* Criando nova base de dados

create database livraria

character set utf8mb4

collate utf8mb4\_0900\_ai\_ci;

* Create table

create table editora (

ideditora int not null auto\_increment,

nomeeditora varchar(45) not null,

primary key (ideditora));

-- SQL MAX and MIN

-- Problem: Find the cheapest product

SELECT \*,UnitPrice

FROM Product ORDER BY UNITPRICE ASC;

SELECT MIN(UnitPrice)

FROM Product;

-- Problem: Find the largest order placed in 2014

SELECT MAX(TotalAmount)

FROM cliente2.order

WHERE YEAR(OrderDate) = 2012

SELECT MAX(TotalAmount)

FROM Order

WHERE YEAR(OrderDate) = 2012

-- Entre colchete apenas no SQL SERVER e se retirar vai entender que eh palavra reservada do Order by e vai dar erro.

SELECT MAX(TotalAmount)

FROM cliente2.Order

WHERE YEAR(OrderDate) = 2012;

-- Para resolver, colocar antes da tabela o nome do banco

-- SQL SELECT COUNT, SUM, and AVG

SELECT COUNT(Id)

FROM Customer;

SELECT SUM(TotalAmount) as soma

FROM cliente2.Order

WHERE YEAR(OrderDate) = 2012;

SELECT AVG(TotalAmount) as media

FROM cliente2.Order;

-- SQL WHERE AND, OR, NOT Clause

SELECT Id, FirstName, LastName, City, Country

FROM Customer

WHERE FirstName = 'Ana' AND LastName = 'rosa';

SELECT Id, FirstName, LastName, City, Country

FROM Customer

WHERE Country = 'Spain' or Country = 'France';

SELECT Id, FirstName, LastName, City, Country

FROM Customer

WHERE NOT Country = 'USA';

-- The SQL WHERE IN

-- Problem: List all suppliers from the USA, UK, OR Japan

SELECT Id, CompanyName, City, Country

FROM Supplier

WHERE Country IN ('USA', 'UK', 'Japan');

-- Problem: List all products that are not exactly $10, $20, $30, $40, or $50

SELECT Id, ProductName, UnitPrice

FROM Product

WHERE UnitPrice NOT IN (10,20,30,40,50);

-- Problem: List all orders that are between $50 and $15000

SELECT Id, OrderDate, CustomerId, TotalAmount

FROM cliente2.Order

WHERE (TotalAmount >= 50 AND TotalAmount <= 15000)

ORDER BY TotalAmount DESC;

-- Problem: List all orders that are not between $50 and $15000

SELECT Id, OrderDate, CustomerId, TotalAmount

FROM cliente2.Order

WHERE NOT (TotalAmount >= 50 AND TotalAmount <= 15000)

ORDER BY TotalAmount DESC;

-- SQL WHERE BETWEEN

-- Problem: List all products between $10 and $20

SELECT Id, ProductName, UnitPrice

FROM Product

WHERE UnitPrice BETWEEN 10 AND 20

ORDER BY UnitPrice;

SELECT Id, ProductName, UnitPrice

FROM Product

WHERE UnitPrice NOT BETWEEN 5 AND 100

ORDER BY UnitPrice;

-- Problem: Get the number of orders and amount sold between Jan 1, 2013 and Jan 31, 2013.

SELECT COUNT(Id) as qtvendas , SUM(TotalAmount) as valortotal

FROM cliente2.Order

WHERE OrderDate BETWEEN '2010/01/01' AND '2013/12/31';

-- SQL WHERE LIKE

-- Problem: List all products with names that start with 'Ca'

SELECT Id, ProductName, UnitPrice, Package

FROM Product

WHERE ProductName LIKE %ca%;

-- Problem: List all products that start with 'Cha' or 'Chan' and have one more character.

SELECT Id, ProductName, UnitPrice, Package

FROM Product

WHERE ProductName LIKE 'Cha\_' OR ProductName LIKE 'Chan\_';

-- SQL WHERE IS NULL

-- Problem: List all suppliers that have no fax number

SELECT Id, CompanyName, Phone, Fax

FROM Supplier

WHERE Fax IS NULL;

-- Problem: List all suppliers that do have a fax number

SELECT Id, CompanyName, Phone, Fax

FROM Supplier

WHERE Fax IS NOT NULL;

-- SQL GROUP BY

-- Problem: List the number of customers in each country.

select id, country, firstname from Customer;

SELECT Country , count(country) as quantPais

FROM Customer

GROUP BY Country;

-- SQL Alias

-- Problem: List total customers in each country. Display results with easy to understand column headers.

SELECT C.Country AS Nation, COUNT(C.Id) AS TotalCustomers

FROM Customer C

GROUP BY C.Country;

-- Problem: List the number of customers in each country sorted high to low

SELECT Country , COUNT(Id) as numberclients

FROM Customer

GROUP BY Country

ORDER BY COUNT(Id) asc;

select \* from cliente2.Order;

-- Problem: Trazer numero de vendas e soma de vendas por cliente

SELECT customerid, COUNT(Id) as qtvendas , SUM(TotalAmount) as valortotal, AVG(TotalAmount) as mediavendas, MIN(TotalAmount) as menorvenda, MAX(TotalAmount) as maiorvenda

FROM cliente2.Order

WHERE OrderDate BETWEEN '2010/01/01' AND '2013/12/31'

GROUP BY customerid

order by qtvendas desc;

-- INNER JOIN

-- Vamos gerar o modelo de dados do banco cliente2 como apoio as consultas com join

select \*

from cliente2.customer C JOIN cliente2.order O

ON C.id=O.customerid;

SELECT C.firstname, C.lastname, O.totalamount

FROM cliente2.customer C INNER JOIN cliente2.order O

ON C.id = O.customerid;

SELECT OrderNumber, TotalAmount, FirstName, LastName, City, Country

FROM cliente2.Order JOIN Customer

ON cliente2.Order.CustomerId = cliente2.Customer.Id;

-- OU UMA OUTRA FORMA DE RODAR O MESMO CODIGO SERIA COM A UTILIZACAO DOS ALIAS, CONFORME DEMONSTRADO NA VIDEO AULA

select o.ordernumber, o.totalamount, c.firstname,c.lastname, c.city,c.country

from cliente2.order O INNER JOIN cliente2.customer C

ON O.customerid = C.ID;

-- E AGORA SE QUISEEMOS TRAZER DADOS DE MAIS TABELAS

SELECT O.OrderNumber, OrderDate AS Datetime,

P.ProductName, I.Quantity, I.UnitPrice

FROM cliente2.Order O

INNER JOIN OrderItem I ON O.Id = I.OrderId

INNER JOIN Product P ON P.Id = I.ProductId

ORDER BY O.OrderNumber;

SELECT O.OrderNumber, CONVERT(date,O.OrderDate) AS Data,

P.ProductName, I.Quantity, I.UnitPrice

FROM cliente2.Order O

JOIN OrderItem I ON O.Id = I.OrderId

JOIN Product P ON P.Id = I.ProductId

ORDER BY O.OrderNumber;

-- A FUNCAO CONVERT no SQL SERVER, primeiro vem para qual tipo quer converter, mas se rodar desta forma no musql da errp.

SELECT O.OrderNumber, CONVERT(O.OrderDate, date) AS Data,

P.ProductName, I.Quantity, I.UnitPrice

FROM cliente2.Order O

JOIN OrderItem I ON O.Id = I.OrderId

JOIN Product P ON P.Id = I.ProductId

ORDER BY O.OrderNumber;

-- A mesma funcao no MYSQL, mas deve invester a order dos parametros da funcao CONVERT

-- Problema: Listas o total de pedidos ordenado dos maiores para menores pedidos

SELECT C.FirstName, C.LastName, SUM(O.TotalAmount) AS SOMA

FROM cliente2.Order O

INNER JOIN Customer C ON O.CustomerId = C.Id

GROUP BY C.FirstName, C.LastName

ORDER BY SOMA DESC;

-- LEFT JOIN

-- Problema: liste todos os clientes, independentemente de terem feito pedidos ou não

SELECT c.FirstName, c.LastName, c.City, c.Country, o.OrderNumber, o.TotalAmount

FROM Customer C LEFT JOIN cliente2.Order O

ON O.CustomerId = C.Id

ORDER BY TotalAmount;

-- RIGHT JOIN

-- Problema: Liste os clientes que não fizeram pedidos

SELECT FirstName, LastName, City, Country, TotalAmount

FROM cliente2.Order O RIGHT JOIN Customer C

ON O.CustomerId = C.Id

WHERE TotalAmount IS NULL;

-- SQL UNION

-- Problema: Liste todas as empresas, incluindo fornecedores e clientes.

SELECT 'Customer' As Type,

FirstName + ' ' + LastName AS ContactName,

City, Country, Phone

FROM Customer

UNION

SELECT 'Supplier',

ContactName, City, Country, Phone

FROM Supplier;

-- Mesmo comando que o SQL SERVER, nao da erro mas repare quanto tento concatenar as strings firstname e lastiname com espaco.

-- usar no mysql a funcao CONCAT

SELECT 'Customer' As Type,

CONCAT( FirstName, ' ', LastName ) AS ContactName,

City, Country, Phone

FROM Customer

UNION

SELECT 'Supplier',

ContactName, City, Country, Phone

FROM Supplier;

-- SQL SUBQUERIE

-- Problema: liste produtos com quantidades de pedido maiores que 10.

SELECT ProductName

FROM Product

WHERE Id IN (1,2);

SELECT ProductName

FROM Product P

WHERE Id IN (SELECT O.ProductId

FROM OrderItem O

WHERE Quantity > 10);

-- Problema: Liste todos os clientes com seu número total de pedidos

SELECT c.ID, FirstName, LastName,

OrderCount = (SELECT COUNT(O.Id)

FROM cliente2.Order O

WHERE O.CustomerId = C.Id)

FROM Customer C ;

-- NO SQL SERVER poderia ser assim, mas no MYSQL vai dar erro devido a criacao do campo ORDERCOUNT

-- ira para cada cliente, ler a tabela de ordens e vai veirificar a quantidade de pedidos.

SELECT FirstName, LastName,

(SELECT avg

FROM cliente2.Order O

WHERE O.CustomerId = C.Id) as OrderCount

FROM Customer C ;

-- SQL EXISTS SUBQUERIE

-- Problema: Encontre fornecedores com produtos acima de $ 100.

SELECT CompanyName

FROM Supplier

WHERE EXISTS

(SELECT 1 -- aqui como nao vou trazer productname poderia colocar valor 1

FROM Product

WHERE Product.SupplierId = Supplier.Id

AND UnitPrice > 100) ;

-- Problema: E com NOT EXIST, posso encontrar fornecedores com produtos abaixo e igual de $ 100.

SELECT CompanyName

FROM Supplier

WHERE NOT EXISTS

(SELECT ProductName

FROM Product

WHERE Product.SupplierId = Supplier.Id

AND UnitPrice > 100) ;

-- SQL HAVING

-- Problema: liste o número de clientes em cada país, exceto os EUA, classificados do alto para o baixo. Inclui apenas países com 9 ou mais clientes.

SELECT Country , COUNT(Id) as qt

FROM Customer

WHERE Country <> 'USA'

GROUP BY Country

HAVING qt >= 9

ORDER BY qt DESC;

-- Problema: liste todos os clientes com pedidos médios entre $ 1000 e $ 1200.

SELECT FirstName, LastName, AVG(TotalAmount) as media

FROM cliente2.Order O

JOIN Customer C ON O.CustomerId = C.Id

GROUP BY FirstName, LastName

HAVING AVG(TotalAmount) BETWEEN 1000 AND 1200;

-- SQL SELECT INTO

-- Problema: Copie todos os fornecedores dos EUA para uma nova tabela SupplierUSA.

SELECT \* INTO SupplierUSA

FROM Supplier

WHERE Country = 'USA';

-- NO SQL SERVER, voce consegue crar uma tabela inexistente com select into mas no MYSQL nao funciona

CREATE TABLE SupplierUSA SELECT \* FROM Supplier WHERE Country = 'USA';

-- Para criar uma tabela nova no MYSQL a partir de dados vindos de outra tabela

select \* from SupplierUSA;

select \* from Supplier;

drop table SupplierUSA;

-------------------------------

------ UPDATES

-------------------------------

select \* from Supplier where Id = 2;

UPDATE Supplier

SET City = 'Oslo',

Phone = '(0)1-953530',

Fax = '(0)1-953555'

WHERE Id = 2;

select \* from Supplier where Id = 2;

-- TRANSACOES

select \* from Product;

-- begin transaction, so funciona no SQL SERVER

-- Para iniciar uma transação, você usa a instrução START TRANSACTION. O BEGIN é o apelido de START TRANSACTION.

START TRANSACTION;

UPDATE Product

SET IsDiscontinued = 0

where IsDiscontinued=1;

select \* from Product;

-- SE AO RODAR RECEBER ESTE ERRO:

-- Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column.

-- Para nao dar este erro, no where teria que usar o campo de chave primaria ou incluir o SET abaixo para desligar esta restricao e depois ligar

-- Poderiamos forcar tambem no where passar uma busca com a PK da tabela, desta forma, desde que nao tenha valor de pk com 0

START TRANSACTION;

UPDATE Product

SET IsDiscontinued = 0

where IsDiscontinued=1 and id<>0;

select \* from Product;

-- para desfazer antes de gravar definitivamente, rodar ROLLBACK denteo da TRANSACAO

ROLLBACK;

select \* from Product;

-- E AGORA DE UMA FORMA DIFERENTE SEM COLOCAR A PK NO WHERE

SET SQL\_SAFE\_UPDATES = 0;

START TRANSACTION;

UPDATE Product

SET IsDiscontinued = 0

where IsDiscontinued=1;

select \* from Product;

SET SQL\_SAFE\_UPDATES = 1;

-- Rollback tran Funnciona apenas no SQL SERBER

-- Para MYSQL apenas ROLLBACK ou COMMIT

ROLLBACK;

select \* from Product;

-- UMA OUTRA FORMA ALTERNATIVA PARA RESOLVER DE FORMA GERAL PARA TODOS OS COMANDOS QUE CHEGAM, É NO WORKBENCH MUDAR O PARAMETRO

-- MySQL Workbench => [Edit] => [Preferences] => [SQL EDITOR]. NA ULTIMA OPCAO DESLIGAR E DAR RESTART NO SERVIC MYSQL.

START TRANSACTION;

UPDATE Product

SET IsDiscontinued = 0

where IsDiscontinued=1;

select \* from Product;

-- ALTERAR O PARAMETRO no workbench, ABRIR SERVICOS DO WINDOWS, PARAR O MYSQL E DAR START E RODAR NOVAMENTE

START TRANSACTION;

UPDATE Product

SET IsDiscontinued = 0

where IsDiscontinued=1;

select \* from Product;

ROLLBACK;

-- --------

-- POSSO USAR APENAS O BEGIN AO INVES DO START TRANSACTON, AGORA NAO PRECISO MAIS COLOCAR A PK NO WHERE OU USAR SQL\_SAFE\_UPDATES = 0 PARA NAO DAR ERRO

BEGIN ;

UPDATE Product

SET IsDiscontinued = 1, ProductName = 'TESTE'

WHERE UnitPrice = 97.00;

select \* from Product;

COMMIT;

select \* from Product;

-------------------------------

------ DELETES

-------------------------------

select \* from orderitem;

BEGIN;

DELETE FROM orderitem ;

select \* from orderitem;

ROLLBACK;

select \* from orderitem;

-------------------------------

------ TRUNCATE TABLE

-------------------------------

CREATE TABLE neworderitem SELECT \* FROM orderitem;

SELECT \* FROM neworderitem;

TRUNCATE TABLE neworderitem;

SELECT \* FROM neworderitem;

-------------------------------

------ INSERT INTO

-------------------------------

INSERT INTO Customer (id, FirstName, LastName, City, Country, Phone)

VALUES (100,'Craig', 'Smith', 'New York', 'USA', '1-01-993 2800');

select \* from Customer;

-- vamos inserir algumas linhas na tabela abaixo, comecando proximo valor livre para pk id

-- Podemos dar insert com resultado de um select e vamos aprender mais duas functions

INSERT INTO Customer (FirstName, LastName, City, Country, Phone)

SELECT LEFT(ContactName, 5),

SUBSTRING(ContactName, 2, 2),

City, Country, Phone

FROM Supplier

WHERE CompanyName = 'Bigfoot Breweries';

-- FUNCIONOU?

-- vamos alterar a coluna id que é uma PK em customer. Vamos alterar para colocar a caracteristica auto incremental na columa

ALTER TABLE Customer MODIFY id INTEGER NOT NULL AUTO\_INCREMENT;

-- se der erro e uma mensagem que nao posso alterar a coluna id por ser uma fk de outra tabela

-- Error Code: 1833. Cannot change column 'id': used in a foreign key constraint 'fk\_order\_customer1' of table 'cliente2.order'

-- desabilite na sessao a fk. O ideal eh fazer sem o sistema estar no ar para evitar qualquer problema de integridade referencial no milisegundo que esta sendo desligado

SET FOREIGN\_KEY\_CHECKS = 0;

ALTER TABLE `cliente2`.`customer` CHANGE COLUMN `id` `id` INT NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT = 200;

SET FOREIGN\_KEY\_CHECKS = 1;

-- agora ja poder rodar o mesmo comando que o id sera gerado a partir do valor 102

INSERT INTO Customer (FirstName, LastName, City, Country, Phone)

SELECT LEFT(ContactName, 5),

SUBSTRING(ContactName, 11, 3),

City, Country, Phone

FROM Supplier

WHERE CompanyName = 'Bigfoot Breweries';

select \* from Customer;