DETA Database PLSQL

Mr. Yaoguang. Luo

Liu Yang Deta Software Development Limited Company, Hunan, China,

313699483@qq.com

Outline: *this document paper makes a pretty explanation of how does DETA database works by using PLSQL Method. At the same time, I will spend more care about the DETA PLSQL runs in the command line or rest call service, with a lot of real world samples.*

DETA PLSQL Commands

- **setRoot**:[path];
- **baseName**:[baseName];
- **tableName**:[tableName]:[operation];
- **getCulumns**:[difinition1]:[difinition2]:[difinition3]:[difinition4]:[difinition5]:.....;
- **culumnName**:[culumnName]:[dataType];
- **changeCulumnName**:[newCulumnName]:[oldCulumnName];
- **culumnValue**:[culumnName]:[culumnValue];
- **condition**:[operation]:[difinition1]:[difinition2]:[difinition3]:...;
- **join**:[baseName]:[tableName];
- **relation**[operation]:[difinition1]:[difinition2]:[difinition3]:...;
- **aggregate**[operation]:[difinition1]:[difinition2]:[difinition3]:...;

Commands Definition

- **setRoot:[path];**

The setRoot:[path]; is mostly used for set the database path.

- **baseName:[baseName];**

The baseName:[baseName]; is mostly used for set the current database name in the PLSQL language compiler system.

- **tableName:[tableName]:[operation];**

The tableName:[tableName]'[operation]; is mostly used for set the current table name in current database with the operations. For example tableName:tableName:select; this command will tell PLSQL system, now begin to do the select function section.

- **getCulumns:[difinition1]:[difinition2]:[difinition3]:[difinition4]:[difinition5]:.....;**

The getCulumns:[difinition1]:[difinition2]:[difinition3]:[difinition4]:[difinition5]:.....; mostly be used for select columns.

- **culumnName:[culumnName]:[dataType];**

The culumnName:[culumnName]:[dataType]; mostly be used for create the table columns.

- **changeCulumnName:[oldCulumnName]:[newCulumnName];**

The changeCulumnName:[newCulumnName]:[oldCulumnName]; mostly be used for change the table columns.

- **culumnValue:[culumnName]:[culumnValue];**

The culumnValue:[culumnName]:[culumnValue]; mostly be used for update the columns value.

- **condition:[operation]:[difinition1]:[difinition2]:[difinition3]:...;**

The condition:[operation]:[difinition1]:[difinition2]:[difinition3]:...; mostly be used for

- **join:[baseName]:[tableName];**

The join:[baseName]:[tableName]; mostly be used for select and update of delete with conditions.

- **relation[operation]:[difinition1]:[difinition2]:[difinition3]:...;**

The relation[operation]:[difinition1]:[difinition2]:[difinition3]:...; mostly be used for join section condition

- **aggregate[operation]:[difinition1]:[difinition2]:[difinition3]:...;**

The aggregate[operation]:[difinition1]:[difinition2]:[difinition3]:...; mostly be used for limit, sort or addition operations.

Command Samples

select samples

tableName:test:select;

condition:or:testCulumn1|<|20:testCulumn2|==|fire;

condition:and:testCulumn1|>|100:testCulumn2|==|fire;

select where in samples

setRoot:C:/DetaDB;

baseName:backend;

tableName:usr:select;

condition:or:u_id|in|3,4,5;

select join samples

tableName:utest:select;

condition:or:testCulumn1|<|20:testCulumn2|==|fire;

condition:and:testCulumn1|>|100:testCulumn2|==|fire;

join:stest;

relation:or:uid|==|sid:ussd|==|sssd;

relation:and:utoken|=!|stoken:umap|==|smap;

select join samples

tableName:utest:select;

condition:or:utestCulumn1|<|20:utestCulumn2|==|fire;

condition:and:utestCulumn1|>|100:utestCulumn2|==|fire;

getCulumns:utestCulumn1|as|uid::utestCulumn2|as|ussd:utoken:umap;

join:backend:stest;

condition:and:stestCulumn1|>|100:stestCulumn2|==|fire;

getCulumns:stestCulumn1|as|sid|:stestCulumn2|as|sssd:stoken:smap;

relation:or:uid|==|sid:ussd|==|sssd;

relation:and:utoken|=!|stoken:umap|==|smap;

aggregation:limit:2|~|10;

insert samples

tableName:test:insert;

culumnValue:date0:19850525;

culumnValue:date1:19850526;

culumnValue:date2:19850527;

culumnValue:date3:19850528;

culumnValue:date4:19850529;

update samples

tableName:test:update;

condition:or:testCulumn1|<|20:testCulumn2|==|fire;

condition:and:testCulumn1|>|100:testCulumn2|==|fire;

culumnValue:date0:19850525;

culumnValue:date1:19850526;

update samples

tableName:test:update;

condition:or:testCulumn1|<|20:testCulumn2|==|fire;

condition:and:testCulumn1|>|100:testCulumn2|==|fire;

join:backend:utest;

condition:and:uCulumn3|<|20;

relation:and:testCulumn1|==|uCulumn1:testCulumn2|!=|uCulumn2;

culumnValue:date0:19850525;

culumnValue:date1:19850526;

delete samples

tableName:test:delete;

condition:or:testCulumn1|<|20:testCulumn2|==|fire;

condition:and:testCulumn1|>|100:testCulumn2|==|fire;

create samples

tableName:test:create;

culumnName:pk:culumn1:string;

culumnName:uk:culumn1:long;

culumnName:uk:culumn1:obj;

culumnName:nk:culumn1:double;

drop samples

tableName:test:drop;

change samples

tableName:test:change;

changeCulumnName:oldCulumnName:newCulumnName;

Real World Samples By Using DETA PLSQL Database

setRoot:C:/DetaDB;

baseName:backend;

tableName:usr:select;

condition:or:u_id|<=|3:u_id|>|7;

condition:and:u_email|!equal|321:u_name|!equal|123;

getCulumns:u_id|as|detaId:u_email|as|detaEmail;

join:backend:usrToken;

condition:and:u_level|equal|low;

getCulumns:u_id|as|sId:u_level:u_password|as|SSID;

relation:and:detaId|==|sId;

aggregation:limit:0|~|1;

compare Tranditioanl SQL:

```
SELECT u.u_id as detaId，u.u_email as detaEmail, t.u_id as sId，t.u_level，t.u_password as SSID
FROM usr as U
INNER JOIN (SELECT t.u_id as sId，t.u_level，t.u_password as SSID
            FROM usrToken as t
            WHERE t.u_level equal "low") AS B on U.detaId == B.sId;
WHERE (u.u_id <=3 || u.u_id>7 ) && (u.u_email !equal '321' && u.u_name !equal 123);
LIMIT 0,1;
```


Acknowledgement

The DETA PLSQL database system source code link:
https://github.com/yaoguangluo/DETA_DataBase