

Hostname: yangguang.iok.la

Port: 24015

Username: serveradmin

Password: 6o3k

## 함수:

### Uni\_Inquire('table\_name','Column\_name','condition') 검색 함수

첫 번째 parameter 는 테이블 이름.

두 번째 parameter 는 column 이름 ,"0"일 때는 모든 열을 return 한다. 아니면 조건에 맞는 열을 return 한다.

세 번째 parameter 는 조건이다. "0"일 때는 조건 없이 해당 열을 다 return 한다. 아니면 조건에 따라 return 한다.

### uni\_insert('table\_name','data1,data2..') 테이블에다가 정보 추가

첫 번째 parameter 는 테이블 이름.

두 번째 parameter 는 테이블에다가 저장할 데이터들이다. "data1,data2..." 데이터는 테이블 열이름의 순서를 맞춰서 입력해야 한다. 문자열을 저장하려면 ' 문자열 ' 로 써야 된다.

예:

uni\_insert("user","101,'uname','upw',201") user 테이블에다가 정보 추가

### uni\_delete('table\_name','condition') 삭제 함수

첫 번째 parameter 는 테이블 이름.

두 번째 parameter 는 조건이다. 조건은 "0"이면 해당 하는 테이블의 모든 데이터를 삭제 한다.

예:

call uni\_delete('user','user\_id=101') user 테이블에서 user\_id 가 "101"인 유저의 정보를 다 삭제 한다.

call uni\_delete('user','0') user 테이블의 모든 데이터를 삭제 한다.

### uni\_update('table\_name','column\_name=changed\_value','condition') 데이터 수정 함수

첫 번째 parameter 는 테이블 이름.

두 번째 parameter 는 열이름=저장할 값. 저장할 값은 문자열이면 "" 로 입력해야 된다.

세 번째 parameter 는 조건이다.

예:

uni\_update('user','user\_pw="6o3k","user\_id=102') user 테이블 중의 user\_id 가 102 인 유저의 비밀번호 '6o3k'로 바꿔 준다.

dist(user\_latitude,user\_longitude,range) 유저의 위치 정보(경위도)를 받아서 range 범위내의 대여소 정보를 반환하는 함수다.

이 PROCEDURE 는 Haversine 알고리즘을 이용해서, 조건에 해당하는 대여소 정보들이 rental\_spot 와 같은 형식의 테이블 return 한다. 이 알고리즘은 두 위치 간의 직선거리만 계산할 수 있고 해발이랑 실제 노선이 참고하지 않는다.

첫 번째 parameter 는 유저의 위도 값, 소수점 후 4 번째자리까지 있어야 하고 정

수의 최대값이 999 다.

두 번째 parameter 는 유저의 경로 값, 소수점 후 4 번째자리까지 있어야 하고 정수의 최대값이 999 다.

세 번째 parameter 는 유저위치가 중심인 원의 반지름이다. 단위는 m(미터)고 최대값은 99999 다.

구글지도랑 오차가 조금 있다. 랜덤으로 test 했는 데 다음 결과가 나타났다:

2200m 오차:2m

5210m 오차:12m

220m 오차:1m

736m 오차:5.3m

87m 오차:5m

구글지도에서 정확한 거리를 알고 싶으면 다음 사이트 이용하면 된다.

<http://www.gpsspg.com/distance.htm>

## Table:

### User

열 이름                      type                      길이    소수점    null 인지?    key                      주해

名	类型	长度	小数点	不是 null	虚拟	键	注释
user_id	int	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	unique id
user_acc	varchar	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		user account
user_pw	varchar	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		user password
umbrella	int	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		대여중인 우산 id
hash_code	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		hash_code

User 테이블은 유저정보 저장하는 테이블이다. user\_id 는 primary key 이다, 이걸로 유저의 신분을 식별 할 수 있다. 이 column 의 값들은 서버에서 결정한다, 그래서 일단 DB 에서는 자동증가로 설치 했다. 각 열의 default value 는 0 or '0', 그리고 모든 열은 값이 들어가야 한다. NULL 이면 안 된다. 예를 들면 유저가 우산을 쓰고 있지 않은 경우, 해당하는 umbrella 칸에 0 이 들어가야 한다. hash\_code 열은 똑 같다. hash code 없거나 서버가 지금 존재하는 hash code 를 삭제 하려면 '0'이 들어 가야 하다.

### User table 예|:

user_id	user_acc	user_pw	umbrella	hash_code
1001	testAcc1	123456	2001	0
1002	testAcc2	123465	0	0

## Umbrella

名	类型	长度	小数点	不是 null	虚拟	键	注释
▶ umbrella_id	int	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	우산id
umbrella_status	int	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		현재 우산 상태(1 is 보관, 0 is 대여)
borrower_id	int	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		대여종일 경우 대여하는 user의 id
slot_label	int	2	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		우산이 몇 번째 자리에 있다.

**Umbrella** 테이블은 우산정보 저장하는 테이블이다. **umbrella\_id** 는 **primary key** 이다. 모든 열은 값이 들어가야 하고 **NULL** 이면 안 된다. **umbrella\_status** 열의 **default value** 는 **1** 이고 남은 열의 **default value** 는 다 **0** 이다.

각 열의 의미는 주해에서 간단하게 설명했다. 이 중의 **borrower\_id** 랑 **slot\_label** 두 열은 더 자세히 설명 하면,

우산을 사용할 때 **borrower\_id** 열에 저장하는 데이터는 우산 사용자의 **user\_id** 다. 우산을 사용하고 있지 않으면 해당하는 대여소의 **rs\_id** 로 저장 해야 된다.

**slot\_label** 는 **2bits int** 형 값으로 우산이 몇 번째 자리에 있는 지 알려 준다. **slot\_label** 의 **default** 값은 **0** 이다. 그래서 자리 번호는 **1** 로부터 설정하고, 만약에 우산 보관기계가 **2\*3** 형태이면, 첫 줄의 첫 번째자리의 값은 **1**, 두 번째 줄의 첫 번째자리의 값은 **4** 다. 그리고 만약에 빈자리이면(해당 자리에 우산이 없다.) **0** 이 들어 가야 한다.

## Umbrella table 예:

umbrella_id	umbrella_status	borrower_id	slot_label
▶ 2001	0	1001	0
2002	1	3001	2
2003	1	3001	3
2004	1	3001	4
2005	1	3001	5
2006	1	3001	6

## rental\_spot

名	类型	长度	小数点	不是 null	虚拟	键	注释
rs_id	int	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	대여지점id
umbrella_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		소유중인 우산 id를
structure	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		대여기 배열(x X y)
▶ vacancy	varchar	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1 is 포착상태, 0 is 빈 자리
longitude	double	7	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>		좌표,경도
latitude	double	7	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>		좌표,위도

**rental\_spot** 테이블은 대여소의 정보를 저장하는 테이블이다. **rs\_id** 는 **primary key** 이다. 모든 열의 **default** 값은 **0('0')**이다, **null** 이면 안된다.

각 열의 의미는 주해에서 간단하게 설명했다.

**structure** 열은 해당하는 우산보관기계의 **n\*n** 형태를 의미한다. 예를 들면 우산보관

기계가 총 2 줄, 3 열이면, '2\*3' or '2,3' 로 저장한다. (나중에 어떻게 저장 하는지 서버 결정)

**vacancy** 열은 6bits(2\*3 형 우산보관기계)의 문자열로 해당하는 대여소(우산보관기계) 사용가능한 우산이 있는지를 알려 준다. 예를 들면 '011011'는 첫 줄의 첫 번째자리하고 두 번째 줄의 첫 번째자리가 비어 있다는 뜻이다.

**longitude** 열은 경도고 **latitude** 열은 위도다. **double** 형이고(정수범위 최대 999) 소수 범위는 0.0001 까지 표시 할 수 있다.

**rental\_spot table 예:**

rs_id	umbrella_id	structure	vacancy	longitude	latitude
3001	2002,2003,2004,2005,2006	2x3	011111	37.2974	127.9724

검색 함수:

```
1 call Uni_Inquire('user', '0', '0');
```

信息	Result 1	剖析	状态	
user_id	user_acc	user_pw	umbrella	hash_code
1001	testAcc1	123456	2001	0
1002	testAcc2	123465	0	0
1003	test3	123456	0	0
1009	dapang	19950110	9999	0

```
1 call Uni_Inquire('umbrella','0','0');
2 call Uni_Inquire('rental_spot','0','0');
3
4
```

信息	Result 1	Result 1 (3)	Result 1 (5)	剖析	状态
umbrella_id	umbrella_status	borrower_id	slot_label		
2001	0	1001	0		
2002	1	3001	2		
2003	1	3001	3		
2004	1	3001	4		
2005	1	3001	5		
2006	1	3001	6		

1	call Uni_Inquire('umbrella','0','0');
2	call Uni_Inquire('rental_spot','0','0');
3	
4	

信息	Result 1	Result 1 (3)	Result 1 (5)	剖析	状态
rs_id	umbrella_id	structure	vacancy	latitude	longitude
▶ 3001	2002,2003,2004,2005,2006	2x3	011111	37.2974	127.9724
3002	0	0	111111	37.295	126.9828
3003	0	0	111111	37.3036	126.9783
3004	0	0	111111	37.3036	126.978
3005	0	0	111111	37.2976	126.9732

테이블에다가 정보 추가:

1	call uni_insert('user',"2014','lipin','lipin2014',2014,0");
2	call Uni_Inquire('user','0','0');

信息	Result 1	剖析	状态
user_id	user_acc	user_pw	umbrella hash_code
▶ 1001	testAcc1	123456	2001 0
1002	testAcc2	123465	0 0
1003	test3	123456	0 0
1009	dapang	19950110	9999 0
2014	lipin	lipin2014	2014 0

삭제 함수:

1	call uni_delete('user','user_id=1009');
2	call Uni_Inquire('user','0','0');

信息	Result 1	剖析	状态
user_id	user_acc	user_pw	umbrella hash_code
▶ 1001	testAcc1	123456	2001 0
1002	testAcc2	123465	0 0
1003	test3	123456	0 0
2014	lipin	lipin2014	2014 0



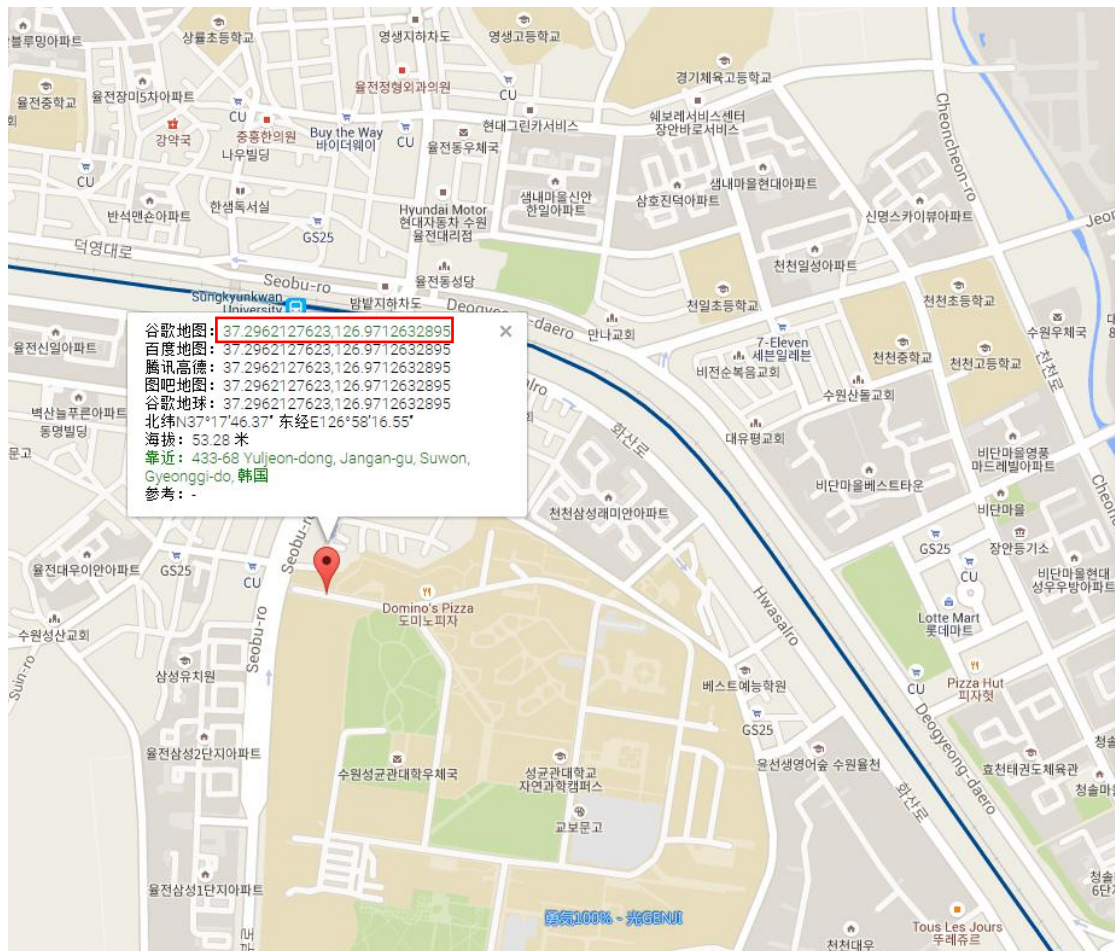
데이터 수정 함수:

```
1 call uni_update('user', 'hash_code=2014', 'user_id=2014');
2 call Uni_Inquire('user', '0', '0');
```

信息	Result 1	剖析	状态	
user_id	user_acc	user_pw	umbrella	hash_code
1001	testAcc1	123456	2001	0
1002	testAcc2	123465	0	0
1003	test3	123456	0	0
2014	lipin	lipin2014	2014	2014

dist(user\_latitude,user\_longitude,range) 함수:

유저의 위치 정보:



range 범위내의 대여소 정보를 반환한다:

Range 가 2000 이면:

```
1  call dist(37.2962,126.9712,2000);
2
```

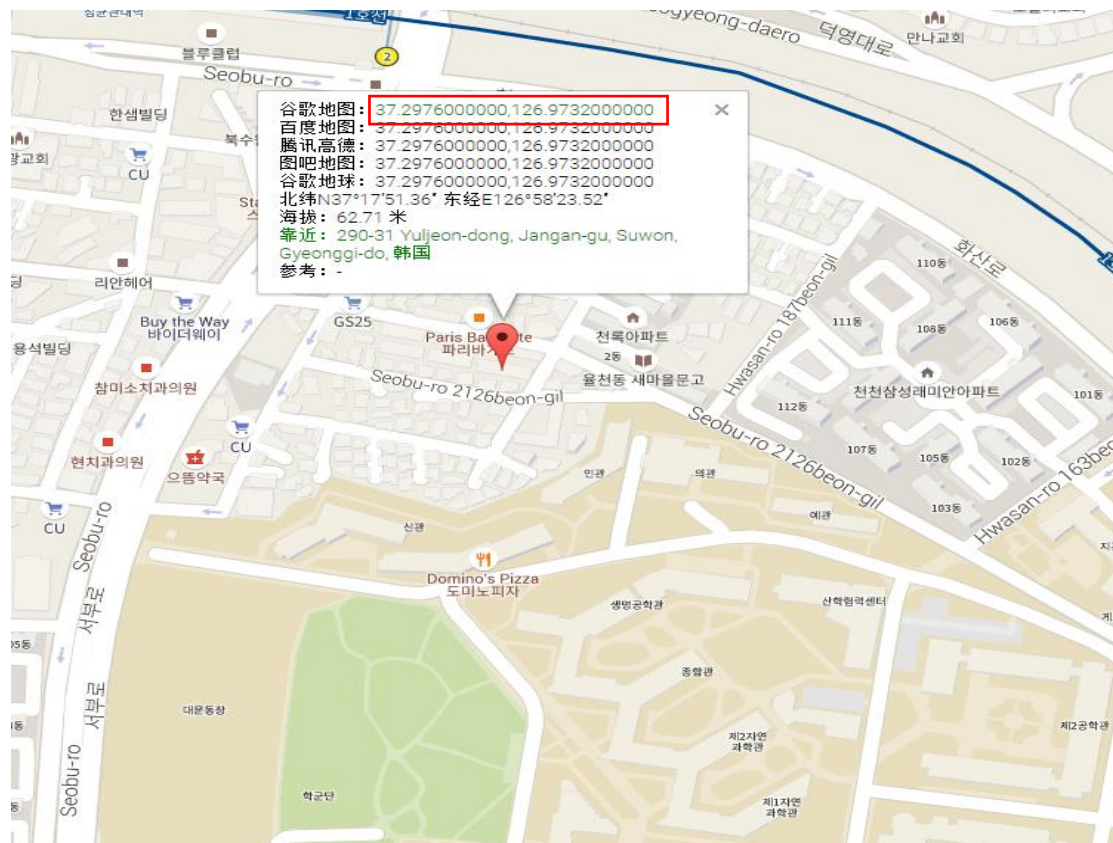
信息	Result 1	剖析	状态		
rs_id	umbrella_id	structure	vacancy	latitude	longitude
3002	0	0	111111	37.295	126.9828
3003	0	0	111111	37.3036	126.9783
3004	0	0	111111	37.3036	126.978
3005	0	0	111111	37.2976	126.9732

Range 가 1000 이면:

```
1  call dist(37.2962,126.9712,1000);
2
```

信息	Result 1	剖析	状态		
rs_id	umbrella_id	structure	vacancy	latitude	longitude
3005	0	0	111111	37.2976	126.9732

제일 가까운 대여소 위치:



실제 직선 거리:

2 점	235.92m	0.24km
2 个点	235.92 米	0.24 公里

