CONCORSO PUBBLICO, PER ESAMI, A N. 1 UNITÀ DI CATEGORIA C, POSIZIONE ECONOMICA C1, AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, PER LE ESIGENZE CENTRO INTERDIPARTIMENTALE DI RICERCA LABORATORIO DI URBANISTICA E PIANIFICAZIONE TERRITORIALE (LUPT) "RAFFAELE D'AMBROSIO" DELL'UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II (COD. RIF. 2246)

QUESITI NON ESTRATTI PROVA ORALE DEL 03 MARZO 2023

TRACCIA B

1. Si descriva la procedura di geo-codifica (geocoding).
2. Si descriva il comando CRONTAB in Linux.
3. Con riferimento alla pubblicazione in allegato, si legga e si traduca il paragrafo 2.2.


```
select from ax_flurstueck;
alter table ax_flurstueck add column flurstueck_ges character varying (50);
update ax_flurstueck set flurstueck_ges=
case
when nenner is null then (zaehler)
else zaehler ||'/' || nenner)
end
select from ap_pto;
create table strassen_namen as select * from ap_pto where art='Starsse';
alter table strassen_namen add column dw_grad numeric
update strassen_namen set dw_grad=(((drehwinkel)*180)/3.1416);
create table Hausen_nummern as select * from ap_pto where art='HNR';
alter table  Hausen_nummern add column dw_grad numeric
update  Hausen_nummern set dw_grad=(((drehwinkel)*180)/3.1416);
create table Delfisau_BRW (ID char(2),(BRW char(5),WNUM varchar(100),
ENTW char(6),WGFZ INT8, Latitude float8, longitude float8);
select * from Delfisau_BRW;
select addgeometryColumn ('public','Delfisau_BRW','geometry_column',
                          31467,'POLYGON',2);
update Delfisau_BRW
```

Figure 3: SQL code for layer preparation in the PostgreSQL database.

2.2 QField plugin configuration

QField allows to work efficiently on GIS data on-site. Its optimized user interface for Android devices hides the full power of QGIS under the hood (OPENGIS.ch 2019). This extension gives a chance to collect the relevant data related to the geometry of each building or land parcel according to the purpose of the city planner. By transferring the newly collected data into the PostgreSQL database, which has a connection to the QField layers, a Python script using the PYQGIS library and its extensions ensures the creation of a specific map layout on the fly.

QField serves as a platform and user interface, but most of the configurations must be done in QGIS desktop (see Figure 4). Figure 5 shows the configuration of the layer attributes afterwards presented on the QField interface, which then can be used by city planners for assigning and editing the data on-site. Furthermore, it offers the possibility to define certain widget types of the attributes. Its drop-down menu option is user friendly and easy to use. Some other presentation options such as check boxes will make the data collection process on-site more effective. For the monitoring process of the renovation measures, photos must be taken by the planners which will be saved in a defined directory according to the chosen widget type.

After specifying the QGIS project the "Package for QField" must be configured for the synchronization of the relevant geodata for the device used on-site (see Figure 4). After opening the QField application and loading the configuration file, the same map which was created before in QGIS desktop will be shown on the device. Now the QField application offers most of the well-known functionalities of QGIS desktop, but simpler and more intuitive.

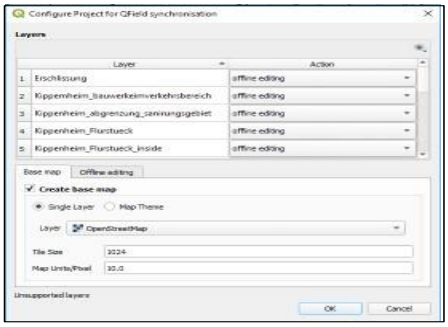

Figure 4: QField plugin configuration in QGIS.

Figure 5: The form section of the layer properties in QGIS enables to define widget types.

Figure 6 shows the QField application interface on a tablet. The table of content on the left site of the display lists all layers, their symbology, and legend description. By using the touch screen and selecting the geometry of a land parcel or a building, the drop-down menu opens and offers defined operations, i.e. adding attributes or editing the geometry of the features. The developed Python script running in the background renders the edits on the fly by changing the symbology or geometry.

TRACCIA C

1. Si descriva la SEQUENCE in Postgresql.
2. Si descriva la principale differenza tra NTFS (New Technology File System) e FAT (File Allocation Table) in Windows.
3. Con riferimento alla pubblicazione in allegato, si legga e si traduca il paragrafo 2.4.

Figure 6: QField interface of intuitive data acquisition on-site.

After finishing data acquisition and update on-site the new or updated geodata and its attributes will be synchronized via the QField plugin in QGIS desktop and finally in the PostgreSQL database. Hence, all data will be accessible from the PostgreSQL database, where acquired geometries can be checked afterwards using the visualization option (Figure 7).

The storage of the new acquired data in PostgreSQL provides the possibility to be directly accessed through an accounting management system based on PostgreSQL tables as well. These new acquired data consists of information being of great importance to the accounting procedure.

Figure 7: Updating process of the acquired data from QField in PostgreSQL database.

2.3 Python code development for map creation

Python, available under an open-source license approved by the Open Source Initiative (OSI), is freely accessible and distributable, even for commercial aspects (Python, 2018; Lawhead, 2017). The code developed in this project focuses on categorizing, labelling, and assigning the appropriate symbology for the layers of the maps in QGIS desktop. Figure 8 shows an example of a Python code categorizing spatial features according to different properties, i.e. usage of buildings.

Figure 8: Python code for layer categorization. Different colors are applied for the visualization in QField (see Figure 6).

2.4 Supporting the compensation process by SQL coding

When the former described procedure of a renovation area – which could last for a decade – is legally terminated, accounts for the compensation payments from the parcel owners have to be set up by the municipalities. Such payments become due if the renovation measures increased the real estate value of a land parcel. For efficient data proliferation a compilation of all relevant owners' data is performed. As the information from the ALKIS® data is stored in several relational tables, the relationships must be evaluated using SQL for final export of the owner data.

The SQL script in Figure 9 evaluates the relationship between six tables with owners' data. The relational join A, a one-to-one relationship, defines the connection of each parcel owner, his address and birthday and other personal data. Relational join B, a one-to-many relationship, evaluates the number of owners for each land parcel. The owners can be individuals, companies, or officials. The full join C (many-to-many) represents an intermediate step, because there is no direct connection between the owners and the land charge register (relational join D), therefore the script links the land charge register to the geocoded location stored in a so-called accounting sheet (Buchungsblatt). The last relational join E (many-to-many) evaluates the connection between the parcel number, consisting of a counter and denominator, and the results achieved so far. To put it in a nutshell, it means each land parcel owner can own more than one parcel as well as each land parcel can have more than one owner (Behr, 2011). The final compilation can be exported in CSV format for further processing.

Figure 9: SQL code for automatic relational connection between the tables according to the specified relation types.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-679-2020 | © Authors 2020. CC BY 4.0 License.

PER ORDINE DEL PRESIDENTE
IL SEGRETARIO DELLA COMMISSIONE
F.TO SIG. GIANNIELLO NICOLA