

ICS 29.240.30

K 07

备案号: 50776-2015

**DL**

# 中华人民共和国电力行业标准

DL/T 1456 — 2015

---

## 电力系统数据库通用访问接口规范

General call interface specification of database for electric power system

2015-07-01 发布

2015-12-01 实施

---

国家能源局 发布

目 次

前言 ..... II

1 范围 ..... 1

2 规范性引用文件 ..... 1

3 术语和定义 ..... 1

4 缩略语 ..... 2

5 数据库功能和类型 ..... 2

6 实时数据库访问接口 ..... 3

7 时间序列实时数据库访问接口 ..... 10

8 时间序列历史数据库访问接口 ..... 14

9 关系数据库访问接口（DCI） ..... 18

附录 A（规范性附录） 实时数据库访问接口数据结构 ..... 30

附录 B（规范性附录） 时间序列实时数据库访问接口数据结构 ..... 31

附录 C（规范性附录） 时间序列历史数据库访问接口数据结构 ..... 32

附录 D（规范性附录） 关系数据库访问接口（DCI）中使用的句柄及描述符 ..... 33

# 前 言

为规范电网调度控制系统、配电自动化系统等应用软件对各类数据库的访问接口，指导相关系统及其应用功能的设计开发、测试验证和升级改造，制定本标准。

本标准按照 GB/T 1.1—2009《标准化工作导则 第 1 部分：标准的结构和编写》给出的规则起草。

本标准由中国电力企业联合会提出。

本标准由全国电网运行与控制标准化技术委员会（SAC/TC 446）归口。

本标准起草单位：南京南瑞集团公司、国家电网公司国家电力调度控制中心、南方电网有限责任公司电力调度控制中心、中国电力科学研究院、北京四方继保自动化股份有限公司、武汉达梦数据库有限公司、北京人大金仓信息技术股份有限公司、国家电网公司华中分部、国网江苏省电力公司、国网江西省电力公司、国网浙江省电力公司。

本标准主要起草人：严亚勤、梁寿愚、翟明玉、陶洪铸、彭晖、陈鹏、张伟、葛云鹏、王瑾、吴庆曦、谢晓冬、张杰、李海华、马志斌、厉启鹏、花静、李宇佳、闫湖、詹庆才、刘涛、金学成、徐春雷、占震滨、于宏文。

本标准在执行过程中的意见或建议请反馈至中国电力企业联合会标准化管理中心（北京市白广路二条一号，100761）。

# 电力系统数据库通用访问接口规范

## 1 范围

本标准规定了电力系统实时数据库、时间序列实时数据库、时间序列历史数据库和关系数据库的通用访问接口，明确了接口功能和参数定义等技术要求。

本标准适用于电网调度控制系统、配电自动化系统及其他与电力生产运行相关的应用软件的设计开发、测试验证、升级改造等工作，厂站监控系统可参照执行。

## 2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 12991.1 信息技术 数据库语言 SQL 第 1 部分：框架（ISO/IEC 9075-1，IDT）

GB/T 30149 电网通用模型描述规范

ISO/IEC 9075-2 Information technology—Database languages—SQL—Part 2: Foundation

ISO/IEC 9075-3 Information technology—Database languages—SQL—Part 3: Call-Level Interface

## 3 术语和定义

下列术语和定义适用于本标准。

### 3.1

#### 应用 application

在电力系统应用软件中，用于完成特定任务、互相紧密关联的一组功能集合，包括稳态监控应用、网络分析应用、自动发电控制应用等。

### 3.2

#### 态 context

在电力系统应用软件中，用于描述应用的运行环境，通常与时间相关，包括实时态、研究态、规划态、测试态、反演态、培训态等。每个态中包含一组应用集合，如实时态中可包含稳态监控应用和网络分析应用等。

### 3.3

#### 实时数据 real-time data

不断更新、对处理实时性要求较高的数据，用于电力系统实时监视、控制和分析计算等。

### 3.4

#### 时间序列数据 time-series data

带有时间顺序标记或按照时间顺序排列的数据，能够反映事物随时间变化的情况，适合描述电网运行动态或暂态过程，简称时序数据。

### 3.5

#### 关系型数据 relational data

可表示为关系数学模型行列二维表的数据，主要用于保存告警、事件记录、采样数据、统计分析等大批量历史数据，也用于保存电网模型、系统配置等其他数据，由关系数据库进行存储与管理。

4 缩略语

下列缩略语适用于本标准。

SQL：结构化查询语言（structured query language）。

DCI：数据库访问接口（database call interface）。

CIM/E：基于 E 的 CIM 模型（E based CIM model）。

5 数据库功能和类型

电力系统的实时数据库、时间序列实时数据库、时间序列历史数据库和关系数据库应通过本标准定义的通用访问接口，为电力系统软件提供数据访问和管理功能，其技术要求如下：

- a) 实时数据库应提供稳态实时数据的高效存取功能，支持电网运行实时监视和分析计算。
- b) 时间序列实时数据库应提供实时断面数据的高速访问功能，支持对电网运行动态过程的监视与在线分析计算。
- c) 时间序列历史数据库应提供时间序列数据的高效压缩归档与快速查询功能，支持对电网运行动态过程的历史重演与离线分析计算。
- d) 关系数据库应提供历史采样数据、各类业务模型和历史数据等非实时数据的存储功能，适用于数据保存时间长、实时性不高的场合。

数据库通用访问接口中所使用的基本数据类型见表 1。

表 1 基本数据类型

类 型	描 述
CHAR	字符型数据，长度一个字节
UCHAR	无符号字符型数据，长度一个字节
STRING	字符串，声明时指定长度
INT2	带符号双字节整数
INT4	带符号四字节整数
UINT2	无符号双字节整数
UINT4	无符号四字节整数
FLOAT	单精度四字节浮点数
MEM	内存缓冲区
ADDR	内存缓冲区地址
.....	.....

数据库通用访问接口中使用的数据结构描述见表 2，具体定义见附录 A～附录 C。

表 2 数据结构描述

数据结构名称	中文名称	描 述
RTDB_TB_DESCR	表描述符	定义实时数据库中的数据表
RTDB_TABLE_PARA_STRU	表信息结构	定义实时数据库数据表的各类属性
WRDB_DESCR	连接描述符	定义时间序列实时数据库的连接信息和连接状态
WRDB_POINTCONFIG	测点结构	定义时间序列实时数据库中数据测点的信息

表 2（续）

数据结构名称	中文名称	描 述
WHDB_DESCR	库描述符	定义时间序列历史数据库的连接信息和连接状态
WHDB_MEASUREMENT_CONFIG	测点配置结构	定义时间序列历史数据库的测点配置信息
WHDB_VALUE	遥测量结构	定义时间序列历史数据库的遥测量信息
WHDB_DIG_VALUE	遥信量结构	定义时间序列历史数据库的遥信量信息

附录 D 定义了关系数据库 DCI 接口中使用的句柄及描述符。

6 实时数据库访问接口

6.1 访问接口功能

实时数据库访问接口应支持以下功能：

- a) 打开实时数据库表。
- b) 读取实时数据，支持全表读取、记录读取、表域读取；读取数据字典，包括表属性和表域属性。
- c) 更新实时数据，包括表域数据更新和记录更新。
- d) 插入、删除和清空实时数据库表记录。
- e) 关闭、删除实时数据库表。
- f) 支持通过 SQL 语句读取、操作实时数据，SQL 语句的格式应遵循 GB/T 12991.1 和 ISO/IEC 9075-2 的规定。

实时数据库访问接口应是线程安全的，应至少提供表 3 所列的访问接口。

表 3 实时数据库主要访问接口

序号	接口名	描 述
1	CreateTable	创建表
2	OpenTableByName	按名称打开数据表
3	GetTablePara	读取表参数
4	GetFieldsPara	读取表域参数
5	ClearTable	清空记录
6	CloseTable	关闭数据表
7	DropTable	删除表
8	GetTableRecs	读全表记录
9	GetTableFields	读表的一个或多个表域数据
10	GetFieldsByID	通过关键字读一条记录的一个或多个表域
11	GetRecByID	通过关键字读一条记录
12	GetRecsByID	通过关键字读多条记录
13	InsertRecs	插入一条或多条记录
14	UpdateRecByID	通过关键字更新一条记录
15	UpdateRecsByID	通过关键字批量更新多条记录

表 3（续）

序号	接口名	描 述
16	UpdateFieldsById	通过关键字更新一条记录一个或多个表域
17	DeleteRecById	通过关键字删除一条记录
18	DeleteRecsById	通过关键字批量删除多条记录
19	ManipulateDataBySql	按 SQL 语句的方式操纵记录
20	GetValueAndStatus	读测点的数值和状态

6.2 访问接口定义

6.2.1 接口返回值

实时数据库访问接口的返回值一般为 INT4 类型，应至少支持表 4 所列的返回值。

表 4 实时数据库访问接口返回值

返 回 值	含 义
DB_OK	执行正确
ERR_PSWD_ERR	错误的用户名或口令
ERR_RTDB_PARA	非法或者异常的参数
ERR_RTDB_MEMORY	分配内存时出错
ERR_RTDB_INVSIZE	缓冲大小错误
ERR_RTDB_INVKEY	非法的关键字
ERR_RTDB_KEYFLEN	关键字长度错误
ERR_RTDB_EXISTREC	记录已存在
ERR_RTDB_NEXISTREC	记录不存在
ERR_RTDB_EMPTY	数据表没有数据
ERR_RTDB_DBFULL	表中数据达到容量上限
ERR_RTDB_TABLE_STA	数据表状态异常
ERR_RTDB_SQL	非法的 SQL 语句
ERR_RTDB_FIELD_NO	非法的域标识
ERR_RTDB_TABLE_NO	非法的数据表标识
ERR_RTDB_APP_NO	非法的应用标识
ERR_RTDB_CONTEXT_NO	非法的态标识
ERR_HOST_INVALID	连接位置错误
ERR_RTDB_EXISTTAB	表已存在
ERR_DUP_RECORD	存在重复记录

6.2.2 创建表 CreateTable

接口原型：

```
CreateTable ( table_para    IN  RTDB_TABLE_PARA_STRU,    //表属性
              field_para    IN  MEM,                    //表域属性
              user           IN  STRING,                 //用户名
              passwd         IN  STRING                  //用户口令或数字证书
            )
RETURN INT4
```

注：引用自 ISO/IEC 9075-3 的规定，CreateTable 为接口名称，table\_para、field\_para、user 和 passwd 为接口的参数名称，IN、OUT 和 INOUT 为输入参数、输出参数和输入/输出参数，RTDB\_TABLE\_PARA\_STRU、MEM 和 STRING 为参数的数据类型，双斜杠后的内容为参数描述，以下同。RTDB\_TABLE\_PARA\_STRU 的详细数据结构说明见附录 A。

功能描述：创建一张数据表。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_EXISTTAB 等。

6.2.3 按名称打开表 OpenTableByName

接口原型：

```
OpenTableByName ( host_name    IN      STRING,          //表所在的主机名
                  context_name IN      STRING,          //表所属的态标识
                  app_name     IN      STRING,          //表所属的应用标识
                  table_name    IN      STRING,          //表名
                  user          IN      STRING,          //用户名
                  passwd        IN      STRING,          //用户口令或数字证书
                  tb_descr      OUT     RTDB_TB_DESCR    //表描述符
                )
RETURN INT4
```

注：RTDB\_TB\_DESCR 的详细数据结构说明见附录 A。

功能描述：按名称打开表。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_NO、ERR\_RTDB\_APP\_NO、ERR\_RTDB\_CONTEXT\_NO、ERR\_HOST\_INVALID、ERR\_PSWD\_ERR 等。

6.2.4 读取表参数 GetTablePara

接口原型：

```
GetTablePara ( tb_descr      IN      RTDB_TB_DESCR,    //表描述符
               table_para    OUT     RTDB_TABLE_PARA_STRU //表参数信息
            )
RETURN INT4
```

功能描述：读取表参数，获取数据表的可访问属性。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA 等。

6.2.5 读取表域参数 GetFieldsPara

接口原型：

```
GetFieldsPara ( tb_descr      IN      RTDB_TB_DESCR,    //表描述符
                field_para    OUT     ADDR               //返回的表域属性数据地址
            )
```



RETURNS INT4

功能描述：读取表域参数，获取数据表域的可访问属性。

返回值说明：执行成功返回表域信息具体大小，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA 等。

6.2.6 清空记录 ClearTable

接口原型：

```
ClearTable ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
             user          IN      STRING,         //用户名
             passwd        IN      STRING          //用户口令或数字证书
            )
```

RETURNS INT4

功能描述：清空记录。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_PSWD\_ERR 等。

6.2.7 关闭数据表 CloseTable

接口原型：

```
CloseTable ( tb_descr      IN      RTDB_TB_DESCR //表描述符
            )
```

RETURNS INT4

功能描述：关闭数据表。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY 等。

6.2.8 删除数据表 DropTable

接口原型：

```
DropTable ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
            user          IN      STRING,         //用户名
            passwd        IN      STRING          //用户口令或数字证书
            )
```

RETURNS INT4

功能描述：删除一张数据表。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_PSWD\_ERR 等。

6.2.9 读取全表记录 GetTableRecs

接口原型：

```
GetTableRecs ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
                buf_size      OUT     INT4,          //返回的数据长度
                data_buffer    OUT     ADDR          //返回的数据地址
               )
```

RETURNS INT4

功能描述：读取数据表全部记录数据。

返回值说明：执行成功返回获取数据缓冲的具体大小，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_EMPTY 等。

6.2.10 读取表域数据 **GetTableFields**

接口原型:

```
GetTableFields ( tb_descr          IN          RTDB_TB_DESCR, //表描述符
                  query_fields     IN          STRING,         //被查询的表域名
                  buf_size         OUT         INT4,            //返回的数据长度
                  data_buffer      OUT         ADDR             //返回的数据地址
                )
                RETURNS INT4
```

功能描述: 读表的一个或多个表域数据。

返回值说明: 执行成功获取数据缓冲的具体大小, 执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_FIELD\_NO、ERR\_RTDB\_EMPTY 等。

6.2.11 按关键字读取表域数据 **GetFieldsByID**

接口原型:

```
GetFieldsByID ( tb_descr          IN          RTDB_TB_DESCR, //表描述符
                  keyword         IN          MEM,             //用来查询的关键字
                  query_fields     IN          STRING,         //被查询的表域名
                  buf_size         OUT         INT4,            //返回的数据长度
                  data_buffer      OUT         ADDR             //返回的数据地址
                )
                RETURNS INT4
```

功能描述: 通过关键字读一条记录的一个或多个表域数据。

返回值说明: 执行成功返回 DB\_OK, 执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_FIELD\_NO、ERR\_RTDB\_INVSIZE、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

6.2.12 按关键字读取一条记录 **GetRecByID**

接口原型:

```
GetRecByID ( tb_descr          IN          RTDB_TB_DESCR, //表描述符
              keyword         IN          MEM,             //用来查询的关键字
              buf_size         OUT         INT4,            //返回的数据长度
              data_buffer      OUT         ADDR             //返回的数据地址
            )
            RETURNS INT4
```

功能描述: 通过关键字读一条记录。

返回值说明: 执行成功返回 DB\_OK, 执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_INVSIZE、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

6.2.13 按关键字读取多条记录 **GetRecsByID**

接口原型:

```
GetRecsByID ( tb_descr          IN          RTDB_TB_DESCR, //表描述符
               keyword         IN          MEM,             //用来查询的关键字
               key_num         IN          INT4,            //用来查询的关键字个数
               buf_size         OUT         INT4,            //返回的数据长度
               data_buffer      OUT         ADDR             //返回的数据地址
            )
```

)  
RETURNS INT4

功能描述：通过单个或多个关键字读取多条记录。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_INVSIZE、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

6.2.14 插入一条或多条记录 **InsertRecs**

接口原型：

```
InsertRecs ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
             rec_num       IN      INT4,           //插入的记录个数
             buf_size      IN      INT4,           //插入的数据长度
             data_buffer   IN      MEM              //插入的数据内容
           )
RETURNS INT4
```

功能描述：插入一条或多条记录。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_INVSIZE、ERR\_RTDB\_INVKEY、ERR\_RTDB\_EXISTREC 等。

6.2.15 按关键字更新一条记录 **UpdateRecByID**

接口原型：

```
UpdateRecByID ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
                keyword       IN      MEM,           //更新记录的关键字
                buf_size      IN      INT4,           //更新的数据长度
                data_buffer   IN      MEM              //更新的数据内容
              )
RETURNS INT4
```

功能描述：通过关键字更新一条记录。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_INVSIZE、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

6.2.16 按关键字批量更新记录 **UpdateRecsByID**

接口原型：

```
UpdateRecsByID ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
                 keyword       IN      MEM,           //批量更新的关键字结构
                 key_num       IN      INT4,           //更新记录的关键字个数
                 buf_size      IN      INT4,           //用于更新的数据长度
                 data_buffer   IN      MEM              //用于更新的数据
               )
RETURNS INT4
```

功能描述：通过关键字批量更新记录。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_INVSIZE、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

6.2.17 按关键字更新一条记录的表域数据 **UpdateFieldsByID**

接口原型:

```
UpdateFieldsByID ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
                  keyword      IN      MEM,             //更新记录的关键字
                  update_fields IN      STRING,          //更新的表域名
                  buf_size     IN      INT4,             //更新的数据长度
                  data_buffer   IN      MEM              //更新的数据内容
                  )
RETURN INT4
```

功能描述: 通过关键字更新一条记录的一个或多个表域数据。

返回值说明: 执行成功返回 DB\_OK, 执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_INVSIZE、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY、ERR\_DUP\_RECORD 等。

6.2.18 按关键字删除一条记录 **DeleteRecByID**

接口原型:

```
DeleteRecByID ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
               keyword      IN      MEM              //删除记录的关键字
               )
RETURN INT4
```

功能描述: 通过关键字删除一条记录。

返回值说明: 执行成功返回 DB\_OK, 执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

6.2.19 按关键字批量删除记录 **DeleteRecsByID**

接口原型:

```
DeleteRecsByID ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
                keyword      IN      MEM ,            //批量删除记录的关键字结构
                key_num      IN      INT4              //删除记录的个数
                )
RETURN INT4
```

功能描述: 通过关键字批量删除记录。

返回值说明: 执行成功返回 DB\_OK, 执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

6.2.20 以 SQL 语句的方式操纵记录 **ManipulateDataBySql**

接口原型:

```
ManipulateDataBySql ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
                    sql_str      IN      STRING ,          //操纵记录的SQL语句
                    sql_type     IN      INT2,             //SQL语句的类型
                    buf_size     OUT     INT4,             //返回的记录数据长度
                    data_buffer   OUT     ADDR             //返回的记录数据地址
                    )
RETURN INT4
```

注: SQL 语句的类型应支持查询、插入和更新, 不应支持删除。查询类型的 SQL 语句应支持带 where 语句的简单过滤查询, 至少应包括=、<、>、<>、≤、≥、like。

功能描述：以 SQL 语句的方式操纵记录。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_SQL、ERR\_RTDB\_FIELD\_NO、ERR\_RTDB\_APP\_NO 等。

6.2.21 获取多个测点的数值和状态 GetValueAndStatus

接口原型：

```
GetValueAndStatus ( tb_descr      IN      RTDB_TB_DESCR, //表描述符
                    keyid         IN      MEM,           //存储一个或多个测点标识
                                                            //的内存缓冲区
                    key_num       IN      INT4,          //测点个数
                    value         OUT     ADDR,          //指向测点数值的地址
                    status        OUT     ADDR          //指向测点状态的地址
                    )
    RETURNS INT4
```

功能描述：获取多个测点的数值和状态。

返回值说明：执行成功返回 DB\_OK，执行错误时可能的返回值包括 ERR\_RTDB\_PARA、ERR\_RTDB\_MEMORY、ERR\_RTDB\_TABLE\_STA、ERR\_RTDB\_FIELD\_NO、ERR\_RTDB\_NEXISTREC、ERR\_RTDB\_INVKEY 等。

7 时间序列实时数据库访问接口

7.1 访问接口功能

时间序列实时数据库访问接口应支持以下功能：

- a) 打开时间序列实时数据库。
- b) 插入、删除、修改和查询测点。
- c) 读取和写入测点的时序数据。
- d) 关闭时间序列实时数据库。

时间序列实时数据库提供的接口应是线程安全的，应至少提供表 5 所列的访问接口。

表 5 时间序列实时数据库主要访问接口

序号	接口名	描 述
1	WRDB_Open	打开数据库
2	WRDB_Close	关闭数据库
3	WRDB_Insert	插入测点
4	WRDB_Delete	删除测点
5	WRDB_Update	修改测点信息
6	WRDB_List	获取全部测点信息
7	WRDB_Write	写入遥测测点数据
8	WRDB_Read	读取遥测测点数据
9	WRDB_WriteDig	写入遥信测点数据
10	WRDB_ReadDig	读取遥信测点数据
注：WRDB 为 WAMS realtime database。		

7.2 访问接口定义

7.2.1 接口返回值

时间序列实时数据库访问接口的返回值一般为 INT4 类型，应至少具有表 6 所列的返回值。

表 6 时间序列实时数据库访问接口返回值

返回值	含 义
RETURN_OK	正确返回
ERROR_PSWD_ERR	错误的用户名或口令
ERROR_PARAMETER	非法的参数
ERROR_WRDB_OPEN	数据库打开错误
ERROR_RECORD_INVAILD	非法的测点标识
ERROR_POINT_CONFIG	非法的测点定义
ERROR_POINT_FULL	测点达到最大值，无法定义新测点
ERROR_POINT_DELETE	无法删除测点
ERROR_POINT_NONE	未找到该测点
ERROR_POINT_TIME	非法的查询时间
ERROR_CONNECT_ERROR	数据库连接错误
ERR_HOST_INVALID	非法的数据库服务器

7.2.2 打开数据库 **WRDB\_Open**

接口原型：

```
WRDB_Open ( server_url    IN      STRING,           //数据库连接位置
             descr        OUT    WRDB_DESCR,        //连接描述符
             user         IN      STRING,           //用户名
             passwd       IN      STRING            //用户口令或数字证书
           )
           RETURNS INT4
```

注：WRDB\_DESCR 的详细数据结构说明见附录 B。

功能描述：打开数据库。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_PARAMETER、ERR\_HOST\_INVALID、ERROR\_PSWD\_ERR 等。

7.2.3 关闭数据库 **WRDB\_Close**

接口原型：

```
WRDB_Close ( descr    IN  WRDB_DESCR    //连接描述符
           )
           RETURNS INT4
```

功能描述：关闭数据库。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_PARAMETER 等。

7.2.4 插入测点 **WRDB\_Insert**

接口原型：

```
WRDB_Insert ( descr      IN      WRDB_DESCR,           //连接描述符
```

```
                pconfig      IN      WRDB_POINTCONFIG    //测点信息
            )
        RETURNS INT4
```

注：WRDB\_POINTCONFIG 的详细数据结构说明见附录 B。

功能描述：在数据库中插入一个测点。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_RECORD\_INVAILD、ERROR\_POINT\_CONFIG、ERROR\_POINT\_FULL、ERROR\_PARAMETER 等。

7.2.5 删除测点 **WRDB\_Delete**

接口原型：

```
        WRDB_Delete ( descr      IN      WRDB_DESCR,      //连接描述符
                    pconfig      IN      WRDB_POINTCONFIG  //测点信息
                )
        RETURNS INT4
```

功能描述：在数据库中删除一个测点。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_RECORD\_INVAILD、ERROR\_POINT\_DELETE、ERROR\_POINT\_NONE、ERROR\_PARAMETER 等。

7.2.6 修改测点信息 **WRDB\_Update**

接口原型：

```
        WRDB_Update ( descr      IN      WRDB_DESCR,      //连接描述符
                    pconfig      IN      WRDB_POINTCONFIG  //测点信息
                )
        RETURNS INT4
```

功能描述：修改数据库中一个测点的信息。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_RECORD\_INVAILD、ERROR\_POINT\_NONE、ERROR\_PARAMETER 等。

7.2.7 列出全部测点信息 **WRDB\_List**

接口原型：

```
        WRDB_List ( descr      IN      WRDB_DESCR ,      //连接描述符
                    pconfig      OUT     ADDR,            //测点信息
                    count        OUT     INT4              //测点个数
                )
        RETURNS INT4
```

功能描述：列出数据库中所有的测点信息。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_PARAMETER 等。

7.2.8 写入遥测测点数据 **WRDB\_Write**

接口原型：

```
        WRDB_Write ( descr      IN      WRDB_DESCR,      //连接描述符
                    meas_id      IN      INT4,            //测点标识
                    begin_tm      IN      INT4,            //写入起始时刻
                    end_tm        IN      INT4,            //写入结束时刻
                    size           IN      INT4,            //写入的数据大小
                    values         IN      MEM,             //数据值
                )
```

```
        status      IN      MEM      //数据质量
    )
    RETURNS INT4
```

功能描述：写入遥测测点的时序数据。

返回值：执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD、ERROR\_PSWD\_ERR 等。

7.2.9 读取遥测测点数据 WRDB\_Read

接口原型：

```
WRDB_Read ( descr      IN      WRDB_DESCR,    //连接描述符
            meas_id     IN      INT4,          //测点标识
            begin_tm     IN      INT4,          //查询起始时刻
            end_tm       IN      INT4,          //查询结束时刻
            size         OUT     INT4,          //查询到的数据长度
            values       OUT     ADDR,          //查询到的数据地址
            status       OUT     ADDR           //数据质量
    )
    RETURNS INT4
```

功能描述：读取遥测测点的时序数据。

返回值：执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD、ERROR\_POINT\_TIME 等。

7.2.10 写入遥信测点数据 WRDB\_WriteDig

接口原型：

```
WRDB_WriteDig ( descr      IN      WRDB_DESCR,    //连接描述符
                meas_id     IN      INT4,          //测点标识
                begin_tm     IN      INT4,          //写入起始时刻
                end_tm       IN      INT4,          //写入结束时刻
                size         IN      INT4,          //写入的数据大小
                values       IN      MEM,           //数据值
                status       IN      MEM           //数据质量
    )
    RETURNS INT4
```

功能描述：写入遥信测点的时序数据。

返回值：执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD 等。

7.2.11 读取遥信测点数据 WRDB\_ReadDig

接口原型：

```
WRDB_ReadDig ( descr      IN      WRDB_DESCR,    //连接描述符
               meas_id     IN      INT4,          //测点标识
               begin_tm     IN      INT4,          //查询起始时刻
               end_tm       IN      INT4,          //查询结束时刻
               size         OUT     INT4,          //查询到的数据长度
               values       OUT     ADDR,          //查询到的数据地址
               status       OUT     ADDR           //数据质量
    )
```



)

RETURNS INT4

功能描述：读取遥信测点的时序数据。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD、ERROR\_POINT\_TIME 等。

8 时间序列历史数据库访问接口

8.1 访问接口功能

时间序列历史数据库访问接口应支持以下功能：

- a) 打开数据库。
- b) 插入、删除测点信息。
- c) 查询测点信息。
- d) 读取测点的时序数据，包括通过 SQL 语句读取数据。
- e) 写入测点的时序数据。
- f) 关闭数据库。

时间序列历史数据库提供的接口应是线程安全的，应至少提供表 7 所列的访问接口。

表 7 时间序列历史数据库主要访问接口

序号	接口名	描 述
1	WHDB_Open	打开数据库
2	WHDB_Close	关闭数据库
3	WHDB_Insert	插入测点
4	WHDB_Delete	删除测点
5	WHDB_List	查询全部测点信息
6	WHDB_Write	写入非等间隔遥测测点数据
7	WHDB_WriteETI	写入等间隔遥测测点数据
8	WHDB_Read	读取遥测测点数据
9	WHDB_ReadETI	读取等间隔遥测测点数据
10	WHDB_WriteDig	写入遥信测点数据
11	WHDB_ReadDig	读取遥信测点数据
12	WHDB_GetBySql	按照 Sql 语句读取记录
注：WHDB 为 WAMS history database。		

8.2 访问接口定义

8.2.1 接口返回值

时间序列历史数据库访问接口的返回值一般为 INT4 类型，应至少具有表 8 所列的返回值。

表 8 时间序列历史数据库访问接口返回值

返 回 值	含 义
RETURN_OK	执行成功

表 8（续）

返 回 值	含 义
RETURN_FAIL	执行失败
ERROR_PSWD_ERR	错误的用户名或口令
ERROR_PARAMETER	参数错误
ERROR_OPEN	数据库打开错误
ERROR_RECORD_INVAILD	非法的测点标识
ERROR_POINT_CONFIG	非法的测点定义
ERROR_POINT_FULL	测点达到最大值，无法定义新测点
ERROR_POINT_NONE	未找到该测点
ERROR_TIME	非法的查询时间
ERROR_CONNECT_ERROR	数据库连接错误

8.2.2 打开数据库 **WHDB\_Open**

接口原型：

```
WHDB_Open ( server_url    IN      STRING,           //数据库连接位置
            descr         OUT    WHDB_DESCR,        //数据库描述符
            user          IN      STRING,           //用户名
            passwd        IN      STRING            //用户口令或数字证书
            )
            RETURNS INT4
```

注：WHDB\_DESCR 的详细数据结构说明见附录 C。

功能描述：打开数据库。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_PARAMETER、ERROR\_PSWD\_ERR 等。

8.2.3 关闭数据库 **WHDB\_Close**

接口原型：

```
WHDB_Close ( descr        IN      WHDB_DESCR        //数据库描述符
            )
            RETURNS INT4
```

功能描述：关闭数据库。

返回值：执行成功返回 RETURN\_OK，执行错误时返回 RETURN\_FAIL。

8.2.4 插入测点 **WHDB\_Insert**

接口原型：

```
WHDB_Insert ( descr        IN      WHDB_DESCR,           //数据库描述符
            meas_config    IN      WHDB_MEASUREMENT_CONFIG //数据库测点数据结构
            )
            RETURNS INT4
```

注：WHDB\_MEASUREMENT\_CONFIG 的详细数据结构说明见附录 C。

功能描述：插入一个新的测点。

返回值：执行成功返回 RETURN\_OK，执行错误时可能的返回值包括 ERROR\_RECORD\_INVAILD、

ERROR\_POINT\_CONFIG 和 ERROR\_POINT\_FULL 等。

8.2.5 删除测点 WHDB\_Delete

接口原型:

```
WHDB_Delete ( descr      IN   WHDB_DESCR,    //数据库描述符
               meas_id   IN   UINT4           //删除测点的标识
             )
            RETURNS INT4
```

功能描述: 删除已有的测点。

返回值: 执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_RECORD\_INVAILD、ERROR\_POINT\_NONE 等。

8.2.6 查询全部测点信息 WHDB\_List

接口原型:

```
WHDB_List ( descr      IN       WHDB_DESCR ,    //数据库描述符
             meas_config OUT     ADDR,           //保存测点信息的数组地址
             count      OUT     INT4             //实际读出的测点信息记录数量
           )
          RETURNS INT4
```

功能描述: 查询数据库中的全部测点信息。

返回值: 执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_PARAMETER 等。

8.2.7 写入遥测测点数据 WHDB\_Write

接口原型:

```
WHDB_Write ( descr      IN       WHDB_DESCR,    //数据库描述符
              count      IN       INT4,          //写入时间序列数据值的数量
              whdb_value IN       ADDR           //指向时间序列数据值数组的地址
            )
            RETURNS INT4
```

注: whdb\_value 参数所指向的时间序列数据值数组应采用 WHDB\_VALUE 数据结构定义, WHDB\_VALUE 的详细信息结构说明见附录 C。

功能描述: 向数据库中写入多个时间序列数据值, 可支持写入不等间隔的时序数据。

返回值: 执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD 等。

8.2.8 写入等间隔遥测测点数据 WHDB\_WriteETI

接口原型:

```
WHDB_WriteETI ( descr      IN   WHDB_DESCR,    //数据库描述符
                 meas_count IN   INT4,          //写入测点数量
                 meas_ids   IN   MEM,           //写入测点标识数组
                 begin_tm    IN   INT4,          //以世纪秒表示的数据开始时间
                 end_tm      IN   INT4,          //以世纪秒表示的数据结束时间
                 interval    IN   INT4,          //写入数据记录的时间间隔
                 count       IN   INT4,          //写入每个测点数据值的数量
                 value       IN   ADDR,          //指向时间序列数据值数组的地址
                 val_flag    IN   ADDR           //指向时间序列数据质量数组的地址
               )
```

```
)  
RETURNS INT4
```

功能描述：向数据库中写入多个遥测测点的一段时间数据，时间序列数据值数组按测点顺序和时间顺序依次排列。

返回值：执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD 等。

8.2.9 读取遥测测点数据 WHDB\_Read

接口原型：

```
WHDB_Read ( descr          IN      WHDB_DESCR,    //数据库描述符  
             meas_count    IN      INT4,          //查询测点数量  
             meas_id       IN      INT4,          //查询测点标识  
             begin_tm      IN      INT4,          //以世纪秒表示的数据开始时间  
             end_tm        IN      INT4,          //以世纪秒表示的数据结束时间  
             count         OUT     INT4,          //读取数据值的数量  
             whdb_value    OUT     ADDR           //指向时间序列数据数组的地址  
)  
RETURNS INT4
```

功能描述：从数据库中读取多个遥测测点的一段时间内数据，时间序列数据值采用 WHDB\_VALUE 数据结构定义，按照测点顺序和时间顺序排列。

返回值：执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD 等。

8.2.10 读取等间隔遥测测点数据 WHDB\_ReadETI

接口原型：

```
WHDB_ReadETI ( descr      IN      WHDB_DESCR,    //数据库描述符  
               meas_count  IN      INT4,          //查询测点数量  
               meas_ids    IN      MEM,          //查询测点标识数组  
               begin_tm    IN      INT4,          //以世纪秒表示的数据开始时间  
               end_tm      IN      INT4,          //以世纪秒表示的数据结束时间  
               interval    IN      INT4,          //读取数据记录的时间间隔  
               count       OUT     INT4,          //读取数据值的数量  
               value       OUT     ADDR,          //指向时间序列数据值数组的地址  
               val_flag    OUT     ADDR           //指向时间序列数据质量数组的地址  
)  
RETURNS INT4
```

功能描述：从数据库中读取多个遥测测点一段时间的等间隔数据，数据值数组按照测点顺序和时间顺序排列。

返回值：执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD 等。

8.2.11 写入遥信测点数据 WHDB\_WriteDig

接口原型：

```
WHDB_WriteDig ( descr      IN      WHDB_DESCR,    //数据库描述符  
               count       IN      INT4,          //写入数据值的数量  
               whdb_dig_value IN      ADDR         //指向时间序列数据值数组的地址
```

)

RETURNS INT4

注: whdb\_dig\_value 参数所指向的时间序列数据值数组应采用 WHDB\_DIG\_VALUE 数据结构定义, WHDB\_DIG\_VALUE 的详细数据结构说明见附录 C。

功能描述: 向数据库中写入多个遥信测点数据值。

返回值: 执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD 等。

8.2.12 读取遥信测点数据 WHDB\_ReadDig

接口原型:

```
WHDB_ReadDig ( descr          IN      WHDB_DESCR,    //数据库描述符
               meas_count     IN      INT4,          //查询测点数量
               meas_id        IN      INT4,          //查询测点标识
               begin_tm       IN      INT4,          //以世纪秒表示的数据开始时间
               end_tm         IN      INT4,          //以世纪秒表示的数据结束时间
               count          OUT     INT4,          //读取数据值的数量
               whdb_dig_value OUT     ADDR           //指向时间序列数据值数组的地址
               )
```

RETURNS INT4

功能描述: 从数据库中读取多个遥信测点的一段时间内数据, 时间序列数据值采用 WHDB\_DIG\_VALUE 数据结构定义, 按照测点顺序和时间顺序排列。

返回值: 执行成功返回 RETURN\_OK, 执行错误时可能的返回值包括 ERROR\_CONNECT\_ERROR、ERROR\_POINT\_NONE、ERROR\_PARAMETER、ERROR\_RECORD\_INVAILD 等。

8.2.13 以 SQL 语句的方式获取测点数据 WHDB\_GetBySql

接口原型:

```
WHDB_GetBySql ( db_descr      IN      WHDB_DESCR,    //数据库描述符
               sql_stmt       IN      STRING,         //用于查询的SQL语句
               buf_size       OUT     INT4,          //返回的数据长度
               data_buffer    OUT     ADDR           //返回的结果数据地址
               )
```

RETURNS INT4

功能描述: 以 SQL 语句的方式获取测点数据。

返回值说明: 执行成功返回 RETURN\_OK, 执行错误时返回 RETURN\_FAIL。

9 关系数据库访问接口 (DCI)

9.1 访问接口功能

DCI 接口是一套访问和操作关系数据库中关系型数据的接口, 主要功能应包括:

- a) 创建 DCI 环境。
- b) 分配句柄。
- c) 登录数据库系统。
- d) 执行 SQL 语句, 并获取结果。其中, SQL 语句的格式应遵循 GB/T 12991.1 和 ISO/IEC 9075-2 的规定, SQL 接口的格式应遵循 ISO/IEC 9075-3 的规定。
- e) 退出数据库系统。
- f) 释放句柄。

- g) CIM/E 数据文件的导入导出，其中 CIM/E 文件格式应遵循 GB/T 30149 的规定。
  - h) 基于 CIM/E 模板文件创建数据表，其中 CIM/E 文件格式应遵循 GB/T 30149 的规定。
- 使用 DCI 接口的基本流程如图 1 所示。

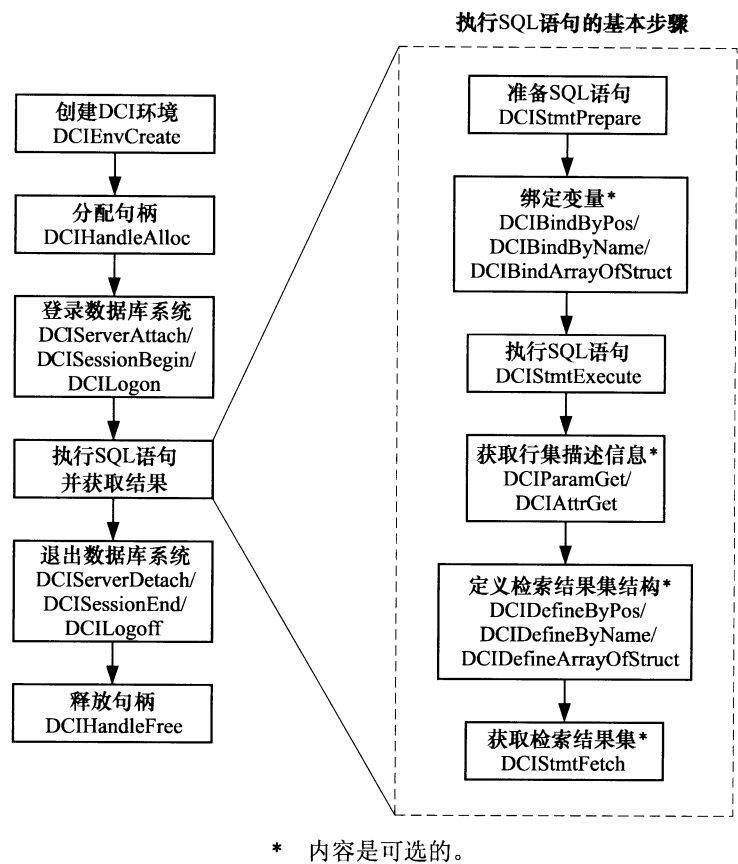


图 1 DCI 接口流程基本框架示意图

关系数据库提供的接口应是线程安全的，应至少提供表 9 所列的访问接口。

表 9 关系数据库主要 DCI 接口列表

序号	接口名	简要功能描述
1	DCIEnvCreate	创建并初始化 DCI 接口的工作环境
2	DCILogon	登录到数据库，并初始化相关句柄
3	DCILogoff	退出和数据库的连接
4	DCIServerAttach	建立连接句柄与数据库的关联
5	DCIServerDetach	解除连接句柄与数据库的关联
6	DCISessionBegin	打开连接句柄与数据库的连接
7	DCISessionEnd	关闭连接句柄与数据库的连接
8	DCIAttrGet	获取句柄上的属性值
9	DCIAttrSet	设置句柄上的属性值
10	DCIDescriptorAlloc	分配描述符

表 9（续）

序号	接口名	简要功能描述
11	DCIDescriptorFree	释放描述符
12	DCIHandleAlloc	分配并初始化各类应用句柄
13	DCIHandleFree	释放已分配的各类句柄
14	DCIParamGet	返回参数描述符
15	DCIParamSet	为句柄设置描述符
16	DCIBindArrayOfStruct	指定参数绑定数组的间隔大小
17	DCIBindByName	按名称绑定 SQL 语句中的参数
18	DCIBindByPos	按位置绑定 SQL 语句中的参数
19	DCIDefineArrayOfStruct	指定结果行集数组中每行的大小
20	DCIDefineByPos	按位置指定结果集中每列的存储空间
21	DCIDescribeAny	获取数据库对象的详细信息
22	DCIStmtPrepare	准备 SQL 语句
23	DCIStmtExecute	执行 SQL 语句
24	DCIStmtFetch	获取 SQL 语句执行后的结果
25	DCITransStart	启动事务
26	DCITransCommit	提交事务
27	DCITransRollback	回滚事务
28	DCIErrorGet	获取 DCI 操作失败时的错误信息
29	DCIServerVersion	返回服务器的版本信息
30	DCIServerSession	返回服务器当前连接数
31	DCICIMDataIn	将一个符合 CIM/E 标准的数据文件导入到数据库中
32	DCICIMDataOut	将表中数据导出成符合 CIM/E 标准的文件
33	DCICIMECreateTable	根据 CIM/E 标准的模板文件在数据库中建表

9.2 访问接口定义

9.2.1 接口返回值

关系数据库 DCI 接口的返回值一般为 INT2 类型，应至少具有表 10 所列的返回值。

表 10 DCI 接口返回值

返回值	描 述
DCI_SUCCESS	接口成功完成
DCI_SUCCESS_WITH_INFO	接口成功完成，带有附加信息
DCI_NO_DATA	接口成功完成，没有更多的数据
DCI_ERROR	接口执行失败
DCI_INVALID_HANDLE	非法的输入句柄参数

9.2.2 创建工作环境 **DCIEnvCreate**

接口原型:

```
DCIEnvCreate ( envhpp      OUT      ADDR,  //环境句柄
               mode       IN       INT4,   //初始化模式
               ctxp       IN       ADDR,   //用户自定义内存管理接口的地址
               malocfp    IN       ADDR,   //用户自定义内存分配接口的地址
               ralocfp    IN       ADDR,   //用户自定义内存重分配接口的地址
               mfreefp    IN       ADDR,   //用户自定义内存释放接口的地址
               xtramemsz  IN       UINT4,  //所分配内存的大小
               usrmempp   OUT      ADDR    //所分配内存的地址
            )
    RETURNS INT2
```

功能描述: 创建并初始化 DCI 接口的工作环境。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.3 登录数据库 **DCILogon**

接口原型:

```
DCILogon ( envhp      IN      ADDR,      //环境句柄
            errhp     INOUT   ADDR,      //错误信息句柄
            svchp     INOUT   ADDR,      //服务上下文句柄
            username   IN      ADDR,      //登录用户名的地址
            uname_len  IN      UINT4,     //登录用户名长度
            password   IN      ADDR,      //登录口令的地址
            passwd_len IN      UINT4,     //登录口令的长度
            dbname     IN      ADDR,      //数据库服务名的地址
            dbname_len IN      UINT4      //数据库服务名长度
        )
    RETURNS INT2
```

功能描述: 登录到数据库, 并初始化相关句柄。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.4 断开数据库 **DCILogoff**

接口原型:

```
DCILogoff ( svchp      IN      ADDR,      //服务上下文句柄
            errhp     INOUT   ADDR      //错误信息句柄
        )
    RETURNS INT2
```

功能描述: 断开与数据库的连接。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.5 建立连接句柄与数据库的关联 **DCIServerAttach**

接口原型:

```
DCIServerAttach ( srvhp      INOUT   ADDR,  //数据库服务句柄
                 errhp     INOUT   ADDR,  //错误信息句柄
                 dblink    IN      ADDR,  //数据库服务名的地址
                 dblink_len IN      INT4,  //数据库服务名的长度
                 mode      IN      UINT4   //数据库连接模式
            )
    RETURNS INT2
```



功能描述：建立连接句柄与数据库的关联。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.6 解除连接句柄与数据库的关联 **DCIServerDetach**

接口原型：

```
DCIServerDetach ( srvhp      IN      ADDR, //数据库服务句柄
                  errhp      INOUT   ADDR, //错误信息句柄
                  mode       IN      UINT4 //数据库连接模式
                )
                RETURNS INT2
```

功能描述：解除连接句柄与数据库的关联。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.7 打开连接句柄与数据库的连接 **DCISessionBegin**

接口原型：

```
DCISessionBegin ( svchp      IN      ADDR, //服务上下文句柄
                  errhp      IN      ADDR, //错误信息句柄
                  usrh       INOUT   ADDR, //用户会话句柄
                  credt       IN      UINT4, //会话认证方式
                  mode        IN      UINT4 //会话模式
                )
                RETURNS INT2
```

功能描述：打开连接句柄与数据库的连接。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.8 关闭连接句柄与数据库的连接 **DCISessionEnd**

接口原型：

```
DCISessionEnd ( svchp      INOUT   ADDR, //服务上下文句柄
                 errhp      INOUT   ADDR, //错误信息句柄
                 usrh       IN      ADDR, //用户会话句柄
                 mode        IN      UINT4 //会话模式
               )
               RETURNS INT2
```

功能描述：关闭连接句柄与数据库的连接。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.9 获取句柄属性 **DCIAttrGet**

接口原型：

```
DCIAttrGet ( trgthndlp      IN      ADDR, //获取属性的句柄
             trghndltyp      IN      UINT4, //获取属性句柄的类型
             attributep       OUT     ADDR, //属性内存的地址
             sizep            OUT     UINT4, //属性内存的大小
             attrtype         IN      UINT4, //属性类型
             errhp            INOUT   ADDR //错误信息句柄
           )
           RETURNS INT2
```

功能描述：获取句柄上的属性值。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.10 设置句柄属性 DCIAttrSet

接口原型：

```
DCIAttrSet ( trghndlp      IN      ADDR,  //设置属性的句柄
             trghndltyp    IN      UINT4,  //设置属性句柄的类型
             attributep    IN      ADDR,  //属性内存的地址
             size          IN      UINT4,  //属性内存的大小
             attrtype      IN      UINT4,  //属性类型
             errhp         INOUT  ADDR    //错误信息句柄
           )
           RETURNS INT2
```

功能描述：设置句柄上的属性值。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.11 分配描述符的内存空间 DCIDDescriptorAlloc

接口原型：

```
DCIDDescriptorAlloc ( parenth  IN      ADDR,  //环境句柄
                     descpp    OUT     ADDR,  //描述符的地址
                     type      IN      UINT4,  //描述符类型
                     xtrmem_sz IN      UINT4,  //用户自定义分配内存的大小
                     usrmempp  OUT     ADDR    //用户自定义分配内存的地址
                   )
                   RETURNS INT2
```

功能描述：为描述符分配内存空间。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.12 释放描述符的内存空间 DCIDDescriptorFree

接口原型：

```
DCIDDescriptorFree ( descp     IN      ADDR,  //释放描述符的地址
                    type       IN      UINT4   //释放描述符的类型
                  )
                  RETURNS INT2
```

功能描述：释放描述符的内存空间。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.13 分配并初始化各类应用句柄 DCIHandleAlloc

接口原型：

```
DCIHandleAlloc ( parenth  IN      ADDR,  //环境句柄
                hndlpp     OUT     ADDR,  //新分配句柄的地址
                type       IN      UINT4,  //新分配句柄的类型
                xtrmem_sz  IN      UINT4,  //用户自定义分配内存的大小
                usrmempp   OUT     ADDR    //用户自定义分配内存的地址
              )
              RETURNS INT2
```

功能描述：分配并初始化各类应用句柄。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.14 释放已分配的各类句柄 **DCIHandleFree**

接口原型:

```
DCIHandleFree ( hndlp      IN      ADDR, //释放的句柄
                type       IN      UINT4 //释放句柄的类型
                )
                RETURNS INT2
```

功能描述: 释放已分配的各类句柄。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR、DCI\_INVALID\_HANDLE。

9.2.15 获取参数描述符信息 **DCIParamGet**

接口原型:

```
DCIParamGet ( hndlp      IN      ADDR, //参数描述符句柄
              htype      IN      UINT4, //参数描述符句柄的类型
              errhp      INOUT   ADDR, //错误信息句柄
              parmdpp    OUT     ADDR, //参数描述符的地址
              pos        IN      UINT4 //参数描述符在句柄中的位置
              )
              RETURNS INT2
```

功能描述: 获取参数描述符信息。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.16 为句柄设置描述符信息 **DCIParamSet**

接口原型:

```
DCIParamSet ( hndlp      INOUT   ADDR, //描述符句柄
              htype      IN      UINT4, //描述符句柄的类型
              errhp      INOUT   ADDR, //错误信息句柄
              dscp       IN      ADDR, //描述符的地址
              dtyp       IN      UINT4, //描述符类型
              pos        IN      UINT4 //描述符在句柄中的的位置
              )
              RETURNS INT2
```

功能描述: 为句柄设置描述符信息。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.17 指定参数绑定数据的间隔大小 **DCIBindArrayOfStruct**

接口原型:

```
DCIBindArrayOfStruct ( bindp      INOUT   ADDR, //绑定信息句柄
                       errhp      INOUT   ADDR, //错误信息句柄
                       pvskip     IN      UINT4, //参数值间隔的长度
                       indskip    IN      UINT4 //空值指示符间隔大小
                       )
                       RETURNS INT2
```

功能描述: 指定参数绑定数组的间隔大小。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.18 按名称绑定参数 **DCIBindByName**

接口原型:

```
DCIBindByName ( stmtp      INOUT      ADDR,    //绑定的语句句柄
                bindp      INOUT      ADDR,    //绑定信息句柄
                errhp      INOUT      ADDR,    //错误信息句柄
                placeholder IN        ADDR,    //绑定参数名称的地址
                placeh_len IN        INT4,    //绑定参数名称长度
                valuep      INOUT      ADDR,    //参数值缓冲区的地址
                value_sz    IN        INT4,    //参数类型单个值大小
                dtty        IN        UINT2,    //参数数据类型
                indp        INOUT      ADDR,    //空指示符的地址
                alenp       INOUT      ADDR,    //数据长度指示符的地址
                rcodep      OUT        ADDR,    //返回码指示符的地址
                maxarr_len  IN        UINT4,    //数据最大长度
                curelep     INOUT      ADDR,    //实际参数个数的地址
                mode        IN        UINT4     //绑定模式
                )
                RETURNS INT2
```

功能描述: 按名称绑定 SQL 语句中的参数。  
返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.19 按位置绑定参数 **DCIBindByPos**

接口原型:

```
DCIBindByPos ( stmtp      INOUT      ADDR,    //绑定的语句句柄
                bindp      INOUT      ADDR,    //输出的绑定信息句柄
                errhp      INOUT      ADDR,    //错误信息句柄
                position    IN        UINT4,    //参数在SQL中出现的位置
                valuep      INOUT      ADDR,    //参数值缓冲区的地址
                value_sz    IN        INT4,    //参数类型单个值的大小
                dtty        IN        UINT2,    //参数数据类型
                indp        INOUT      ADDR,    //空指示符的地址
                alenp       INOUT      ADDR,    //数据长度指示符的地址
                rcodep      OUT        ADDR,    //返回码指示符的地址
                maxarr_len  IN        UINT4,    //数据最大长度
                curelep     INOUT      ADDR,    //实际参数个数的地址
                mode        IN        UINT4     //绑定模式
                )
                RETURNS INT2
```

功能描述: 按位置绑定 SQL 语句中的参数。  
返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.20 指定结果行集数组中每行数据大小 **DCIDefineArrayOfStruct**

接口原型:

```
DCIDefineArrayOfStruct ( defnp      INOUT      ADDR,    //结果集列绑定结构的地址
                        errhp      INOUT      ADDR,    //错误信息句柄
```

```
        pvskip      IN      UINT4,    //数据值间隔大小
        indskip     IN      UINT4,    //空值指示符间隔大小
        rlskip      IN      UINT4     //结果长度指示符间隔大小
    )
```

RETURNS INT2

功能描述：指定结果行集数组中每行数据的大小。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.21 按位置设置结果集的列存储空间 DCIDefineByPos

接口原型：

```
DCIDefineByPos ( stmtp      INOUT      ADDR,    //绑定的语句句柄
                 defnpp     INOUT      ADDR,    //结果集列绑定句柄
                 errhp      INOUT      ADDR,    //错误信息句柄
                 position    IN         UINT4,    //结果集中的列位置
                 valuep     INOUT      ADDR,    //结果集缓冲区的地址
                 value_sz    IN         INT4,     //参数类型单个值大小
                 dty         IN         UINT2,    //绑定的数据类型
                 indp        IN         ADDR,    //指示符缓冲区的地址
                 rlenp       INOUT      ADDR,    //返回值长度指示符的地址
                 rcoddep     OUT        ADDR,    //返回码指示符的地址
                 mode        IN         UINT4     //结果集列绑定模式
    )
```

RETURNS INT2

功能描述：按位置设置结果集中每个列的存储空间。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.22 获取数据库对象信息 DCIDescribeAny

接口原型：

```
DCIDescribeAny ( svchp      IN         ADDR,    //服务上下文句柄
                 errhp      INOUT      ADDR,    //错误信息句柄
                 objptr     IN         ADDR,    //描述对象的地址
                 objnm_len   IN         UINT4,    //描述对象的字符串长度
                 objptr_typ  IN         UCHAR,    //描述对象地址的数据类型
                 info_level  IN         UCHAR,    //信息等级
                 objtyp      IN         UCHAR,    //描述对象的类型
                 dschp       INOUT      ADDR     //数据库对象描述句柄
    )
```

RETURNS INT2

功能描述：获取数据库对象的详细信息。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.23 准备需执行的 SQL 语句 DCIStmtPrepare

接口原型：

```
DCIStmtPrepare ( stmtp      IN         ADDR,    //准备执行的语句句柄
                 errhp      INOUT      MEM,     //错误信息句柄
                 stmt        IN         ADDR,    //准备执行的SQL语句
                 stmt_len    IN         UINT4,    //SQL语句长度
    )
```

```
        language    IN      UINT4, //设置版本兼容语言
        mode        IN      UINT4   //准备模式
    )
    RETURNS INT2
```

功能描述：准备需要执行的 SQL 语句。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.24 执行 SQL 语句 DCIStmtExecute

接口原型：

```
DCIStmtExecute ( svchp      INOUT      ADDR, //服务上下文句柄
                 stmtp      INOUT      ADDR, //执行语句的句柄
                 errhp      INOUT      ADDR, //错误信息句柄
                 iters       IN         UINT4, //执行影响的行数
                 rowoff      IN         UINT4, //结果集的偏移量
                 snap_in     IN         ADDR, //输入快照描述符的地址
                 snap_out    OUT        ADDR, //输出快照描述符的地址
                 mode        IN         UINT4  //执行模式
    )
    RETURNS INT2
```

功能描述：执行准备好的 SQL 语句。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR、DCI\_SUCCESS\_ WITH\_INFO、DCI\_NO\_DATA。

9.2.25 获取 SQL 语句的执行结果 DCIStmtFetch

接口原型：

```
DCIStmtFetch ( stmtp      IN      ADDR, //结果集的语句句柄
               errhp      IN      ADDR, //错误信息句柄
               nrows      IN      UINT4, //获取的数据行数
               orientation IN      UINT2, //结果集提取方式
               mode        IN      UINT4  //提取模式
    )
    RETURNS INT2
```

功能描述：获取 SQL 语句执行后的结果。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR、DCI\_SUCCESS\_ WITH\_INFO、DCI\_NO\_DATA。

9.2.26 启动事务 DCITransStart

接口原型：

```
DCITransStart ( svchp      INOUT      ADDR, //服务上下文句柄
                errhp      INOUT      ADDR, //错误信息句柄
                timeout     IN         UINT4, //超时等待时间
                flags       IN         UINT4  //事务类型
    )
    RETURNS INT2
```

功能描述：启动一个事务。

返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.27 提交事务 DCITransCommit

接口原型：

```
DCITransCommit ( svchp      IN      ADDR, //服务上下文句柄
```

```
errhp      INOUT  ADDR,  //错误信息句柄
flags      IN     UINT4   //提交方式
)
RETURNS INT2
```

功能描述：提交一个事务。  
返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.28 回退事务 DCITransRollback

接口原型：

```
DCITransRollback ( svchp      IN          ADDR,  //服务上下文句柄
errhp      INOUT  ADDR,  //错误信息句柄
flags      IN     UINT4   //回滚方式
)
RETURNS INT2
```

功能描述：回退一个事务。  
返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.29 获取操作失败时的错误信息 DCIErrorGet

接口原型：

```
DCIErrorGet ( hndlp      IN          ADDR,  //错误信息句柄
recordno    IN          UINT4,  //错误信息位置
sqlstate    OUT         UCHAR,  //错误状态
errcodep    OUT         INT4,   //错误返回值
bufp        OUT         ADDR,   //错误描述信息缓冲区的地址
buf_sz      IN          UINT4,  //错误描述信息缓冲区的大小
type        IN          UINT4   //错误信息类型
)
RETURNS INT2
```

功能描述：获取 DCI 接口操作失败时的错误信息。  
返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.30 获取数据库实例版本 DCIServerVersion

函数原型：

```
DCIServerVersion ( hndlp      IN          ADDR,  //上下文句柄
errhp        INOUT  ADDR,  //错误信息句柄
bufp         INOUT  ADDR,  //版本信息缓冲区的地址
bufsz        IN          UINT4,  //版本信息缓冲区的大小
hndltype     IN          UCHAR  //上下文句柄类型
)
RETURNS INT2
```

功能描述：获取数据库实例的版本信息。  
返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.31 获取数据库当前连接数 DCIServerSession

函数原型：

```
DCIServerSession ( hndlp      IN          ADDR,  //上下文句柄
errhp          INOUT  ADDR,  //错误信息句柄
bufp           INOUT  ADDR,  //连接数缓冲区的地址
bufsz          IN          UINT4,  //连接数缓冲区的大小
hndltype       IN          UCHAR  //上下文句柄类型
)
RETURNS INT2
```

功能描述：获取数据库实例当前的连接数。  
返回值：执行成功返回 DCI\_SUCCESS，执行错误时可能的返回值包括 DCI\_ERROR。

9.2.32 导入 CIM/E 数据文件 **DCICIMEDDataIn**

函数原型:

```
DCICIMEDDataIn ( svchp      INOUT      ADDR,  //服务上下文句柄
                  stmp      INOUT      ADDR,  //执行的语句句柄
                  schmeaname IN        ADDR,  //导入数据库模式名的地址
                  scnamelen  IN        UINT4,  //导入数据库模式名长度
                  path       IN        ADDR,  //导入文件路径（含名称）的地址
                  pathlen    IN        UINT4,  //导入文件路径长度
                  errhp      INOUT      ADDR   //错误信息句柄
                )
                RETURNS INT2
```

功能描述: 将一个 CIM/E 标准的数据文件导入到数据库中。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.33 导出 CIM/E 数据文件 **DCICIMEDDataOut**

函数原型:

```
DCICIMEDDataOut ( svchp      INOUT      ADDR,  //服务上下文句柄
                  stmp      INOUT      ADDR,  //执行的语句句柄
                  schmeaname IN        ADDR,  //导出数据库模式名的地址
                  scnamelen  IN        UINT4,  //导出数据库模式名长度
                  tabelname  IN        ADDR,  //导出表名的地址
                  tnamelen   IN        UINT4,  //导出表名长度
                  path       IN        ADDR,  //导出文件路径（含名称）的地址
                  pathlen    IN        UINT4,  //导出文件路径的长度
                  errhp      INOUT      ADDR   //错误信息句柄
                )
                RETURNS INT2
```

功能描述: 将表中的数据导出成 CIM/E 标准的数据文件。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。

9.2.34 根据 CIM/E 模板文件建表 **DCICIMECreateTable**

函数原型:

```
DCICIMECreateTable ( svchp      INOUT      ADDR,  //服务上下文句柄
                    stmp      INOUT      ADDR,  //执行的语句句柄
                    schmeaname IN        ADDR,  //导入数据库模式名的地址
                    scnamelen  IN        UINT4,  //导入数据库模式名的长度
                    path       IN        ADDR,  //导入文件路径（含名称）的地址
                    pathlen    IN        UINT4,  //导入文件路径长度
                    errhp      INOUT      ADDR   //错误信息句柄
                  )
                  RETURNS INT2
```

功能描述: 根据 CIM/E 标准的模板文件在数据库中建表。

返回值: 执行成功返回 DCI\_SUCCESS, 执行错误时可能的返回值包括 DCI\_ERROR。



附 录 A  
(规范性附录)  
实时数据库访问接口数据结构

A.1 表描述符

表描述符 RTDB\_TB\_DESCR 用于标识实时数据库中的数据表，数据结构声明如下：

```
{  
    STRING      context_name;    //字符串形式表示的态标识  
    UINT4       context_id;      //数值形式表示的态标识  
    STRING      app_name;        //字符串形式表示的应用标识  
    UINT4       app_id;          //数值形式表示的应用标识  
    STRING      table_name;      //字符串形式表示的表标识  
    UINT4       table_id;        //数值形式表示的表标识  
} RTDB_TB_DESCR
```

A.2 表信息结构

表信息结构 RTDB\_TABLE\_PARA\_STRU 定义了实时数据库数据表的各类属性，数据结构声明如下：

```
{  
    STRING      table_name_eng;  //表英文名  
    STRING      table_name_chn;  //表中文名  
    UINT4       key_len;         //关键字长度  
    UINT4       subkey_num;      //关键字个数  
    UINT4       field_sum;       //表域个数  
    UINT4       record_size;     //记录长度  
    UINT4       record_number;   //记录个数  
} RTDB_TABLE_PARA_STRU
```

附 录 B  
(规范性附录)  
时间序列实时数据库访问接口数据结构

B.1 连接描述符

连接描述符 WRDB\_DESCR 用于描述时间序列实时数据库的连接信息和连接状态，数据结构声明如下：

```
{
    UINT4      link_mode;    //连接模式，包括网络连接或文件方式连接
    STRING     db_addr;      //连接位置，描述数据库连接的物理位置
} WRDB_DESCR
```

B.2 测点结构

测点结构 WRDB\_POINTCONFIG 描述了时间序列实时数据库中数据测点的信息，数据结构声明如下：

```
{
    INT4      meas_id;       //测点标识
    STRING    pnt_name;      //测点名
    INT4      frames;        //数据帧数
} WRDB_POINTCONFIG
```

附 录 C  
(规范性附录)  
时间序列历史数据库访问接口数据结构

C.1 库描述符

库描述符 WHDB\_DESCR 用于标识所操作的时间序列历史数据库，数据结构声明如下：

```
{
    UINT4      mode;    //连接模式
    STRING     addr;    //连接位置
} WHDB_DESCR
```

C.2 测点配置数据结构

测点配置数据结构 WHDB\_MEASUREMENT\_CONFIG 描述了时间序列历史数据库的测点配置信息，数据结构声明如下：

```
{
    UINT4      id;      // 数据 ID
    UINT4      type;    // 数据类型
    STRING     name;    // 数据名称
} WHDB_MEASUREMENT_CONFIG
```

C.3 遥测量数据结构

遥测量数据结构 WHDB\_VALUE 描述了时间序列历史数据库的单个遥测量数据值的结构信息，数据结构声明如下：

```
{
    UINT4      id;      //数据 ID
    INT4       secs;    //秒级时间
    INT4       usecs;   //微秒级时间
    FLOAT      value;   //数据值
    UCHAR      val_tag;  //数据质量
} WHDB_VALUE
```

C.4 遥信量数据结构

遥信量数据结构 WHDB\_DIG\_VALUE 描述了时间序列历史数据库的单个遥信量数据值的结构信息，数据结构声明如下：

```
{
    UINT4      id;      //数据 ID
    INT4       secs;    //秒级时间
    INT4       usecs;   //微秒级时间
    UCHAR      value;   //数据值
    UCHAR      val_tag;  //数据质量
} WHDB_DIG_VALUE
```

附 录 D  
(规范性附录)

关系数据库访问接口（DCI）中使用的句柄及描述符

D.1 句柄

句柄是指向某一特定存储区域、对外部透明的内存地址。这些区域用来保存上下文信息、连接信息、错误信息或绑定信息等。DCI 接口中至少应提供表 D.1 所列的句柄。

表 D.1 DCI 句 柄 类 型

句柄名称	描 述
DCI_HTYPE_ENV	DCI 环境句柄
DCI_HTYPE_ERROR	DCI 错误描述句柄
DCI_HTYPE_SVCCTX	DCI 服务上下文句柄
DCI_HTYPE_STMT	DCI 语句句柄
DCI_HTYPE_BIND	DCI 参数绑定句柄
DCI_HTYPE_DEFINE	DCI 结果集列绑定句柄
DCI_HTYPE_DESCRIBE	DCI 数据库对象描述句柄
DCI_HTYPE_SERVER	DCI 数据库服务句柄
DCI_HTYPE_SESSION	DCI 登录用户信息连接句柄

D.2 描述符

描述符是一种用于定义特定数据类型的、对外部透明的数据结构，用来维护特定的数据信息。DCI 接口中至少应提供表 D.2 所列的描述符。

表 D.2 DCI 描 述 符 类 型

描述符名称	描 述
DCI_DTYPE_PARAM	DCI 参数类型描述符
DCI_DTYPE_TIMESTAMP	DCI TIMESTAMP 类型描述符
DCI_DTYPE_INTERVAL_YM	DCI 日期间隔类型描述符
DCI_DTYPE_INTERVAL_DS	DCI 日期时间间隔类型描述符

中 华 人 民 共 和 国  
电 力 行 业 标 准  
电力系统数据库通用访问接口规范  
DL/T 1456 — 2015

\*

中国电力出版社出版、发行  
(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)  
北京九天众诚印刷有限公司印刷

\*

2016 年 2 月第一版 2016 年 2 月北京第一次印刷  
880 毫米×1230 毫米 16 开本 2.25 印张 66 千字  
印数 001—100 册

\*

统一书号 155123·2767 定价 19.00 元

敬 告 读 者

本书封底贴有防伪标签，刮开涂层可查询真伪  
本书如有印装质量问题，我社发行部负责退换

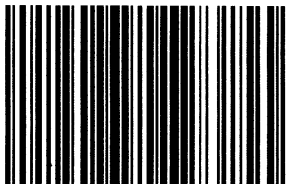
版 权 专 有 翻 印 必 究



中国电力出版社官方微信



掌上电力书屋



155123.2767