Re: bit−wise and

Source:
http://www.tech−archive.net/Archive/SQL−Server/microsoft.public.sqlserver.programming/2004−12/1514.html

---

From: MGFoster (*me_at_privacy.com*)
Date: 12/06/04

Date: Mon, 06 Dec 2004 22:27:30 GMT

−−−−−BEGIN PGP SIGNED MESSAGE−−−−−
Hash: SHA1

A bit more layman−like explanation than MC's and −CELKO−'s:

If you're using Access as the front−end (forms, reports) of your
database you can set object security by using Access' user security (on
the main menu toolbar: Tools > Security > User and Group Permissions).

Unfortunately, Access doesn't provide any security ability to allow/deny
access to controls (aka Fields) on the forms. In SQL'r (2000) you can
set up allow/deny permissions on columns. I've never used that feature
so I can't really comment on it − read the books on line (BOL) for more
info.

```
−−
MGFoster:::mgf00 <at> earthlink <decimal−point> net
Oakland, CA (USA)
−−−−−BEGIN PGP SIGNATURE−−−−−
Version: PGP for Personal Privacy 5.0
Charset: noconv
iQA/AwUBQbTc2YechKqOuFEgEQI9OQCdGw1dmB0R4yxla3H8kEewysl3IggAoKm4
nu1sbbfLJaL1OOdRbyhN1Tew
=XvwG
−−−−−END PGP SIGNATURE−−−−−
MC wrote:
> The access control in SQL will rarely map properly to the permissions
> control of a non−trivial application.
> It is silly and naive to think it would. Access control secures database
> tables, views, stored procedures, etc. As such they provide a base layer of
> security that cannot be circumvented by the application(s).
>
> Frequently applications will require a UI layer of permissions that map to
> tasks that one can and cannot do, resulting in disabling buttons, hiding
> forms etc.
>
>
> "−−CELKO−−" <remove.jcelko212@earthlink.net> wrote in message
> news:O0AoKzy2EHA.3236@TK2MSFTNGP15.phx.gbl...
>
>>>>I was thinking of creatng a table with a user name and a column for
```

>>
>>representing my application's security. <<
>>
>>SQL has three sub−languages: DDL, DML and DCL. This is usually
>>mentioned in the first 20 minutes of an "Intro to SQL" lecture. The
>>last one is Data Control Language and it handles access control in a
>>standard, portable fashion.
>>
>>
>>>> So when the user opens the application some forms [sic] won't be
>>
>>shown to the user because I will AND (&) some known binary to the stored
>>data in that field [sic]. <<
>>
>>Thre are no fields in SQL; columns are a totally different concept. You
>>neither know nor care how a column is implemented PHYSICALLY, so there
>>are no bits and bytes to worry about. That kind of stuff is for
>>assembly language programming, not SQL.
>>
>>There are no forms in SQL; they exist in the front end where the user
>>does his input and output. SQL is a data retrieval language.
>>
>>
>>>>Does this sound reasonable? Does anyone have any suggestions or
>>
>>examples of how this is done? How big should this column be? <<
>>
>>No it does not. Use DCL. It should not exist at all. It might also be
>>a good idea to get the basics of a tiered architecture. You seem to be
>>thinking in terms of a monolithic application environment.
>>
>>−−CELKO−−
>>Please post DDL in a human−readable format and not a machne−generated
>>one. This way people do not have to guess what the keys, constraints,
>>Declarative Referential Integrity, datatypes, etc. in your schema are.
>>Sample data is also a good idea, along with clear specifications.
>>
>>
>>*** Sent via Developersdex http://www.developersdex.com ***
>>Don't just participate in USENET...get rewarded for it!
>
>
>