CLASSIC MODEL DATABSE

Introduction: Exploring the Classic Model Database
Welcome to the Classic Model Database, a rich and comprehensive dataset designed to capture and analyze data from a classic model business scenario. This relational database provides a structured representation of various aspects of a business, including customers, employees, offices, orders, payments, and products.

Objectives:

Throughout our exploration, we aim to derive meaningful insights by posing diverse queries and analyses on this dataset. From understanding customer credit limits to identifying top-performing products, our exploration will provide valuable perspectives on the business model represented in this classic dataset.

DATABASE

1. Find the total number of customers in the database.

```sql
SELECT COUNT(*) AS total_customers
FROM classicmodels.customers;
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

total_customers
122

2. Find the average credit limit of customers in each country.

```sql
SELECT country , AVG(creditLimit) AS AvgCriditLimit
FROM classicmodels.customers
GROUP BY country;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country	AvgCreditLimit
France	77691.666667
USA	78102.777778
Australia	86060.000000
Norway	91200.000000
Poland	0.000000
Germany	19776.923077
Spain	73971.428571
Sweden	84750.000000
Denmark	102100.000000
Singapore	67233.333333
Portugal	0.000000

3. Retrieve the details of the products with the highest quantity in stock.

```sql
SELECT * FROM classicmodels.products
ORDER BY quantityInStock DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

productCode	productName	productLine	productScale	productVendor	productDescription	quantityIn
S12_2823	2002 Suzuki XREO	Motorcycles	1:12	Unimax Art Galleries	Official logos and insignias, saddle bags located ...	9997
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Find the product lines with the highest and lowest average buy prices.

```sql
SELECT productLine, AVG(buyPrice) AS  AvgBuyPrice
FROM classicmodels.products
GROUP BY productLine
ORDER BY AvgBuyPrice DESC/ASC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch row

productLine	AvgBuyPrice
Classic Cars	64.446316

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

productLine	AvgBuyPrice
Trains	43.923333

5. List the names and email addresses of employees who report to a specific manager (reportsTo).

```sql
SELECT firstName , lastName , email
FROM classicmodels.employees
WHERE reportsTo = 1056;
```

	firstName	lastName	email
▶	William	Patterson	wpatterson@classicmodelcars.com
	Gerard	Bondur	gbondur@classicmodelcars.com
	Anthony	Bow	abow@classicmodelcars.com
	Mami	Nishi	mnishi@classicmodelcars.com

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 𝐈A

6. Find the top 3 customers with the highest credit limits.

```
SELECT customerName, creditLimit
FROM classicmodels.customers
ORDER BY creditLimit DESC
Limit 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customerName	creditLimit
Euro + Shopping Channel	227600.00
Mini Gifts Distributors Ltd.	210500.00
Vida Sport, Ltd	141300.00

7. List the employees who have not reported to anyone.

SELECT emp.*
FROM classicmodels.employees emp
WHERE emp.reportsTo IS NULL;

	employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
▶	1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1	NULL	President
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

8. Find the customers who have not made any payments.

```sql
SELECT c.customerName
FROM classicmodels.customers c
LEFT JOIN classicmodels.payments p ON
c.customerNumber = p.customerNumber
WHERE p.customerNumber IS NULL;
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

customerName
Havel & Zbyszek Co
American Souvenirs Inc
Porto Imports Co.
Asian Shopping Network, Co
Natürlich Autos
ANG Resellers
Messner Shopping Network
Franken Gifts, Co
BG&E Collectables
Schuyler Imports
Der Hund Imports

9. List the product codes and names of products that have not been ordered.

```sql
SELECT p.productCode, p.productName
FROM classicmodels.products p
LEFT JOIN classicmodels.orderdetails od ON
p.productCode = od.productCode
WHERE od.productCode IS NULL;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:

	productCode	productName
▶	S18_3233	1985 Toyota Supra

10. Find the average order quantity for each product line.

```sql
SELECT pr.productLine, AVG(od.quantityOrdered) AS AvgOrderQuantity
FROM classicmodels.orderdetails od
JOIN classicmodels.products pr ON od.productCode = pr.productCode
GROUP BY pr.productLine;
```

Result Grid			Filter Rows:		Export:		Wrap Cell Content:

productLine	AvgOrderQuantity
Classic Cars	35.2297
Motorcycles	35.5933
Planes	35.3333
Ships	34.8245
Trains	34.7901
Trucks and Buses	35.7175
Vintage Cars	34.9056