

INFORMÁTICA PARA CONCURSOS

SUMÁRIO

1-	FERRAMENTA ATINGIR META NO EXCEL	3
2-	EXCEL AVANÇADO	6
3-	ACCESS	34
4-	PLANEJAMENTO DE BANCO DE DADOS	39
5-	CRIAÇÃO DE BANCO DE DADOS	47
	REFERÊNCIAS	

1- FERRAMENTA ATINGIR META NO EXCEL

Se você conhece o resultado que deseja obter de uma fórmula, mas não tem certeza sobre o valor de entrada necessário para chegar a esse resultado, use o recurso Atingir Meta. Por exemplo, suponha que você precise pedir algum dinheiro emprestado. Você sabe quanto dinheiro quer, quanto tempo deseja usar para pagar o empréstimo e quanto pode pagar a cada mês. Você pode usar o recurso Atingir Meta para determinar qual taxa de juros você precisará garantir para atingir seu objetivo de empréstimo.

Se você conhece o resultado que deseja obter de uma fórmula, mas não tem certeza sobre o valor de entrada necessário para chegar a esse resultado, use o recurso Atingir Meta. Por exemplo, suponha que você precise pedir algum dinheiro emprestado. Você sabe quanto dinheiro quer, quanto tempo deseja usar para pagar o empréstimo e quanto pode pagar a cada mês. Você pode usar o recurso Atingir Meta para determinar qual taxa de juros você precisará garantir para atingir seu objetivo de empréstimo.

Observação: O Atingir Meta funciona somente com um valor de entrada de variável. Se quiser aceitar mais de um valor de entrada, por exemplo, o valor do empréstimo e o valor do pagamento mensal, use o suplemento Solver.

Passo a passo com um exemplo

Vamos examinar o exemplo anterior, passo a passo.

Como você deseja calcular a taxa de juros de empréstimo necessária para cumprir sua meta, use a função PMT. A função PMT calcula um valor de pagamento mensal. Neste exemplo, o valor de pagamento mensal é a meta que você procura.

Preparar a planilha

1. Abra uma nova planilha em branco.
2. Primeiro, adicione alguns rótulos à primeira coluna para facilitar a leitura da planilha.
 - a. Na célula A1, digite **Valor do Empréstimo**.

- b. Na célula A2, digite **Prazo em Meses**.
- c. Na célula A3, digite **Taxa de Juros**.
- d. Na célula A4, digite **Pagamento**.
- 3. Em seguida, adicione os valores conhecidos.
 - a. Na célula B1, digite **100000**. Esse é o valor que você deseja pedir emprestado.
 - b. Na célula B2, digite **180**. Esse é o número de meses em que você deseja liquidar o empréstimo.

Observação: Embora você saiba o valor do pagamento desejado, ele não é digitado como um valor já que o valor do pagamento é um resultado da fórmula. Em vez disso, você adiciona a fórmula à planilha e especifica o valor do pagamento em uma etapa posterior, ao usar o recurso Atingir Meta.

- 4. Em seguida, adicione a fórmula para a qual você tem uma meta. Por exemplo, use a função PGTO:
 - a. Na célula B4, digite **=PMT(B3/12,B2,B1)**. Essa fórmula calcula o valor de pagamento. Neste exemplo, você deseja pagar US\$ 900 por mês. Você não insere esse valor aqui, pois deseja usar o Goal Seek para determinar a taxa de juros, e o Goal Seek exige que você comece com uma fórmula.

A fórmula faz referência às células B1 e B2, que contêm os valores que você especificou nas etapas anteriores. A fórmula também faz referência à célula B3, que é onde você irá especificar que Atingir Meta deve inserir a taxa de juros. A fórmula divide o valor em B3 por 12 porque você especificou um pagamento mensal, e a função PGTO assume uma taxa de juros anual.

Já que não há valores na célula B3, o Excel assume uma taxa de juros de 0% e, usando os valores do exemplo, retorna um pagamento de R\$ 555,56. Ignore esse valor por enquanto.

Usar Atingir Meta para determinar a taxa de juros

- 1. Na guia **Dados**, no grupo **Ferramentas de Dados**, clique em **Teste de Hipóteses** e clique em **Atingir Meta**.

2. Na caixa **Definir célula**, insira a referência da célula que contém a fórmula que você deseja resolver. No exemplo, essa referência é a célula B4.
3. Na caixa **Até o valor**, digite o resultado da fórmula desejado. No exemplo, o resultado é -900. Observe que esse número é negativo porque representa um pagamento.
4. Na caixa **Alterando a célula**, insira a referência da célula que contém o valor que você deseja ajustar. No exemplo, essa referência é a célula B3.

Observação: A célula que Atingir Meta altera deve ser referenciada pela fórmula na célula que você especificou na caixa **Definir célula**.

5. Clique em **OK**.
- 6.

O Goal Seek é executado e produz um resultado, conforme mostrado na ilustração a seguir.

	A	B
1	Valor do empréstimo	\$100,00
2	Termo em Meses	180
3	Taxa de Juros	7.02%
4	Pagamento	(\$900.00)

As células B1, B2 e B3 são os valores do valor do empréstimo, do comprimento do termo e da taxa de juros.

A célula B4 exibe o resultado da fórmula =PMT(B3/12,B2,B1).

7. Por fim, formate a célula de destino (B3) para exibir o resultado como uma porcentagem.
 - a. Na guia **Página Inicial**, no grupo **Número**, clique em **Porcentagem**.
 - b. Clique em **Aumentar Casas Decimais** ou **Diminuir Casas Decimais** para definir o número de casas decimais.

2- EXCEL AVANÇADO

Você provavelmente sente um calafrio quando ouve o nome “Excel”, certo? Tudo bem, fique tranquilo. Aqui é um lugar seguro! O **tutorial de Excel** que você sempre sonhou, já é uma realidade!

Nós entendemos que a maior parte dos profissionais tem receio quanto ao uso de planilhas. Elas parecem mesmo complicadas, tem formulas indecifráveis e funções complexas.

Especialmente profissionais de Marketing, que não tem muita familiaridade com a ferramenta.

Mas não precisa ser assim!

O Excel — e, mais recentemente, o Google Spreadsheets — são ferramentas fantásticas. Vou te mostrar como usá-las e levar sua capacidade de análise ao próximo nível.

Veja tudo o que você precisa para saber como usar Excel:

- Tutorial de Excel: a lógica de uma planilha
- Como navegar uma planilha
- Como trabalhar em múltiplas folhas nas planilhas
- Como usar corretamente uma célula
- Como usar Excel para facilitar seu trabalho
- Fórmulas básicas de Excel
- Fórmulas de Excel: intermediárias
- Truques avançados de Excel
- Visualize seus números: como fazer gráficos no Excel

Faça o download deste post inserindo seu e-mail abaixo

Tudo isso você confere agora:

Tutorial de Excel: a lógica de uma planilha

Ao abrir o Excel você já é jogado de cara em uma interface que não é muito amigável.

Mas não se preocupe, mexer no Excel é muito mais simples do que parece.

Em uma planilha de Excel, **o bloco básico de informação é uma célula**. Cada célula contém uma informação.

Ela pode ser uma data, um número, um bloco de texto, porcentagem ou fórmula, dentre outros. **Esta célula faz parte de uma folha.**

Cada planilha pode ser composta de várias folhas, cada uma delas com seu próprio conjunto de células. Assim, a nossa hierarquia é:

Arquivo (nome.xls ou nome.xlsx) > Folha (sheet) > Célula (cell).

** **Observação:** no Calc (do OpenOffice e BrOffice) a extensão de planilha é .ods. No Google Spreadsheets, ela é identificada por um endereço de internet, como “docs.google.com/xxxxxxxxxxxxx”*

Voltando: dentro de uma planilha, você pode ter várias folhas, em dentro de cada folha, várias células. É uma hierarquia bem simples, que rege qualquer conjunto de dados que você está planilhando.

Um exemplo prático seria:

- Planilha: Investimentos em Marketing
- Folha: agosto
- Célula: Investimento em Adwords.

Bem mais simples, certo?

Como navegar uma planilha

Trabalhando com planilhas, você pode referenciar (ou buscar) informações de células em outros locais para executar cálculos, comparações e outras operações. Assim, cada célula possui um “endereço” único dentro da sua própria folha.

Este endereço é composto de uma coordenada, que contém sua linha (a dimensão vertical) e coluna (sua orientação horizontal).

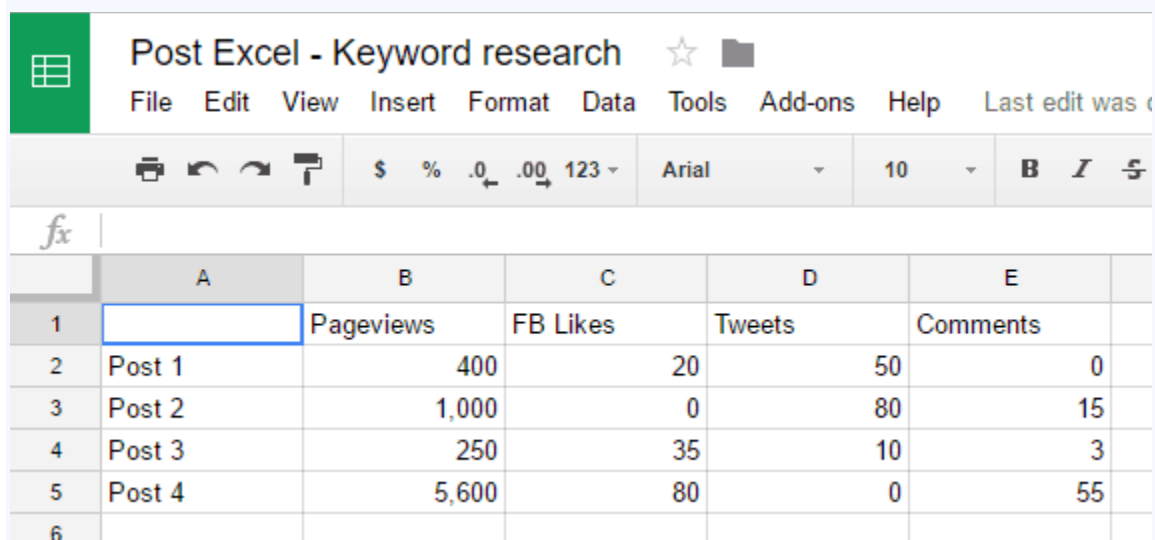
As linhas são identificadas por números inteiros, em ordem crescente.

Já as colunas, por letras, em ordem alfabética.

Este sistema extremamente versátil permite que você organize as suas informações e busque os seus dados de um lugar para outro de forma muito ágil e prática.

Este sistema é consistente no Excel, [Google Drive](#), Calc e em qualquer outro software de planilhas.

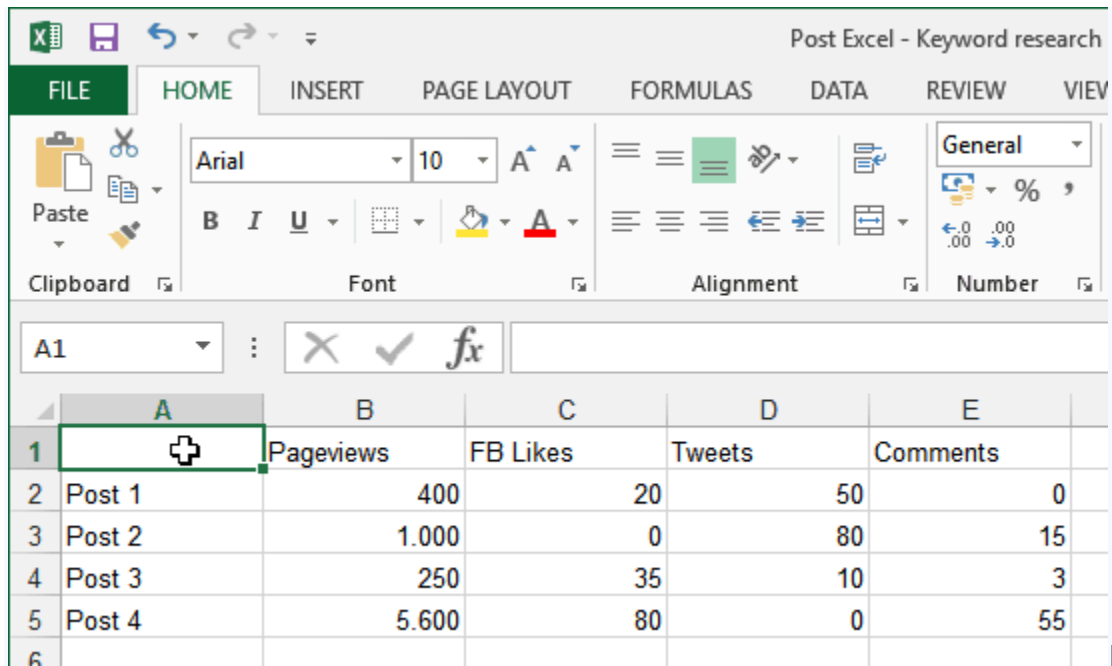
Veja no exemplo abaixo:



	A	B	C	D	E
1		Pageviews	FB Likes	Tweets	Comments
2	Post 1	400	20	50	0
3	Post 2	1,000	0	80	15
4	Post 3	250	35	10	3
5	Post 4	5,600	80	0	55
6					

[Vista

do Google Spreadsheets]



Post Excel - Keyword research

	A	B	C	D	E
1		Pageviews	FB Likes	Tweets	Comments
2	Post 1	400	20	50	0
3	Post 2	1.000	0	80	15
4	Post 3	250	35	10	3
5	Post 4	5.600	80	0	55

[Vista

dentro do Excel]

Na “coluna A” estão listados os posts e na “linha 1” as interações que cada um deles teve. Veja como é fácil navegar esta informação:

- A “linha 3”, dá as interações do “Post 2”. O título do post, tem o endereço “A3”. O número de pageviews deste post está em “B3”. O número de FB Likes deste post está em “C3”. E assim por diante...

Desta maneira, eu sei que todas as informações relacionadas à “coluna D” são “Tweets” e, na medida que eu desço para D2, D3, D4 e D5, eu vejo a dimensão “Tweets” de propriedades diferentes, identificadas na “coluna A”.

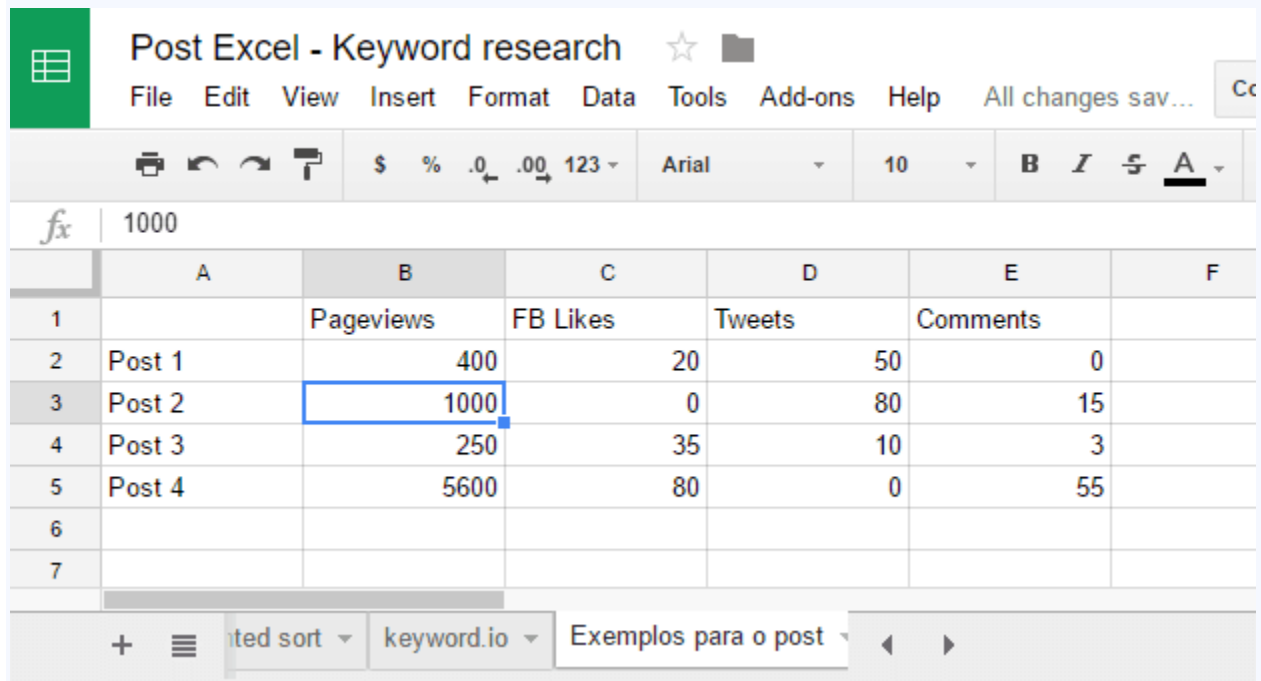
Você também pode referenciar um intervalo inteiro de uma só vez, usando “:” (dois pontos) como separador entre o começo e o fim do intervalo, assim ficamos com estes exemplos:

- A2:A5 – O intervalo vertical que compreende A2, A3, A4 e A5.
- A2:E2 – O intervalo horizontal que compreende A2, B2, C2, D2 e E2.
- A2:E5 – O quadrado que tem um vértice em A2 e outro em E5.
- A2:A – O intervalo vertical que começa em A2 e vai até o final da coluna A.
- A2:2 – O intervalo horizontal que vai de A2 até o final da linha 2.

Agora que você já consegue identificar qualquer célula em uma folha, vamos dar o próximo passo:

Como trabalhar em múltiplas folhas nas planilhas

Veja a visão do print screen acima, de forma um pouco mais aberta:

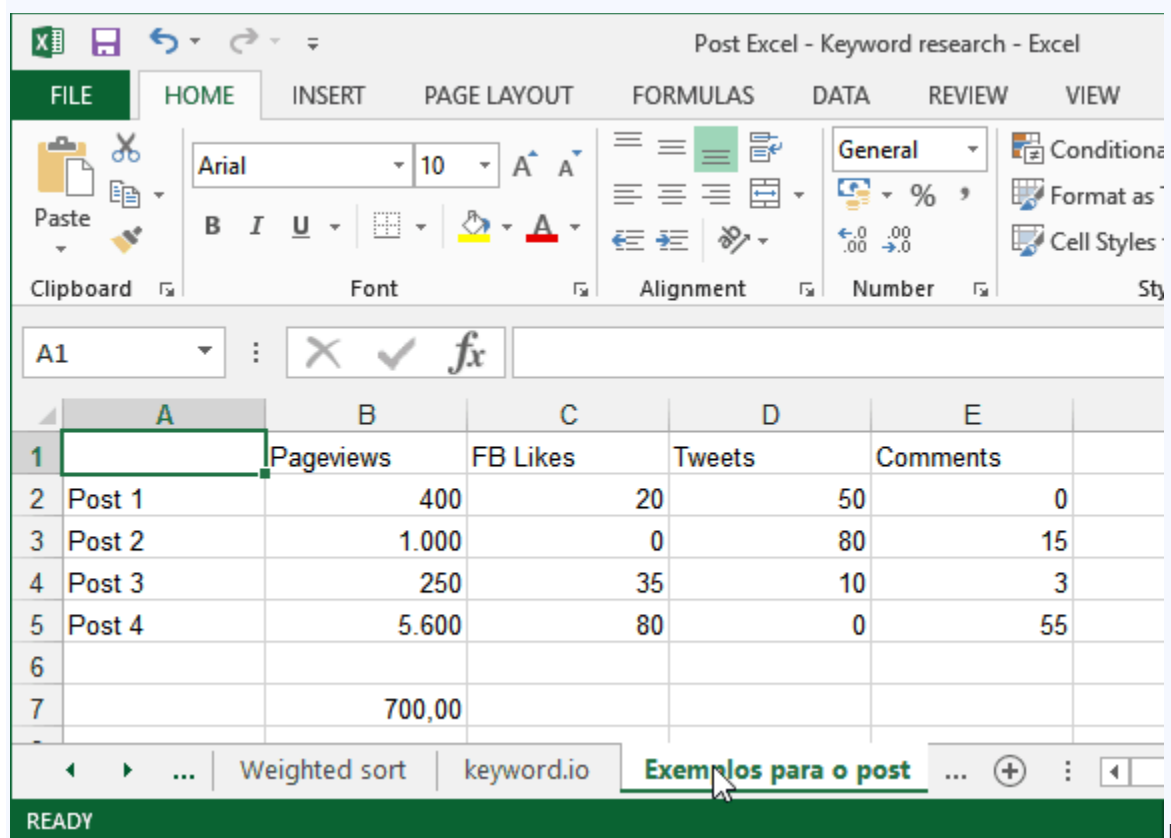


The screenshot shows the Google Sheets interface. The title bar reads 'Post Excel - Keyword research'. The menu bar includes File, Edit, View, Insert, Format, Data, Tools, Add-ons, Help, and a status bar 'All changes sav...'. The toolbar shows various icons for undo, redo, and formatting. The spreadsheet has columns A through F and rows 1 through 7. The data is as follows:

	A	B	C	D	E	F
1		Pageviews	FB Likes	Tweets	Comments	
2	Post 1	400	20	50	0	
3	Post 2	1000	0	80	15	
4	Post 3	250	35	10	3	
5	Post 4	5600	80	0	55	
6						
7						

At the bottom, there are tabs for '+', 'Weighted sort', 'keyword.io', and 'Exemplos para o post'.

[Vista do Google Spreadsheets]



The screenshot shows the Microsoft Excel 2013 interface. The title bar reads 'Post Excel - Keyword research - Excel'. The ribbon includes FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, and VIEW. The HOME ribbon is active, showing options for Clipboard, Font, Alignment, Number, and Styles. The spreadsheet data is identical to the Google Sheets version, with the following table:

	A	B	C	D	E	F
1		Pageviews	FB Likes	Tweets	Comments	
2	Post 1	400	20	50	0	
3	Post 2	1.000	0	80	15	
4	Post 3	250	35	10	3	
5	Post 4	5.600	80	0	55	
6						
7		700,00				

At the bottom, there are tabs for 'Weighted sort', 'keyword.io', and 'Exemplos para o post'. The status bar at the bottom left says 'READY'.

do Excel 2013]

[Vista

Agora, além de ver as informações de célula, você vê também o nome do arquivo (spreadsheet) “Post Excel – Keyword research”. Olhando na parte de baixo do print, você também vê que estou trabalhando com várias folhas. Estão visíveis “Exemplos para o post” e também “keyword.io”.

É muito comum usar folhas diferentes para agregar dados em comum, como os números de um mês, departamento ou ação específicas.

O bacana é que você pode referenciá-los de um lugar para outro.

Então é possível ter uma folha “números do ano”, por exemplo, que reúne a somatória de cada mês em um resumo.

A sintaxe para isto é (sem aspas) “Nome_da_folha!EndereçoDaCélula”. Uma exclamação separa o nome da folha do endereço da célula.

Então, o endereço completo dos pageviews do “Post 1” citado acima é *Exemplos para o post!B3*.

Jóia! Agora você já sabe como identificar todas as células, em todas as folhas, de qualquer planilha que fizer. Vamos dar então uma olhada em como fazer coisas úteis com isto:

Como usar corretamente uma célula

Agora que você já sabe identificar uma célula dentro da sua planilha, é hora de entender como trabalhar corretamente com ela. **A maior parte das pessoas não entende o trecho abaixo, e fracassa por isto!**

Uma célula pode conter informações de três naturezas diferentes:

- **Texto (labels):** Um conjunto de letras e números (string) que serve apenas para identificar ou organizar um conjunto. No exemplo acima, B1, C1, D1 e E1 são células de texto, assim como A2, A3, A4 e A5.
- **Números (values):** Composto apenas por números, um “value” contém um dado que pode ser entendido de várias maneiras, como número, percentual, data, duração, entre

outros. A natureza de um value precisa ser declarada para evitar que o Excel faça interpretações incorretas! Para entender isto, pense no número “100586”. Este número, puro, pode ser entendido como:

- Moeda: 100586 é R\$100.586,00
- Data: 10/maio/1986
- Data: 05/outubro/0086
- Perceba que todas as interpretações estão corretas. O que falta é contexto!
- **Fórmula:** É uma instrução, que diz ao Excel ou Spreadsheet o que fazer para ocupar aquele lugar. Todas as fórmulas começam com o sinal “=” (igual) e transformam, de alguma maneira, o conteúdo de outras células.
- Uma fórmula pode contar várias funções, em conjunto. Dentro de uma função, os parâmetros para seu uso são colocados em parênteses, com uma vírgula ou ponto e vírgula os separando, dependendo da linguagem do seu Excel.

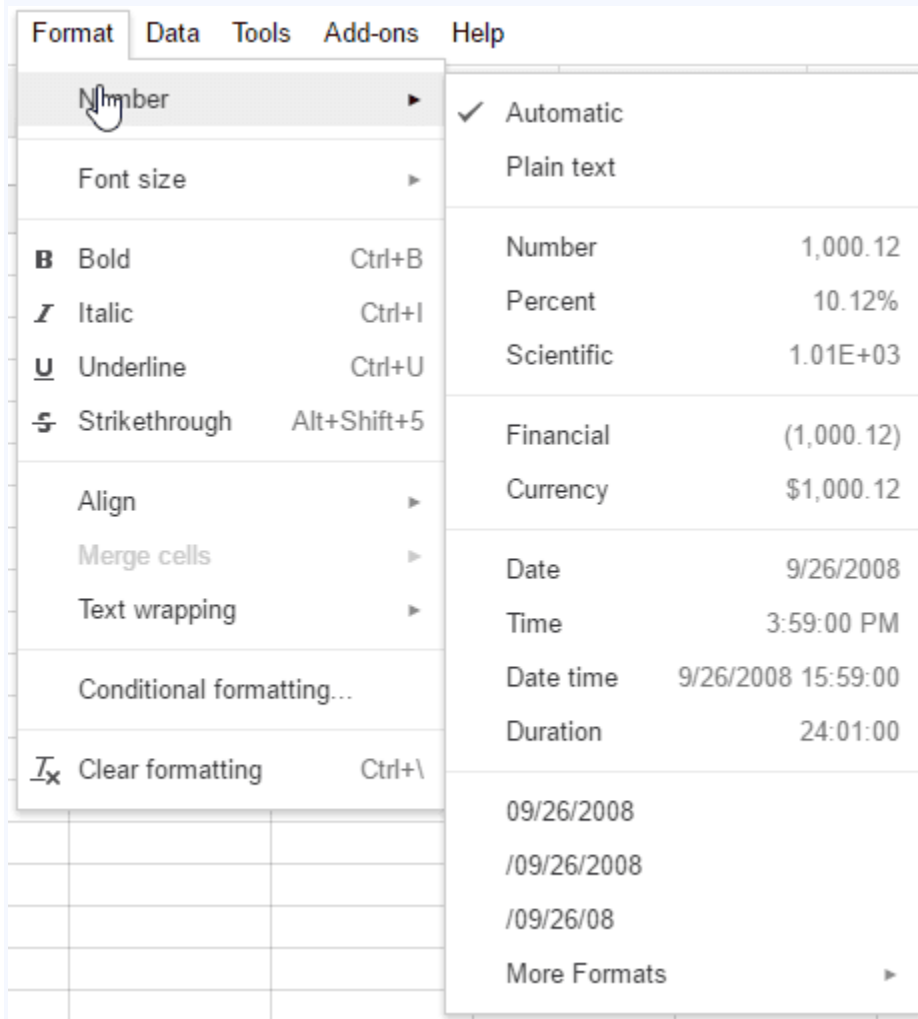
The image shows two overlapping screenshots. The top one is the Microsoft Excel ribbon interface, with the 'FORMULAS' tab selected. The formula bar at the bottom of the ribbon is highlighted with a yellow box and contains the 'fx' icon. The bottom screenshot is a Google Sheets interface for a spreadsheet titled 'Post Excel - Keyword research'. It also shows the 'fx' icon in the formula bar highlighted with a yellow box. Below the formula bar is a table with 5 columns (A-E) and 5 rows (1-5).

	A	B	C	D	E
1		Pageviews	FB Likes	Tweets	Comments
2	Post 1	400	20	50	0
3	Post 2	1,000	0	80	15
4	Post 3	250	35	10	3

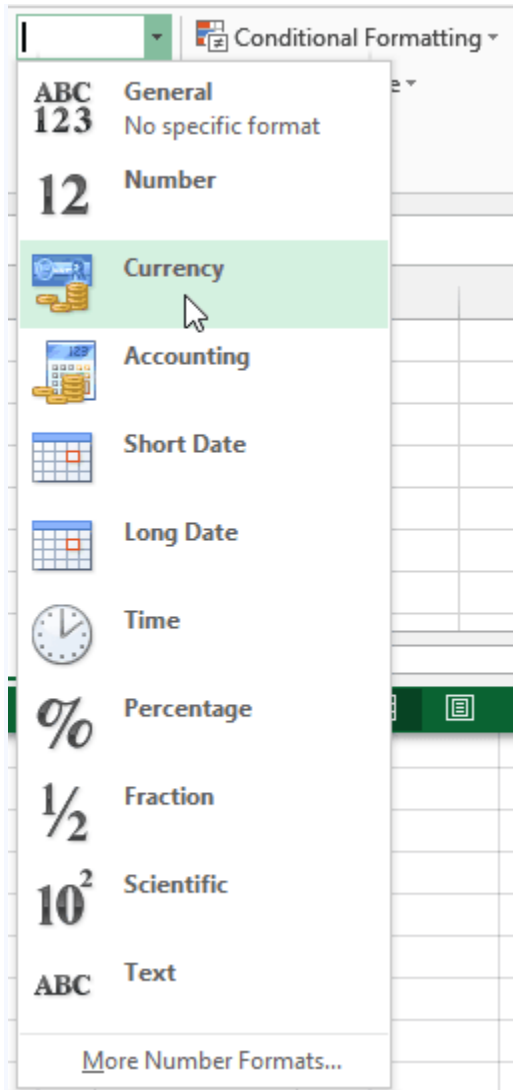
[Indicador de função (fórmula) no Excel e Spreadsheets]

Registrando números corretamente em células

Como já vimos, muitas vezes é necessário dizer ao Excel que tipo de número estamos entregando a ele. Para isto, basta selecionar as células, clicar em “Format” e depois em “Number”.



[Escolha de formato, no Spreadsheets]



[Escolha de formato, no Excel]

“Automatic” é o padrão, e vai tentar adivinhar qual o tipo de informação ali. Eu particularmente não recomendo usar este modo, pois ele é bem propenso a falhas.

“Plain text” ignora o número, e o trata como uma informação neutra, parte de um texto, mas desprovido de significado matemático qualquer.

Um código de barras é um bom exemplo de número “plain text”. É apenas uma sequência de dígitos.

De novo no nosso exemplo:

	A	B
1		Pageviews
2	Post 1	400.00
3	Post 2	\$1,000.00
4	Post 3	9/6/1900
5	Post 4	5.60E+03

Aqui, cliquei para que cada célula fosse vista com as identificações:

- B2: "Number". O número 400 ganhou duas casas decimais. (O número de casas decimais pode ser configurado)
- B3: "Currency": O número 1000 ganhou um sinal de dólar e também um marcador para centavos
- B4: "Date" O número "250" foi convertido em 6 de setembro de 1900. (250 dias após o 01/01/1900.)
- B5: "Scientific" O número 5600 em notação científica. Ou seja 5.6×10^3 (5.6 vezes 10 elevado ao cubo)

Você não precisa se preocupar demais em entender cada uma destas identificações agora!

O mais importante é que você saiba que, dependendo da notação entregue ao programa, o seu número será tratado de maneira diferente.

Sabendo disto, certamente você poderá escolher a notação correta na medida em que estiver trabalhando.

Como usar Excel para facilitar seu trabalho

Se você chegou até aqui, meus parabéns! Você já entende mais sobre o funcionamento do Excel (e Google Spreadsheets, Calc e outras planilhas) do que a maior parte das pessoas!

É hora de trazer então a ferramenta mais legal para a conversa: **fórmulas**.

Como já vimos, uma fórmula é armazenada sempre dentro de uma célula. Ela rege o valor que será mostrado naquele endereço (e, em alguns casos, em endereços adjacentes também) e pode usar informações de vários lugares para isto.

Existem centenas de funções possíveis, que podem ser combinadas de infinitas maneiras, de modo a automatizar —ou pelo menos simplificar— praticamente qualquer tarefa. Para entendermos a variedade de operações disponíveis, vamos ver as suas categorias principais:

String / Texto: Para unir, separar, combinar ou manipular blocos de texto ou suas letras separadamente.

Data / Hora: Operações relacionadas ao cálculo do tempo, intervalo entre datas, dias úteis e a identificação dos dias, semanas e meses.

Matemática / Trigonometria: Além das operações aritméticas, permite calcular equações complexas, constantes (como Pi), gerar números aleatórios e alterar números com arredondamento, conversão de unidades e operações com ângulos.

Estatística: Para análise de conjuntos. Identifica automaticamente percentis, médias, medianas, distribuições, contagens, permutações, variância, entre outras.

Funções lógicas: Permite que outras funções sejam combinadas e analisadas em função de uma regra descrita pelo usuário. Retorna “TRUE” ou “FALSE” para funções que checam por uma determinada condição e tomam uma ação de acordo com o resultado.

Funções de informação: São usadas basicamente para avaliar o resultado de outras funções. Podem responder se uma determinada célula está em branco, é um erro, uma data ou um texto, por exemplo. Também identificam e manipulam os identificadores de formato (data, hora, percentual, moeda, etc.) que falamos acima.

Database: Entregam valores relacionados à um grupo de células (range). Podem ser usadas para contar o número de elementos, apontar o seu maior (ou menor) membro, a variância dentro do grupo, entre outros.

Financeiras: O nome diz tudo! São usadas para simplificar o cálculo de juros, depreciação, períodos de investimento e número de pagamentos, por exemplo.

Engenharia: Basicamente usada para converter números entre suas diversas formas de representação, como binários, hexa, octal e algumas unidades de medida.

Referência / Lookup: Varrem um determinado espaço em busca de um valor dado pelos usuários. Ideais para encontrar correspondências entre fontes de dados diferentes, assim como relações entre índices diversos que possuem elementos em comum.

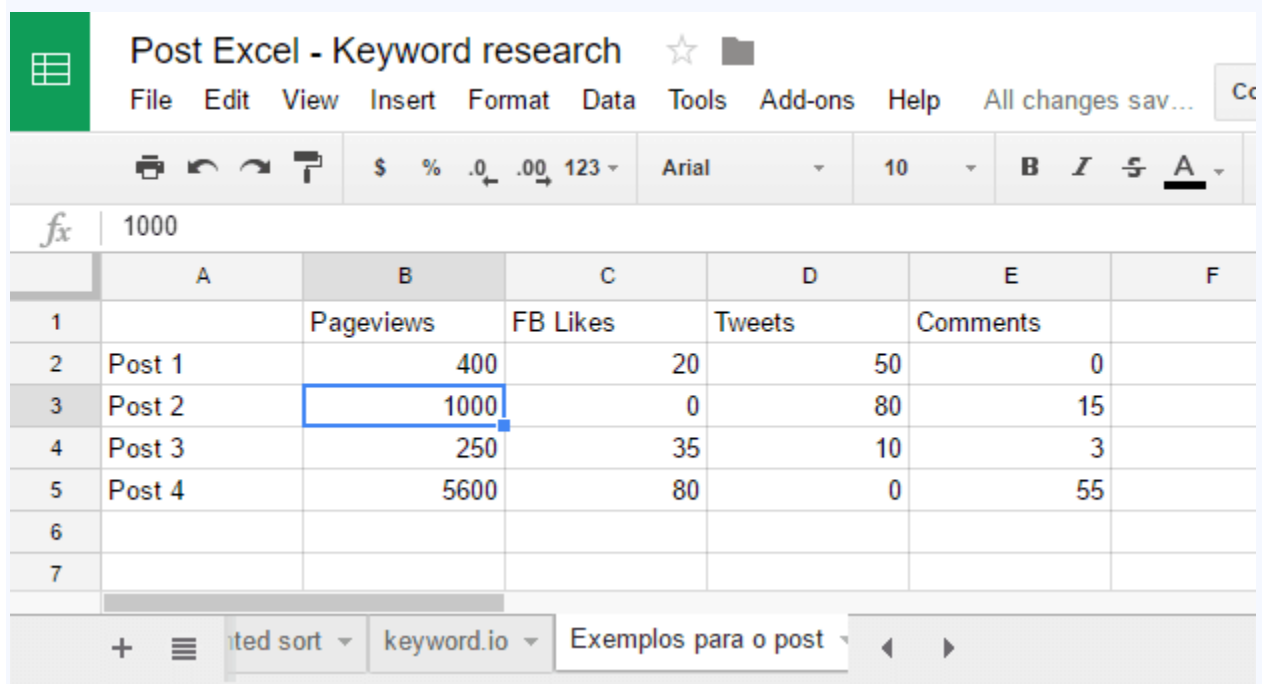
Quero deixar algo claro. Tenho plena certeza que a listagem acima, em um primeiro momento, pode ter de deixado um pouco confuso.

Afinal, com tantas possibilidades, fica complicado saber: **por onde começar?** Com tantas opções em mãos, quais delas eu devo usar para facilitar o meu trabalho?

Fique tranquilo. Selecionei algumas das fórmulas mais comumente usadas e vamos passar por elas juntos agora.

Vamos lá?

Para isto, vamos lembrar o nosso conjunto de dados de exemplo:



	A	B	C	D	E	F
1		Pageviews	FB Likes	Tweets	Comments	
2	Post 1	400	20	50	0	
3	Post 2	1000	0	80	15	
4	Post 3	250	35	10	3	
5	Post 4	5600	80	0	55	
6						
7						

Fórmulas básicas de Excel

Para aprender como usar Excel é fundamental entender a saber utilizar as suas fórmulas.

Aqui vamos apresentar algumas das fórmulas básicas mais utilizadas na ferramenta.

Confira!

SUM (ou SOMA)

A função SUM faz a adição de um grupo de valores determinados pelo usuário. Para somar o número total de pageviews nos quatro posts do nosso exemplo, a função seria escrita assim:

`=SUM(B2; B3; B4;B5).`

De forma resumida: `=SUM(B2:B5)`

Neste caso, o resultado seria “7250”, a somatória de todos os termos indicados.

O mesmo resultado seria obtido escrevendo:

`=B2+B3+B4+B5`

Entretanto, em um conjunto maior de dados, esta notação seria muito mais trabalhosa e susceptível a erros.

AVERAGE (ou MÉDIA)

Retorna a média numérica de um conjunto. Aplicando esta função ao mesmo intervalo anterior, temos:

`=AVERAGE(B2:B5) ou`

`=AVERAGE(B2;B3;B4;B5)`

O resultado aqui seria 1812,50, que é a soma dos 4 termos divididos pelo número de termos do conjunto, que é 4. $7250/4 = 1812,50$.

MEDIAN (ou MEDIANA)

Média e mediana são rotineiramente confundidas, mas elas têm funções diferentes. Como vimos, a média é a soma dos valores dividida pelo número de termos. Já a mediana é o valor central de um conjunto.

Quer dizer, é o ponto onde metade dos valores estão acima dele e metade estão abaixo.

No nosso exemplo (que tem um número par de termos), encontraríamos a média somando os dois valores centrais (400 e 1000) e dividindo por 2, com resultado 700.

Em um conjunto ímpar, o valor mediano é listado diretamente. Para executar esta função, basta usar:

`=MEDIAN(A2:A5)`

COUNT, COUNTA, COUNTUNIQUE e COUNTBLANK

Estas duas funções servem para contar quantos elementos existem em um conjunto. COUNT vai buscar apenas as células que tem números dentro do intervalo dado, e retornar quantas entradas existem.

Já COUNTA retorna à quantidade de elementos de qualquer natureza. Pode ser usado para saber quantas URLs estão contidas em uma lista, por exemplo.

COUNTUNIQUE também faz uma contagem de lista, mas ignora itens repetidos. Então se você busca entradas únicas num conjunto onde há duplicatas, esta é sua escolha.

Por último, COUNTBLANK lista apenas as células em branco de um intervalo. Use-a para identificar falhas em grandes conjuntos ou erros no preenchimento.

A sintaxe de todas elas é semelhante:

`=COUNT(intervalo a ser pesquisado)`

DATEDIF

Esta função calcula quanto tempo existe entre duas datas. Funciona assim:

=DATEDIF(data de início; data final, unidade de tempo)

A data de início ou final podem ser digitadas direto dentro da função ou referenciada de outros lugares, como vimos acima.

Já a unidade de tempo precisa ser escolhida dentre algumas opções. Para listar em anos, é “Y”. Meses inteiros “M” e, dias é “D”.

Estas são apenas algumas das operações básicas do Excel. Seguindo em frente, podemos usar este conhecimento para construir fórmulas mais sofisticadas.

Fórmulas de Excel: intermediárias

Agora que você já aprendeu as fórmulas básicas é hora de avançarmos mais um pouco!

VLOOKUP, ou PROCV

Esta fórmula vai ser uma grande aliada quando você precisar comparar dados de duas fontes diferentes. Por exemplo, em uma folha você tem os nomes e emails de um grupo de pessoas, e no outro você tem os nomes e data de aniversário.

O VLOOKUP vai varrer um intervalo de cima abaixo, na horizontal, em busca de um valor que você determinou.

Uma vez que este valor é encontrado, ele retorna o conteúdo de qualquer célula que esteja à frente daquele valor.

Veja uma folha com um exemplo semelhante ao nosso:

	A	B	C	D	E
1	Nome	Email		Nome	Aniversário
2	Hugo	h@email.com		Luis	27/05/1986
3	José	j@email.com		Hugo	23/04/1988
4	Luis	l@email.com		José	19/01/1990

Imagine que estas duas listas (A:B e D:E) têm, na verdade, centenas de nomes. Seria um pesadelo trazer o email de cada aniversariante, certo? Veja como o VLOOKUP resolve.

=VLOOKUP(valor a pesquisar, intervalo onde pesquisar, coluna de retorno, [tipo de busca])

Calma que não é complicado! Vamos ver cada termo:

- **Valor a pesquisar (lookup value):** No nosso caso, seria o nome. Assim, se queremos achar o email do Luis, este termo é D2.
- **Intervalo a pesquisar (table array):** Onde a pesquisa deve ser feita, incluindo as células onde está a informação buscada. Aqui é A2:B4.
- **Coluna de retorno (index):** Qual a distância horizontal entre a busca e o valor que queremos retornar, onde “1” é a própria coluna. Aqui, como o email de Luis está logo à direita, este valor é 2. Este valor só aceita números positivos, então não consegue fazer buscas à esquerda, apenas à direita do termo buscado.
- **Tipo de busca (is_sorted):** Se este parâmetro for 0 (zero) ou “FALSE”, ele traz resultado apenas se a identificação for exata. “Luís” não seria retornado no lugar de “Luis”, por exemplo. Se for TRUE, o Excel parte do princípio que a lista está em ordem alfabética, e vai tratar equivalências incompletas como verdadeiras.

Substituindo na nossa fórmula, temos:

=VLOOKUP(D2, A2:B4; 2; TRUE)

Esta fórmula retorna, sem dúvidas, “l@email.com” como resultado!

<i>fx</i>	=VLOOKUP(D2, A2:B4, 2, TRUE)					
	A	B	C	D	E	F
1	Nome	Email		Nome	Aniversário	
2	Hugo	h@email.com		Luis	27/05/1986	l@email.com
3	José	j@email.com		Hugo	23/04/1988	
4	Luis	l@email.com		José	19/01/1990	

Dica: existe uma função HLOOKUP (ou PROCH) que funciona exatamente da mesma maneira, mas realiza a busca na horizontal, em vez de vertical. Use a variação adequada em função dos seus dados! □

IF Statements

Uma fórmula “IF” nos permite testar como uma célula ou intervalo respondem a um **teste lógico**. Um teste lógico só pode trazer duas repostas: TRUE ou FALSE. Ou seja, ele verifica se uma condição é, ou não verdadeira.

Apesar da aparente simplicidade, esta fórmula é uma das mais versáteis, uma vez que existem uma infinidade de testes que podem ser feitos dessa maneira.

Vejamos antes a sintaxe da fórmula:

IF(teste lógico, O que fazer se TRUE; O que fazer se FALSE)

Vamos usar um IF para saber se o email de Hugo é l@email.com. Ficaria assim:

IF(B2="l@email.com", "Este é o email de Hugo", "Este não é o email de Hugo")

O resultado:

=IF(B2="l@email.com", "Este é o email de Hugo", "Este não é o email de Hugo")						
A	B	C	D	E	F	G
Nome	Email		Nome	Aniversário		
Hugo	h@email.com		Luis	27/05/1986	l@email.com	
José	j@email.com		Hugo	23/04/1988	Este não é o email de Hugo	
Luis	l@email.com		José	19/01/1990		

Num teste lógico não precisamos verificar apenas igualdade. Os símbolos de maior e menor (< e >) também são muito úteis para fazer comparações lógicas. Para usar

“igual ou maior” e “igual ou menor”, basta colocar o sinal “=” na frente, como em “>=” e “<=”.

Acompanhe um exemplo mais avançado comigo: quero saber em um grupo de pessoas, quem tem mais de 30 anos de idade.

Então vou comparar as datas de aniversário (usando o DATEDIF para saber quais resultados são maiores que 30. Também vou usar uma função nova, o TODAY(), que apenas retorna a data de hoje.

Veja só:

<i>fx</i>	=IF(DATEDIF(E3,TODAY(),"Y")>=30, "Mais de 30 anos", "Menos de 30 anos")					
	A	B	C	D	E	F
1	Nome	Email		Nome	Aniversário	
2	Hugo	h@email.com		Luis	27/5/1986	Mais de 30 anos
3	José	j@email.com		Hugo	23/4/1988	Menos de 30 anos
4	Luis	l@email.com		José	19/1/1990	Menos de 30 anos

Nestes exemplos, o meu resultado para o teste lógico trouxe apenas como resultado um texto. Mas também é possível usar qualquer fórmula para realizar uma ação tanto em caso negativo, quanto positivo.

COUNTIF, SUMIF e AVERAGEIF

Agora que você já sabe usar as COUNT, SUM, AVERAGE e IF, você pode combiná-las em três novas funções. Em todas elas, a lógica é a mesma.

A função só é executada quando um teste lógico retorna “TRUE” para uma condição. A sintaxe de todas elas é a mesma.

Por exemplo, se eu quiser saber quantas pessoas de um grupo tem menos de 30 anos, eu posso usar:

=COUNTIF(Intervalo a ser verificado, condição para verificar no intervalo)

No nosso exemplo, usando as mesmas funções de antes, ficaria assim:

=COUNTIF(F2:F4,"=Menos de 30 anos")					
A	B	C	D	E	F
Nome	Email		Nome	Aniversário	
Hugo	h@email.com		Luis	27/5/1986	Mais de 30 anos
José	j@email.com		Hugo	23/4/1988	Menos de 30 anos
Luis	l@email.com		José	19/1/1990	Menos de 30 anos
					2

Truques avançados de Excel

Quando temos apenas duas grandezas organizadas em uma planilha, a sua própria worksheet (ou folha de trabalho) organiza todas as informações que você precisa.

Afinal, toda planilha tem duas dimensões: linhas e colunas. Mas muitas vezes é preciso ir além disto. Por isto, existem alguns truques para lidar com um número maior de variáveis sem perder o controle.

Na verdade, estas funções vão além!

Com os recursos abaixo você consegue ordenar seus dados em diferentes maneiras, e pode tirar insights das novas visualizações que seriam um tormento com os dados desorganizados.

Vamos a elas:

SORT (ou ORDENAR)

A função sort organiza, em ordem crescente ou decrescente, uma coluna ou intervalo de dados. Para dados em números, datas, porcentagens e moeda, a ordem será Bem... numérica!

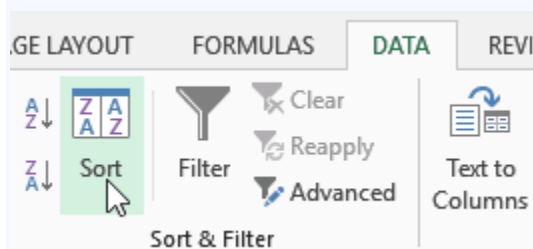
Em caso de dados alfanuméricos, contendo qualquer combinação de letras e números, a ordenação alfabética será usada.

É possível acionar o SORT através de uma célula de fórmula. Neste caso, você usa a sintaxe:

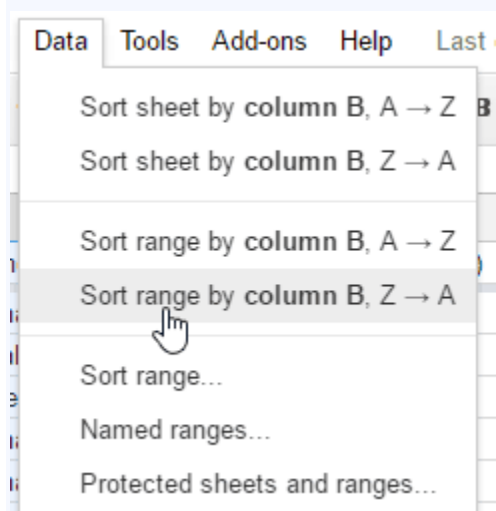
=SORT(intervalo, ordenação)

Este acionamento é útil para quando você não quer mexer nos dados originais, pois ela traz o intervalo ordenado para outra parte da folha, ou mesmo para uma folha diferente no seu arquivo, preservando o conjunto inicial intacto.

Para quando você quer ordenar os dados originais, você pode acionar o SORT no menu, clicando em:



[Sort no excel]



[Sort no spreadsheets]

Note que sempre que você utilizar um intervalo para seu SORT, precisa de uma coluna de referência. Assim, as relações entre colunas são mantidas sempre com suas associações corretas. Uma baita mão na roda!

Como fazer TABELA DINÂMICA (ou PIVOT TABLE)

Uma tabela dinâmica permite que você agrupe e analise dados de forma inteligente com poucos cliques. É ideal para quando existem variáveis diferentes em um mesmo conjunto de dados.

Na prática, ela facilita que você separe e agrupe seus números em segmentos específicos, contabilizando apenas os valores relevantes por soma, média, contagem e outras funções matemáticas.

Calma! Não é tão complicado quanto parece.

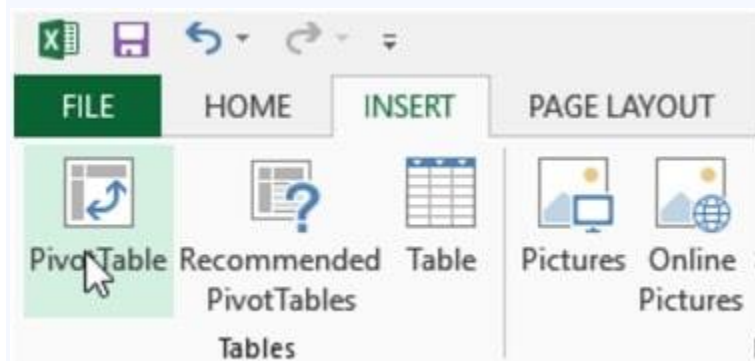
Veja o conjunto de dados abaixo:

	A	B	C	D	E	F
1	Post	Autor	Pageviews	FB Likes	Tweets	Comments
2	Post 1	Renato	400	20	50	0
3	Post 2	Jeremias	1.000	0	80	15
4	Post 3	Jeremias	250	35	10	3
5	Post 4	Renato	5.600	80	0	55

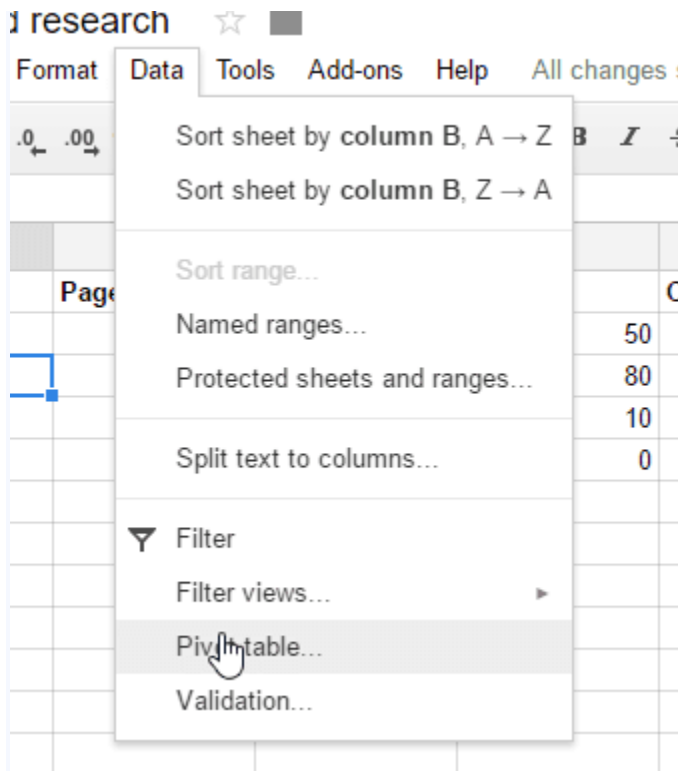
Agora, além dos posts e dados sobre eles, temos uma dimensão adicional, que é o autor de cada post. Aqui só temos 4 linhas de exemplo.

Mas já imaginou como seria para computar os resultados de dezenas de autores diferentes, em centenas de posts? Uma trabalhadeira sem tamanho, susceptível a erros e chata de fazer.

É para este tipo de análise que as tabelas dinâmicas vão salvar sua vida!



[Pivot Table no Excel]



[Pivot Table no Spreadsheet]

1- Escolha quais dados serão tabulados

O primeiro passo é definir qual o intervalo será considerado na tabela. No nosso caso, a A1 a F5, ou “\$A\$1:\$F\$5”, para garantir que o intervalo é fixo. Confira sempre se a primeira linha do intervalo possui os rótulos de cada coluna! Isto ajuda muito nos próximos passos.

2- Escolha quais campos serão contabilizados

Imagine que eu quero saber o total e a média de acessos que cada autor teve em seus posts. Assim, o primeiro campo que vou escolher para as minhas linhas (rows) é “Autor”. Isto diz ao Excel que eu vou agrupar os meus valores em função daqueles que já existem nesta linha. Neste exemplo “Renato” e “Jeremias”.

Sabendo disso, eu já posso escolher quais valores serão contados para estas linhas. Aqui são “Pageviews”.

	A	B
1		
2		
3	Row Labels	Sum of Pageviews
4	Jeremias	1250
5	Renato	6000
6	Grand Total	7250
7		
8		
9		

PivotTable Fields

Choose fields to add to report:

- ☐ Post
- ☒ Autor
- ☒ Pageviews
- ☐ FB Likes

Perceba que os pageviews foram automaticamente contabilizados como somas. (SUM), por serem números. Para mudar isto, você pode clicar com o botão direito sobre o valor “Sum of Pageviews” e escolher “Value Field Settings”. Então é possível trocar a soma por contagem, média, máxima, mínima, variância, entre outros.

No Spreadsheets, é só clicar em “Summarize by:”, também no campo value:

Post Excel - Keyword research
☆
📁

File Edit View Insert Format Data Tools Add-ons
Comments
Share

More

Report Editor

fx
405.6

	A	B	C
1	Jeremias	251	
2	Renato	405.6	
3	Grand Total	656.6	
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

SUM
COUNTA
COUNT
COUNTUNIQUE
AVERAGE
MAX
MIN
MEDIAN
PRODUCT
STDEV
STDEVP
VAR
VARP

Summarize by: SUM

Filter - Add field

3- Indo além

Fizemos um exemplo muito simples para ilustrar. As tabelas dinâmicas podem te ajudar a resolver problemas muito mais complexos! Você pode empilhar múltiplas linhas e também acrescentar uma segunda variável em colunas para plotar seus valores.

Quer mais? Você também pode usar qualquer uma das suas colunas para filtrar o conjunto inteiro, “cortando” seu conjunto de dados em qualquer segmentação que faça sentido para você.

Não importa o quão grande seu dataset seja, fazer cortes e segmentos com planilhas dinâmicas é uma das habilidades mais úteis que você pode adquirir!

Visualize seus números: como fazer gráficos no Excel

Seres humanos não são muito bons relacionando números. Afinal, a habilidade de dimensionar e comparar um monte de valores é algo muito novo, que nossos ancestrais nunca tiveram o tempo (ou necessidade) para evoluir.

Quer dizer, essa análise abstrata precisa ser treinada conscientemente. Não faz parte das nossas habilidades inatas.

Mas por que mencionar a evolução do cérebro em um artigo sobre Excel? Ainda mais depois de você ter lido mais de 3000 palavras sobre como organizar seus números aqui?

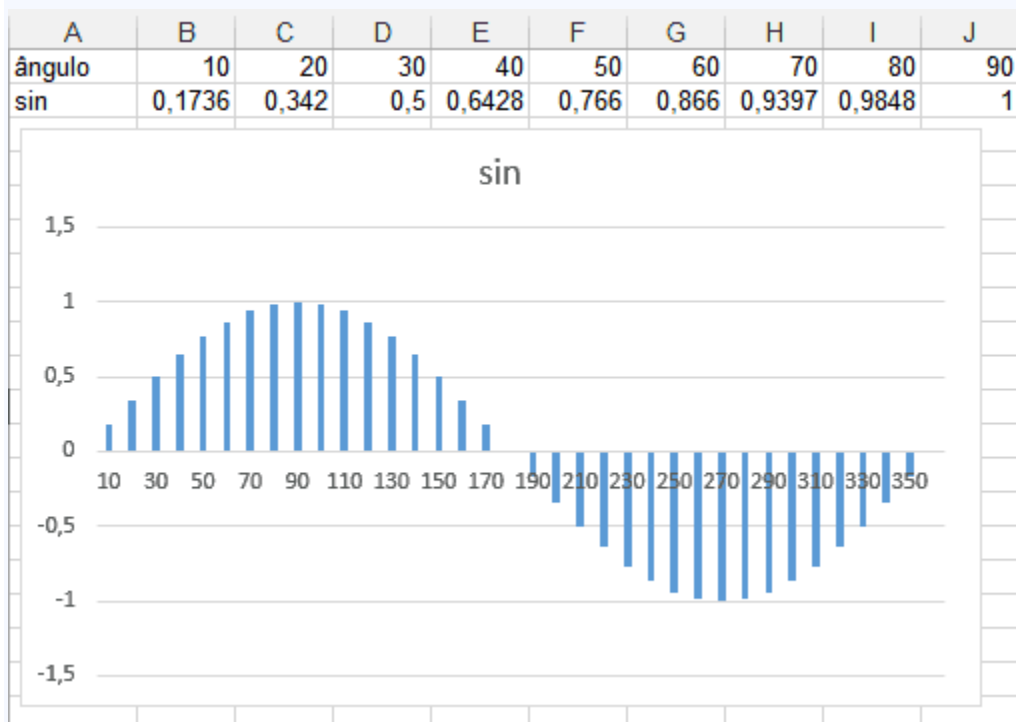
Oras, por que há uma maneira de tornar conjuntos gigantescos de dados muito mais compreensíveis para os nossos cérebros primatas! Conheçam os gráficos!

Um gráfico é uma representação visual de um conjunto numérico. Eles relacionam diferentes grandezas a um eixo numerado (em geral X e Y, horizontal e vertical) e mostram através de uma imagem, como um valor em X se relaciona a um valor em Y.

Vamos começar com um exemplo matemático para deixar este conceito bem claro.

B2		:				=SIN(RADIANS(B1))				
	A	B	C	D	E	F	G	H	I	J
1	ângulo	10	20	30	40	50	60	70	80	90
2	sin	0,1736	0,342	0,5	0,6428	0,766	0,866	0,9397	0,9848	1
3										

Na linha 1 temos valores para os ângulos, e na linha 2 uma fórmula para calcular o seno do ângulo acima (alô, trigonometria!). Assim, quando eu seleciono este conjunto e clico em “Insert Graph”, isto acontece:



Cada valor da nossa linha 1 foi marcado no eixo X (horizontal) e um valor correspondente a eles foi calculado no eixo Y (vertical). O resultado neste exemplo é uma onda senoidal, do jeito que sua professora disse que seria na oitava série.

Mas como isso se aplica à minha vida?

Uma coisa é mostrar ao seu gerente que seu número de pedidos aumentou de 331 para 503 de um trimestre para outro, finalmente alcançando 765 no trimestre seguinte. Você pode dizer esta frase com toda a empolgação do mundo, mas provavelmente não vai passar sua mensagem com a ênfase que gostaria.

Agora olhe para isto:

INFORMÁTICA PARA CONCURSOS

Ticket médio	Taxa de crescimento mensal (%)	Número de pedidos no primeiro mês
R\$ 130,00	15,00%	250
Evolução mês a mês		
#/mês	Número de pedidos	Faturamento com vendas
1	250	R\$ 32.500,00
2	288	R\$ 37.375,00
3	331	R\$ 42.981,25
4	380	R\$ 49.428,44
5	437	R\$ 56.842,70
6	503	R\$ 65.369,11
7	578	R\$ 75.174,47
8	665	R\$ 86.450,65
9	765	R\$ 99.418,24
10	879	R\$ 114.330,98
11	1.011	R\$ 131.480,63
12	1.163	R\$ 151.202,72



MUITO melhor, certo?

Agora, para fechar, vou fazer algo que não deveria.

Vou **mudar a escala do eixo Y direito**, do faturamento, e veja o que acontece:

Ticket médio	Taxa de crescimento mensal (%)	Número de pedidos no primeiro mês
R\$ 130,00	15,00%	250
Evolução mês a mês		
#/mês	Número de pedidos	Faturamento com vendas
1	250	R\$ 32.500,00
2	288	R\$ 37.375,00
3	331	R\$ 42.981,25
4	380	R\$ 49.428,44
5	437	R\$ 56.842,70
6	503	R\$ 65.369,11
7	578	R\$ 75.174,47
8	665	R\$ 86.450,65
9	765	R\$ 99.418,24
10	879	R\$ 114.330,98
11	1.011	R\$ 131.480,63
12	1.163	R\$ 151.202,72



Percebeu a diferença?

Com o mesmíssimo conjunto de dados, eu consegui a aparência de um resultado muito diferente. Como a escala de faturamento, que antes ia até R\$200.000,00 agora vai até R\$750.000,00, o crescimento da curva de faturamento quase desaparece. O número de pedidos, que não foi alterado, parece ter crescido em uma proporção superior!

Mas você pode ver nos números que **ambos obedeceram a mesma taxa de 10%** mês a mês.

O que guardar disto?

Números não mentem. Mas as pessoas que trabalham com números podem mentir!

Você é potencial vítima deste efeito o tempo todo. Em jornais, revistas, blogs e toda outra mídia, todo dado que é apresentado ao público passa pelo filtro de alguém que sabe fazer isto aí, se quiser. Conjuntos de dados em si são desinteressantes para o grande público, mas um recorte adequado pode tirar deles grandes histórias.

3- ACCESS

Microsoft Access (nome completo **Microsoft Office Access**), conhecido por **MSAccess**, é um sistema de gerenciamento de banco de dados da Microsoft, incluído no pacote do Microsoft Office Professional, que combina o Microsoft Jet Database Engine com uma interface gráfica do utilizador (graphical user interface). Ele permite o desenvolvimento rápido de RAD (Rapid Application Development), que envolvem tanto de dados como também a interface a ser utilizada pelos usuários. Microsoft Access é capaz de usar dados guardados em qualquer recipiente de dados compatível com ODBC.

Primeiramente foi o nome de um programa de comunicação da Microsoft, destinado à competir com ProComm e outros programas. Esse produto fracassou e foi abandonado. No segundo semestre de 1992 a Microsoft lançou seu primeiro Sistema de Gerenciamento de Banco de Dados e reusou o nome: o Microsoft Access (MS Access).

Para a Microsoft havia uma grande vantagem quanto ao mercado, como era a dominadora do seu próprio mercado, foi a primeira a lançar um software executável em plataforma Windows, enquanto que os outros programas deste segmento, liderados pela dBase, eram voltados para o ambiente DOS. Ao ser lançado por um preço bastante atrativo o MS Access 1.0 para Windows foi logo tomando conta do seu espaço. Um dos fatores que muito contribuíram para isto foi o seu preço de apenas noventa e nove dólares e também contamos com os investimentos que os administradores da Microsoft fizeram como a compra da Fox Software por cento e setenta milhões de dólares.

Embora as vantagens de lançamentos, faltava de um pacote de ferramenta para desenvolvedores, o que foi princípio para muitas reclamações, assim como a limitação do software de 128MB de memória. Já em 1993 é lançado no mercado MS Access 1.1 com a ampliação para 1GB de memória e junto a ele veio Distribution Kit e os incentivos da para que os desenvolvedores criassem seus aplicativos e os vendessem

sem a necessidade do cliente ter uma versão do Access em seu computador, bastava apenas utilizar o Access Runtime.

No entanto o programa ainda precisava de reajustes e dois anos depois modificações apareceram com a versão 2.0. Agora era possível trabalhar com 254 tabelas ao mesmo tempo e no mesmo arquivo de dados. O novo ambiente de programação dava facilidades ao desenvolvedor e agora também não era mais necessário digitar imensas linhas de códigos. Neste mesmo ano chega ao Brasil a versão em português do Access.

A nova versão do MS Access, em 1995, pulou para 7.0 acompanhando o pacote de programas Microsoft Office 7.0. Introduzida na era de 32 bits a nova versão do Access vinha acompanhada com a linguagem Visual Basic for Applications (VBA), com o seu próprio ambiente de programação. Além do recurso de replicar banco de dados, dando a possibilidade de trabalhar off-line em cópias de banco de dados, podendo ser sincronizada com a base central.

Em março de 1997 foi lançada a versão 8.0, preparada para internet, capaz de gravar hiperlinks e salvar arquivos em formato HTML. E em 1999 lança a versão 9.0, também conhecido como MS Access 2000, com suporte a OLE DB e um produto independente até hoje. Com isso, em 2002, a nova versão com suporte a uma linguagem mais comum, o XML (Extensible Markup Language), que é uma linguagem de marcação de dados e também dá uma maior integralidade com o browser, podendo criar páginas em formato HTML de acesso aos dados do banco.

O Access 2003 na sua versão 11 traz maior integração com browser, além da linguagem VBA, marcação em XML e incorporação da linguagem SQL nas consultas de tabelas do banco.

A partir da versão 2007, a extensão padrão dos arquivos do Access é **.ACCDB**. Arquivos .MDB criados em versões anteriores a 2003 passaram a ser incompatíveis. Para permitir a edição desses arquivos, é necessário a utilização de ferramentas de terceiros, como o MDB Admin, MDB Viewer, entre outros.^[1]

Usos

Geralmente uma aplicação desenvolvida com o Access através da linguagem de programação VBA (Visual Basic for Applications) consiste em dois arquivos: Um que se denomina BackEnd, onde ficam armazenadas todas as tabelas com seus respectivos relacionamentos, e outro denominado FrontEnd, onde ficam armazenados os códigos fontes, formulários, módulos, consultas, macros, etc.

O código fonte pode ser compilado, mas não é possível gerar um executável. Para rodar os aplicativos desenvolvidos é necessário que o usuário possua em sua estação de trabalho o **MSAccess** instalado ou pelo menos o seu **Access Runtime** que vem a ser uma versão semi completa do MSAccess que servirá apenas para rodar os aplicativos sem a possibilidade de desenvolvimento e poderá usar aplicativos criado em Access.

Com o **Microsoft Access** é possível desenvolver desde aplicações simples como por exemplo, um cadastro de clientes, controle de pedidos, até as aplicações mais complexas, como por exemplo, todo o controle operacional, administrativo e financeiro de uma pequena ou até mesmo de uma média ou grande empresa, pois os aplicativos desenvolvidos podem rodar perfeitamente numa rede de computadores e os dados armazenados pelo sistema podem ser publicados na Intranet ou até mesmo na Internet.

Características

Um dos benefícios do Access do ponto de vista do programador é sua relativa compatibilidade com o PHP – buscas podem ser visualizadas e editadas como sendo indicações de SQL, e estes por sua parte podem ser usados diretamente em Macros e Módulos VBA para manipular tabelas do próprio Access. Usuários podem misturar e usar ao mesmo tempo VBA e Macros para formulários de programação e lógica, além de serem oferecidas possibilidades com técnicas de orientação-objeto.

O escritor de relatórios do Access, mesmo sendo capaz da criação sofisticada de relatórios, não é tão potente quanto outro escritor de relatórios de dados chamado Crystal Reports. MSDE (Microsoft SQL Server Desktop Engine) 2000, a mini-versão do MS SQL Server 2000, é incluído com a edição de desenvolvedor do Office XP e pode ser usado ao lado do Access como uma alternativa ao Jet Database Engine.

A função "cortar e colar"(cut and paste) do Access pode torná-lo uma ferramenta útil para a conexão entre diversos bancos de dados (por exemplo, entre o Oracle e o Microsoft SQL Server durante conversões de dados ou bancos de dados).

O Access disponibiliza várias funções de exportação e importação (import and export), que permitem a integração entre o Windows e outras aplicações de plataforma, muitas das quais podem ser executadas dentro das próprias aplicações ou manualmente pelo usuário. Serve, como exemplo, o bastante compacto formato SNP para a compartilhamento de relatórios perfeitamente formatados entre usuários que não dispõem de uma versão completa do Access.

Diferentemente dos RDBMS completos, o motor de banco de dados Jet da Microsoft não dispõe de gatilhos de bancos de dados e procedimentos armazenados (stored procedures).

A partir do MS Access 2000 (Jet 4.0), há um sintaxe que permite a criação de pedidos de busca com parâmetros, semelhante à criação de procedimentos.

O Microsoft Access permite que formulários contenham códigos que são ativados à medida que mudanças são feitas a uma tabela subjacente (contanto que as modificações sejam feitas apenas com aquele formulário), e é comum usar buscas "pass-through" e outras técnicas no Access para ativar procedimentos armazenados em RDBMSs que suportam este tipo de sistema.

Em arquivos de projeto de banco de dados do Access ADP (suportado em MS Access 2000 e adiante), as ferramentas relacionadas aos bancos de dados são completamente diferentes, uma vez que este tipo de arquivo é conectado ao MSDE ou ao Microsoft SQL Server, ao invés de usar o motor Jet. Deste modo, ele suporta a criação de quase qualquer objeto no servidor subjacente (tabelas com restrições e gatilhos, visualizações, procedimentos armazenados e UDF's).

Contudo apenas formulários, relatórios, macros e módulos são armazenados no arquivo ADP (outros objetos são armazenados no banco de dados "back end").

Desenvolvimento

A linguagem de programação disponível no Access é a Microsoft Visual Basic for Applications, igualmente a outros produtos da série Microsoft Office.

Duas bibliotecas de componentes COM de acesso de banco de dados são providenciadas: a legacy Data Access Objects (DAO), apenas disponível com o Access, e o novo ActiveX Data Objects (ADO).

O Microsoft Access é facilmente aplicado à projetos pequenos porém sofre problemas de escala com projetos maiores se as aplicações forem mal desenvolvidas.

Todas as buscas no banco de dados, formulários e relatórios são armazenados no banco de dados, e visando os ideais de um modelo relacional, não há possibilidade de montar uma hierarquia estruturada fisicamente a partir deles.

Uma técnica de desenvolvimento é dividir uma aplicação Access entre dados e programas. Um banco de dados deve conter apenas tabelas e relações, enquanto outro contém todos os programas, formulários, relatórios e pedidos de busca, assim como as conexões para as tabelas do primeiro banco de dados.

Infelizmente, o Access não permite trilhas quando são feitas as conexões, e o ambiente de desenvolvimento deve ter a mesma trilha do ambiente de produção (embora seja possível escrever sua própria rotina de "conexão dinâmica" em VBA, que pode procurar um arquivo "back end" específico através de uma pesquisa dentro da árvore do diretório, se não puder encontrá-lo no caminho atual). Esta técnica também permite que o desenvolvedor divida a aplicação entre arquivos diferentes, sendo então possível uma certa estrutura.

Access 2010 (Service Pack 1)

No total foram 70 melhorias e correções realizadas no SP1 para o Access, entre elas:^[2]

- Adicionado um novo recurso para integrar o conteúdo da comunidade na Galeria de partes do aplicativo;
- Corrigido um problema que ocorre quando você tentar exportar um arquivo do Access para uma pasta de trabalho do Excel;

- Melhorado o desempenho quando for publicar formulários do cliente de acesso que contém imagens incorporadas.

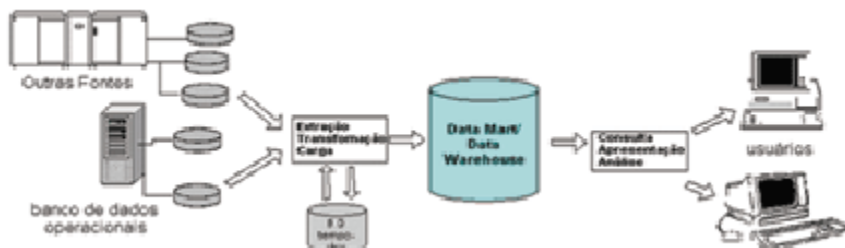
4- PLANEJAMENTO DE BANCO DE DADOS

Uma metodologia para desenvolvimento de data warehouse.

Atualmente, em muitas empresas, o processo manual de produção de informações gerenciais é demorado, dispendioso e cansativo, pois reúne uma grande quantidade de dados que precisam ser coletados de diversas fontes e convertidos em um formato apropriado que possibilite a sua análise. Por isso, criação de um ambiente de data warehouse surge como uma alternativa viável. Seu princípio está na criação de um banco de dados especializado, capaz de manipular grande volume de informações com bom desempenho, melhorando a gerência, o controle e o acesso aos dados. A função do data warehouse é tornar as informações corporativas, obtidas a partir de bancos de dados operacionais e de fontes de dados externas à organização, acessíveis para entendimento e uso das áreas estratégicas.

O projeto de data warehouse é uma tarefa complexa envolvendo um conjunto de conceitos e tecnologias. O sucesso de um projeto de data warehouse está estreitamente relacionado com o entendimento e domínio destes conceitos e tecnologias. A causa principal que resulta em falha e insucesso de um projeto de data warehouse está relacionada à ausência de uma metodologia abrangente capaz de fornecer uma visão geral do processo envolvendo estes conceitos e tecnologias.

O objetivo principal deste estudo é elaborar uma metodologia consistente caracterizada pela sua aplicabilidade prática. Em especial, concentramo-nos na clara identificação e descrição das várias fases do projeto aliada à possibilidade do processo todo ser suportado por uma ferramenta de desenvolvimento.



O Ambiente - o aspecto fundamental na criação de uma data warehouse reside na separação dos dados do ambiente operacional para o ambiente de data warehouse.

Obs: para melhor visualização clique na ilustração.

Metodologia

Embora o desenvolvimento de um data warehouse possua aspectos diferenciados com relação aos sistemas tradicionais, muitas das lições aprendidas no desenvolvimento de sistemas OLTP são de grande valia e devem ser utilizadas no projeto de um sistema de data warehouse. O principal aspecto a ser considerado é a natureza iterativa do desenvolvimento, característica que distingue o ciclo de vida de um projeto de data warehouse de outros projetos de desenvolvimento, e que permite rapidamente liberar partes do banco de dados para o usuário, enquanto outra parte pode estar sendo desenvolvida.

Planejamento é fundamental

O componente de gerência tem a responsabilidade de estabelecer o plano geral do projeto. Este plano deve ser conhecido por todos os membros que farão parte da equipe de desenvolvimento. O plano deve estabelecer prazos, recursos disponíveis e principalmente a expectativa dos usuários. O gerente tem a responsabilidade de estabelecer as principais variáveis do projeto, incluindo: 1) funções que o data warehouse irá disponibilizar; 2) alocação de recursos (máquinas, ferramentas, pessoas); 3) qualidade (definição de prazos não realísticos podem levar a equipe a seguir atalhos e comprometer a qualidade do data warehouse).

Na definição são estabelecidos os objetivos maiores com o intuito de prevenir as constantes mudanças durante as fases do ciclo de desenvolvimento à medida que novos requisitos são identificados, tendo, contudo, o desafio de construir um data warehouse flexível e que tenha a habilidade de absorver as futuras expansões. Esta fase inclui também o entendimento dos conceitos e tecnologias relacionados ao ambiente de inserção do data warehouse, sendo recomendado um planejamento prévio para determinar a escolha da arquitetura e infra-estrutura necessária para possibilitar o pleno desenvolvimento do data warehouse.

Um aspecto importante na definição do projeto é a escolha da abordagem de desenvolvimento. A decisão de usar a estratégia botton up ou top down deve ser

tomada com cuidado. Como modelo foi adotada a abordagem top down para a fase de definição do projeto e botton up para as demais fases que, juntas, representam a fase de desenvolvimento. Para tanto, foi desenvolvido um sistema automatizado apresentado como alternativa ao sistema atual de geração de dados analíticos, eliminando as atuais dificuldades e limitações apresentadas. Neste caso, sugere-se a utilização de ferramentas de desenvolvimento de data warehouse a fim de auxiliar o projetista nas várias fases do projeto. Estas ferramentas estão divididas nas seguintes categorias: servidor de banco de dados IBM DB2 V7.2; ferramenta de Data warehouse IBM DB2 Warehouse Manager V7.2; ferramenta OLAP IBM DB2 OLAP STARTER KIT V7.2. O IBM DB2 Warehouse Manager também oferece um conjunto de recursos que auxiliam na criação dos metadados ao longo do desenvolvimento do data warehouse.



Fases da metodologia de desenvolvimento de data warehouse.

Modelo dimensional no projeto gráfico

A técnica utilizada para a criação do projeto lógico do data warehouse é a da modelagem dimensional. Esta técnica é caracterizada pela criação do esquema estrela a partir do esquema conceitual criado na fase anterior. Esta fase é inteiramente desenvolvida através da utilização de uma ferramenta que suporta a construção do esquema estrela. O Centro de Data warehouse do IBM DB2 guia o projetista através das várias etapas do projeto lógico. A primeira etapa é a definição de um assunto que, por sua vez, compreende um conjunto de processos relacionados a uma área específica do negócio. O objetivo principal de um assunto é a elaboração de um esquema de data warehouse (esquema estrela). Este esquema é construído

gradativamente através dos processos que estão relacionados ao assunto. Um processo tem a finalidade de transformar os dados que estão armazenados nos sistemas fonte, cuja origem dos dados pode derivar de várias bases de dados e podem estar armazenadas em sistemas diferentes.

Um exemplo seria o processo de transformação dos dados de um arquivo texto para uma tabela relacional. A tabela relacional resultante é armazenada no banco de dados do data warehouse e compreende a dimensão tempo do esquema estrela resultante.

Através do uso do centro de data warehouse podemos modelar processos complexos de transformação de dados. Neste caso, não é necessário mostrar as potencialidades e limitações do centro de data warehouse, mas apenas os recursos utilizados na fase de criação. Os passos para a criação da tabela de fatos do esquema estrela é descrito a seguir. A tabela é criada a partir de uma série de processos que são executados em seqüência. O resultado do processo de transformação é a criação de uma tabela que reúne os atributos que compreendem as medidas do fato. Neste exemplo, os atributos compreendem a quantidade de alunos inscritos, classificados, aprovados, não aprovados e suplentes. O processo poderia ser simplificado através de um Join único entre as tabelas de origem. Esta opção foi adotada para exemplificar o uso de vários passos e a possibilidade de uso de tabelas temporárias no processo de transformação dos dados que é bastante comum em ambiente de data warehouse.

Ao final da execução de todos os processos de transformação dos dados fonte, obtemos então o esquema estrela resultante. Este esquema será então utilizado posteriormente por uma ferramenta OLAP, que será utilizada para a construção de aplicações, que serão utilizadas pelos usuários do data warehouse.

O esquema estrela descrito como exemplo compreende as seguintes tabelas: uma tabela de fatos (DW.TB_FATVES) e quatro tabelas dimensionais correspondentes às dimensões Campus (DW.TB_CAM), Regime (DW.TB_REG), Curso (DW.TB_CUR), e Tempo (DW.TB_TIME).

Principais aspectos do projeto físico

Os principais aspectos a serem considerados no projeto físico do data warehouse são: 1) indexação; 2) materialização de visões; 3) particionamento, paralelismo; 4) nível de redundância dos dados; 5) sintonia dos parâmetros do banco de dados. A sintonia do banco de dados é fundamental no ambiente de data warehouse, visto que a natureza

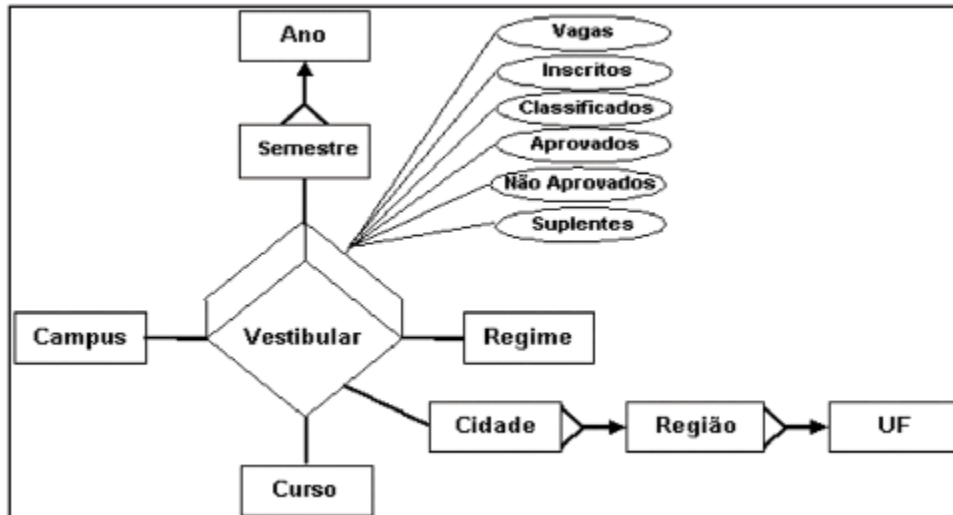
da carga é diferente do ambiente transacional. Os ambientes OLTP são configurados para realizar as transações dos vários usuários simultâneos no menor tempo possível. Os parâmetros de configuração ajustados são responsáveis pela melhora do desempenho em 20 a 25%. Os 75% restantes derivam dos ajustes dos comandos (SQL) de consulta. Isso envolve alterações no projeto físico do banco de dados, disponibilidade e características dos índices, replicação e particionamento de tabelas.

Exemplo de data warehouse com arquitetura centralizada

Como demonstrativo, vamos utilizar o banco de dados operacional de uma universidade, limitando-se ao sistema de concurso vestibular. Para implementar o data warehouse vamos utilizar a arquitetura centralizada. Esta solução poderá posteriormente ser expandida através da incorporação de outros módulos.

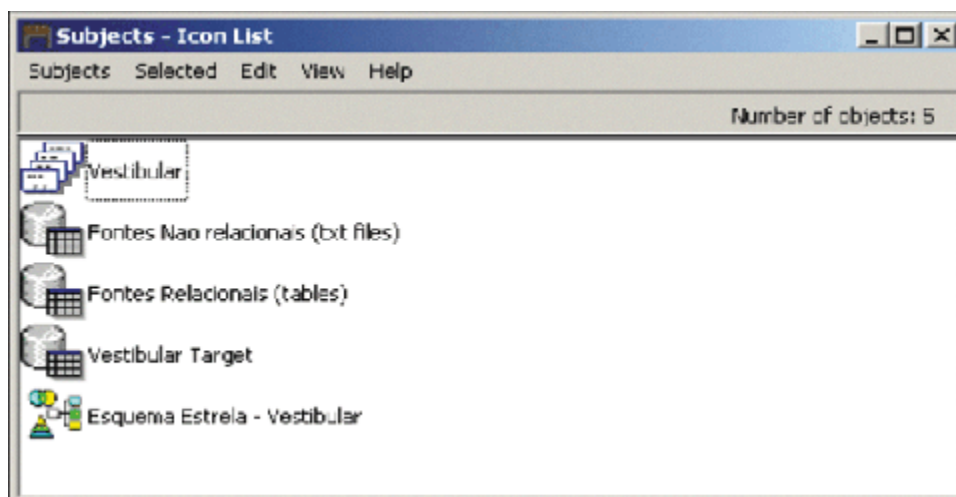
Na modelagem conceitual de data warehouse não basta apenas realizar o levantamento de requisitos dos usuários. Adicionalmente, as estruturas dos bancos de dados operacionais devem ser consideradas. Os requisitos dos usuários e as estruturas dos bancos de dados possuem influência estática e dinâmica, caracterizadas pelas possíveis alterações nos requisitos dos usuários e pela mudança na estrutura do banco de dados em questão. Existem basicamente duas abordagens para obter os requisitos do data warehouse. A primeira alternativa concentra-se mais diretamente no usuário. A segunda alternativa dá maior ênfase aos dados existentes nos sistemas da organização. A fase de levantamento de requisitos deve ter como base as informações do usuário e os dados existentes nos sistemas operacionais.

Ainda como exemplo, vamos utilizar uma extensão ao modelo E/R para o paradigma multidimensional. Neste caso, vamos incluir novos elementos gráficos para estender o modelo E/R. O objetivo é auxiliar o projetista na construção de modelos de dados para data warehouse, contemplando a já conhecida e amplamente difundida modelagem E/R. Entretanto, essa extensão ainda não foi incluída nas ferramentas e produtos disponíveis no mercado. A partir dos requisitos levantados, para atender o fato vestibular, temos um modelo conceitual composto por seis medidas: 1) vagas oferecidas; 2) candidatos inscritos; 3) número de candidatos classificados no vestibular; 4) número de candidatos aprovados no vestibular; 5) candidatos não aprovados no vestibular; 6) número de candidatos suplentes.



Além da tabela de fatos, o esquema conceitual possui cinco dimensões: 1) a dimensão campus, onde a Universidade oferece curso de graduação em vários campus; 2) a dimensão regime, onde um curso pode pertencer ao regime regular (normal) ou especial (período de férias, meses de janeiro, fevereiro e julho); 3) a dimensão curso, sobre os vários cursos oferecidos a cada vestibular; 4) dimensão tempo, referindo-se aos concursos semestrais de vestibular; 5) dimensão cidade, com o objetivo de realizar a estatística sobre a origem dos candidatos.

Implantação de metadados



Pelo menos três outros aspectos importantes merecem ser considerados num projeto de data warehouse: os metadados, com informações sobre o conteúdo que está armazenado no data warehouse. Eles são elaborados gradativamente ao longo de todo

o processo de desenvolvimento. À medida em que cada passo vai sendo criado, os metadados também são elaborados. Ao final do projeto, os metadados podem ser explorados através de uma ferramenta chamada Information Catalog Manager. Esta ferramenta está incluída no Centro de data warehouse do IBM DB2. Ela nos expõe ícones que representam um grupo de informações que podem ser exploradas pelo usuário. Ao clicar no ícone vestibular, por exemplo, a ferramenta abre todos os processos que fazem parte deste assunto. O ícone “fontes relacionais” descreve todas as tabelas fonte que foram utilizadas no processo de criação do data warehouse. Através da exploração dos metadados, os usuários podem encontrar as tabelas que originaram os dados do data warehouse. A granularidade do data warehouse registra em que nível de detalhe os dados estarão disponíveis para a análise do usuário, isto é, determina a sua dimensionalidade, possuindo influência direta no tamanho do banco de dados. A decisão sobre a granularidade dos dados é um dos aspectos mais importantes na construção do data warehouse. A escolha de um nível de granularidade inadequada pode comprometer e até inviabilizar o uso do data warehouse. Neste estudo de caso foi definido o nível de granularidade semestral, ou seja, os dados são agrupados semestralmente através das dimensões Campus, Regime e Curso.

A etapa de atualização de dados deve ser suportada pela ferramenta de desenvolvimento que deve suportar as seguintes atividades: automação do processo de extração, conversão e carga dos dados; definição da periodicidade da atualização e possibilidade de integração com outras ferramentas.

Conclusão

Para concluir, vale dizer que o desenvolvimento de um data warehouse constitui um avanço em relação as metodologias anteriores, pois apresenta uma sistemática mais apropriada baseada na realidade dos sistemas existentes nas empresas. Essa metodologia também valoriza a experiência da equipe no desenvolvimento de sistemas transacionais, pois as fases que a compõem já são largamente utilizadas no desenvolvimento de sistemas OLTP. Também é importante que a metodologia seja suportada por uma ferramenta de desenvolvimento que aumente a produtividade, simplificando e automatizando tarefas complexas no processo de data warehousing. O estudo do caso em questão evidencia algumas questões que merecem uma avaliação mais aprofundada. É o caso da metodologia, considerando diferentes arquiteturas de data warehouse e a exploração detalhada dos níveis conceitual e lógico. Também os

aspectos relacionados à implementação física do data warehouse (particionamento, indexação, materialização de visões) devem ser acrescentados à metodologia.

5- CRIAÇÃO DE BANCO DE DADOS

O processo de criação de banco de dados

Um banco de dados bem-estruturado:

- Economiza espaço em disco ao eliminar dados redundantes.
- Mantém a exatidão e a integridade dos dados.
- Oferece acesso aos dados de maneiras úteis.

Criar um banco de dados eficiente e útil é uma questão de seguir o processo adequado, incluindo as fases a seguir:

1. Análise de requisitos, ou identificação do objetivo do banco de dados
2. Organizando dados em tabelas
3. Especificando chaves primárias e analisando relações
4. Normalizando para padronizar as tabelas

Vamos analisar cada passo mais de perto. Observe que este guia trata do modelo de banco de dados relacional de Edgar Codd, conforme escrito em SQL (em vez de modelos de dados hierárquicos, rede ou objetos). Para saber mais sobre os modelos de bancos de dados, leia nosso guia [aqui](#).

Análise de requisitos: identificando o objetivo do banco de dados

Compreender a finalidade do banco de dados servirá de base para informar suas escolhas durante todo o processo de criação. Certifique-se de considerar o banco de dados de todas as perspectivas. Por exemplo, se você estiver criando um banco de dados para uma biblioteca pública, seria importante considerar as maneiras pelas quais os usuários e bibliotecários precisariam acessar os dados.

Aqui estão algumas maneiras de coletar informações antes de criar o banco de dados:

- Entreviste as pessoas que o usarão
- Analise formulários corporativos, como faturas, quadros de horários e pesquisas
- Faça um pente fino em todos os sistemas de dados existentes (incluindo arquivos físicos e digitais)

Comece reunindo todos os dados existentes que serão incluídos no banco de dados. Em seguida, liste os tipos de dados que você deseja armazenar e as entidades, ou pessoas, coisas, locais e eventos que esses dados descrevem, assim:

Clientes

- Nome
- Endereço
- Cidade, Estado, CEP
- Endereço de e-mail

Produtos

- Nome
- Preço
- Quantidade em estoque
- Quantidade em pedidos

Pedidos

- ID do pedido
- Representante de vendas
- Data
- Produto(s)
- QUANTIDADE

- Preço
- Total

Essas informações mais tarde se tornarão parte do dicionário de dados, que descreve as tabelas e os campos dentro do banco de dados. Certifique-se de dividir as informações em pequenas partes úteis. Por exemplo, considere separar o endereço do país para que você possa filtrar mais tarde as pessoas pelo país de residência. Além disso, evite colocar o mesmo ponto de dados em mais de uma tabela, o que acrescenta complexidade desnecessária.

Depois de saber quais os tipos de dados que o banco de dados incluirá, de onde esses dados vêm e como eles serão usados, você estará pronto para começar a planejar o banco de dados real.

Estrutura de banco de dados: os blocos de construção de um banco de dados

O próximo passo é estabelecer uma representação visual do seu banco de dados. Para fazer isso, você precisa entender exatamente como os bancos de dados relacionais são estruturados.

Dentro de um banco de dados, os dados relacionados são agrupados em tabelas, cada uma das quais consiste em linhas (também chamadas tuplas) e colunas, como uma planilha.

Para converter suas listas de dados em tabelas, comece criando uma tabela para cada tipo de entidade, como produtos, vendas, clientes e pedidos. Eis um exemplo:

Cada linha de uma tabela é chamada de registro. Os registros incluem dados sobre algo ou alguém, como um cliente em particular. Como comparação, as colunas (também conhecidas como campos ou atributos) contêm um único tipo de informação que aparece em cada registro, como os endereços de todos os clientes listados na tabela.

1º nome	Sobrenome	Idade	Código postal
---------	-----------	-------	---------------

Roger	Williams	43	34760
-------	----------	----	-------

Jerrica	Jorgensen	32	97453
---------	-----------	----	-------

Samantha	Hopkins	56	64829
----------	---------	----	-------

Para manter os dados consistentes de um registro para o outro, atribua o tipo de dados apropriado a cada coluna. Os tipos de dados comuns incluem:

- CHAR - um tamanho específico de texto
- VARCHAR - texto de tamanhos variáveis
- TEXT - grandes quantidades de texto
- INT - número inteiro positivo ou negativo
- FLOAT, DOUBLE - também podem armazenar números de pontos flutuantes
- BLOB - dados binários

Alguns sistemas de gestão de banco de dados também oferecem o tipo de dados Autonumeração, que gera automaticamente um número exclusivo em cada linha.

Para criar uma visão geral do banco de dados, conhecido como um diagrama entidade-relacionamento, você não incluirá as tabelas reais. Em vez disso, cada tabela se torna uma caixa no diagrama. O título de cada caixa deve indicar o que os dados nessa tabela descrevem, enquanto os atributos são listados abaixo, assim:

Finalmente, você deve decidir qual atributo ou quais atributos servirão como chave primária para cada tabela, se houver. Uma chave primária (PK) é um identificador exclusivo para uma determinada entidade, o que significa que você poderia escolher um cliente exato, mesmo se você só conhecesse o valor.

Os atributos escolhidos como chaves primárias devem ser únicos, imutáveis e sempre presentes (nunca NULOS ou vazios). Por esta razão, os números de pedido e nomes de usuários são boas chaves primárias, ao contrário de números de telefone ou endereços. Você também pode usar vários campos em conjunto como chave primária (isso é conhecido como uma chave composta).

Quando se trata de criar o banco de dados real, você colocará a estrutura de dados lógicos e a estrutura de dados físicos na linguagem de definição de dados suportada pelo seu sistema de gestão de banco de dados. Nesse ponto, você também deve estimar o tamanho do banco de dados para ter certeza de que você pode obter o nível de desempenho e espaço de armazenamento que serão necessários.

Criando relações entre entidades

Com suas tabelas de banco de dados agora convertidas em tabelas, você está pronto para analisar as relações entre essas tabelas. A cardinalidade se refere à quantidade de elementos que interagem entre duas tabelas relacionadas. Identificar a cardinalidade ajuda a garantir que você tenha dividido os dados em tabelas de forma mais eficiente.

Cada entidade pode potencialmente ter uma relação com todas as outras, mas essas relações normalmente são uma de três tipos:

Relações uma a uma

Quando há apenas uma instância da Entidade A para cada instância da Entidade B, diz-se que elas têm uma relação uma para uma (frequentemente escrito 1:1). Você pode indicar esse tipo de relação em um diagrama ER com uma linha com um traço em cada extremidade:

A menos que você tenha uma boa razão em contrário, uma relação 1:1 geralmente indica que seria melhor combinar os dados das duas tabelas em uma única tabela.

No entanto, convém criar tabelas com uma relação 1:1 sob um determinado conjunto de circunstâncias. Se você tiver um campo com dados opcional, como "descrição", que esteja em branco para muitos dos registros, você pode mover todas as descrições para sua própria tabela, eliminando o espaço vazio e melhorando o desempenho do banco de dados.

Para garantir que os dados correspondam corretamente, você teria então de incluir pelo menos uma coluna idêntica em cada tabela, provavelmente a chave primária.

Relações uma para muitas

Essas relações ocorrem quando um registro em uma tabela é associado com várias entradas em outra tabela. Por exemplo, um único cliente pode ter feito muitos pedidos, ou um cliente pode ter vários livros retirados da biblioteca de uma só vez. As relações uma para muitas (1:M) são indicadas com o que é chamado de "notação pé de galinha", como neste exemplo:

Para implementar uma relação 1:M à medida que você configura um banco de dados, basta adicionar a chave primária do lado "um" da relação como um atributo na outra tabela. Quando uma chave primária é listada em outra tabela desta forma, ela é chamada de chave estrangeira. A tabela no lado "1" da relação é considerada a tabela primária em relação à tabela secundária no outro lado.

Relações muitas para muitas

Quando várias entidades de uma tabela podem ser associadas a várias entidades em outra tabela, diz-se que elas têm uma relação muitas para muitas (M:N). Isso pode acontecer no caso de alunos e aulas, uma vez que um aluno pode ter muitas aulas e uma aula pode ter muitos alunos.

Em um diagrama ER, essas relações são retratadas com estas linhas:

Infelizmente, não é diretamente possível implementar esse tipo de relação em um banco de dados. Em vez disso, você tem que dividi-lo em duas relações uma para muitas.

Para fazer isso, crie uma nova entidade entre essas duas tabelas. Se a relação M:N existir entre vendas e produtos, você pode chamar essa nova entidade de "produtos_vendidos", uma vez que ela mostraria o conteúdo de cada venda. As tabelas de vendas e produtos teriam uma relação 1:M com produtos_vendidos. Esse tipo de entidade intermediária é chamada de tabela de ligação, entidade associativa ou tabela de junção em vários modelos.

Cada registro na tabela de ligação corresponderia às duas entidades nas tabelas vizinhas (também pode incluir informações suplementares). Por exemplo, uma tabela de ligação entre alunos e aulas pode ter esta aparência:

Obrigatório ou não?

Outra maneira de analisar as relações é considerar qual lado da relação tem que existir para que o outro exista. O lado não obrigatório pode ser marcado com um círculo sobre linha onde estaria um traço. Por exemplo, um país tem que existir para que ele tenha um representante nas Nações Unidas, mas o oposto não é verdade:

Duas entidades podem ser mutuamente dependentes (uma não poderia existir sem a outra).

Relações recursivas

Às vezes, uma tabela aponta para si mesma. Por exemplo, uma tabela de funcionários pode ter um atributo "gerente" que se refere a outro indivíduo na mesma tabela. Isso é chamado de relação recursiva.

Relações redundantes

Uma relação redundante é aquela expressa mais de uma vez. Normalmente, é possível remover uma das relações sem perder nenhuma informação importante. Por exemplo, se uma entidade "alunos" tem uma relação direta com outra entidade chamada "professores", mas também tem uma relação com os professores indiretamente através de "aulas", seria bom remover a relação entre "alunos" e "professores". É melhor excluir essa relação, porque a única maneira que os alunos são atribuídos aos professores é através das aulas.

Normalização de banco de dados

Depois de ter um design preliminar para o banco de dados, você pode aplicar regras de normalização para certificar-se de que as tabelas estão estruturadas corretamente. Pense nestas regras como padrões da indústria.

Dito isto, nem todos os bancos de dados são bons candidatos para a normalização. Em geral, os bancos de dados de processamento de transações on-line (OLTP, na abreviação em inglês), nos quais os usuários se preocupam com a criação, a leitura, a atualização e a exclusão de registros, devem ser normalizados.

Bancos de dados de processamento analítico on-line (OLAP, na sigla em inglês), que favorecem a análise e a elaboração de relatórios podem ser melhorados com um certo grau de desnormalização, uma vez que a ênfase é na velocidade de cálculo. Estes incluem aplicativos de suporte à decisão onde os dados precisam ser analisados rapidamente, mas não alterados.

Cada formulário, ou nível de normalização, inclui as regras associadas com os formulários inferiores.

Primeiro formulário normal

O primeiro formulário normal (abreviado como 1NF) especifica que cada célula na tabela pode ter apenas um valor, nunca uma lista de valores, portanto, uma tabela como esta é incorreta:

ID do produto	Cor	Preço
1	marrom, amarelo	US\$ 15
2	vermelho, verde	US\$ 13
3	azul, laranja	US\$ 11

Você pode ser tentado a contornar isso dividindo esses dados em colunas adicionais, mas isso também é contra as regras: uma tabela com grupos de atributos repetidos ou estreitamente relacionados não atende ao primeiro formulário normal. A tabela abaixo, por exemplo, está incorreta:

Em vez disso, divida os dados em várias tabelas ou registros até que cada célula contenha apenas um valor e não haja colunas extras. Nesse ponto, diz-se que os dados são atômicos, ou divididos até o menor tamanho útil. Para a tabela acima, você poderia criar uma tabela adicional chamada "Detalhes de vendas" que corresponderia a produtos específicos com vendas. "Vendas" então teria uma relação 1:M com "Detalhes de vendas".

Segundo formulário normal

O segundo formulário normal (2NF) diz que cada um dos atributos deve ser totalmente dependente de toda a chave primária. Isso significa que cada atributo deve depender diretamente da chave primária, e não indiretamente, através de algum outro atributo.

Por exemplo, um atributo "idade" que depende de "data de nascimento", que por sua vez depende de "ID do aluno", é dito como tendo uma dependência funcional parcial, e uma tabela contendo esses atributos não cumpriria o segundo formulário normal.

Além disso, uma tabela com uma chave primária composta de vários campos viola o segundo formulário normal se um ou mais dos outros campos não dependerem de cada parte da chave.

Assim, uma tabela com esses campos não atenderia ao segundo formulário normal, porque o atributo "nome do produto" depende de ID do produto, mas não do número do pedido:

- Número do pedido (chave primária)
- ID do produto (chave primária)
- Nome do produto

Terceiro formulário normal

O terceiro formulário normal (3NF) adiciona a essas regras a exigência de que cada coluna sem chave seja independente de qualquer outra coluna. Se a alteração de um

valor em uma coluna sem chave faz com que outro valor seja alterado, essa tabela não atende ao terceiro formulário normal.

Isso evita que você armazene quaisquer dados derivados na tabela, como a coluna "imposto" abaixo, que depende diretamente do preço total do pedido:

Pedido	Preço	Imposto
14325	US\$ 40,99	US\$ 2,05
14326	US\$ 13,73	US\$ 0,69
14327	US\$ 24,15	US\$ 1,21

Formulários adicionais de normalização foram propostos, incluindo o formulário normal de Boyce-Codd, o quarto a sexto formulários normais, e o formulário normal de domínio-chave, mas os três primeiros são os mais comuns.

Embora esses formulários expliquem as práticas recomendadas a serem seguidas em geral, o grau de normalização depende do contexto do banco de dados.

Diagramar é rápido e fácil com o Lucidchart. Faça uma avaliação gratuita hoje mesmo para começar a criar e colaborar.

Dados multidimensionais

Alguns usuários podem querer acessar várias dimensões de um único tipo de dados, especialmente em bancos de dados OLAP. Por exemplo, eles podem querer saber as vendas por cliente, estado e mês. Nesta situação, é melhor criar uma tabela central de fatos que outras tabelas de clientes, estados e meses podem consultar, como esta:

Regras de integridade de dados

Você também deve configurar seu banco de dados para validar os dados de acordo com as regras apropriadas. Muitos sistemas de gestão de banco de dados, como o Microsoft Access, aplicam algumas dessas regras automaticamente.

A regra de integridade da entidade diz que a chave primária nunca pode ser NULA. Se a chave é composta de várias colunas, nenhuma delas pode ser NULA. Caso contrário, ele poderia não conseguir identificar o registro de forma exclusiva.

A regra de integridade referencial requer que cada chave estrangeira listada em uma tabela seja correspondida com uma chave primária na tabela que faz referência. Se a chave primária mudar ou for excluída, essas alterações precisarão ser implementadas sempre que essa chave for referenciada em todo o banco de dados.

As regras de integridade da lógica de negócios garantem que os dados se encaixem dentro de certos parâmetros lógicos. Por exemplo, o horário de consulta teria de estar dentro do horário comercial normal.

Adicionando índices e exibições

Um índice é essencialmente uma cópia classificada de mais uma ou colunas, com os valores em ordem ascendente ou descendente. Adicionar um índice permite aos usuários localizar registros mais rapidamente. Em vez de reclassificar para cada consulta, o sistema pode acessar registros na ordem especificada pelo índice.

Embora os índices acelerem a recuperação de dados, eles podem diminuir a inserção, atualização e exclusão, uma vez que o índice precisa ser reconstruído sempre que um registro é alterado.

Uma exibição é simplesmente uma consulta salva sobre os dados. Eles podem unir dados úteis de várias tabelas ou mostrar parte de uma tabela.

Propriedades estendidas

Depois de concluir o layout básico, é possível refinar o banco de dados com propriedades estendidas, como texto instrucional, máscaras de entrada e regras de formatação que se aplicam a um determinado esquema, exibição ou coluna. A vantagem é que, como essas regras são armazenadas no próprio banco de dados, a apresentação dos dados será consistente entre os vários programas que acessam os dados.

SQL e UML

A UML (Unified Modeling Language) é outra forma visual de expressar sistemas complexos criados em uma linguagem orientada a objetos. Vários dos conceitos mencionados neste guia são conhecidos na UML sob diferentes nomes. Por exemplo, uma entidade é conhecida como uma classe na UML.

A UML não é usada tão frequentemente hoje como era antes. Hoje em dia, ela é frequentemente usada academicamente e em comunicações entre projetistas de software e seus clientes.

Sistemas de gestão de bancos de dados

Muitas das escolhas de design que você fará dependem de qual sistema de gestão de banco de dados você usa. Alguns dos sistemas mais comuns incluem:

- Oracle DB
- MySQL
- Microsoft SQL Server

- PostgreSQL
- IBM DB2

Quando houver escolha, selecione um sistema de gestão de banco de dados apropriado baseado em custos, sistemas operacionais, recursos e muito mais.

REFERÊNCIAS

<https://support.microsoft.com/pt-br/office/use-atingir-meta-para-encontrar-o-resultado-desejado-ajustando-um-valor-de-entrada-320cb99e-f4a4-417f-b1c3-4f369d6e66c7><acesso em 04/12/2023>

<https://rockcontent.com/br/blog/como-usar-excel/><acesso em 04/12/2023>

https://pt.wikipedia.org/wiki/Microsoft_Access<acesso em 04/12/2023>

<https://www.batebyte.pr.gov.br/Pagina/Como-organizar-um-banco-de-dados-moderno-e-eficiente><acesso em 06/12/2023>

<https://www.lucidchart.com/pages/pt/tutorial-de-criacao-e-estruturacao-de-banco-de-dados><acesso em 06/11/2023>