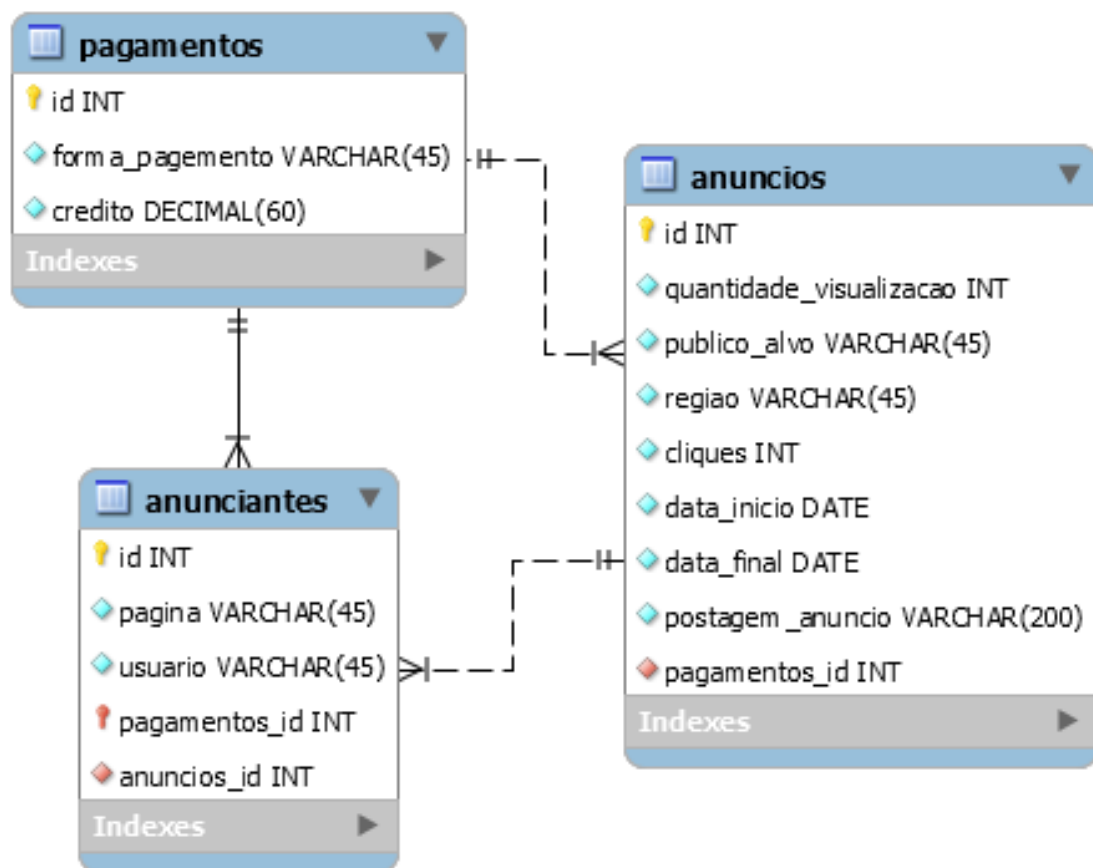


Figura 1: DER do Banco da Dados

## 3 Como funciona?

Cada anunciantes tem sua página e seu usuário, com isso ele pode criar um anuncio que conta a quantidade de visualizações, o publico alvo, a regiao, os cliques,

a data de início e fim e a postagem que foi anunciada, além disso conta com a forma de pagamento e o crédito prévio para propagandas.

### 3.1 anuncio.sql

SQL do Banco de Dados.

```
1 CREATE DATABASE anuncio;
2
3 USE anuncio;
4
5 CREATE TABLE pagamentos(
6     id INT NOT NULL AUTO.INCREMENT,
7     valor DECIMAL(60) NOT NULL,
8     forma_pagamento VARCHAR(45) NOT NULL,
9     credito DECIMAL(60),
10    PRIMARY KEY (id)
11 );
12
13 CREATE TABLE anuncios(
14     id INT NOT NULL AUTO.INCREMENT,
15     quantidade_visualizacao INT NOT NULL,
16     publico_alvo VARCHAR(45),
17     regioao VARCHAR(45) NOT NULL,
18     cliques INT NOT NULL,
19     data_inicio DATE NOT NULL,
20     data_final DATE NOT NULL,
21     postagem_anuncio VARCHAR(200) NOT NULL,
22     pagamentos_id INT NOT NULL,
23     PRIMARY KEY (id),
24     INDEX fk_anuncios_pagamentos_idx (pagamentos_id ASC),
25     CONSTRAINT fk_anuncios_pagamentos1
26     FOREIGN KEY (pagamentos_id)
27     REFERENCES pagamentos (id)
28 );
29
30 CREATE TABLE anunciantes(
31     id INT NOT NULL AUTO.INCREMENT,
32     pagina VARCHAR(45) NOT NULL,
33     usuario VARCHAR(45) NOT NULL,
34     pagamentos_id INT NOT NULL,
35     anuncios_id INT NOT NULL,
36     PRIMARY KEY (id, pagamentos_id),
37     INDEX fk_anunciantes_pagamentos_idx (pagamentos_id ASC),
38     INDEX fk_anunciantes_anuncios1_idx (anuncios_id ASC),
39     CONSTRAINT fk_anunciantes_pagamentos
40     FOREIGN KEY (pagamentos_id)
41     REFERENCES anuncio.pagamentos (id),
42     CONSTRAINT fk_anunciantes_anuncios1
43     FOREIGN KEY (anuncios_id)
44     REFERENCES anuncio.pagamentos (id)
45 );
```

recursos/anuncio.sql

### 3.2 inserts.sql

Inserts do Banco de Dados

```

1 INSERT INTO pagamentos(valor, forma_pagamento, credito)
2 VALUES (200, 'Boleto', 155),
3         (2500, 'Cartao de credito', 10),
4         (100, 'Boleto', 1),
5         (20, 'Cartao de credito', 0),
6         (340, 'Cartao de credito', 85),
7         (2050, 'Boleto', 155),
8         (400, 'Boleto', 103),
9         (60, 'Boleto', 122),
10        (2088, 'Boleto', 20),
11        (987, 'Cartao de credito', 845);
12
13
14
15 INSERT INTO anuncios(quantidade_visualizacao, publico_alvo, regioao,
16 cliques, data_inicio, data_final, postagem_anuncio, pagamentos_id)
17 VALUES (800, 'lojistas', 'centro-oeste', 740, '2017-11-25', '
18 2017-10-30', 'pao de batata', 1),
19        (8400, 'varejo', 'sul', 2740, '2017-01-25', '2017-11-30', '
20 churrasco', 2),
21        (100, 'americanas', 'norte', 140, '2017-02-25', '
22 2017-11-30', 'pao', 3),
23        (500, 'seu ze', 'centro-oeste', 240, '2017-03-25',
24 '2017-11-30', 'arroz', 4),
25        (400, 'atacado', 'centro-oeste', 440, '2017-04-25',
26 '2017-11-30', 'feijao', 5),
27        (200, 'lojistas', 'nordeste', 640, '2017-05-25', '
28 2017-11-30', 'carro', 6),
29        (600, 'lojistas', 'centro-oeste', 840, '2017-06-25'
30 , '2017-11-30', 'moto', 7),
31        (100, 'if goiano', 'sul', 40, '2017-07-25', '
32 2017-11-30', 'barco', 8),
33        (1800, 'lojistas', 'centro-oeste', 420, '017-08-25'
34 , '017-11-30', 'aviao', 9),
35        (50, 'lojistas', 'centro-oeste', 10, '2017-09-25',
36 '2017-10-30', 'pao de batata', 10);
37
38
39
40 INSERT INTO anunciantes(pagina, usuario, pagamentos_id, anuncios_id)
41 VALUES ('renova filmes', 'Andrey Silva', 1, 1),
42        ('conteudi', 'Jeferson Rosini', 2, 2),
43        ('if goiano', 'Adriano Braga', 3, 3),
44        ('sistemas', 'Ronne', 4, 4),
45        ('conteudo', 'Rossini Jeferson', 5, 5),
46        ('renova filmes', 'Andrey Silva', 6, 6),
47        ('renova filmes', 'Andrey Silva', 7, 7),
48        ('marcafrutas', 'Jose silva', 8, 8),
49        ('lojista', 'Thalia Santana', 9, 9),
50        ('renova filmes', 'Andrey Silva', 10, 10);

```

recursos/inserts.sql

### 3.3 view.sql

View utilizadas no Banco de Dados

```

1 — Visualizar o nome da pagina, o nome do usuario e quantidade de
   visualizacao do anuncio
2 CREATE VIEW visualizacao_anunciante AS

```

```

3 SELECT anunciantes.id, anunciantes.pagina, anunciantes.usuario,
   anuncios.quantidade_visualizacao FROM anunciantes
4 INNER JOIN anuncios ON anunciantes.anuncios_id = anuncios.id;
5
6 — Seleciona a quantidade de cliques por anuncio
7 CREATE VIEW cliques_anuncio AS
8 SELECT anuncios.cliques, anuncios.postagem_anuncio, anuncios.id FROM
   anuncios
9 INNER JOIN anunciantes ON anuncios_id = anunciantes.anuncios_id
10 GROUP BY anuncios.id;
11
12 — Seleciona o metodo a pagina e o metodo de pagamento de cada anuncio
13 CREATE VIEW pagamento_anuncio AS
14 SELECT anunciantes.id, anunciantes.pagina, anuncios.postagem_anuncio,
   forma_pagamento FROM anunciantes
15 INNER JOIN anuncios ON anunciantes.anuncios_id = anuncios.id
16 INNER JOIN pagamentos ON anuncios.pagamentos_id = pagamentos.id;
17
18 — Seleciona o nome da pagina e credito de anuncio de cada usuario
19 CREATE VIEW qntd_credito AS
20 SELECT anunciantes.id, anunciantes.pagina, anunciantes.usuario,
   anuncios.postagem_anuncio, pagamentos.credito FROM anunciantes
21 INNER JOIN anuncios ON anunciantes.anuncios_id = anuncios.id
22 INNER JOIN pagamentos ON anuncios.pagamentos_id = pagamentos.id;
23
24 — Seleciona a pagina, o usuario as datas de inicio e fim de cada
   anuncio
25 CREATE VIEW datas_anuncio AS
26 SELECT anunciantes.id, anunciantes.pagina, anunciantes.usuario,
   anuncios.postagem_anuncio, anuncios.data_inicio, anuncios.
   data_final FROM anunciantes
27 INNER JOIN anuncios ON anunciantes.anuncios_id = anuncios.id;
28
29 — Seleciona a pagina, o usuario e o publico alvo de cada anuncio
30 CREATE VIEW publico_alvo_anuncio AS
31 SELECT anunciantes.id, anunciantes.pagina, anunciantes.usuario,
   anuncios.postagem_anuncio, anuncios.publico_alvo FROM anunciantes
32 INNER JOIN anuncios ON anunciantes.anuncios_id = anuncios.id;
33
34 — Seleciona a pagina, o usuario e regioa escolhida de cada anuncio
35 CREATE VIEW regioa_anuncio AS
36 SELECT anunciantes.id, anunciantes.pagina, anunciantes.usuario,
   anuncios.postagem_anuncio, anuncios.regiao FROM anunciantes
37 INNER JOIN anuncios ON anunciantes.anuncios_id = anuncios.id;

```

recursos/views.sql