How to Define a Regional Arbitrary Geodetic Datum in Oracle Spatial

Christian Manthe, Lothar Gruendig

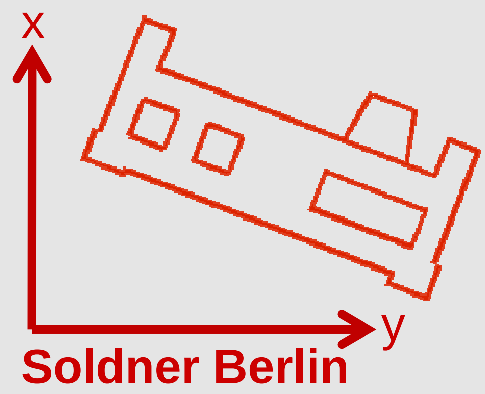
Institute for Geodesy and Geoinformation Science

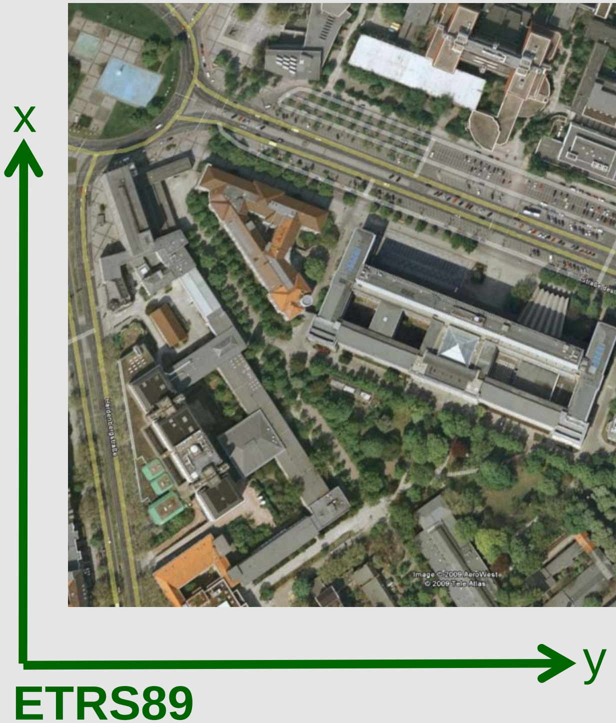
Technische Universität Berlin, Germany

Motivation

More and more data available from a variety of different sources.

Geoinformation referenced in different coordinate systems





Picture data based on GPS and use of the ETRS 89 as global system

Geoinformation based on terrestrial measurements and use of the Soldner Berlin 88

Combination of all -> **mash up**

Motivation

- Transformation in Oracle Spatial
  - We insert a user defined system into the model of coordinate systems
  - Therefore we only need some general information on
    - Projection
    - Ellipsoid
    - Orientation with respect to a geocentric system
  - If Oracle knows our system we insert the data with respect to the system
  - Then we can transform the data into the available systems in Oracle with an simple SQL-statement.

Motivation

- SQL-statement to transform local Soldner Berlin coordinates into ETRS89 coordinates

```
SELECT
        P.PKTNUM Point_number,
        SDO_CS.TRANSFORM(P.geom,83033).sdo_point.x X,
        SDO_CS.TRANSFORM(P.geom,83033).sdo_point.y Y,
FROM
        SOLDNER_BERLIN P
```

SRID of the defined ETRS89 system

Table with Soldner Berlin coordinates in the SDO_Geometry

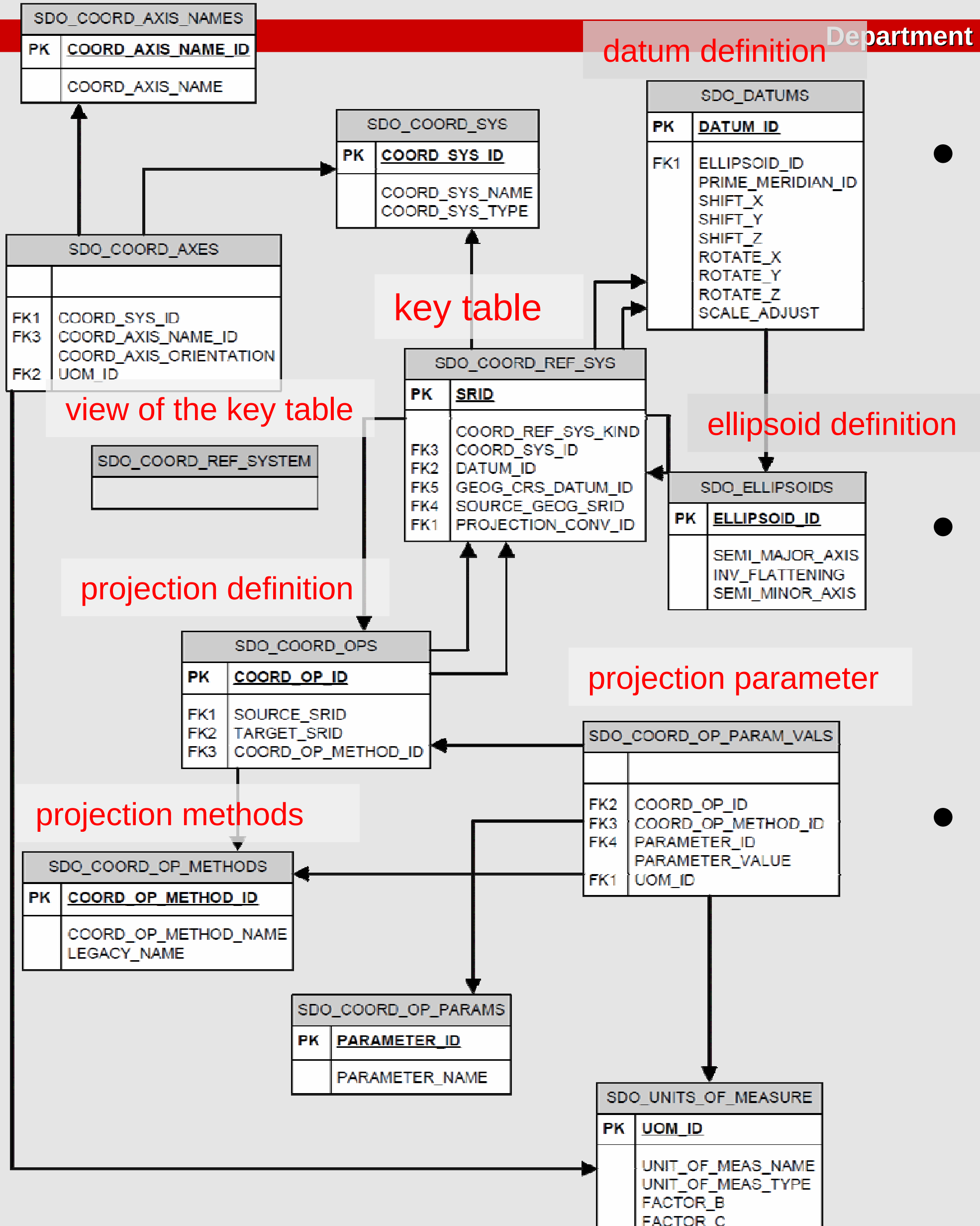
- That results in a table with three columns
  - Point_number (out of the Soldner Berlin table)
  - X (ETRS89 coordinates in UTM)
  - Y (ETRS89 coordinates in UTM)

Overview

- This presentation will explain stepwise how to set your own user defined system within Oracle

  - Data model in Oracle Spatial
  - Features of the local system in Berlin
  - Estimation of the datum parameter
  - SQL- statements to set a special local Soldner Berlin System within Oracle
  - SQL- statements to use special local Pulkovo 1942 System within Oracle
  - Conclusion

Data model in oracle spatial

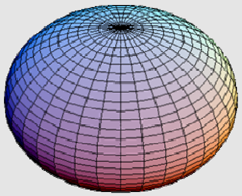


- based on the data model form the European Petroleum Survey Group (EPSG)
- Coordinate systems described in an Entity-Relationship Model
- The key-table is called SDO_COORDREF_SYS, via foreign keys all needed information is linked to this table

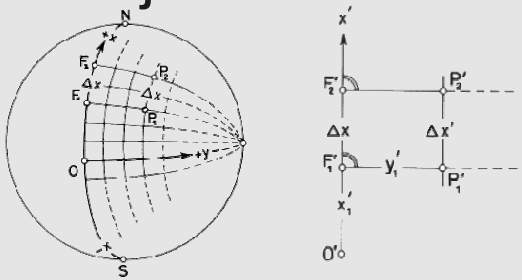
Coordinate systems in oracle

- Oracle classifies different kinds of systems

  - Vertical — for physical height systems
  - Geodetic — systems based on a individual reference surface



  - Projected — map systems based on geodetic systems (to transform the curved geodetic coordinates in a 2D drawing map)



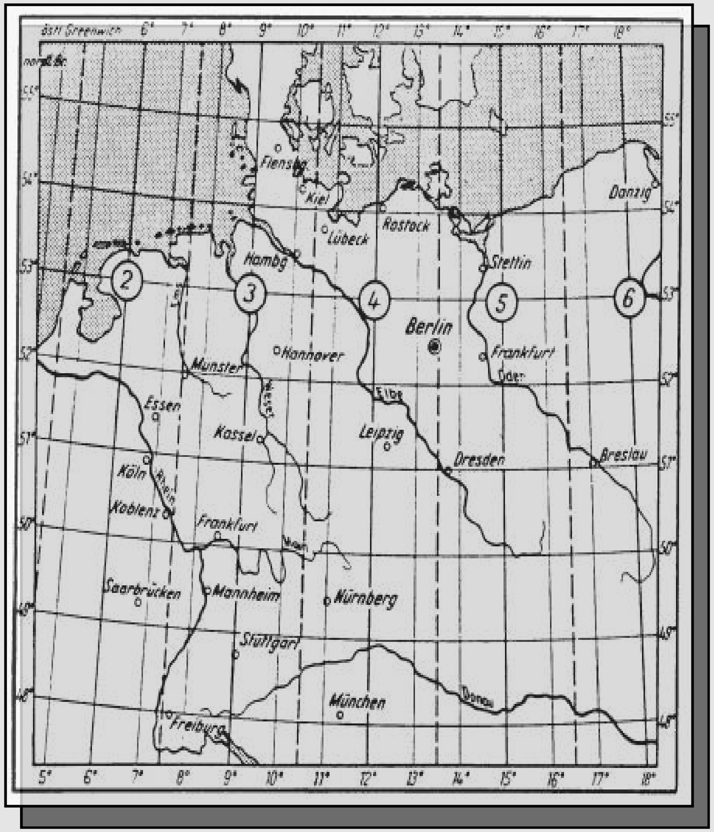
  - Compound — to define a projected system with physical heights

Coordinate systems in oracle

- Geocentric systems as WGS84 or ETRS89
  - Typically predefined
- Local systems like Soldner Berlin
  - Unknown and has to be set into the ER-model
  - Therefore the projection and datum information are needed

Lets have a closer look to our local system in Berlin.

Features of the local system in Berlin

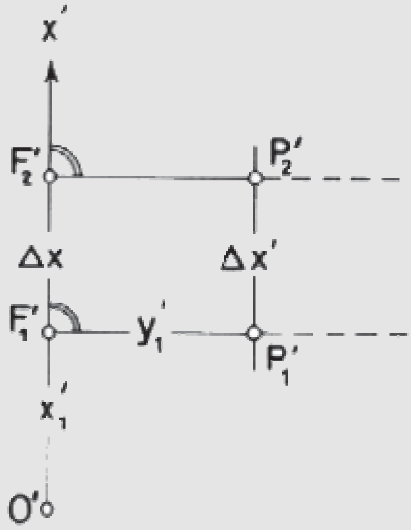


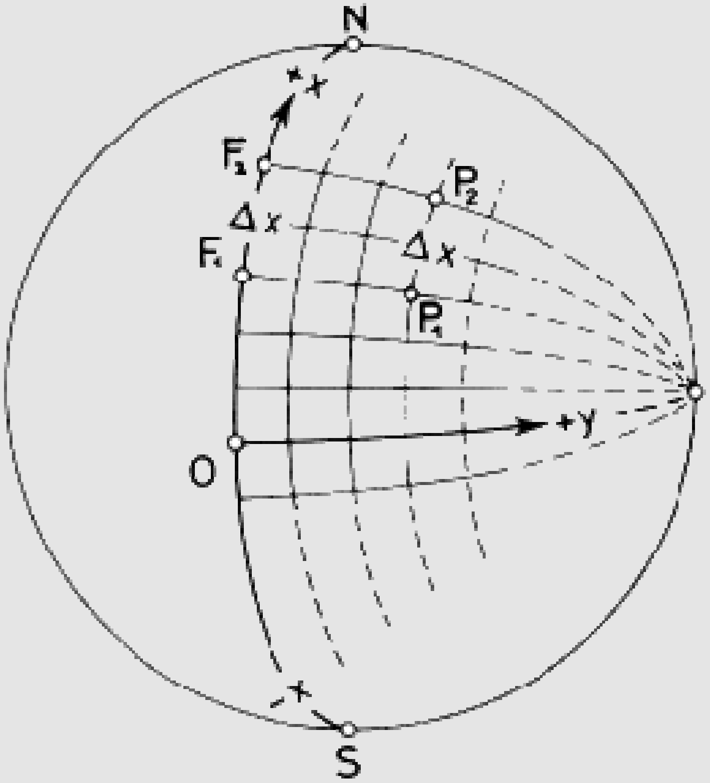
[Source: http://www.vermessung-sopart.de/Vermessung-Dateien/image006.jpg]

- Historical information
  - The geoinformation in northern Germany was in the 18th and 19th century represented in 42 different soldner projections.
  - With the development in projections the most geodata are represented in transversal cylindric projections (Gauß-Krüger projection, 3° large zones)
  - Because of the location exactly between two zones the soldner projection is still official in use

Features of the local system in Berlin

- Features of the Soldner/Cassini projection in Berlin
  - Non-conformal projection
  - Very easy to use
  - Mathematically clear defined





Features of the local system in Berlin

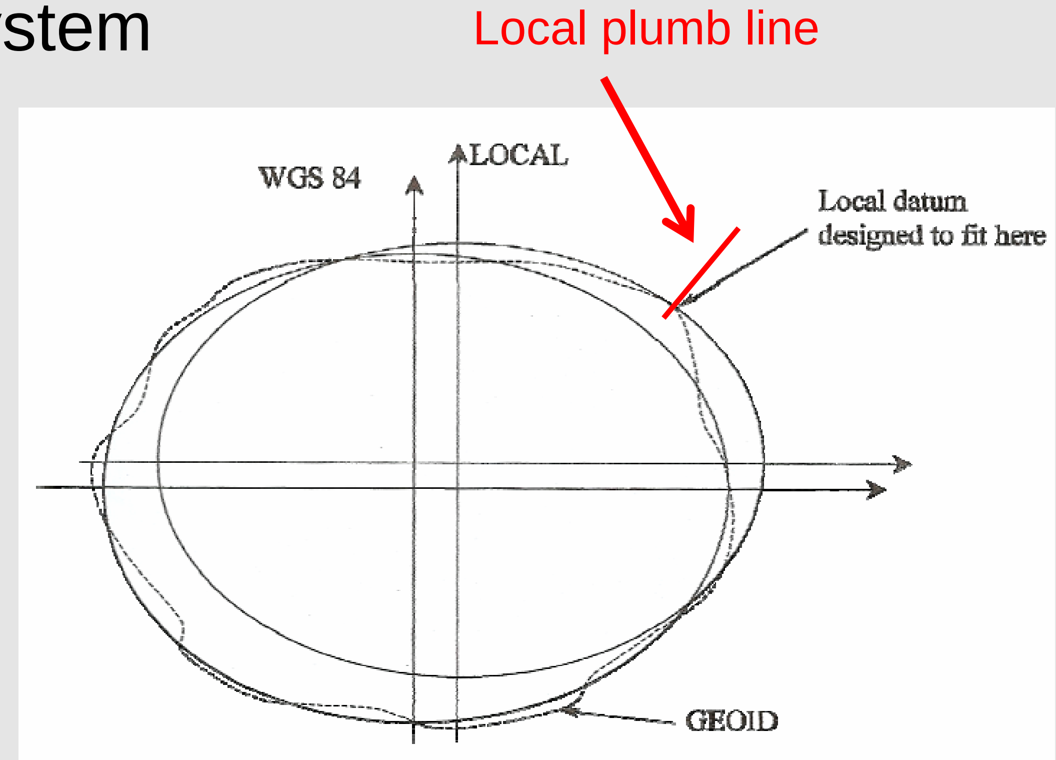
Department of Engineering Surveying and Adjustment Techniques

Picture background source: Google maps

Origin realized at the topological point “Müggelberg”
longitude λ = **13° 37’ 37.9332’’ East** and latitude ϕ = **52° 25’ 7.1338’’ North**

Features of the local system in Berlin

- Underlying geodetic system
  - Is called “Netz88”
  - Based on the DHDN “Deutsches Hauptdreicksnetz”
  - DHDN is based on triangulations
  - with a defined vertical deflection in the fundamental point “Rauenberg” as zero
  - different orientation to a geocentric system (different datum)



Features of the local system in Berlin

- With some DHDN points and new local terrestrial and GPS observations a new adjustment solution “Netz 88” was calculated

  - Without distortions in the data set
  - Realized on a different unknown reference frame
  - Different datum parameter from DHDN

How we can estimate such datum parameters?

- **With an parametric adjustment computation**

Estimation of the datum parameter

- Pre-processing to 3D world coordinates
  - Step 1
    - Project the coordinates back to the underlying geodetic system
    - Results: 2D geodetic coordinates on the underlying surface
  - Step 2
    - Calculat the 3D world coordinates with respect to each used ellipsoid
    - Results: two sets of 3D coordinates

Because of the geocentric origin and orientation of the ETRS89 solution, we get the datum parameter of the local system out of the transformation parameter between these both data sets.

Estimation of the datum parameter

- **Assumptions and restrictions**
  - We used only **2-dimensional Soldner Berlin coordinates**
  - ETRS89 as **geocentric target system**
  - **ETRS89 coordinates** were considered as non-stochastic **errorless values**
  - The **Soldner system** is expected to be nearly **homogeneous**
  - used were **14 Homologous points** known in the local and target system (evenly distributed over Berlin)

Estimation of the datum parameter

- Used adjustment model
  - 3D-Helmert transformation

Translation parameter; scale; 3D world coordinates from the local system; Geocentric coordinates; Rotation matrix

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{ETRS89} = \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} + mR \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{local}$$

  - **At first** the linear description of the rotation matrix
    - Assumptions here: for very small angles sinα=α and cosα=1
    - It easy to implement
    - But the solved parameter are **not precise enough**

$$R = \begin{pmatrix} 1 & \alpha_3 & -\alpha_2 \\ -\alpha_3 & 1 & \alpha_1 \\ \alpha_2 & -\alpha_1 & 1 \end{pmatrix}$$

Estimation of the datum parameter

- Good solution brings the use of the full Euler-Rotation-Matrix

$$R = \begin{pmatrix} \cos\beta\cos\gamma & -\cos\gamma\sin\alpha\sin\beta - \cos\alpha\sin\gamma & -\cos\alpha\cos\gamma\sin\beta + \sin\alpha\sin\gamma \\ \cos\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\cos\gamma\sin\alpha - \cos\alpha\sin\beta\sin\gamma \\ \sin\beta & \cos\beta\sin\alpha & \cos\alpha\cos\beta \end{pmatrix}$$

- It creates a non lineare adjustment problem were approximated values were needed
  - Translations T_x=T_y=T_z=0m
  - Rotation angles α=β=γ=0°
  - Scale m=1

- To **control** the Euler solution and to be **independent from the need of approximated** values we used also quaternion's

$$R=\begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3 \\ 2q_1q_2+2q_0q_3 & q_0^2-q_1^2+q_2^2-q_3^2 & -2q_0q_1+2q_2q_3 \\ -2q_0q_2+2q_1q_3 & 2q_0q_1+2q_2q_3 & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}$$

$0=q_0^2+q_1^2+q_2^2+q_3^2-1$ Condition between the unknowns

- Parametric adjustment model with restriction between the unknowns

Estimation of the datum parameter

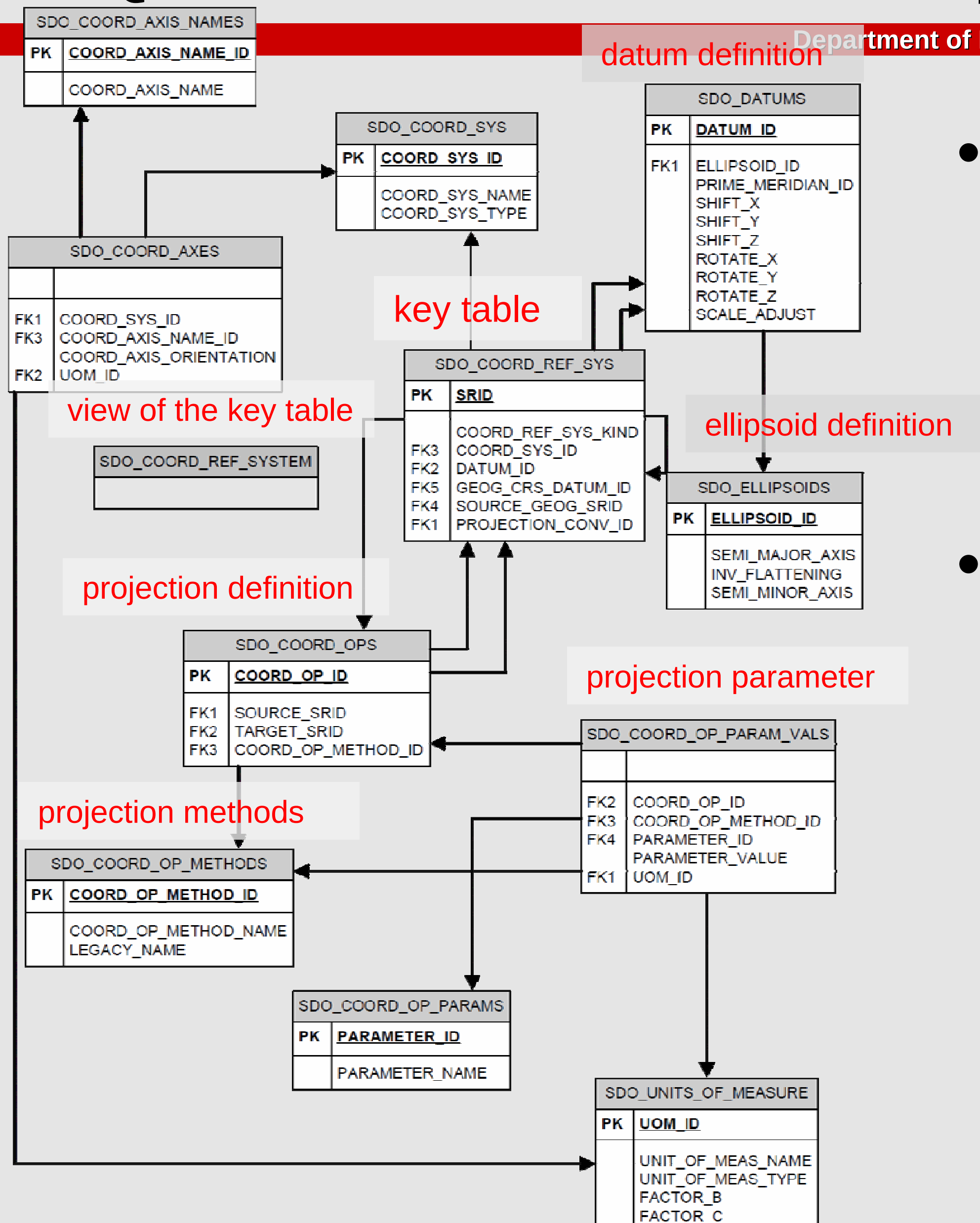
- Both adjustment solutions results the same parameters after some iterations

```
Shift in X                675.239155 m
Shift in Y                 25.303490 m
Shift in Z                422.544682 m
Rotation in X              -0.717994 sec
Rotation in X              -1.766241 sec
Rotation in X              -0.719541 sec
Scale                      -0.245916 ppm
```

- Oracle use the unit [sec] for the angles and parts per million [ppm] for the scale

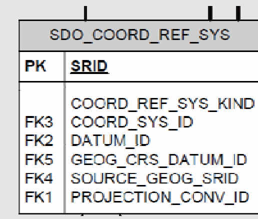
SQL- statements to set a special local system

Department of Engineering Surveying and Adjustment Techniques



- Presented SQL-statements are in the inverse order
- By setting a system in oracle start defining first with tables without undefined foreign keys

Definition of the projected Soldner Berlin System




```
INSERT
  INTO MDSYS.SDO_COORD_REF_SYSTEM(
    …
  )
  VALUES(
```

Self-proclaimed

predefined

7000001,	-- SRID *for defined Soldner CS*	
'Soldner Berlin',	-- COORD_REF_SYS_NAME	
'PROJECTED',	-- COORD_REF_SYS_KIND	*Type: Projected*
4530,	-- COORD_SYS_ID	*Type of coordinates System: 4530 – metric (X-North and Y-East)*
NULL,	-- DATUM_ID	*Datum → NULL, because of 'PROJECTED' non geodetic system*
6000000	-- GEOG_CRS_DATUM_ID	*Datum of the underlying geodetic system*
7000000,	-- SOURCE_GEOG_SRID	*ID of the used geodetic CS (Netz88)*
5000000,	-- PROJECTION_CONV_ID	*Type of projection → Soldner same ID like COORD_OP_ID*

```
    …
);
```

Undefined information

Definition of the underlying geodetic System "Netz 88"

```
INSERT
  INTO MDSYS.SDO_COORD_REF_SYSTEM(
     ...
  )
 VALUES(
  7000000,            -- SRID for defined Netz88 CS
  'geodetic net88',   -- COORD_REF_SYS_NAME
  'GEOGRAPHIC2D',     -- COORD_REF_SYS_KIND
  6422,               -- COORD_SYS_ID
  6000000,            -- DATUM_ID
  NULL,               -- GEOG_CRS_DATUM_ID
  NULL,               -- SOURCE_GEOG_SRID
  NULL,               -- PROJECTION_CONV_ID
     ...
);
```

Self-proclaimed: 7000000 (SRID for defined Netz88 CS), 'geodetic net88' (COORD_REF_SYS_NAME)

predefined: 'GEOGRAPHIC2D', 6422

'GEOGRAPHIC2D'	COORD_REF_SYS_KIND	*Type: geodetic*
6422	COORD_SYS_ID	*Type of coordinates System: (latitude and longitude)*
6000000	DATUM_ID	*Datum parameter*
NULL	GEOG_CRS_DATUM_ID	*NULL-> no other underlying system exists system*
NULL	SOURCE_GEOG_SRID	*NULL-> no other underlying system exists system*
NULL	PROJECTION_CONV_ID	*NULL-> no projection*

Undefined information: 6000000 (DATUM_ID)

SDO_COORD_REF_SYS	
PK	SRID
	COORD_REF_SYS_KIND
FK3	COORD_SYS_ID
FK2	DATUM_ID
FK5	GEOG_CRS_DATUM_ID
FK4	SOURCE_GEOG_SRID
FK1	PROJECTION_CONV_ID

Definition of the projected Soldner Berlin System

```
INSERT
  INTO MDSYS.SDO_COORD_REF_SYSTEM(
    …
  )
  VALUES(
    7000001,          -- SRID for defined Soldner CS
    'Soldner Berlin‘, -- COORD_REF_SYS_NAME
    'PROJECTED',      -- COORD_REF_SYS_KIND      Type: Projected
    4530,             -- COORD_SYS_ID            Type of coordinates System: 4530 –
                                                 metric (X-North and Y-East)
    NULL,             -- DATUM_ID                Datum → NULL, because of ‘
                                                 PROJECTED‘ non geodetic system
    6000000           -- GEOG_CRS_DATUM_ID       Datum of the underlying geodetic
                                                 system
    7000000,          -- SOURCE_GEOG_SRID        ID of the used geodetic CS (Netz88)
    5000000,          -- PROJECTION_CONV_ID      Type of projection → Soldner
                                                 same ID like COORD_OP_ID
    …
);
```

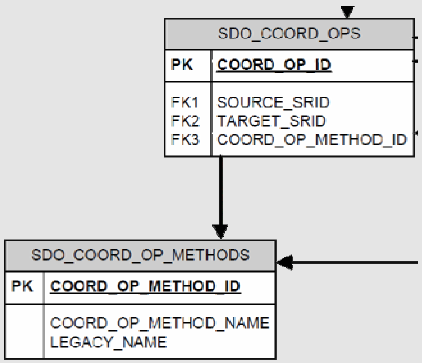
Self-proclaimed

predefined

Undefined information

SQL- statements to set a special local system

Definition of the local Soldner Berlin projection




```
INSERT
  INTO MDSYS.SDO_COORD_OPS(
     …
  )
  VALUES(
     5000000,          -- COORD_OP_ID           projection ID
     'Soldner Berlin',  -- COORD_OP_NAME         Name of the projection
     'CONVERSION,      -- COORD_OP_TYPE         Type of operation: translation
     …
     …
     9806,             -- COORD_OP_METHOD_ID    predefined projection Method: Cassini
     …
);
```

Self-proclaimed

predefined

Parameter Undefined!!!

Definition of the local Soldner Berlin projection parameters

How to set these parameters??

```
SELECT
    parameter_id || ': ' ||
    legacy_param_name
FROM
    sdo_coord_op_param_use
WHERE
    coord_op_method_id = 9806;
```

Picture background source: Google maps

Result:

8801: Latitude_Of_Origin
8802: Central_Meridian
8806: False_Easting
8807: False_Northing

SQL- statements to set a special local system

Definition of the local Soldner Berlin projection parameters

How to set these parameters??

SELECT

parameter_id || ': ' ||

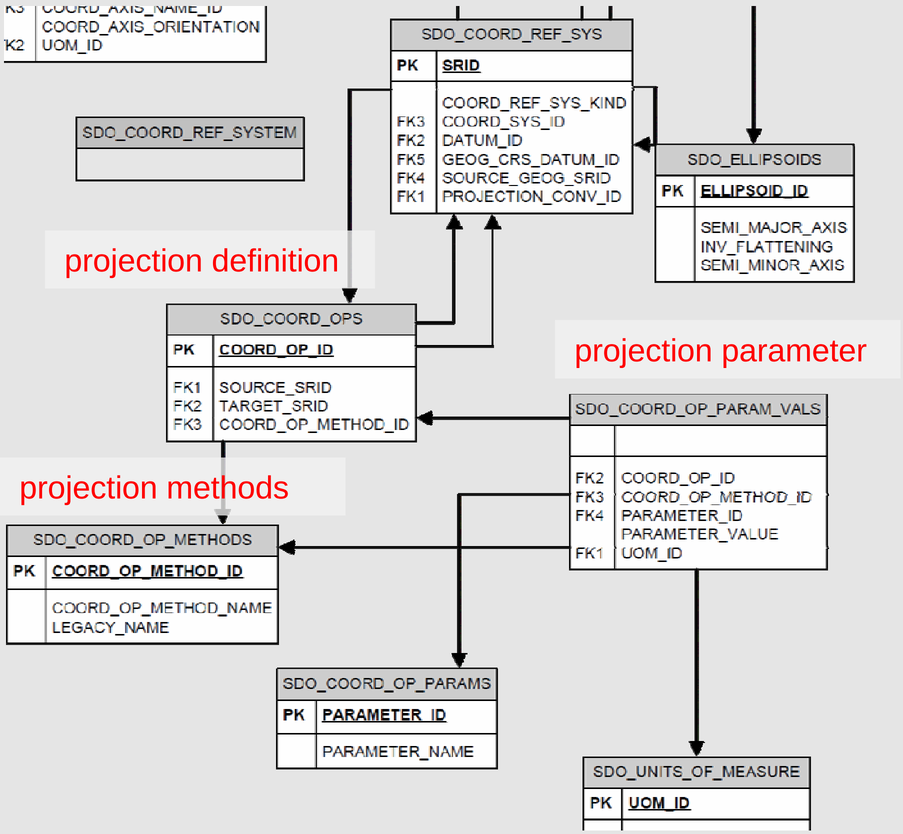
legacy_param_name

FROM

sdo_coord_op_param_use

WHERE

coord_op_method_id = **9806**;



Result:

8801: Latitude_Of_Origin

8802: Central_Meridian

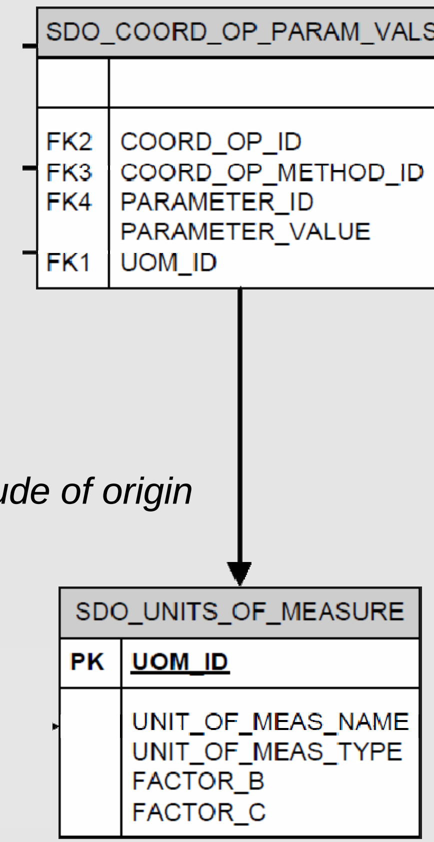
8806: False_Easting

8807: False_Northing

Definition of the local Soldner Berlin projection parameters

LATITUDE of origin

projection parameter



INSERT
  INTO MDSYS. **COORD_OP_PARAM_VALS**(
    …
)
VALUES(

5000000,	-- *COORD_OP_ID*	*of the Soldner Berlin*
9806,	-- COORD_PO_METHOD_ID	Cassini projection
8801,	-- PARMETER_ID	*Parameter ID, 8801: latitude of origin*
52.418648277	-- PARAMETER_VALUE	*Value of the parameter*
10001	-- UOM_ID	unit of the value

…
);

For every parameter!

Definition of the projected Soldner Berlin System

SDO_COORD_REF_SYS	
PK	SRID
	COORD_REF_SYS_KIND
FK3	COORD_SYS_ID
FK2	DATUM_ID
FK5	GEOG_CRS_DATUM_ID
FK4	SOURCE_GEOG_SRID
FK1	PROJECTION_CONV_ID

INSERT

INTO MDSYS.**SDO_COORD_REF_SYSTEM**(

…

)

VALUES(

7000001,	-- SRID *for defined Soldner CS*	
'Soldner Berlin‘,	-- COORD_REF_SYS_NAME	
'PROJECTED',	-- COORD_REF_SYS_KIND	*Type: Projected*
4530,	-- COORD_SYS_ID	*Type of coordinates System: 4530 – metric (X-North and Y-East)*
NULL,	-- DATUM_ID	*Datum → NULL, because of ‘ PROJECTED‘ non geodetic system*
6000000	-- GEOG_CRS_DATUM_ID	*Datum of the underlying geodetic system*
7000000,	-- SOURCE_GEOG_SRID	*ID of the used geodetic CS (Netz88)*
5000000,	-- PROJECTION_CONV_ID	*Type of projection → Soldner same ID like COORD_OP_ID*

…

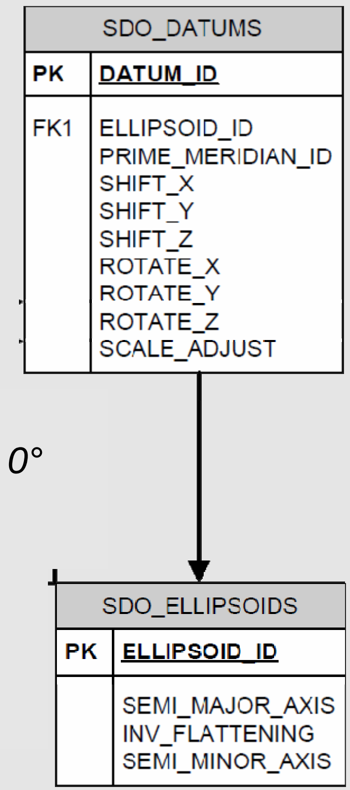
);

Undefined information

SQL- statements to set a special local system

Datum Definition of the Soldner Berlin System

```
INSERT
  INTO MDSYS. SDO_DATUMS(
     …
)
VALUES(
     6000000,              -- DATUM_ID
     'Netz88 (Berlin)',    -- DATUM_NAME            Name of the datum
     'GEODETIC',           -- DATUM_TYPE            Type of the Datum
     8804                  -- ELLIPSOID_ID          Ellipsoid ID, 8804: Bessel
     8901                  -- PRIME_MERIDIAN_ID     Central meridian Greenwich 0°
   …                          estimated parameter
     675.2392              -- SHIFT_X               unit [m]
     25.3035,              -- SHIFT_Y               unit [m]
     422.5457,             -- SHIFT_Z               unit [m]
     -0.71799386,          -- ROTATE_X              unit [sec]
     -1.76624146,          -- ROTATE_Y              unit [sec]
     -0.71954061,          -- ROTATE_Z              unit [sec]
     -0.245916,            -- SCALE                 [ppm]
);
```



Conclusion

- Oracle now knows the local Soldner Berlin system
- We can create tables with local coordinates
- That table information can be transformed into other Oracle systems only by changing the SRID number.

```
SELECT
        P.PKTNUM Point_number,
        SDO_CS.TRANSFORM(P.geom,83033).sdo_point.x X,
        SDO_CS.TRANSFORM(P.geom,83033).sdo_point.y Y,
FROM
        SOLDNER_BERLIN P
```

GK 6° / Pulkovo 1942 to UTM 48 / WGS84

- Steps for a good transformation with Oracle
  - **Step 1:** Is the GK 6° System predefined in Oracle Spatial?

```
select SRID, COORD_REF_SYS_NAME
from SDO_COORD_REF_SYSTEM
where COORD_REF_SYS_NAME
like '%GK%Zone 13%';
```

  - Yes it is!

Request:

SRID	COORD_REF_SYS_NAME
82008	GK Zone 13 (Pulkovo 1942)

So you can use it?

GK 6° / Pulkovo 1942 to UTM 48 / WGS84

Department of Engineering Surveying and Adjustment Techniques

- **<u>Step 2:</u>** Insert the coordinates as SDO_Geometry with respect to the coordinate system

```
CREATE TABLE GK13_Pulkovo (
pktnum NUMBER(10),
geom sdo_geometry);

INSERT INTO user_sdo_geom_metadata
(TABLE_NAME,COLUMN_NAME,DIMINFO,SRID)
  VALUES
  ( GK13_Pulkovo ', 'GEOM',
  SDO_DIM_ARRAY( SDO_DIM_ELEMENT('East', 0, 1000000, 0.001),
  SDO_DIM_ELEMENT('North', 0, 10000000, 0.001)), 82008);

insert into GK13_Pulkovo values( 1, SDO_GEOMETRY(2001, 82008,
SDO_POINT_TYPE(  4600939.887  ,   5800939.300  ,  NULL), NULL, NULL) );

insert into GK13_Pulkovo values( 2, SDO_GEOMETRY(2001, 82008,
SDO_POINT_TYPE(  4610938.727  ,   5801164.053  ,  NULL), NULL, NULL) );
```

Metadata entry to assign oracle the used coordinate system

insert some points

GK 6° / Pulkovo 1942 to UTM 48 / WGS84

- **Step 3:** perform the transformation and compare the result with homological points

SELECT P.PKTNUM GK13_TO_UTM48
to_char(SDO_CS.TRANSFORM(... 99999.999') East,
to_char(SDO_CS.TRANSFORM ... 9999.999') North
FROM GK13_Pulkovo P;

- If the result is not precis... ...k the parameter of the system defi...

SELECT wktext FROM cs_srs WHERE sr...

Request:
PROJCS["GK Zone 13 (Pulkovo 1942)",
GEOGCS ["Pulkovo 1942", DATUM ["Pulkovo 1942",
SPHEROID ["Krassovsky", 6378245, 298.3]],
PRIMEM ["Greenwich", 0.000000], UNIT ["Decimal Degree", 0.01745329251994330]],
PROJECTION ["Transverse Mercator"], PARAMETER ["Scale_Factor", 1.000000],
PARAMETER ["Central_Meridian", 75.000000],
PARAMETER ["False_Easting", 13500000.000000],
UNIT ["Meter", 1.000000000000]]

Well known text of the definitions
name of the system
underlying
projection parameter

GK 6° / Pulkovo 1942 to UTM 48 / WGS84

Department of Engineering Surveying and Adjustment Techniques

- **Step 4:** Check the datum definition

```
select srid, COORD_REF_SYS_NAME, GEOG_CRS_DATUM_ID  from
SDO_COORD_REF_SYSTEM
where COORD_REF_SYS_NAME like '%GK%Zone 13%';
```

Request:

SRID	COORD_REF_SYS_NAME	GEOG_CRS_DATUM_ID
82008	GK Zone 13 (Pulkovo 1942)	10094

- **Step 4.1:** Check the datum parameter

```
select DATUM_NAME name, ELLIPSOID_ID ell, SHIFT_X DX, SHIFT_Y DY,
SHIFT_Z DZ, ROTATE_X RX, ROTATE_Y RY, ROTATE_Z RZ,
SCALE_ADJUST m from SDO_DATUMS Where DATUM_ID=10094;
```

Request:

name	ellipsoid_ID	DX	DY	DZ	Rx	RY	RZ	m
Pulkovo 1942	8031	24	-123	-94	-0,02	0,25	0,13	1,1

Krassovsky ellipsoid

GK 6° / Pulkovo 1942 to UTM 48 / WGS84

- **Step 5:** Estimate the new datum parameter with the adjustment computation
- **Step 6:** Update the datum parameter

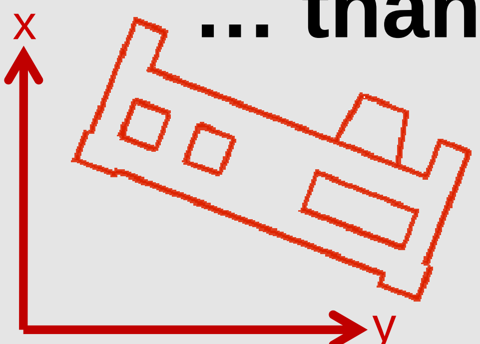
```
UPDATE MDSYS.SDO_DATUMS SET
SHIFT_X      =  26.93176  WHERE DATUM_ID = 10094;
UPDATE MDSYS.SDO_DATUMS SET
SHIFT_Y      =-123.92726 WHERE DATUM_ID = 10094;
…
```

- **Step 7:** Perform the transformation like at Step 3

Conclusion

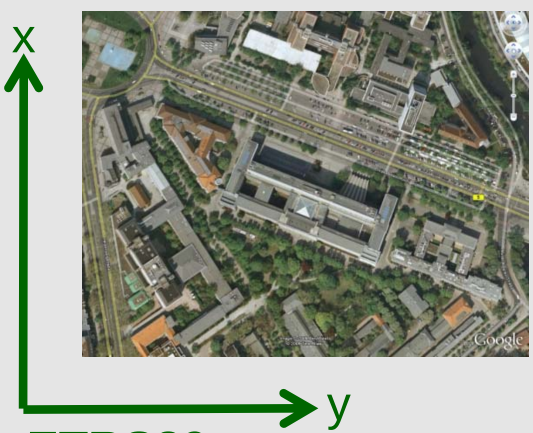
- Steps in overview
  - Collect all information
    - Projections, underlying ellipsoid,…
  - Estimate the datum parameter by using homological points
  - Insert or update the new system in Oracle data model
  - Paste the original coordinates into the SDO-Geometry with the link to the system
  - Transform the data into different systems only by changing the SRID number

... thank you. Are there any questions?



Soldner Berlin

ETRS89



ETRS89

Christian.manthe@tu-berlin.de