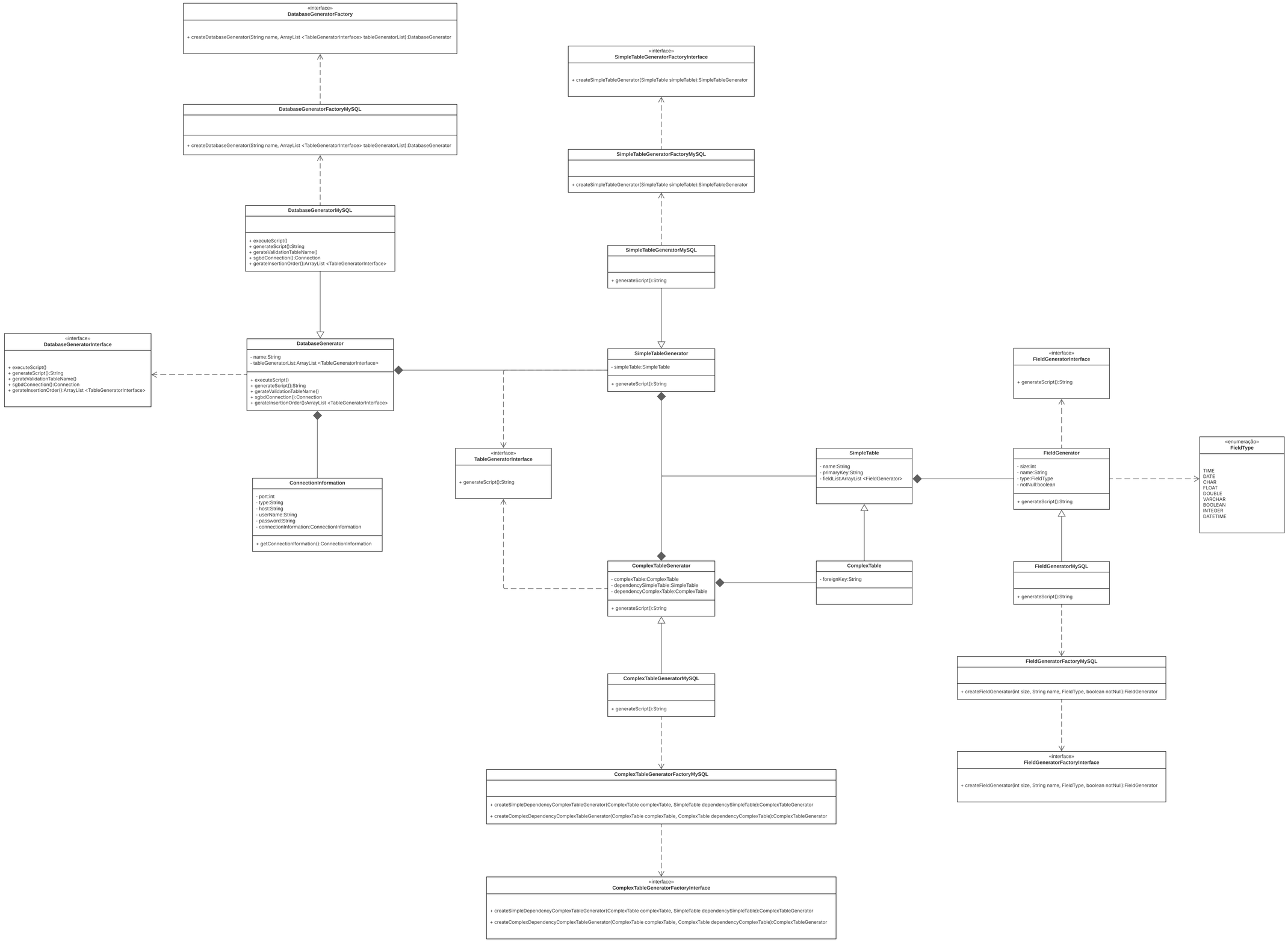
«interface»
DatabaseGeneratorFactory
+ createDatabaseGenerator(String name, ArrayList <TableGeneratorInterface> tableGeneratorList):DatabaseGenerator
DatabaseGeneratorFactoryMySQL
+ createDatabaseGenerator(String name, ArrayList <TableGeneratorInterface> tableGeneratorList):DatabaseGenerator
DatabaseGeneratorMySQL
+ executeScript()
+ generateScript():String
+ gerateValidationTableName()
+ sgbdConnection():Connection
+ gerateInsertionOrder():ArrayList <TableGeneratorInterface>
«interface»
DatabaseGeneratorInterface
+ executeScript()
+ generateScript():String
+ gerateValidationTableName()
+ sgbdConnection():Connection
+ gerateInsertionOrder():ArrayList <TableGeneratorInterface>
DatabaseGenerator
- name:String
- tableGeneratorList:ArrayList <TableGeneratorInterface>
+ executeScript()
+ generateScript():String
+ gerateValidationTableName()
+ sgbdConnection():Connection
+ gerateInsertionOrder():ArrayList <TableGeneratorInterface>
ConnectionInformation
- port:int
- type:String
- host:String
- userName:String
- password:String
- connectionInformation:ConnectionInformation
+ getConnectionInformation():ConnectionInformation
«interface»
TableGeneratorInterface
+ generateScript():String
«interface»
SimpleTableGeneratorFactoryInterface
+ createSimpleTableGenerator(SimpleTable simpleTable):SimpleTableGenerator
SimpleTableGeneratorFactoryMySQL
+ createSimpleTableGenerator(SimpleTable simpleTable):SimpleTableGenerator
SimpleTableGeneratorMySQL
+ generateScript():String
SimpleTableGenerator
- simpleTable:SimpleTable
+ generateScript():String
SimpleTable
- name:String
- primaryKey:String
- fieldList:ArrayList <FieldGenerator>
ComplexTableGenerator
- complexTable:ComplexTable
- dependencySimpleTable:SimpleTable
- dependencyComplexTable:ComplexTable
+ generateScript():String
ComplexTable
- foreignKey:String
ComplexTableGeneratorMySQL
+ generateScript():String
ComplexTableGeneratorFactoryMySQL
+ createSimpleDependencyComplexTableGenerator(ComplexTable complexTable, SimpleTable dependencySimpleTable):ComplexTableGenerator
+ createComplexDependencyComplexTableGenerator(ComplexTable complexTable, ComplexTable dependencyComplexTable):ComplexTableGenerator
«interface»
ComplexTableGeneratorFactoryInterface
+ createSimpleDependencyComplexTableGenerator(ComplexTable complexTable, SimpleTable dependencySimpleTable):ComplexTableGenerator
+ createComplexDependencyComplexTableGenerator(ComplexTable complexTable, ComplexTable dependencyComplexTable):ComplexTableGenerator
«interface»
FieldGeneratorInterface
+ generateScript():String
FieldGenerator
- size:int
- name:String
- type:FieldType
- notNull:boolean
+ generateScript():String
«enumeração»
FieldType
TIME
DATE
CHAR
FLOAT
DOUBLE
VARCHAR
BOOLEAN
INTEGER
DATETIME
FieldGeneratorMySQL
+ generateScript():String
FieldGeneratorFactoryMySQL
+ createFieldGenerator(int size, String name, FieldType, boolean notNull):FieldGenerator
«interface»
FieldGeneratorFactoryInterface
+ createFieldGenerator(int size, String name, FieldType, boolean notNull):FieldGenerator