2008 ESRI User Conference
Technical Workshops
August 4–8, 2008

Please!
Turn OFF cell phones
and paging devices

ArcSDE Administration for PostgreSQL

Ale Raza, Brijesh Shrivastav, Derek Law
ESRI - Redlands

Outline

- Introduce ArcSDE technology for PostgreSQL
- Implementation
- PostgreSQL performance – tips and tricks
- Common tasks
- Summary
- Additional Resources

- **Prerequisities:**
  1. Working knowledge of the geodatabase
  2. Basic DBMS knowledge

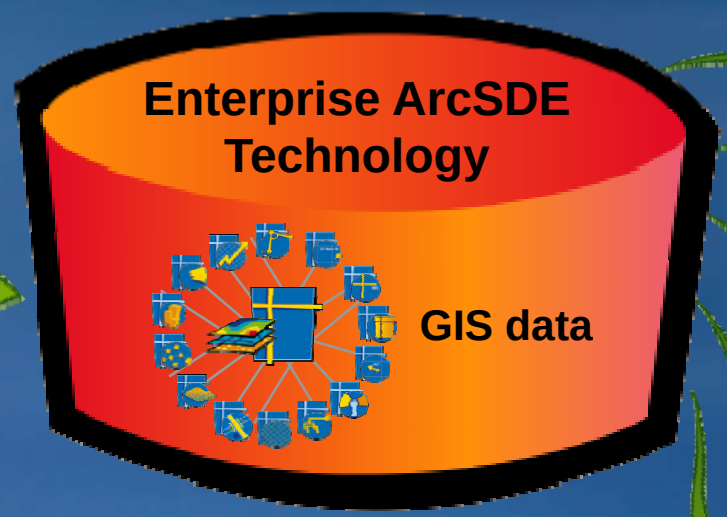
Outline

- Introduce ArcSDE technology for PostgreSQL
  - Review: enterprise geodatabase
  - Enterprise ArcSDE technology
  - PostgreSQL DBMS
- Implementation
- PostgreSQL performance – tips and tricks
- Common tasks
- Summary
- Additional Resources

ArcGIS Server Enterprise

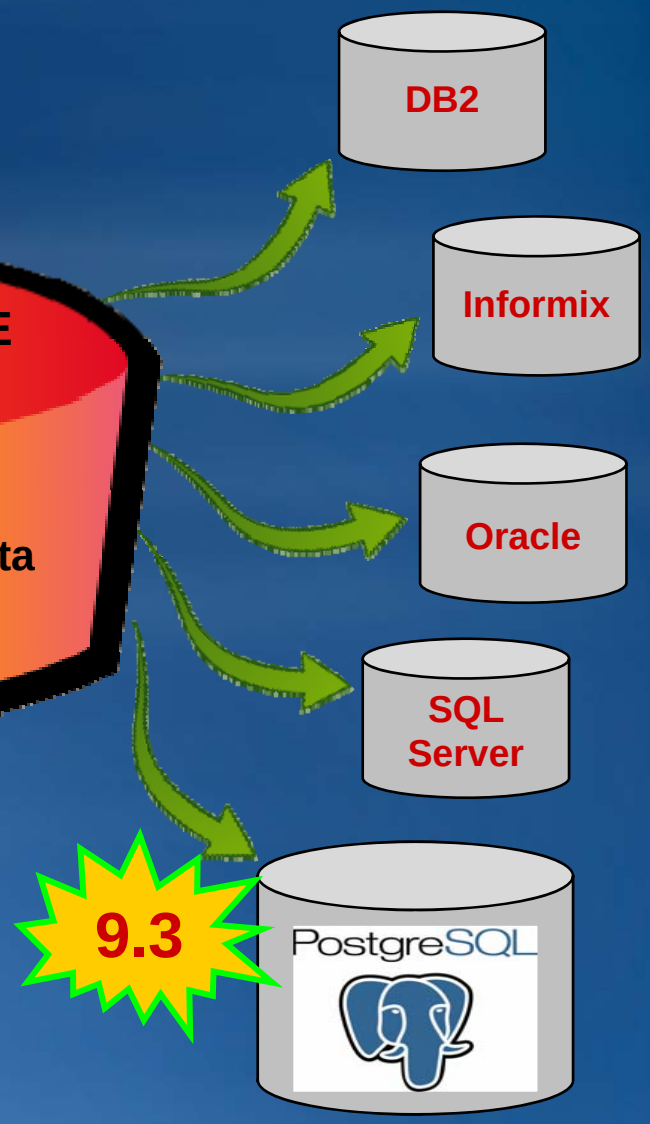
All editions (Basic, Standard, Advanced)

GIS clients



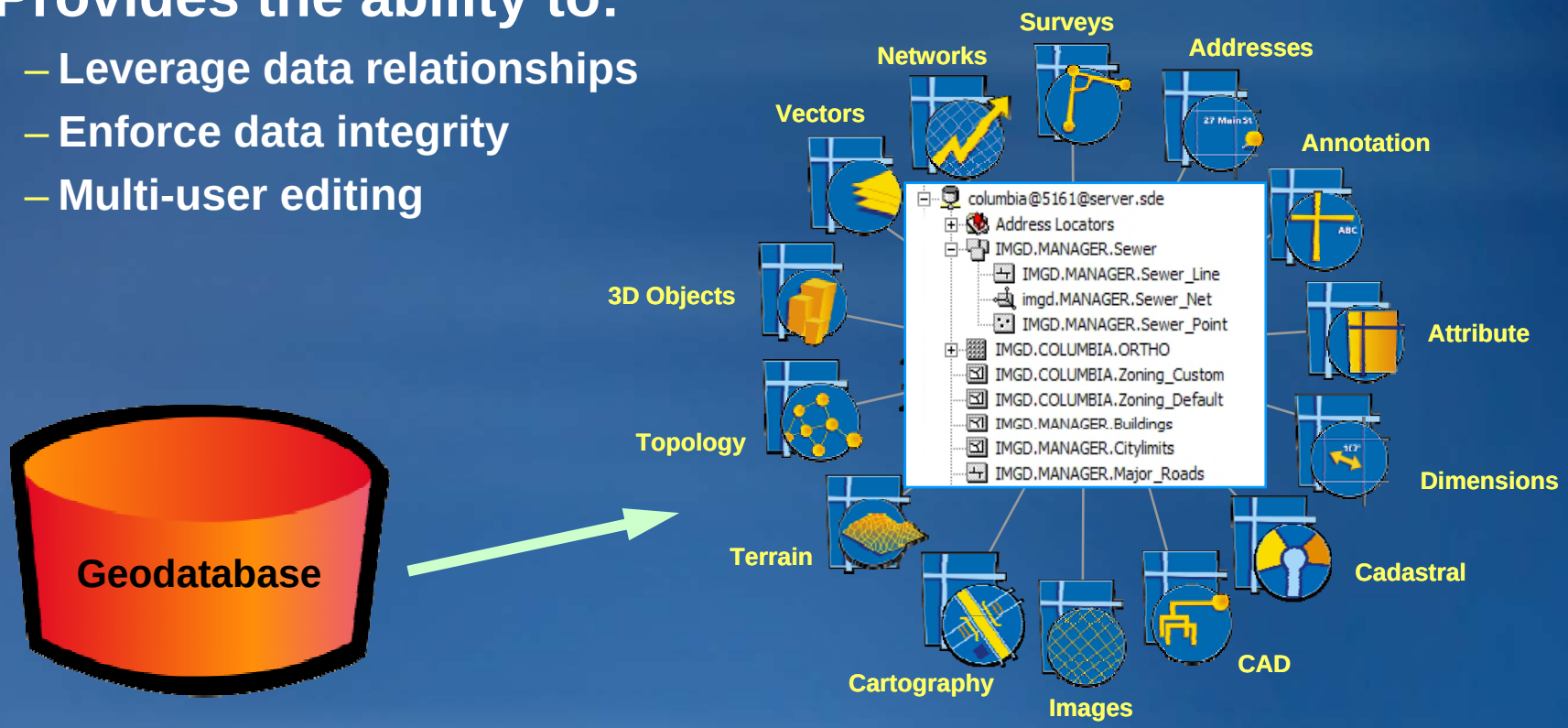
Enterprise Geodatabase

Supported DBMS platforms



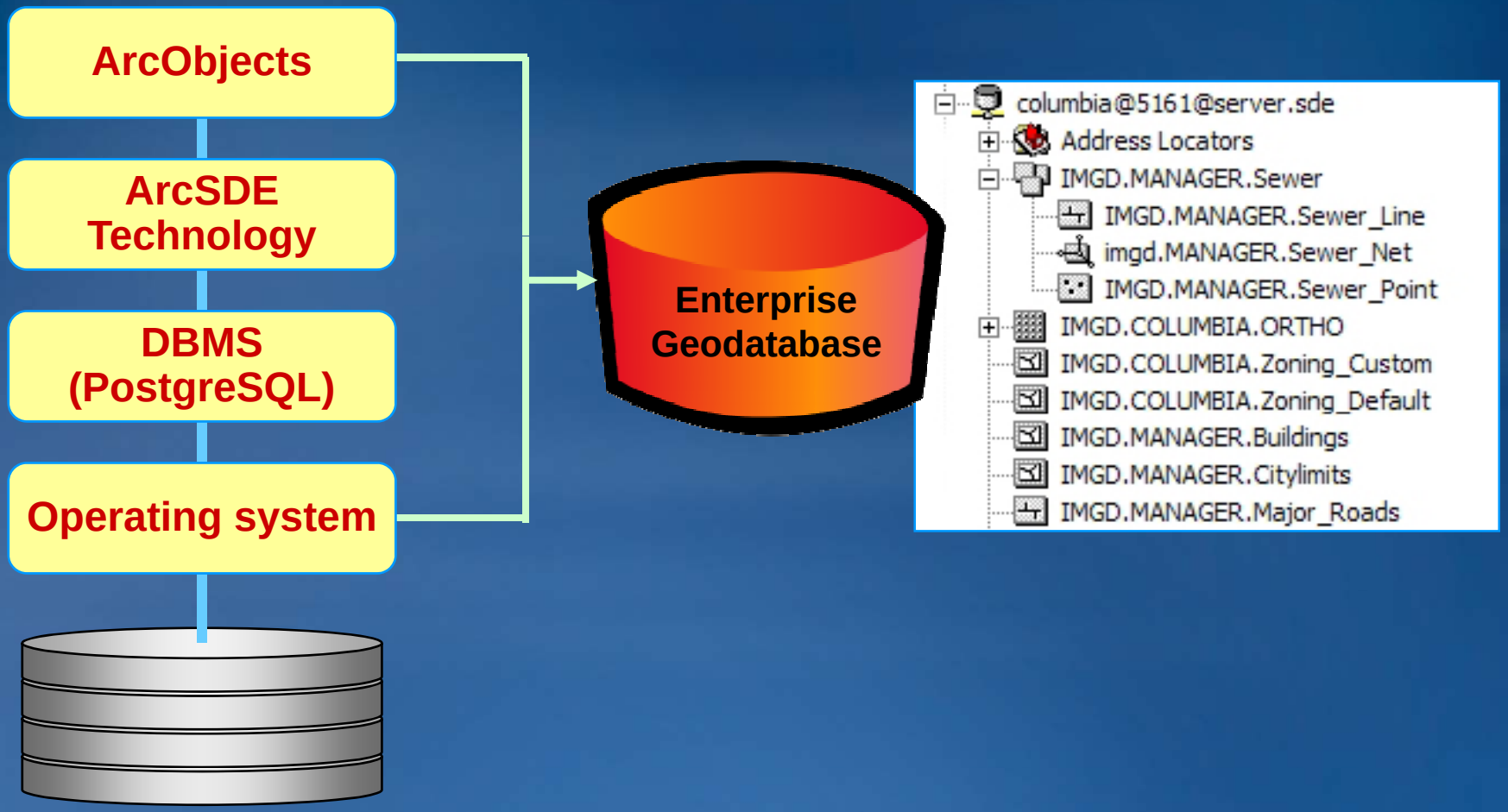
Defining the geodatabase

- Native data structure for ArcGIS
- Container of spatial & attribute data
  - Collection of geographic datasets
- Provides the ability to:
  - Leverage data relationships
  - Enforce data integrity
  - Multi-user editing



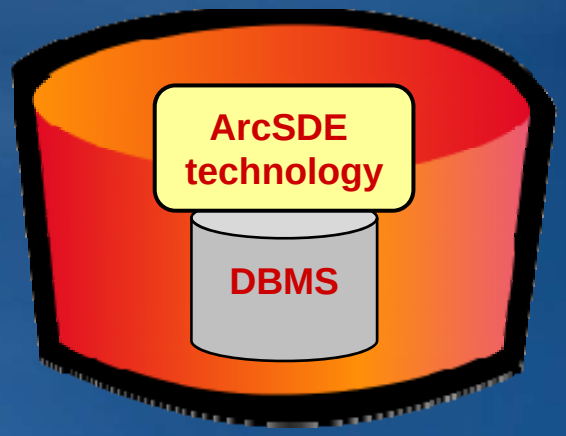
Enterprise geodatabase

Technology stack



Introducing ArcSDE technology

- Spatial extension for DBMSs
  - Storage & management of spatial data & associated attributes
    - Vector data
    - Raster data
  - Fast retrieval & display of spatial data
    - Utilizes spatial indexes
  - Part of the geodatabase data model
  - Enables multi-user editing framework
    - Versioning
- Leverages DBMS functionality
  - Security
  - Backup & recovery
  - Scalability



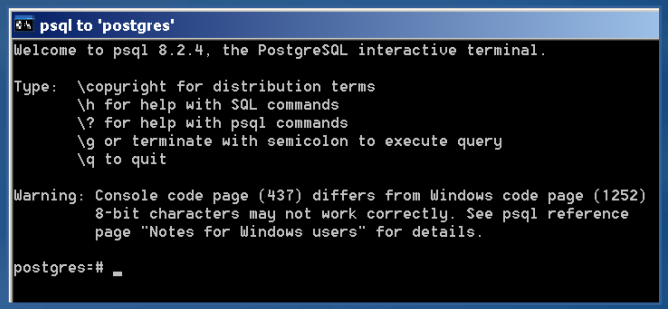
Enterprise Geodatabase

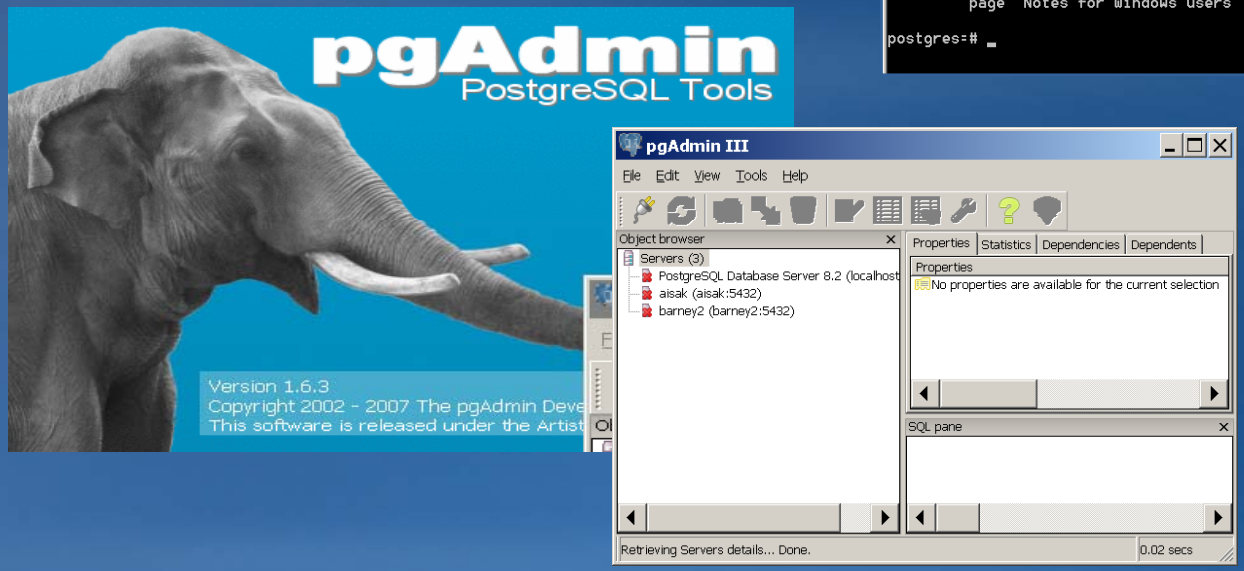
Introducing PostgreSQL

- Open Source DBMS
  - Developed by Online Community
    - http://www.postgresql.org/about/
  - Distributed with BSD license = Free
  - Started as *Ingres* at UC Berkeley
- Conforms to SQL 92/99 standards
- Comparable to leading commercial DBMS platforms
  - Supports complex database features such as UDT, views, table inheritance, stored procedures, extensible index framework, etc.
  - Client library interface available in many languages (C,C++, Java, Perl, Python, Lisp etc.)

PostgreSQL administrator tools

- Many Open Source DBMS management tools available:
  - pgAdmin III → like SQL Server Enterprise Manager
    - Included with ArcGIS Server Enterprise
  - psql → like SQL*Plus
  - Resources:
    - http://pgfoundry.org/



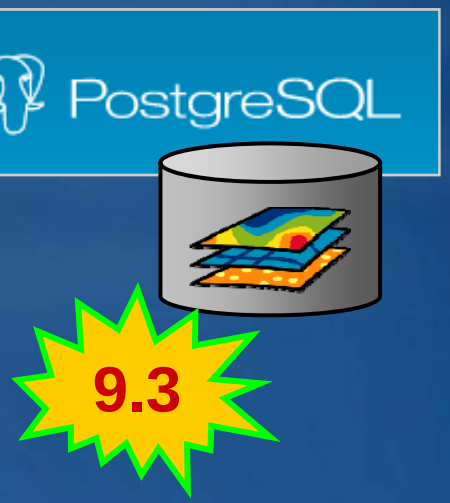


Outline

- Introduce ArcSDE technology for PostgreSQL
- Implementation
  - Enterprise ArcSDE technology for PostgreSQL
  - Spatial types
- PostgreSQL performance – tips and tricks
- Common tasks
- Summary
- Additional Resources

ArcSDE technology for PostgreSQL

- ArcGIS Server Enterprise will support geodatabases on PostgreSQL
  - PostgreSQL v8.3.0 software included
  - Only accessible with 9.3 client
- Supported for
  - Enterprise geodatabases only
  - Not available for Desktop or Workgroup geodatabases
- Operating systems:
  - Windows 2000 server, 2003 server
  - Linux: Red Hat 4 es/as, Suse 10



ArcSDE technology for PostgreSQL

- Single database model
- Two supported spatial types
  1. `ST_GEOMETRY` (ESRI)
  2. `GEOMETRY` (PostGIS)
- No SDEBinary storage for vector data
- Backup /Restore
  - Currently backup entire database only
    - Pg_dump/ pg_restore

ArcSDE technology for PostgreSQL

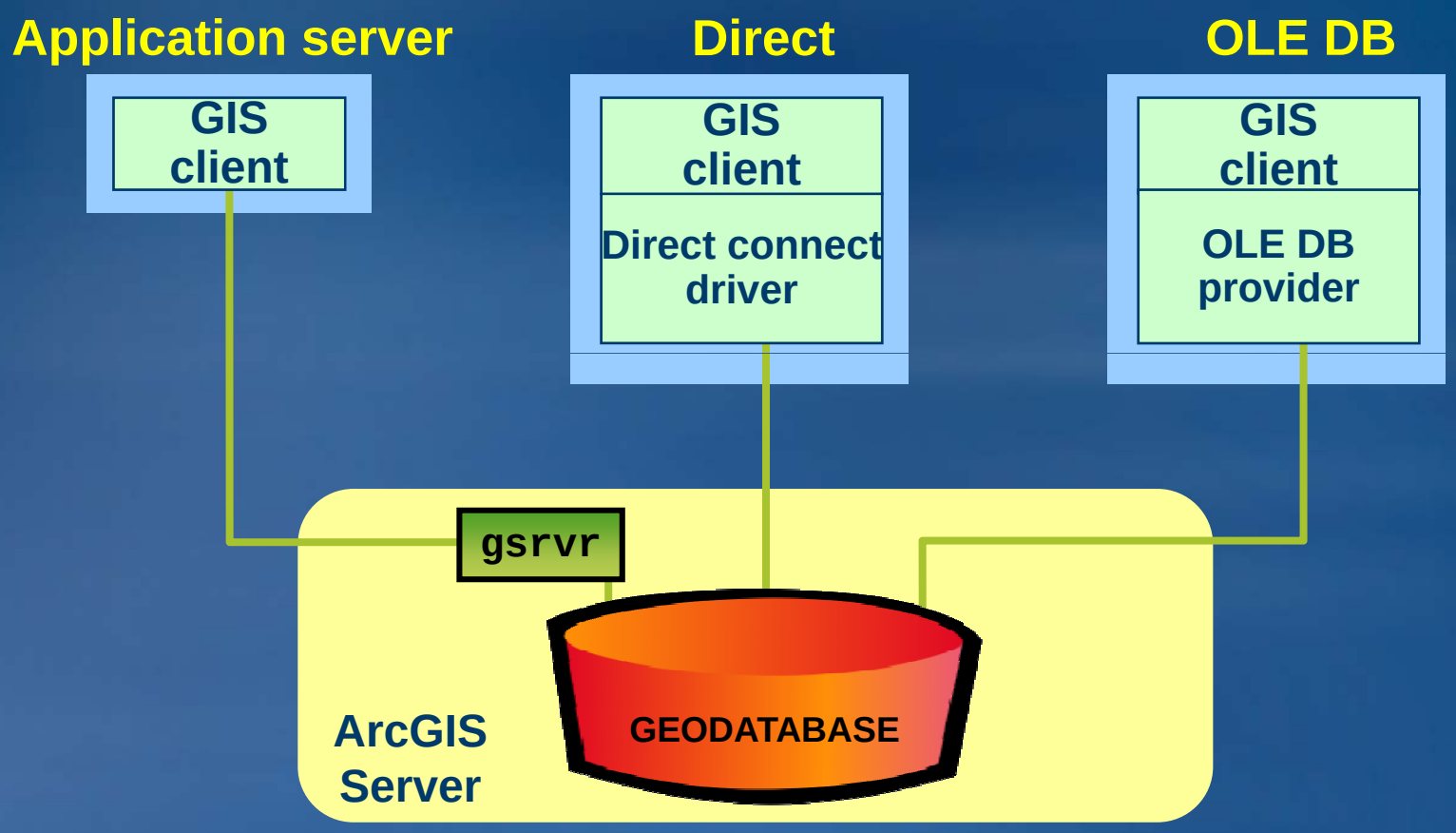
- **ArcSDE administrative tasks**
  - ArcCatalog
  - ArcSDE Command Line
- **List connected user**

```
Z:\pgexe\sdeexe94\bin>sdemon -o info -i 9400 -I users

ArcSDE Instance 9400 Registered Server Tasks on playground at Mon Aug 04 10:43:0
7 2008
-----------------------------------------------------------------------------
S-ID   User     Host:OS                              Started
----- -------- ---------------------------------- -------------------------
49     map      playground:Win32                   Mon Aug 04 10:42:05 2008
50     map      playground:Win32                   Mon Aug 04 10:42:33 2008
```

- **Alter server configuration parameter**

```
Z:\pgexe\sdeexe94\bin>sdeconfig -o alter -p sde -i 9400 -D sde94 -u sde -v MAXBUFSIZE=131072


ArcSDE 9.4  for PostgreSQL Build 113 Thu Jul 10 12:00:54  2008
SDE Server Configuration Tool    Administration Utility
-----------------------------------------------------
Alter SERVER_CONFIG Table. Are you sure? (Y/N): y

        Successfully altered SERVER_CONFIG Table.
```

Connection types to enterprise geodatabases



- No PostgreSQL client installation necessary for direct connect

Spatial types in PostgreSQL

- Two spatial types
  1. `ST_GEOMETRY` (ESRI)
  2. `GEOMETRY` (POSTGIS)
- Both are OGC/ISO compliant
  - Support standard constructor, accessor, & analytical functions
- Full geodatabase functionality supported on both spatial types
  - E.g., versioning, topology, geometric networks, historical archiving, geodatabase replication, etc.
- Both types provide spatial index functionality

What is different between the 2 spatial types?

ST_GEOMETRY	GEOMETRY
• Resides under 'sde' schema	• Resides under 'public' schema
• Consistent implementation across DBMSs (Oracle, Informix, DB2, PostgreSQL)	• Only available in PostgreSQL
• Supports parametric curves, surfaces, & point-id	• Not supported
• Stored as compressed shape (less data transfer over network and no conversion required in geodatabase)	• Stored as Well Known Binary

Outline

- Introduce ArcSDE technology for PostgreSQL
- Implementation
- PostgreSQL performance – tips and tricks
- Common tasks
- Summary
- Additional Resources

PostgreSQL performance – tips and tricks

- Autovaccum, vacuum analyze
  - Vacuum: Permanently removes deleted records
  - Autovacuum: a background process
    - Defined in *postgresql.conf*
  - Analyze: updates index statistics
- Postgres memory allocation
  - Defined in *postgresql.conf*
    - Shared buffers
    - Work memory
    - Effective cache size

```
# - Memory -

shared_buffers = 32MB

#temp_buffers = 8MB
#max_prepared_transactions = 5

# Note:  Increasing max_prepar
# per transaction slot, plus l
#work_mem = 1MB
#maintenance_work_mem = 16MB
#max_stack_depth = 2MB

# - Free Space Map -

max_fsm_pages = 204800

#max_fsm_relations = 1000


# - Kernel Resource Usage -

#max_files_per_process = 1000

shared_preload_libraries = '$l

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0
#vacuum_cost_page_hit = 1
#vacuum_cost_page_miss = 10
#vacuum_cost_page_dirty = 20
#vacuum_cost_limit = 200
```

PostgreSQL performance – tips and tricks

- **Log directory location**

  <Postgresql_location>\data\pg_log

- **Log settings to enable performance monitoring**

  (Defined in *postgresql.conf*)

  - **log_min_duration_statement = 25**
  - **log_duration = on**
  - **log_line_prefix = '%t [%p]: [%l-1] '**
  - **log_statement = 'all'**
  - **stats_start_collector = on**

- **Use PGFouine to process performance log files**

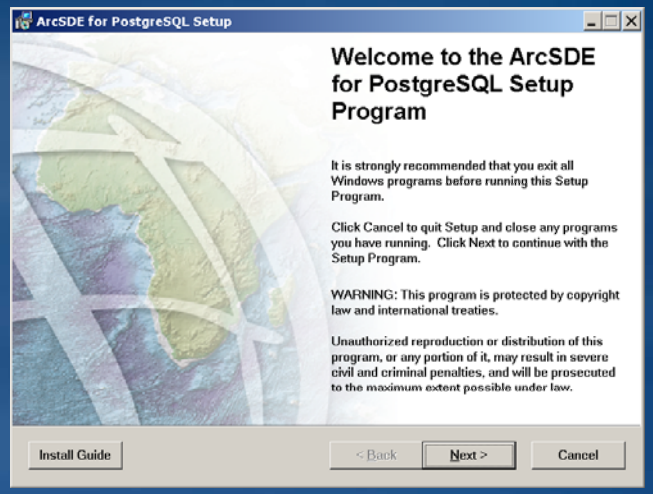
  pgfouine.php -file pgsql.log -top 40 -report
  queries.html=overall,bytype,slowest,n-mosttime,n-mostfrequent
  -logtype stderr

Outline

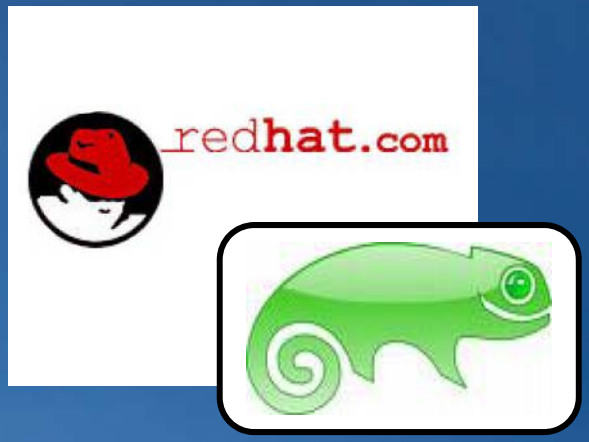
- Introduce ArcSDE technology for PostgreSQL
- Implementation
- PostgreSQL performance – tips and tricks
- Common tasks
  - Installation
  - Creating users and assigning privileges
  - Connecting to a PostgreSQL database
  - Data loading
  - Data editing
  - Registering spatial data with geodatabase
  - Tips: psql commands
- Summary
- Additional Resources

Installation: Included on software DVD

- **Windows**: 2000 server & 2003 server
  - PostgreSQL 8.3.0
  - Post installation for PostgreSQL
  - ArcSDE
  - Post installation for ArcSDE



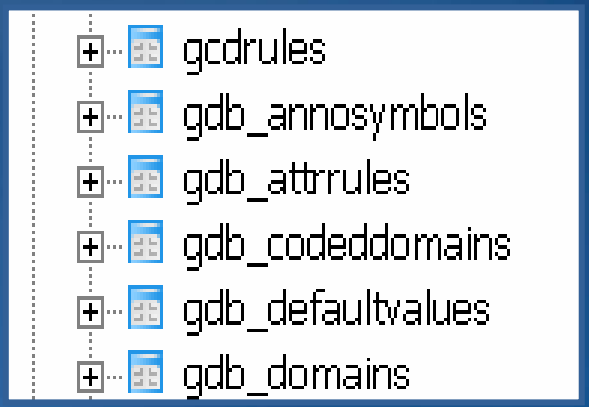
- **Linux**: Red Hat Linux 4 & Suse10
  - *Create_pgdb.sde* (Red Hat only)
  - *Setup_pgdb.sde*
  - Install
  - Manual post installation



ArcSDE for PostgreSQL installation

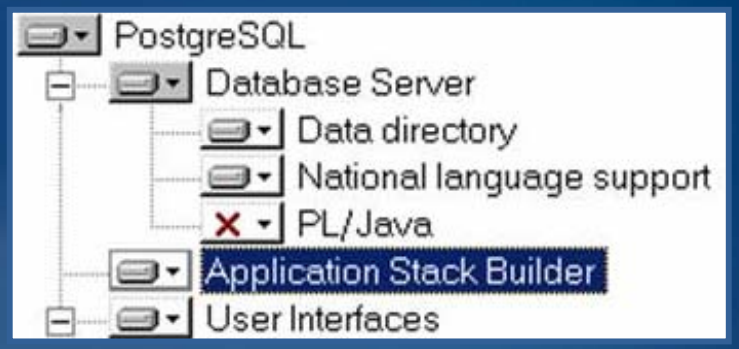
- ArcSDE can be installed on
  - Same machine as DBMS, or
  - Remotely

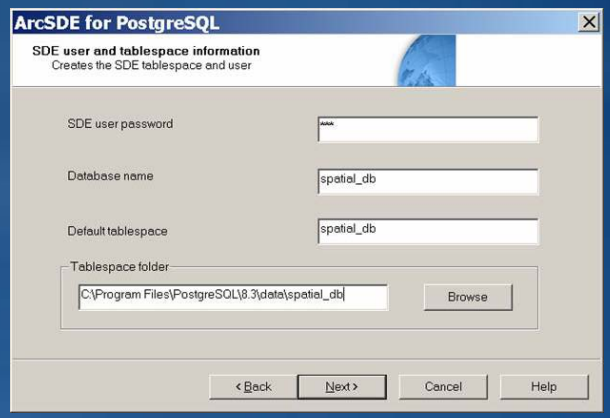
- What does installation do?
  - Copies PostgreSQL software
  - Copies ArcSde Software
  - Creates PostgreSQL database (Optional)
  - Creates 'SDE' role and SCHEMA
  - Creates ST_GEOMETRY type
  - Creates geodatabase repository



Installation: ArcSDE & PostGIS in one database

- **Install PostgreSQL**
  - Under installation options choose: Application Stack Builder (ASB)
- **Install PostGIS**
  - ASB will connect to the internet & allow for PostGIS download
  - Choose PostGIS v1.3.2
  - Install PostGIS & create a database
- **Install ArcSDE**
  - Execute the ArcSDE installation wizard
- **Post Installation**
  - Execute the Post installation wizard
  - Use database with PostGIS installed



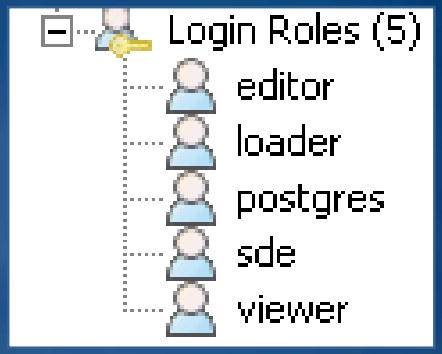


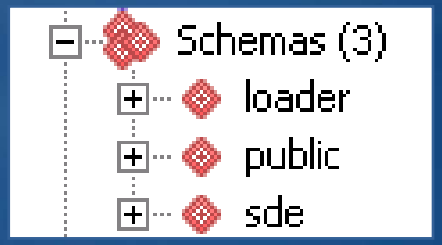
- **Manual Installation: PostGIS to database**
  - psql –d *yourdatabase* -f lwpostgis.sql

- **ESRI Knowledge Base article: 35128**

Creating users and assigning privileges

- PostgreSQL has:
  - Roles
    - Login roles: database accounts
    - Group roles: database roles
  - Schemas
    - Data logically resides in a schema
    - For data editors: login role name = schema name
    - Granted “usage” to PUBLIC/USER





- Types of users
  - Data Editors: Select, Insert, delete and update privileges
  - Data Viewers : Select privileges

Creating users and assigning privileges

- **For data editors:**

```
CREATE ROLE user1 LOGIN ENCRYPTED PASSWORD 'user1' CREATEDB;
CREATE SCHEMA user1 AUTHORIZATION user1;
GRANT SELECT, INSERT, UPDATE, DELETE ON public.Geometry_columns
  to user1; (PostGIS only)
```

- **For data viewers:**

```
CREATE ROLE user2 LOGIN ENCRYPTED PASSWORD 'user2';
GRANT USAGE ON SCHEMA user1 TO user2;
```

- **SQL scripts are provided as part of ArcSDE for PostgreSQL installation:**

C:\Program Files\ArcGIS\ArcSDE\pgexe\tools\postgres

Privileges: schema privileges

- Common oversight when setting up privileges
- Scenario:
  - *User1* owns a feature class named “lakes”
  - *User1* gives *User2* read/write privileges to “lakes”
  - Usage privilege has not been granted to user2 on *User1* schema
- Solution: grant Usage to user2 for *User1* schema

Connecting to a PostgreSQL database

- After installing PostgreSQL **must enable** connectivity to cluster:
  - *Postgresql.conf*
  - *Pg_hba.conf*
- Otherwise will get:
  - "Bad login user" error
  - "Server not accepting connections" error
- Reload the server if you modify these files.

postgresql.conf - Notepad

File Edit Format View Help

```
#------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------

# - Connection Settings -

listen_addresses = '*'
```

pg_hba.conf - Notepad

File Edit Format View Help

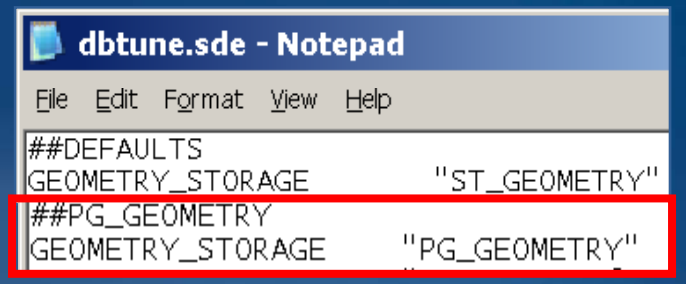
```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHO

# IPv4 local connections:
host    all         all         127.0.0.1/32          md5
host    all         all         10.0.0.0/8            md5
# IPv6 local connections:
#host    all         all         ::1/128               md5
```

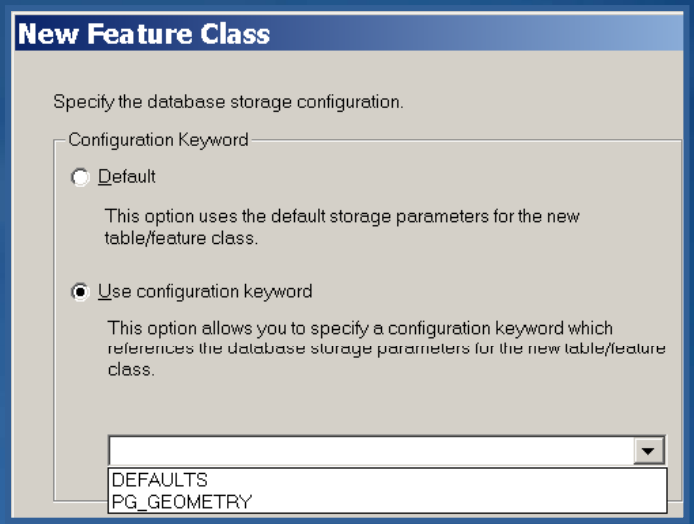
Loading data into the enterprise geodatabase

- **Methods**
  - Create new data
  - Import existing data
  - Append into existing feature class
- **Tools**
  - ArcGIS
    - Append Tool
    - Simple Data Loader
    - Object Loader
  - Manually
    - ArcSDE administration commands
    - SQL API in PostgreSQL

Controlling storage in the enterprise geodatabase

- Use configuration keyword to control object placement
  - Stored in `sde.sde_db_tune`
  - Specify during loading
- DBTUNE parameters set:
  - 18 default keywords
    - Tablespace for index
    - Tablespace for table
    - Spatial type(s)
  - Can create additional keywords
- Default geometry storage: `ST_GEOMETRY`





Creating spatial data in PostgreSQL

- Creating a table with a spatial attribute

```
// ST_GEOMETRY type
   CREATE TABLE john.blocks_st
     (objectid INTEGER NOT NULL,
      block    VARCHAR(24),
      shape    st_geometry);


// POSTGIS GEOMETRY Type
  // Create table
   registration
   CREATE TABLE john.blocks_pg
     (objectid INTEGER NOT NULL,
      block VARCHAR(24));

  // Add spatial column
   Select AddGeometryColumn(‘john',
       ‘blocks_pg', ‘shape',1 ,‘GEOMETRY',2);
```

Working with spatial data in PostgreSQL

- Inserting a row with a spatial attribute

```
INSERT INTO john.blocks_st VALUES (1,‘block',
st_geometry('polygon((52 28,58 28,58 23,52 23, 52 28))',1));

INSERT INTO john.blocks_st VALUES (2,‘block',
st_geometry('polygon ((12 28,18 28,18 23,12 23,12 28))',1));
```

- Creating the spatial index

```
// ST_GEOMETRY TYPE
CREATE INDEX blockssp_idx ON blocks_st USING gist(shape);

// GEOMETRY TYPE
CREATE INDEX blockssp_idx ON blocks_pg USING gist(shape);
```

ST_Geometry type functions

- **Relational functions**
  - ST_Contains(), ST_Within(), ST_Intersects(), ST_Overlaps(), ST_Touches(), ST_Crosses(), ST_Equals(), ST_Disjoint(), …
- **Geometric functions**
  - Constructors: ST_Geometry(), ST_Point(), ST_LineString(), ST_Polygon(), ST_MultiPoint(), ST_MultiLineString(), ST_GeomFromWKB(), ST_GeomFromShape(), …
  - Accessors: ST_AsText(), ST_AsBinary(), ST_AsShape(), ST_AsSDEComp(), …
  - Analysis: ST_MinX(), ST_MaxM(), ST_Distance(), ST_GeometryType(), ST_SRID(), ST_Boundary(), ST_Buffer(), ST_Intersection(), ST_Difference (), ST_IsClosed(), ST_Centroid(), …
- **Misc. functions**
  - ST_Geometry_Version(), ST_Geometry_Release(), ST_MBR(), ST_register_spatial_column(), ST_unregister_spatial_column(), ST_isregistered_spatial_column(), …

Registering spatial data with geodatabase

- **Creating table with spatial type**

```
create table sde.test2 (id integer, shape sde.st_geometry);

insert into sde.test2 values(1, sde.st_multipoint('multipoint(10 30, 10 30)',0));
```

- **Registering with ArcSDE**

```
C:\>sdelayer -o register -l sde.test2,shape -C ID,SDE -e 1 -t PG_GEOMETRY -i 5153
-D production -u map -p map -x 400,400,1000000000
```

- **Register with geodatabase**
- **Register as versioned (optional)**
- **Grant privileges to other users**

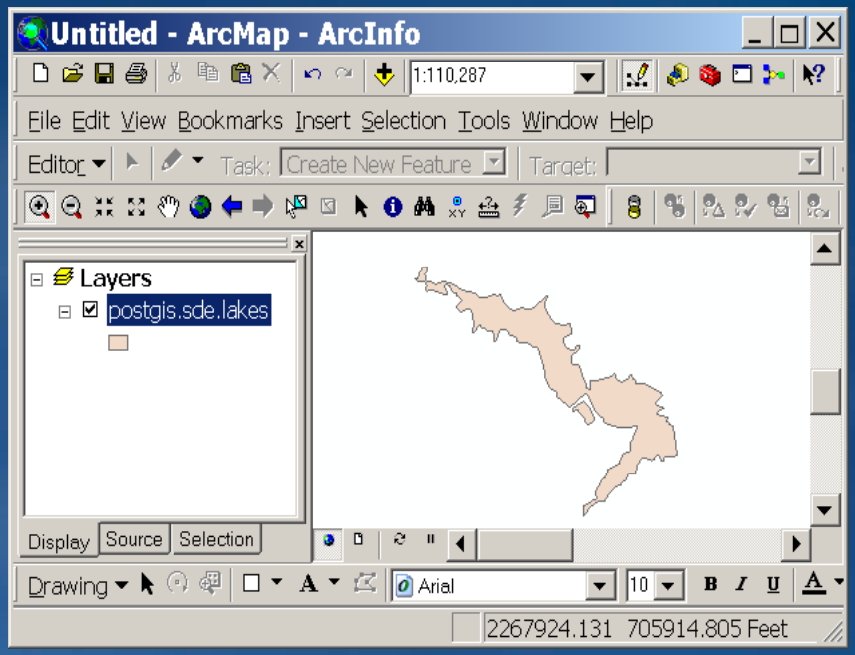
Method applies to both spatial types

Registering existing PostGIS data with geodatabase

- Enables access to geodatabase functionality
  1. Ensure the PostgreSQL version is supported by ArcSDE: v8.3.0
  2. Ensure the PostGIS version is supported by ArcSDE: v1.3.2
  3. Register the PostGIS layers with ArcSDE
  4. Register the PostGIS layers with geodatabase

Data editing options

- Vector data can be edited:
  - ArcGIS client
    - Accessing spatial data in the geodatabase
    - Non versioned editing
    - Versioning
  - SQL API
    - Accessing spatial data in the DBMS
    - Inserting & updating geometry
    - Do not edit data that participates in geodatabase functionality (i.e. topology, networks, terrain etc.)



Tips: Psql commands (shortcuts)

- \c[onnect] [DBNAME|- USER|- HOST|- PORT|-]
- \d [NAME] describe table, index, sequence, or view
- \db [PATTERN] list tablespaces
- \df [PATTERN] list functions
- \dD [PATTERN] list domains
- \dg [PATTERN] list groups
- \dn [PATTERN] list schemas
- \du [PATTERN] list users
- \l list all databases
- \H toggle HTML output mode
- \q quit psql
- \? Help
- \h [NAME] help on syntax of SQL commands

Summary

- Introduce ArcSDE technology for PostgreSQL
- Implementation
- PostgreSQL DBMS administration
- Common tasks

Additional Resources

- PostgreSQL Resources:
  - User forums
  - Documentation on line
  - Help in PgAdminIII
- ArcSDE Resources:
  - Podcast
  - Knowledge Base Article 35128. How to install PostgreSQL 8.3.0, ArcSDE 9.3 and PostGIS 1.3.2 on Windows
  - ArcGIS help