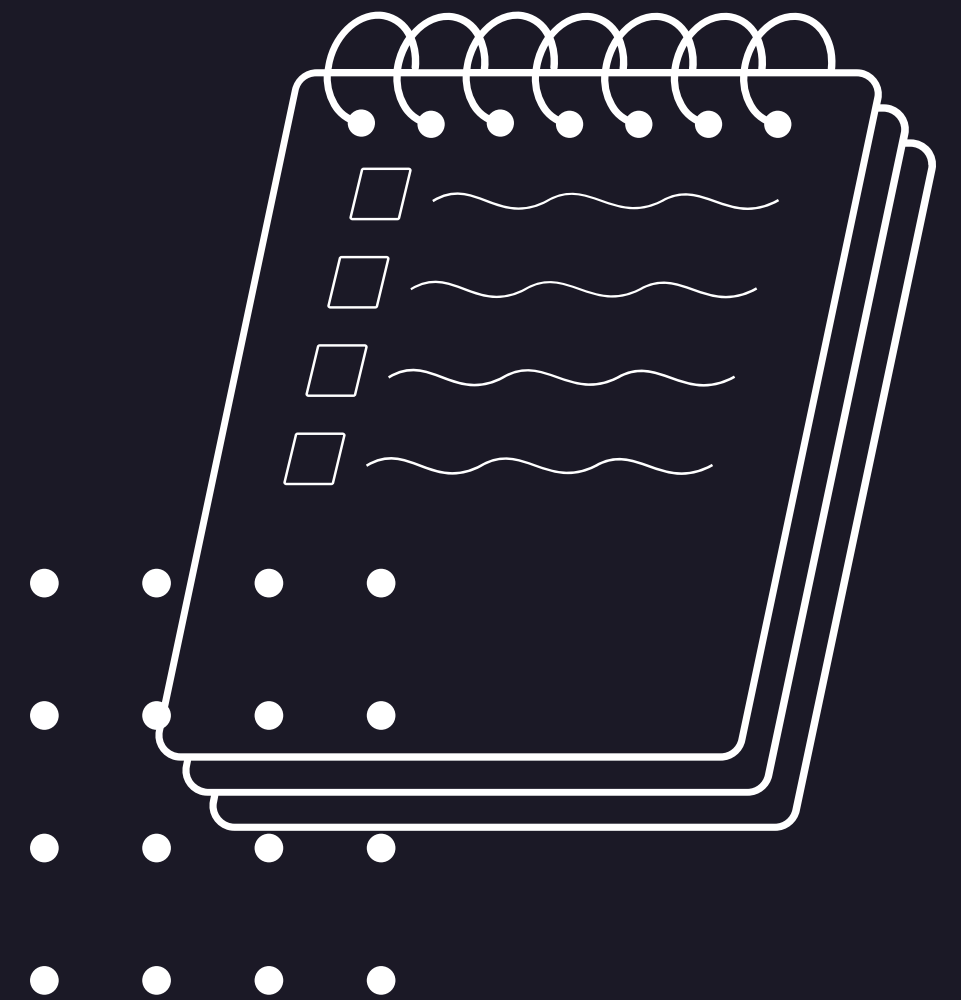




# ABOUT

MariaDB was first released in late 2009 to permanently consolidate the MySQL code base as a free database management system.

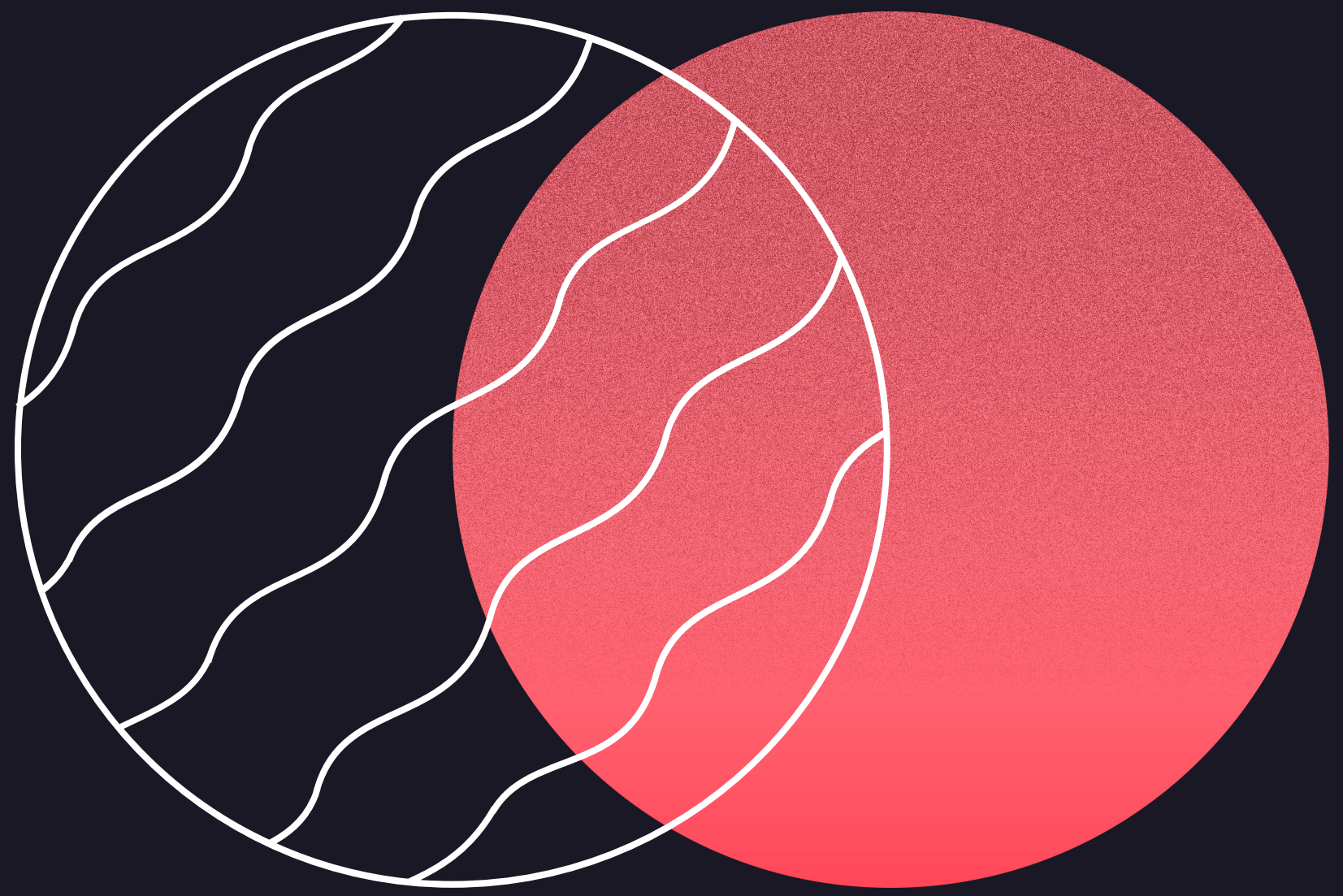
MariaDB is a fork of the MySQL database management system. It is a multi-model database system, mainly used as Relational DBMS, but it can also be used as document store, graph DBMS and spatial DBMS.



# DATA MODEL



RDBMS	Relational
<p><b>ColumnStore</b></p>	<p>columnar storage engine that utilizes a massively parallel distributed data architecture.</p> <p><b>InfiniDB</b></p>
<p><b>Graph DMBS</b></p>	<p>The Open Query GRAPH computation engine, or OQGRAPH as the engine itself is called, allows you to handle hierarchies (tree structures) and complex graphs (nodes having many connections in several directions).</p>



**QUERY LANGUAGE**

# DDL - CREATE, DROP

```
CREATE DATABASE Demo;  
USE Demo;
```

**CREATE TABLE table\_name (columnName columnType);**

```
CREATE TABLE Book(  
id INT NOT NULL AUTO_INCREMENT,  
name VARCHAR(100) NOT NULL,  
PRIMARY KEY (id));
```

```
DROP TABLE Book;  
DROP DATABASE Demo;
```

# DDL - ALTER, TRUNCATE

**ALTER TABLE table\_name ADD/MODIFY column\_name definition;**

```
ALTER TABLE Book ADD (author VARCHAR(30));
```

```
ALTER TABLE Book MODIFY (name VARCHAR(30));
```

**TRUNCATE TABLE table\_name;**

```
TRUNCATE TABLE Book;
```

07

# DML - INSERT, UPDATE, DELETE

**INSERT INTO table\_name (column\_1, column\_2, ... )  
VALUES (value1, value2, ... ), (value1, value2, ... ), ...;**

INSERT INTO Book (id, name) VALUES (1, 'MariaDB Book');

**UPDATE table\_name SET [column\_1= value1,...column\_N = valueN]  
[WHERE CONDITION]**

UPDATE Book SET name='MariaDB' WHERE id=1;

**DELETE FROM table\_name [WHERE condition];**

DELETE FROM Book WHERE name='MariaDB';

# SELECTION AND PROJECTION

```
SELECT [DISTINCT | ALL] { * | select_list }  
FROM { table_name [alias] | view_name } [ { table_name [alias] |  
view_name } ] ...  
[WHERE condition]  
[GROUP BY condition_list]  
[HAVING condition]  
[ORDER BY { column_name | column_# [ASC | DESC] } ...
```



# AGREGATE FUNCTIONS

```
SELECT COUNT(*) FROM products;
```

```
SELECT AVG(unitsinstock) FROM products;
```

```
SELECT MIN(unitsinstock) FROM products;
```

```
SELECT MAX(unitsinstock) FROM products;
```

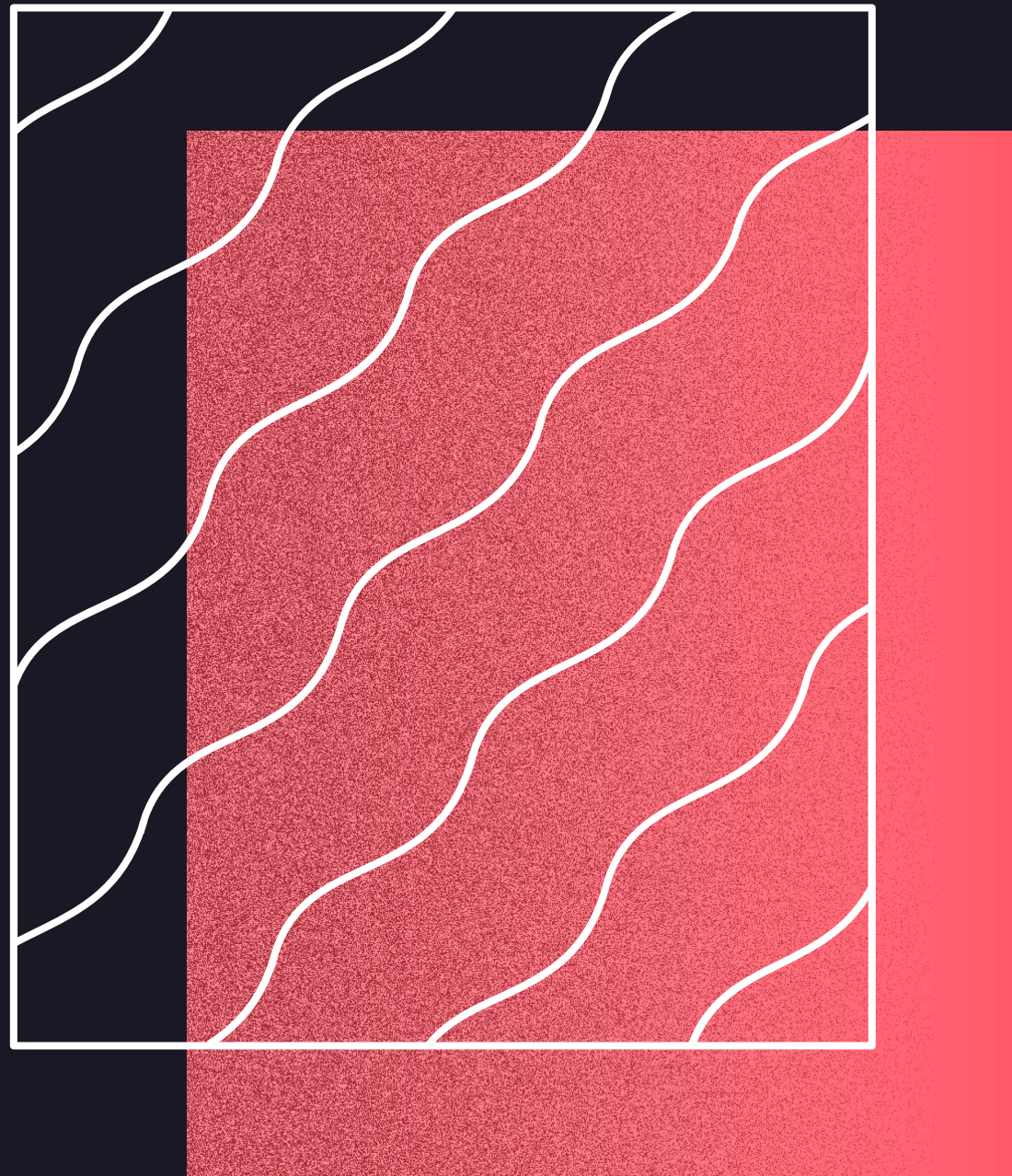
```
SELECT categoryid, SUM(unitsinstock) FROM products GROUP BY categoryid;
```

# JOINS AND NESTING

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

CustomerID	CustomerName	ContactName
1	Alfreds Futterkiste	Maria Anders
2	Ana Trujillo Emparedados y helados	Ana Trujillo
3	Antonio Moreno Taquería	Antonio Moreno

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID
WHERE Orders.CustomerID > (SELECT AVG(CustomerID) FROM Orders);
```



## MARIADB FLAWS

MariaDB doesn't support Data Masking and Dynamic column while MySQL supports it

MariaDB lacks some of the features provided by the MySQL enterprise edition. To address this, it offers alternative open-source plugins.

# Pros and Cons

## Pros:

ColumnStore

Better performance in Flash storage

MariaDB supports PHP;

Supports standard query languages  
and high performance storage  
systems

Security

## Cons:

Long-term cost

Support

