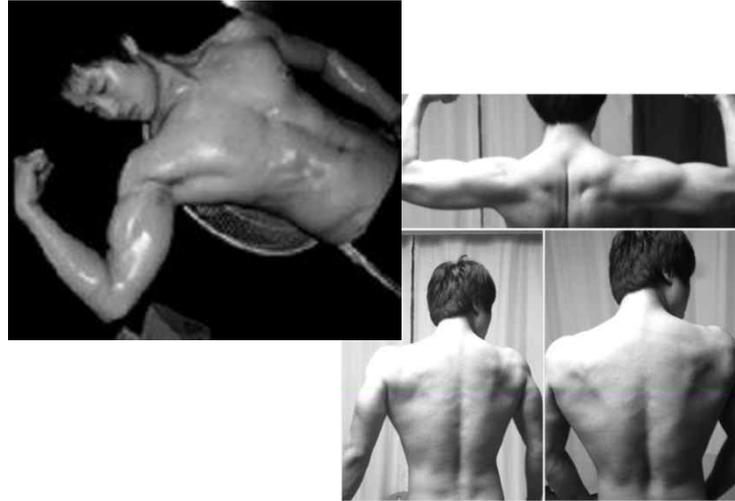




for MySQL

정 주원

정 주원



NIPA Global Open Frontier Lab

Korea Sencha User Group

<https://www.facebook.com/jungjuwon>

jjwcom@nate.com

https://github.com/jeongjuwon/planche

jeongjuwon / **planche**

MySQL GUI client tool based on ExtJS

78 commits 1 branch 0 releases 1 contributor

branch: master **planche** / +

History, Info 탭 스크롤링 수정

jeongjuwon authored 3 days ago latest commit 5067cebbee

codemirror	코드미러 버전 업데이트	a month ago
controller	테이블 생성 오류 수정	3 days ago
engine	alter method 추가	6 days ago
extjs	첫번째 미완성본 전체 소스 커	a month ago
images	sql icon 추가	9 days ago
lib	LIMIT 파싱 오류 수정	3 days ago
LICENSE	Initial commit	2 months ago
README.md	no message	9 days ago
app.js	History, Info 탭 스크롤링 수정	3 days ago
config-host.js	Example Host 설정 추가	a month ago



Planche for MySQL

Alter Table 'tbl_blocksql_list' in 'WhitesQL'

Field Name	Datatype	Length	Default	PK	Not Null	Unsigned	Auto Incr	Zerofill	Comment
agent_id	int	10			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			서버내부니(=tbl_agent_info.u...
policy_name	varchar	255			<input checked="" type="checkbox"/>				정책이름
policy_type	enum	'1','2','3'			<input checked="" type="checkbox"/>				정책 종류(1:BLOCK_SQL, 2:US...
policy_id	int	10			<input checked="" type="checkbox"/>				정책이디
apply_starttime	int	10			<input checked="" type="checkbox"/>				정책적용시작시간
apply_endtime	int	10			<input checked="" type="checkbox"/>				정책적용종료시간
event_level	tinyint	4			<input checked="" type="checkbox"/>				이벤트 레벨 (이력사용하지 않...
description	varchar	255			<input checked="" type="checkbox"/>				설명
reg_time	int	10			<input checked="" type="checkbox"/>				정책 등록시간
reg_user_id	int	10			<input checked="" type="checkbox"/>				
state	char	1	A		<input checked="" type="checkbox"/>				
on_off	enum	'0','1'	0		<input checked="" type="checkbox"/>				
sync_flag	enum	'0','1'	0		<input checked="" type="checkbox"/>				
sync_succ_flag	enum	'0','1'	0		<input checked="" type="checkbox"/>				
sync_time	int	10			<input checked="" type="checkbox"/>				
sync_user_id	int	10			<input checked="" type="checkbox"/>				
last_work_time	int	10	0		<input checked="" type="checkbox"/>				
last_work_user_id	varchar	20			<input checked="" type="checkbox"/>				

Messages

policy_id	policy	type	target
11	tbl_blocksql_list	4	tbl_blocksql_list
2	tbl_blocksql_list	4	tbl_blocksql_list
3	정책이름	5	tbl_blocksql_list
4	정책이디	1	tbl_blocksql_list
5	정책이디	4	tbl_blocksql_list
6	타입	1	tbl_blocksql_list
7	SQL 변경	2	tbl_blocksql_list
8	IP	3	tbl_blocksql_list
12	개인정보 Table	7	tbl_blocksql_list

Result

properties	agent_id	alarm_level	block	reg_time	reg_user_id	state	on_off
MYISAM	0	1397792249	1	1397792249	1	N	1
MRG_MyISAM	0	1395296817	1	1395296817	1	A	1
CSV	1	1395297405	1	1395297405	1	A	0
BLACKHOLE	0	1395289995	1	1395289995	1	A	0
MEMORY	1	1395299032	1	1395299032	1	A	0
FEDERATED	0	1395301352	1	1395301352	1	A	0
ARCHIVE	0	1395301382	1	1395301382	1	A	0
INNODB	0	1395303527	1	1395303527	1	A	0
PERFORMANCE_SCHEMA	0	1398073470	1	1398073470	1	N	1

웹기반의 MySQL GUI Client Tool

FRAMEWORK



Sencha Ext JS

Windows

Window Variations
A collection of Windows in different configurations, showing headers attached to any side of the window.

Layout Window
A window containing a basic BorderLayout with nested TabPanel.

MessageBox
Different styles include confirm, alert, prompt, progress and wait and also...

Trees

Drag and Drop Reordering
A TreePanel loaded asynchronously via a JSON TreeLoader that shows drag and drop with container scroll.

Locking TreeGrid (New)
TreeGrid with lockable columns

Custom Drop Lock
Apply custom determine when can be moved

Grids

Author	Title
Sidney Sheldon	Master of the Deceit
Sidney Sheldon	Are You Ahead of Tomorrow?
Sidney Sheldon	Tell Me Your Dream
Sidney Sheldon	Deadline
Sidney Sheldon	The Other Side of the Mountain

Basic Array Grid
A basic read-only grid loaded from local array data that demonstrates the use of custom column renderer functions.

Author	Title
Sidney Sheldon	Master of the Deceit
Sidney Sheldon	Are You Ahead of Tomorrow?
Sidney Sheldon	Tell Me Your Dream
Sidney Sheldon	Deadline
Sidney Sheldon	The Other Side of the Mountain

XML Grid
A simple read-only grid loaded from XML data.

Company
3m Co
Alcoa Inc
Altria Group Inc
American Express Company
American International Group

Paging
A grid with paging, cross-domain data loading and custom-rendered expandable row bodies.

Grid Plugins
Multiple grids customized via plugins: expander rows, checkbox selection and row numbering.

Grid Data Binding (advanced)
Refactoring the basic data binding example to use a class-based application design model.

Grouped Header Grid
A basic grouping grid showing collapsible data groups that can be customized via the 'Group By' header menu option.

Combination Examples

Web Desktop
Demonstrates how one could build a desktop in the browser using Ext components including a module plugin system.

Portal Demo
A page layout using several custom extensions to provide a web portal interface.

Ext JS Calendar
Example Calendar application. Demonstrates the new Day, Week and Month views and how to combine them.

Ext JS 3 & 4 on one page
This example demonstrates Ext JS 4's sandboxing behavior which allows you to run Ext JS 3 & 4 on the same page.

Image Viewer
DataView and TreePanel example that demonstrates dragging data items from a DataView into a TreePanel.

Theme Viewer
View and test every Ext component against bundled Ext themes.

Simple Tasks
Complete personal task management application example

<http://www.sencha.com/products/extjs/>



Why Ext JS?

(완벽한)컴포넌트 기반

마크업(X)

데스크탑 APP(같은..)

<http://youtu.be/QmQzCi6MqU4>

COMBINATION EXAMPLES

**Kitchen Sink**

Showcase of Ext JS components and features

**Sencha Charts Kitchen Sink**

Showcase of Sencha Charts

**Executive Dashboard**

A Tablet-Friendly Responsive Dashboard Application

**Feed Viewer**

RSS feed reader example application

**Ext JS Calendar**

Example Calendar application

**Web Desktop**

Demonstrates how one could build a desktop in the browser using Ext JS

**Ticket App**

Simple ticket application that demonstrates MVC concepts and data binding

**Portal Demo**

A page layout using several custom extensions to provide a web portal interface.

**Simple Tasks**

Complete personal task management application example

**Responsive Design Demo**

A simple application that shows how to implement a responsive UI.

**Image Viewer**

An image viewer with drag/drop functionality

**Theme Viewer**

View and test every Ext component against bundled Ext themes.

**Right-to-Left (RTL)**

Demonstrates Ext JS support for right-to-left languages

**Ext JS Legacy Charts Kitchen Sink**

Showcase of Ext JS Legacy Charts

```
1 // The bindings defined specifically in the Sublime Text mode
2 var bindings = {
3   "Alt-Left": "goSubwordLeft",
4   "Alt-Right": "goSubwordRight",
5   "Ctrl-Up": "scrollLineUp",
6   "Ctrl-Down": "scrollLineDown",
7   "Shift-Ctrl-L": "splitSelectionByLine",
8   "Shift-Tab": "indentLess",
9   "Esc": "singleSelectionTop",
10  "Ctrl-L": "selectLine",
11  "Shift-Ctrl-K": "deleteLine",
12  "Ctrl-Enter": "insertLineAfter",
13  "Shift-Ctrl-Enter": "insertLineBefore",
14  "Ctrl-D": "selectNextOccurrence",
15  "Shift-Ctrl-Space": "selectScope",
16  "Shift-Ctrl-M": "selectBetweenBrackets",
17  "Ctrl-M": "goToBracket",
18  "Shift-Ctrl-Up": "swapLineUp",
19  "Shift-
20  "Ctrl-
21  "Ctrl-
22  "Shift
23  "Ctrl-
24  "F9":
25  "Ctrl-
26  "F2":
27  "Shift
28  "Ctrl-
29  "Shift
30  "Alt-f
31
```

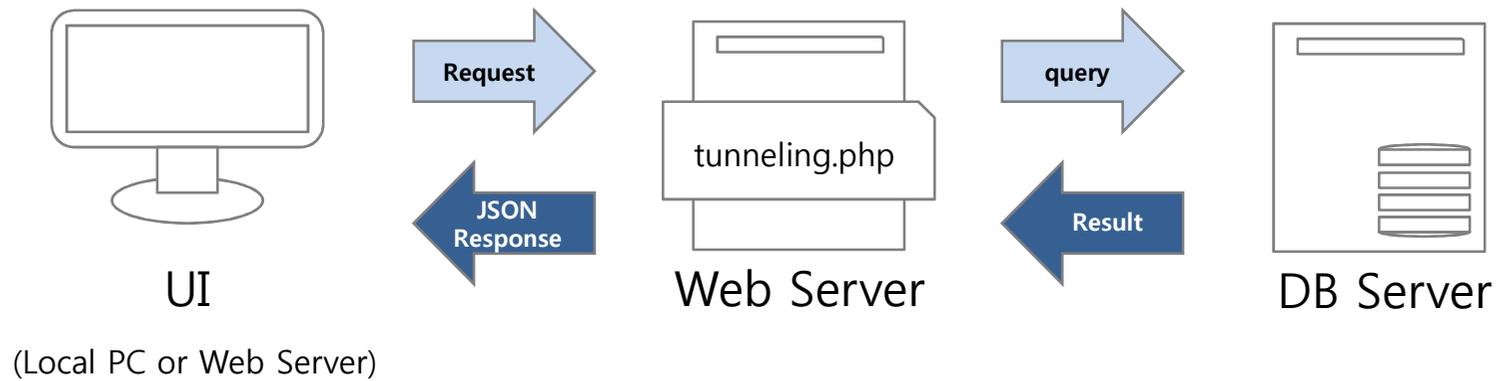
- Javascript기반의 Text Editor
- **Planche에서 Query Editor로 쓰임**
- 60개 이상의 다양한 언어 편집 지원
- 다양한 API, PLUGIN 지원



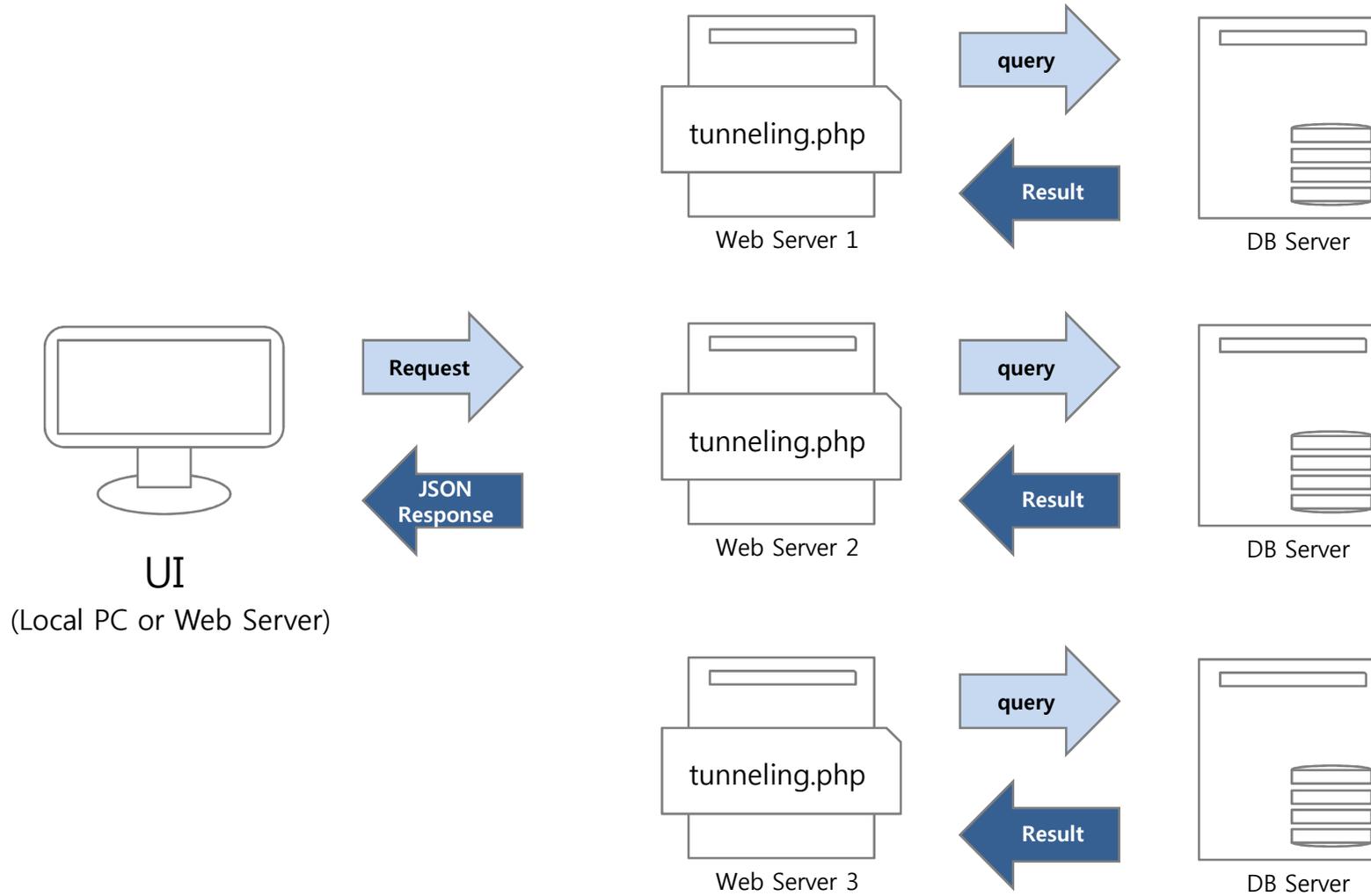
```
1 SELECT
2 `group_id`,`group_name`,`description`,`last_work_time`,`last_work_user_id`
3 FROM `WhiteSQL`.`tbl_agentgroup_info`;
4
5
6
7
8 SELECT
9 `eventlog_id`,`event_type`,`event_time`,`event_level`,`event_kind`,`policy_t
10 FROM `WhiteSQL`.`tbl_eventlog`;
11
12
13
14
15 SELECT
16 `policy_id`,`policy_name`,`policy_type`,`policy_properties`,`agent_id`,`alarm_level`,`block`,`reg_time`,`reg_user_id`,`state`,`on_
17 FROM `WhiteSQL`.`tbl_policy_list`;
```

CONCEPT

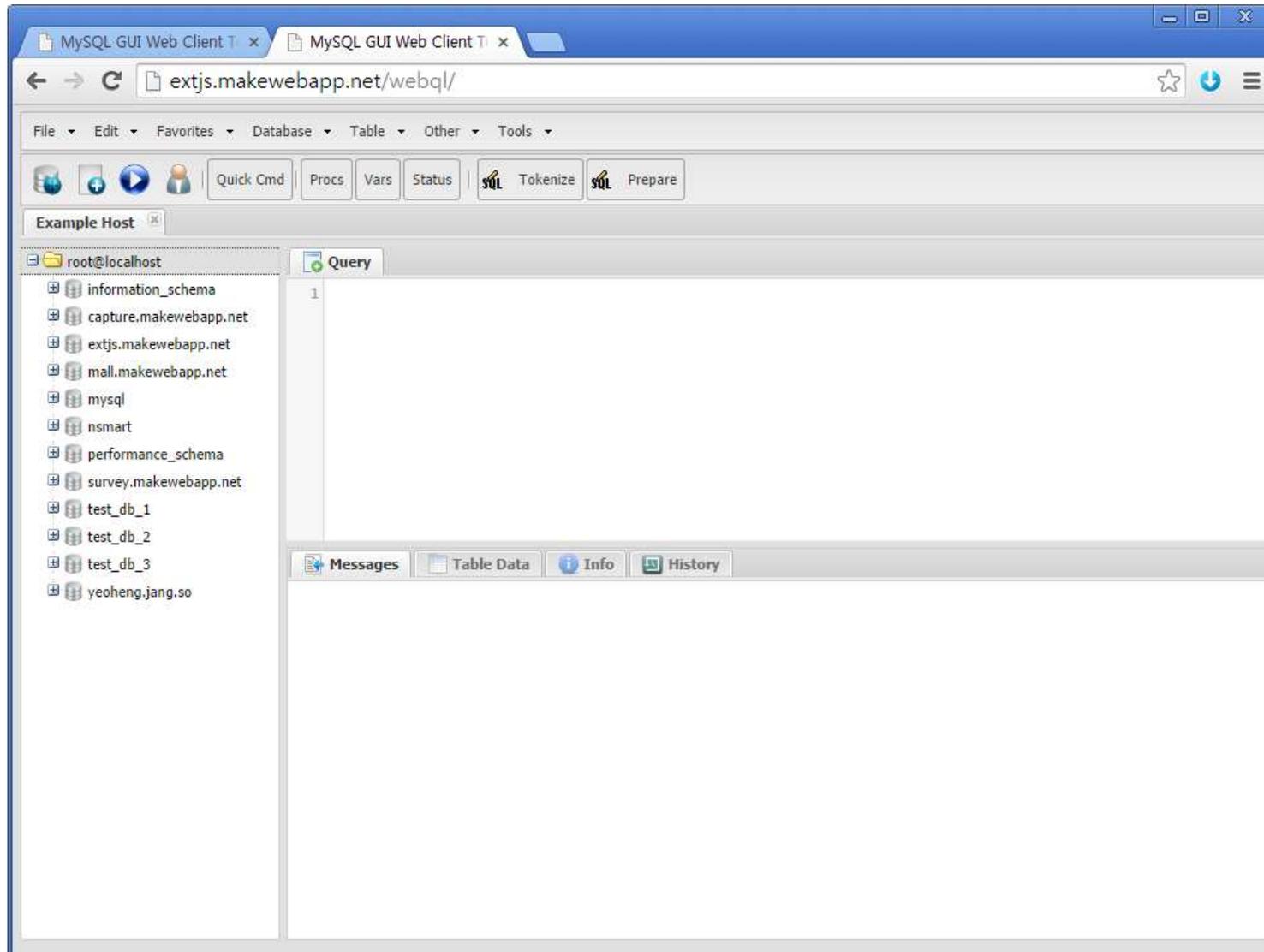
- HTTP 터널링



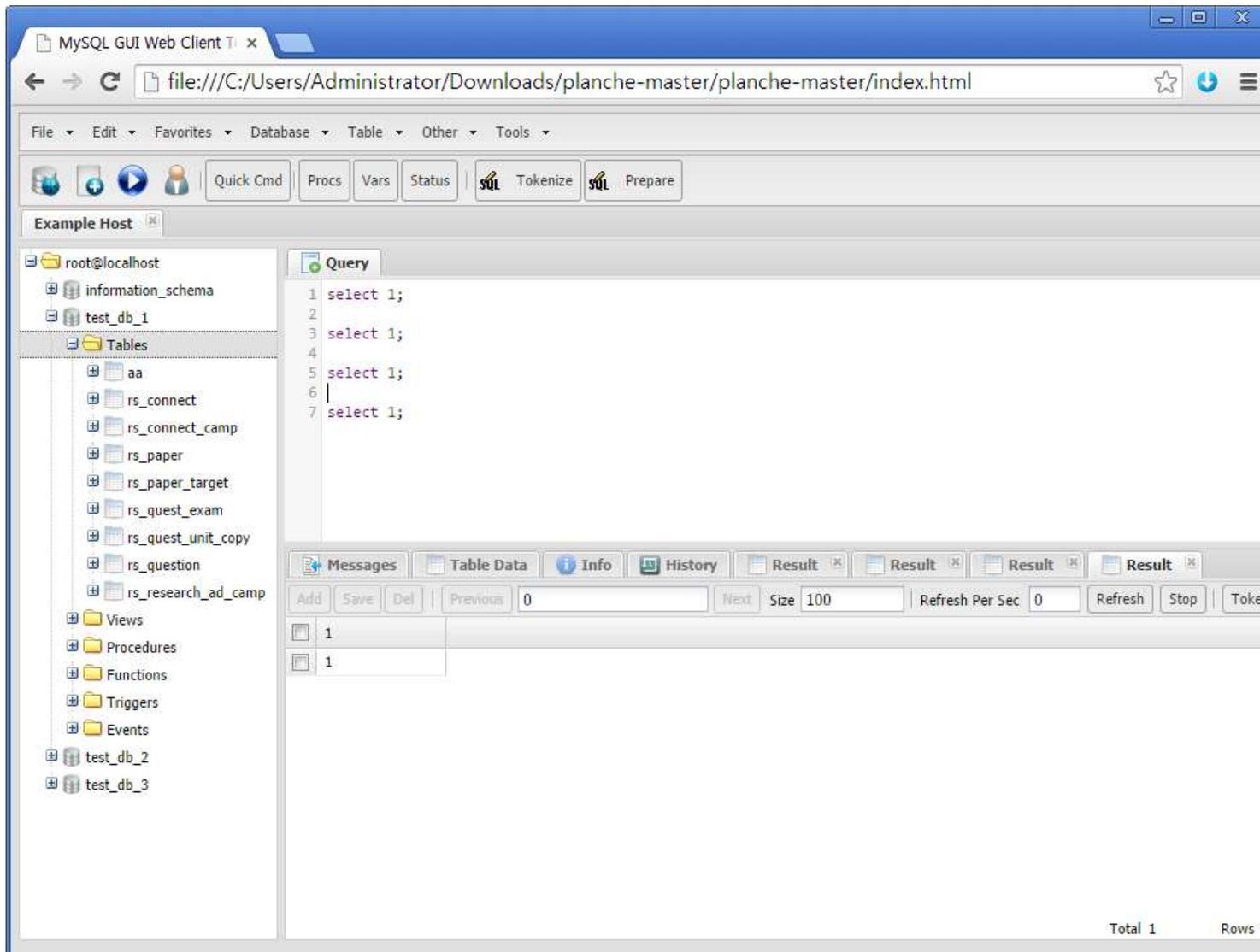
- HTTP 터널링



- 기존 처럼 웹서버에 설치 후 URL 입력



- PC에 다운 받은 상태에서 바로 실행



- AJAX를 사용하지 않음



Cross origin requests are only **supported for HTTP**.

- JSONP 사용



```

1 Ext.namespace('Planche');
2
3 Ext.application({
4   appFolder      : '.',
5   name           : 'Planche',
6   history        : [],
7   launch         : function() {
8
9     ///  
10     ///  
11     Ext.create('Planche.lib.Window', {
12       width : 800,
13       height : 500
14     });
15
16     ///  
17     Ext.Ajax.request({
18       url: 'tunneling.php',
19       params: {
20         host      : 'host',
21         user      : 'user',
22         pass      : 'pass',
23         db         : 'db',
24         query     : 'query'
25       },
26       success: function(response){
27
28         ///  
29       }
30     });
31   }
32 });
33
34

```



```

1 Ext.namespace('Planche');
2
3 Ext.application({
4   appFolder      : '.',
5   name           : 'Planche',
6   history        : [],
7   launch         : function() {
8
9     ///  
10     ///  
11     Ext.require('Planche.lib.Window', function(){
12
13       Ext.create('Planche.lib.Window', {
14         width : 800,
15         height : 500
16       });
17
18     }, this);
19
20     ///  
21     Ext.data.JsonP.request({
22       url: 'tunneling.php',
23       params: {
24         host      : 'host',
25         user      : 'user',
26         pass      : 'pass',
27         db         : 'db',
28         query     : 'query'
29       },
30       callbackKey : 'callback',
31       success: function(response) {
32
33         ///  
34       }
35     });
36   }
37 });

```

CONNECTION



Connection

호스트 설정

Host Name	Host	User	Charset	Port	HTTP Tunneling URL
Local Example	localhost	root	utf8	3306	http://192.168.0.20/tunnel.php
Example Host	localhost	planche	utf8	3306	http://extjs.makewebapp.net/webq/planche_tnl.php

```
1 var Planche = Planche || {};  
2 Planche.config = {  
3   hosts : [  
4     {  
5       host_name : 'Local Example',  
6       http_tunneling : 'http://192.168.0.20/tunnel.php',  
7       host      : 'localhost',  
8       user      : 'root',  
9       pass      : 'password',  
10      charset   : 'utf8',  
11      port      : 3306  
12     },  
13     {  
14      host_name : 'Example Host',  
15      http_tunneling : 'http://extjs.makewebapp.net/webq/planche_tnl.php',  
16      host      : 'localhost',  
17      user      : 'planche',  
18      pass      : 'planche',  
19      charset   : 'utf8',  
20      port      : 3306  
21     }  
22   ]  
23 }  
24  
25
```

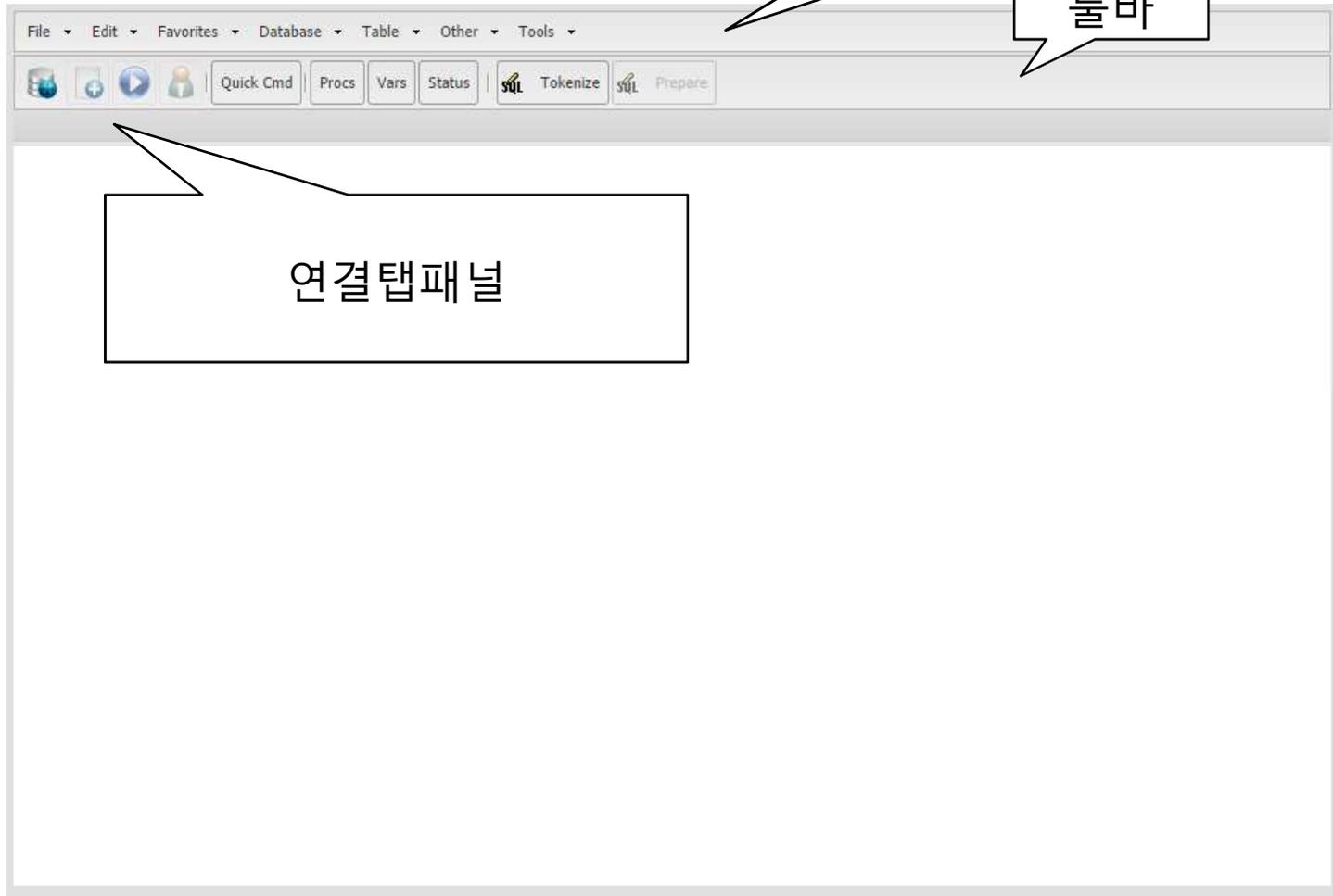
LAYOUT



메뉴

툴바

연결탭패널



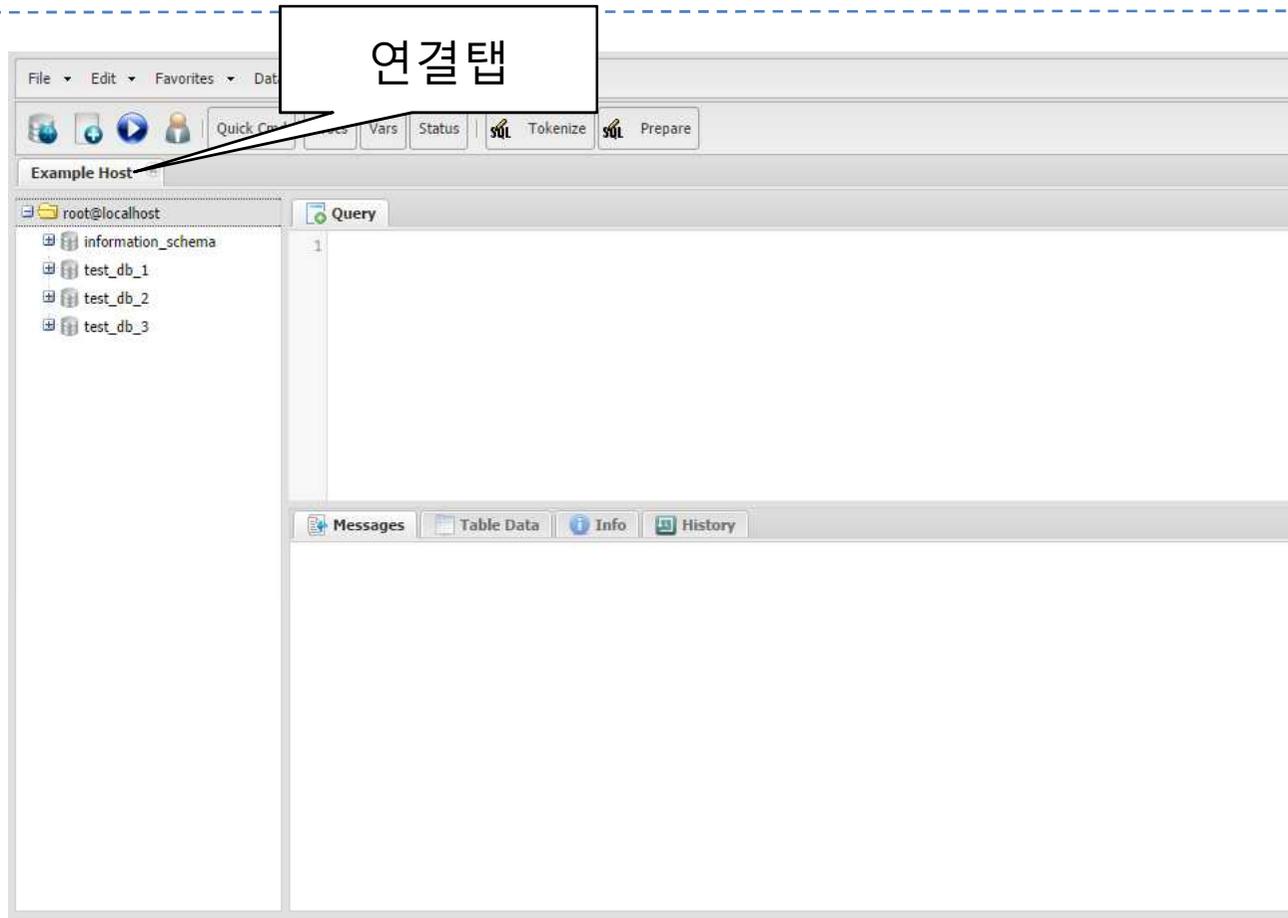
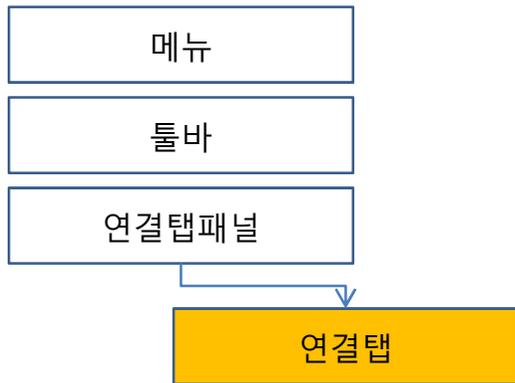
메뉴

툴바

연결탭패널

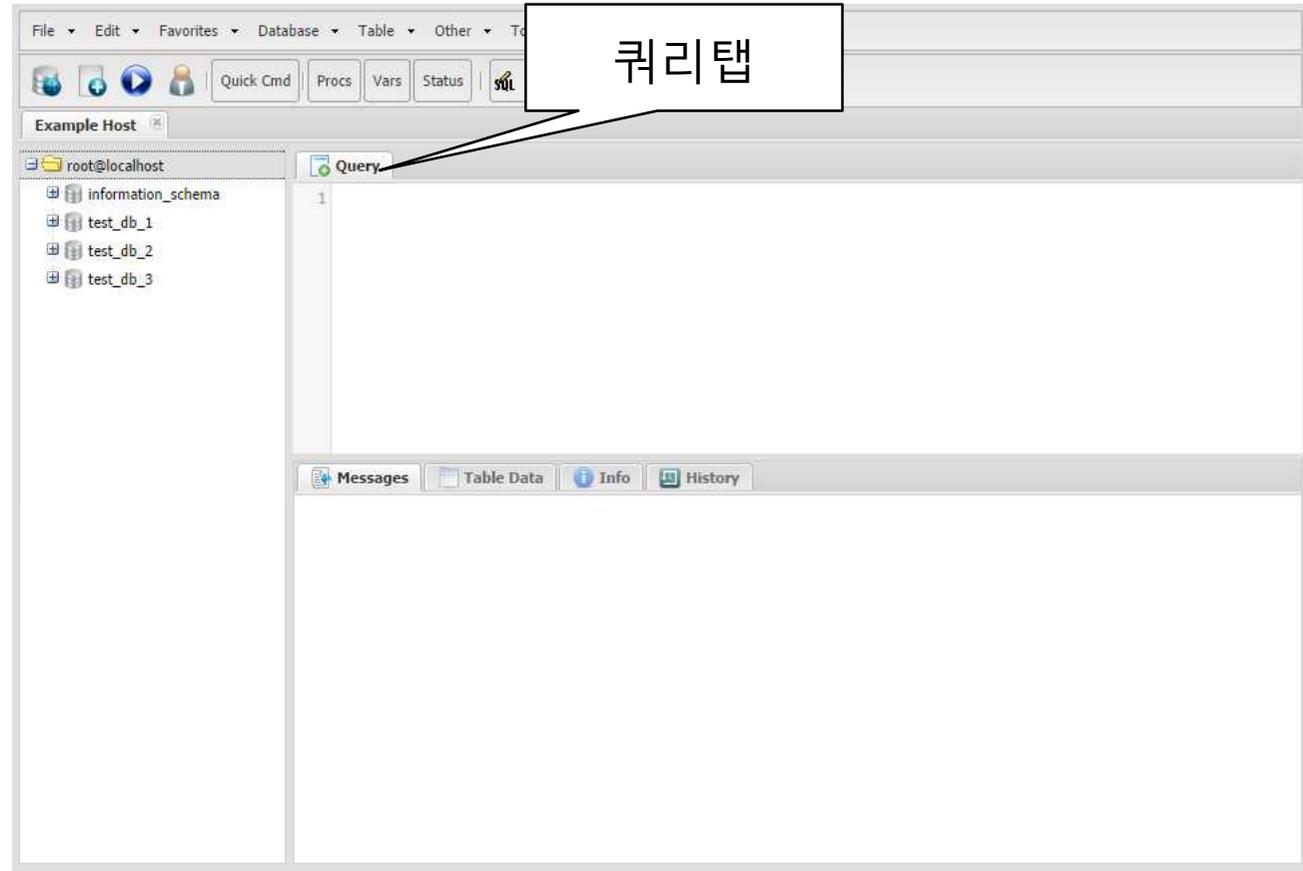
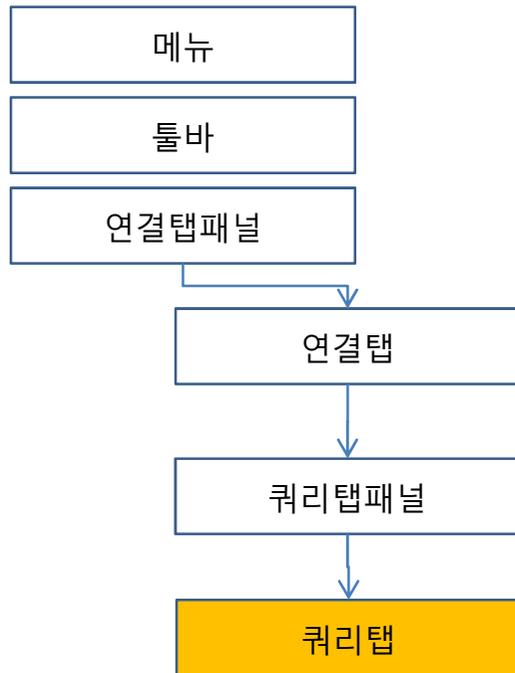


레이아웃



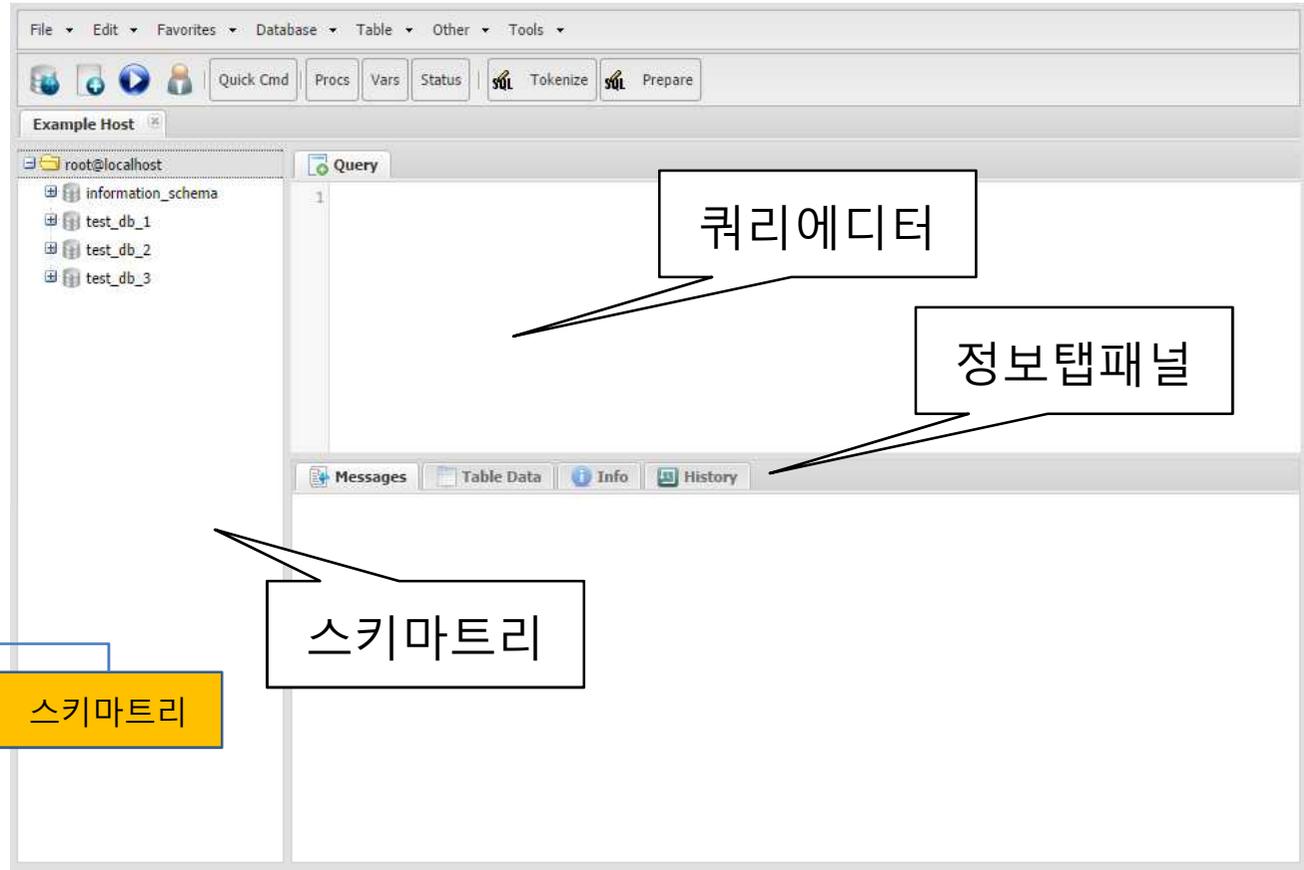
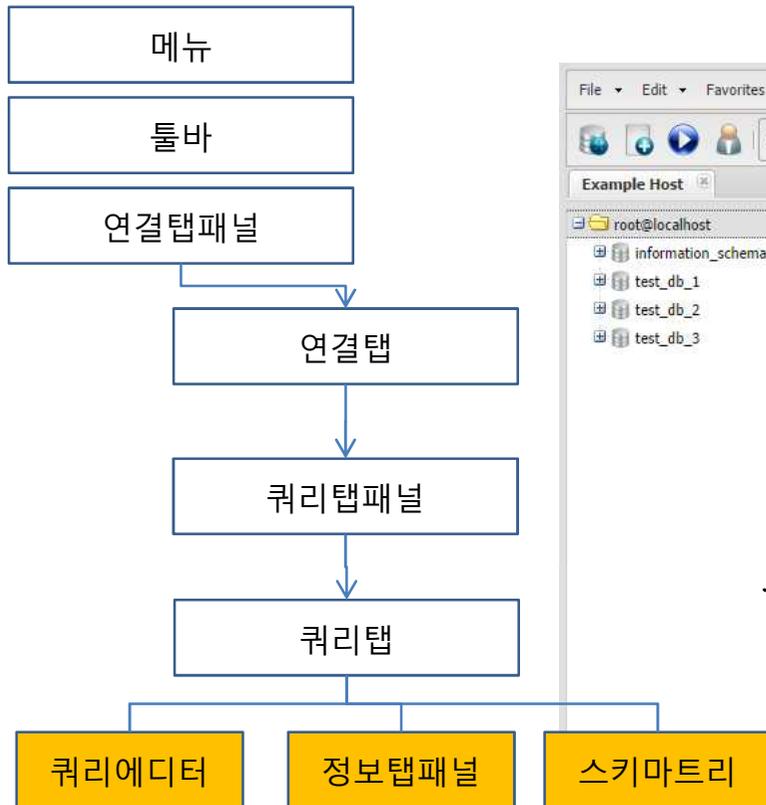


레이아웃



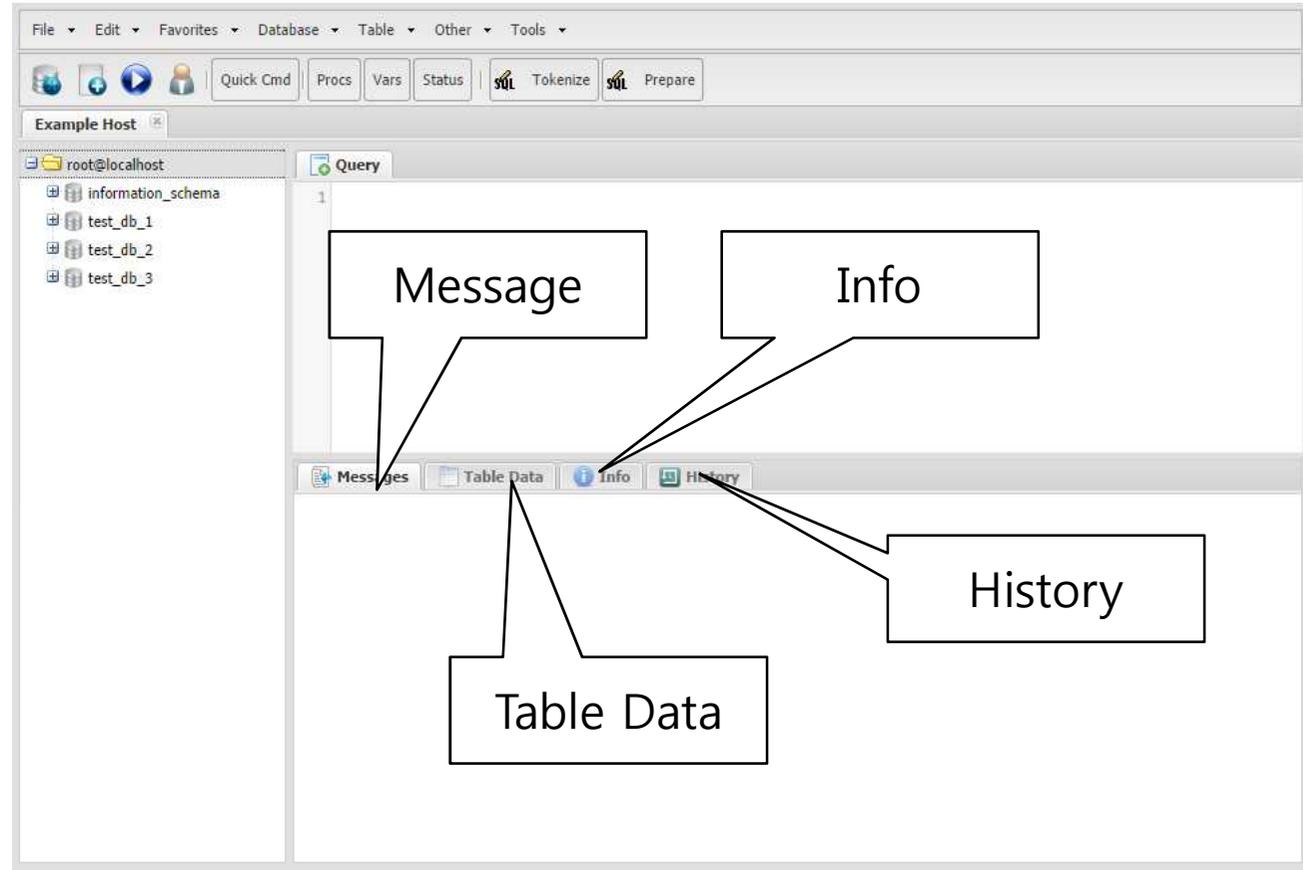
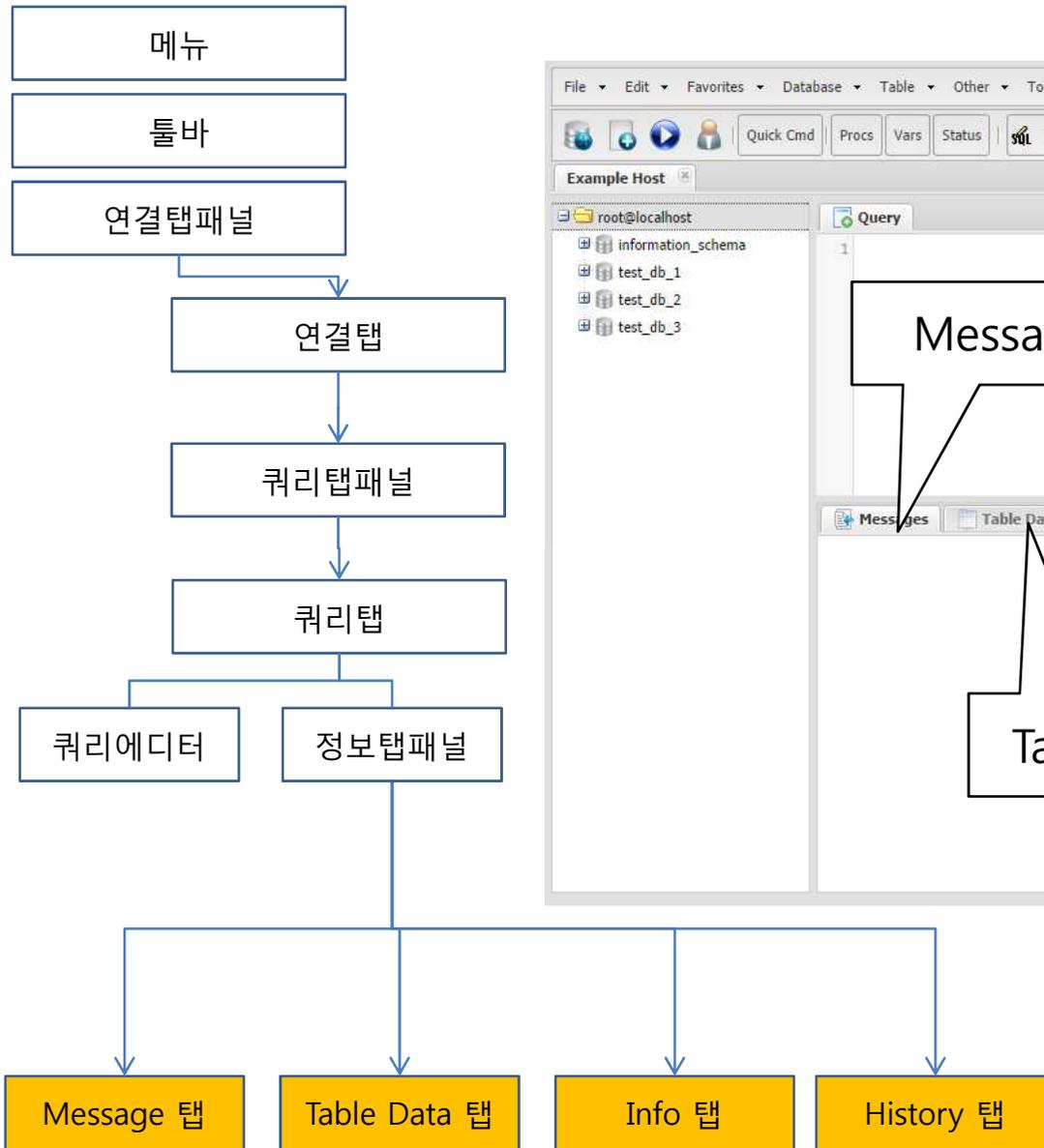


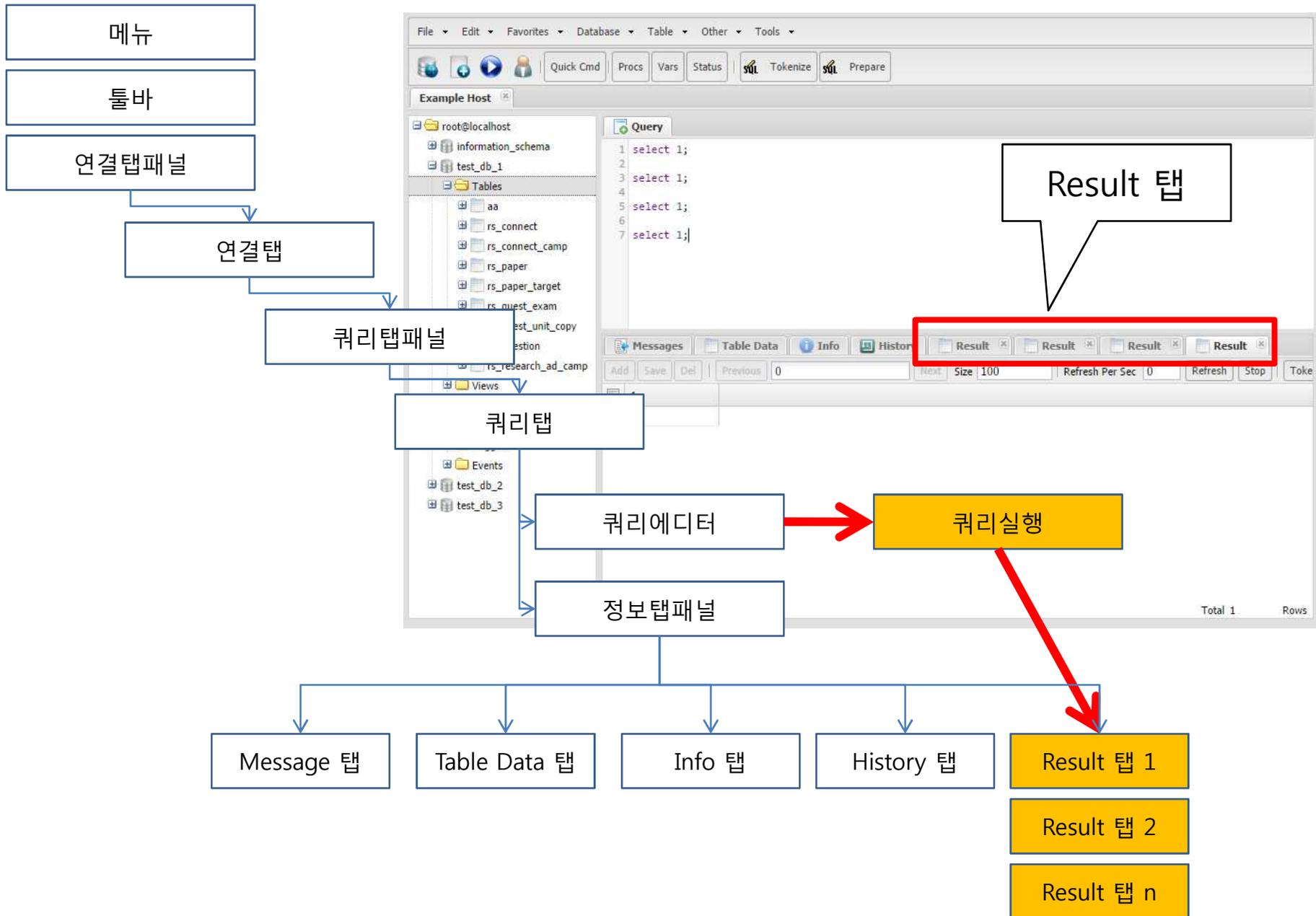
레이아웃





레이아웃

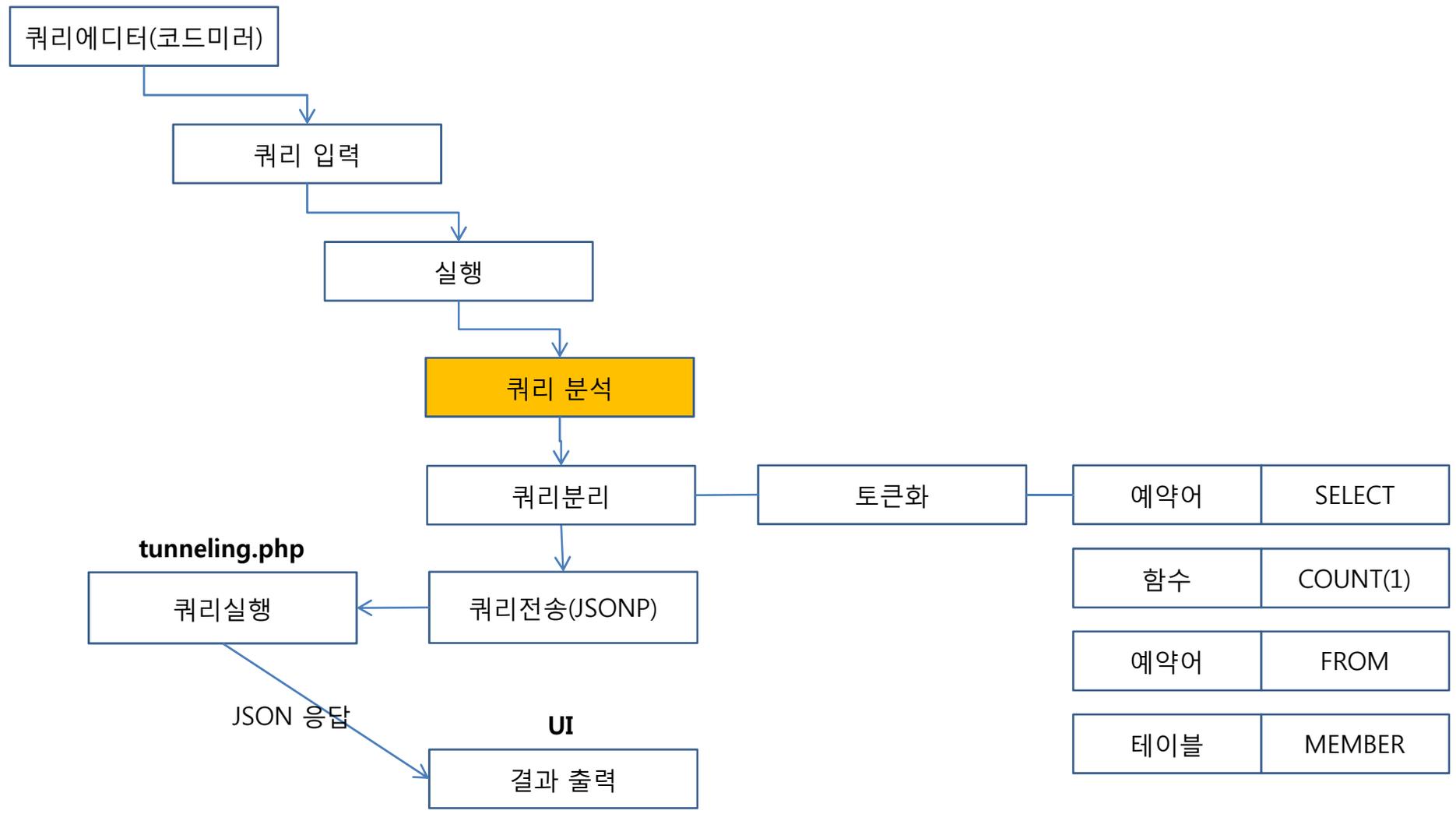




쿼리 실행



쿼리 실행 구조

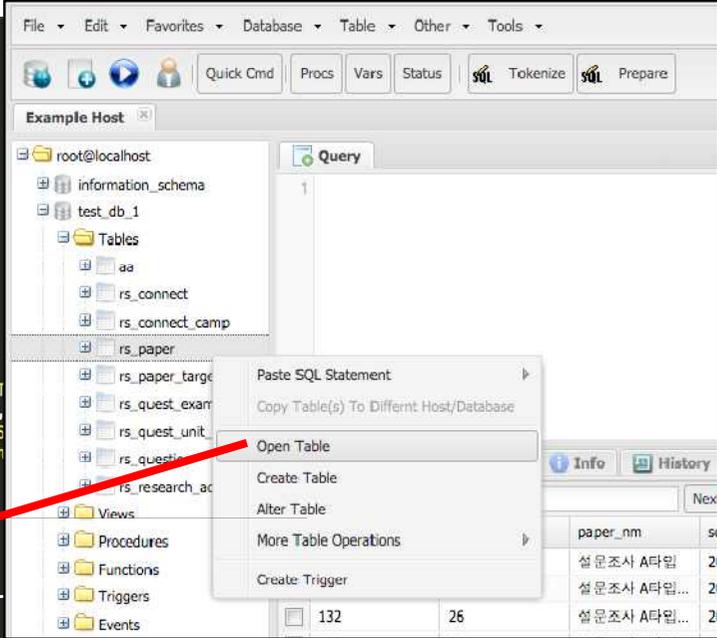




UI에 연결된 쿼리 처리 방법

Ext.String.format을 이용한 쿼리 치환

```
1 Ext.define('Planche.engine.MySQL', function(){
2
3   var queries = {
4     SELECT_USER : 'SELECT * FROM `mysql`.`user`',
5     SHOW_FULL_FIELDS : 'SHOW FULL FIELDS FROM `{0}`.`{1}`',
6     SHOW_ADVANCED_PROPERTIES : 'SHOW TABLE STATUS FROM `{0}` LIKE "{1}"',
7     SHOW_DATABASE : 'show databases',
8     OPEN_TABLE : 'SELECT * FROM `{0}`.`{1}`',
9     SHOW_PROCEDURES : 'SHOW PROCEDURE STATUS WHERE DB = "{0}"',
10    SHOW_FUNCTIONS : 'SHOW FUNCTION STATUS WHERE DB = "{0}"',
11    SHOW_TRIGGERS : 'SHOW TRIGGERS FROM `{0}`',
12    SHOW_VARIABLES : 'SHOW VARIABLES',
13    SHOW_STATUS : 'SHOW STATUS',
14    SHOW_GLOBAL_STATUS : 'SHOW GLOBAL STATUS',
15    SHOW_SESSION_STATUS : 'SHOW SESSION STATUS'
16  };
17
18  var joins = ['NATURAL JOIN', 'INNER JOIN', 'LEFT INNER JOIN', 'RIGHT INNER JOIN', 'LEFT
19  var functions = ['ABS', 'ACOS', 'ADDDATE', 'ADDTIME', 'AES_DECRYPT', 'AES_ENCRYPT', 'AREA',
20  var reserved_words = ['ACCESSIBLE', 'ADD', 'ALL', 'ALTER', 'ANALYZE', 'AND', 'AS', 'ASC', 'ASENS
21  var data_types = ['tinyint', 'int', 'varchar', 'float', 'double', 'timestamp', 'bit', 'bigin
22
23  var regexplimit = "LIMIT\\s+[0-9]+((\\s+?,|,)?(\\s+)?[0-9]+)";
24
25  return {
26    singleton: true,
27    constructor: function(config) {
28
29      this.callParent(arguments);
30    },
31    getQuery : function(query){
32
33      var args = Ext.Array.slice(arguments, 1);
34      args.unshift(queries[query]);
35      return Ext.String.format.apply(this, args);
36    },
37    getDataTypes : function(){
38
39      return data_types;
40    },
41    getDataTypesToJSON : function(){
42
43      var json = [];
44      Ext.Array.each(data_types, function(type, idx){
45
46        json.push([type, type]);
47      })
48
49      return json;
50    }
51  };
52});
```



```
openTable : function(node){
  var tab = this.getActiveTableDataTab();
  db = this.getParentNode(node),
  //파서 생성
  parser = Ext.create('Planche.lib.QueryParser', this.getEngine()),
  //선택된 MySQL 엔진에서 쿼리 가져옴
  query = this.getEngine().getQuery('OPEN_TABLE', db, node.data.text),
  //쿼리 파싱
  queries = parser.parse(query);
  query = queries[0];
  this.tunneling({
    db : db,
    query : query.getSQL(),
  });
}
```



실행되는 케이스

- ▶ 커서가 위치해 있는 곳에서 바로 실행

```
Query
1 select 1;
2
3 select 2;
4
5 select 3;
6
7 select 4;
8 |
9 select 5;
10
```

- ▶ 선택되어 있는 쿼리 실행

```
Query
1 select 1;
2
3 select 2;
4
5 select 3;
6
7 select 4;
8
9 select 5;
10
```

- ▶ 다중 선택(하나는 잘못 선택) 쿼리 실행

```
Query
1 select 1;
2
3 select 2;
4
5 select 3;
6
7 select 4;
8
9 select 5;
10
```

Messages Table Data Info History Result x

ect 2

▶ You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'ect 2' at line 1



Delimiter

예) 프로시저 생성 쿼리 – delimiter에 의한 쿼리구분자 변경 (; => \$\$)

```
1 DELIMITER $$
2
3 USE `test_db_1` $$
4
5 DROP PROCEDURE IF EXISTS `proc3` $$
6
7 CREATE DEFINER=`websql`@`localhost` PROCEDURE `proc3`()
8 BEGIN
9
10     SELECT `uniq_id`, `camp_id`, `freq`, `uid`
11     FROM `test_db_1`.`rs_connect_camp`;
12
13 END $$
14
15 DELIMITER ;
16
17 SELECT 1;
18
```



Delimiter

변경된 쿼리 구분자를 파악하여 제대로 완성된 쿼리만 보내야 한다.

```
1 DELIMITER $$
2
3 USE `test_db_1`$$
4
5 DROP PROCEDURE IF EXISTS `proc3`$$
6
7 CREATE DEFINER=`websql`@`localhost` PROCEDURE `proc3`()
8 BEGIN
9
10     SELECT `uniq_id`,`camp_id`,`freq`,`uid`
11     FROM `test_db_1`.`rs_connect_camp`;
12
13 END$$
14
15 DELIMITER;
16
17 SELECT 1;
18
```

쿼리배열에 저장된 쿼리를 하나씩 꺼내며 재귀한다.

```
executeQuery : function(){
    var queries = this.getParsedQuery();

    var tunneling;
    var messages = [];
    (tunneling = Ext.Function.bind(function(){
        var query = queries.shift();

        if(query) {
            this.tunneling({
                db : db,
                query : query.getSQL(),
                success : function(config, response){
                    if(response.is_result_query == true){
                        var grid = this.initQueryResult({
                            icon : 'images/icon_table.png',
                            closable : true,
                            title : 'Result'
                        }, db, query, response),
                    }
                    else {
                        var msg = response.affected_rows+' row(s) affected<br/><br/>';
                        msg += 'Execution Time : '+response.exec_time+'<br/>';
                        msg += 'Transfer Time : '+response.transfer_time+'<br/>';
                        msg += 'Total Time : '+response.total_time;
                        messages.push(query.getSQL()+'<br/><br/>'+msg);
                    }
                    this.getActiveMainTab().setLoading(false);
                    tunneling();
                },
                failure : function(config, response){
                    messages.push(query.getSQL()+'<span class=\'query_err\'>> '+response.message+'</span>');
                    this.getActiveMainTab().setLoading(false);
                    tunneling();
                }
            });
        }
    })
}
else {
    this.afterExecuteQuery(messages);
}
```



Tokenizing

쿼리 분석전 실행할 문자열(쿼리)을 토큰화하여 토큰별 정보를 구축한다.

Token Type	Token
DELIMITER	DELIMITER \$\$
SPACE	
RESERVED_WORD	USE
SPACE	
BOUNDARY	'
STRING	test_db_1
BOUNDARY	'
QUERY_END	\$\$
SPACE	
RESERVED_WORD	DROP
SPACE	
RESERVED_WORD	PROCEDURE
SPACE	
RESERVED_WORD	IF
SPACE	
RESERVED_WORD	EXISTS
SPACE	
BOUNDARY	'
STRING	proc3
BOUNDARY	'



Alignment

(정렬 전)

```
Query Query Query
1
2 SELECT `res_id`, (select 1) as type, `paper_id`, (SELECT `res_id` FROM `test_db_1`.`rs_conne
3 FROM `test_db_1`.`rs_connect` a left outer join ( select `uniq_id`, `start_time`, `end_time`
```

(정렬 후) 구축된 토큰별 정보로 적당한 곳에서 들여쓰기와 개행을 해준다.

```
Query Query Query
1
2 SELECT `res_id`, (
3     SELECT 1
4 ) as type, `paper_id`, (
5     SELECT `res_id`
6     FROM `test_db_1`.`rs_connect`
7 ) as aa, `uniq_id`, `start_time`, `end_time`, `ip`, `complete_yn`
8 FROM `test_db_1`.`rs_connect` a
9 LEFT OUTER JOIN (
10
11     SELECT `uniq_id`, `start_time`, `end_time`
12     FROM `rs_connect`
13     WHERE 1=1
14 ) b on 1=1
```



LIMIT

LIMIT 절이 없는 경우

The screenshot shows a SQL IDE interface with two query windows. The first query is a simple SELECT statement without a LIMIT clause. The second query is identical but includes a LIMIT clause. Below the queries, the 'Result' tab is active, showing a table with columns: res_id, paper_id, uniq_id, start_time, end_time, ip, and complete. The first two rows of the result are visible.

```
1  
2 SELECT `res_id`, `paper_id`, `uniq_id`, `start_time`, `end_time`, `ip`, `complete_yn`  
3 FROM `test_db_1`.`rs_connect`;  
4  
5  
6 SELECT `res_id`, `paper_id`, `uniq_id`, `start_time`, `end_time`, `ip`, `complete_yn`  
7 FROM `test_db_1`.`rs_connect` limit 200, 100;
```

res_id	paper_id	uniq_id	start_time	end_time	ip	complete
48	253	000150E0-8277...	2011-01-13 22:...	0000-00-00 00:...	118.103.195.153	n
38	218	000299A0-7008...	2010-08-08 13:...	2010-08-08 13:...	118.217.120.140	y



쿼리 파싱 - LIMIT 절 추가

LIMIT 절이 있는 경우

The screenshot shows a SQL IDE interface. At the top, there are menu options 'Other' and 'Tools', and buttons for 'Status', 'SQL Tokenize', and 'SQL Prepare'. Below this, there are tabs for 'Query', 'Query', and 'Query'. The main area contains a SQL query:

```
1  
2 SELECT `res_id`, `paper_id`, `uniq_id`, `start_time`, `end_time`, `ip`, `complete_yn`  
3 FROM `test_db_1`.`rs_connect`;  
4  
5  
6 SELECT `res_id`, `paper_id`, `uniq_id`, `start_time`, `end_time`, `ip`, `complete_yn`  
7 FROM `test_db_1`.`rs_connect` limit 200, 100;
```

The second query is highlighted in blue, and the 'limit 200, 100;' part is enclosed in a red box. Below the query editor, there are tabs for 'Messages', 'Table Data', 'Info', 'History', and 'Result'. The 'Result' tab is active, showing a table with columns: res_id, paper_id, uniq_id, start_time, end_time, ip, and complete. The table has two rows of data. The 'Previous' and 'Next' buttons are visible, with '200' and '100' respectively highlighted in red boxes. The 'Size' field is also set to '100'.

res_id	paper_id	uniq_id	start_time	end_time	ip	complete
34	203	00B7EC05-A6A...	2010-06-01 15:...	0000-00-00 00:...	61.99.183.91	n
135	391	00B7EF8D-D42...	2011-11-29 14:...	2011-11-29 15:...	211.178.131.12	y



Demo

Demo

<http://github.com/jeongjuwon/planche>

Q & A

<https://www.facebook.com/groups/korea.sencha/>