母线智能监测系统数据库设计

作者：邢永陈 时间：2019年6月7日 版本：v1

**引 言**

母线槽在使用过程中，部分用户往往忽视母线槽的安全管理，由此出现诸如绝缘加速老化、母线安全缺乏有效监控措施等问题，威胁到供电安全，甚至引发电气火灾事故，母线槽安装规模庞大，排查隐患点如大海捞针，运维人员难为对母线槽进行常态化精细化的停电维护检查。由于缺乏长期深入的检测维护，无法评估母线槽状态，由此造成的后果是电气火灾往往突发成灾，且难以控制。根据电力安全监督部门提出数据分析，全国每年因为母线槽故障引发的重大事故上前起，给生产和运营造成巨大经济损失。目前市场上的母线监控系统多是流于形式，未能真正的使母线的管理实现精细化、自动化、智能化，因此急需一套切实有效的系统来对母线安全进行准确的检测管理，提高事故预警能力。

“母线温度在线监测系统”负责对母线槽运行情况及所处环境的全面监控和信息记录，对母线工作时的安全性能、运行效率、负荷状况进行评估，预测可能存在的风险，让工作人员能及时有效的掌握整个系统的运行状况。

系统主要通过传感器采集若干个母线相关的参数信息（包括温度、湿度、电流、电压、功率因数、谐波等）来对现场母线的运行情况做准确监控，并实时收集运行数据及设备状态，将信息数据通过网络传送到服务器，由服务器端的监控软件产生报表及状态汇总，以作为操作人员直观的监控参考，读取相关数据并加以分析，还可进一步进行远程控制或设备维护，可以减少人力资源、缩短修护时间并节省成本。

**第一部分 需求分析**

项目的总体目标是建立一套对数据中心母线内的环境和电气参量进行统一监测和预警管理的智能监控管理系统。

按照标准实际的母线系统进行监控设计，主要通过对环境要求监测对传感器部署，对母线系统的温、湿度等环境指标建立一套较完整的信息监控、环境监控、安全防范的管理体系，实现对母线系统运行环境科学管理，及时发现母线设备的故障和故障隐患，做到及时发现、及时处理。

母线智能控制探测系统对母线运行过程能进行全天候无间断的数据监测，并能将监测的数据进行传输和保存，还能对母线系统运行过程中的突发事件进行预警和报告，从而能达到对系统的不间断的值守。

母线温度智能监测系统将实现以下目标：

* 为母线内各系统及设备运行提供高度稳定可靠的监控信息资源；
* 节省母线运行管理费用，达到短期投资长期受益的目的；
* 确保提高母线系统管理工作效率并提供安全适用的工作环境；
* 系统软/硬件均采用模块化结构设计，适应发展需要，做到具有可扩展性、可变性，适应环境的变化和工作性质的多样化；
* 用户通过浏览器、安卓、IOS各端都能实时查询、监测母线温度数据。

**数据模型：**



**数据库对象**：

（1）项目：项目编号，项目名称，项目位置，项目图片路径，项目归属单位，项目说明。

（2）楼层：编号，位置，楼层图片路径。

（3）数据发送器：编号，名称，MAC，位置。

（4）数据点：编号，名称，位置，数据单位，报警上界，报警下界。

（5）用户：编号，用户名，密码，电话，邮箱，身份。

（6）操作记录：编号，内容记录。

（7）报表：编号，内容。

**数据流图：**



**第二部分 概念设计**

本数据库系统一共有7个实体：

分别为：项目、楼层、数据发送器、数据点、用户、操作记录、报表。

经过分析研究，可以得到以下的E-R图：



第三部分 模式设计

根据上述E-R图，可以发现：项目与楼层是1对n关系，楼层与数据发送器是1对n关系，数据发送器与数据点之间是1对n关系，项目与报表之间是1对n关系，用户与操作记录之间是1对n关系；项目与用户之间是m对n关系。

通过以上分析，建立如下关系模式：

项目（编号，项目名称，项目位置，项目图片路径，项目归属单位，项目说明）。

楼层（编号，位置，楼层图片路径，项目编号）。

数据发送器（编号，名称，MAC，位置，楼层编号）。

数据点（编号，名称，位置，数据单位，报警上界，报警下界，数据发送器编号）。

用户（编号，用户名，密码，电话，邮箱，身份）。

操作记录（编号，内容记录，用户名）。

报表（编号，内容，项目编号）。

项目用户关系表（编号，项目编号，用户名）。

第四部分 定 义

由以上的工作，得到各模式，则可进行SQL定义，现将定义语句罗列如下：

**1、项目**

CREATE TABLE project

(

id int NOT NULL AUTO\_INCREMENT,

name varchar(255),

palce varchar(255),

image\_path varchar(255),

belong varchar(255),

introduction varchar(1000),

PRIMARY KEY (id)

)

**2、楼层**

CREATE TABLE floor

(

id int NOT NULL AUTO\_INCREMENT,

palce varchar(255),

image\_path varchar(255),

project\_id int NOT NULL,

PRIMARY KEY (id)

)

**3、数据发送器**

CREATE TABLE sender

(

id int NOT NULL AUTO\_INCREMENT,

name varchar(255),

mac varchar(255),

palce varchar(255),

floor\_id int NOT NULL,

PRIMARY KEY (id)

)

**4、数据点**

CREATE TABLE data\_point

(

id int NOT NULL AUTO\_INCREMENT,

name varchar(255),

palce varchar(255),

unit varchar(255),

upper\_bound float,

lower\_bound float,

sender\_id int NOT NULL,

PRIMARY KEY (id)

)

**5、用户**

CREATE TABLE user

(

id int NOT NULL AUTO\_INCREMENT,

name varchar(255) NOT NULL,

password varchar(255) NOT NULL,

tel varchar(255),

email varchar(255),

capacity varchar(255),

PRIMARY KEY (name)

)

**6、操作记录**

CREATE TABLE operation\_log

(

id int NOT NULL AUTO\_INCREMENT,

content varchar(500),

user\_name varchar(255),

PRIMARY KEY (id)

)

**7、报表**

CREATE TABLE report\_form

(

id int NOT NULL AUTO\_INCREMENT,

content varchar(500),

project\_id int NOT NULL,

PRIMARY KEY (id)

)

**8、项目用户关系表**

CREATE TABLE project\_user\_relation

(

id int NOT NULL AUTO\_INCREMENT,

project\_id int,

user\_name varchar(255),

PRIMARY KEY (id)

)

第五部分 小 结

在关系数据库设计中，我们首先要明确设计的最终目标，再根据目标决定哪些数据要持久化存储; 对于这些数据，要按照功能和逻辑来进行拆分，并且存放在不同的表中，并且明确之间的关系; 对于设计好的表，要进行重构，根据设计范式对大表进行拆分和优化; 对于每个表要增加对应的完整性检查，关键是实体完整性和参照完整性；最后在实际使用中，对于高频查询的记录构建索引提升效率，以及其他因地制宜的优化。