# Lesson 27

21.10.2024

```
public class ex1 {
    int num = 100;
    public void calc(int num)   { num = num * 10;   }
    public void printNum()      { System.out.println(num); }

    public static void main(String[] args)
    {
        ex1 obj = new ex1();
        obj.calc(2);
        obj.printNum();
    }
}
```

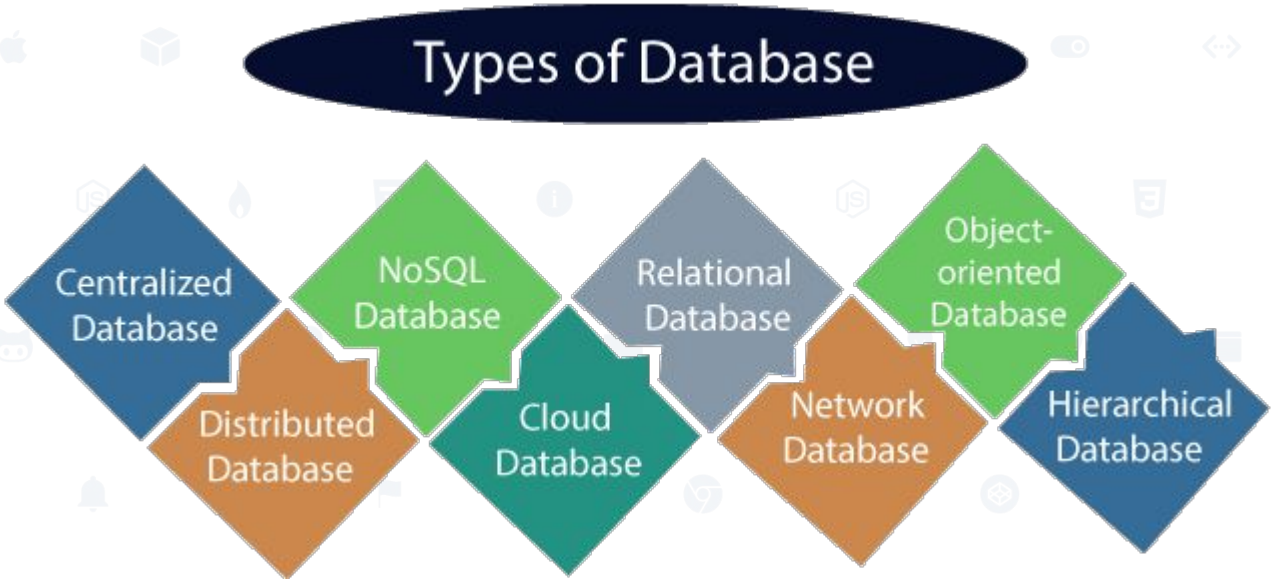
```
class First {
    public First() { System.out.println("a"); }
}

class Second extends First {
    public Second() { System.out.println("b"); }
}

class Third extends Second {
    public Third() { System.out.println("c"); }
}

public class ex2 {
    public static void main(String[] args) {
        Third c = new Third();
    }
}
```

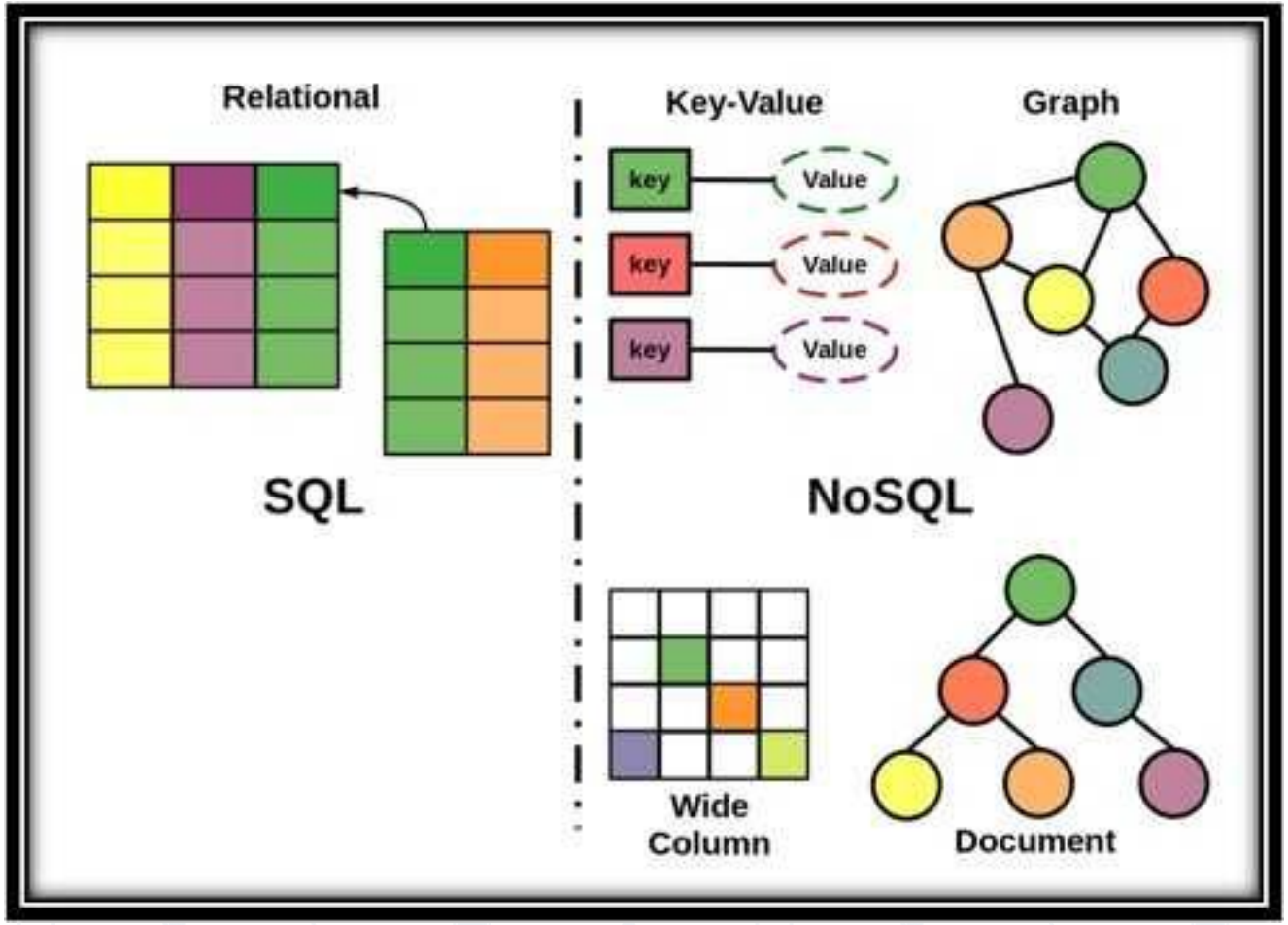
```
public class ex3 {
    public static void main(String[] args) {
        var list = List.of(new String[]{"A", "BB", "CCC"},
                new String[]{"DD", "E"});
        list.forEach(x ->
                System.out.print(x.length));
    }
}
```

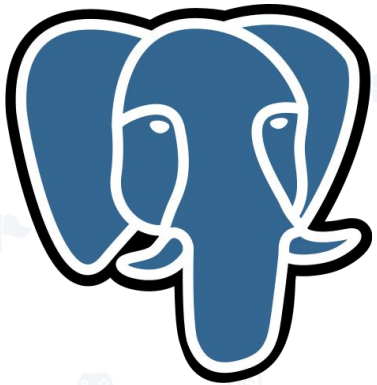
```
public class ex4 {
    public static void main(String[] args) {
        char[][] arr = {
                {'A', 'B', 'C'},
                {'D', 'E', 'F'},
                {'G', 'H', 'I'}
        };
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr[i].length; j++) {
                System.out.print(arr[i][1]);
            }
            System.out.println();
        }
    }
}
```

```
class Car {
    void speed(Byte val) {System.out.println("DARK");}

    void speed(byte... vals) {System.out.println("LIGHT");}
}

public class ex5 {

    public static void main(String[] args) {
        byte b = 10;
        new Car().speed(b);
    }
}
```

**База даних** - сукупність даних, що зберігаються відповідно до схеми даних, маніпулювання якими виконують відповідно до правил засобів моделювання даних.




Relational
Key-Value
Graph
key
Value
key
Value
key
Value
SQL
NoSQL
Wide
Column
Document




MariaDB


Microsoft®
SQL Server®


SQLite


H2

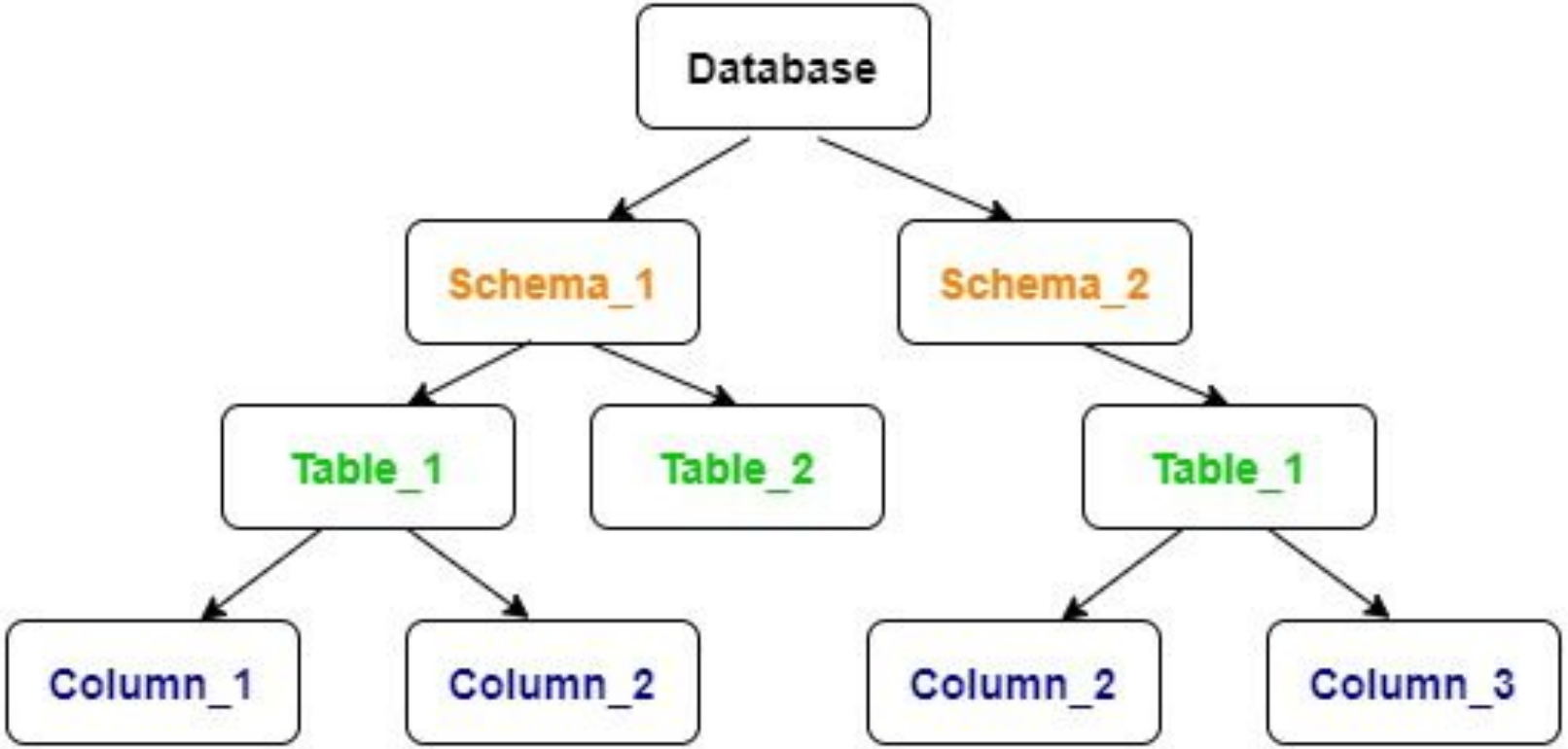

MySQL™


ORACLE®
DATABASE

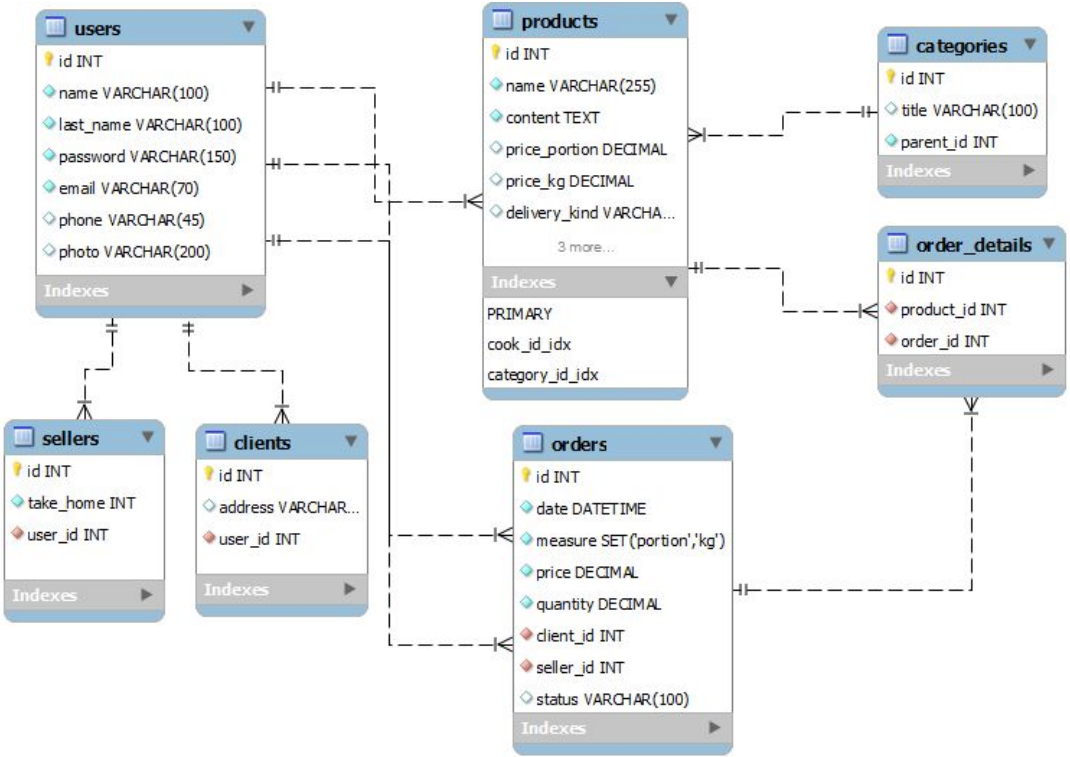
https://db-engines.com/en/ranking

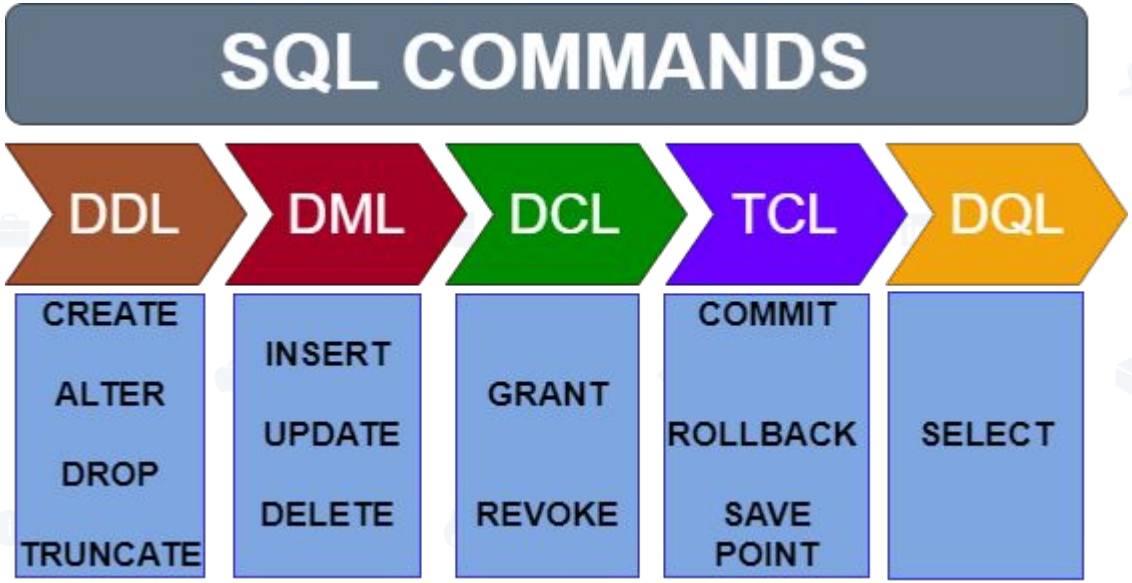
| Rank Feb 2024 | Rank Jan 2024 | Rank Feb 2023 | DBMS | Database Model | Score Feb 2024 | Score Jan 2024 | Score Feb 2023 |
|---|---|---|---|---|---|---|---|
| 1. | 1. | 1. | Oracle | Relational, Multi-model | 1241.45 | -6.05 | -6.08 |
| 2. | 2. | 2. | MySQL | Relational, Multi-model | 1106.67 | -16.79 | -88.78 |
| 3. | 3. | 3. | Microsoft SQL Server | Relational, Multi-model | 853.57 | -23.03 | -75.52 |
| 4. | 4. | 4. | PostgreSQL | Relational, Multi-model | 629.41 | -19.55 | +12.90 |
| 5. | 5. | 5. | MongoDB | Document, Multi-model | 420.36 | +2.88 | -32.41 |
| 6. | 6. | 6. | Redis | Key-value, Multi-model | 160.71 | +1.33 | -13.12 |
| 7. | 7. | ↑ 8. | Elasticsearch | Search engine, Multi-model | 135.74 | -0.33 | -2.86 |
| 8. | 8. | ↓ 7. | IBM Db2 | Relational, Multi-model | 132.23 | -0.18 | -10.74 |
| 9. | 9. | ↑ 12. | Snowflake | Relational | 127.45 | +1.53 | +11.80 |
| 10. | ↑ 11. | ↓ 9. | SQLite | Relational | 117.28 | +2.08 | -15.38 |
| 11. | ↓ 10. | ↓ 10. | Microsoft Access | Relational | 113.17 | -4.50 | -17.86 |
| 12. | 12. | ↓ 11. | Cassandra | Wide column, Multi-model | 109.27 | -1.77 | -6.95 |
| 13. | 13. | 13. | MariaDB | Relational, Multi-model | 97.23 | -2.00 | +0.42 |
| 14. | 14. | 14. | Splunk | Search engine | 91.65 | -1.07 | +4.57 |
| 15. | ↑ 16. | 15. | Amazon DynamoDB | Multi-model | 82.90 | +1.96 | +3.21 |
| 16. | ↓ 15. | 16. | Microsoft Azure SQL Database | Relational, Multi-model | 79.56 | -1.51 | +0.81 |
| 17. | 17. | ↑ 19. | Databricks | Multi-model | 76.91 | -3.62 | +16.58 |
| 18. | 18. | ↓ 17. | Hive | Relational | 65.81 | -1.15 | -6.31 |
| 19. | 19. | ↑ 22. | Google BigQuery | Relational | 63.63 | +0.15 | +11.17 |
| 20. | 20. | ↓ 18. | Teradata | Relational, Multi-model | 51.24 | -1.94 | -11.79 |

## Sql database structure



# Схема бази даних






1. DDL – Data Definition Language
2. DQL – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language
5. TCL – Transaction Control Language

- **CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- **DROP**: This command is used to delete objects from the database.
- **ALTER:** This is used to alter the structure of the database.
- **TRUNCATE:** This is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT**: This is used to add comments to the data dictionary.
- **RENAME:** This is used to rename an object existing in the database.

- **SELECT:** It is used to retrieve data from the database.

- **INSERT**: It is used to insert data into a table.
- **UPDATE:** It is used to update existing data within a table.
- **DELETE**: It is used to delete records from a database table.
- **LOCK:** Table control concurrency.
- **CALL:** Call a PL/SQL or JAVA subprogram.
- **EXPLAIN PLAN:** It describes the access path to data.

```
CREATE TABLE [schema.]table
        (column datatype [DEFAULT expr][, ...]);
```

| Data Type | Description |
|---|---|
| VARCHAR2(*size*) | Variable-length character data |
| CHAR(*size*) | Fixed-length character data |
| NUMBER(*p*,*s*) | Variable-length numeric data |
| DATE | Date and time values |
| LONG | Variable-length character data (up to 2 GB) |
| CLOB | Character data (up to 4 GB) |
| RAW and LONG RAW | Raw binary data |
| BLOB | Binary data (up to 4 GB) |
| BFILE | Binary data stored in an external file (up to 4 GB) |
| ROWID | A base-64 number system representing the unique address of a row in its table |

```sql
ALTER TABLE table_name
ADD column_name datatype;
```

```sql
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

```sql
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

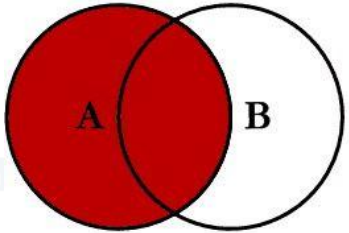
```sql
SELECT column1, column2, ...
FROM table_name;
```

```sql
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

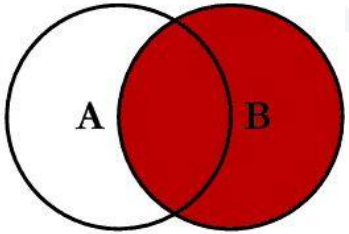
```sql
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

```sql
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

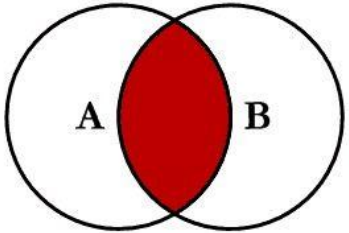
# SQL JOINS



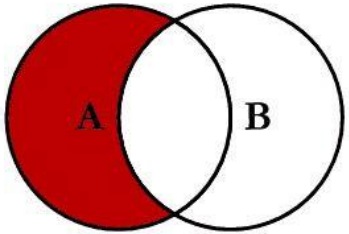

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



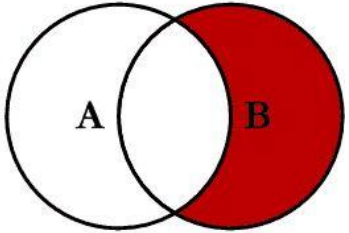

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



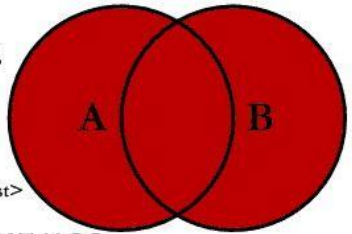

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



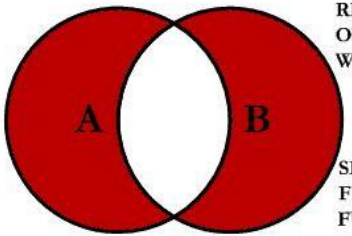

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



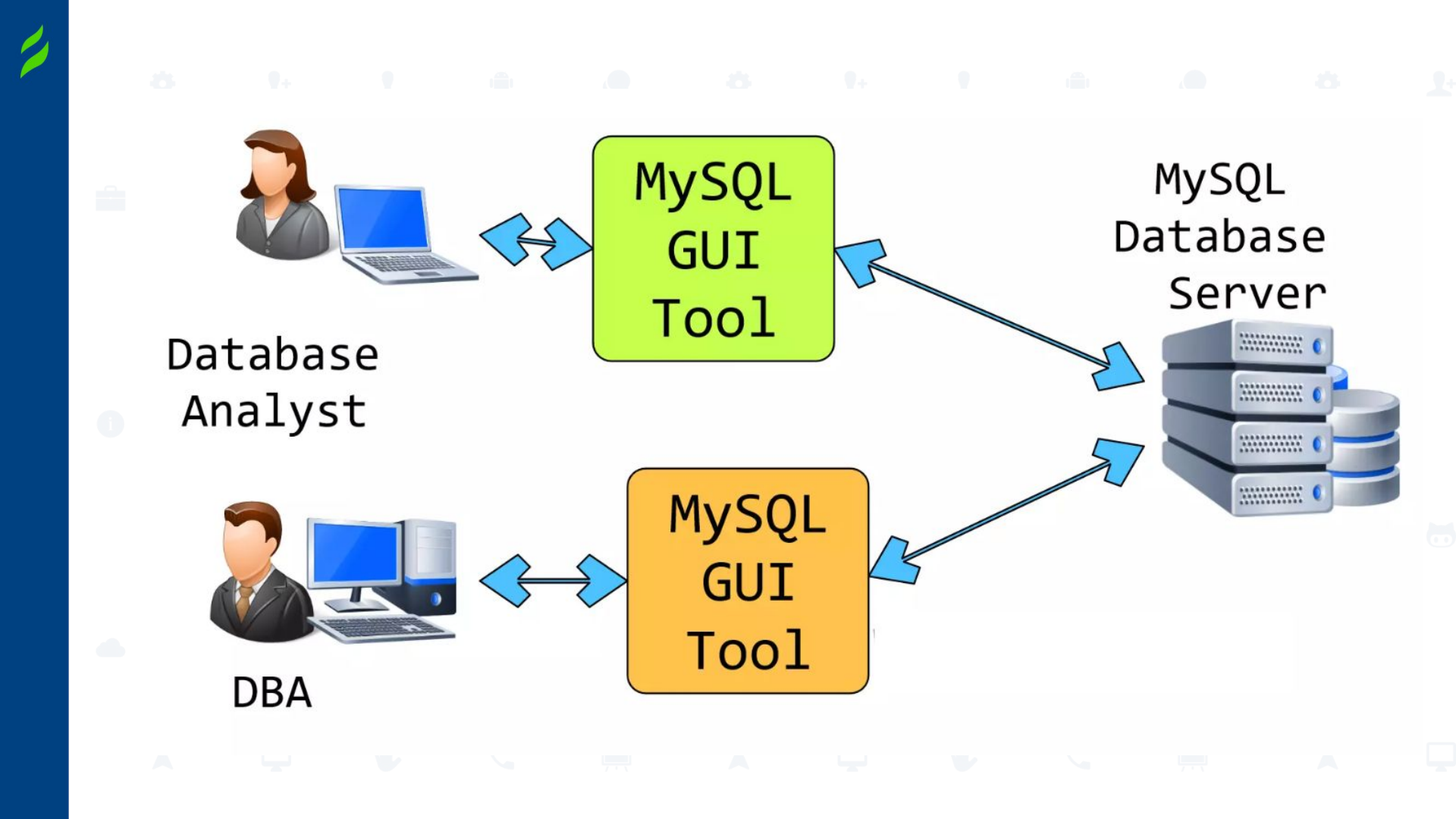

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```


MySQL
GUI
Tool
MySQL
Database
Server
Database
Analyst
MySQL
GUI
Tool
DBA