

Automation in shell

Run SQL

```
counter=`mysql -u $DB_USER -p $DB_PASS $DB_NAME -e "select count(*) from tablename where id=XX;"`

# -B: Suppress table borders
counter=`mysql -u $DB_USER -p $DB_PASS $DB_NAME -B -e "select count(*) from tablename where id=XX;"`

# Multiple SQLs
mysql -u USER -pPASSWORD <<EOF
SQL_QUERY 1
SQL_QUERY 2
SQL_QUERY N
EOF
```

For cron job (my.cnf)

```
mysql --defaults-extra-file=/path/to/specific/.my.cnf -e 'SELECT something FROM sometable'
```

`~/my.cnf` :

```
# This will be passed to all MariaDB clients
[client]
user=root
password="p@$@"
```

Parsing SQL Output

```
local sql="
SELECT
    DATE(t1.starttime) AS day,
    sum(t1.sessiontime) AS calltime,
    sum(t1.sessionbill) AS cost,
```

```

count(*) as nbcall,
sum(t1.buycost) AS buy,
sum(case when t1.sessiontime>0 then 1 else 0 end) as success_calls
FROM
cc_call t1
LEFT OUTER JOIN cc_trunk t3 ON t1.id_trunk = t3.id_trunk
LEFT OUTER JOIN cc_ratecard t4 ON t1.id_ratecard = t4.id
WHERE
t1.starttime >= ('$from_date $from_time') AND
t1.starttime <= ('$to_date $to_time') AND
t1.terminatecauseid = 1
GROUP BY day
ORDER BY day
LIMIT 50;
"

```

```

# Execute SQL query and store results in temporary file
local tmp=$(mysql -u $db_user $db_name -s -e "$sql" | grep -v "day")

```

```

while IFS= read -r line; do
    local c1 c2 c3 c4 c5 c6
    read -r c1 c2 c3 c4 c5 c6 <<< "$line"
    local day="$c1"
    local duration="$c2"
    local ncalls="$c4"
    local scalls="$c6"

    if [ -n "$scalls" ] && [ -n "$ncalls" ]; then
        local asr=$(bc -l <<< "scale=2; $scalls * 100 / $ncalls")
    fi

    date_list+="$day"
    duration_list+="$duration"
    asr_list+="$asr"
    calls_list+="$ncalls"
done <<< "$tmp"

```

CRUD Functions

```
#!/bin/bash

# MySQL login details
MYSQL_USER="user"
MYSQL_PASSWORD="password"

# Function to execute a MySQL query and print the results
execute_query() {
    local query=$1
    local database=$2
    mysql -u "$MYSQL_USER" -p"$MYSQL_PASSWORD" "$database" -e "$query"
}

# Function to create a new database
# Usage: create_database {db-name}
create_database() {
    local database=$1
    execute_query "CREATE DATABASE $database"
}

# Function to create a new table in a database
# Usage: create_table {db-name} {table-name} {schema}
# {schema}: "id INTEGER PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255), salary INTEGER"
create_table() {
    local database=$1
    local table=$2
    local schema=$3
    execute_query "CREATE TABLE $table ($schema)" "$database"
}

# Function to insert data into a table
# Usage: insert_data {db-name} {table-name} ${data}
# ${data}: "num1, 'str1', num2, 'str2'"
insert_data() {
    local database=$1
    local table=$2
    local data=$3
    execute_query "INSERT INTO $table VALUES ($data)" "$database"
}

```

```

# Function to update data in a table
# Usage: update_data {db-name} {table-name} {data} ${condition}
# {data}: "salary=7000"
# ${condition}: "id=2"
update_data() {
    local database=$1
    local table=$2
    local data=$3
    local condition=$4
    execute_query "UPDATE $table SET $data WHERE $condition" "$database"
}

# Function to delete data from a table
# Usage: delete_data ${db-name} ${table-name} ${condition}
# ${condition}: "id=2"
delete_data() {
    local database=$1
    local table=$2
    local condition=$3
    execute_query "DELETE FROM $table WHERE $condition" "$database"
}

# Function to display data from a table
# Usage: display_data ${db-name} ${table-name}
display_data() {
    local database=$1
    local table=$2
    execute_query "SELECT * FROM $table" "$database"
}

```

Revision #10

Created 2025-07-18 15:25:43 CST by A-Lang (Admin)

Updated 2025-07-25 16:27:47 CST by A-Lang (Admin)