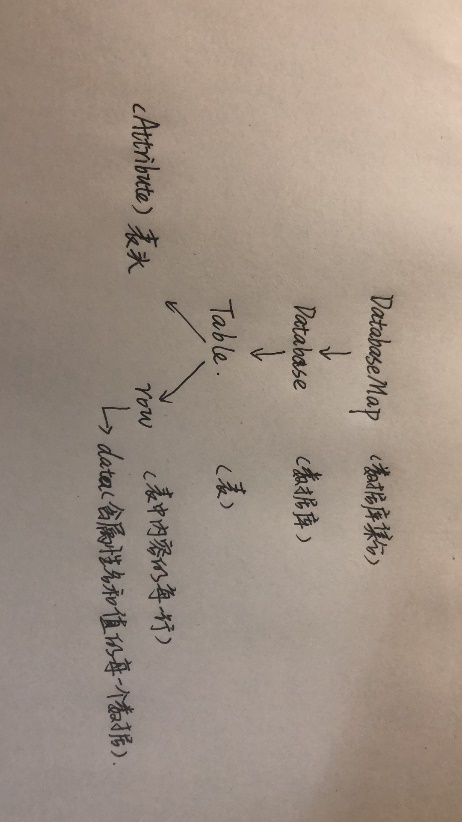
说明文档

**总体结构：**

DatabaseMap是多个数据库Database的组合，Database是多个表Table的组合， Table是多行Row的组合，不同类型属性的数据均以string类存储于表中，具体操作时再转化为相应类型。



**main.cpp:**

主函数在main.cpp，每条命令的读取以；结尾，如果文件末尾有\n,仅能处理只有一个\n的情况，如需解析其它情况可在函数中做修改。

**Command .h和Command.cpp:**

Command .h和Command.cpp 为对读入命令行的解析处理。其中，class Command 中有私有变量command存输入命令行，公有成员函数进行构造，operator函数是对命令行进行处理，operator函数中首先读取命令的第一个单词，分类并分配到相应的函数中进行处理。

所有涉及到的数据输出，均使用OutputData(string value,string type)函数，将value转化为相应类型。如需增加数据类型，可在此函数中添加新的输出方式；如有输出长度、精度等要求，也可在此函数中实现。

对于WHERE子句的处理，调用where\_clause函数，传入string型参数表示子句，拆分为多个条件句，通过DatabaseMap类的KeyWhereClause函数得到分别符合每个条件句的多个记录主键的集合，通过交集、并集来实现and和or的处理，最终得到符合子句条件的记录主键的集合。

**DatabaseMap.h和DatabaseMap.cpp:**

DatabaseMap.h和DatabaseMap.cpp为建立数据库集，是对多个数据库的集合这一整体进行操作。

其中，class DatabaseMap私有成员有dbs数据库集,current\_name记当前处理的数据库的名字，current\_db是指向当前处理的数据库（当前的Database）的指针；

公有成员函数中:

CreateDatabase是创建数据库，调用类Database中的创建表的函数CreateTable;

DropDatabase是删除数据库，调用类Database中的删除表的函数DropTable;

UseDatabase是切换数据库，切换到当前数据库；

ShowDatabases是输出数据库名；

InsertInto是在名为“table\_name”的表中加入名为“attr\_name”的属性及其对应的数据值“value”；

CreateTable是创建表，调用类Database中的创建表的函数CreateTable;

DropTable是删除表，调用类Database中的删除表的函数DropTable;

UseTable是切换表，切换到当前表；

ShowTables是输出表名；

ShowColumns是将名为“table\_name”的表中各项表头信息即属性名、属性类型、主键信息等输出；

GetAllKeys是返回所有主键的集合；

KeyWhereClause是查找名为“table\_name”的表的各项符合条件c的所有行Row，并且返回这些行的主键的集合；

DeleteRow是删除名为“table\_name”的表中主键为key的行；

Set是将名为“table\_name”的表中主键为key的行中的名为“attr\_name”的属性对应的数据值改为“value”；

GetValue是返回名为“table\_name”的表中主键为key的那一行的属性为“attr”的数据；

GetType 是返回名为“table\_name”的表中属性名为“attr”的属性类型；

OutPutAttr是调用指定表Table的OutPutAttr函数，输出Table的表头（即属性名）；

OutPutRow是根据主键key在名为“table\_name”的表中找到对应的Row，并调用其output函数，输出一行Row的数据；

getkeytype是调用Database类中的函数getkeytype；

**Database.h和Database.cpp:**

Database.h和Database.cpp为单个数据库，是对某一确定的数据库进行处理，中有类Database。

其中，私有成员为db\_name存数据库的名字，table\_list为数据库中表的集合——map<表名，表对象>；

公有成员函数中，

deleterow是删除主键值为key的那一行Row；

重载的两个output函数分别表示输出表头（属性名）和数据值；

**Table.h和Table.cpp：**

Table.h和Table.cpp为单个表，是对当前数据库的确定表的处理，中有类Table；

Table.h中的Data为包含主键key的值和类型的结构体，在此处实现对不同类型（如int、double、char）的key的分类处理，主要是通过重载“<””==”实现，这样做为了避免在排序比较key时由于将int或double类型的key误当作字符串按照字典序排列而产生错误；

类Table中的row\_map为表中行Row的集合（map），attr\_list为结构体属性（Attribute）的集合（vector）；

find函数为查找表中主键值为key的行Row；

**Row.h和Row.cpp：**

Row.h和Row.cpp为某行，是对确定表中具体某一行的处理，中有类Row；

其中结构体Clause为要处理的（已拆分的）条件；left和right分别为要进行比较的左右值，cmp\_op是比较运算符；

结构体Attribute是包括了属性名、属性类型、是否为NOT NULL、是否为主键、Default、Extra等于属性相关的一系列信息；

类Row中，into为插入函数，先构造为空的一行,然后为对应的空格赋值，未被赋值的仍为NULL。