Metadata, Security, and the DBA

SET08104 Database Systems

Metadata

- DBMS table structure is more than user schema.
- Most DBMS systems use tables internally too.
- Good independence and granularity.
- Once table security is finished, everything is secure.

Oracle Metadata

- Oracle holds system metadata in SYS. E.g.

USER_OBJECTS
USER_TABLES
ALL_TABLES
USER_CONSTRAINTS
USER_CATALOG
TAB
USER_VIEWS
USER_TAB_COLUMNS
USER_TRIGGERS
DBA_USERS

Example: DBA_USERS

- USERNAME
- USER_ID
- DEFAULT_TABLESPACE
- TEMPORARY_TABLESPACE
- CREATED
- etc

Example: USER_CONSTRAINTS

- OWNER
- CONSTRAINT_NAME
- CONSTRAINT_TYPE
- TABLE_NAME
- DEFERRABLE
- DEFERRED
- LAST_CHANGE

```
Select owner,table_name,constraint_name,constraint_type
From all_constraints
Where owner = ‘DBRW’
And table_name IN
    (‘EMPLOYEE’,’JOBHISTORY’,’DEPARTMENT’)
```

OWNER	TABLE_NAME	CONSTRAINT_NAME	CON
DBRW	DEPARTMENT	SYS_C0010801	P
DBRW	EMPLOYEE	SYS_C0010803	P
DBRW	EMPLOYEE	SYS_C0010804	R
DBRW	JOBHISTORY	SYS_C0010807	P
DBRW	JOBHISTORY	SYS_C0010808	R

MySQL: creating user accounts

- mysql> **use mysql;**
- mysql> INSERT INTO user (host, user) VALUES ('localhost', ‘someusername');
- mysql> SET PASSWORD FOR someusername@localhost = PASSWORD(‘jremf6b1’);
- mysql> create database someusername;
- mysql> GRANT ALL ON someusername.* TO someusername@localhost;
- mysql> FLUSH PRIVILEGES;

MySQL System tables

- mysql> show databases;
- mysql> show tables;
- mysql> select user, host from mysql.user;
- mysql> help create table;

MySQL Metadata

- USE information_schema;

- SELECT * FROM **tables**
WHERE **table_type = 'VIEW'** ;

- Other tables in information_schema: schemata, tables, columns, statistics, views, user_privileges, …

Security

Database Security involves protection against:

- unauthorised disclosures
- alteration
- destruction

The protection which security gives is usually directed against two classes of users

- Stop people without db access from having any access.
- Stop people with database access from performing actions on the database which they are not supposed to.

Security cont…

There are many aspects to security

- Legal, social and ethical aspects
- Physical controls
- Policy questions
- Operational problems
- Hardware controls
- Operating system security
- Database system security

Granularity of DBMS Security

The unit of data used in specifying security in the database can be, for example;

- the entire database
- a set of relations
- individual relation
- a set of tuples in a relation
- individual tuple
- a set of attributes of all tuples
- an attribute of an individual tuple.

DBMS-level Protection

- Data encryption:

  Often it is hard to prevent people from copying the database and then hacking into the copy at another location. It is easier to simply make copying the data a useless activity by encrypting the data. This means that the data itself is unreadable unless you know a secret code. The encrypted data in combination with the secret key is needed to use the DBMS.

- Audit Trails:

  If someone does penetrate the DBMS, it is useful to find out how they did it and what was accessed or altered. Audit Trails can be set up selectively to minimise disk usage, identify system weaknesses, and finger naughty users.

User-level Security for SQL

- Each user has certain access rights on certain objects.
- Different users may have different access rights on the same object.

In order to control the granularity of access rights, users can

- Have rights of access (authorisations) on a table
- Have rights of access on a view. Using views, access rights one can control horizontal and vertical subsets in a table, and dynamically generated data from other tables.

The GRANT command

GRANT is used to grant privileges to users:

```
GRANT privileges ON tablename
TO { grantee ... }
[ WITH GRANT OPTION ]
```

Possible privileges include:

- SELECT - user can retrieve data
- UPDATE - user can modify existing data
- DELETE - user can remove data
- INSERT - user can insert new data
- REFERENCES - user can make references to the table

GRANT cont...

The WITH GRANT OPTION permits the specified user can grant privileges which that user possesses on that table to other users. This is a good way to permit users to look after permissions for certain tables, such as allowing a manager to control access to a table for their subordinates.

Grantee need not be a username or a set of usernames. It is permitted to specify PUBLIC, which means that the privileges are granted to everyone.

```
GRANT SELECT ON userlist TO PUBLIC;
```

GRANT and VIEWs

- VIEW security validated at view create time

```
GRANT select on employee to jim;

Create view empjim as
    select empno,surname,forenames from employee;

GRANT select on empjim to jim;

REVOKE select on employee for jim;
```

Restrict Rows

```
CREATE table checker (
    username            varchar(200)
  , secretinfo          varchar(100)
);
CREATE VIEW userview as
    select * from checker
    where username = USER;

Select * from userview;
```

The DBA

- DBA – Database Administrator
- Maintain system usability
- Provide security
- Target for hate mail…

Area of concern

- System performance and tuning
- Data backup and recovery
- Product+tool selection, install, maintain
- System documentation
- Support
- Education
- Fortune Telling / Future Prediction