Database Management System

# Insert record (*.trd file)

## Content

- Function
- Design
- Implementation

# Copyright Declaration

# Function



Insert a row of record in a specified table, and save to the record file (*trd) of the table.

**Input**

(1) Select a table in the tree view to get the table name.

(2) Select the menu “record->insert record” and it will pop up the dialog box, then enter the record data.

**Process**

(1) If the record file (*.trd) does not exist, then create the file.

Path: [DBMS_ROOT]\data\Ruanko\TABLE_NAME.trd

(2) Save the record data to the *.trd file.

**Output**

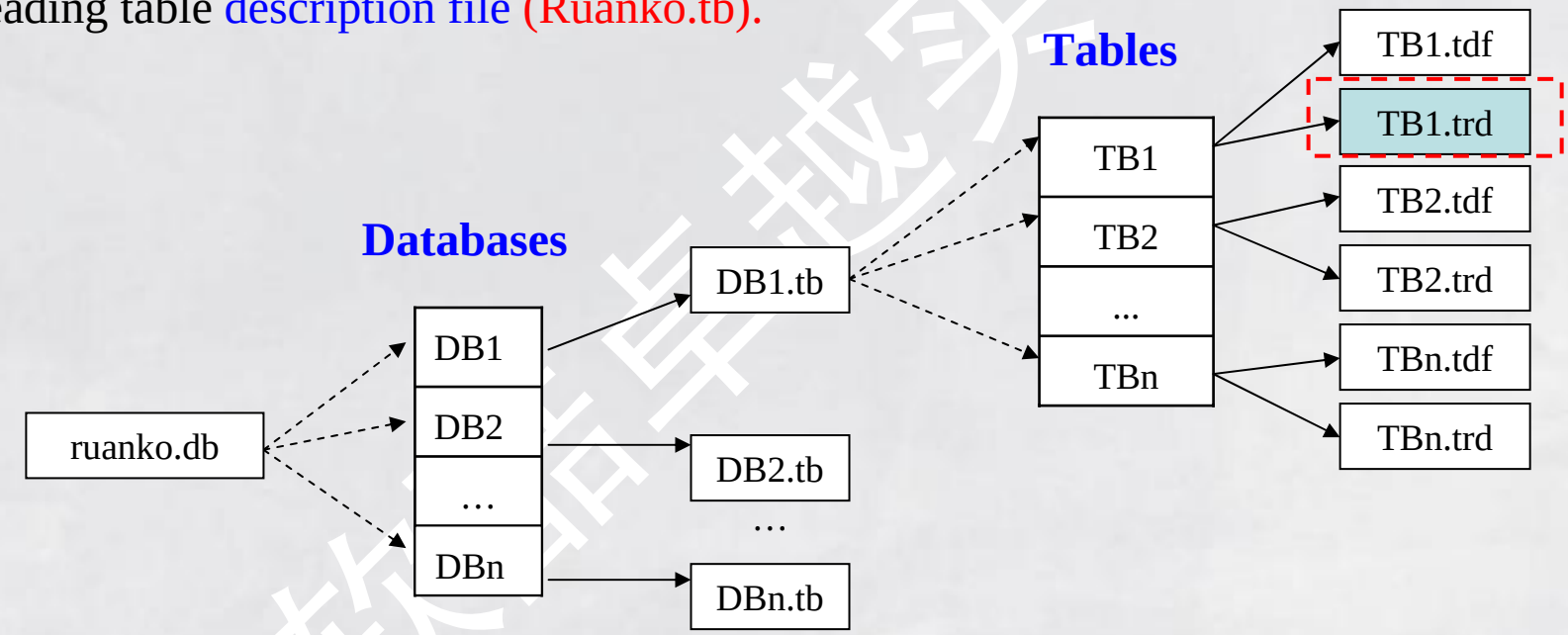
Insert a row of record in the record file (*.trd) of the specified table.

# Design

Iterative development based on the “Show table structure” function.

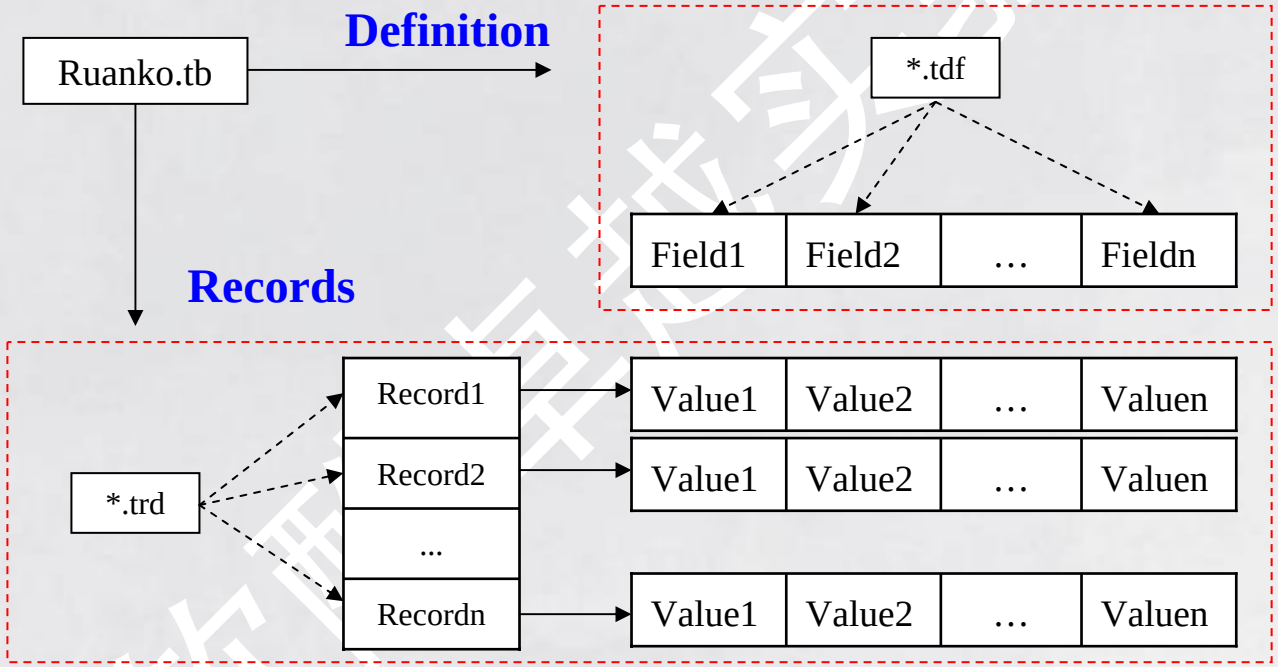
Create dialog box CRecordDlg, and receive user entered record data to assemble to CRecordEntity object. Call document class CRKDBMSDoc::InsertRecord() fucntion, and transfer the object to the logic layer to processing.

Create record file (*.trd) according to the table name to save record data in the table; save the file path to the table description file, and then we can find the table data file (*.trd) by reading table description file (Ruanko.tb).



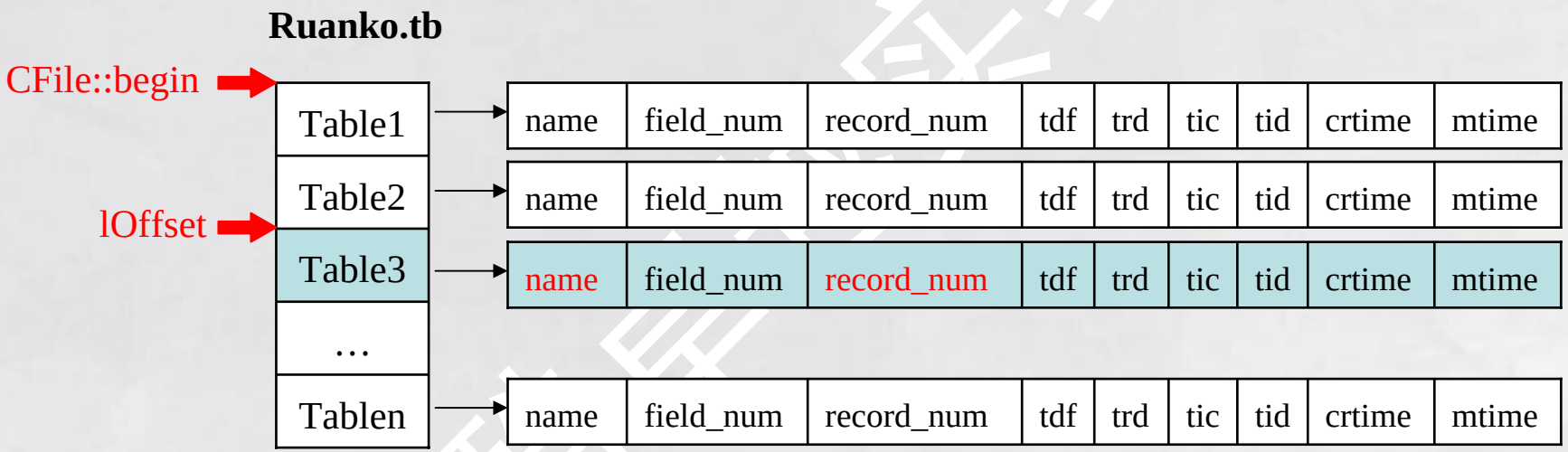
## 1. File structure

Multiple file paths of table are saved in the table description file “Ruanko.tb” : *.tdf file is to save table definition information, while *.trd file for saving record data. When inserting the record, save the data record to this file according to the recorded filed sequence in the table definition file (*.tdf).



# System design

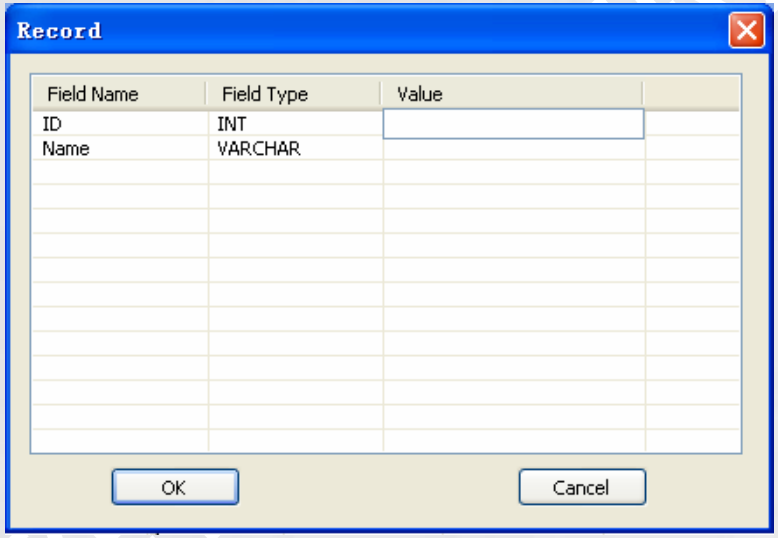
The "Ruanko.tb" file is also saving record number of each table (record_num). When inserting the record, find the corresponding table in the file based on the table name (name), locate the location of table in the file by CFile::Seek(lOffset, CFile::begin) function, and then modify the record_num in the table record_num.



## 2. Interface design

### (1) Dialog box CRecordDlg

Create **a dialog box** CRecordDlg, display field name, type and value of the table by list control. When click on the content of a certain item in the "value" column, we can enter the value of this field.

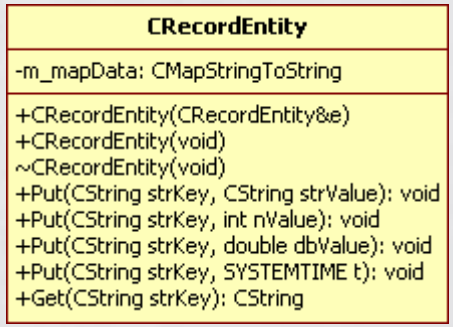



### (2) "Insert record" menu

Add response function CMainFrame::OnInsertRecord() to the **menu** "record ->insert record". Pop up the "record" dialog box after click; enter the record and then save to the record file (*trd).

### 3. CRecordEntity class design

Define entity class CRecordEntity to transfer record data among layers.




| Key | Value |
|---|---|
| Field1 | Value1 |
| Field2 | Value2 |
| … | … |
| Fieldn | Valuen |

**(1) data member**

CMap StringToString m_mapData;

Save record data by using key-value pair <key, value>

**2) member function**

1) void Put(CString strKey, CString strValue);

Set the field name as Key; the value of field is Value; save a record based on the field sequence.

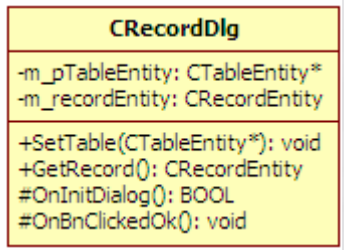
2) CString Get(CString strKey);

Set the field name as Key, and take out the value of each field in the record one by one.

**Note: as the type of returned value of Get() function is Cstring, we need to transfer to the real data type before the record data is saved to the file.**

## 4. CRecordDlg class design

Create record dialog box class CRecordDlg to correspond to the IDD_RRCORD_DIALOG resource for entering a record.




**(1) data member**

CTableEntity* m_pTableEntity: table information entity

CRecordEntity m_recordEntity： record entity

**(2) member function**

void SetTable(CTableEntity* pTable): set the value of m_pTableEntity

virtual BOOL OnInitDialog(): Get the field name and data type from m_pTableEntity and display them in the list.

void OnBnClickedOk()： Take out the record data in the list, and assemble to the m_recordEntity.

CRecordEntity GetRecord(void): return to the record entity.

# Implementation



Iterative development based on the “Show table structure” function. Receive entered record information from CRecordDlg, and compose to the record entity CRecordDlg object, then save this record to the record file (*.trd).

The implementation steps are as follows:

**Step 1: define entity class CRecordEntity**

**Step 2: add record dialog box CRecordDlg**

**Step 3: receive entered record information in the view layer**

**Step 4: insert record in the logic layer CRecordLogic**

**Step 5: insert record in the data access layer CRecordDao**

**Step 6: modify record number in the data access layer CTableDao**

www.ruanko.com

Thanks

Insert record