**Module 7 – Java – RDBMS & Database Programming with JDBC**

**1: INTRODUCTION TO JDBC**

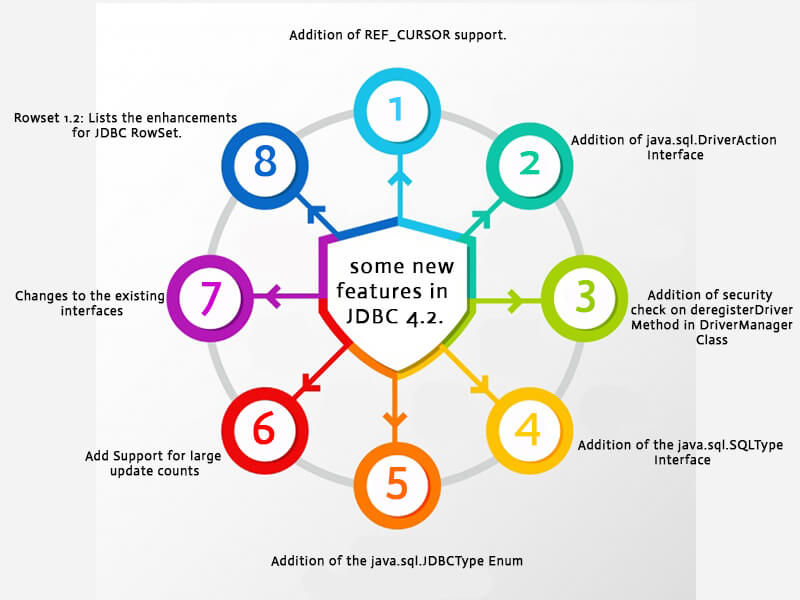
1. **What is JDBC (Java Database Connectivity)?**

JDBC (Java Database Connectivity) is a Java-based API that enables Java applications to interact with databases. It provides a standard interface for connecting to relational databases, executing SQL queries, and retrieving results.

1. **Importance of JDBC in Java Programming**

JDBC (Java Database Connectivity) plays a crucial role in Java programming, particularly when it comes to data management and interaction with databases. Here are some key reasons highlighting its importance:

1. Database Interaction
2. Portability
3. Support for Multiple Databases
4. Ease of Use
5. Transaction Management
6. Error Handling
7. Integration with Other Java Technologies
8. Performance Optimization
9. Flexibility
10. Large Community and Resources



1. **JDBC Architecture: Driver Manager, Driver, Connection, Statement, and ResultSet**

**1. Driver Manager**

* The DriverManager class is responsible for managing a list of database drivers. It matches connection requests from the application with the appropriate driver using the connection URL.

**2. Driver**

* A JDBC driver is a software component that enables Java applications to interact with a database. Different drivers are available for different database management systems.

**3. Connection**

* The Connection interface represents a session with a specific database. It is used to create Statement objects and manage transactions.

**4. Statement**

* The Statement interface is used to execute SQL queries against the database.

**5. ResultSet**

* The ResultSet interface represents the result set of a query. It provides methods to retrieve data from the results returned by executing a statement.

