Views in database systems, including MySQL, are an important feature that offer several benefits:

1. Simplification of complex queries:
   - Views allow you to save complex SQL queries as a virtual table.
   - This simplifies future data retrieval, as you can query the view instead of rewriting complex joins or subqueries.

2. Data security:
   - Views can restrict access to specific columns or rows, allowing you to control what data users can see.
   - This is crucial for maintaining data privacy and adhering to security policies.

3. Data abstraction:
   - Views provide a layer of abstraction over the underlying table structure.
   - If you change the structure of base tables, you can often update the view definition without changing application code.

4. Consistent data representation:
   - Views ensure that data is presented consistently across different applications or users.
   - This is particularly useful for standardizing calculations or data transformations.

5. Reduced storage requirements:
   - Unlike materialized views, regular views don't store data physically.
   - They compute data on-the-fly, saving storage space.

6. Aggregation and summarization:
   - Views can pre-compute aggregations, making it easier to retrieve summary data quickly.

7. Backward compatibility:
   - When making schema changes, views can help maintain compatibility with older versions of applications.

8. Modular query design:
   - Complex queries can be broken down into simpler views, promoting code reusability and easier maintenance.

9. Performance optimization:
   - In some cases, views can be optimized by the database engine, potentially improving query performance.

Here's a simple example of creating a view in MySQL:

```
CREATE VIEW high_value_customers AS
SELECT customer_id, name, total_purchases
FROM customers
WHERE total_purchases > 10000;
```

This view could then be queried like a regular table:

```
SELECT * FROM high_value_customers;
```