

# Corso ITS: *ARTIFICIAL INTELLIGENCE SPECIALIST*

## Modulo: Programmazione ad oggetti in Python e librerie esterne

Docente: *Andrea Ribuoli*

---

**Martedì 11 Marzo 2025**

**09:00 - 14:00**

---

## Accesso ai database manager da Python

- ORM (ad esempio [SQLAlchemy](#))
- pure Python (ad esempio [asyncpg](#)) or binary-driver-based interface (ad esempio [psycopg2](#))
- DB Manager (ad esempio [postgresql](#))

### Avvio del database manager

```
% /usr/local/opt/postgresql@14/bin/postgres -D /usr/local/
var/postgresql@14
2025-03-10 15:57:51.099 CET [26659] LOG:  starting PostgreSQL
14.17 (Homebrew) on x86_64-apple-darwin23.6.0, compiled by
Apple clang version 16.0.0 (clang-1600.0.26.6), 64-bit
2025-03-10 15:57:51.100 CET [26659] LOG:  listening on IPv6
address ":::1", port 5432
2025-03-10 15:57:51.100 CET [26659] LOG:  listening on IPv4
address "127.0.0.1", port 5432
2025-03-10 15:57:51.101 CET [26659] LOG:  listening on Unix
socket "/tmp/.s.PGSQL.5432"
2025-03-10 15:57:51.113 CET [26661] LOG:  database system was
shut down at 2025-03-09 16:57:56 CET
2025-03-10 15:57:51.121 CET [26659] LOG:  database system is
ready to accept connections
```

---

## approfondimento su ORM da Python via SQLAcademy V2

```
In [1]: from typing import List
from typing import Optional
from sqlalchemy import ForeignKey
from sqlalchemy import String
from sqlalchemy.orm import DeclarativeBase, Mapped, mapped_column, relationships

class Base(DeclarativeBase):
    pass

class Provincia(Base):
    __tablename__ = "provincia"
    id: Mapped[str] = mapped_column(String(2), primary_key=True)
    name: Mapped[str] = mapped_column(String(30), nullable=True)
    regione_id: Mapped[int] = mapped_column(ForeignKey("regione.id"), nullable=True)
    regione: Mapped["Regione"] = relationship(back_populates="province")
    residenti: Mapped[int] = mapped_column(nullable=True)

    def __repr__(self) -> str:
        return f"Provincia(id={self.id!r}, name={self.name!r}, residenti={self.residenti!r})"

class Regione(Base):
    __tablename__ = "regione"
    id: Mapped[int] = mapped_column(primary_key=True)
    name: Mapped[str] = mapped_column(String(30))
    province: Mapped[List["Provincia"]] = relationship(
        back_populates="regione", cascade="all, delete-orphan"
    )

from sqlalchemy import create_engine
engine = create_engine("postgresql+psycopg2://localhost/italy")

Base.metadata.create_all(engine)
```

## inserimenti tramite psql

```
italy=# INSERT INTO provincia (id, name) VALUES('PU', 'Pesaro
e Urbino');
INSERT 0 1
italy=# INSERT INTO regione (name) VALUES('Marche');
INSERT 0 1
italy=# SELECT * FROM provincia;
 id |          name          | regione_id | residenti
----+-----+-----+-----
  PU | Pesaro e Urbino       |           |
(1 row)
```

```

italy=# UPDATE provincia SET regione_id = 1 WHERE id = 'PU';
UPDATE 1
italy=# SELECT * FROM provincia;
 id |      name      | regione_id | residenti
----+-----+-----+-----
  PU | Pesaro e Urbino |          1 |
(1 row)

italy=# \q

```

```

In [ ]: from sqlalchemy.orm import Session
        from sqlalchemy import select
        session = Session(engine)
        stmt = select(Provincia).where(Provincia.regione_id == 1)
        print(stmt)
        for provincia in session.scalars(stmt) :
            print(provincia)

```

```

SELECT provincia.id, provincia.name, provincia.regione_id, provincia.residen
ti
FROM provincia
WHERE provincia.regione_id = :regione_id_1
Provincia(id='PU', name='Pesaro e Urbino', residenti=None, regione=1)

```

```

In [ ]: an = Provincia()
        # dir(an)
        an.id = "AN"
        an.name = "Ancona"
        an.regione_id = 1

```

```

In [ ]: print(an)

```

```

Provincia(id='AN', name='Ancona', residenti=None, regione=1)

```

```

In [ ]: session.add_all([an])
        session.commit()

```

```

In [ ]: stmt = select(Provincia).where(Provincia.regione_id == 1)
        print(stmt)
        for provincia in session.scalars(stmt) :
            print(provincia)

```

```

SELECT provincia.id, provincia.name, provincia.regione_id, provincia.residen
ti
FROM provincia
WHERE provincia.regione_id = :regione_id_1
Provincia(id='PU', name='Pesaro e Urbino', residenti=None, regione=1)
Provincia(id='AN', name='Ancona', residenti=None, regione=1)

```

```

In [ ]: stmt = (
        select(Provincia)
        .join(Provincia.regione)
        .where(Regione.name == "Marche")
        )
        print(stmt)

```

```
for provincia in session.scalars(stmt) :
    print(provincia)
```

```
SELECT provincia.id, provincia.name, provincia.regione_id, provincia.residen
ti
FROM provincia JOIN regione ON regione.id = provincia.regione_id
WHERE regione.name = :name_1
Provincia(id='PU', name='Pesaro e Urbino', residenti=None, regione=1)
Provincia(id='AN', name='Ancona', residenti=None, regione=1)
```

```
In [ ]: from csv import reader
infile = open("regioni.csv")
csvReader = reader(infile)
regioni = set()
header = True
for riga in csvReader :
    if header :
        header = False
        continue
    regioni.add(riga[1])
infile.close()
print(regioni)
```

```
{'Sardegna', 'Emilia-Romagna', 'Valle d'Aosta', 'Lombardia', 'Liguria', 'Cal
abria', 'Basilicata', 'Trentino-Alto Adige', 'Molise', 'Sicilia', 'Lazio', '
Puglia', 'Abruzzo', 'Campania', 'Toscana', 'Marche', 'Umbria', 'Friuli-Venez
ia Giulia', 'Veneto', 'Piemonte'}
```

```
In [ ]: regs = []
for r in regioni :
    regs.append(Regione(name=r))
session.add_all(regs)
session.commit()
```

```
In [ ]: !echo "DROP DATABASE italy;\q" | psql postgres
```

```
DROP DATABASE
```

```
In [4]: !echo "CREATE DATABASE italy;\q" | psql postgres
```

```
CREATE DATABASE
```

## Ripeto il tutto

```
In [16]: from typing import List
from typing import Optional
from sqlalchemy import ForeignKey
from sqlalchemy import String
from sqlalchemy.orm import DeclarativeBase, Mapped, mapped_column, relations

class Base(DeclarativeBase):
    pass

class Provincia(Base):
```

```

__tablename__ = "provincia"
id: Mapped[str] = mapped_column(String(2), primary_key=True)
name: Mapped[str] = mapped_column(String(30), nullable=True)
regione_id: Mapped[int] = mapped_column(ForeignKey("regione.id"), nullable=True)
regione: Mapped["Regione"] = relationship(back_populates="province")
residenti: Mapped[int] = mapped_column(nullable=True)

def __repr__(self) -> str:
    return f"Provincia(id={self.id!r}, name={self.name!r}, residenti={se

class Regione(Base):
    __tablename__ = "regione"
    id: Mapped[int] = mapped_column(primary_key=True)
    name: Mapped[str] = mapped_column(String(30))
    province: Mapped[List["Provincia"]] = relationship(
        back_populates="regione", cascade="all, delete-orphan"
    )

from sqlalchemy import create_engine
engine = create_engine("postgresql+psycopg2://localhost/italy")

Base.metadata.create_all(engine)

```

```

In [7]: from csv import reader
infile = open("regioni.csv")
csvReader = reader(infile)
regioni = set()
header = True
for riga in csvReader :
    if header :
        header = False
        continue
    regioni.add(riga[1])
infile.close()
print(regioni)

```

```

{'Friuli-Venezia Giulia', 'Molise', 'Sardegna', 'Trentino-Alto Adige', 'Sicilia', 'Emilia-Romagna', 'Umbria', 'Puglia', 'Basilicata', 'Toscana', 'Liguria', 'Calabria', 'Campania', 'Abruzzo', 'Veneto', 'Lazio', 'Piemonte', 'Valle d'Aosta', 'Lombardia', 'Marche'}

```

```

In [1]: from sqlalchemy.orm import Session
session = Session(engine)
regs = []
for r in regioni :
    regs.append(Regione(name=r))
session.add_all(regs)
session.commit()

```

```

In [12]: !echo "SELECT * FROM regione;\q" | psql italy

```

id	name
1	Friuli-Venezia Giulia
2	Molise
3	Sardegna
4	Trentino-Alto Adige
5	Sicilia
6	Emilia-Romagna
7	Umbria
8	Puglia
9	Basilicata
10	Toscana
11	Liguria
12	Calabria
13	Campania
14	Abruzzo
15	Veneto
16	Lazio
17	Piemonte
18	Valle d'Aosta
19	Lombardia
20	Marche

(20 rows)

```
In [18]: from csv import reader
infile = open("regioni.csv")
csvReader = reader(infile)
prov = []
regs = []
from sqlalchemy.orm import Session
from sqlalchemy import select
session = Session(engine)

header = True
for riga in csvReader :
    if header :
        header = False
        continue
    stmt = select(Regione).where(Regione.name == riga[1])
    prov.append(Provincia(id=riga[0], regione=session.scalars(stmt).one()))

infile.close()
session.add_all(prov)
session.commit()
```

```
/var/folders/_3/3xzqsl3977v6k_0_c7rq6pr80000gn/T/ipykernel_51866/179900146
2.py:16: SAWarning: Object of type <Provincia> not in session, add operation
along 'Regione.province' will not proceed (This warning originated from the
Session 'autoflush' process, which was invoked automatically in response to
a user-initiated operation. Consider using ``no_autoflush`` context manager
if this warning happended while initializing objects.)
    prov.append(Provincia(id=riga[0], regione=session.scalars(stmt).one()))
```

```
In [17]: !echo "DELETE FROM provincia;\q" | psql italy
```

```
DELETE 107
```

```
In [1]: !echo "SELECT * FROM provincia;\q" | psql italy
```

id	name	regione_id	residenti
CH		14	
TE		14	
AQ		14	
PE		14	
PZ		9	
MT		9	
VV		12	
CS		12	
CZ		12	
RC		12	
KR		12	
SA		13	
AV		13	
BN		13	
CE		13	
NA		13	
PR		6	
MO		6	
FE		6	
FC		6	
BO		6	
RA		6	
RE		6	
RN		6	
PC		6	
TS		1	
UD		1	
GO		1	
PN		1	
RI		16	
LT		16	
RM		16	
FR		16	
VT		16	
SV		11	
GE		11	
IM		11	
SP		11	
MI		19	
BS		19	
PV		19	
SO		19	
MB		19	
MN		19	
VA		19	
LO		19	
BG		19	
LC		19	
CO		19	
CR		19	
PU		20	
MC		20	
AN		20	
AP		20	



FM		20
CB		2
IS		2
AL		17
BI		17
NO		17
TO		17
VC		17
CN		17
AT		17
VB		17
BA		8
BT		8
BR		8
FG		8
TA		8
LE		8
SS		3
OR		3
NU		3
CA		3
SU		3
SR		5
TP		5
AG		5
RG		5
EN		5
ME		5
CL		5
PA		5
CT		5
GR		10
LI		10
LU		10
MS		10
AR		10
PI		10
SI		10
FI		10
PT		10
PO		10
TN		4
BZ		4
PG		7
TR		7
A0		18
BL		15
VI		15
TV		15
VE		15
RO		15
VR		15
PD		15
(107 rows)		

```
In [91]: from csv import reader
infile = open("regioni.csv")
known = []
csvReader0 = reader(infile)
header = True
for riga in csvReader0 :
    if header :
        header = False
        continue
    known.append(riga[0])
infile.close()
infile = open("province.csv")
csvReader = reader(infile)
from sqlalchemy.orm import Session
session = Session(engine)
from sqlalchemy import update
actions = []

header = True
for riga in csvReader :
    if header :
        header = False
        continue
    if riga[0] in known :
        actions.append({"id": riga[0], "name": riga[1], "residenti": int(rig
infile.close()
# session.bulk_update_mappings(Provincia, actions)
# session(commit)
print(actions)
```

```
[{'id': 'RM', 'name': 'Roma', 'residenti': 4353738}, {'id': 'MI', 'name': 'Milano', 'residenti': 3218201}, {'id': 'NA', 'name': 'Napoli', 'residenti': 3107006}, {'id': 'TO', 'name': 'Torino', 'residenti': 2277857}, {'id': 'PA', 'name': 'Palermo', 'residenti': 1268217}, {'id': 'BS', 'name': 'Brescia', 'residenti': 1262318}, {'id': 'BA', 'name': 'Bari', 'residenti': 1260142}, {'id': 'CT', 'name': 'Catania', 'residenti': 1113303}, {'id': 'BG', 'name': 'Bergamo', 'residenti': 1109933}, {'id': 'SA', 'name': 'Salerno', 'residenti': 1104731}, {'id': 'FI', 'name': 'Firenze', 'residenti': 1014423}, {'id': 'BO', 'name': 'Bologna', 'residenti': 1009210}, {'id': 'PD', 'name': 'Padova', 'residenti': 936274}, {'id': 'CE', 'name': 'Caserta', 'residenti': 924166}, {'id': 'VR', 'name': 'Verona', 'residenti': 921557}, {'id': 'VA', 'name': 'Varese', 'residenti': 890043}, {'id': 'TV', 'name': 'Treviso', 'residenti': 885972}, {'id': 'MB', 'name': 'Monza e della Brianza', 'residenti': 868859}, {'id': 'VI', 'name': 'Vicenza', 'residenti': 865082}, {'id': 'VE', 'name': 'Venezia', 'residenti': 854275}, {'id': 'GE', 'name': 'Genova', 'residenti': 850071}, {'id': 'LE', 'name': 'Lecce', 'residenti': 802082}, {'id': 'CS', 'name': 'Cosenza', 'residenti': 711739}, {'id': 'MO', 'name': 'Modena', 'residenti': 700862}, {'id': 'PG', 'name': 'Perugia', 'residenti': 660690}, {'id': 'ME', 'name': 'Messina', 'residenti': 636653}, {'id': 'FG', 'name': 'Foggia', 'residenti': 628556}, {'id': 'CO', 'name': 'Como', 'residenti': 600190}, {'id': 'CN', 'name': 'Cuneo', 'residenti': 589108}, {'id': 'TA', 'name': 'Taranto', 'residenti': 583479}, {'id': 'LT', 'name': 'Latina', 'residenti': 574891}, {'id': 'CA', 'name': 'Cagliari', 'residenti': 560373}, {'id': 'RC', 'name': 'Reggio Calabria', 'residenti': 553861}, {'id': 'PV', 'name': 'Pavia', 'residenti': 547251}, {'id': 'TN', 'name': 'Trento', 'residenti': 538604}, {'id': 'RE', 'name': 'Reggio Emilia', 'residenti': 532483}, {'id': 'UD', 'name': 'Udine', 'residenti': 531466}, {'id': 'BZ', 'name': 'Bolzano', 'residenti': 524256}, {'id': 'FR', 'name': 'Frosinone', 'residenti': 493067}, {'id': 'AN', 'name': 'Ancona', 'residenti': 474124}, {'id': 'PR', 'name': 'Parma', 'residenti': 448899}, {'id': 'AG', 'name': 'Agrigento', 'residenti': 442049}, {'id': 'TP', 'name': 'Trapani', 'residenti': 434476}, {'id': 'AL', 'name': 'Alessandria', 'residenti': 426658}, {'id': 'AV', 'name': 'Avellino', 'residenti': 423506}, {'id': 'PI', 'name': 'Pisa', 'residenti': 421851}, {'id': 'MN', 'name': 'Mantova', 'residenti': 412610}, {'id': 'SR', 'name': 'Siracusa', 'residenti': 402822}, {'id': 'BR', 'name': 'Brindisi', 'residenti': 397083}, {'id': 'FC', 'name': 'Forlì-Cesena', 'residenti': 394067}, {'id': 'BT', 'name': 'Barletta-Andria-Trani', 'residenti': 392546}, {'id': 'RA', 'name': 'Ravenna', 'residenti': 391414}, {'id': 'LU', 'name': 'Lucca', 'residenti': 390042}, {'id': 'CH', 'name': 'Chieti', 'residenti': 389169}, {'id': 'PZ', 'name': 'Potenza', 'residenti': 370680}, {'id': 'NO', 'name': 'Novara', 'residenti': 370143}, {'id': 'CZ', 'name': 'Catanzaro', 'residenti': 362343}, {'id': 'PU', 'name': 'Pesaro e Urbino', 'residenti': 360711}, {'id': 'CR', 'name': 'Cremona', 'residenti': 359388}, {'id': 'FE', 'name': 'Ferrara', 'residenti': 348362}, {'id': 'AR', 'name': 'Arezzo', 'residenti': 344374}, {'id': 'LC', 'name': 'Lecco', 'residenti': 339238}, {'id': 'LI', 'name': 'Livorno', 'residenti': 337334}, {'id': 'RN', 'name': 'Rimini', 'residenti': 336786}, {'id': 'SS', 'name': 'Sassari', 'residenti': 333116}, {'id': 'RG', 'name': 'Ragusa', 'residenti': 321359}, {'id': 'PE', 'name': 'Pescara', 'residenti': 321309}, {'id': 'VT', 'name': 'Viterbo', 'residenti': 319008}, {'id': 'MC', 'name': 'Macerata', 'residenti': 318921}, {'id': 'PN', 'name': 'Pordenone', 'residenti': 312051}, {'id': 'TE', 'name': 'Teramo', 'residenti': 309859}, {'id': 'AQ', 'name': 'L'Aquila', 'residenti': 301910}, {'id': 'PT', 'name': 'Pistoia', 'residenti': 291839}, {'id': 'PC', 'name': 'Piacenza', 'residenti': 286758}, {'id': 'BN', 'name': 'Benevento', 'residenti': 279675}, {'id': 'SV', 'name': 'Savona', 'residenti': 279408}, {'id': 'CL', 'name': 'Caltanissetta', 'residenti': 269710}, {'id': 'SI', 'name': 'Siena', 'residenti': 2683}
```

```
41}, {'id': 'PO', 'name': 'Prato', 'residenti': 254608}, {'id': 'RO', 'name': 'Rovigo', 'residenti': 238588}, {'id': 'TS', 'name': 'Trieste', 'residenti': 234682}, {'id': 'LO', 'name': 'Lodi', 'residenti': 229338}, {'id': 'TR', 'name': 'Terni', 'residenti': 228218}, {'id': 'CB', 'name': 'Campobasso', 'residenti': 224644}, {'id': 'GR', 'name': 'Grosseto', 'residenti': 223045}, {'id': 'SP', 'name': 'La Spezia', 'residenti': 220698}, {'id': 'AT', 'name': 'Asti', 'residenti': 216677}, {'id': 'IM', 'name': 'Imperia', 'residenti': 215130}, {'id': 'AP', 'name': 'Ascoli Piceno', 'residenti': 209450}, {'id': 'BL', 'name': 'Belluno', 'residenti': 205781}, {'id': 'MT', 'name': 'Matera', 'residenti': 199685}, {'id': 'MS', 'name': 'Massa-Carrara', 'residenti': 196580}, {'id': 'SO', 'name': 'Sondrio', 'residenti': 181437}, {'id': 'BI', 'name': 'Biella', 'residenti': 178551}, {'id': 'KR', 'name': 'Crotone', 'residenti': 175566}, {'id': 'FM', 'name': 'Fermo', 'residenti': 174849}, {'id': 'VC', 'name': 'Vercelli', 'residenti': 173868}, {'id': 'EN', 'name': 'Enna', 'residenti': 168052}, {'id': 'VV', 'name': 'Vibo Valentia', 'residenti': 161619}, {'id': 'OR', 'name': 'Oristano', 'residenti': 160746}, {'id': 'VB', 'name': 'Verbano-Cusio-Ossola', 'residenti': 159664}, {'id': 'RI', 'name': 'Rieti', 'residenti': 157420}, {'id': 'NU', 'name': 'Nuoro', 'residenti': 156096}, {'id': 'GO', 'name': 'Gorizia', 'residenti': 139673}, {'id': 'AO', 'name': 'Aosta', 'residenti': 126883}, {'id': 'IS', 'name': 'Isernia', 'residenti': 85805}]
```

```
In [10]: session = Session(engine)
         session.bulk_update_mappings(Provincia, actions)
         session.commit()
```

```
In [13]: !echo "SELECT p.id, p.name, r.name, p.residenti FROM provincia p, regione r"
```

id	name	name	residenti
SU		Sardegna	
RM	Roma	Lazio	4353738
MI	Milano	Lombardia	3218201
NA	Napoli	Campania	3107006
TO	Torino	Piemonte	2277857
PA	Palermo	Sicilia	1268217
BS	Brescia	Lombardia	1262318
BA	Bari	Puglia	1260142
CT	Catania	Sicilia	1113303
BG	Bergamo	Lombardia	1109933
SA	Salerno	Campania	1104731
FI	Firenze	Toscana	1014423
BO	Bologna	Emilia-Romagna	1009210
PD	Padova	Veneto	936274
CE	Caserta	Campania	924166
VR	Verona	Veneto	921557
VA	Varese	Lombardia	890043
TV	Treviso	Veneto	885972
MB	Monza e della Brianza	Lombardia	868859
VI	Vicenza	Veneto	865082
VE	Venezia	Veneto	854275
GE	Genova	Liguria	850071
LE	Lecce	Puglia	802082
CS	Cosenza	Calabria	711739
MO	Modena	Emilia-Romagna	700862
PG	Perugia	Umbria	660690
ME	Messina	Sicilia	636653
FG	Foggia	Puglia	628556
CO	Como	Lombardia	600190
CN	Cuneo	Piemonte	589108
TA	Taranto	Puglia	583479
LT	Latina	Lazio	574891
CA	Cagliari	Sardegna	560373
RC	Reggio Calabria	Calabria	553861
PV	Pavia	Lombardia	547251
TN	Trento	Trentino-Alto Adige	538604
RE	Reggio Emilia	Emilia-Romagna	532483
UD	Udine	Friuli-Venezia Giulia	531466
BZ	Bolzano	Trentino-Alto Adige	524256
FR	Frosinone	Lazio	493067
AN	Ancona	Marche	474124
PR	Parma	Emilia-Romagna	448899
AG	Agrigento	Sicilia	442049
TP	Trapani	Sicilia	434476
AL	Alessandria	Piemonte	426658
AV	Avellino	Campania	423506
PI	Pisa	Toscana	421851
MN	Mantova	Lombardia	412610
SR	Siracusa	Sicilia	402822
BR	Brindisi	Puglia	397083
FC	Forlì-Cesena	Emilia-Romagna	394067
BT	Barletta-Andria-Trani	Puglia	392546
RA	Ravenna	Emilia-Romagna	391414
LU	Lucca	Toscana	390042

CH	Chieti	Abruzzo	389169
PZ	Potenza	Basilicata	370680
NO	Novara	Piemonte	370143
CZ	Catanzaro	Calabria	362343
PU	Pesaro e Urbino	Marche	360711
CR	Cremona	Lombardia	359388
FE	Ferrara	Emilia-Romagna	348362
AR	Arezzo	Toscana	344374
LC	Lecco	Lombardia	339238
LI	Livorno	Toscana	337334
RN	Rimini	Emilia-Romagna	336786
SS	Sassari	Sardegna	333116
RG	Ragusa	Sicilia	321359
PE	Pescara	Abruzzo	321309
VT	Viterbo	Lazio	319008
MC	Macerata	Marche	318921
PN	Pordenone	Friuli-Venezia Giulia	312051
TE	Teramo	Abruzzo	309859
AQ	L'Aquila	Abruzzo	301910
PT	Pistoia	Toscana	291839
PC	Piacenza	Emilia-Romagna	286758
BN	Benevento	Campania	279675
SV	Savona	Liguria	279408
CL	Caltanissetta	Sicilia	269710
SI	Siena	Toscana	268341
PO	Prato	Toscana	254608
RO	Rovigo	Veneto	238588
TS	Trieste	Friuli-Venezia Giulia	234682
LO	Lodi	Lombardia	229338
TR	Terni	Umbria	228218
CB	Campobasso	Molise	224644
GR	Grosseto	Toscana	223045
SP	La Spezia	Liguria	220698
AT	Asti	Piemonte	216677
IM	Imperia	Liguria	215130
AP	Ascoli Piceno	Marche	209450
BL	Belluno	Veneto	205781
MT	Matera	Basilicata	199685
MS	Massa-Carrara	Toscana	196580
SO	Sondrio	Lombardia	181437
BI	Biella	Piemonte	178551
KR	Crotone	Calabria	175566
FM	Fermo	Marche	174849
VC	Vercelli	Piemonte	173868
EN	Enna	Sicilia	168052
VV	Vibo Valentia	Calabria	161619
OR	Oristano	Sardegna	160746
VB	Verbano-Cusio-Ossola	Piemonte	159664
RI	Rieti	Lazio	157420
NU	Nuoro	Sardegna	156096
GO	Gorizia	Friuli-Venezia Giulia	139673
AO	Aosta	Valle d'Aosta	126883
IS	Isernia	Molise	85805
(107 rows)			