Constraint for phone number in SQL Server

Ask Question



Constraint for phone number to be of 7 digits. How to check if it is of 7 digits in SQL Server?





C ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY, C Name VARCHAR(255) NOT NULL, Phone INT);



sgl-server

CREATE TABLE Customer

check-constraints

edited Mar 16 '16 at 20:13



asked Mar 16 '16 at 20:04

Serena Gale **14** 1 1 5

You don't need to store phone numbers as integers. You will never be doing math on phone numbers. Store them as varchar. You can easily add a constraint to check LEN = 7 and value not like '^[0-9][0-9][0-9][0-9] [0-9][0-9]' - Sean Lange Mar 16 '16 at 20:08

try decimal(7,0) which trows error for storing more length – Sravan Kumar Mar 16 '16 at 20:08

- it worked with Phone VARCHAR CHECK(DATALENGTH(Phone)=7) Serena Gale Mar 16 '16 at 20:11
- You could make it CHAR(7), but I think your desired constraint is a bad idea in the first place. In the US, for example, phone numbers are actually 10 digits with the first three being optional. When dealing with foreign countries, you may want to store the country code along with however many digits the number is. WingedPanther73 Mar 16 '16 at 20:12

What happens when an Australian person wants to store their phone number? In that country, phone numbers can be up to 8 characters. – Greenstone Walker Mar 17 '16 at 0:14

5 Answers

Do not store phone numbers as integers. Some valid numbers, for instance, could start with a 0 -- if not today, perhaps in the future. To do the validation check, you can use like:



);

CREATE TABLE Customer (
 C_ID INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
 C_Name VARCHAR(255) NOT NULL,
 Phone CHAR(7), -- you might not want to have such a precise leng
 CONSTRAINT chk_phone CHECK (phone not like '%[^0-9]%') -- check
not a digit

Alternatively, you could write:

answered Mar 16 '16 at 20:16

Gordon Linoff

Home

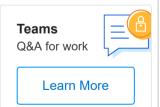
PUBLIC

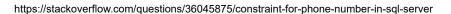


Tags

Users

Jobs









According to the <u>NANP</u>, seven-digit phone numbers in the US are NXX-XXXX, where N is 2-9 and X is 0-9. Additionally the second and third numbers cannot both be 1. Assuming you want to store only real NANP phone numbers, you could use the following constraint:



ALTER TABLE Customer

ADD CONSTRAINT CHK_Phone_valid

CHECK (Phone >= 2000000 AND Phone <= 9999999 and Phone / 10000 % 100

If you wanted to add an area code, you would need to use a bigint and different/additional constraints. Additionally, any sort in internal phone numbers, extensions, area codes, international phone numbers, etc., will have different requirements.

As someone who frequently deals with a large number of (exclusively NANP) phone numbers in databases, I do find that keeping them in a number format is ideal, as it is faster, smaller, and has the added benefit of preventing various formatting problems by default. Though for most users, this is probably all moot.

answered May 3 '18 at 14:16



Paul

873 1 10 18



Let's say you have a student table. In order to check the contact length, the following code works:





ALTER TABLE Student ADD CONSTRAINT SD CHK check(contact like '[0-9]*:

Hope this may help you

```
edited Sep 9 '18 at 19:16
      Ageax
answered Sep 9 '18 at 18:27
      YASH OSWAL
      1 1
```

it worked with Phone VARCHAR CHECK(DATALENGTH(Phone)=7)



answered Mar 16 '16 at 20:12 Serena Gale





You'd better store phone number as varchar2 or char type. In Oracle, you can check the validation of phone number by Regular

Expression:



```
CREATE TABLE Customer
    C_ID INT NOT NULL PRIMARY KEY,
    C_Name VARCHAR(255) NOT NULL,
    Phone char(10),
    CONSTRAINT valid_phone_number
    CHECK (REGEXP_LIKE(p_number, '^0\d{9}|\d{10}$'))
);
```

'^0\d{9}|\d{10}\$' means phone number must start with digit 0, then followed by 9 or 10 digits (i.e 01646947314 (11 digits) or 0123456789 (10 digits) is VALID, and 123456789 or 0123456 is not valid). You can modify this code to 7 digits by dropping "|" in regular expression and change to d{7}. Hope this help you (or someone else's having the similar problem)!

answered May 3 '17 at 5:07



Do Long Bien