2010 Esri 中国区域用户大会

高效的使用SQL操作空间数据库

刘锋

主要内容

- 为什么要使用SQL
- 空间数据类型
- 如何使用SQL操作GEODATABASE

为什么要使用SQL

- 快?

- 无所不能
  - 能做坐标转换吗?
  - 维护拓扑和网络结构吗?
  - 添加进去的记录ArcMap中一定能正常显示吗?

- 适合什么场景?

主要内容

- 为什么要使用SQL
- 空间数据类型
- 如何使用SQL操作GEODATABASE

空间存储类型是？

C:Struct, C++,C#,Java,Flex:Class

属性:

- DBMS Datatypes

函数和操作子

- Constructor
- Accessor
- Relational
- Geometry

空间索引

为什么使用空间类型

效率：

- 空间数据和函数都存储在DBMS中
- 空间操作子包括在空间类型中
- 应用程序可以直接操作空间数据类型

访问方法：

- C, C++, C#, Java, OLEDB
- SQL Extensions

GeodatabaseV空间存储类型

SQL Server

- GEOMETRY
- GEOGRAPHY

Oracle

- ST_GEOMETRY
- SDO_GEOMETRY

DB2/Informix

- ST_GEOMETRY

PostgreSQL

- ST_GEOMETRY
- PostGIS
- GEOMETRY

GeodatabaseR空间存储类型

SQL Server

ST_RASTER

Oracle

ST_RASTER

SDO_GEORASTER

PostgreSQL

ST_RASTER

空间类型的配置

Listener.ora

Path to st_shapelib.dll,libst_raster_ora.dll

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME = d:\oracle\product\10.2.0\db_1)
      (SID_NAME = PLSExtProc)
      (PROGRAM = extproc)
      (ENVS = "EXTPROC_DLLS=d:\program files\arcgis\arcsde\ora10gexe\bin\st_shapelib.dll;
           d:\program files\arcgis\arcsde\ora10gexe\bin\libst_raster_ora.dll")
    )
    (SID_DESC =
      (GLOBAL_DBNAME = ceshi)
      (ORACLE_HOME = d:\oracle\product\10.2.0\db_1)
      (SID_NAME = ceshi)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 1f)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
    )
  )
```

空间类型的配置

tnsnames.ora

```
EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(Key = EXTPROC0))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )
```

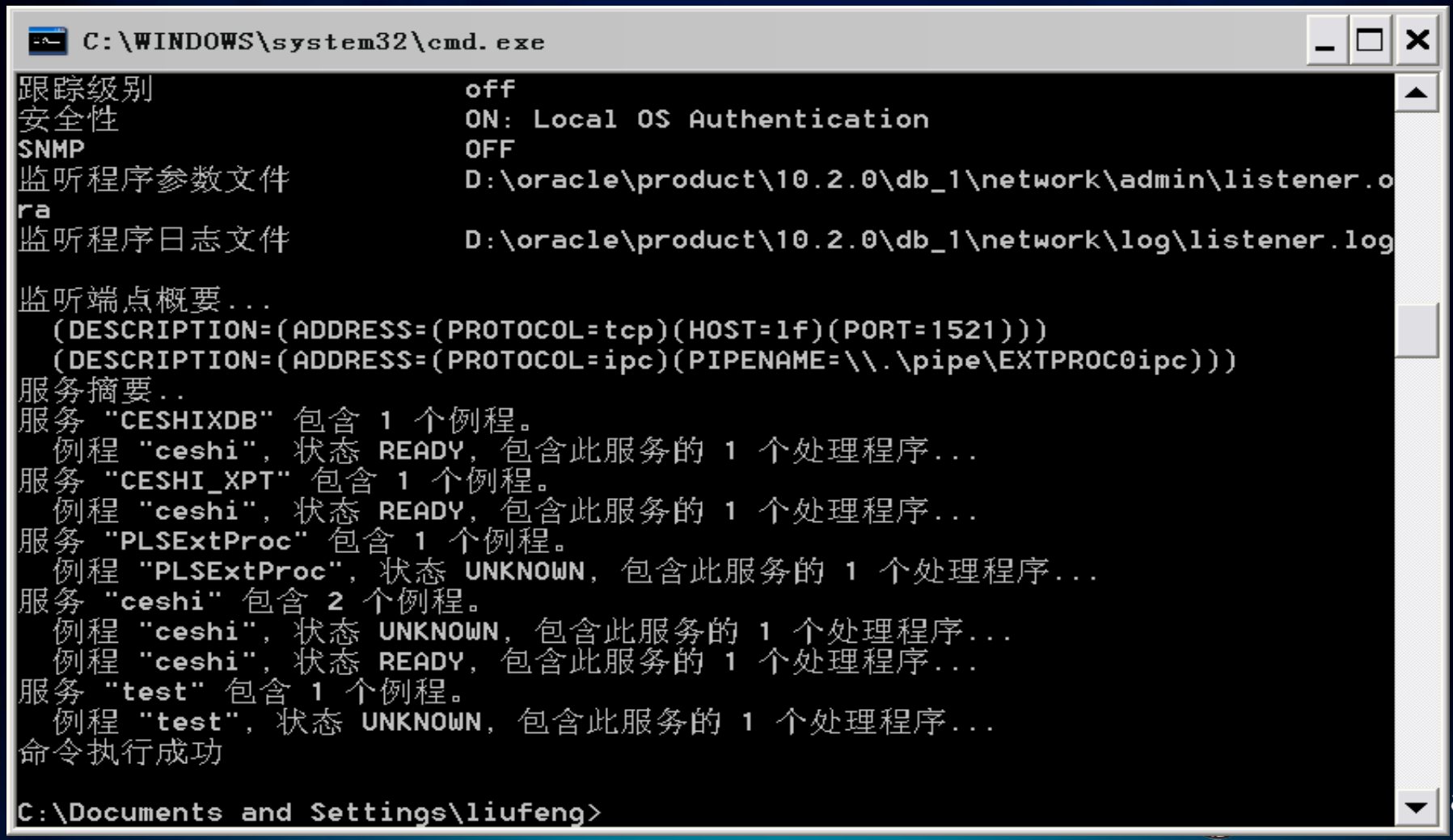
Key value must match listener.ora file

ST_RASTER

```
sdesetup
-o install_st_raster
            -d {ORACLE10G|ORACLE11G|SQLSERVER|POSTGRESQL}
          [-H <sde_directory>] [-u <DB_Admin_user>] [-p <DB_Admin_password>]
          [-D <database>] [-s <datasource>] [-N] [-l <key>] [-q]
```

NOTE

Lsnrctl status



ST_GEOMETRY

- Geometry Type
- Extent (MBR)
- Characteristics
  - Area, Len
- Spatial Reference
- Coordinates

Name	Type
ENTITY	NUMBER(38)
NUMPTS	NUMBER(38)
MINX	FLOAT(64)
MINY	FLOAT(64)
MAXX	FLOAT(64)
MAXY	FLOAT(64)
MINZ	FLOAT(64)
MAXZ	FLOAT(64)
MINM	FLOAT(64)
MAXM	FLOAT(64)
AREA	FLOAT(64)
LEN	FLOAT(64)
SRID	NUMBER(38)
POINTS	BLOB

空间索引

Rtree or Grid

Modeled as separate spatial index table

Associated to geometry type and operators

Enabled to the optimizer using statistics

函数

Constructors

```
ST_POINT (X, Y, SRID)
ST_POINT (X, Y, Z, M, SRID)
ST_POINT (“X Y”, SRID)
```

```
CREATE TABLE accidents (ID number, shape ST_GEOMETRY);
INSERT INTO accidents (1,ST_POINT(10,10,1));
```

函数

Accessor

- Well-Known Text

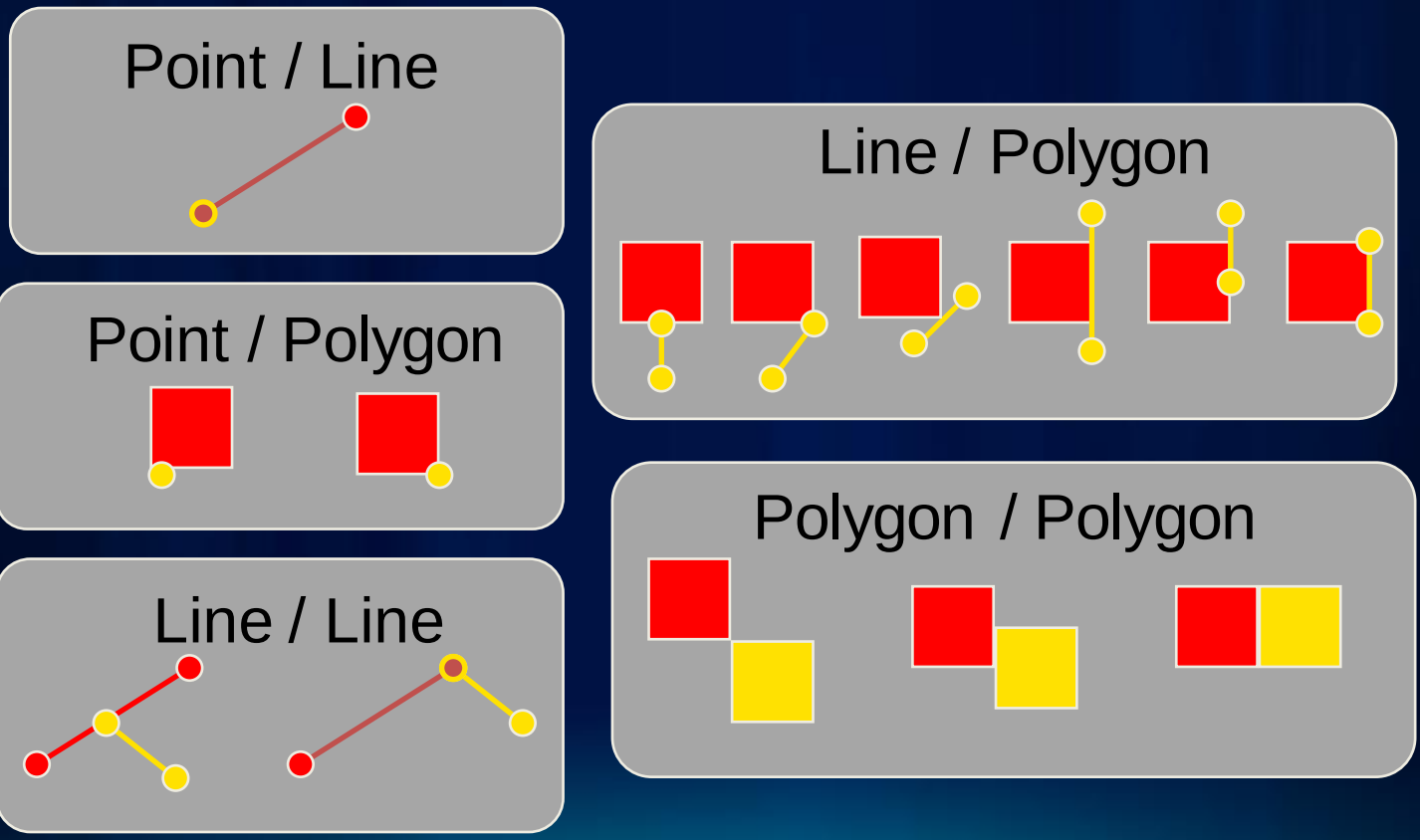
```
SELECT ST_ASTEXT(SHAPE)FROM districts

ST_ASTEXT(ST_GEOMFROMWKB(ST_ASBINARY(SHAPE),0))
----------------------------------------
POLYGON ((10.0 10.0, 50.0 10.0, 50.0 50.0, 10.0
          50.0, 10.0 10.0))
```

关系操作符(Relationship)

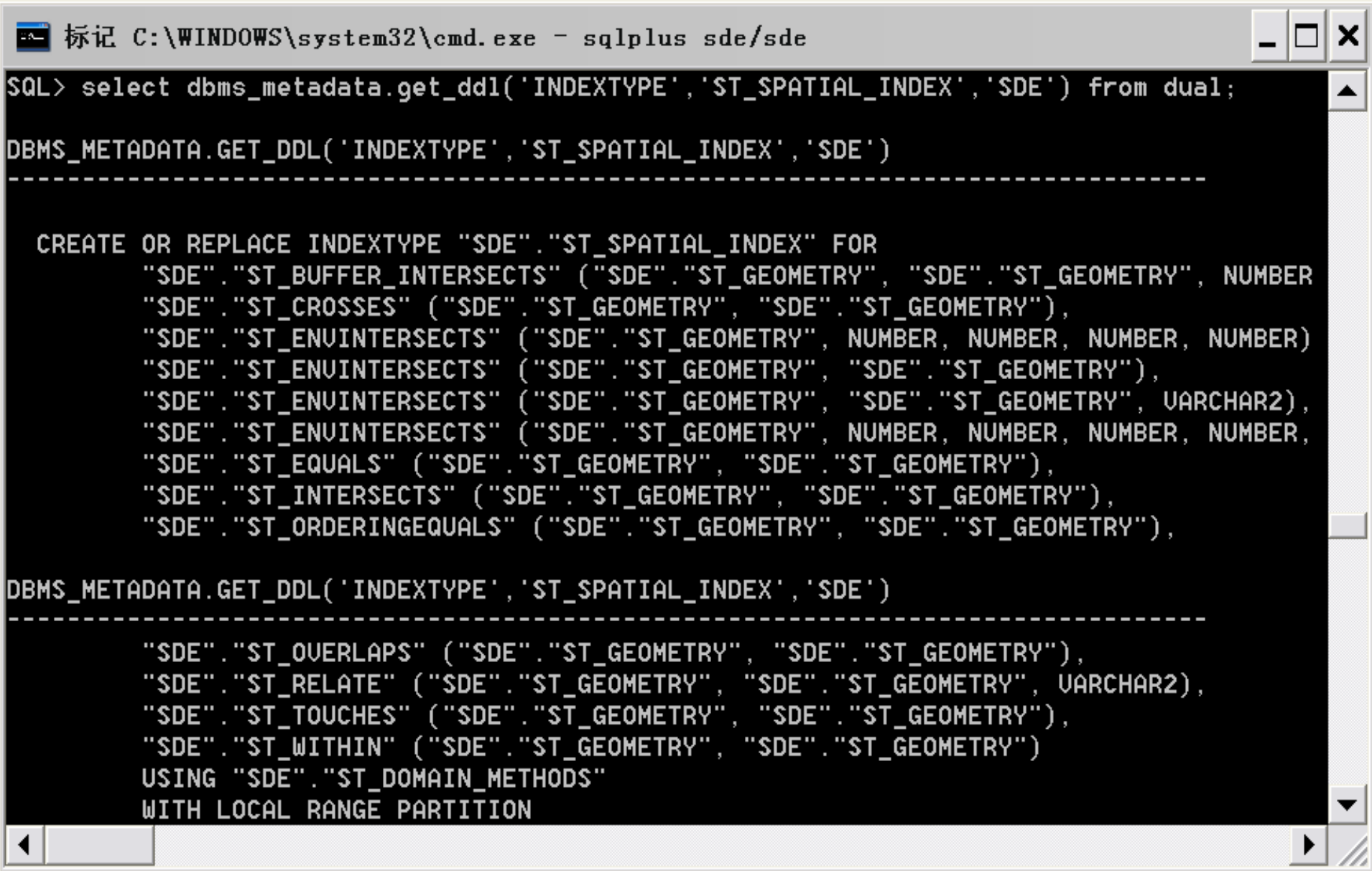
- 检测两个图形之间的关系
  - Input (GEOMETRY_A, GEOMETRY_B)
  - Returns TRUE / 1 or FALSE / 0
- 例子
  - ST_INTERSECTS
  - ST_TOUCHES
  - ST_CONTAINS

关系操作符

- ST_TOUCHES (Geometry_A, Geometry_B) = 0 or 1



空间索引与操作符的关系



空间索引与操作符的关系

Comparison function	Can use spatial grid index
EnvelopesIntersect	Yes
ST_Contains	Yes
ST_Crosses	Yes
ST_Distance	Yes
ST_EnvIntersects	Yes
ST_Equals	Yes
ST_Intersects	Yes
ST_MBRIntersects	Yes
ST_Overlaps	Yes
ST_Touches	Yes
ST_Within	Yes

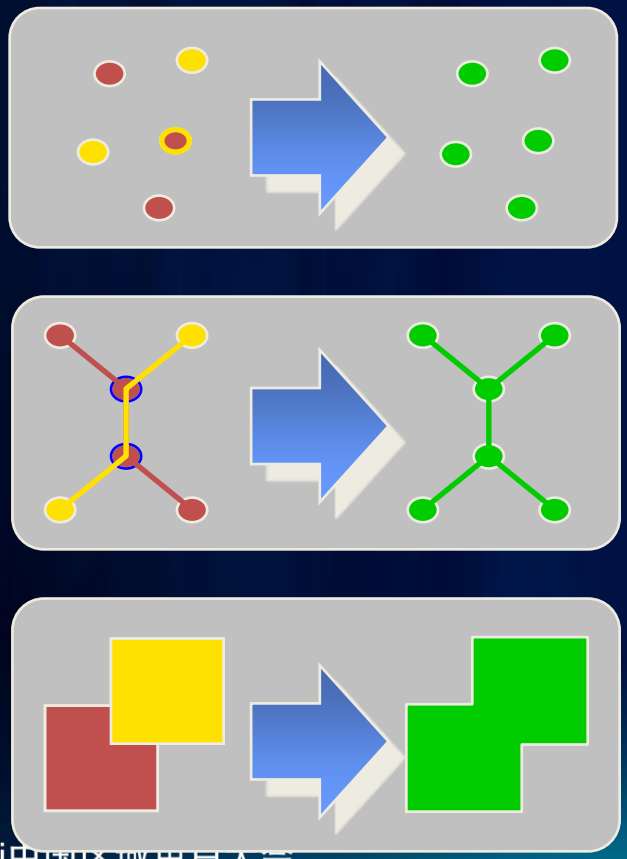
Geometry操作符

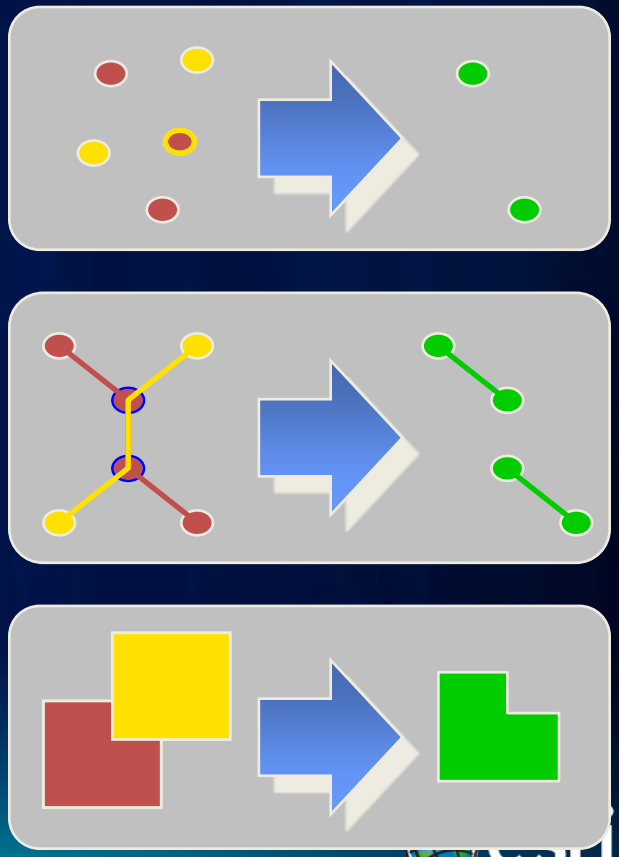
- 创建一个新的Geometry
  - Input: (GEOMETRY_A, *args*…)
  - Returns: (GEOMETRY)

- 例子
  - ST_BUFFER
  - ST_UNION
  - ST_DIFFERENCE

Geometry操作符

- ST_Union (geom1,geom2)
- ST_Difference (geom1,geom2)





演示

ST_RASTER

- Type
- Extent
- Properties
- BandInfo
- PixelData

```
Name                 Type
-------------        ----------
RASTERID             NUMBER
TYPE                 NUMBER
EXTENT               SDE.SE_EXTENT
PROPERTIES           SDE.SE_RASTERPROPERTY
                     _LIST
RASTERBAND_LIST      SDE.ST_RASTERBAND_LIST

DATA                 SDE.ST_PIXELDATA
```

函数

Constructors

```
ST_RASTER (filename)
ST_RASTER (filename, options)
ST_RASTER (data)
ST_RASTER (data, options)
```

```
CREATE TABLE NOVA (ID number, image ST_RASTER);
INSERT INTO NOVA (1,sde.st_raster('nova.tif',
  'compression=lz77'));
```

函数

```
getCompressionType
getInterleaveType
GetInterpolationType
getPixelData
getPixelType
getPyramidLevel
getValue
getVersion
hasColormap
hasStats
.....
```

函数

Editor:

```
Crop
deleteColormap
deleteStats
Subset
setColormap
Mosaic
export
……
```

主要内容

- 为什么要使用SQL
- 空间数据类型
- 如何使用SQL操作GEODATABASE

SQL编辑FeatureClass

- 编辑简单对象
  - 点，线，面（single or multipart)
  - 不包含空间数据的行为
    - 不参与网络，不参与拓扑
- 非版本
  - 直接应用到B表中(没有Delta table)
- 版本
  - 需要预定义一个操作流程

编辑非多版本FeatureClass

- 添加记录(objectid)
  - 通过sequence 或者 procedure获得
- 可以充分利用数据的功能
  - Unique indexes, constraints, referential integrity, default values, triggers

获得row_id值

```
//Oracle

  SQL> SELECT registration_id FROM sde.table_registry
       WHERE owner = 'TOMB' AND table_name = 'PARCELS';

  SQL> SELECT sde.version_user_ddl.next_row_id('TOMB', 114)
      FROM dual;


//SQL*Server

  SELECT registration_id FROM sde.sde_table_registry
  WHERE owner = 'TOMB' AND table_name = 'PARCELS'

  DECLARE @id AS INTEGER
  DECLARE @num_ids AS INTEGER
  exec sde.i114_get_ids 2, 1, @id OUTPUT, @num_ids OUTPUT
```

SQL编辑版本FeatureClass

- 使用SQL访问多版本视图
  - 可以修改空间数据
  - 详细文档在ArcGIS Desktop Help
- 流程
  - 创建一个多版本视图
  - 创建一个新的版本
  - 在该版本中进行编辑
  - 使用ArcMap reconcile/post

SQL编辑版本FeatureClass

创建一个多版本view

```
sdetable -o create_mv_view -T <view_name> -t <table_name>
  [-i <service>] [-s <server_name>] [-D <database>]
  -u <DB_User_name> [-p <DB_User_password>] [-N] [-q]

sdetable -o create_mv_view -T parcels_mv -t parcels -i 5151
  -s sde -u sde -N
```

SQL编辑版本FeatureClass

创建一个新版本

```
db2 CALL sde.create_version ('SDE.DEFAULT', 'versionname',
 1, 1, 'multiversioned view edit test1 version', ?, ?)
```

设定当前编辑的版本

```
db2 CALL sde.set_current_version('versionname',?,?)
```

SQL编辑版本FeatureClass

开始编辑

```
db2 CALL sde.edit_version('versionname',1,?,?)
```

SQL进行相关的编辑

```
db2 Insert into test values (……)
Commit;
db2 Delete from test where ……
Savepoint a;
db2 Update test set shape=…….
Rollback to a;
……
```

SQL编辑版本FeatureClass

结束编辑

```
db2 CALL sde.edit_version('versionname',2,?,?)
```

利用ArcGIS产品调和并合并

删除版本

```
db2 CALL sde.delete_version('versionname',?,?)
```

SQL编辑版本FeatureClass

```
sdeversion -o compress
```

```
sdetable -o delete_mv_view -t <table_name>
[-i <service>] [-s <server_name>] [-D <database>]
 [-u <DB_User_name>] [-p <DB_User_password>] [-N] [-q]

sdetable -o delete_mv_view  -t parcels -i 5151
  -s sde -u sde -N
```

SQL编辑版本FeatureClass

注意:

- 创建多版本视图前必须先注册版本
- 添加记录的时候不需要自己确定ID
- 不要使用SQL来更新已有记录的ID
- 尽量不要在Default版本下编辑

演示

SQL操作栅格数据

- 步骤
- 安装ST_RASTER

```
SELECT sde.st_raster_util.describe()
FROM dual;

SDE.ST_RASTER_UTIL.DESCRIBE()

No data found.
```

```
SELECT sde.st_raster_util.describe()
FROM dual;

ERROR:
ORA-28575: unable to open RPC connection to external proced
ORA-06512: at "SDE.ST_RASTER_UTIL", line 190
ORA-06512: at "SDE_ST_RASTER_UTIL", line 377
```

SQL操作栅格数据

创建一张包含ST_GEOMETRY的表

```
CREATE TABLE urban_area (sde.st_raster raster, name varchar2(255))
```

初始化raster列

```
BEGIN SDE.ST_RASTER_UTIL.INITALIZE ('urban_area','raster',4326,'defaults');
END; /
```

SQL操作栅格数据

- 插入影像到该表中

```
INSERT INTO URBAN_AREA
(SDE.ST_RASTER('C:\milwakee.tif'),'MILWAUKEE');
```

- ArcCatalog注册成geodatabase

演示

小结

- VECTOR
- RASTER
- Geo SQL使用方法
- 效率

参考资料

- ArcGIS Desktop Help
- Oracle Official Document
- DB2 Official document

下一个讲座

- 《海量空间数据库的实施策略》
  吴泳锋，刘锋

谢 谢！