PS 2: Code-reading and design questions

The Table and Column classes

1) opens the table so that it can be accessed by SQL commands, opens the Database handle for the underlying BDB database, and fills in the Table object with the metadata stored in the catalog for this table;
   return OperationStatus.NOTFOUND if the table does not exist;
   we can use CREAT TABLE to open a new table
2) primaryKeyColumn
3) getColumn; getColumn(0)
4) getType; the type (one of the constants defined in this class)
5) getLength; the maximum number of bytes

The SQLStatement class and its subclasses

6) getTable
7) getColumn

Marshalling

8) -2
9) -1
10) by inserting (1, 'hello'), 1 is the primary key and the value is ‘hello’ with a header of offsets;
    by inserting (2, NULL), 2 is the primary key and the value is only a header of offsets -2 and -1 because NULL means the other fields are empty
11) by inserting (1, '1234', 'hello', 12.5), ‘1234’ is the primary key and the value is 1, 'hello', 12.5 with a header of offsets;
    by inserting (2, '4567', 'wonderful', NULL), ‘4567’ is the primary key and the value is 2, 'wonderful' with a header of offsets
12) writeByte when the value is a string, writeShort when the value is a offset, writeInt when the value is a int

Unmarshalling

13) we first determine how many bytes is that offset from the start, then perform at an offset of that specific bytes
14) readByteAtOffset
15) we read the next offset and then compute the length

Miscellaneous

16) it creates an InsertRow object for the row to be inserted and use that object to marshall the row
17) we need to perform the actual insertion, and to print the appropriate message after it has occurred; Linear Hashing
18) next() and getColumnVal(int colIndex)