

# IBM | Opensource to native IBM i


John Derr  
Craig Jacquez



Goal = Determine client Country location

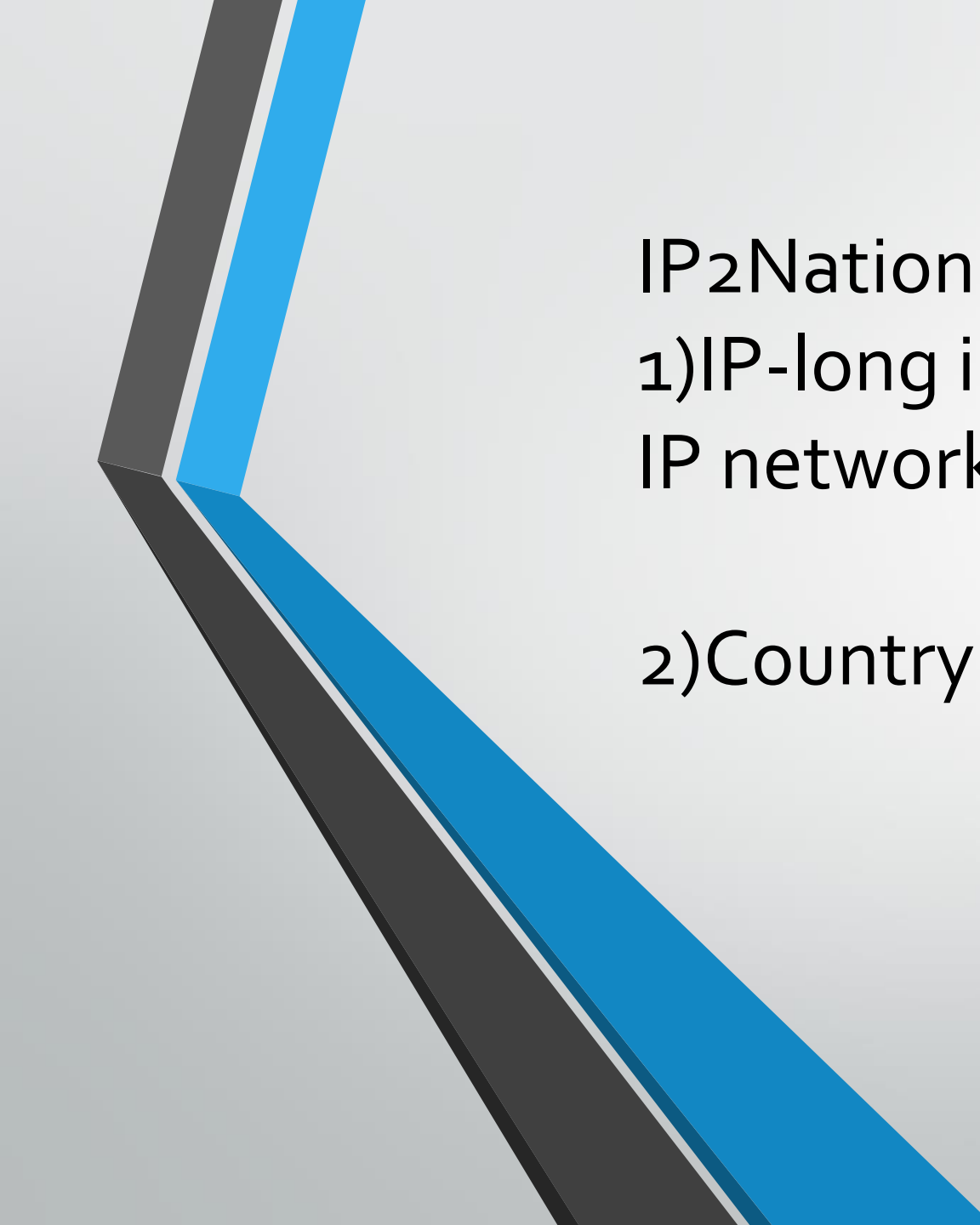
Convert MySQL Project

Implement with native IBM i



Our starting point:  
<http://www.ip2nation.com>  
Download includes MySQL script  
commands

IP2NATION.SQL download is a text file



IP2Nation table contains 2 field/columns:

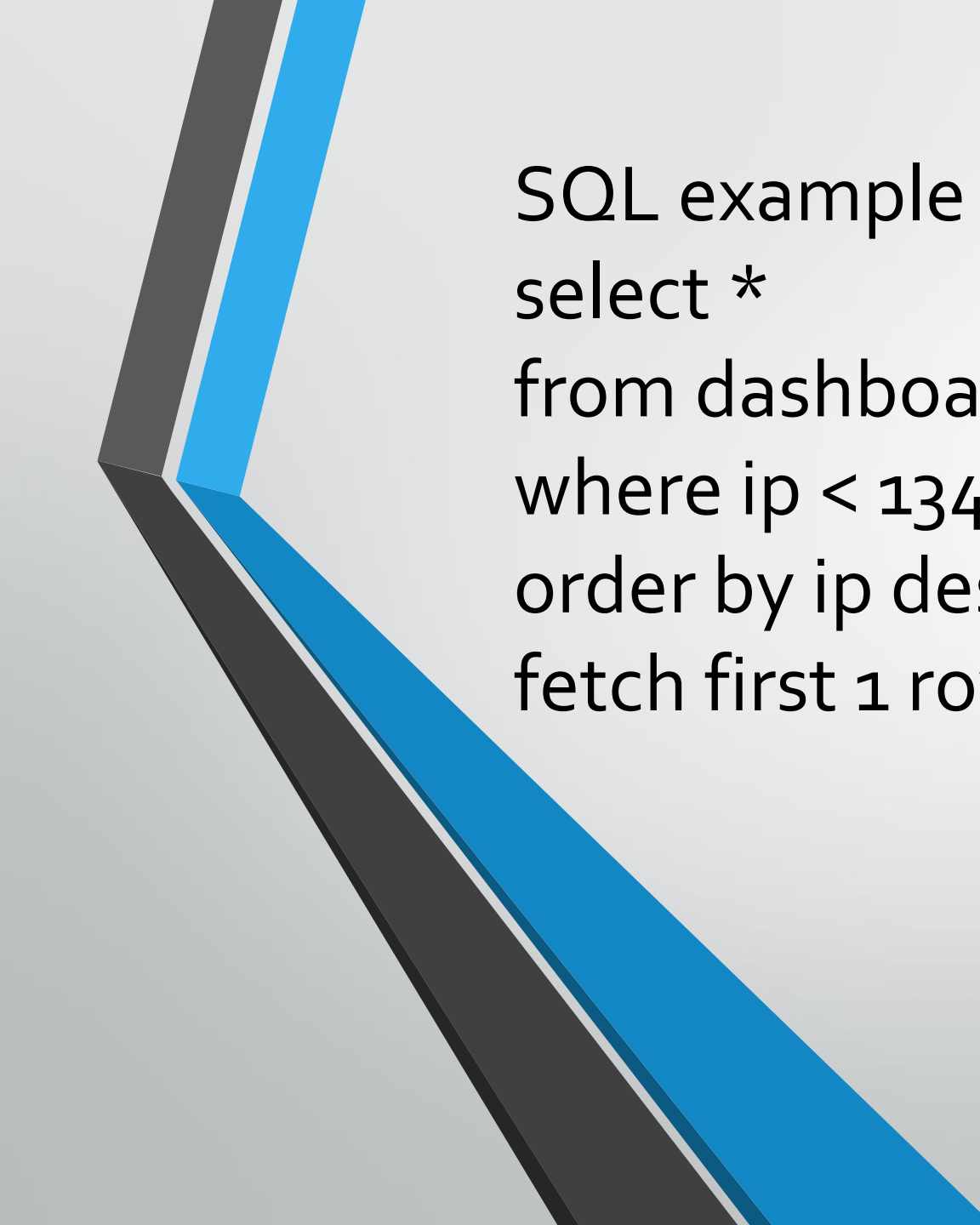
- 1)IP-long integer/decimal form (starting IP network address)
- 2)Country ID



How to use IP2Nation:

Convert ip v4 address into integer/decimal  
Position on the IP address in the table  
Read the previous IP (starting address)

Gives us the country the IP is in

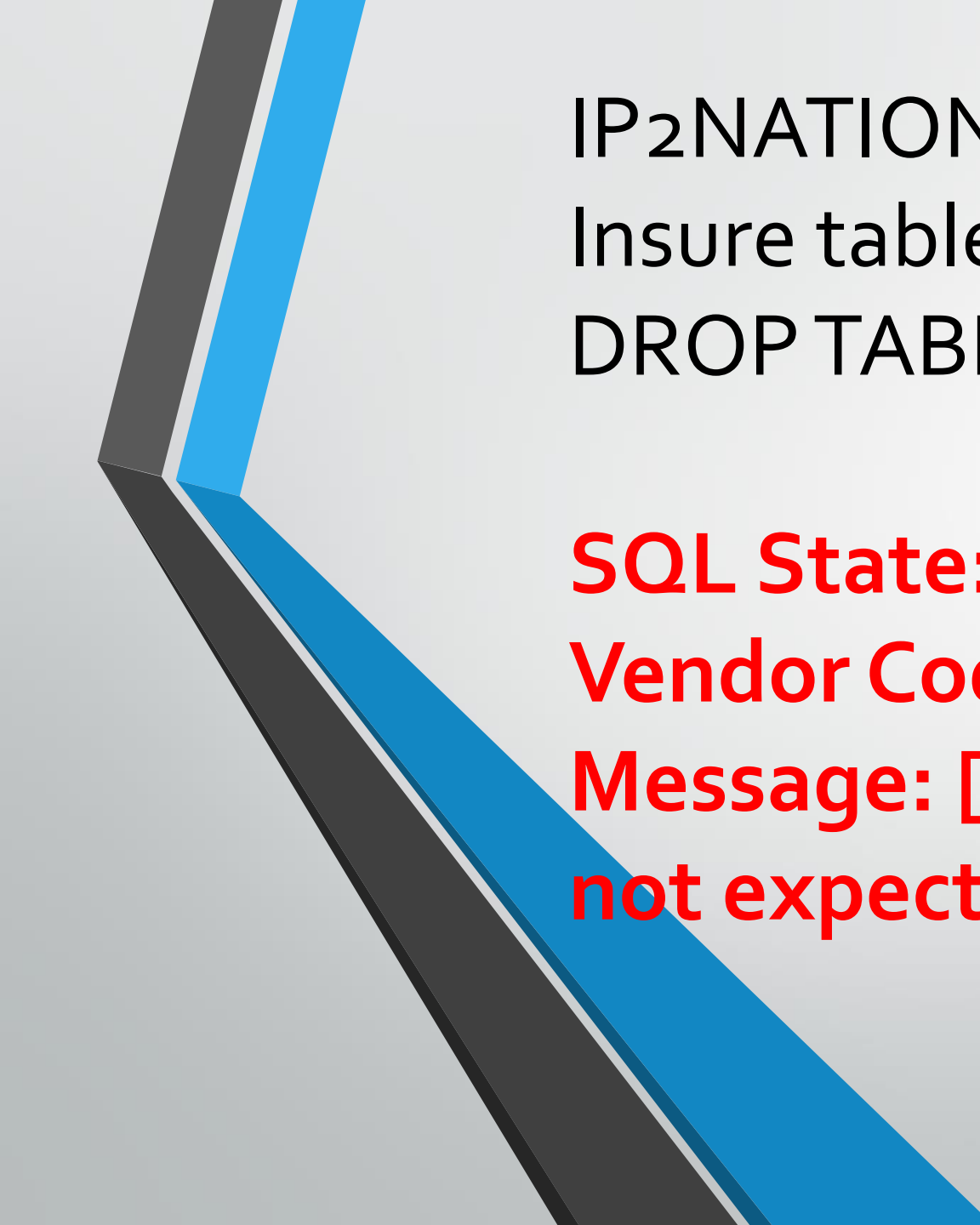


SQL example 8.8.8.8 converted to 134744072:  
select \*  
from dashboard.ip2nation  
where ip < 134744072  
order by ip desc  
fetch first 1 rows only;



IP2NATION (IP start IP integer 11 digits):  
DROP TABLE IF EXISTS schema.ip2nation;

CREATE TABLE ip2nation ( ip int(11)  
unsigned NOT NULL default '0', country  
char(2) NOT NULL default "", KEY ip (ip));



IP2NATION (IP start IP integer 11 digits):  
Insure table doesn't exist:  
DROP TABLE IF EXISTS lib.ip2nation;

**SQL State: 42601**

**Vendor Code: -199**

**Message: [SQL0199] Keyword EXISTS  
not expected.**

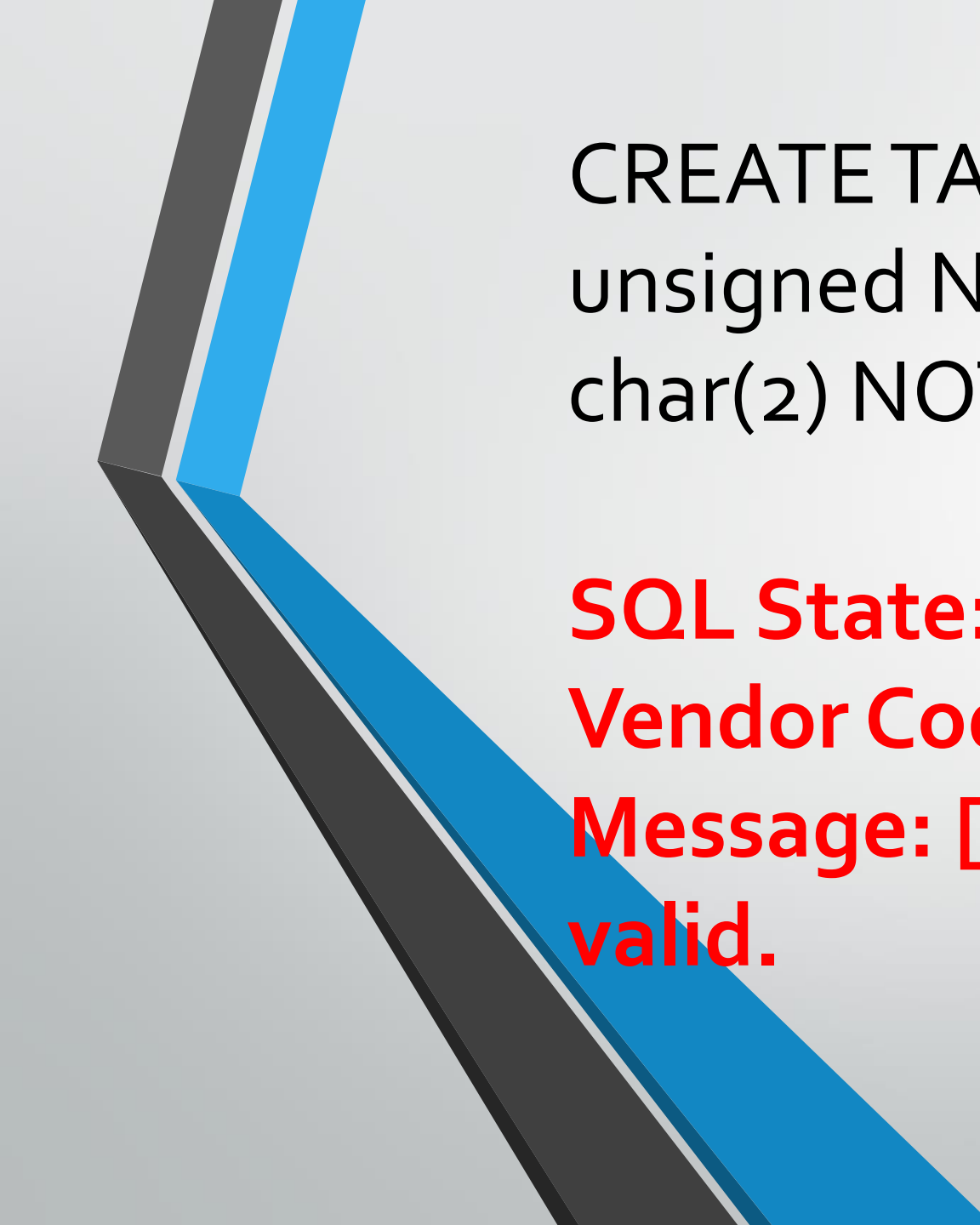




Convert to native

Insure table doesn't exist:

```
DROP TABLE lib.ip2nation;
```



```
CREATE TABLE lib.ip2nation ( ip int(11)
unsigned NOT NULL default '0', country
char(2) NOT NULL default "", KEY ip (ip));
```

**SQL State: 42601**

**Vendor Code: -104**

**Message: [SQLo104] Token INT was not  
valid.**



Convert to native:

```
CREATE TABLE lib.ip2nation ( IP  
decimal(11,0) unsigned NOT NULL default  
0, country char(2) NOT NULL default '',  
primary KEY (IP));
```



Let's load some data!

```
INSERT INTO ip2nation (ip, country)  
VALUES(o, 'us');
```

....

(63,646 rows of sql insert)



Let's load some data!

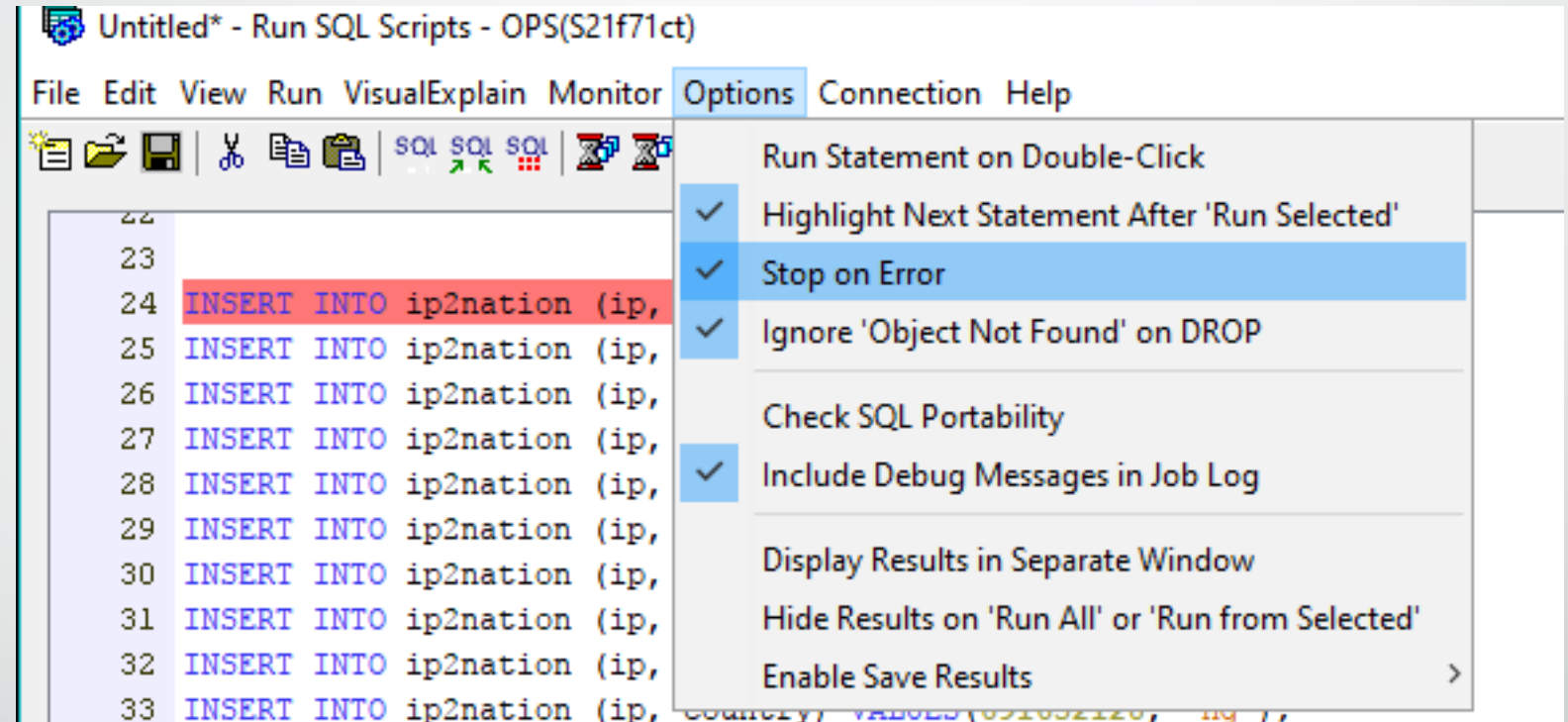
Doh!! Found some duplicate data!

**SQL State: 23505**

**Vendor Code: -803**

**Message: [SQLo803] Duplicate key value specified.**

Hmmm.





**IP2Country (Country definition 2 characters):**

**DROP TABLE IF EXISTS lib.ip2Country;**

**SQL State: 42601**

**Vendor Code: -104**


**Message: [SQL0104] Token EXIST was not valid.**




**IP2Country (Country definition 2  
characters):**

```
DROP TABLE lib.ip2Country;
```





```
CREATE TABLE lib.ip2Country ( code  
varchar(4) NOT NULL default "", iso_code_2  
varchar(2) NOT NULL default "", iso_code_3  
varchar(3) default "", iso_country  
varchar(255) NOT NULL default "", country  
varchar(255) NOT NULL default "", lat float  
NOT NULL default '0', lon float NOT NULL  
default '0', PRIMARY KEY (code), KEY  
code (code);
```




```
CREATE TABLE lib.ip2Country ( code
varchar(4) NOT NULL default "", iso_code_2
varchar(2) NOT NULL default "", iso_code_3
varchar(3) default "", iso_country
varchar(255) NOT NULL default "", country
varchar(255) NOT NULL default "", lat
decimal(9,6) NOT NULL default 0, lon
deicmal(9,6) NOT NULL default 0,
PRIMARY KEY (code));
```

There are data types: smallint, integer, bigint, short float

Note: latitude 0-90 & longitude 0-180 + precision

[https://www.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i\\_73/rzajp/rzajpequivsqlirpg.htm](https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzajp/rzajpequivsqlirpg.htm)



Load some data:

```
INSERT INTO dashboard.ip2country (code,  
iso_code_2, iso_code_3, iso_country,  
country, lat, lon) VALUES('ad', 'AD', 'AND',  
'Andorra', 'Andorra', 42.3, 1.3);
```

insert value error example:

Democratic People\'s Republic

convert native db2 sql syntax:

Democratic People"s Republic

**note: ACS sql script editor will help you find  
syntax issues**



Load some data:

```
INSERT INTO dashboard.ip2country (code,  
iso_code_2, iso_code_3, iso_country,  
country, lat, lon) VALUES('ad', 'AD', 'AND',  
'Andorra', 'Andorra', 42.3, 1.3);
```

Successful statements: 256

Note native adjustments for the “next refresh”



## PHP snippet:

**1) // convert v4 ip to integer**

```
$long_ip = sprintf('%u', ip2long($ip2nat));
```



**PHP snippet:**

```
2) $sql = "SELECT upper(N.country)  
FROM dashboard.ip2nation as N  
where N.ip < $long_ip  
order by ip desc  
fetch first 1 rows only  
";
```

# DB2 SQL IP2Nation/Country

- CREATE OR REPLACE FUNCTION **derrja.fnGetCountry** (  
    IPAddress VARCHAR(20))  
    RETURNS VARCHAR(256)  
    LANGUAGE SQL  
    SPECIFIC **derrja.fnGetCountry**  
    NOT DETERMINISTIC  
    SET OPTION  
        DBGVIEW = \*SOURCE  
    --RETURN  
    START:  
  
BEGIN

# DB2 SQL IP2Nation/Country Cont'd

- RETURN
- WITH
- -- CTE T1 - Get the long IP based on formula (
  - --    octet 1 \* 256 cubed)
  - --    + octet 2 \* 256 squared
  - --    + octet 3 \* 256
  - --    + octet 4
  - --    -----
  - --    Long IP
- T1 AS (
  - SELECT
  - CAST(SUBSTR(IPAddress,
  - 1,
  - REGEXP\_INSTR(IPAddress, '\.', 1, 1) - 1) AS BIGINT) \* (256 \* 256 \* 256)
  - + CAST(SUBSTR(IPAddress,
  - REGEXP\_INSTR(IPAddress, '\.', 1, 1) + 1,
  - REGEXP\_INSTR(IPAddress, '\.', 1, 2) - (REGEXP\_INSTR(IPAddress, '\.', 1, 1) + 1)) AS BIGINT) \* (256 \* 256)
  - + CAST(SUBSTR(IPAddress,
  - REGEXP\_INSTR(IPAddress, '\.', 1, 2) + 1,
  - REGEXP\_INSTR(IPAddress, '\.', 1, 3) - (REGEXP\_INSTR(IPAddress, '\.', 1, 2) + 1)) AS BIGINT) \* 256
  - + CAST(SUBSTR(IPAddress,
  - REGEXP\_INSTR(IPAddress, '\.', 1, 3) + 1) AS BIGINT) AS LongIP
- FROM    sysibm.sysdummy1),



## DB2 SQL IP2Nation/Country Cont'd

- -- CTE T2 Get the country code belonging to the file's long IP that is largest or equal to the searched for long ip
- T2 AS (  
• SELECT (SELECT MAX(ip) FROM derrtest.ip2Nation WHERE ip <= LongIP) AS LongIP  
• FROM T1)  
•
- -- Result, get the country name based on the country code from CTE T2
- SELECT Cntry.Country || ': ' || IPAddress || ': ' || LongIP  
• FROM T2 LEFT OUTER JOIN **derrTest**.ip2Nation Nat  
• ON LongIP = Nat.ip  
• LEFT OUTER JOIN **derrtest**.ip2NationCountries Cntry  
• ON Nat.Country = Cntry.Code;  
• END;

# DB2 SQL IP2Nation/Country Results

- values **derrja.fnGetCountry('209.85.227.147');** --  
3512066963 United States -- yes, verified
- values **derrja.fnGetCountry('146.185.223.45');** --  
2454978560 Russia -- yes, verified
- values **derrja.fnGetCountry('12.189.103.242');** --  
213739506 United States -- yes, verified
- values **derrja.fnGetCountry('146.185.223.45');** --  
2461654829 Russia -- yes, verified
- values **derrja.fnGetCountry('10.0.0.3');** Private? --  
yes, verified



## Take aways:

SQL is not too difficult to migrate from MySQL to Db2

DB2 tables accessible by SQL or native row access

RPGILE, PHP and many more languages can access Db2

Don't shy away from open source for native IBM i implementations