Security II

Using Principals, Permissions and Boundaries

Scaling Management

- **Fortress dba**
  - think of a single server limited by max scale up
- **Islands of responsibility**
  - scaling out management
- **Schemas, roles, execution contexts**

Schemas

- Schema is set of database objects
- Management boundary

```
darkmatter5.AdventureWorks.HumanResources.Department
darkmatter5.AdventureWorks.HumanResources.Employee
darkmatter5.AdventureWorks.HumanResources.EmployeeAddress
…
```

server name

Schemas Summary

- Use schema to "scale out" management of database objects
- Schema level permission
- Schema owner manages
- May need to add Grant Create *

Roles

- **Has permissions for particular kinds of takes**
  - principal assumes of role they are assigned to
- **Server and database level roles**

Server Roles

- **Fixed only**

```
sysadmin
```

Database Roles

- **Fixed and flexible**

```
db_owner
```

Roles Summary

- **Roles group permissions**
- **Fixed roles for general management tasks**
- **Flexible roles for specific tasks**
- **Roles can deny or grant**
  - server level applies to everything on the server
  - database level applies to everything in the database
- **Deny takes precedence over grant**

Execution context

- **Execution context - authority of user**
- **Impersonation changes execution context**
  - login - server level impersonation
  - user - database level impersonation

Changing Execution Context

- **Imperative, as you go**
  - execute as user or login
- **Declarative, as part of create**
  - execute as clause
  - stored procedures, functions, triggers, queues
- **Revert, undoes impersonation**

Execution Context Scope

- Execute as user -- database
- Execute as login -- server

```
execute as user = 'joe' // securityII is database in use
```

```
select * from AdventureWorks.HumanResources.employee
```

fails

Execution Context Summary

- Execution context can be imperatively or declaratively controlled
- Execution context scope
- Care must be taken when use dynamic sql and execute as

Summary

- Schemas partition off database objects into manageable sets
- Roles are sets of permissions needed for tasks
- Execution context manages impersonation
- Execution context make dynamic sql more useful