  - Use of CHANGE *col_name*, DROP *col_name*, or DROP INDEX, IGNORE or RENAME in ALTER TABLE statements. Use of multiple ADD, ALTER, DROP, or CHANGE clauses in an ALTER TABLE statement. See Section 13.1.9, “ALTER TABLE Statement”.
  - Use of index names, indexes on a prefix of a column, and use of INDEX or KEY in CREATE TABLE statements. See Section 13.1.20, “CREATE TABLE Statement”.
  - Use of TEMPORARY or IF NOT EXISTS with CREATE TABLE.
  - Use of IF EXISTS with DROP TABLE and DROP DATABASE.
  - The capability of dropping multiple tables with a single DROP TABLE statement.
  - The ORDER BY and LIMIT clauses of the UPDATE and DELETE statements.
  - INSERT INTO *tbl_name* SET *col_name* = ... syntax.
  - The DELAYED clause of the INSERT and REPLACE statements.
  - The LOW_PRIORITY clause of the INSERT, REPLACE, DELETE, and UPDATE statements.
  - Use of INTO OUTFILE or INTO DUMPFILE in SELECT statements. See Section 13.2.10, “SELECT Statement”.
  - Options such as STRAIGHT_JOIN or SQL_SMALL_RESULT in SELECT statements.
  - You don't need to name all selected columns in the GROUP BY clause. This gives better performance for some very specific, but quite normal queries. See Section 12.20, “Aggregate Functions”.
  - You can specify ASC and DESC with GROUP BY, not just with ORDER BY.
  - The ability to set variables in a statement with the := assignment operator. See Section 9.4, “User-Defined Variables”.
- Data types
  - The MEDIUMINT, SET, and ENUM data types, and the various BLOB and TEXT data types.
  - The AUTO_INCREMENT, BINARY, NULL, UNSIGNED, and ZEROFILL data type attributes.
- Functions and operators
  - To make it easier for users who migrate from other SQL environments, MySQL Server supports aliases for many functions. For example, all string functions support both standard SQL syntax and ODBC syntax.
  - MySQL Server understands the || and && operators to mean logical OR and AND, as in the C programming language. In MySQL Server, || and OR are synonyms, as are && and AND. Because of this nice syntax, MySQL Server doesn't support the standard SQL || operator for string concatenation; use CONCAT() instead. Because CONCAT() takes any number of arguments, it is easy to convert use of the || operator to MySQL Server.
  - Use of COUNT(DISTINCT *value_list*) where *value_list* has more than one element.
  - String comparisons are case-insensitive by default, with sort ordering determined by the collation of the current character set, which is utf8mb4 by default. To perform case-sensitive comparisons instead, you should declare your columns with the BINARY attribute or use the BINARY cast, which