

Indicatori di sostenibilità, Sviluppo urbano e Aree vincolate e protette

Relazione tecnica di Markus Hedorfer (CM Milano, Progetto DeSK)

Contenuto

Obiettivi.....	1
Base di dati utilizzata.....	4
Indicatori.....	5
Trattamento dei dati.....	10
Unità amministrative.....	12
Comuni.....	12
Zone omogenee.....	18
Dati censuari.....	21
Sezioni.....	22
Variabili.....	23
Popolazione 1951-2018.....	25
Isolati urbani.....	27
Uso del suolo derivato dal DBTR.....	29
Uso del suolo 1954-2015.....	31
Preparazione e strutturazione dei dati.....	31
Trasformazione dei dati.....	34
Uso del suolo 1954-2015 in formato raster.....	35
Uso del suolo unificato.....	36
Gradi di impermeabilizzazione e di naturalità a livello comunale 1954-2015.....	37
Edifici.....	39
Aree vincolate e protette.....	42
Visualizzazione dei dati e delle informazioni.....	45
Lizmap.....	46
Superset.....	47

Obiettivi

Obiettivo dell'attività era l'evoluzione di alcune procedure e prodotti informativi già realizzati precedentemente al Progetto DeSK, principalmente presso la Città metropolitana di Milano. Tra i criteri più importanti lungo i quali sviluppare l'attività di evoluzione c'era la replicabilità delle procedure attraverso una codifica "soft" anziché "hard" delle variabili utilizzate (per esempio la possibilità di sostituire i dati censuari di un'annata con un'altra senza dover riscrivere il relativo codice). Questo poneva il problema di risalire ai dati originali utilizzati nelle procedure messe a punto in prece-

denza, aggiornarle laddove necessario (per esempio nel frattempo si sono fusi due comuni, rendendo così necessarie aggregazioni di dati diverse in alcune situazioni), ricostruire gli algoritmi sulla cui base erano state prodotte — in alcuni casi presumibilmente — le prime implementazioni e infine costruire degli script che fossero eseguibili in qualunque momento per aggiornare dati, informazioni e visualizzazioni più a valle, anche da parte di personale GIS e DB con competenze tecniche più superficiali. I prodotti da realizzare erano i seguenti.

- **Indicatori di sostenibilità**, che possono essere suddivisi in quattro categorie:
 - **Indicatori morfologici**, calcolati a partire dal DBTR (Database topografico regionale della Lombardia del 2015) per isolati, sezioni di censimento, comuni e zone omogenee; nello specifico
 - ▷ **densità edilizia**, definita come rapporto tra il volume costruito e l'area del territorio di riferimento (sezione di censimento, comune, zona omogenea, non viene calcolata per gli isolati),
 - ▷ **densità edilizia residenziale**, definita come rapporto tra il volume costruito con utilizzazione residenziale e l'area del territorio di riferimento (non viene calcolata per gli isolati),
 - ▷ **altezza media pesata degli edifici**, definita come rapporto tra il volume costruito e l'area delle superfici coperte,
 - ▷ **indice di eterogeneità volumetrica**, definito come scarto assoluto medio dei volumi costruiti (è però dubbio il livello di significatività del suo valore a causa di una discretizzazione metodologicamente debole nel DBTR),
 - ▷ **indice di eterogeneità pesata delle altezze**, definito come scarto assoluto pesato medio delle altezze degli edifici,
 - ▷ **distanza media tra edifici** (limitatamente ai soli isolati);
 - **Indicatori morfo-demografici**, calcolati incrociando dati a partire dal DBTR e del censimento della popolazione del 2011 per sezioni di censimento, comuni e zone omogenee; nello specifico
 - ▷ **densità di popolazione**, definita come rapporto tra la popolazione residente e l'area del territorio di riferimento,
 - ▷ **densità di popolazione delle superfici coperte**, definita come rapporto tra la popolazione residente e l'area delle superfici coperte con destinazione residenziale,
 - ▷ **volume residenziale pro capite**, definito come rapporto tra il volume delle costruzioni residenziali e la popolazione residente;
 - **Indicatori socio-demografici**, calcolati a partire dai dati del censimento della popolazione del 2011 e, in secondo ordine, del DBTR per sezioni di censimento, comuni e zone omogenee; nello specifico
 - ▷ **densità di addetti**, calcolato come rapporto tra il numero di addetti e l'area del territorio di riferimento,
 - ▷ **indice di residenzialità**, calcolato come rapporto tra la popolazione residente e gli addetti nel territorio di riferimento,

- ▷ **indice di residenzialità volumetrica**, calcolato come rapporto tra il volume delle costruzioni residenziali e il volume delle costruzioni non residenziali,
- ▷ **numero di aziende**, espresso come numero adimensionale associato a ciascun territorio di riferimento,
- ▷ **densità di aziende**, calcolato come rapporto tra il numero di aziende e l'area del territorio di riferimento;
- **Indicatori di sostenibilità in senso stretto**, calcolati a partire dal DBTR e dati demografici, in parte censuari (ISTAT) e in parte di altre fonti, per sezioni di censimento, comuni e zone omogenee; nello specifico
 - ▷ **grado di impermeabilizzazione 1954-2015**, definito come rapporto tra la somma pesata dei gradi di impermeabilizzazione determinati sulla base delle classi di copertura del suolo e l'area del territorio di riferimento,
 - ▷ **grado di naturalità 1954-2015**, definito come rapporto tra la somma pesata dei gradi di naturalità determinati sulla base delle classi di copertura del suolo e l'area del territorio di riferimento,
 - ▷ **indice del consumo di suolo ponderato**, definito nell'ambito del SDG 11.3 delle Nazioni Unite come rapporto tra l'incremento del consumo di suolo (rapporto tra le somme pesate dei gradi di impermeabilizzazione tra due annate) e l'incremento demografico (rapporto tra le popolazioni residenti tra le medesime due annate);
- **Carta delle aree vincolate e protette;**
- **Carte dello sviluppo urbano dal 1954 al 2015**, applicando una visualizzazione con le convenzioni cromatiche del CORINE Land Cover per le annate 1954 (GAI), 1980 (TEM1), 1999 (DUSAF 1.1), 2007 (DUSAF 2.1), 2009 (DUSAF 3.0), 2012 (DUSAF 4.0) e 2015 (DUSAF 5.0).

Tutti i prodotti erano da realizzarsi il più possibile su piattaforme condivise via Internet (nello specifico in ambiente VPN attraverso connessioni protette), dando la possibilità a tutto il personale autorizzato all'accesso di operare con i dati e di aggiornarli all'occorrenza. Quest'obiettivo è stato raggiunto nella sostanza, in quanto è stato possibile limitare le operazioni non eseguibili online alla preparazione iniziale di taluni dati territoriali (in alcune situazioni è stato necessario effettuare delle conversioni di formato prima di poter caricare i dati nell'ambiente online) e alla predisposizione di script (l'ambiente Postgres/PostGIS non permette di archiviare contestualmente al database le catene di istruzioni SQL necessarie per la creazione e l'aggiornamento di tabelle e dati territoriali, se non nel caso delle "viste materializzate", che però sono state utilizzate molto raramente a causa delle catene di "dipendenze" risultate troppo rigide).

Come ambienti di visualizzazione dei dati e delle informazioni dovevano essere utilizzate "Lizmap" per la visualizzazione cartograficamente avanzata di prodotti cartografici e "Superset" per la visualizzazione interattiva di dati e informazioni in forma tabellare, grafica e — con potenzialità decisamente inferiori a quelle di "Lizmap" — cartografica. Anche questo obiettivo è stato raggiunto.

Base di dati utilizzata

Per le elaborazioni eseguite sono state utilizzate fonti di dati provenienti da due soggetti distinti: Regione Lombardia e ISTAT. Per quanto riguarda l'ISTAT gli insiemi di dati utilizzati sono i seguenti.

- Confini delle sezioni di censimento con alcuni attributi morfologici, tematici e di gerarchia delle aggregazioni (comune, provincia, regione, località, lago, zona di contestazione ecc.) per un totale di 23 variabili, oltre all'identificatore poligonale e l'informazione geometrica stessa.
- Confini comunali aggiornati a prima della fusione dei comuni di Vermezzo e Zelo Surrigone nel comune di Zelo con Vermezzo. Il dato è stato utilizzato soltanto durante le fasi iniziali di esplorazione della base di dati e informativa messa a disposizione o prodotta precedentemente al subentro dell'autore della presente relazione nel Progetto DeSK. Per le elaborazioni successive sono poi stati utilizzati i limiti amministrativi prodotti da parte dell'amministrazione regionale.
- Dati attributi del censimento della popolazione del 2011 a livello di sezione di censimento con un totale di 141 variabili su popolazione e abitazioni, oltre a 12 variabili relativi alla gerarchia delle aggregazioni territoriali.
- Dati demografici delle serie storiche dei censimenti della popolazione dal 1951 al 2011 e delle ricostruzioni intercensuarie e postcensuarie a livello comunale al 1° gennaio delle annate dal 1982 al 1991 e al 31 dicembre delle annate dal 1991 al 2018.

Invece i dati prodotti dall'amministrazione regionale lombarda sono i seguenti.

- DBTR – Database topografico regionale, costituito, nella versione consegnata all'équipe del Progetto DeSK da parte dell'amministrazione metropolitana di Milano, da complessivamente 139 insiemi di dati, di cui uno tabellare (senza diretta controparte geometrica/geografica).
- Serie storica dell'uso del suolo dal 1954 al 2015, con classificazione coordinata e orientata a quella della copertura del suolo CORINE, nello specifico:
 - per il 1954 i dati sono stati ottenuti, con una qualità posizionale corrispondente alla scala 1:10.000, dai fotogrammi in bianco e nero del volo GAI (Gruppo Aereo Italiano) del 1954; il lavoro è stato eseguito dall'ERSAF (Ente regionale per i servizi all'agricoltura e foreste);
 - per il 1980 i dati sono stati ottenuti dalle carte dell'uso del suolo in scala 1:50.000, a loro volta ottenute mediante fotointerpretazione del volo TEM 1 (finalizzato alla realizzazione di carte tematiche, prima edizione) del 1980;
 - per il 1999 i dati sono stati ottenuti, nell'ambito del Progetto DUSAF (Destinazione d'uso dei suoli agricoli e forestali), versione 1.1, a partire dalle ortofoto "IT2000" della CGR (Compagnia Generale Riprese aeree di Parma), con scala nominale 1:10.000;
 - per il 2007 i dati fanno riferimento al Progetto DUSAF 2.1, hanno sempre una scala nominale di 1:10.000 e sono stati prodotti dall'ERSAF (Ente Regionale per i servizi all'Agricoltura e Foreste) a partire dalle ortofoto "IT2007" della CGR, integrate con fonti varie prodotte o

messe a disposizione nell'ambito dell'IIT (Infrastruttura per l'informazione territoriale) della Lombardia;

- per il 2009 i dati fanno riferimento al Progetto DUSAF 3.0 e sono stati prodotti, sempre dall'ERSAF e sempre per una scala nominale di 1:10.000, sulla base di immagini aeree realizzate dall'AGEA (Agenzia per le Erogazioni in Agricoltura) e integrate, come per il 2007, con fonti varie dell'IIT;
 - per il 2012 i dati sono stati prodotti secondo il medesimo schema del 2009, vale a dire Progetto DUSAF 4.0, a cura dell'ERSAF, con scala nominale 1:10.000, sulla base di immagini AGEA integrate con dati IIT;
 - per il 2015 i dati sono stati prodotti sempre dall'ERSAF, Progetto DUSAF 5.0, con scala nominale 1:10.000, sulla base di immagini AGEA, ma con “possibili variazioni nella classificazione delle aree rispetto alle precedenti versioni dei DUSAF dovute al maggior dettaglio delle ortofoto 2015”.
- Limiti amministrativi 2019, ufficiali, validati tra comuni limitrofi e aggiornati a dopo la fusione dei comuni di Vermezzo e Zelo Surrigone nel comune di Zelo con Vermezzo.
 - Aree vincolate per legge e aree naturali protette.

È stato inoltre tentato di costruire una base di dati demografici per il periodo 1954-2015, da affiancare alla serie storica dell'uso/copertura del suolo per il calcolo dell'indicatore SDG 11.3, utilizzando quale base territoriale la suddivisione comunale odierna, integrandola con una suddivisione territoriale del Comune di Milano simile, per estensione territoriale, agli altri comuni della Città metropolitana. Nonostante gli sforzi anche dell'Ufficio Servizi statistici della Città metropolitana (si desidera, in questa sede, rivolgere un ringraziamento particolare al dott. Pietro Marino che ha cercato in modo esemplare di soddisfare questa esigenza informativa interfacciandosi anche con una serie di altri enti e uffici), non è stato possibile predisporre nei tempi e con le risorse a disposizione una base omogenea e consistente. Non è stato possibile produrre una serie storica completa della popolazione residente relativa a unità territoriali subcomunali a Milano, mentre per il restante territorio metropolitano è stato possibile con alcuni compromessi metodologici che verranno illustrati più avanti nella relazione.

Indicatori

Qui di seguito si riporta in modo sintetico la caratterizzazione di ciascuno degli indicatori predisposti, con il relativo simbolo (“Sy.”), la formula per il suo calcolo, l'unità di misura dei valori, le unità territoriali rispetto ai quali l'indicatore in questione viene calcolato (I = isolati urbani, S = sezioni di censimento, C = comuni, Z = zone omogenee) nonché una breve descrizione del significato dell'indicatore.

Indicatore	Sy. Formula	Unità	I	S	C	Z	Descrizione
Media delle distanze minime tra gli edifici	$\bar{d}_{\min} = \sum ST_Distance(C_i C_j) / n$	m	×				Rapporto tra la somma delle distanze minime, calcolata con la funzione SQL/MM ST_Distance tra gli edifici (tra un edificio <i>i</i> -esimo e l'edificio ad esso più vicino, quindi indicato come edificio <i>j</i> -esimo) e il numero di edifici. L'indicatore non è stato ricalcolato.
Altezza media degli edifici	$\bar{h} = \sum(V_i) / \sum(A_i)$	m	×	×	×	×	Rapporto tra la somma di tutti i volumi costruiti e la somma di tutte le superfici coperte.
Indice di eterogeneità volumetrica	$E_V = \sum V_i - \bar{V} / n$	m ³	×	×	×	×	Scarto assoluto medio dei volumi costruiti.
Indice di eterogeneità pesata delle altezze	$E_h = \sum(A_i \cdot h_i - \bar{h}) / \sum(A_i)$	m	×	×	×	×	Scarto assoluto medio delle altezze pesate con le superfici coperte.
Densità edilizia	$\rho_C = \sum(V_i) / A$	m ³ /m ²	×	×	×		Rapporto tra la somma di tutti i volumi costruiti e l'area del territorio di riferimento.
Densità edilizia residenziale	$\rho_{CR} = \sum(V_{Ri}) / A$	m ³ /m ²	×	×	×		Rapporto tra la somma di tutti i volumi costruiti a uso residenziale e l'area del territorio di riferimento.
Densità di popolazione	$\rho_P = P / A$	ab/km ²	×	×	×		Rapporto tra la popolazione residente e l'area del territorio di riferimento.
Densità di popolazione delle superfici coperte	$\rho_{PC} = P / \sum(A_i)$	ab/ha	×	×	×		Rapporto tra la popolazione residente e la somma di tutte le superfici coperte.
Volume residenziale pro capite	$v_P = \sum(V_{Ri}) / P$	m ³ /ab	×	×	×		Rapporto tra la somma di tutti i volumi a uso residenziale

Indicatore	Sy. Formula	Unità	I S C Z	Descrizione
				e la popolazione residenziale.
Densità di occupati	$\rho_Q = Q / A$	ab/km ²	× × ×	Rapporto tra gli occupati (“addetti” + “altri retribuiti” + “volontari”) e l’area del territorio di riferimento.
Indice di residenzialità	$R = P / Q$	ab/oc	× × ×	Rapporto tra gli occupati e gli occupati nel territorio di riferimento
Indice di residenzialità volumetrica	$R_V = \sum(V_{Ri}) / \sum(V_i - V_{Ri})$	m ³ /m ³	× × ×	Rapporto tra la somma di tutti i volumi costruiti a uso residenziale e la somma di tutti i volumi costruiti a uso non residenziale.
Numero di azienda	$F = F$	az	× × ×	Numero di aziende nel territorio di riferimento.
Densità di aziende	$\rho_F = F / A$	az/km ²	× × ×	Rapporto tra il numero di aziende e l’area del territorio di riferimento.
Grado di impermeabilizzazione	$S = \sum(A_j S_j) / A$		× × ×	Rapporto tra la somma pesata dei gradi di impermeabilizzazione determinati sulla base delle classi di copertura del suolo e l’area del territorio di riferimento.
Gradi di naturalità	$N = \sum(A_j N_j) / A$		× × ×	Rapporto tra la somma pesata dei gradi di naturalità determinati sulla base delle classi di copertura del suolo e l’area del territorio di riferimento.
Indice del consumo di suolo ponderato	$C_{ab} = (N_b / N_a) / (P_b / P_a)$ $= (N_b P_a) / (N_a P_b)$		× × ×	Rapporto tra l’incremento del consumo di suolo (rapporto tra le somme pesate dei gradi di impermeabilizzazione

Indicatore	Sy. Formula	Unità I S C Z	Descrizione
			ne tra due annate a e b) e l'incremento demografico (rapporto tra le popolazioni residenti tra le medesime due annate).

I simboli utilizzati nelle formule sono per quanto possibile ricondotti alle convenzioni o alla consuetudine internazionali. Di seguito se ne riporta l'elenco, indicando nella colonna più a destra anche la parte finale dei nomi dei campi (attributi) delle tabelle nelle quali i valori possono essere incontrati.

Simbolo	Significato	Attributo
i	Indice per un elemento i -esimo di n elementi	—
j	Indice per un elemento j -esimo di m elementi	—
n	Numero di elementi (qui sempre edifici)	—
C	Edificio (costruzione)	—
R	Residenziale o Indice di residenzialità	*_IRPQ
h_i	Altezza nell'edificio i -esimo	*_IHGV
\bar{h}	Altezza media (ponderata con l'area coperta)	*_IHGM
V_i	Volume costruito (valore singolo i -esimo)	*_IVOV
$\sum V_i$	Volume costruito (somma)	*_IVOS
\bar{V}	Volume costruito medio (non ponderato)	*_IVOM
V_{Ri}	Volume costruito a uso residenziale (valore singolo i -esimo)	*_IVRV
$\sum V_{Ri}$	Volume costruito a uso residenziale (somma)	*_IVRS
A	Area del territorio di riferimento	*_IARE
A_i	Area coperta dell'edificio i -esimo	*_IARV
$\sum A_i$	Area coperta di tutti gli edifici	*_IARS
P	Popolazione residente (nella sezione di censimento)	*_SPTO
P	Popolazione residente (nel comune o nella zona omogenea)	*_IPTO
Q	Occupati (nella sezione di censimento)	*_SOTO
Q	Occupati (nel comune o nella zona omogenea)	*_IOTO
F	Aziende (nella sezione di censimento)	*_SANU
F	Aziende (nel comune o nella zona omogenea)	*_IANU
S	Grado di impermeabilizzazione – annata XX – rapporto nell'intervallo [1; 0]	*_XXRS
S	Grado di impermeabilizzazione – annata XX – percentuale (decimi di %)	*_XXPS
S	Grado di impermeabilizzazione – annata XX – percentuale (numeri interi)	*_XXQS
N	Grado di naturalità – annata XX – rapporto nell'intervallo [1; 0]	*_XXRN
N	Grado di naturalità – annata XX – percentuale (decimi di %)	*_XXPN
N	Grado di naturalità – annata XX – percentuale (numeri interi)	*_XXQN

Per quanto riguarda i nomi degli attributi (campi) delle tabelle nei quali sono scritti i valori degli indicatori sono riportati nella tabella dopo il presente capoverso. I simboli, le formule e i significati indicati nelle prime tre colonne sono quelle della prima tabella dell'inizio del paragrafo. Nella colonna "Attributo" si riportano, come nella tabella precedente, soltanto i suffissi dei nomi dei campi nei quali vengono scritti i relativi valori in quanto la parte iniziale dei nomi può variare in base all'insieme di dati o di attributi nel quale sono collocati. Il nome del suffisso inizia per tutti gli indicatori con una X per simboleggiare la parola inglese index. La seconda lettera indica l'oggetto spaziale al quale si riferisce: A = area del territorio di riferimento, B = edifici (buildings), C = superficie coperta, P = popolazione, Q = occupati, R = edifici residenziali, S = superficie impermeabilizzata (sealed), N = superficie naturale. La seconda lettera indica nuovamente un oggetto spaziale, specie nei valori che rappresentano dei rapporti tra due grandezze, oppure la prima di una coppia di lettere che funge da abbreviazione al significato dell'indice in questione. Le coppie di lettere possono essere DA = distanze medie (average), EV = eterogeneità volumetrica, EH = eterogeneità pesata delle altezze (heights), RV = residenzialità volumetrica. L'ultima lettera, quando non si tratta della seconda lettera di una delle coppie elencate poc'anzi, rappresenta il tipo di indice: N = numero di unità, R = rapporto tra due grandezze. Un'eccezione a queste regole di denominazione è rappresentata dall'indice del consumo di suolo ponderato che è contraddistinto dall'abbreviazione 113, richiamando così l'SDG 11.3 delle Nazioni Unite.

Simbolo	Formula	Significato	Attributo
\bar{d}_{\min}	$= \sum ST_Distance(C_i C_j) / n$	Media delle distanze minime tra gli edifici	*_XBDA
\bar{h}	$= \sum (V_i) / \sum (A_i)$	Altezza media degli edifici	*_XBCR
EV	$= \sum V_i - \bar{V} / n$	Indice di eterogeneità volumetrica	*_XBEV
E_h	$= \sum (A_i \cdot h_i - \bar{h}) / \sum (A_i)$	Indice di eterogeneità pesata delle altezze	*_XBEH
ρ_C	$= \sum (V_i) / A$	Densità edilizia	*_XBAR
ρ_{CR}	$= \sum (V_{Ri}) / A$	Densità edilizia residenziale	*_XRAR
ρ_P	$= P / A$	Densità di popolazione	*_XPAR
ρ_{PC}	$= P / \sum (A_i)$	Densità di popolazione delle superfici coperte	*_XPCR
v_P	$= \sum (V_{Ri}) / P$	Volume residenziale pro capite	*_XRBR
ρ_Q	$= Q / A$	Densità di occupati	*_XQAR
R	$= P / Q$	Indice di residenzialità	*_XPQR
RV	$= \sum (V_{Ri}) / \sum (V_i - V_{Ri})$	Indice di residenzialità volumetrica	*_XBRV
F	$= F$	Numero di azienda	*_XFFN
ρ_F	$= F / A$	Densità di aziende	*_XFAR
S	$= \sum (A_j S_j) / A$	Grado di impermeabilizzazione	*_XSAR
N	$= \sum (A_j N_j) / A$	Gradi di naturalità	*_XNAR
C_{ab}	$= (N_b / N_a) / (P_b / P_a)$ $= (N_b P_a) / (N_a P_b)$	Indice del consumo di suolo ponderato	*_X113

Trattamento dei dati

Il **primo di due principi** guida nel trattamento dei dati è stato quello della localizzazione di possibilmente tutti i dati, geometrici e attributi, nel database Postgres/PostGIS per la Città metropolitana di Milano nell'ambito del Progetto DeSK, temporaneamente localizzato in un server virtuale presso il Servizio Informatica/SIT della Città metropolitana di Venezia, al fine di garantire la massima trasparenza dei processi di produzione, manipolazione, analisi e visualizzazione dei dati. Questo principio, tuttavia, non è stato possibile applicare con rigore assoluto, in quanto alcune operazioni e passaggi di elaborazione richiedevano necessariamente il lavoro su computer locali. Si tratta delle seguenti operazioni e passaggi:

- Acquisizione dei dati, che è avvenuta manualmente (per esempio accedendo ai siti Internet dell'ISTAT, della Regione Lombardia e di altri soggetti) con scarico dei dati. È teoricamente possibile automatizzare le procedure di acquisizione dei dati in modo da importare direttamente i dati dai portali web nel database Postgres, ma lo sviluppo del software necessario a tal fine non era contemplato dall'incarico di consulenza in questione.
- Preparazione di alcuni dati. Si è trattato di alcune operazioni, la cui esecuzione in remoto nel database Postgres avrebbe eccessivamente risentito della macchinosità di alcune procedure Postgres o PostGIS (come per esempio l'assegnazione dei comuni della Città metropolitana di Milano alle singole zone omogenee, che è stata eseguita in locale in modo interattivo) o la cui esecuzione in remoto avrebbe risentito del collo di bottiglia della connessione Internet (come per esempio la generazione, trasferimento o manipolazione dei dati raster della serie storica dell'uso del suolo, il cui trattamento in formato raster è poi stato sostanzialmente abbandonato proprio a causa di queste difficoltà tecnologiche).
- Procedure di trasferimento dal computer locale sul server del database. Una volta scaricati i dati recuperati sul computer locale, c'erano due possibilità operative per il loro trasferimento sul server: eseguire il trasferimento manualmente oppure eseguirlo tramite script salvabili su disco ed riutilizzabili anche successivamente. Il trasferimento manuale ha il vantaggio di essere generalmente più immediato e veloce, ma ha lo svantaggio di non essere direttamente riutilizzabile. Per il suo riutilizzo occorre predisporre una documentazione scritta accurata con istruzioni precise su quale software, con quali parametri utilizzare. Invece, utilizzando gli script il riutilizzo delle procedure è più robusto, in quanto il software necessario viene direttamente richiamato dalla riga di comando e in quanto i parametri variabili in base alla configurazione del computer in uso (per esempio cartelle di installazione del software o cartelle in cui risiedono localmente i dati, così come i parametri di connessione al server del database). Laddove possibile (non in tutte le situazioni era però possibile) si è adottato il metodo per script che, nello specifico, si è tradotto sostanzialmente nell'uso dei programmi `psql` (ambiente di lavoro per Postgres da riga di comando per database in locale e in remoto, utilizzabile anche in forma parametrica senza entrare in modalità interattiva), `shp2pgsql` (software di trasferimento di dati spaziali in formato Shapefile nel database PostGIS) e `raster2pgsql` (software di trasferimento di dati raster in

formato Shapefile nel database PostGIS). Gli script sono stati scritti per il “Prompt dei comandi” (cmd.exe) dei sistemi Microsoft e salvati sotto forma di file “batch” (file di testo ANSI o UTF-8 con estensione convenzionale *.bat). Nel caso dell’utilizzo degli script per una delle shell dei sistemi Unix, Linux o Mac OS, per i quali i programmi psql, shp2pgsql e raster2pgsql sono disponibili, occorrerà eseguire una trasposizione del codice batch, possibilmente (per questioni di portabilità tra le shell) verso codice Bourne shell. Si tratta di un’operazione relativamente facile, che però non è stata predisposta nell’ambito del presente incarico.

- Procedure di manipolazione delle tabelle Postgres e PostGIS. L’interazione con i dati può avvenire anche qui in due modi: interattivamente, utilizzando gestori come pgAdmin (ambiente indicato più di frequente per l’interazione con i database Postgres/PostGIS), dove però non rimane alcuna traccia documentale automatica delle operazioni eseguite, oppure con il linguaggio PostgreSQL, che è una variante del linguaggio SQL (Structured Query Language) specifica per i database Postgres, che implementa il linguaggio SQL standard, alcune estensioni, comprensive delle estensioni di analisi spaziale note con il nome “PostGIS”, che trasformano Postgres in un ambiente GIS, privo però di funzioni di disegno interattivo e privo di funzioni avanzate di visualizzazione dei dati spaziali. Salvo rarissime eccezioni (per esempio la manipolazione dei nomi dei comuni), tutte le operazioni di creazione, manipolazione e analisi dei dati sono state eseguite tramite istruzioni SQL salvate in file script (file di testo UTF-8 con estensione convenzionale *.sql). Postgres non dà la possibilità di salvare le istruzioni SQL in remoto (salvo implicitamente per le *viste* e le *viste materializzate*, il cui uso, inizialmente previsto e implementato, e poi stato abbandonato perché, con l’adozione dello scripting SQL, era diventato superfluo e, anzi, troppo macchinoso in quanto *viste* e *viste materializzate* impediscono l’eliminazione o la ridefinizione della struttura delle tabelle da cui derivano, cosa che è stato necessario fare frequentemente durante il lavoro con i dati), per cui è stato necessario salvare la raccolta — piuttosto voluminosa — degli script sul computer locale. Tutti gli script batch e SQL realizzati e utilizzati verranno messi a disposizione dei responsabili del Progetto DeSK.

Il **secondo principio** nel trattamento dei dati — oltre all’archiviazione centralizzata nel database Postgres/PostGIS — è la replicabilità di tutte le operazioni eseguite. In questo contesto, il termine “replicare” non deve essere confuso con lo stesso termine utilizzato nel contesto delle archiviazioni ridondanti di dati e database. Nella terminologia in lingua inglese, i due concetti sono espressi con parole differenti: “replication” per l’archiviazione ridondante e “reuse” per il riutilizzo di dati e procedure, dove nella problematica qui posta ci si riferisce alla replicabilità (riutilizzabilità) delle procedure di trattamento dei dati. A tal fine era necessario adoperare la massima trasparenza sul recupero dei dati alla fonte, sulle modalità della loro manipolazione e trasformazione fino alla visualizzazione e, più in generale, trasformazione in informazione. Grazie all’adozione sistematica dello scripting — già illustrato all’elenco qui sopra — la trasparenza delle operazioni eseguite e della costruzione degli algoritmi è ampiamente garantita. Nelle poche situazioni in cui non era possibile lavorare con lo scripting, le operazioni sono documentate in modo dettagliato nell’ambito di file ausi-

liari sul disco locale (che verranno messi a disposizione assieme agli script), come classici file README (“leggimi”) o altro, ma anche nella presente relazione tecnica.

Unità amministrative

Le unità amministrative che interessavano nell’ambito della parte milanese del Progetto DeSK sono sostanzialmente soltanto i Comuni e le Zone omogenee, queste ultime previste dalla Legge Delrio (Legge n. 56/2014, articolo unico, comma 11, lettera c, richiamate anche dai commi 22 e 57), mentre le Province non interessavano nella misura in cui l’orizzonte territoriale del trattamento dei dati è la Città metropolitana di Milano, che territorialmente ha rango di Provincia (“ente territoriale di area vasta”, secondo la terminologia della Legge Delrio). Le procedure di acquisizione e di trattamento di dati comunali e zonali erano tra loro completamente differenti.

Comuni

Sebbene i limiti/poligoni comunali possano essere facilmente derivati dai poligoni delle sezioni di censimento (attraverso i campi COD_ISTAT e PRO_COM, poi convertiti nei rispettivi campi AUIT_CD e ZK11_CD), è concettualmente più corretto utilizzare il dato messo a disposizione dalla Regione Lombardia attraverso il suo Geoportale (<http://www.geoportale.regione.lombardia.it>, “Download Dati”, “Accedi al servizio”, “Limiti amministrativi 2019”, “Vai al metadato”, “Scarica dato”, “Regione_Lombardia.zip”). A differenza delle geometrie delle sezioni di censimento, costruite dall’ISTAT a partire da una serie di informazioni eterogenee, il dato regionale poggia su una base legale solida che viene brevemente descritta anche nei metadati del Geoportale, alla voce “Modalità di realizzazione del prodotto” nella sezione finale “Informazioni sul prodotto”. Per chiarezza di esposizione si cita per intero il capoverso in questione.

La legenda indica le fonti e le modalità di acquisizione dei limiti amministrativi rappresentati. In particolare già dal 2018 era stata aggiunta la voce “Validati tra Comuni limitrofi” per evidenziare che il tracciato cartografico dei limiti amministrativi è stato rivisto e validato tra Comuni limitrofi. I limiti amministrativi derivanti dalla CTR o da DBT non verificato con Catasto sono oggetto di verifiche puntuali per migliorarne la qualità.

Nell’ambito del file archivio compresso REGIONE_LOMBARDIA.zip recuperato dal Geoportale, è stato utilizzato l’insieme di dati nel formato Shapefile Comuni_2019_poligonalι.shp, mentre gli altri insiemi — “Comuni lineari” (vale a dire limiti comunali), “Comunità montane”, “Province” e “Regione” — non sono stati utilizzati. Da un punto di vista concettuale, l’importazione nel database Postgres/PostGIS dei poligoni comunali ha comportato i seguenti due passaggi.

1. Caricamento sul server con contestuale conversione dal formato Shapefile nel formato PostGIS dei file scaricati dal Geoportale lombardo.
2. Normalizzazione degli attributi e loro conformazione allo standard di denominazione adottato.

Per una maggiore flessibilità nell’impostazione dei parametri di caricamento e di conversione, nonché per garantire la replicabilità delle procedure eseguite tramite script riutilizzabili anche in futuro,

si è optato per l'impiego del programma `shp2pgsql`, fornito assieme al pacchetto `pgAdmin` per la gestione (in locale e in remoto) dei database Postgres/PostGIS. È un programma che viene eseguito dalla riga di comando ("Bourne", "C", "K", "Z" o altre shell per sistemi Unix, Linux e Mac OS, o "Prompt dei comandi" per i sistemi Microsoft) e come tale può essere facilmente incluso in script eseguibili. Nel caso specifico, le operazioni sono state eseguite su sistemi Microsoft e, pertanto, è stato usato l'ambiente "Prompt dei comandi" (`cmd.exe`) e, in qualche situazione, la più performante "PowerShell" (`powershell.exe`). Tutte le operazioni eseguite con `shp2pgsql` — così come anche quelle eseguite con `raster2pgsql` nei tentativi iniziali con la serie storica dell'uso del suolo in formato raster — sono state codificate all'interno di script "batch" (file di testo ANSI o UTF-8 con estensione convenzionale `*.bat`) associandovi in modo statico anche i relativi parametri (cartelle dei programmi eseguibili, dei file di origine e di destinazione, parametri di connessione al server del database, parametri di conversione e via dicendo). Il caricamento sul server è avvenuto sostanzialmente eseguendo i seguenti sei comandi.

```
%_ppath%\shp2pgsql ^
-s 32632 ^
-d ^
-g mpl ^
-D ^
-k ^
-I ^
-t 2D ^
%fpath%\%_fname% ^
public.%_tname%> tmp.sql
echo comment on table public.%_tname% is>> tmp.sql
echo '%_tdesc%';>> tmp.sql
echo alter table public.%_tname% rename column gid to "%_cname%";>> tmp.sql
echo alter index public.%_tname%_%_gname%_idx rename to %_iname%;>> tmp.sql
%_ppath%\psql -h %_phost% -d %_pname% -U %_puser% -f tmp.sql
```

I sette parametri del programma `shp2pgsql` indicano rispettivamente il sistema di riferimento delle coordinate (`-s 32632` imposta il sistema UTM, fuso 32 nord, orientato al WGS84), la modalità di creazione della tabella PostGIS (`-d` elimina, nel caso esista già, ricrea e popola la tabella), il nome del campo contenente i dati geometrici (`-g mpl` imposta il nome a "mpl", che nelle convenzioni qui adottate indica geometrie multi-poligonali), la modalità di trasferimento dei dati (`-D` impone l'uso del comando SQL `copy` anziché `insert`, che è più performante), il trattamento dei nomi dei campi (`-k` mantiene la distinzione dei nomi in maiuscole minuscole, così come incontrate nei dati di origine, anziché convertire tutti i nomi in soli caratteri minuscoli), la creazione o meno di un indice spaziale (`-I` lo crea assegnandogli un nome predefinito, che poi viene modificato nella penultima riga dello script riportato sopra) e la dimensionalità delle coordinate (`-t 2D` riproietta, se necessario, le coordinate sul piano x-y). Nella quintultima e quartultima riga viene impostato il commento Postgres della tabella, nella terzultima tabella viene rinominato il campo `gid` (ID dell'elemento geometrico secondo i valori predefiniti di Postgres) utilizzando il valore assegnato alla variabile d'ambien-

te `%_cname%` e nella penultima riga viene cambiato il nome dell'indice spaziale. Infine, nell'ultima riga viene eseguito il programma `psql` — anch'esso facente parte del pacchetto `pgAdmin` — con il quale viene creata una connessione al database Postgres ed eseguita, tramite il reindirizzamento dell'output dal file temporaneo `tmp.sql` (parametro `-f` del programma `psql`), la serie di istruzioni SQL ivi precedentemente indirizzati (la prima istruzione utilizza l'operatore di redirectione `>` che elimina, qualora esista già, il file `tmp.sql`, poi lo crea/ricrea e vi scrive l'output dell'istruzione, mentre le successive istruzioni utilizzano l'operatore `>>` che appende gli output alla fine del file specificato).

Le istruzioni elencate sopra sono salvate nel file batch `load_epo1.bat` assieme a tre istruzioni di impostazione delle variabili `%_cname%` (nome del campo contenente l'identificatore numerico degli elementi multi-poligonal), `%_tname%` (nome della tabella PostGIS di destinazione, utilizzando altre le variabili `%_dname%` e `%_vdest%` precedentemente impostate) e `%_iname%` (nome dell'indice spaziale che impiega le medesime due variabili precedentemente impostate). Il file `load_epo1.bat` viene eseguito al termine delle istruzioni archiviate nel file batch `load_auco_epo1_01pg.bat`, il cui compito è di chiamare lo script di impostazione di alcune variabili di sistema, come le cartelle in cui si trovano file eseguibili e la cartella radice del progetto GIS sul computer locale (script `system_vars.bat`) e di impostare altre variabili specifiche per il caricamento dell'insieme di dati dei poligoni comunali. Il suffisso `_01pg` dello script batch, così come anche della tabella di destinazione `auco_epo1_01pg`, indica la versione 1 (distinta dalla versione 0 di acquisizione del dato, con suffisso `_00ac`) dell'importazione nell'ambiente Postgres.

La tabella PostGIS creata in questo modo corrisponde alla prima versione importata, ma non ancora normalizzata e standardizzata, dell'insieme AUCO (“Unità amministrative – Comuni”) al quale viene assegnato nell'implementazione PostGIS l'identificatore `auco_epo1` (`epo1` = elementi *poli*gonali), qui provvisto del suffisso di versione `_01pg`. Nel passaggio successivo, eseguito nell'ambito di due script SQL distinti, sono state eseguite le operazioni di normalizzazione e di standardizzazione dei dati di identificazione e di qualificazione dei comuni. Si tratta delle seguenti operazioni:

- Limitazione dei dati alla sola Città metropolitana di Milano.
- Separazione dei dati di qualificazione (per esempio nomi e associazioni alle zone omogenee) da quelli di identificazione (codici identificativi), creando, da una parte, una tabella degli attributi delle istanze comunali `auco_i` senza informazioni geometriche e, dall'altra, una “tabella” (nella terminologia PostGIS) degli elementi poligonal `auco_epo1` associandovi i soli codici identificativi (in realtà, per praticità e per ovviare ad alcune macchinosità dell'architettura di Postgres, nonostante rappresenti una violazione delle regole di integrità dei database relazionali, si sono incluse nella tabella `auco_epo1` anche diversi dati di tipo qualitativo).
- Eliminazione di ridondanze informative. Per esempio i codici identificativi dei comuni sono stati ridotti a uno solo corrispondente allo schema di denominazione `rppppccc`, dove *r* è un numero a una o due cifre indicante il codice ISTAT della Regione (qui sempre 3 per la Lombardia), *ppp* è il codice ISTAT della Provincia (per una questione di retro-compatibilità con i dati più vecchi, qui si scrive sempre 015 come per la vecchia Provincia di Milano anziché 215, che

corrisponde al nuovo codice introdotto dall'ISTAT per le città metropolitane a partire dal 1° gennaio 2017 che maggiore semplicemente i vecchi codici delle province trasformate in città metropolitane di 200) e *ccc* è il numero progressivo del Comune.

- Trasformazione delle geometrie multi-polygonali in poligoni semplici. La maggior parte dei dati ottenuti nel formato Shapefile erano strutturati secondo il paradigma multi-polygonale, che viene riconosciuto e accettato da PostGIS, ma che viola le regole di integrità referenziale tra elementi geometrici di base e attributi. Al fine di mantenere la compatibilità anche verso altri prodotti software GIS, è più corretto gestire situazioni multi-polygonali attraverso i codici identificativi (per esempio nel caso di esclave amministrative, dove il territorio di un comune è costituito da due o più poligoni, ciascuno dei poligoni sarà dotato di autonomo codice identificativo poligonale, qui sempre espresso nell'ambito del campo *EPOL_ID*, mentre a tutti i poligoni sarà associato il medesimo codice identificativo comunale, qui sempre espresso nell'ambito del campo *AUCO_ID*). L'archiviazione dei dati con impianto areale sotto forma di poligoni semplici è inoltre necessaria per creare visualizzazioni dei dati territoriali con le librerie *deck.gl* per l'ambiente *Superset*.

Oltre alle operazioni appena elencate, sono state anche calcolati una serie di campi geometrici aggiuntivi al fine di rendere possibile le visualizzazioni nell'ambiente *Superset*. Nello specifico si tratta dei seguenti campi, inclusi in *auco_epol*: *EPOL_IAREK* = area del poligono in km²; *EPOL_DECK* = geometria poligonale nel formato *deck.gl* per *Superset*, in coordinate geografiche; *EPOL_DECK1* = come *EPOL_DECK*, ma con geometria semplificata con tolleranza 1 m; *EPOL_DECK2* = geometria semplificata con tolleranza 2 m; *EPOL_DECK5* = tolleranza 5 m; *EPOL_DECKA* = tolleranza 20 m; *EPOL_JSONC* = geometria puntuale del centroide poligonale nel formato *GeoJSON*, in coordinate geografiche; *EPOL_ECOX* = coordinata x del centroide poligonale (in coordinate geografiche); *EPOL_ECOY* = coordinata y del centroide poligonale.

Prima dell'esecuzione delle operazioni sui dati geometrici poligonali, è stata creata la tabella degli attributi di qualificazione delle istanze comunali *auco_i*. L'istruzione SQL principale di questa operazione è

```
create table auco_i as
select
  "COD_ISTATN"::integer    as "AUCO_ID",
  "NOME_COM"::varchar(64) as "AUCO_INAI",
  0::integer              as "AUZO_ID"
from auco_epol_pg01
where "COD_PRO" = 15;
```

che trasforma il "codice ISTAT" (campo *COD_ISTATN*) nel campo numerico intero *AUCO_ID* e trasforma il campo dei nomi comunali (*NOME_COM*, di tipo testo variabile con profondità iniziale di 100 caratteri) nel campo testuale *AUCO_INAI* con profondità iniziale di 64 caratteri. L'istruzione crea poi il campo intero *AUZO_ID* vuoto (tutti i record sono inizialmente popolati con il valore 0), che riceverà successivamente i codici identificativi delle zone omogenee a cui sono associati i singoli comuni.

La clausola `where "COD_PRO" = 15` seleziona soltanto i comuni ricadenti nella Città metropolitana di Milano. L'istruzione è stata salvata, assieme a una serie di altre istruzioni di contorno, nello script SQL `create_auco_i.sql`. Successivamente è stata eseguita una sessione di editing manuale sulla tabella, nell'ambito della quale sono stati cambiati i nomi di tutti i comuni rendendoli tipograficamente più idonei alla lettura umana. Nello specifico, nei dati ottenuti dall'ISTAT, i nomi erano scritti con caratteri tutti maiuscoli, con apostrofi al posto degli accenti e con segni gravi al posto degli apostrofi (per esempio "TREZZO SULL`ADDA" è stato corretto in "Trezzo sull'Adda"). Come ultima operazione sono stati assegnati, sempre interattivamente ma utilizzando il software QGIS collegato a PostGIS, i codici identificativi delle zone omogenee ai singoli comuni. L'operazione è meglio descritta al paragrafo successivo che tratta nello specifico la creazione delle tabelle relative alle zone omogenee.

Le operazioni di standardizzazione — e di calcolo dei campi aggiuntivi utili per l'ambiente Superset — sono state salvate nello script SQL `create_auco_epo1.sql`. La sua istruzione principale (di creazione della tabella PostGIS `auco_epo1`) è costituita da quattro clausole `from` nidificate che eseguono, da quella più interna verso quella più esterna, le seguenti operazioni.

1. Selezione dei soli poligoni della Città metropolitana di Milano (`auco_epo1_01pg."COD_PRO" = 15`) e relazionamento con la tabella degli attributi dei comuni precedentemente creata (`left join auco_i on auco_epo1_01pg."COD_ISTATN"::integer = auco_i."AUCO_ID"`).
2. Selezione di tutti i campi così recuperati in un "alias" di tabella ("table alias" = tabella temporanea che cessa di esistere al termine dell'istruzione nell'ambito della quale è stata definita).
3. Separazione dei multi-poligoni in poligoni semplici con la funzione PostGIS `st_dump()`.
4. Standardizzazione dei campi attraverso:
 - a) calcolo dell'identificatore numerico degli elementi poligonal `EPOL_ID` con la funzione Postgres `row_number()` ordinandoli in base all'identificatore numerico comunale `AUCO_ID`;
 - b) inserimento nella tabella finale `auco_epo1` dei campi `po1` (geometrie poligonal), `AUCO_ID`, `AUCO_INAI` (la prima I nel suffisso del nome del campo indica che si tratta di istanze comunali, il frammento NA sta per *name* = nome, mentre l'ultima I indica che si tratta di nomi in lingua italiana, ipotizzando che potrebbero essere scritti anche in altre lingue, come per esempio in inglese, con E per *English*) e `AUZO_ID`;
 - c) calcolo dell'area poligonale con la funzione Postgres `st_area()` e registrando il risultato nei due campi `EPOL_IARE` (istanze poligonal, *area*, espressa in m²) ed `EPOL_IAREK` (istanze poligonal, *area*, espressa in km², dove la K finale viene aggiunta al nome di `EPOL_IARE` per simboleggiare che si tratta di una particolare "vista" di quel dato piuttosto che di un dato a sé stante e dove, con l'aggiunta della K finale, viene raggiunto anche la lunghezza massima dei nomi dei campi permessa dallo standard `dBASE III`, che ancora oggi dovrebbe essere buona norma rispettare);
 - d) costruzione della famiglia di campi `EPOL_DECK*` (sei campi, di cui uno principale e cinque derivati come "viste", in modo analogo a `EPOL_IARE` ed `EPOL_IAREK`, con i caratteri finali 1, 2, 5, A e K aggiunti alla radice del nome) che codificano le geometrie poligonal prelevate dal



Figura 1: Insieme di dati AUCO – Unità amministrative – Comuni
(tabella PostGIS auco_epo1)

campo po1 nel formato richiesto dalle librerie deck.gl dell'ambiente Superset; il formato in questione è simile al formato GeoJSON, del quale utilizza però soltanto la componente delle coordinate in formato testuale, che viene recuperata applicando la formula di conversione `st_asgeojson(geometria)::json->>'coordinates'::text`, dove il segnaposto geometria indica diverse formule di trasformazione del campo po1 con

- i) `st_transform(po1, 4236)` per il campo principale EPOL_DECK che applica semplicemente una trasformazione dal sistema di coordinate UTM 32 nord orientato al WGS84 in coordinate geografiche (latitudine/longitudine), come richiesto dalle librerie deck.gl,
- ii) `st_transform(st_simplifypreservetopology(po1, 1::double precision), 4236)` per il campo “vista” EPOL_DECK1, dove, oltre alla trasformazione di coordinate, viene anche eseguita una semplificazione (generalizzazione) del perimetro poligonale con tolleranza un (1) metro al fine di rendere le operazioni di vestizione grafica nell'ambiente Superset più celeri e dove il carattere aggiunto 1 alla radice del nome del campo EPOL_DECK indica la tolleranza di 1 m,
- iii) altre quattro operazioni di trasformazione e semplificazione geometrica per i campi “vista” EPOL_DECK2, EPOL_DECK5, EPOL_DECKA ed EPOL_DECKK con tolleranze rispettiva-

- mente di 2, 5, 10 e 20 metri (A e K indicano “cifre” nel sistema numerico esatrigesimale, vale a dire con base 36 = 10 cifre da 0 a 9 più 26 “cifre” da A a Z);
- e) costruzione del campo EPOL_JSONC contenente le coordinate, nel formato GeoJSON e in coordinate geografiche, del centroide (punto d’area posizionato in modo da ricadere all’interno del poligono, che non è da confondere con il baricentro, che può anche ricadere al suo esterno) con le funzioni PostGIS `st_pointonsurface()`, `st_transform()` e `st_asgeojson()`;
 - f) costruzione dei campi EPOL_ECOX ed EPOL_ECOY contenenti rispettivamente le coordinate *x* e *y*, espresse in longitudine e latitudine, dei centroidi poligonali, con le funzioni PostGIS `st_pointonsurface()`, `st_transform()`, `st_x()` e `st_y()`.

Oltre all’istruzione `select` principale, con la quale viene creata la tabella `auco_epol`, anche in questo caso lo script `create_auco_epol.sql` contiene una serie di altre istruzioni, come `drop table if exists auco_epol`; che elimina la tabella `auco_epol` nel caso esista già, la definizione della chiave primaria `auco_epol_pkey` sul campo `EPOL_ID`, l’indice spaziale `auco_pidx` sul campo `pol`, la definizione del commento Postgres della tabella (“Unità amministrative – Comuni”) e la definizione dei commenti Postgres su ciascuno dei campi della tabella.

In Figura 1 è rappresentato l’insieme dei dati AUCO, al quale sono stati sovrimpresi i numeri identificativi dei comuni, impostati con l’espressione `AUCO_ID % 1000` (cioè il resto della divisione di `AUCO_ID` per 1000, che equivale a sottrarre il codice 3 della Lombardia e il codice 015 della Provincia di Milano lasciando solo il numero progressivo univoco all’interno della Città metropolitana). La figura è stata prodotta con QGIS.

Zone omogenee

Nella Città metropolitana di Milano, le zone omogenee sono costituite per aggregazione di territori comunali, mentre *non* sono state istituite zone omogenee subcomunali (per esempio a Milano) o che aggregassero tra loro parti di comuni. Dal punto di vista della tecnica geoinformatica — sebbene nei dati ISTAT o anche negli stessi limiti comunali della Regione Lombardia non sia presente l’attributo dell’appartenenza alle zone omogenee — la genesi del dato territoriale poligonale è piuttosto semplice ed è stata eseguita con il passaggi seguenti.

1. Come prima operazione è stata costruita una tabella identificativa delle istanze delle zone omogenee. Secondo le regole di denominazione applicate al database, l’insieme di dati di riferimento è stato chiamato AUZO (“Unità amministrative – Zone omogenee”) e, pertanto, la sua tabella identificativa delle istanze `auzo_i`. La tabella contiene due campi: un codice identificativo numerico `AUZO_ID` (numero intero) e il nome italiano della zona omogenea `AUZO_INAI` (testo con 32 caratteri, dove la prima *I* indica “istanze” come in `ID`, `NA` sta per “name” e la *I* finale per la lingua italiana, ipotizzando che tutti i nomi potrebbero teoricamente essere rappresentati anche in altre lingue). La tabella è stata creata tramite lo script SQL `create_auzo_i.sql` che verifica l’esistenza della tabella e in caso affermativo la cancella e la ricrea, aggiungendovi una chiave primaria sul campo identificativo e delle descrizioni Postgres della tabella stessa e dei due cam-

pi. Come ultima istruzione, sotto forma di dati codificati in modo statico (“hard-encoded”) direttamente nello script, vengono inseriti otto record con le coppie di valori dei campi AUZO_ID e AUZO_INAI corrispondenti alle otto zone omogenee della Città metropolitana. I codici identificativi assegnati alle singole zone si conformano allo schema di denominazione *rpppzz*, dove *r* è un numero a una o due cifre indicante il codice ISTAT della Regione (qui sempre 3 per la Lombardia), *ppp* è il codice ISTAT della Provincia (per la questione di retro-compatibilità citata in precedenza) e *zz* è il numero progressivo della Zona omogenea. Non esistendo alla data attuale una numerazione ufficiale delle zone omogenee, se n’è adottata una con ordinamento geografico a partire dalla ZO di Milano (ID 301501) corrispondente al solo Comune di Milano e poi in senso orario ponendo il nord in alto: Nord Milano; Adda Martesana; Sud Est; Sud Ovest; Nord Ovest; Magentino e Abbiatense; Alto Milanese. L’ultima ZO elencata è identificata dal codice 301508.

2. Come seconda operazione è stata costruita una tabella delle corrispondenze (“look-up table”) tra numeri identificativi dei comuni e numeri identificativi delle zone omogenee. Dato che la tabella identificativa delle istanze dei comuni *auco_i* utilizza la stessa chiave primaria (numero identificativo comunale AUZO_ID) di questa ipotetica tabella delle corrispondenze — e dato che sia la tabella identificativa dei comuni, sia la tabella delle corrispondenze hanno ciascuna un solo campo aggiuntivo (AUZO_INAI per il nome italiano del comune nella tabella *auco_i* e AUZO_ID per il numero identificativo della zona omogenea nella tabella delle corrispondenze) — è parso logico e doveroso in termini di normalizzazione del database relazionale riunire le due tabelle in una sola, aggiungendo semplicemente il campo AUZO_ID alla tabella *auco_i*. L’operazione è stata eseguita manualmente aggiungendo dapprima il campo AUZO_ID con l’istruzione SQL `alter table auco_i add column "AUZO_ID" integer` e poi, con il software QGIS collegato al database Postgres, selezionando geograficamente e interattivamente i comuni appartenendo alle singole zone omogenee e assegnando in blocco i relativi numeri identificativi.
3. Come terza operazione sono stati fusi tra loro i poligoni comunali dell’insieme *auco_i* appartenenti alla stessa zona omogenea. Le istruzioni SQL necessarie a tal fine sono state riunite nello script `create_auzo_epol.sql` che elimina — nel caso esista già — e crea o ricrea la tabella *auzo_epol*, che è l’implementazione PostGIS dell’insieme di dati (astratto) AUZO, costruisce la chiave primaria *auzo_epol_pkey* sul campo *EPOL_ID* (numero identificativo dell’elemento poligonale), crea l’indice spaziale *auzo_pidx* (*index poligonale*) sul campo *pol* (geometria poligonale, che nelle convenzioni PostGIS viene generalmente denominato *geom* o *the_geom*, ma che qui si è preferito distinguere per il fatto che alcuni insiemi di dati sono caratterizzati dalla compresenza di più campi geometrici, come *mp1* per le geometrie multi-poligonali o *cnt* per i centroidi poligonali o “punti d’area” secondo la terminologia SDTS dello Spatial Data Transfer Standard), imposta il commento Postgres della tabella a “Unità amministrative – Zone omogenee” e imposta i commenti Postgres anche a tutti e 15 i campi attributi: *EPOL_ID* = “ID del poligono (istanza) – chiave primaria”; *pol* = “Geometria poligonale”; *AUZO_ID* = “ID della Zona

omogenea (istanza)”; AUZO_INAI = “Nome italiano della Zona omogenea”; EPOL_IARE = “Area del poligono in m²”; EPOL_IAREK = “Area del poligono in km²”; EPOL_DECK = “Geometria poligonale nel formato deck.gl per Superset (in coordinate geografiche)”; EPOL_DECK1 = “Geometria poligonale nel formato deck.gl per Superset (in coordinate geografiche) – semplificata con tolleranza 1 m”; EPOL_DECK2 = “Geometria poligonale nel formato deck.gl per Superset (in coordinate geografiche) – semplificata con tolleranza 2 m”; EPOL_DECK5 = “Geometria poligonale nel formato deck.gl per Superset (in coordinate geografiche) – semplificata con tolleranza 5 m”; EPOL_DECK10 = “Geometria poligonale nel formato deck.gl per Superset (in coordinate geografiche) – semplificata con tolleranza 10 m”; EPOL_DECK20 = “Geometria poligonale nel formato deck.gl per Superset (in coordinate geografiche) – semplificata con tolleranza 20 m”; EPOL_JSONC = “Geometria puntuale del centroide poligonale nel formato GeoJSON (in coordinate geografiche)”; EPOL_ECOX = “Coordinata x del centroide poligonale (in coordinate geografiche)”; EPOL_ECOY = “Coordinata y del centroide poligonale (in coordinate geografiche)”. L’istruzione di creazione della tabella preleva i dati dall’insieme di dati dei poligoni comunali (insieme AUZO, implementato in PostGIS sotto forma della tabella `auco_epo1`) e dalla tabella identificativa delle zone omogenee creata al primo passaggio (il campo AUZO_ID degli identificatori delle zone omogenee era nel frattempo stato incluso nella tabella `auco_epo1` in virtù del suo aggiornamento a seguito della semplice riesecuzione dello script `create_auco_epo1.sql`, quale principio portante delle attività di evoluzione nell’ambito del Progetto DeSK), crea una relazione esterna sinistra (“left outer join”) tra `auco_epo1` (tabella di sinistra) e `auzo_i` (tabella di destra) sul campo AUZO_ID sul quale — assieme al campo AUZO_INAI — vengono eseguiti il raggruppamento (clausola `group by`) e la funzione di aggregazione poligonale di PostGIS `st_union()` che, in una clausola `from` più esterna dell’istruzione `select` (nidificata in `create table auzo_epo1 as select...`), viene poi suddivisa tramite la funzione `st_dump()` per garantire la creazione di poligoni semplici anziché multi-poligoni (la Zona omogenea Sud Est verrebbe restituita sotto forma di multi-poligono in quanto il territorio del Comune di San Colombano al Lambro è geograficamente distaccato dal resto del territorio della Zona omogenea, come anche dallo stesso territorio metropolitano). Nell’istruzione `select` nidificata vengono inoltre calcolati l’identificatore numerico poligonale EPOL_ID tramite la funzione Postgres `row_number()`, l’area poligonale con la funzione PostGIS `st_area()` e le geometrie nei formati `deck.gl` per l’ambiente software *Superset*, *GeoJSON* nonché le coordinate geografiche dei centroidi generati “al volo” tramite la funzione PostGIS `st_pointonsurface()`. Per i dettagli si rimanda allo studio dello script `create_auzo_epo1.sql`.

In Figura 2 è rappresentato l’insieme dei dati AUZO, al quale sono stati sovrimpressi per migliorarne la leggibilità i codici (campo AUZO_ID) e i nomi (campo AUZO_INAI) delle zone omogenee nonché, con tratto molto sottile, i confini comunali dell’insieme AUZO. La figura è stata prodotta con QGIS.

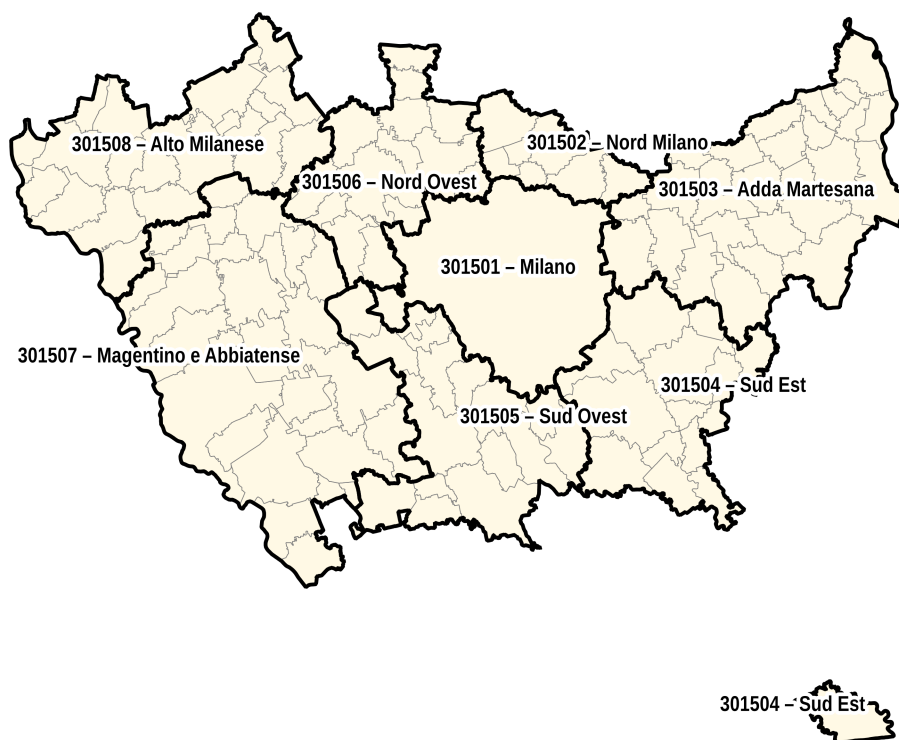


Figura 2: Insieme di dati AUZO – Unità amministrative – Zone omogenee (tabella PostGIS auzo_epo1)

Dati censuari

Nell'ambito delle attività del sistema DECIMETRO (Sistema Decisionale della Città Metropolitana di Milano) e delle fasi precedenti del Progetto DeSK erano già stati utilizzati ed elaborati dati censuari relativi alla Città metropolitana di Milano, alle quali si è avuto accesso anche durante la fase del progetto e nell'ambito dell'incarico professionale qui documentati. Tuttavia, al fine di garantire il principio della trasparenza delle procedure durante la fase di evoluzione delle pratiche informative del Progetto DeSK, si è ritenuto indispensabile ripartire con le elaborazioni dai dati originali anziché utilizzando quelli già trasformati e dei quali non erano più disponibili i passaggi intermedi del trattamento o una documentazione esaustiva su tutte le operazioni eseguite. Pertanto, si è deciso di recuperare i dati in questione direttamente dall'ISTAT.

L'ISTAT mette i dati censuari a disposizione sotto forma di dati distinti in dati geometrico-identificativi (poligoni delle sezioni censuarie con identificatori numerici di sezioni e altre aggregazioni territoriali) e dati attributi (variabili censuarie), a loro volta separati in due gruppi di tabelle distinte, uno per il censimento della popolazione e delle abitazioni e l'altro per il censimento dell'industria e dei servizi. In questo modo, si è partiti da tre insiemi di dati facenti tutti e tre parte della serie "Censimento 2011", alla quale è stata associato l'identificatore ZK11 (Z per dati censuari e K11 per l'anno

2011, con K che rappresenta il numero 20 nel sistema numerico esatrigesimale). Nello specifico gli insiemi istanziati sono ZK11_EPOL per i dati geometrico-identificativi (“Sezioni di censimento”, ottenuti dall’ISTAT sotto forma di file nel formato Shapefile), ZK11_VINS per i dati attributi (“variabili” nella terminologia ISTAT) del censimento dell’industria e dei servizi del 2011 (messi a disposizione da parte dell’ISTAT sotto forma di tabella nel formato Microsoft Excel) e ZK11_VPOA per i dati attributi del censimento della popolazione e delle abitazioni del 2011 (messi a disposizione sotto forma di tabella nel formato testuale CSV – “comma-separated values”, in italiano “valori separati da virgole”). I dati sono stati salvati sul disco fisso di un computer locale e poi trasferiti sul database Postgres utilizzando procedure simili a quelle già descritte in dettaglio per i dati comunali.

Sezioni

Per i dati geometri (sezioni di censimento) è stata utilizzata una procedura di caricamento del database Postgres del tutto analoga a quella dei poligoni comunali, dalla quale si differenzia soltanto per i nomi dei file e le posizioni nelle cartelle locali, nonché per il fatto che per i dati poligonali ottenuti dall’ISTAT era possibile una conversione immediata in poligoni semplici, anziché multi-poligoni, come imporrebbero i parametri predefiniti della conversione tra il formato Shapefile e PostGIS. La chiamata al programma shp2pgsql conteneva pertanto un ottavo parametro -S, inserito tra gli switch -I e -t, che forza la creazione di poligoni semplici o che ferma l’esecuzione del programma nel caso la procedura incontri una struttura geometrica che impone la gestione sotto forma di multi-poligono. La procedura, salvata nello script batch load_zk11_epol_01pg.bat, crea una tabella PostGIS poligonale con 26 campi, di cui 22 campi ISTAT, un campo creato nell’ambito della procedura di importazione (l’identificatore numerico poligonale EPOL_ID), due campi provenienti dal formato Shapefile (Shape_Leng e Shape_Area) nonché il campo geometrico pol. L’insieme di dati contiene 53.173 record (poligoni) e si riferisce ancora a tutto il territorio regionale lombardo.

In un secondo passaggio, di standardizzazione dei nomi e dei tipi dei campi salvato nello script create_zk11_epol_02st.sql, è stata creata la struttura corrispondente alle convenzioni di denominazione adottate in questa sede. Per una chiarezza documentale soprattutto della corrispondenza dei nomi dei campi, si porta qui sotto integralmente l’istruzione SQL principale dello script.

```
create table zk11_epol_02st as
select
  row_number() over (order by "EPOL_ID") as "EPOL_ID",
  pol,
  "COD_REG"::smallint      as "ZK11_RD",
  "COD_ISTAT"::integer     as "AUIT_CD",
  "PRO_COM"::integer       as "ZK11_CD",
  "SEZ2011"::bigint        as "ZK11_SD",
  "SEZ"::integer           as "ZK11_CSEZ",
  "COD_STAGNO"::integer    as "ZK11_XSTD",
  "COD_FIUME"::integer     as "ZK11_XFID",
  "COD_LAGO"::integer      as "ZK11_XLOD",
  "COD_LAGUNA"::integer    as "ZK11_XLND",
  "COD_VAL_P"::integer     as "ZK11_XVPD",
```



```
"COD_ZONA_C"::smallint as "ZK11_XZCD",
"COD_IS_AMM"::smallint as "ZK11_XIAD",
"COD_IS_LAC"::integer as "ZK11_XILD",
"COD_IS_MAR"::integer as "ZK11_XIMD",
"COD_AREA_S"::integer as "ZK11_XASD",
"COD_MONT_D"::integer as "ZK11_XMDD",
"LOC2011"::bigint as "ZK11_LD",
"COD_LOC"::integer as "ZK11_CLOC",
"TIPO_LOC"::smallint as "ZK11_LTD",
"COM_ASC"::integer as "ZK11_QD",
"COD_ASC"::smallint as "ZK11_CQSC",
(("PRO_COM"*1000) + "ACE")::integer as "ZK11_AD",
"ACE"::integer as "ZK11_CACE"
from zk11_epol_01pg;
```

Oltre all'istruzione qui elencata, lo script contiene anche le solite istruzioni di verifica se la tabella esiste già, di creazione della chiave primaria (zk11_epol_02st_pkey sul campo EPOL_ID), di creazione dell'indice spaziale (zk11_pidx_02st sul campo pol) nonché di impostazione dei commenti Postgres della tabella e di ciascuno dei suoi campi.

Il terzo e ultimo passaggio di manipolazione dell'insieme ZK11_EPOL è consistito nella mera selezione delle sezioni di censimento della Città metropolitana di Milano. L'istruzione

```
select *
from zk11_epol_02st
where ("AUIT_CD" / 1000)::integer = 3015;
```

è stata inserita nello script create_zk11_epol.sql assieme alle altre istruzioni consuete. La tabella PostGIS risultante zk11_epol è pertanto costituita dai medesimi 26 campi attributi, ma solo dai 14.309 poligoni della Città metropolitana di Milano.

Variabili

Nel caso delle tabelle di attributi si trattava di caricare nel database Postgres file CSV. Gli attributi del censimento dell'industria e dei servizi (salvato in locale con il nome z2011ast_zk11_vpoa_00ac.csv) era già fornito in quel formato, mentre gli attributi del censimento della popolazione e delle abitazioni, fornito sotto forma di file Microsoft Excel (salvato in locale come z2011ast_zk11_vins_00ac.xls) doveva necessariamente essere prima convertito — manualmente — nel formato CSV, attribuendogli poi il nome z2011ast_zk11_vins_01cv.csv. A parte questa differenza iniziale, la procedura in seguito è stata concettualmente identica e mostra alcune somiglianze con quelle adoperate per trasferire i dati dal formato Shapefile nel database Postgres. Il trasferimento avviene anche qui chiamando da riga di comando con parametri il programma psql facendo reindirizzare sul suo standard input (stdin) una serie di istruzioni SQL di creazione della tabella e di inserimento, tramite l'istruzione copy, dei record letti dal file CSV. Per l'insieme ZK11_VINS, le istruzioni sono inserite nello script load_zk11_vins_02pg.bat, mentre per l'insieme ZK11_VPOA nello script load_zk11_vpoa_01pg.bat. Inizialmente, i due script chiamano system_vars.bat, impo-

stano una serie di variabili specifiche, per poi cominciare a popolare passo dopo passo il file temporaneo `tmp.sql` con le istruzioni SQL che nell'ultima della riga dello script vengono passati al programma `psql` con l'opzione `-t tmp.sql`. Come per i poligoni comunali, anche qui vengono forniti i parametri di connessione (la password di accesso al database `lm_milano` deve tuttavia essere fornita interattivamente) direttamente sulla riga di comando. Per chiarezza si riporta qui di seguito la parte centrale dello script.

```
echo set client_encoding to utf8;> tmp.sql
echo set standard_conforming_strings to on;>> tmp.sql
type %_qpath%\create_%_tname%.sql>> tmp.sql
echo[>> tmp.sql
echo[>> tmp.sql
echo copy %_tname% (>> tmp.sql
echo     "TIPO_SOGGETTO",>> tmp.sql
echo     "CODREG", "REGIONE", "CODPRO", "PROVINCIA", "CODCOM", "COMUNE",>> tmp.sql
echo     "PROCOM", "SEZ2011", "NSEZ", "ACE", "CODLOC", "CODASC",>> tmp.sql
echo     "NUM_UNITA", "ADDETTI", "ALTRI_RETRIB", "VOLONTARI">> tmp.sql
echo     )>> tmp.sql
echo from stdin>> tmp.sql
echo with (format csv, delimiter ';', header true);>> tmp.sql
type %_fpath%\%_fname%>> tmp.sql
echo \.>> tmp.sql
echo analyze %_tname%;>> tmp.sql
```

```
%_ppath%\psql -h %_phost% -d %_pname% -U %_puser% -f tmp.sql
```

Come si vede, sulla terza riga viene incluso anche il contenuto di uno script SQL predisposto in via preliminare e che contiene l'istruzione di creazione della tabella in questione (`zk11_vins_02pg` oppure `zk11_vpoa_01pg`, a seconda dello script specifico) con la definizione dei suoi campi, con i nomi incontrati nel file CSV, ma ancora senza i dati. Come in altri script di creazione, anche qui vengono creati chiavi primarie, indici e commenti Postgres. Dalla sesta alla tredicesima riga viene richiamato il comando SQL `copy`, con il quale vengono copiati i valori da una sorgente nella tabella Postgres e che contiene, tra le due parentesi, l'elenco dei campi ai quali si riferiscono i valori. Tali valori vengono scritti dal file CSV sullo standard output (`stdout`) attraverso il comando batch `type`, che viene a sua volta reindirizzato come standard input (`stdin`) sul programma `psql`. Il flusso di standard output/input a seguito del comando `copy` viene terminato con il carattere punto (`.`), qui preceduto da un carattere di escape, la barra inversa (`\`). L'esempio qui riportato si riferisce ai dati del censimento dell'industria e dei servizi, ma lo script relativo ai dati del censimento della popolazione e delle abitazioni si differenzia soltanto per l'elenco dei campi. Anche lo script SQL richiamato dallo script batch si differenzia soltanto per la definizione dei campi.

Le due tabelle `zk11_vins_02pg` e `zk11_vpoa_01pg` così importate hanno rispettivamente 18 e 153 campi, tra cui sempre gli identificatori numerici dei record di dati `RDAT_ID`, che sono stati creati dagli script SQL di creazione delle tabelle. Le tabelle hanno inoltre rispettivamente 65.373 e 48.220 record, riferendosi entrambe ancora a tutto il territorio regionale, ma con un numero differente di

sezioni prive di dati. A partire da queste due tabelle sono state costruite altre due tabelle denominate `zk11_vins` e `zk11_vpoa` — cioè senza suffisso di versione perché considerate versioni finali — contenenti soltanto i campi di dati necessari ai fini delle elaborazioni successive e limitando i dati alle sole sezioni della Città metropolitana di Milano. Per la tabella `zk11_vpoa` del censimento della popolazione e delle abitazioni si tratta dei seguenti tre campi.

`RDAT_ID` ID del record (istanza) – chiave primaria;

`ZK11_SD` ID della Sezione di censimento (istanza);

`ZK11_SPT0` P1 – Popolazione totale.

Invece per la tabella `zk11_vins` del censimento dell'industria e dei servizi si tratta dei seguenti sette campi.

`RDAT_ID` ID del record (istanza) – chiave primaria;

`ZK11_SD` ID della Sezione di censimento (istanza);

`ZK11_SANU` Aziende – Numero di unità;

`ZK11_SOAD` Occupati – Addetti;

`ZK11_SOAR` Occupati – Altri retribuiti;

`ZK11_SOV0` Occupati – Volontari;

`ZK11_SOT0` Occupati – Totale.

I numeri di record delle due tabelle risultanti sono rispettivamente 13.557 e 19.759. In un ultimo passaggio, salvato nello script `create_zk11_epo1_v.sql`, è stata creata una tabella poligonale espansa delle sezioni di censimento contenente anche i sei campi di dati attribuiti su popolazione (`ZK11_SPT0`), aziende (`ZK11_SANU`) e occupati (`ZK11_SOAD`, `ZK11_SOAR`, `ZK11_SOV0` e `ZK11_SOT0`). La necessità della creazione di una tabella siffatta, non normalizzata, risiede nella macchinosità delle procedure Postgres/PostGIS per il relazionamento “al volo” in fase di visualizzazione dei dati. L'istruzione di creazione utilizza un doppio relazionamento tra `zk11_epo1` da una parte e le tabelle `zk11_vpoa` e `zk11_vins` dall'altra utilizzando il campo identificatore comune `ZK11_SD` a tutti e tre le tabelle.

Popolazione 1951-2018

L'Obiettivo di sviluppo sostenibile (SDG, Sustainable Development Goal) 11 delle Nazioni Unite, intitolato “Città e comunità sostenibili – Rendere le città e gli insediamenti urbani inclusivi, sicuri, resilienti e sostenibili” sono definiti dieci obiettivi specifici, per ciascuno dei quali sono stati definiti uno o più indicatori. Il sotto-obiettivo 3 — generalmente indicato con l'espressione “SDG 11.3” — è quello più pertinente per la problematica qui posta e al quale anche l'Unione Europea ha dedicato varie attività, specialmente attraverso il suo Comitato delle Regioni, come per esempio nello sviluppo di strumenti per la TIA (Territorial Impact Analysis = Analisi di impatto territoriale) applicata alle aree urbane, dove i territori di riferimento corrispondono principalmente alle aree funzionali urbane (functional urban areas). La descrizione del sotto-obiettivo è, nella sua versione originale in lingua inglese “By 2030, enhance inclusive and sustainable urbanization and capacity for participat-

ory, integrated and sustainable human settlement planning and management in all countries” (in italiano “Entro il 2030 incrementare l’urbanizzazione inclusiva e sostenibile e la capacità di una pianificazione e gestione degli insediamenti umani sostenibile in tutti i paesi”). Per l’SDG 11.3, le Nazioni Unite hanno definito due indicatori: 11.3.1 “Rapporto tra consumo di suolo e crescita demografica” e 11.3.2 “Proporzione di città con una struttura di partecipazione diretta della società civile nella pianificazione e gestione urbanistica che opera regolarmente e democraticamente”. Le traduzioni dall’inglese a cura dell’autore della presente relazione. L’indicatore 11.3.2 è privo di significato per la realtà lombarda e milanese, in quanto si può partire dal presupposto che tutte le città — in teoria anche tutte le città italiane — sono dotate di *strutture di partecipazione diretta della società civile nella pianificazione e gestione urbanistica che operano regolarmente e democraticamente*. Pertanto, l’indice in questione risulterebbe pari a 1 (valore massimo possibile) per tutti i comuni della Città metropolitana. Non è stato così in passato, ma la ricostruzione di questo indice in chiave storica per ciascuno dei comuni della Città metropolitana fuoriuscirebbe chiaramente dal quadro dell’incarico al quale si riferisce la presente relazione. Invece la conoscenza dei valori che assume l’indicatore 11.3.1 nei vari comuni nelle varie annate è utile conoscere ed è possibile calcolarlo.

Per il calcolo effettivo dell’indicatore — qui ribattezzato “indice del consumo di suolo ponderato” — era necessario disporre, oltre a dati spaziali sull’uso e consumo di suolo in un arco di tempo più ampio possibile, di dati demografici (popolazione residente o presente) per lo stesso periodo di tempo. Dati sull’uso e consumo di suolo sono disponibili attraverso la serie storica 1954-2015 del DUSAF (Destinazione d’uso dei suoli agricoli e forestali) lombardo e, più precisamente, per le annate 1954, 1980, 1999, 2007, 2009, 2012 e 2015. I valori di incremento del suolo “consumato” — quantificabile attraverso il dato proxy delle quantità di suolo impermeabilizzato (a volte, l’impermeabilizzazione del suolo viene utilizzato a tutti gli effetti come sinonimo di consumo di suolo) — tra due annate consecutive (dato che il dato del 1980 riporta un livello di dettaglio molto inferiore a quello delle altre annate, si sono calcolati anche gli incrementi medi annui dal 1954 direttamente sul 1999, ampliando però così sensibilmente l’arco temporale abbracciato) devono essere messe a confronto con i valori di incremento della popolazione, per la quale è possibile predisporre una serie storica più densa, con dati a frequenza almeno annuale.

È stato possibile ottenere dati continui dal censimento della popolazione del 1951 (con data convenzionale 4 ottobre) fino al 31 dicembre 2018, con i dati dei censimenti del 1961, 1971, 1981, 2001 e 2011, nonché degli aggiornamenti a cura dell’ISTAT al 1° gennaio di ciascun anno dal 1981 al 1991 e al 31 dicembre di ciascuno anno dal 1991 al 2018. Il problema principale nella ricostruzione storica dei dati demografici è consistito nelle variazioni territoriali delle circoscrizioni comunale intervenute dal 1951 (o 1954) fino a oggi, con l’ultima delle variazioni avvenuta l’8 febbraio 2019 quando i comuni di Vermezzo e Zelo Surrigone si sono riuniti nel nuovo Comune di Vermezzo con Zelo. Utilizzando anche altre fonti (per molti versi sono state molto utili le ricostruzioni alle date censuarie recuperate dalla Wikipedia italiana) è stato possibile, con un certo grado di approssimazione, ricostruire le popolazioni comunali per tutto il periodo preso in esame. L’unica eccezione è costituita dal Comune di Baranzate, per il quale l’operazione non è riuscita e, pertanto, nel calcolo

dell'indice del consumo di suolo ponderato, il territorio baranzatese continua a far parte di quello di Bollate, dal quale si era staccato in via definitiva l'8 giugno 2004. È comprensibile che nella popolazione in questione — viste le forti tensioni emotive registrate negli anni immediatamente prima del distacco comunale — tale compromesso non incontrerà molti favori, ma una ricostruzione più accurata avrebbe richiesto tempi ed energie qui non disponibili.

Dopo aver scaricato dal sito Internet dell'ISTAT tutti i file di dati necessari — disponibili principalmente sotto forma di tabelle Microsoft Excel, ma con strutture non completamente identiche — il lavoro sui dati è stato eseguito con fogli di calcolo LibreOffice Calc in quanto era necessario eseguire diverse prove sulla coerenza dei dati che sono più agevoli da eseguire in modo interattivo. La versione finale della tabella — denominata `mpophhet_02xp.odt` (`m` = dati *modellati*, `poph` = *popolazione storica-historical*, `he` = autore *HESC*, `t` = forma *tabellare*, `02` = versione 2, `xp` = versione per *esportazione-export* in altro formato) e poi definitivamente convertita nel formato CSV con il nome `mpophhet_03cv.csv` (versione 3, conversione di formato) — è stata caricata nel database Postgres in modo interattivo con QGIS, dove ha assunto la denominazione `poph_i` (`i` per *istanze*).

Isolati urbani

L'insieme di dati degli isolati urbani — identificato con la sigla URBL per “urban block” — è utile per la costruzione di alcuni indicatori morfo-demografici a livello micro-urbano (al di sotto della sezione di censimento e al netto delle superfici stradali e non edificate/urbanizzate). Nell'ambito del sistema DECIMETRO e delle fasi del Progetto DeSK precedenti a quella presente era già stato costruito un insieme di dati poligonale rappresentante gli isolati urbani. In base ai principi del trattamento dei dati esposti all'inizio del presente capitolo, anche l'insieme URBL sarebbe dovuto essere rigenerato a partire dai suoi dati originali. Tuttavia, dalla documentazione a disposizione al momento della costruzione del database di evoluzione non sono emersi elementi conoscitivi sufficienti per poter replicare la creazione dell'insieme di dati e documentare le relative procedure informatiche e analitiche. Si è inoltre constatato come una considerevole quantità di edifici — anche soltanto tenendo in considerazione i centroidi degli edifici — si trovano all'esterno di qualunque isolato urbano, per cui il livello di significatività dell'informazione associata agli indicatori morfo-demografici costruiti sulla base degli isolati è relativamente bassa. Pertanto si è ritenuto sufficiente utilizzare l'insieme dei dati degli isolati urbani “così com'erano” (“as is”), importando semplicemente il dato dal formato File Geodatabase (GDB, formato proprietario di ArcGIS di ESRI che consiste in una cartella con estensione convenzionale `*.gdb` contenente potenzialmente una moltitudine di dati spaziali e attributi) nel database PostGIS, assegnandogli il nome di tabella `urb1_epo1`.

Anche nel caso degli isolati urbani è stato necessario effettuare questa importazione in più passaggi. L'insieme di dati era disponibile in congiunto (nella medesima cartella GDB) con altri dati di base e analitici, per cui è stato acquisito nella cartella locale del progetto GIS tutto il Geodatabase, assegnandogli il nome `bimdscoo_00ac.gdb` (`b` = dati di *base* acquisiti da altre fonti, `imds` = *indicatori morfo-demo-sociologici*, `co` = autore *Corvallis*, `o` = combinazioni di oggetti spaziali vari, “other”, `00` = versione “zero”, `ac` = insieme di dati *acquisito*). Dalla cartella GDB è quindi stato

estratto l'insieme degli isolati che era denominato `its_03_isolati_idx`. La cartella conteneva anche altri undici insiemi di dati che erano irrilevanti ai fini qui posti. Il trasferimento sul database Postgres è avvenuto in modo interattivo con QGIS. Prima di poter effettuare il trasferimento in sé era però necessario salvare l'insieme di dati in un formato intermedio in quanto il caricamento diretto nel database Postgres falliva a causa di incoerenze tra i dati geometrici dichiarati nei file (multi-poligoni) e i dati geometrici effettivamente incontrati (alcuni elementi erano del tipo multi-superficie). Il formato intermedio scelto era Spatialite, attraverso il quale era stato possibile ripianare queste inconsistenze. È stato pertanto necessario salvare in locale un ulteriore file di dati intermedio, al quale è stato assegnato il nome `burb1coo_01s1` (01 = versione 1, s1 = importazione nel formato Spatialite).

Dopodiché era possibile eseguire la procedura di caricamento interattivo con QGIS. I parametri applicati, diversi da quelli predefiniti, riguardavano il nome della tabella PostGIS da creare (`urb1_epo1_02pg` anziché una combinazione tra il nome della cartella GDB e il singolo insieme), il nome del campo identificativo da creare e utilizzare come chiave primaria (`EPOL_ID` anziché `id`), il nome del campo contenente i dati geometrici (`mp1` anziché `geom`), la sostituzione della tabella nel caso esista già (per impostazione predefinita la procedura di importazione si ferma con un messaggio di errore nel caso esista già una tabella con lo stesso nome), la creazione di un indice spaziale (per impostazione predefinita la procedura non crea un indice spaziale) e il commento Postgres che è stato impostato su "Isolati urbani – Importato da Spatialite con QGIS". Le altre opzioni di importazione (nessuna limitazione degli elementi da importare, schema "public", sistema di riferimento delle coordinate di origine e destinazione letta dal file d'origine, codifica UTF-8, nessuna conversione dei nomi dei campi in caratteri minuscoli) erano state lasciate con l'impostazione predefinita.

La tabella PostGIS risultante contiene, oltre all'identificatore multi-poligonale `EPOL_ID` creato durante la procedura di importazione e ai campi `shape_length` e `shape_area` creati precedentemente dai prodotti software ESRI, soltanto campi ausiliari e finali relativi agli indici calcolati nell'ambito delle attività del sistema DECIMETRO e durante le prime fasi del Progetto DeSK. Non è invece presente alcun campo identificatore che consenta di risalire alla posizione amministrativa (comune, attraverso il codice ISTAT) o censuaria (sezione di censimento) del singolo isolato urbano. Dato che gli indicatori erano tutti da ricalcolare a partire dai dati morfologici e demografici, era possibile rimuovere nel prosieguo delle operazioni tutti i campi attribuiti presenti. L'operazione successiva, salvata nello script `create_urb1_epo1.sql`, si limita pertanto a suddividere — laddove necessario — i multi-poligoni in poligoni semplici e a ricalcolare l'identificatore numerico poligonale `EPOL_ID`. Inoltre, non era necessario calcolare, come nel caso dei poligoni comunali, i campi geometrici nei formati GeoJSON e deck.gl, in quanto il numero elevato di elementi geometrici (26.020 multi-poligoni, poi convertiti in 26.043 poligoni semplici) non avrebbe consentito una visualizzazione in tempi ragionevoli nell'ambiente Superset. Era invece necessario includere dall'insieme di dati preesistente un indicatore che, data l'impossibilità di rieseguire tutte le operazioni di analisi spaziale necessari per replicarne la costruzione a causa della mancanza di una documentazio-

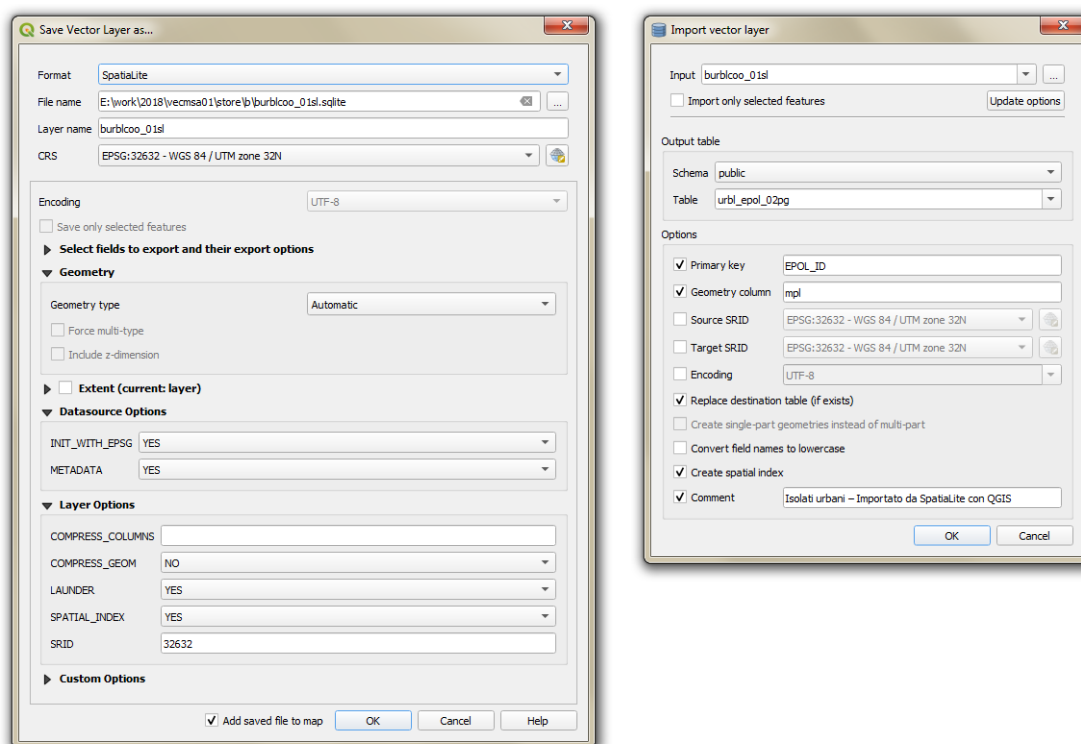


Figura 3: Parametri delle procedure di esportazione, eseguite con QGIS, nel formato SpatiaLite e di importazione nel database PostGIS dell'insieme di dati URBL – Isolati urbani (tabella PostGIS urb1_epo1)

ne esauriente con algoritmi, formule e script, non poteva più essere ricalcolato: la media delle distanze minime tra gli edifici, contraddistinta dal simbolo \bar{d}_{\min} e dalla formula astratta

$$\bar{d}_{\min} = \sum ST_Distance(C_i C_j) / n$$

(si veda Tabella 1 per maggiori dettagli). La tabella PostGIS definitiva degli isolati urbani contiene, pertanto, soltanto i seguenti tre campi:

EPOL_ID ID del poligono (istanza) – chiave primaria;
URBL_XBDA Indicatore – Media delle distanze minime tra gli edifici;
pol Geometria poligonale.

Uso del suolo derivato dal DBTR

Con l'inizio della fase di attività del Progetto DeSK qui documentata era già disponibile un insieme di dati dell'uso del suolo — o meglio “copertura del suolo” per analogia con il “CORINE Land Cover” — della Città metropolitana di Milano. Era stato predisposto da parte di Corvallis durante una precedente fase direttamente a partire dal DBTR della Lombardia. Le operazioni che erano state effettuate sono tutte di tipo analitico (nessun intervento correttivo manuale) e ben documentate, tali

da rendere probabilmente non complicata — sebbene complessa — la replicabilità delle procedure. Poiché si tratta tuttavia di un dato statico (“Uso del suolo derivato dal DBTR 2015”) si è ritenuto superfluo rieseguire la sua genesi, trattandolo pertanto come dato di base in ingresso. Le uniche operazioni che sono state eseguite erano il caricamento nell’ambiente PostGIS dei file disponibili in locale dal formato File Geodatabase (GDB) attraverso l’interfaccia QGIS la strutturazione e normalizzazione degli attributi nonché l’applicazione di una convenzione di denominazione di tabelle e attributi. Parte di queste operazioni era anche la fusione, in termini sia di attributi, sia di poligoni, dei due Comuni di Vermezzo e Zelo Surrigone nel nuovo Comune di Vermezzo con Zelo, istituito l’8 febbraio 2019, e cioè dopo che era stato predisposto il dato dell’uso del suolo.

L’operazione di caricamento nell’ambiente Postgres è avvenuta in modo interattivo con QGIS., specificando tre parametri di importazione: nome del campo che funge da chiave primaria (“EPOL_ID”); nome del campo geometrico (“mp1” per “multi-poligono”); creazione di un indice spaziale (“sì”). La tabella PostGIS così creata era stata denominata luco_epol_01pg (“lu” per “land-use”, “co” per “Corvallis”, “epol” per “elementi poligonali”, “01” per “1° passaggio” e “pg” per “importazione in Postgres/PostGIS”).

Il secondo passaggio di strutturazione ha poi riguardato una serie di interventi sul dataset: applicazione della convenzione di denominazione; separazione tra attributi identificativi e informativi (con conseguente disarticolazione dell’insieme di dati in un insieme poligonale con attributi identificativi, da una parte, e una tabella di attributi normalizzata dall’altra); separazione dei multi-poligoni in poligoni semplici; riduzione (preventiva) a sole due dimensioni (x e y); associazione dei codici comunali post-fusione a quelli originali (tramite la tabella delle corrispondenze auco_i0 predisposta per l’occasione); fusione dei poligoni in base alla classe di uso del suolo (campo LUMI_L3D) e al codice comunale post-fusione (campo AUCO_ID); ricalcolo del campo della chiave primaria e calcolo dell’area di ciascun poligono. Per una maggiore trasparenza si riporta qui la relativa istruzione SQL:

```
create table luco_epol as
select
  row_number() over (order by j."AUCO_ID") as "EPOL_ID",
  st_area(j.pol) as "EPOL_IARE",
  j."LUMI_L3D",
  j."AUCO_ID",
  j.pol
from (
  select
    d."LUMI_L3D",
    d."AUCO_ID",
    st_union(d.pol) as pol
  from (
    select
      (st_dump(st_force2d(mp1))).geom as pol,
      "L3_Code"::smallint as "LUMI_L3D",
      key_field::integer as "AUCO_ID"
```

```
from luco_epol_01pg as l
left join auco_i0 as c
on l."L3_Code"::smallint = c."AUCO_I0D"
) as d
group by "AUCO_ID", "LUMI_L3D"
order by "AUCO_ID", "LUMI_L3D"
) as j;
```

L'istruzione, assieme ad altre istruzioni collaterali (rimozione della tabella nel caso esista già, creazione della chiave primaria, creazione dell'indice spaziale e definizione di commenti per la tabella e per i campi), è conservata nello script SQL `create_luco_epol.sql`. In quanto risultato dell'ultima operazione di trattamento, la tabella non porta il suffisso di versione che, a rigor di logica, potrebbe essere "02st", con "02" indicante il passaggio numero 2 e "st" l'operazione di strutturazione.

Parallelamente alla creazione della versione finale dell'uso del suolo poligonale, contenente gli attributi metrici e identificativi, è stata creata la tabella degli attributi normalizzata `luco_13` tramite un'istruzione SQL piuttosto semplice che seleziona tre attributi informativi, oltre all'identificatore numerico della classe di 3° livello e a quello della corrispondente classe di 2° livello, e li raggruppa in base all'identificatore di 3° livello. Gli attributi informativi sono il nome italiano della classe (`LUMI_L3NI`, nella versione originale `L3_Desc`), il grado di impermeabilizzazione (`LUMI_L3RS`, originariamente `sealing`) e il grado di naturalità (`LUMI_L3RN`, originariamente `naturalita`). L'istruzione di selezione e le altre istruzioni connesse sono salvate nello script `create_luco_13.sql`.

Uso del suolo 1954-2015

Il trattamento dei dati della serie storica dal 1954 al 2015 dell'uso del suolo, derivata dalla collezione dei dati DUSAF, si è articolato nelle seguenti due fasi distinte.

1. Preparazione e strutturazione dei dati, a partire dai dati acquisiti e caricati nell'ambiente Postgres/PostGIS con successive operazioni di strutturazione, normalizzazione e incrocio spaziale con i poligoni comunali.
2. Trasformazione dei dati al fine di renderli compatibili con quelli derivati direttamente dal DB-TR ad opera di Corvallis, vale a dire riduzione da cinque a tre livelli gerarchici delle classi e redazione di tabelle comparative della serie storica.

Preparazione e strutturazione dei dati

I dati dell'uso del suolo sono stati acquisiti tramite il sito Internet della Regione Lombardia (<https://www.regione.lombardia.it>, "Servizi e Informazioni", "Enti e Operatori", "Territorio", "Sistema Informativo Territoriale (SIT)", "Uso e copertura del suolo in Lombardia") nel formato Shapefile (unico formato disponibile). Prima di caricare i dati nell'ambiente Postgres, è stato aggiunto a ciascuna delle tabelle di attributi il campo univoco `LUMI_ID` contenente la codifica numerica fino a cinque livelli delle classi di uso/copertura del suolo. L'identificativo dell'insieme di attributi `LUMI` sta per *Land-Use Milano*, in quanto nel seguito delle elaborazioni sono state applicate alcune modifiche rispetto alla strutturazione originale ad opera degli uffici regionali, come la trasformazione dell'at-

tributo da testuale in numerico, il riempimento a destra di zeri per le classi di livello superiore alla quinta (per esempio la classe 331 diventa 33100, la classe 3241 diventa 32410, mentre classi come la 31122 restano invariate), così come l'integrazione di alcune classi dei gradi di impermeabilizzazione e di naturalità che nel DBTR non erano contemplate. L'operazione è stata eseguita in locale con QGIS creando il campo intero LUMI_ID e calcolando i suoi valori con la seguente formula:

```
if(CODICE<10, CODICE*10000,  
if(CODICE<100, CODICE*1000,  
if(CODICE<1000, CODICE*100,  
if(CODICE<10000, CODICE*10, CODICE))))
```

Il campo CODICE contiene i codici in formato testuale che l'interprete di QGIS converte direttamente in numeri, per cui non erano necessarie funzioni di type casting. Nelle versioni 2012 e 2015, il campo CODICE assume rispettivamente le denominazioni IDENTIFICA e COD5. Successivamente, i file di dati così modificati sono stati caricati nel database Postgres sul server remoto utilizzando da riga di comando il programma shp2pgsql, fornito assieme al pacchetto pgAdmin. Sono state eseguite sostanzialmente le seguenti sei istruzioni, inserite in uno script (load_lu.bat) per la riga di comando di Windows:

```
%_ppath%\shp2pgsql ^  
-s 32632 ^  
-d ^  
-g pol ^  
-D ^  
-k ^  
-I ^  
-S ^  
-t 2D ^  
%_fpath%\%_fname% ^  
public.%_tname%> tmp.sql  
echo comment on table public.%_tname% is>> tmp.sql  
echo '%_tdesc%';>> tmp.sql  
echo alter table public.%_tname% rename column gid to "%_cname%";>> tmp.sql  
echo alter index public.%_tname%_%_gname%_idx rename to %_iname%;>> tmp.sql  
%_ppath%\psql -h %_phost% -d %_pname% -U %_puser% -f tmp.sql
```

Lo script è fortemente parametrizzato con variabili d'ambiente per consentire il suo riutilizzo con una varietà di dati da caricare sul server Postgres. Per questo motivo, lo stesso script, in realtà, è articolato nei seguenti quattro script distinti.

1. load_lu??_epo1_02pg.bat, una famiglia di sette script, dove ?? corrisponde all'identificativo dell'annata dell'uso del suolo (54 = 1954, 80 = 1980, 99 = 1999, a7 = 2007, a9 = 2009, b2 = 2012, b5 = 2015). Lo script si limita ad assegnare le variabili %_tdesc% (descrizione della tabella Postgres da creare, per esempio "Uso del suolo 1954 – Volo GAI – Importato con shp2pgsql") e %_cname% (identificativo del dataset, per esempio "1u54") e a chiamare lo script successivo.

2. `load_lu.bat`, uno script che chiama lo script `system_vars.bat` (vedi sotto) e poi assegna sei variabili: `%_vorig%` (versione del dataset di origine, per esempio “01st”); `%_vdest%` (versione del dataset di destinazione, per esempio “02pg”); `%_fpath%` (percorso parametrizzato nel computer locale del file Shapefile da caricare, per esempio “%_lpath%\store\b\blu\b%_dname%”); `%_fname%` (nome parametrizzato del file Shapefile da caricare, per esempio “b%_dname%_r1a%_vorig%.shp”); `%_gname%` (nome del campo geometrico da creare in ambiente PostGIS, per esempio “pol”); `%-S%` (opzione `shp2pgsql` per la creazione di geometrie semplici anziché multi-geometrie, per esempio “-S” oppure “-I”). Infine chiama il quarto script `load_epol.bat`.
3. `system_vars.bat`, uno script che si limita ad assegnare cinque variabili a livello di sistema: `%_phost%` (il nome dell’host sul quale risiede il server Postgres); `%_pname%` (il nome del database Postgres, qui sempre “lm_milano”); `%_puser%` (il nome dell’utente con il quale collegarsi al database Postgres); `%_ppath%` (il percorso nel computer locale nel quale risiedono i programmi `pgSQL`, per esempio “C:\usr\lib\pgsql\bin”); `%_lpath%` (il percorso base del progetto GIS nel computer locale, per esempio “E:\work\2018\vecmsa01”).
4. `load_epol.bat`, il “cuore” dello script di caricamento, che, prima di chiamare le sei istruzioni riportate sopra, definisce altre tre variabili: `%_cname%` (nome del campo della tabella che costituirà la chiave primaria, per esempio “EPOL_ID”); `%_tname%` (nome parametrizzato della tabella di destinazione, qui “%_dname%_epol%_vdest%”); `%_iname%` (nome parametrizzato verso il quale rinominare l’indice spaziale che `shp2pgsql` avrà creato con un nome predefinito, qui “%_dname%_pidx%_vdest%”).

Dopo l’importazione dei dati poligonali nell’ambiente PostGIS (il suffisso “pg” della sigla di versione “02pg” fa riferimento a Postgres/PostGIS), si è proceduti a strutturare gli attributi secondo convenzioni di denominazione che facilitano il trattamento automatizzato e parametrizzato dei dati, oltre a rendere omogenei i nomi degli attributi tra le sette diverse annate. A tale proposito è stata predisposta la famiglia di script SQL `create_lu??_epol_03st.sql`, dove ?? indica nuovamente l’identificativo dell’annata. Gli script si limitano a rinominare e a convertire di formato alcuni campi, tralasciandone quelli che rappresentano ridondanze o informazioni non necessarie. I campi definiti per `lu??_epol_03st` sono:

<code>EPOL_ID</code>	ID del poligono (istanza) – chiave primaria – numero intero;
<code>LUMI_L5D</code>	Codice (ID) di uso del suolo (5° livello) – numero intero;
<code>LUMI_L5NI</code>	Nome dell’uso del suolo (5° livello) – testo variabile (varchar) fino a 254 caratteri;
<code>LUMI_L5D</code>	Codice (ID) di uso del suolo (5° livello) – numero intero piccolo (smallint);
<code>LUMI_L5D</code>	Codice (ID) di uso del suolo (5° livello) – numero intero piccolo (smallint);
<code>LUMI_L5D</code>	Codice (ID) di uso del suolo (5° livello) – numero intero piccolo (smallint);
<code>pol</code>	Geometria poligonale – geometria PostGIS.

Nel passaggio successivo è stata effettuata la sovrapposizione con i poligoni dei comuni e sono stati separati in due tabelle distinte gli attributi informativi da quelli identificativi per assicurare la coerenza tematica dei dati territoriali. Con gli script SQL `create_lu??_epol.sql` (famiglia di script) e

create_1u00_15.sql (script unico) sono state create le famiglie di tabelle 1u??_epo1 e 1u??_15. I campi mantenuti assieme alle geometrie poligonali in 1u??_epo1 sono quelli identificativi, vale a dire EPOL_ID e LUMI_L5D, ai quali si aggiunge AUCO_ID (ID dei Comuni) proveniente dal dato auco_epo1 dei poligoni comunali. Invece le tabelle 1u??_15 contengono gli attributi dell'uso del suolo (sempre una tabella per annata) normalizzati, vale a dire LUMI_L5D (ID della classe e chiave primaria della tabella), LUMI_L5NI (nome italiano della classe) e LUMI_L3D (ID della classe di 3° livello) per operazioni relazionali successive.

Trasformazione dei dati

Da un punto di vista della coerenza interna della serie storica dei dati dell'uso del suolo, basata sulla "collezione" DUSAF, i passaggi di elaborazione sono giunti al loro termine. Tuttavia occorre armonizzare i dati storici con quelli derivati direttamente dal DBTR ad opera di Corvallis. Le differenze tra i due insiemi sono significativi, ma gestibili.

La prima differenza sostanziale riguarda il numero di livelli gerarchici in cui sono strutturate le due famiglie di dati: cinque livelli nel caso della serie storica (1u??) e tre livelli nel caso di DBTR/Corvallis (1uco). Da un punto di vista tecnico non ci sono problemi a ridurre il dettaglio di discretizzazione delle classi da cinque a tre livelli, considerando che la loro indicizzazione avviene proprio per codici progressivi, coerentemente a quanto avviene nel CLC (CORINE Land Cover). È sufficiente porre una clausola group by anche dinamicamente su $\text{div}(\text{LUMI_L5D}, 100)$, vale a dire calcolando la divisione di interi (troncando sull'unità) tra il codice della classe di 5° livello e 100. Per comodità, il calcolo era già stato effettuato in precedenza calcolando il nuovo attributo identificativo della classe di 3° livello LUMI_L3D.

La seconda differenza risiede nella denominazione delle classi. A differenza del CLC e del dato prodotto da Corvallis, la serie storica non prevede denominazioni differenziate a seconda del livello specifico, ma soltanto un'unica denominazione per l'ultima classe (in termini di "profondità") istanziata. Il livello dell'ultima classe poteva però anche variare: su un totale di 98 classi si registrano 21 classi di 5° livello, 45 classi (compresa la classe 9999) che si fermano al 4° livello e 32 classi che si fermano al 3° livello. Inoltre, la "profondità" delle classi poteva essere caratterizzata da asimmetrie: per esempio la classe 1123 "Tessuto residenziale sparso" contiene in tutte le annate (eccetto 1980) un'unica sottoclasse — la 11231 "Cascine" — mentre tutte le superfici non classificate come caschine sono istanze direttamente della classe superiore. Tutte queste caratteristiche rendevano la nomenclatura delle classi della serie storiche inadatta a essere adottata come nomenclatura unificata con l'uso del suolo prodotto da Corvallis. Poiché però le categorie di superfici a cui fanno riferimento entrambi gli insiemi di dati sono sufficientemente sovrapponibili, si è optato per l'approccio opposto: adottare la nomenclatura di 1uco_13 anche per la serie storica e integrarla puntualmente laddove necessario.

Analogo ragionamento è stato fatto per i gradi di impermeabilizzazione e di naturalità definiti nell'ambito della procedura analitica di Corvallis. A ogni classe dell'uso del suolo (di 3° livello) era associata una coppia di valori nell'intervallo [0; 1] che indicavano le due grandezze in base alle caratteristiche fisiche ed ecologiche delle superfici che ne fanno parte. La procedura e gli algoritmi so-

no sufficientemente documentati da poter formulare delle ipotesi — approssimative — di integrazione per le classi di 3° livello presenti nella serie storica, ma assenti dal dato DBTR/Corvallis.

Da un punto di vista logico-analitico, il primo passaggio è consistito nella costruzione di una tabella comparativa tra le classi presenti in ciascuna delle sette annate. Lo script `create_lumi_15_comp.sql` contiene un'istruzione SQL con sette livelli di nidificazione per consentire di relazionare tra di loro i codici di identificazione con le relative denominazioni (ad eccezione del dato del 2015 che era privo di denominazioni) e verificare la consistenza della presenza delle classi e delle denominazioni attraverso le annate. Le differenze sono relativamente piccole, eccezion fatta per il dato del 1980 che, essendo costruito per una scala di rappresentazione cartografica più piccola, è privo di alcuni dettagli. Per il resto, le denominazioni sono consistenti attraverso le annate, senza eccezione alcuna.

Nella successiva — e definitiva — tabella delle classi di 5° livello `lumi_15` si è pertanto mantenuto soltanto un campo con le denominazioni (`LUMI_L5NI`), mentre il resto della tabella è stata trasformata in una tabella di valori binari, dove il valore 1 indica “classe presente” e il valore 0 “classe assente”. I relativi campi sono stati denominati `LUMI_L554` (L5 per il 5° livello e 54 per l'anno 1954), `LUMI_L580`, `LUMI_L599`, `LUMI_L5A7`, `LUMI_L5A9`, `LUMI_L5B2` e `LUMI_L5B5`. A titolo di informazione statistica, delle 98 classi istanziate, una (12126, “Impianti fotovoltaici a terra”) è presente soltanto nel 2012 e nel 2015, un'altra (9999 per i dati non definiti) nel 1999, 2012 e 2015, due (1424 “Aree archeologiche” e 1423 “Parchi divertimento”) sono presenti dal 1999 al 2015, 11 classi sono presenti soltanto nel 1980, 4 soltanto nel 1954, 3 nel 1954 e nel 2015, 6 nel 1954 e nel 1980, 38 sono presenti in tutte le annate eccetto nel 1980, mentre le rimanenti 32 classi sono presenti in tutte le annate.

Uso del suolo 1954-2015 in formato raster

Può essere utile aprire una breve parentesi sulla serie degli insiemi di dati dell'uso del suolo 1954-2015 in formato raster. L'approccio iniziale, visto il verificarsi di alcune incoerenze geometriche e topologiche dei dati vettoriali riscontrati nell'ambito di procedure analitiche eseguite con QGIS (non è chiaro però se i problemi fossero dovuti a incoerenze nei dati utilizzati oppure nel prodotto software impiegato), era quello di eseguire tutte le operazioni di calcolo dei tassi storici di consumo del suolo convertendo i dati poligonali in dati raster, anche considerando il fatto che il formato raster in generale si presta meglio all'esecuzione di operazioni di contabilità sulle aree e considerando, inoltre, che al fine del calcolo del consumo di suolo, non era necessario lavorare con dati ad elevatissima accuratezza geometrica. Si è pertanto proceduto a convertire tutti i dati dell'uso del suolo della serie 1954-2015 in raster, utilizzando sempre la medesima griglia di riferimento con una dimensione delle celle di 5 metri. L'estensione dei dati è stata, in un primo momento, ritagliata su un rettangolo con estensione $(x_{min}, x_{max}, y_{min}, y_{max}) = (470.940, 548.940, 4.998.465, 5.056.965)$ espressa in coordinate UTM, fuso 32 nord, orientato al WGS84. Successivamente, la serie di dati raster è stata ritagliata utilizzando una maschera, creata preliminarmente, del territorio comunali della Città metropolitana di Milano. Le operazioni fin qui descritte sono tutte state eseguite in locale con QGIS. Nel prosieguo delle attività di analisi spaziale dei dati raster si è tuttavia dovuto consta-

tare come le operazioni sui raster fossero particolarmente macchinose e onerose in termini di trasferimenti da locale in remoto (utilizzando il programma raster2pgsql) tra il computer locale e il server del database Postgres/PostGIS e come ci fossero difficoltà a utilizzare i dati raster, archiviati nel database Postgres, da parte dell'ambiente Lizmap. Pertanto si è deciso di cancellare tutti i dati raster caricati in Postgres dal server. Una rappresentazione cartografica dell'uso del suolo storico in formato raster è stata comunque conservata sul server per la consultazione tramite gli strumenti Lizmap, collocando però i dati, convertiti nel formato GeoTIFF, nella sottocartella data della cartella di pubblicazione Lizmap (per questioni di riservatezza dei dati e del sistema si omettono in questa sede i nomi delle cartelle sul server). I nomi dei file TIFF seguono lo schema

blu??mic_r500_lu13.tif

dove ?? rappresenta sempre l'annata (54, 80, 99, a7, a9, b2, b5), sostituendo la cifra delle decine con lettere per numeri superiori a 99), mi indica Milano, r500 la risoluzione al suolo di 500 cm e il suffisso lu13 "land-use, level 3" ("uso del suolo, livello 3").

Uso del suolo unificato

Una volta completate le famiglie di tabelle e dati poligonali per la serie storica dell'uso del suolo e per l'uso del suolo prodotto da Corvallis, è stata eseguita la procedura in cinque passaggi per la creazione di un'unica tabella degli attributi normalizzata.

Il primo passaggio, eseguito nell'ambito dello script `create_lumi_13_comp_01cr.sql` e che ha portato alla creazione della tabella `lumi_13_comp_01cr` ("comp" per "comparativa", "cr" per "creazione") per semplice raggruppamento e aggregazione analitica degli attributi di 5° livello della serie storica. Il commento associato alla tabella è "Uso del suolo – Tabella comparativa delle classi di 3° livello – Primo passaggio: annate 1954-2015". I nomi dei campi sono identici ai nomi della tabella `lumi_15`, con la differenza che "L5" qui è sostituito con "L3".

Il secondo passaggio (`create_lumi_13_comp_02un.sql`) relaziona la tabella appena creata alla tabella degli attributi dell'uso del suolo DBTR/Corvallis (`luco_13`), inserendovi tutti i suoi campi, comprensivi dei gradi di impermeabilizzazione e di naturalità, e aggiungendo il campo con valori binari `LUMI_L3CO` relativo all'uso del suolo "luco" sul modello dei campi `LUMI_L3??` della serie storica 1954-2015. Il commento della tabella Postgres è stato impostato su "Uso del suolo – Tabella comparativa delle classi di 3° livello – Secondo passaggio: DUSAF 1954-2015 + DBTR-Corvallis 2015".

Il terzo passaggio (`create_lumi_13_comp_03ed.sql`) è sostanzialmente un'operazione di editing, in quanto nell'istruzione SQL sono state codificate in modo statico ("hard encoded") alcuni valori del campo `LUMI_L3NI` (nome italiano della classe):

```
create table lumi_13_comp_03ed as
select
  e.*,
  l.*
from lumi_13_comp_02un as l
full outer join (values
```



```
(112, 'Zone residenziali a tessuto discontinuo e rado'),  
(123, 'Aree portuali'),  
(132, 'Discariche'),  
(134, 'Aree degradate non utilizzate e non vegetate (classe DUSAF)'),  
(224, 'Arboricoltura da legno (classe DUSAF)'),  
(314, 'Rimboschimenti recenti (classe DUSAF)'),  
(321, 'Aree a pascolo naturale e praterie'),  
(322, 'Brughiere e cespuglieti'),  
(331, 'Spiagge, dune e sabbie'),  
(332, 'Rocce nude, falesie, rupi, affioramenti'),  
(335, 'Ghiacciai e nevi perenni')  
) as e
```

```
on l."LUMI_L3D" = e.column1  
order by l."LUMI_L3D";
```

Come si può vedere, si tratta dell'integrazione delle denominazioni delle classi non presenti nella tabella `luco_13`. I valori delle classi 112, 123, 132, 321, 322, 331, 332 e 335 sono classi CLC standard, per cui è stato possibile recuperare le denominazioni da varie fonti disponibili nell'Internet. Invece le classi 134, 224 e 314 sono classi definiti soltanto nell'ambito della serie storica DUSAF, per cui le denominazioni sono state recuperate dai dati originali della serie storica. Per rendere esplicita questa operazione, alle classi DUSAF è stata aggiunta la specifica "(classe DUSAF)". Il commento alla tabella Postgres è qui "Uso del suolo – Tabella comparativa delle classi di 3° livello – Terzo passaggio: DUSAF 1954-2015 + DBTR-Corvallis 2015 + integrazioni CLC+DUSAF"

Il quarto passaggio (`create_lumi_13_comp.sql`, senza suffisso di versione, che potrebbe in teoria essere "04ed", in quanto l'ultima versione del processo di trattamento) consiste nuovamente in un'operazione di editing, che però si è voluta tenere distinta da quella precedente per via della diversa origine delle informazioni inserite: sono stati integrati i valori "mancanti" dei gradi di impermeabilizzazione e di naturalità (stime effettuate dall'autore) nonché i valori cromatici e la relativa fonte ("CLC" per valori RGB convenzionali definiti da parte dell'EPA e "HESC" per i tre valori definiti dall'autore a integrazione delle tre classi DUSAF), rispettivamente nei nuovi campi `LUMI_L3VC` e `LUMI_L3VA`). Commento alla tabella Postgres: "Uso del suolo – Tabella comparativa delle classi di 3° livello – Quarto passaggio: DUSAF 1954-2015 + DBTR-Corvallis 2015 + integrazioni CLC+DUSAF+HESC".

Il quinto è l'ultimo passaggio, documentato nello script `create_lumi_13.sql`, è consistito nella selezione dei soli campi informativi degli attributi di 3° livello, tralasciando i campi documentali sulla presenza delle classi nelle varie annate della tabella comparativa. Il commento descrittivo associato alla tabella Postgres è pertanto semplicemente "Uso del suolo DUSAF 1954-2015 e DBTR-Corvallis 2015".

Gradi di impermeabilizzazione e di naturalità a livello comunale 1954-2015

Una volta ottenuti i dati poligonali e gli attributi definitivi della serie storica dell'uso del suolo, sovrapposti ai poligonali comunali (i confini comunali sono tuttavia quelli attuali, risultanti dopo

l'istituzione del Comune di Vermezzo con Zelo l'8 febbraio 2019), è stata costruita la famiglia di tabelle statistiche `auco_1u??` (script `create_auco_1u?? .sql`) che, a partire dai poligoni raggruppati per comuni e classi di 3° livello (`1u??_epo1_13` per la serie storica e `1uco_epo1` per la versione DBTR/Corvallis) con l'insieme di attributi `COLU` ("Comuni / Land-use", identificati da `COLU_ID` che è la concatenazione di `AUCO_ID` con `LUMI_L3D`), dai poligoni comunali (`auco_epo1`) e dalla tabella degli attributi degli usi del suolo di 3° livello (`1umi_13`), raccoglie per ogni annata una serie di statistiche, utili nel prosieguo per la redazione di carte tematiche e grafici, che sono collocate nei seguenti campi (qui si riporta l'esempio del 1954, mentre per le annate 1980, 1999, 2007, 2009, 2012, 2015 nonché per la versione DBTR/Corvallis le sigle sono rispettivamente 80, 99, A7, A9, B2, B5 e CO).

<code>RDAT_ID</code>	ID del record (istanza) – chiave primaria;
<code>LU54_L3D</code>	Codice (ID) della classe di uso del suolo (3° livello) – 1954;
<code>AUCO_ID</code>	Codice (ID) del Comune (istanza);
<code>COLU_ID</code>	Codice (ID) della combinazione tra Comune e Classe di uso del suolo (istanza);
<code>AUCO_IARE</code>	Area comunale in m ² (confini comunali del 2019);
<code>COLU_IARE</code>	Area della combinazione tra Comune e Classe di uso del suolo in m ² ;
<code>COLU_54AS</code>	Area convenzionalmente impermeabilizzata in m ² – 1954;
<code>COLU_54AN</code>	Area convenzionalmente naturale in m ² – 1954;
<code>COLU_54RA</code>	Contributo (valori tra 0 e 1) dell'area all'area comunale – 1954;
<code>COLU_54PA</code>	Percentuale (con una posizione decimale) dell'area sull'area comunale – 1954;
<code>COLU_54QA</code>	Percentuale (numero intero) dell'area sull'area comunale – 1954.

Il raggruppamento degli attributi è eseguito in modo tale da non avere mai più di un record con la medesima combinazione tra ID comunale e ID della classe di uso del suolo di 3° livello. I valori dei campi `COLU_??AS` e `COLU_??AN` (aree, espresse in m², convenzionalmente impermeabilizzate e naturali) sono stati calcolati moltiplicando i m² delle singole superfici per il grado di impermeabilizzazione e di naturalità (valori compresi tra 0 e 1) assegnata in modo parametrico a ciascuna delle classi di uso del suolo.

Le statistiche di questa famiglia di tabelle sono quindi state ulteriormente aggregate — per Comuni — e collocate all'interno di un'unica tabella di attributi con 133 record (uno per ciascun Comune della Città metropolitana di Milano) e ben 70 campi:

<code>AUCO_ID</code>	Codice (ID) del Comune (istanza) – chiave primaria;
<code>AUCO_INAI</code>	Nome italiano del Comune;
<code>AUZO_ID</code>	Codice (ID) della Zona omogenea (istanza);
<code>AUZO_INAI</code>	Nome italiano della Zona omogenea;
<code>AUCO_IARE</code>	Area comunale in m ² (confini comunali del 2019);
<code>AUCO_IAREK</code>	Area comunale in km ² (confini comunali del 2019);
<code>AUCO_??AS</code>	Area comunale convenzionalmente impermeabilizzata in m ² – 8 annate;
<code>AUCO_??AN</code>	Area comunale convenzionalmente naturale in m ² – 8 annate;
<code>AUCO_??RS</code>	Grado di impermeabilizzazione comunale – 8 annate;

AUCO_??RN Grado di naturalità comunale – 8 annate;
AUCO_??PS Percentuale (una posizione decimale) di impermeabilizzazione comunale – 8 annate;
AUCO_??PN Percentuale (una posizione decimale) di naturalità comunale – 8 annate;
AUCO_??QS Percentuale (numero intero) di impermeabilizzazione comunale – 8 annate;
AUCO_??QN Percentuale (numero intero) di naturalità comunale – 8 annate.

Le coppie di metacaratteri ?? indicano anche qui ogni volta una famiglia di campi, ciascuna relativa alle otto annate (54, 80, 99, A7, A9, B2, B5 e C0). L'inclusione ridondante anziché relazionata dei campi AUCO_INAI, AUZO_ID e AUZO_INAI è dettata da esigenze di semplificazione delle procedure di visualizzazione e documentazione nel prosieguo del lavoro. In modo simile anche la creazione statica del campo AUCO_IAREK e dei campi con i valori percentuali (con una e senza posizioni decimali) dei gradi di impermeabilizzazione e di naturalità. La tabella è stata chiamata `auco_u` con descrizione "Gradi di impermeabilizzazione e di naturalità a livello comunale dal 1954 al 2015 (confini comunali del 2019)". Le istruzioni SQL necessarie — molto voluminose (217 righe) sebbene concettualmente semplici — sono state salvate nello script `create_auco_u.sql`.

Come ultimo passaggio è stato creato l'insieme di dati poligonale `auco_epo1_u` ("Unità amministrative – Comuni – con uso del suolo e gradi di impermeabilizzazione e naturalità dal 1954 al 2015", salvato nello script `create_auco_epo1_u.sql`) come semplice giunzione statica ("join") tra l'insieme poligonale dei comuni `auco_epo1` e la tabella degli attributi `auco_u` creata al passaggio precedente. Anche qui le ridondanze sono estreme — sebbene di fatto indispensabili pena il verificarsi di rallentamenti insostenibili durante la visualizzazione dei dati online — tali da rendere necessaria, in caso di modifiche nei dati a monte, la ripetizione dell'esecuzione di tutti gli script SQL più a valle. È stata chiaramente valutata la possibilità di operare con le "viste materializzate" di Postgres (tabelle dinamiche, basate su una serie di istruzioni SQL salvate *nel* database Postgres assieme alla "vista materializzata", che consentono aggiornamenti agevoli), ma le regole di integrità dello stesso Postgres impediscono che una tabella utilizzata da un'istruzione SQL di creazione di una vista materializzata possa subire delle modifiche strutturali (aggiunta, rimozione o nuova definizione di campi). In tali casi è necessario prima rimuove completamente la vista materializzata in questione — assieme alle eventuali viste materializzate da essa dipendenti — per poi ricrearla in un secondo momento. Nell'ambito di flussi di lavoro che prevedono continue ristrutturazioni di tabelle e insiemi di dati questo tipo di comportamento è altamente improduttivo. L'effetto sarebbe quello di dover salvare comunque in locale le istruzioni SQL necessarie per ricreare le viste materializzate, esattamente come avviene con le tabelle statiche classiche. Per questo motivo si è — dopo una fase iniziale consistente — deciso di rinunciare del tutto alle viste dinamiche e alle viste materializzate.

Edifici

Molti degli indicatori morfo-demo-sociologici costruiti utilizzano dati morfologici degli edifici. Nello specifico si tratta del volume, dell'altezza e della superficie coperta (superficie occupata da una costruzione) nonché della distinzione tra edifici in generale ed edifici residenziali. Inoltre, per

alcune operazioni è anche necessario modellare la popolazione, a partire dai dati censuari del 2011, sugli edifici in base al loro volume (ipotizzando che, all'interno di una sezione di censimento, la popolazione tende a essere distribuita sugli edifici in base al volume costruito a uso residenziale). Tutte queste informazioni sono potenzialmente disponibili nell'ambito del DBTR lombardo e, nello specifico, negli strati informativi 020101, identificato anche attraverso l'abbreviazione UN_VOL, e 020102, identificato anche attraverso l'abbreviazione CR_EDF/EDIFC. Lo strato 020101/UN_VOL corrisponde alle "unità volumetriche" e lo strato 020102/CR_EDF/EDIFC ai "corpi edificati" e agli "edifici", così come definito nel *Catalogo dei dati territoriali – Specifiche di contenuto dei database geotopografici*, allegato al Decreto 10 novembre 2011 del Ministro per la pubblica amministrazione e l'innovazione, intitolato "Regole tecniche per la definizione delle specifiche di contenuto dei database geotopografici". Lo strato indicato con la doppia sigla CR_EDF/EDIFC si riferisce agli edifici come corpi complessi, costituiti dai singoli volumi, dove la classe di oggetti DBT EDIFC rappresenta una classe derivata dalla classe astratta CR_EDF. Entrambi gli insiemi di dati contengono, nell'implementazione lombarda messa a disposizione nell'ambito del Progetto DeSK, un attributo, denominato ID_EDIF nello strato UN_VOL e EDIFC_ID nello strato CR_EDF/EDIFC, che non compare né nella specifica nazionale né in quella regionale, ma che a rigor di logica dovrebbe rappresentare l'identificatore numerico dell'edificio, vale a dire dell'elemento logico di base dello strato CR_EDF/EDIFC.

Si sono eseguite varie verifiche, ma non è stato possibile trovare evidenza significativa della corrispondenza tra i due strati nell'identificatore in questione. In molti casi c'è congruenza, ma esiste anche un numero notevole (varie migliaia) di situazioni in cui poligoni sovrapposti tra i due strati sono qualificati da identificatori differenti. Queste situazioni di incongruenza possono riguardare singoli volumi d'edificio, che non si allineano agli altri volumi dello stesso edificio, ma anche interi edifici che, a seconda dello strato, portano identificatori diversi. Questo ha chiaramente reso impossibile — come veniva inizialmente ipotizzato — la creazione di un relazionamento tra le due tabelle PostGIS con un'istruzione della forma

```
select * from edvo_epol full outer join using ("EDIF_ID")
```

assumendo di aver importato nel database PostGIS lo strato UN_VOL con il nome edvo_epol e lo strato CR_EDF/EDIFC con il nome edco_epol e di aver assegnato in entrambe le tabelle il nome EDIF_ID all'identificatore degli edifici.

In Figura 4 è rappresentata una situazione realmente incontrata nel DBTR lombardo relativamente alla Città di Milano. Le superfici rappresentate in grigio corrispondono a poligoni semplici dello strato UN_VOL, mentre le superfici rosse corrispondono a un unico multi-poligono, sempre dello strato UN_VOL, costituito da cinque aree tra loro staccate (quattro grandi e una molto piccola, invisibile nella figura, ma identificata con il codice 277647). I numeri sovrapposti rappresentano i codici identificativi degli edifici nel campo ID_EDIF nello strato UN_VOL. Come si può facilmente vedere, oltre all'edificio multi-poligonale 277647, sono rappresentati altri sette edifici: 277645, 277646, 385985, 387108, 387109, 387110 e 387111. Per maggiore chiarezza, i poligoni ("volumi") appartenenti, secondo questa aggregazione, al medesimo edificio sono stati contornati con un tratto

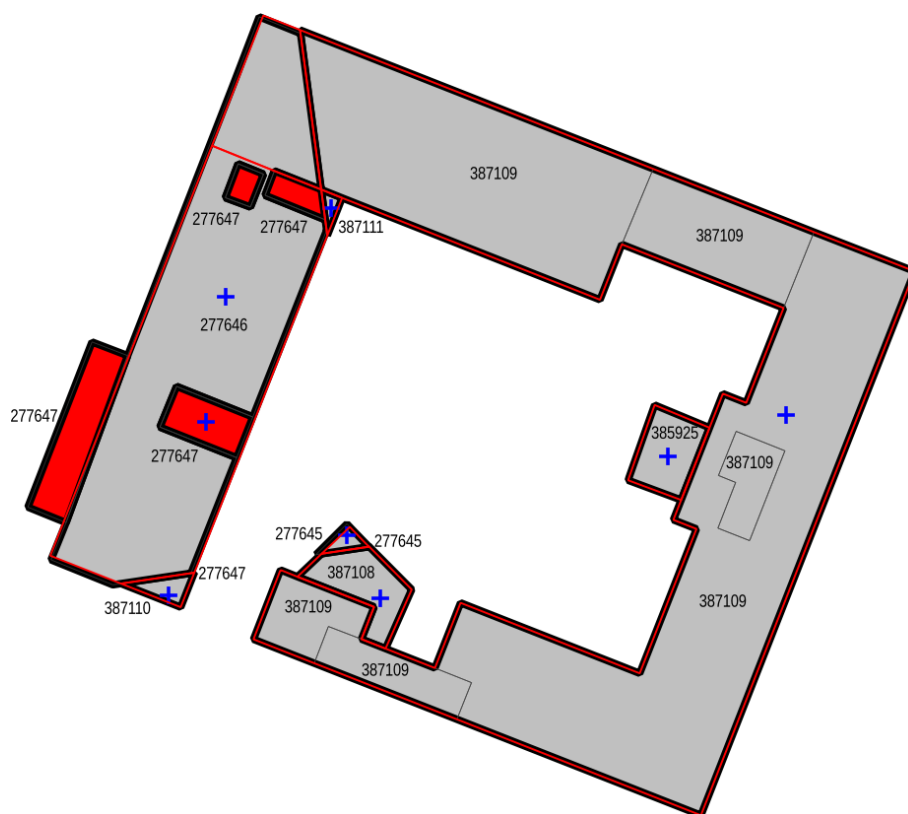


Figura 4: Inconsistenze tematiche e geometriche tra volumi e corpi d'edificio (strati UN_VOL/020101 e CR_EDF/EDIFC/020102) da un lato e il concetto di edificio (campi ID_EDIF e EDIFC_ID rispettivamente in UN_VOL e CR_EDF/EDIFC) dall'altro nel DBTR lombardo

grafico nero più spesso, mentre quelli più sottili in grigio rappresentano le suddivisioni interne nella loro proiezione ortogonale. Invece i tratti rossi, un po' più sottili di quelli neri spessi, rappresentano la suddivisione in "edifici" in base all'attributo EDIFC_ID dello strato CR_EDF/EDIFC. Le incongruenze sono varie: geometriche (parzialmente corrette nello strato CR_EDF/EDIFC), tematiche (gli "edifici" 277645, 387110 e 387111, presenti in entrambi gli strati, sono dei palesi errori di non aggregazione rispettivamente agli edifici 277246 per il primo e 387108 per i secondi due) e topologiche (gli "edifici" 277646 e 277647 corrispondono ad aggregazioni diverse a seconda dello strato). Inoltre sono "materializzate" con molta evidenza le suddivisioni in fogli cartografici che tagliano con delle linee rette gli oggetti spaziali rappresentati.

Data la grande quantità di oggetti in questione (varie centinaia di migliaia) e la grande quantità di incongruenze di questo e di altri tipi (varie decine di migliaia), è stato impossibile ipotizzare un qualche intervento di correzione, anche utilizzando procedure automatiche eventualmente da predisporre ad hoc. Visto, poi, che per la costruzione degli indici sono necessari i valori di altezze e volu-

mi, archiviate soltanto con lo strato UN_VOL nel campo UN_VOL_AV (altezza del volume d'edificio, poi rinominato in EDVO_IHGV e a partire del quale si è anche costruito il volume EDVO_IVOV per moltiplicazione con l'area dell'edificio), si è deciso di predisporre i dati di base per il calcolo degli indicatori a partire da questo strato che, dopo l'importazione (effettuata secondo procedure analoghe a quelle descritte già in più occasioni in precedenza), è stato rinominato in edvo_epo1. Era tuttavia necessaria anche la creazione di un collegamento con l'altro strato, nell'ambito del quale era archiviato l'informazione circa l'uso dell'edificio (campo EDIFC_USO, poi rinominato in EDCO_UD) richiesta per il calcolo del valore della densità edilizia residenziale, del volume residenziale pro capite e dell'indice di residenzialità volumetrica. Non essendo possibile il relazionamento attraverso un campo comune, si è optato per un relazionamento di tipo spaziale, generando in un primo momento dei centroidi per ciascun volume d'edificio attraverso il quale trasferire l'attributo EDCO_UD dalla tabella PostGIS edco_epo1 alla tabella PostGIS edvo_epo1.

Eseguiti tutti i calcoli sui valori metrici dei corpi d'edificio, è stata eseguita l'ultima fase nelle operazioni di manipolazione dei dati degli edifici. Consapevole delle incongruenze rilevate in precedenza, che però a questo punto si limitavano a poche centinaia, si è proceduto ad aggregare i volumi d'edificio in base ai valori nel campo EDIF_ID dell'insieme edvo_epo1 (strato UN_VOL) anziché edco_epo1 (strato CR_EDF/EDIFC), come sarebbe stato più naturale. Dato che i perimetri degli edifici così costruiti (gli edifici dello strato CR_EDF/EDIFC mostrano invece alcune semplificazioni geometriche soprattutto in corrispondenza di sporgenze verso le superfici pubbliche antistanti gli edifici) fuoriescono in diverse situazioni dai perimetri degli isolati urbani, facendo ricadere alcuni centroidi dei volumi d'edificio al loro esterno, si è deciso di generare dei centroidi anche per gli edifici costruiti per aggregazione di volumi (a cui è stata assegnata l'abbreviazione EDIF e il nome per la tabella PostGIS edif_epo1) al fine di ridurre in maniera significativa il rischio di incongruenze geometriche tra edifici e isolati per causa di inaccuratezze geometriche. In questo modo è stato possibile gestire in modo più agevole, senza la necessità di utilizzare sempre i dati poligonali (i campi delle geometrie aumentano sensibilmente la dimensione delle tabelle), le operazioni di calcolo anche per le sezioni di censimento, per i comuni e per le zone omogenee.

Aree vincolate e protette

Per la costruzione delle carte tematiche delle aree vincolate e protette, nonché per il calcolo degli indici (percentuali) comunali di aree naturali protette, era innanzitutto necessario raccogliere i dati territoriali di riferimento. A tale proposito è stato sufficiente effettuare una serie di ricerche nel Geoportale della Regione Lombardia, da dove si sono potuti ottenere i seguenti insiemi di dati.

- “Aree protette” che contengono — citando dai metadati pubblicati sul Geoportale — “i confini delle aree protette in Lombardia, vale a dire: Parchi regionali, Parchi naturali e Parco Nazionale dello Stelvio – Riserve naturali – Monumenti naturali – Parchi locali di interesse sovracomunale (PLIS) – Zone Speciali di Conservazione (ZSC) e Siti di Importanza Comunitaria (SIC) – Zone di Protezione speciale (ZPS)”. Nei metadati si continua a leggere che le date di aggiornamento dei dati in questione sono tutte molto recenti e non risalgono in nessun caso a prima di

luglio 2016, mentre in molti casi le date di aggiornamento sono del 2019. I dati sono forniti nel formato Shapefile con i seguenti nove insiemi di dati singoli:

- Aree Ramsar;
 - Monumenti naturali (areali e puntuali);
 - Parchi locali di interesse sovracomunale;
 - Parchi naturali;
 - Parchi regionali e nazionali;
 - Riserve regionali e nazionali;
 - ZPS e, separatamente, ZSC/SIC.
- PAI – Piano Stralcio per l’Assetto Idrogeologico del Bacino del Po. I dati sono forniti nel formato Shapefile con i seguenti nove insiemi di dati singoli, di cui 5 lineari, 3 areali e 1 puntuale:
- Aree RME (insieme di dati utilizzabile);
 - Dissesti lineari, areali, puntuali (3 insiemi di dati distinti, di cui utilizzabile solo quello con impianto areale);
 - Limiti Fasce A, B, B di progetto e C (4 insiemi di dati lineari distinti che sarebbero utilizzabili previa interventi intensivi di correzione topologica manuali);
 - Stato di attuazione PAI dei comuni.
- “Basi ambientali della pianura – rilevanze naturalistiche e paesaggistiche”. Si cita dai metadati del Geoportale. “Per ciascuna componente informativa delle rilevanze naturalistiche e paesaggistiche, è stato realizzato un raggruppamento di layer con rese grafiche differenti in base alla scala di visualizzazione. Vengono indicati di seguito gli strati informativi presenti: 1) fontanili attivi 2) aree archeologiche 3) architettura rurale 4) architettura religiosa 5) architettura turistica 6) siti preistorici 7) architettura per lavorazione prodotti agricoli 8) architettura fortificata 9) architettura civile 10) opere idrauliche 11) architettura per attività industriali 12) stazioni ferroviarie 13) ponti 14) sistemazioni agrarie 15) rete irrigua 16) tracce organizzazione fondiaria 17) canali navigabili 18) luoghi storici 19) punti panoramici 20) percorsi panoramici 21) giardino 22) insediamenti abitativi 23) rete ferroviaria 24) rete stradale”. Dei 20 insiemi di dati forniti nel formato Shapefile — 14 puntuali, 5 lineari e 1 areale, peraltro non coincidenti con i 24 strati informativi elencati nei metadati — è stato effettivamente utilizzato uno soltanto, in quanto dà luogo a geometrie con impianto areale che possono determinare una percentuale di territorio “protetto”, mentre per esempio le “aree archeologiche”, che sarebbero stati dei “candidati” all’inclusione nella raccolta di dati, hanno impianto puntuale:
- Rilevanze vegetazionali (boschi di conifere, boschi di latifoglie, boschi misti, marcite, rimboschimenti recenti, vegetazione dei greti, vegetazione erbacea e cespuglieti, vegetazione palustre e delle torbiere); la data di ultima revisione del dato è — per tutti i layer della raccolta — il 10 ottobre 2007, per cui si presume che si tratti di un’estrazione di alcuni temi dall’uso del suolo DUSAF 2.1 dello stesso 2007.
- Zone di preservazione e salvaguardia ambientale estratte dalla Tavola 2 del PTR (Piano territoriale regionale). La raccolta di dati, aggiornata secondo i metadati al 10 ottobre 2019, è costitui-

ta dai seguenti 19 insiemi di dati — di cui 10 areali, 7 lineari e 2 puntuali — dati formato Shapefile, in parte già coperti dalla prima e dalla seconda delle raccolte di dati qui documentate:

- Aree a convenzione Ramsar;
 - Aree a rischio idrogeologico molto elevato (PAI);
 - Comprensori di bonifica ai sensi della DGR 19 settembre 2019;
 - Fiume Po (linee);
 - Geositi (punti);
 - Ghiacciai;
 - Laghi;
 - Limite Fascia A (linee);
 - Limite Fascia B di progetto (linee);
 - Limite Fascia B (linee);
 - Limite Fascia C (linee);
 - Parchi Naturali;
 - Parchi Nazionali;
 - Pericolosità RP scenario raro (L);
 - Rete canali SIBITER (linee);
 - Sistema dei Navigli (linee);
 - Siti di importanza comunitaria (ZSC e SIC);
 - Siti Unesco (punti);
 - Zone di Protezione Speciale (ZPS).
- Rete Ecologica Regionale (RER). La RER è stata “ricavata da un progetto del 2008 di Regione Lombardia con Fondazione Lombardia Ambiente (FLA) che ha individuato le aree prioritarie per la biodiversità e la successiva costruzione delle connessioni ecologiche, realizzando tutti gli elementi della rete. I dati sono aggiornati al 2009 (DGR 30 dicembre 2009 n. 8/10962)” (fonte: metadati del Geoportale). Nel campo “Data dell’ultima revisione del dato” si legge “17 ottobre 2011”. La raccolta è costituita dai seguenti sei insiemi di dati nel formato Shapefile, di cui cinque areali e uno lineare:
- Corridoi regionali primari ad alta antropizzazione;
 - Corridoi regionali primari a bassa o moderata antropizzazione;
 - Elementi di primo livello della RER;
 - Elementi di secondo livello della RER;
 - Gangli della RER;
 - Varchi della RER (linee).
- Vincoli paesaggistici. Si tratta degli oggetti tutelati ai sensi del Codice del paesaggio (D.lgs. 42/2004). La raccolta è organizzata nei seguenti 12 insiemi di dati nel formato Shapefile, di cui 10 di tipo areale e 2 di tipo lineare:
- Alvei fluviali tutelati;
 - Area argini maestri fiume Po;

- Aree di interesse pubblico di difficile cartografazione (sic) (linee);
 - Aree rispetto corsi d'acqua tutelati;
 - Beni e immobili di notevole interesse pubblico;
 - Fiumi torrenti e corsi acqua pubblici e relative sponde (linee);
 - Ghiacciai e circhi glaciali;
 - Perimetro e Aree di notevole interesse pubblico;
 - Terreni alpini e appenninici;
 - Territori contermini a i laghi;
 - Territori coperti da foreste e da boschi;
 - Zone umide.
- Aree soggette a vincolo idrogeologico. L'insieme di dati è fornito sotto forma di un unico insieme di file nel formato Shapefile. Il dato è inoltre privo di attributi identificativi e qualitativi, eccezion fatta per i campi GID (identificatore del record), SHAPE_AREA e SHAPE_LEN, generati dal software GIS utilizzato dal produttore dei dati.

I dati selezionati sono stati caricati sul database Postgres/PostGIS e strutturati secondo le modalità illustrate più volte nel corso della presente relazione. Poiché, nel caso dei dati qui in questione, non vi era la necessità di ulteriori elaborazioni, la strutturazione e standardizzazione delle geometrie e degli attributi poteva limitarsi a interventi minimi, tali da rendere le operazioni di interrogazione, necessari ai fini della visualizzazione dei dati cartografici e attributi da parte di QGIS e Superset, compatibili con le convenzioni di denominazione adottate nell'ambito delle attività qui documentate. In altre parole, gli interventi di standardizzazione potevano limitarsi a quei campi oggetto di interrogazione nell'ambito dei progetti QGIS per la pubblicazione con Lizmap e nell'ambito delle visualizzazioni con Superset, mentre per gli altri campi non era necessario alcun intervento.

Visualizzazione dei dati e delle informazioni

I due principi dell'archiviazione centralizzata nel database Postgres/PostGIS e della replicabilità di tutte le procedure eseguite, illustrati all'inizio del capitolo sul trattamento dei dati, impongono che anche per la visualizzazione dei dati e delle informazioni vengano adottati strumenti in grado di garantire la loro applicazione. Per questo era necessario identificare degli strumenti che potessero accedere in modo dinamico ai dati PostGIS (e, limitatamente ai dati raster della serie storica dell'uso del suolo, anche a file di dati caricati su un server e ivi sostituibili in qualunque momento senza provocare contraccolpi nella visualizzazione dei dati e delle informazioni) e che fossero trasparenti a un punto tale da consentire a chiunque (con le necessarie conoscenze tecniche di base) di poter leggere, capire e in caso di bisogno modificare le modalità di visualizzazione. Questi strumenti erano già stati individuati durante le fasi del Progetto DeSK precedenti all'inizio delle attività di consulenza qui documentate. Si trattava dei seguenti cinque strumenti informatici (programmi o ambienti di lavoro).

- SICLA, nato come *Sistema di concertazione dei limiti amministrativi*, si è, nel corso delle attività di evoluzione, trasformato in uno strumento altamente performante per la concertazione tra soggetti di dati territoriali in generale. Per gli scopi posti nell'ambito dell'incarico qui documento, il SICLA non era pertinente.
- SIGEO, il Sistema di validazione dei contenuti e segnalazioni, serve per raccogliere segnalazioni e proposte di modifica e correzione di dati territoriali e procedere alla loro validazione, da parte di un soggetto autorizzato, ed eventualmente alla loro integrazione nel database topografico. Anche questo strumento non era pertinente nei compiti qui posti.
- AppDSS, un *applicativo di tipo decision-support system*, con il quale è possibile visualizzare in modo semplice dati e informazioni complesse, sottoporle a un'utenza non esperta, dando la possibilità di "giocare" con i dati e le visualizzazioni. Questo strumento era potenzialmente molto interessante per supportare alcune delle esigenze comunicative presenti nell'ambito del presente incarico.
- Superset, un ambiente apparentemente simile ad AppDSS, più semplice da configurare, ma anche meno performante dal punto di vista della velocità di accesso ai dati e delle capacità grafiche e cartografiche. Nella scelta tra AppDSS e Superset, alla fine, si è optato per quest'ultimo.
- Lizmap, un ambiente di visualizzazione di dati territoriali molto performante e completamente integrato nel flusso di lavoro di QGIS: è sufficiente predisporre un "progetto" di QGIS, configurare alcuni pochi parametri e caricare il file progetto e due file ausiliari in una cartella specifica del server Internet sul quale sono installati il prodotto "QGIS Server" e lo stesso Lizmap. Con Lizmap possono essere visualizzati sia dati che risiedono nel database Postgres, sia dati sotto forma di file, come Shapefile, TIFF e altro, che risiedono in una specifica cartella del server Internet. Con Lizmap è anche possibile consentire la modifica tramite il protocollo HTTP di taluni dati territoriali. Lizmap è stato scelto come strumento avanzato per la visualizzazione e l'interrogazione di dati territoriali.

Ovviamente non sono state prodotte visualizzazioni statiche, come prodotti cartacei, carte in formato PDF o simili, in quanto non sarebbero stati compatibili con i principi elencati in apertura.

Lizmap

Nel presente documento non si forniscono informazioni precise su nomi di server, database, indirizzi IP, nomi di utenti e, più in generale, credenziali di accesso ai server e ai database Postgres per salvaguardare la sicurezza dei dati online. Le informazioni di accesso vengono fornite in separata sede, assieme al materiale informativo e informatico (principalmente script e file documentali ausiliari). Le visualizzazioni predisposte con Lizmap sono le seguenti.

- Uso del suolo 1954-2015 (versione raster).
- Uso del suolo e gradi di impermeabilizzazione e di naturalità 1954-2015 in forma sia disaggregata (poligoni DUSAF) si aggregata per comuni.

- Indicatori di sostenibilità (morfo-demo-sociologici), riferiti al 2015 (elementi morfologici) e al 2011 (demografia ed economia), aggregati per isolati urbani, sezioni di censimento, comuni e zone omogenee, come da Tabella 1 della presente relazione.
- Indice del consumo di suolo ponderato 1954-2015 (indicatore 11.3.1 degli SDG delle Nazioni Unite), aggregati per comuni (132 territori, con Baranzate aggregata a Bollate).
- Vincoli ex lege ed aree protette e gradi (percentuale) di protezione dei territori comunali.

Superset

A completamento delle informazioni cartografiche pubblicate con Lizmap, sono state predisposte alcune infografiche e carte semplificate con l'ambiente Superset (lo strumento non consente visualizzazioni cartografiche molto complesse o basate su dati molto complessi). Si tratta dei seguenti temi.

- Gradi di impermeabilizzazione e di naturalità nel 2015 su base comunale.
- Evoluzione nel tempo (1954-2015) dell'indice del consumo di suolo ponderato (indicatore SDG 11.3.1).
- Gradi (percentuali) di aree protette per territorio comunale.

Markus Michael Hedorfer · Via Ca' Rossa 118/D · I-30173 Venezia-Mestre VE

PEC: hedorfer@pec.hesc.it → {katia.maretto;giusepperoberto.chiaia}@venezia.pecavvocati.it

avv. Giuseppe Roberto Chiaia
Avvocatura della
Città metropolitana di Venezia
Via Forte Marghera 191
30173 Venezia-Mestre VE

VOSTRO RIF.
prot. n. —

VOSTRA LETTERA

NOSTRO RIF.
vecmsa01/mail/grc9cb1a

DATA
11 dicembre 2019

Oggetto:

Contratto per prestazione di lavoro autonomo dell'11 giugno 2019

Gentili Signori,

con la presente desidero comunicare che, richiamando ogni mia precedente corrispondenza in argomento, sono terminate le attività professionali relative al contratto in oggetto. A causa del documentato (corrispondenza via email con l'ing. Luca Celeghin del 9 e del 10 dicembre 2019) blocco delle attività di sei (6) ore nella giornata del 9 dicembre, dovuto a problemi tecnici sul server dei dati (spazio su disco insufficiente) localizzato presso gli uffici della Città metropolitana di Venezia, è stato necessario posticipare la chiusura delle attività di altrettante ore fino alla giornata lavorativa odierna. Nella mattinata di oggi si è verificato un ulteriore blocco della connettività al database, installato sul su menzionato server dei dati, le cui cause tecniche esatte mi sono ancora ignote e che persiste al momento in cui scrivo, che ha impedito l'esecuzione del calcolo finale del rapporto tra l'area del territorio protetto e l'area del territorio comunale per i comuni della Città metropolitana di Milano. Tuttavia, il dato in sé (aree protette e/o vincolate) è presente ed è stato pubblicato, assieme a tutti gli altri dati, nel database "Postgres" e nell'ambiente di visualizzazione "Lizmap".

Con queste precisazioni, sono stati realizzati tutti i prodotti richiamati dalla dott.ssa Cinzia Davoli nella sua comunicazione via email ordinaria delle ore 8:58 del 7 ottobre 2019, oltre una serie di altri dati non richiesti dagli accordi verbali o nella menzionata email.

Si resta pertanto in attesa della pronta liquidazione del corrispettivo a me dovuto per l'attività compiuta e, pertanto, allego alla presente la nota di pagamento riemessa con data odierna.

Distinti saluti,


Markus Hedorfer