



Transactional Updates to Enterprise GIS data sets

Presented by:
Kelly Ratchinsky
PBC Countywide GIS Coordinator

The Enterprise Environment

- Countywide GIS
 - Centralized GIS Data Repository
- Departmental GIS
 - De-centralized GIS Data Maintenance





GIS Layer Ownership

- Each Department maintains their GIS layers
- Updates are provided to Countywide GIS for posting to Enterprise GIS
- Manual posting imposes time constraint
- Automated method is instant



PZB Example

- Owners of too many layers to count
- City Boundary - Annexations
- Changes to a City Boundary affect FLU, Zoning, etc.
- Changes must be reflected in ePZB applications as soon as possible
 - ePZB applications use Enterprise GIS Data repository



Technical Considerations

- SDE Versioned GIS Layer
- Oracle Spatial GIS data model
- Oracle Triggers & Procedures



SDE Versioned GIS Layer

- SDE Layer (Oracle Spatial)
 - GIS Layer table – Geometry & Attribute table, Oracle stores spatial indexes
 - SDOBinary will have Layer, F & S tables which store the Layer attributes, Feature geometry and Spatial indexes
- Versioned SDE Layer has 2 more tables
 - A table – Adds table (identical to GIS Layer table)
 - D table – Deletes table (stores deleted objectid)



Oracle Spatial GIS Data Model

- Oracle Stores data in sdo_geometry data type
 - ESRI stores as BLOB in F table
- Oracle creates Spatial Indexes
 - ESRI creates S table for spatial index
- Use Oracle SQL operators, packages and procedures
- Use SDE Versioning process to initiate change to Enterprise GIS data repository



Oracle Triggers & Procedures

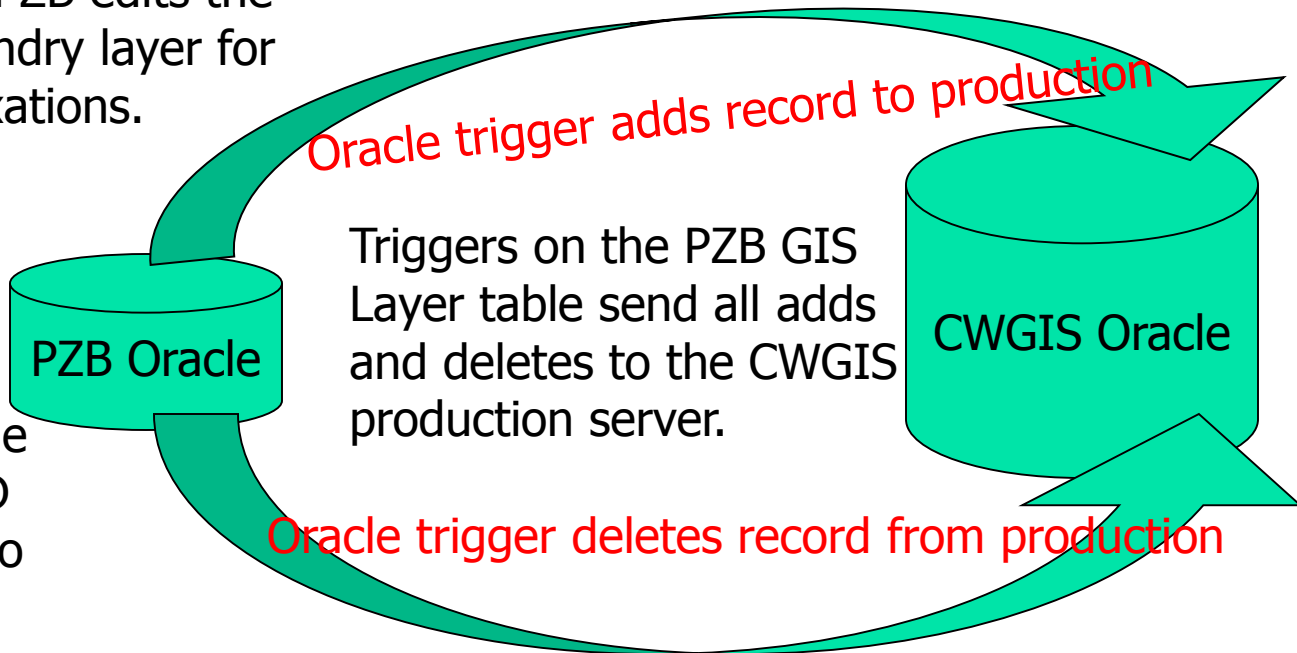
- Maintenance Database
 - Create [dblink](#) to production database
 - Create Oracle [Synonym](#) to update_triggers package
 - Create [Oracle triggers](#) on GIS Layer table
- Production Database
 - Create [update_triggers](#) package
 - Create [dblink](#) to maintenance database

GIS Layer Edits



Sharon at PZB edits the the City_Bndry layer for new annexations.

Matt runs the weekly SDE Compress. This Initiates the adding & deleting of data to the GIS Layer table on PZB Server. The A & D tables are compressed into the GIS Layer table.





Questions?

Kelly Ratchinsky

kratchin@pbcgov.com



Maintenance Server

- Create dblink to production database
 - Create database link pzb_egis connect to pzb identified by password using 'pzb/password@egis'
- Create Oracle Synonym to update_triggers package
 - Create synonym update_triggers for pzb.update_triggers@pzb_egis

[Go Back](#)



Maintenance Server (cont.)

- Delete Trigger
 - CREATE OR REPLACE TRIGGER "PAO".PARCEL_PY_DELETES
 - BEFORE DELETE
 - ON PAO.PARCEL_PY
 - REFERENCING NEW AS NEW OLD AS OLD
 - FOR EACH ROW
 - DECLARE
 - BEGIN
 - update_triggers.delete_trigger(:old.objectid,'PARCEL_PY');
 - END;
- ADD Trigger
 - CREATE OR REPLACE TRIGGER "PAO".PARCEL_PY_ADDS
 - AFTER INSERT
 - ON PAO.PARCEL_PY
 - REFERENCING NEW AS NEW OLD AS OLD
 - FOR EACH ROW
 - DECLARE
 - BEGIN
 - update_triggers.add_trigger(:new.OBJECTID,
 - 'PARCEL_PY','OBJECTID,PCN,CTY,RNG,TWP,SEC,SUBD,BLK,LOT,CREATEDATE,GEOMETRY');
 - END;



Production Server

- Create dblink to maintenance database
 - Create database link egis_pzb connect to pzb identified by password using `'pzb/password@pzbgisprod'`

[Go Back](#)



Production Server

- **Create update_triggers package**

```
CREATE OR REPLACE PACKAGE UPDATE_TRIGGERS as
    PROCEDURE DELETE_TRIGGER(i_objectid IN number,i_table IN varchar2);
    PROCEDURE ADD_TRIGGER(i_objectid IN number,i_table IN varchar2,i_flds IN varchar2);
END UPDATE_TRIGGERS;
```

```
CREATE OR REPLACE PACKAGE BODY UPDATE_TRIGGERS as
PROCEDURE DELETE_TRIGGER(i_objectid IN number,i_table IN varchar2)
AS
mySQL varchar2(2000);
BEGIN
    mySQL := 'DELETE FROM '||i_table||' WHERE OBJECTID = '||i_objectid;
    execute immediate mySQL;
```

```
END DELETE_TRIGGER;
```

```
PROCEDURE ADD_TRIGGER(i_objectid IN number,i_table IN varchar2,i_flds IN varchar2)
```

```
AS
mySQL varchar2(2000);
BEGIN
    mySQL := 'INSERT INTO '||i_table||' ('||i_flds||') SELECT '||i_flds||' FROM PAO.'||i_table||'@ODMDEV WHERE OBJECTID = '||i_objectid;
    execute immediate mySQL;
    mySQL := 'DELETE FROM '||i_table||' where geometry is null';
    execute immediate mySQL;
```

```
END ADD_TRIGGER;
```

```
END;
```

```
/
```