Qt SQL and Application Questions

Your Name

April 20, 2025

1 Core Class Questions

1.1 QSqlDatabase (25 questions)

1. What is the purpose of QSqlDatabase in Qt's SQL module?
2. How do you add a database connection using QSqlDatabase?
3. What is the difference between addDatabase() and cloneDatabase()?
4. How would you check if a database connection is open?
5. What method is used to close a database connection?
6. How do you specify the database driver type when creating a connection?
7. What are the supported database drivers in Qt SQL?
8. How would you handle transaction management with QSqlDatabase?
9. What is the purpose of the connectionName parameter in addDatabase()?
10. How do you retrieve a list of available database connections?
11. What happens if you don't specify a connection name?
12. How would you remove a database connection?
13. What methods are available for error handling in QSqlDatabase?
14. How do you set connection parameters like hostname or username?
15. What is the purpose of the QSqlDatabase::database() static method?
16. How would you check if a particular driver is available?
17. What is the role of QSqlDatabase::tables()?
18. How do you execute a raw SQL statement directly through QSqlDatabase?
19. What is the difference between commit() and rollback()?

20. How would you handle multiple database connections in an application?
21. What security considerations exist when using QSqlDatabase?
22. How do you optimize database connection pooling with QSqlDatabase?
23. What is the impact of not properly closing database connections?
24. How would you implement a connection timeout mechanism?
25. What are the limitations of QSqlDatabase compared to native database APIs?

1.2 QSqlQuery (25 questions)

1. What is the primary function of QSqlQuery?
2. How do you execute a SELECT statement with QSqlQuery?
3. What is the difference between exec() and prepare()/exec()?
4. How would you retrieve the number of rows affected by a query?
5. What methods are available for navigating query results?
6. How do you bind values to a prepared statement?
7. What is positional binding versus named binding?
8. How would you handle SQL injection prevention with QSqlQuery?
9. What is the purpose of QSqlQuery::size()?
10. How do you check if a query executed successfully?
11. What is batch execution and how is it implemented?
12. How would you retrieve metadata about the result set?
13. What is the difference between forward-only and scrollable queries?
14. How do you handle BLOB data with QSqlQuery?
15. What is the role of QSqlQuery::lastInsertId()?
16. How would you implement pagination with QSqlQuery?
17. What are the performance considerations when using QSqlQuery?
18. How do you reuse a QSqlQuery object for multiple statements?
19. What is the lifetime of a QSqlQuery object relative to its database connection?

20. How would you handle stored procedures with QSqlQuery?
21. What error information is available when a query fails?
22. How do you optimize bulk inserts with QSqlQuery?
23. What is the behavior of QSqlQuery with different database backends?
24. How would you implement asynchronous query execution?
25. What are the limitations of QSqlQuery compared to ORM solutions?

1.3 QSqlRecord (25 questions)

1. What does QSqlRecord represent in Qt SQL?
2. How do you retrieve a QSqlRecord from a query result?
3. What information does a QSqlRecord contain?
4. How would you get the number of fields in a record?
5. What methods are available for accessing field values?
6. How do you check if a field is null?
7. What is the purpose of QSqlRecord::indexOf()?
8. How would you retrieve metadata about a field?
9. What is the difference between field() and fieldName()?
10. How do you modify values in a QSqlRecord?
11. What is the role of QSqlRecord in model-view programming?
12. How would you create a new empty QSqlRecord?
13. How do you append or remove fields from a QSqlRecord?
14. What is the relationship between QSqlRecord and QSqlField?
15. How would you implement record validation with QSqlRecord?
16. What is the performance impact of using QSqlRecord?
17. How do you handle different data types in QSqlRecord fields?
18. What is the behavior with binary or large data fields?
19. How would you serialize a QSqlRecord to JSON?
20. What are the limitations of QSqlRecord for complex data structures?
21. How do you compare two QSqlRecord objects?

22. What is the role of QSqlRecord in database schema operations?
23. How would you handle default values in QSqlRecord?
24. What is the impact of database schema changes on QSqlRecord?
25. How do you implement custom field types with QSqlRecord?

1.4 QTableView (25 questions)

1. What is the role of QTableView in Qt's model-view architecture?
2. How do you connect a SQL query result to a QTableView?
3. What is the relationship between QTableView and QSqlTableModel?
4. How would you customize the appearance of a QTableView?
5. What methods are available for sorting columns?
6. How do you handle selection changes in QTableView?
7. What is the purpose of QTableView::setModel()?
8. How would you implement custom delegates in QTableView?
9. What are the performance considerations for large datasets?
10. How do you enable editing in a QTableView connected to SQL data?
11. What is the role of QItemSelectionModel with QTableView?
12. How would you implement filtering in a QTableView?
13. What methods are available for row and column manipulation?
14. How do you handle horizontal and vertical headers?
15. What is the difference between QTableView and QTableWidget?
16. How would you implement context menus in QTableView?
17. What is the behavior of QTableView with different data types?
18. How do you optimize rendering performance for SQL data?
19. What are the keyboard navigation options in QTableView?
20. How would you implement drag and drop with QTableView?
21. What is the role of QAbstractItemView methods in QTableView?
22. How do you handle very wide tables in QTableView?
23. What are the accessibility features of QTableView?
24. How would you export QTableView data to other formats?
25. What are the limitations of QTableView for complex data visualization?

2 Intelligent Application Questions (20 questions)

1. Design a database abstraction layer using Qt SQL classes that can switch between SQLite and PostgreSQL without code changes
2. Implement a data caching system that maintains a local SQLite cache of frequently accessed remote MySQL data
3. Create a thread-safe wrapper for QSqlDatabase that manages connection pooling
4. Design a schema migration system using Qt SQL classes
5. Implement a REST API server that uses QSqlQuery for database access
6. Create a data visualization dashboard that combines multiple QTableView widgets
7. Design an undo/redo system for database edits made through a QTableView
8. Implement a permission system that filters QSqlQuery results based on user roles
9. Create a generic data importer that maps CSV files to SQL tables using QSqlRecord
10. Design a reactive UI that automatically updates QTableView when underlying SQL data changes
11. Implement a full-text search across multiple SQL tables with results in a QTableView
12. Create a database schema inspector tool using QSqlDatabase metadata methods
13. Design a bulk data processing system with progress reporting using QSqlQuery
14. Implement a data validation framework that checks QSqlRecord before insertion
15. Create a reporting system that generates PDFs from QTableView contents
16. Design a synchronization mechanism between a local SQLite and remote database
17. Implement a query builder interface that generates QSqlQuery objects
18. Create a data auditing system that logs all changes made through QTableView
19. Design a high-performance scrolling QTableView for large SQL result sets
20. Implement a mobile application that uses Qt SQL with offline capability