How to Define a Regional Arbitrary Geodetic Datum in Oracle Spatial

Christian Manthe

Institute for Geodesy and Geoinformation Science
Technische Universität Berlin, Germany

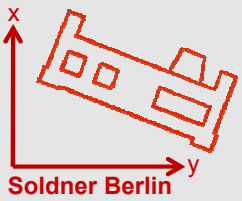
Motivation

- More and more data available from a variety of different sources
- The task is combining different data sets
  - New data along archive material
  - GPS referenced image with data obtained form a published map
- Those are different data in different systems

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 the ETRS 89 as global system

Geoinformation based on terrestrial measurements and use the Soldner Berlin 88

Combination of all -> **mashup**

Motivation

- A lot of different coordinate Systems
  - Historically, every Region was developing its **own system**
  - **Different locations** on earth require different projections
  - Coordinates may be based on a different **datum**
    - even same projection can use a different datum → **Locally best fitting ellipsoid**
- Nowadays:

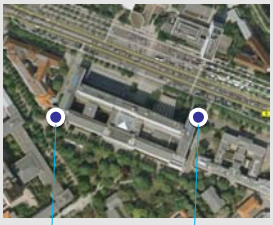
  intention to **harmonize** coordinates from different systems and regions (Europe-> ETRS89)

→ **Transformations** are needed

→ **GIS** (**Geodatabase**) is the solution for transforming coordinates easily

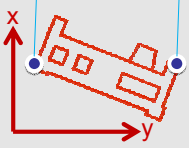
Motivation

How we can combine this different geodata sets?



1.Searching for identical points and estimate specific transformation parameters

2.Transform the scene into the target system with the calculated parameters (stored procedure).



Result:
- Fast solution
- very special parameter
- consider not the projections of the different systems

Motivation

- **Why not stored procedures?**
  - Like in every programming language you can implement a transformation in oracle (java)
  - The complete formalism in a single procedure
  - Quick solution as long as we have one target system and not very complex formulas
  - Is **not reusable with different systems** or in the inverse direction

Motivation

- Transformation in oracle spatial
  - We insert at first a user defined system into the model of coordinate systems
  - Therefore we need only some general information
    - Projection
    - Ellipsoid
    - Orientation with respect to a geocentric system
  - If oracle know our system we insert the data with respect to the system
  - Then we can transform the data into the most in oracle available systems 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

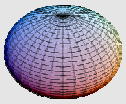
  - That results a table with three columns
    - Point_number (out of the Soldner Berlin table)
    - X (ETRS89 coordinates in UTM)
    - Y (ETRS89 coordinates in UTM)

Overview

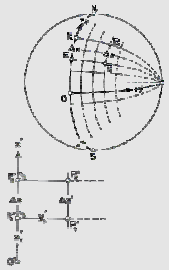
- This presentation will explain stepwise how you set your own user defined system in oracle
  - Data model in oracle spatial
  - Features of the local system in Berlin
  - Estimation of the datum parameter
  - SQL- statements to set a special local Soldner Berlin System in oracle
  - Clonclusion

Overview of coordinate systems

- Geodetic
  - Referring to spheroids (spherical but most elliptical)
    - Geometric parameter
    - Orientation parameters with respect to an geocentric spheroid



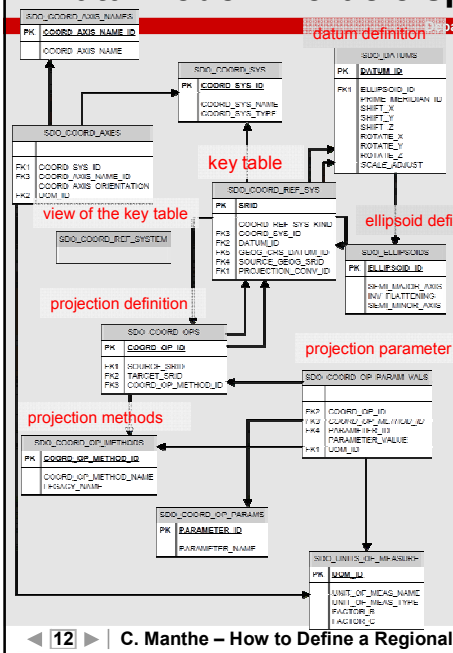
  - coordinates longitude λ and latitude ϕ
- Projected
  - Used for drawing 2-dimensional maps
  - Based on geodetic systems
  - Conversions between geodetic and projected coordinates mathematically clear defined



Overview of coordinate systems

- Divers rotation ellipsoids
  - Historical development in geodesy
    - Bessel
    - Krassowski
    - Hayfort
    - GRS80
  - With different orientations
    - German common general geodetic system is DHDN
    - With different realizations in the different states
    - Can be traced back to individual improvements in the net evaluation

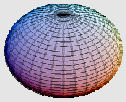
Data model in oracle spatial



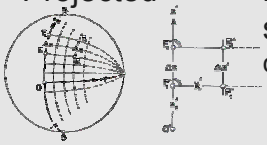
- based on the data model form the European Petroleum Survey Group (EPSG)
- Coordinate systems described in an Entity-Relationship Model
- The key-table is called SDO_COORDREF_SYS, via foreign keys all needed information is linked to this table

Coordinate systems in oracle

- Oracle classify different kind 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

Overview of coordinate systems

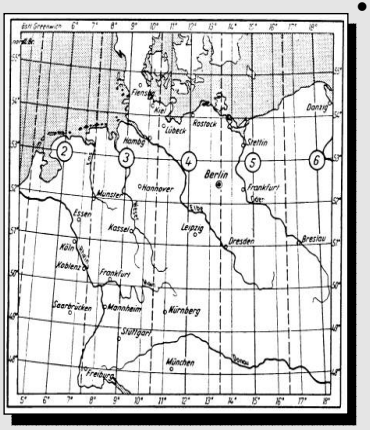
- Geocentric Systems
  - GPS facilitates investigating a global best-approximated rotation ellipsoid (figure)
  - The most recent geodetic datum is the World Geodetic System 84 (WGS84) and in Europe the European terrestrial Reference frame 89 (ETRF 89)
    - Identical orientation parameter
    - Origin in the mass center of the earth
    - Z axis very close to the real earth rotation axis
    - Ellipsoid is fitted over the whole world

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

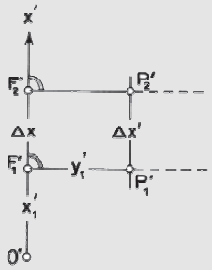


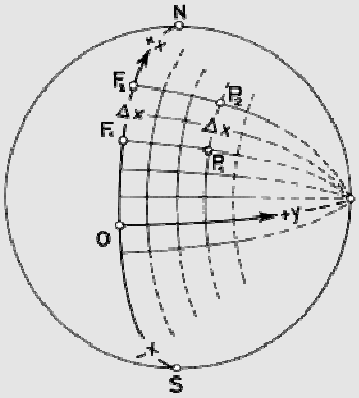
[Source: http://www.vermessung-sopart.de/Vermessung-Dateien/image006.jpg]

- 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

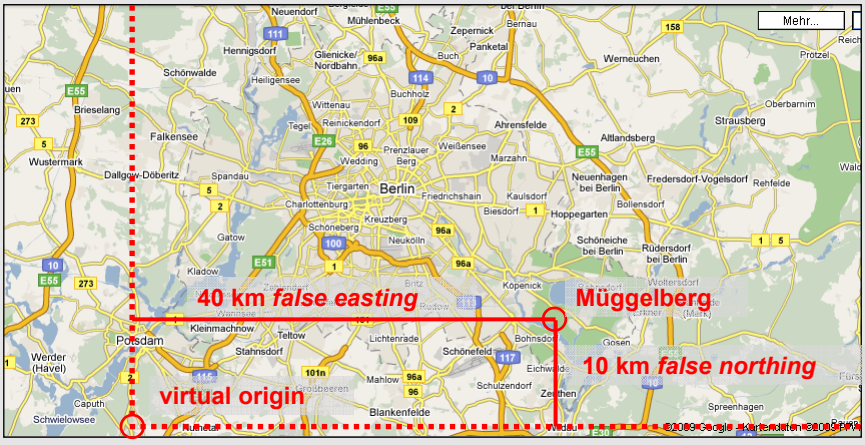
Features of the local system in Berlin

- 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

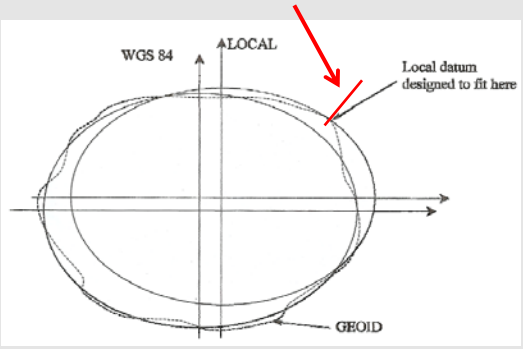


Picture background source: Google maps

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

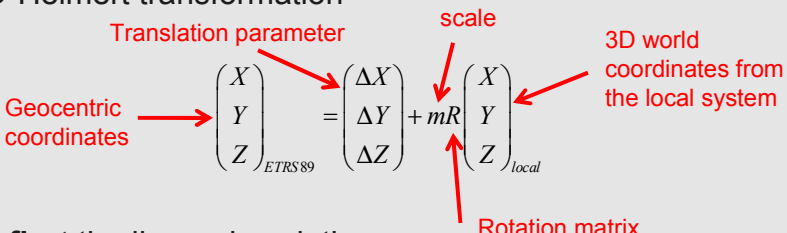
Because of the geocentric origin and orientation of the ETRS89 solution, we get the datum parameter of the local system out of the transformation parameter between these both data sets.

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



  - **At first** the linear description of the rotation matrix
    - Assumptions here: for very small angles $\sin\alpha=\alpha$ and $\cos\alpha=1$
    - 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 $T_x=T_y=T_z=0$m
    - Rotation angles $\alpha=\beta=\gamma=0°$
    - Scale m=1

Estimation of the datum parameter

- 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

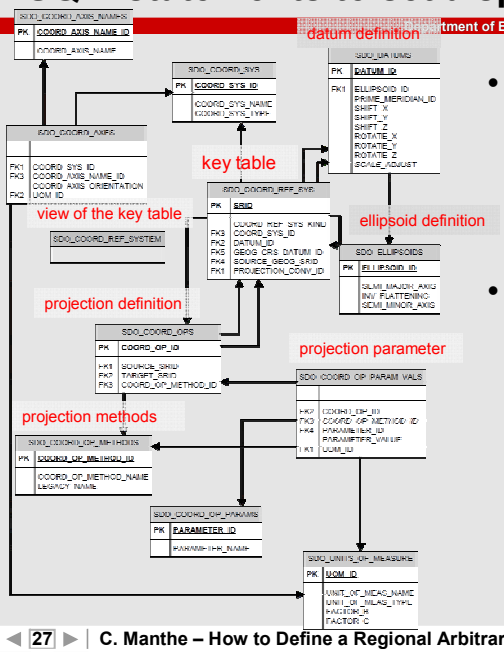
Estimation of the datum parameter

- 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

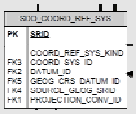
SQL- statements to set a special local system



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

SQL- statements to set a special local system

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
    4530,              -- COORD_SYS_ID
    NULL,              -- DATUM_ID
    6000000            -- GEOG_CRS_DATUM_ID
    7000000,           -- SOURCE_GEOG_SRID
    5000000,           -- PROJECTION_CONV_ID
    …
);
```

Self-proclaimed

predefined

Type: Projected

Type of coordinates System: 4530 – metric (X-North and Y-East)

Datum → NULL, because of 'PROJECTED' non geodetic system

Datum of the underlying geodetic system

ID of the used geodetic CS (Netz88)

Type of projection → Soldner same ID like COORD_OP_ID

Undefined information

SQL- statements to set a special local system

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

predefined

Type: geodetic

Type of coordinates System: (latitude and longitude)

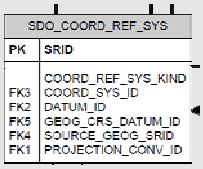
Datum parameter

NULL-> no other underlying system exists system

NULL-> no other underlying system exists system

NULL-> no projection

Undefined information



SQL- statements to set a special local system

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
    4530,             -- COORD_SYS_ID
    NULL,             -- DATUM_ID
    6000000           -- GEOG_CRS_DATUM_ID
    7000000,          -- SOURCE_GEOG_SRID
    5000000,          -- PROJECTION_CONV_ID
    …
);
```

Self-proclaimed

predefined

Type: Projected

Type of coordinates System: 4530 – metric (X-North and Y-East)

Datum → NULL, because of ‘ PROJECTED‘ non geodetic system

Datum of the underlying geodetic system

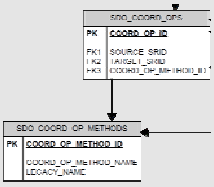
ID of the used geodetic CS (Netz88)

Type of projection → Soldner same ID like COORD_OP_ID

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!!!

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;
```

Picture background source: Google maps

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

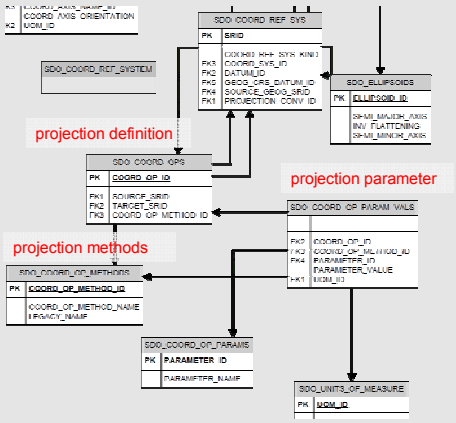
SQL- statements to set a special local system

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



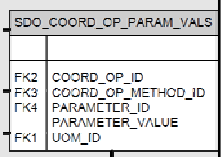
SQL- statements to set a special local system

Department of Engineering Surveying and Adjustment Techniques

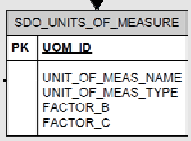
Definition of the local Soldner Berlin projection parameters

LATITUDE of origin

projection parameter



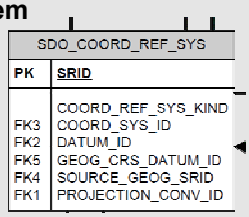

```
INSERT
  INTO MDSYS. COORD_OP_PARAM_VALS(
    …
)
VALUES(
    5000000,        -- COORD_OP_ID          of the Soldner Berlin
    9806,           -- COORD_PO_METHOD_ID   Cassini projection
    8801,           -- PARMETER_ID          Parameter ID, 8801: latitude of origin
    52.418648277    -- PARAMETER_VALUE      Value of the parameter
    10001           -- UOM_ID               unit of the value
…
);
```



For every parameter!

SQL- statements to set a special local system

Definition of the projected Soldner Berlin System



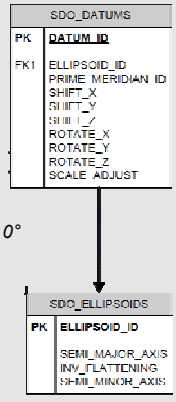

```
INSERT
  INTO MDSYS.SDO_COORD_REF_SYSTEM(
    …
  )
  VALUES(
    7000001,        -- SRID for defined Soldner CS
    'Soldner Berlin‘,  -- COORD_REF_SYS_NAME
    'PROJECTED',    -- COORD_REF_SYS_KIND     Type: Projected
    4530,           -- COORD_SYS_ID           Type of coordinates System: 4530 – 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
    …
);
```

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

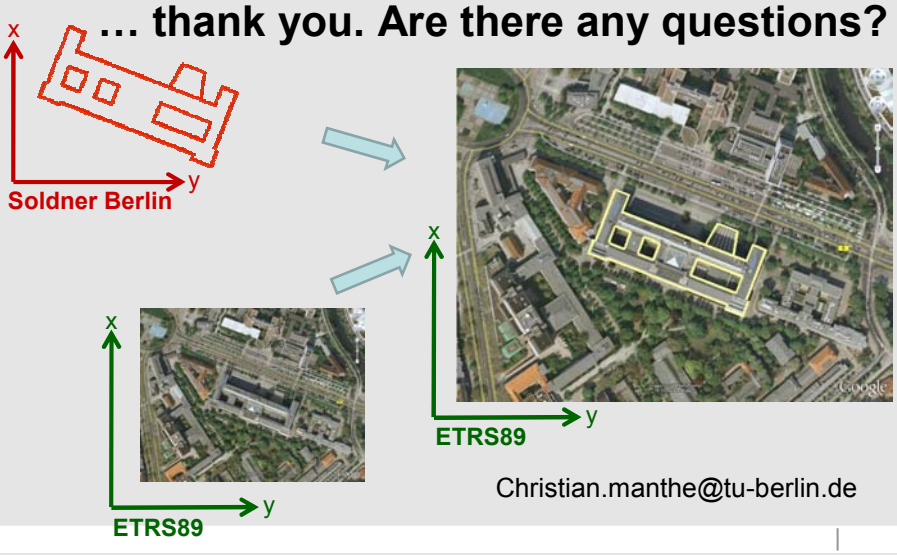
- oracle knows the local Soldner Berlin system
- We can create tables with local coordinates
- That table information can be transform like seen at the beginning into the most other oracle given systems only by changing the SRID number.

```
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
```


Conclusion

- Steps in overview
  - Collect all information
    - Projections
  - Estimate the datum parameter by using homological points
  - Insert 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?



Christian.manthe@tu-berlin.de