UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

BASI DI DATI

GIULIANO DI GIUSEPPE N46004374
SOSSIO CIRILLO N46004229

INDICE

1 INTRODUZIONE

Progetto database sui contagi del COVID-19 dal 25/02/2020 al 03/05/2020 in Italia.

Il coronavirus è un virus influenzale la cui diffusione ha avuto l'epicentro a Wuhan, in Cina, nel dicembre 2019.
Nel gennaio 2020 è stata allertata l'Organizzazione Mondiale della Sanità e solo a fine gennaio/inizio febbraio, a causa dell'elevato numero di contagi nel nord Italia, la pericolosità del virus è diventata nota a tutti e sono iniziate le prime misure restrittive.
Essendo un virus di base influenzale, tra i sintomi si annoverano in particolare febbre alta, difficoltà respiratorie, stanchezza, debolezza, ma ogni individuo può mostrare sintomi diversi o addirittura essere asintomatico.

Questo progetto si occuperà della ricerca, manipolazione e interrogazione dei dati riguardanti i contagi del COVID-19 per darci una visione molto precisa riguardante la pericolosità del virus.

Portale malattie infettive:
http://www.salute.gov.it/portale/malatitudinetieInfettive/dettaglioFaqMalatitudinetieInfettive.jsp?lingua=italiano&id=228

2 CREAZIONE DATABASE

2.1 RACCOLTA E PULIZIA DATI

La raccolta dati riguardanti i contagi giornalieri in ogni provincia italiana si trovano sul seguente sito: https://github.com/pcm-dpc/COVID-19/tree/master/dati-PROVINCIA. Come da traccia, sono stati scaricati i dati del periodo che va dal 25/02/2020 al 03/05/2020.

L'attributo *data* è scritto nella seguente forma 'AAAA-MM-GGTHH-MM-SS' nei file precedentemente scaricati, quindi lo modifichiamo e lo rendiamo nella forma 'AAAA-MM-GG' in modo tale che possiamo considerare l'attributo data come variabile di tipo DATE e non come variabile di tipo VARCHAR(). Per modificare l'attributo *data,* si possono aprire i file .csv in Power Point ed operare la modifica.

2.2 CONVERSIONE .CSV A SQL

Converto i dati scaricati dal precedente sito da CSV in comandi SQL usando il tool https://www.convertcsv.com/csv-to-sql.htm. Creo la tabella e inserisco i dati nel database tramite la console.

Creazione tabella:

```
CREATE TABLE MASTER (
    data                    DATE NOT NULL,
    stato                   VARCHAR(3),
    codice_regione          INTEGER,
    denominazione_regione   VARCHAR(30),
    codice_provincia        INTEGER  NOT NULL,
    denominazione_provincia VARCHAR(40),
    sigla_provincia         VARCHAR(2),
    latitudine              NUMERIC(11,9),
    longitudine             NUMERIC(11,9),
    totale_casi             INTEGER,
    note_it                 VARCHAR(30),
    note_en                 VARCHAR(30),
    PRIMARY KEY (data, codice_provincia )
);
```

Esempio inserimento dati:

```
INSERT INTO MASTER(data,stato,codice_regione,denominazione_regione,codice_provincia,
                   denominazione_provincia,sigla_provincia,latitudine,longitudine,totale_casi,note_it,note_en)
                   VALUES ('2020-02-25','ITA',13,'Abruzzo',069,'Chieti','CH',42.35103167,14.16754574,0,NULL,NULL);
INSERT INTO MASTER(data,stato,codice_regione,denominazione_regione,codice_provincia,
                   denominazione_provincia,sigla_provincia,latitudine,longitudine,totale_casi,note_it,note_en)
                   VALUES ('2020-02-25','ITA',13,'Abruzzo',066,'L''Aquila','AQ',42.35122196,13.39843823,0,NULL,NULL);
```

Un altro metodo è inserire i dati all'interno della tabella direttamente dal file .csv è usando il comando 'Import Data from File'.

2.3 VIOLAZIONE DI VINCOLI

I dati scaricati, presentano alcuni problemi:

- le province autonome Trento e Bolzano hanno lo stesso *codice_regione* ma diversa *denominazione_regione*; ciò provoca una violazione del vincolo d'integrità. Quindi cambio la

denominazione_regione di Trento e Bolzano in 'Trentino-Alto Adige' tramite in seguente comando:

```
UPDATE MASTER SET DENOMINAZIONE_REGIONE='Trentino-Alto Adige'
where DENOMINAZIONE_PROVINCIA='Trento' or denominazione_provincia='Bolzano';
```

- molte provincie non comunicano i dati nei tempi richiesti, quindi abbiamo che alcune tuple hanno all'attributo *denominazione_provincia* il valore 'In fase di definizione/aggiornamento', quindi ci conviene pulire la tabella eliminando queste tuple in eccesso tramite il seguente comando:

```
DELETE FROM MASTER WHERE denominazione_provincia = 'In fase di definizione/aggiornamento';
```

3 VERIFICA DELLA 3NF

3.1 VERIFICA

La tabella da verificare è: **MASTER** (*data, stato, codice_regione, denominazione_regione, codice_provincia, denominazione_provincia, sigla_provincia, latitudine, long, totale_casi, note_it, note_en*)

- La prima forma normale è verificata poiché ogni attributo appartenente a MASTER è un attributo semplice, cioè non è né un attributo multivalore né un attributo composto.

- Per verificare se è in seconda forma normale (2NF) bisogna specificare la chiave dello schema e le relative dipendenze funzionali:
    1. Primary key = (*data*, *codice_provincia*),
    2. FD: (*data*, *codice_provincia*) → (*stato*, *codice_regione*, *denominazione_regione*, *codice_provincia*, *denominazione_provincia*, *sigla_provincia*, *latitudine*, *longitudine*, *totale_casi*, *note_it, note_en*)
    3. *codice_provincia* → (*denominazione_provincia*, *sigla_provincia*, *latitudine*, *longitudine*, *stato, codice_regione, denominazione_regione*)

Lo schema non è in 2NF poiché, per esserlo, ogni attributo dovrebbe dipendere da tutta la chiave primaria, cioè da entrambi gli attributi.

Per normalizzarla bisogna operare la seguente decomposizione:

PROVINCIA (*codice_provincia, denominazione_provincia, sigla_provincia, latitudine, longitudine, codice_regione, denominazione_regione, Stato*)

ANDAMENTO_PROVINCIA (*data, codice_provincia, totale_casi , note_it, note_en*)

- Per verificare se è in terza forma normale (3NF) controllo se ci sono dipendenze transitive.
  La relazione ANDAMENTO_PROVINCIA è già in 3NF, quindi non bisogna fare nessuna ulteriore decomposizione.
  In PROVINCIA si ha una dipendenza transitiva causata dalla seguente dipendenza funzionale
  codice_regione → (*stato, denominazione_regione*)

  Per normalizzarla bisogna operare la seguente decomposizione:
  ANDAMENTO_PROVINCIA (*data, codice_provincia*: PROVINCIA, t*otale_casi , note_it, note_en*)
  REGIONE (*codice_regione, denominazione_regione, Stato*)
  PROVINCIA (*codice_provincia, denominazione_provincia, sigla_provincia, latitudine, longitudine, codice_regione*: REGIONE).

3.2 DECOMPOSIZIONE TRAMIDE COMANDI DDL

Creazione tabelle:

```
CREATE TABLE REGIONI(
    codice_regione INTEGER NOT NULL,
    denominazione_regione VARCHAR(50),
    stato VARCHAR(3),
    PRIMARY KEY (codice_regione)
);

CREATE TABLE PROVINCE(
    codice_provincia INTEGER NOT NULL,
    denominazione_provincia VARCHAR(50) NOT NULL,
    sigla_provincia VARCHAR(2),
    latitudine NUMERIC (11,8) NOT NULL ,
    longitudine NUMERIC (11,9) NOT NULL,
    codice_regione INTEGER NOT NULL,
    PRIMARY KEY (codice_provincia),
    FOREIGN KEY (codice_regione) REFERENCES REGIONI(codice_regione)
);

CREATE TABLE ANDAMENTO_PROVINCE (
    data DATE NOT NULL,
    codice_provincia INTEGER NOT NULL,
    totale_casi INTEGER NOT NULL,
    note_it CLOB,
    note_en CLOB,
    PRIMARY KEY (data, codice_provincia),
    FOREIGN KEY (codice_provincia) REFERENCES PROVINCE(codice_provincia)
);
```

Popolamento tabelle:

```sql
INSERT INTO REGIONI
SELECT DISTINCT codice_regione, denominazione_regione, stato
FROM MASTER
ORDER BY codice_regione;

INSERT INTO PROVINCE
SELECT DISTINCT codice_provincia, denominazione_provincia, sigla_provincia, latitudine, longitudine, codice_regione
FROM MASTER
ORDER BY codice_provincia;

INSERT INTO ANDAMENTO_PROVINCE
SELECT DISTINCT data, codice_provincia, totale_casi, note_it, note_en
FROM MASTER
ORDER BY data;
```

4 ARRICCHIMENTO SCHEMA

L' arricchimento dello schema consiste nell'aggiungere allo schema delle informazioni utili che successivamente saranno utilizzate per un'analisi più dettagliata e profonda del fenomeno in questione.

4.1 ANDAMENTO REGIONI

Si crea una nuova tabella chiamata ANDAMENTO_REGIONE composta dai dati riguardanti l'andamento del virus a livello regionale. I dati vengono presi dal sito: https://github.com/pcm-dpc/COVID-19/blob/master/dati-REGIONE/dpc-covid19-ita-REGIONE.csv e vengono inseriti tramite l'opzione 'Import Data from File ' dal file .csv:

Creazione della tabella:

```sql
CREATE TABLE ANDAMENTO_REGIONI(
    data                        DATE NOT NULL,
    codice_regione              INTEGER NOT NULL,
    ricoverati_con_sintomi      INTEGER  NOT NULL,
    terapia_intensiva           INTEGER  NOT NULL,
    totale_ospedalizzati        INTEGER  NOT NULL,
    isolamento_domiciliare      INTEGER  NOT NULL,
    totale_positivi             INTEGER  NOT NULL,
    variazione_totale_positivi  INTEGER  NOT NULL,
    nuovi_positivi              INTEGER  NOT NULL,
    dimessi_guariti             INTEGER  NOT NULL,
    deceduti                    INTEGER  NOT NULL,
    totale_casi                 INTEGER  NOT NULL,
    tamponi                     INTEGER  NOT NULL,
    casi_testati                INTEGER ,
    note_it                     CLOB,
    note_en                     CLOB,
    PRIMARY KEY (data, codice_regione),
    FOREIGN KEY (codice_regione) REFERENCES REGIONI(codice_regione)
);
```

4.2 INFORMAZIONI PROVINCE

Vengono aggiunti alla relazione PROVINCIA gli attributi *superficie, residenti* e *num_comuni* tramite il seguente comando:

```
alter table PROVINCE ADD ( superficie NUMBER (8,3),
                           residenti integer,
                           num_comuni integer);
```

I nuovi dati vengono inseriti tramite il comando UPDATE.

Esempio di comando UPDATE:

```
UPDATE PROVINCE SET sigla_provincia = 'AG',denominazione_provincia = 'Agrigento',superficie = '3044,85',
                    residenti = 446_081,num_comuni = 43 WHERE sigla_provincia= 'AG';
UPDATE PROVINCE SET sigla_provincia = 'AL',denominazione_provincia = 'Alessandria',superficie = '3560,42',
                    residenti = 427_354,num_comuni = 190 WHERE sigla_provincia= 'AL';
```

4.3 INFORMAZIONI REGIONE

Vengono aggiunti alla relazione REGIONI gli attributi *numero_abitanti, densita_abitativa, superficie, numero_aereoporti, numero_comuni, numero_province* e *presidente* tramite i seguenti comandi:

```
ALTER TABLE REGIONI ADD (numero_abitanti INTEGER,
                         densita_abitativa INTEGER,
                         superficie DECIMAL(7,2),
                         numero_aereoporti INTEGER,
                         numero_comuni INTEGER,
                         numero_province INTEGER,
                         presidente VARCHAR2(50) );
```

I nuovi dati vengono inseriti tramite il comando UPDATE.

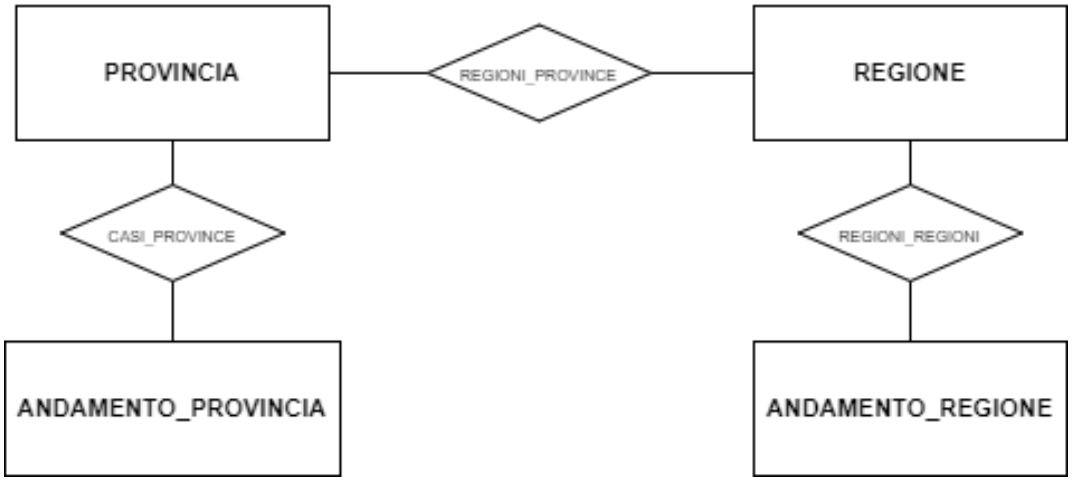
Esempio di comando UPDATE:

```
UPDATE REGIONI SET CODICE_REGIONE = 1,DENOMINAZIONE_REGIONE = 'Piemonte',STATO = 'ITA',NUMERO_ABITANTI = 4404246,
                   DENSITA_ABITATIVA = 173,SUPERFICIE = 25387.07,NUMERO_AEREOPORTI = 10,NUMERO_COMUNI = 1201,NUMERO_PROVINCE = 8,
                   PRESIDENTE = 'Sergio Chiamparino' WHERE CODICE_REGIONE= 1;
UPDATE REGIONI SET CODICE_REGIONE = 2,DENOMINAZIONE_REGIONE = 'Valle d''Aosta',STATO = 'ITA',NUMERO_ABITANTI = 127329,
                   DENSITA_ABITATIVA = 39,SUPERFICIE = 3260.90,NUMERO_AEREOPORTI = 1,NUMERO_COMUNI = 74,NUMERO_PROVINCE = 1,
                   PRESIDENTE = 'Augusto Rollandin' WHERE CODICE_REGIONE= 2;
```

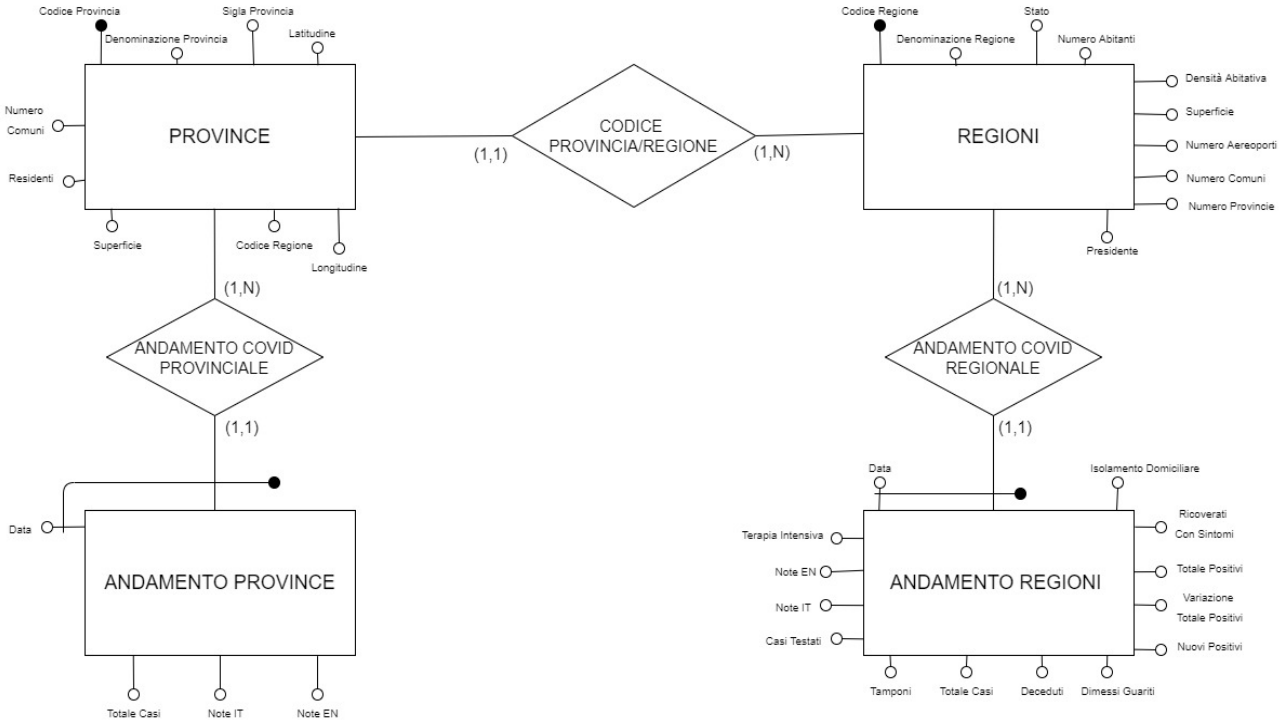
5 SCHEMA CONCETTUALE

Attraverso un processo di reverse engineering è possibile ricondursi ad uno schema concettuale E/R della base di dati.

5.1 SCHEMA SCHELETRO



5.2 SCHEMA COMPLETO



6 SPECIFICHE IN SQL

Ai fini della comprensione globale dell'andamento del contagio, possono essere utili le query, con annesse spiegazioni e risultati, che riportiamo di seguito.

6.1 QUERY

```sql
--QUERY1
--SELEZIONA DATA,DENOMINAZIONE_PROVINCIA E TOTALI CASI IN UNA DETERMINATA GIORNATA
SELECT a.data , p.denominazione_provincia ,a.totale_casi
FROM ANDAMENTO_PROVINCE A JOIN PROVINCE P ON A.codice_provincia=P.codice_provincia
WHERE A.DATA = '10-APR-2020'
ORDER BY P.denominazione_provincia ;
```

Risultato: QUERY\QUERY1.xlsx

```sql
--QUERY2
--SELEZIONA LA PROVINCIA CON IL NUMERO MASSIMO DI CASI
SELECT A.data , p.denominazione_provincia ,a.totale_casi
FROM ANDAMENTO_PROVINCE A JOIN PROVINCE P ON A.codice_provincia=P.codice_provincia
WHERE A.TOTALE_CASI=( --posso specificare anche una determinata giornata
    SELECT MAX(totale_casi)
    FROM ANDAMENTO_PROVINCE
    );
```

Risultato: QUERY\QUERY2.xlsx

```sql
--QUERY3
--SELEZIONA LE PROVINCE DI UNA REGIONE E I RELATIVI ABITANTI
SELECT r.Denominazione_Regione ,p.denominazione_provincia , p.residenti
FROM REGIONI R JOIN PROVINCE P ON R.Codice_Regione=P.COD_REGIONE
WHERE R.DENOMINAZIONE_REGIONE='Campania' ;
```

Risultato: QUERY\QUERY3.xlsx

```sql
--QUERY4
--SELEZIONA LE REGIONI E LA SOMMA DEI NUOVI POSITVI PER OGNUNA IN UN ARCO TEMPORALE
SELECT  R.Denominazione_Regione ,SUM(A.nuovi_positivi) AS SOMMA_NUOVI_POSITIVI
FROM REGIONI R JOIN ANDAMENTO_REGIONI A ON R.Codice_Regione=A.codice_regione
WHERE A.DATA  BETWEEN '10-APR-2020' AND '20-APR-2020'
GROUP BY R.Denominazione_Regione
ORDER BY SUM(A.nuovi_positivi) DESC;
```

Risultato: QUERY\QUERY4.xlsx

```sql
--QUERY5
--SELEZIONA LA PROVINCIA CON PIU' RESIDENTI
SELECT*
FROM PROVINCE
WHERE PROVINCE.residenti=(
    SELECT MAX(PROVINCE.residenti)
    FROM PROVINCE
    ) ;
```

Risultato: QUERY\QUERY5.xlsx

```sql
--QUERY6
-- SELEZIONA PER OGNI REGIONE IL NUMERO TAMPONI,TOTALE_CASI,DECEDUTI,DIMESSI_GUARITI
-- FINO A UNA SPECIFICA DATA E ORDINA PER TAMPONI IN DECRESCENTE
SELECT R.Denominazione_Regione , A.totale_casi ,A.deceduti ,A.dimessi_guariti , A.tamponi
FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
WHERE  A.data ='20-APR-2020'
ORDER BY A.tamponi DESC ;
```

Risultato: QUERY\QUERY6.xlsx

```sql
--QUERY7
--SELEZIONA LA REGIONE CON IL NUMERO MAGGIORE DI PERSONE IN TERAPIA INTENSIVA
--IN UN ARCO TEMPORALE
SELECT A.DATA , R.Denominazione_Regione ,A.terapia_intensiva,  A.totale_casi ,
       A.deceduti ,A.dimessi_guariti , A.tamponi
FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
WHERE A.DATA  BETWEEN '1-APR-2020' AND '20-APR-2020' AND A.terapia_intensiva=(
    SELECT MAX(a.TERAPIA_INTENSIVA)
    FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
    WHERE A.DATA  BETWEEN '1-APR-2020' AND '20-APR-2020'
    );
```

Risultato: QUERY\QUERY7.xlsx

```sql
--QUERY8
--SELEZIONA LA TUPLA IN CUI IN LA 'CAMPANIA' HA IL NUMERO MASSIMO DI TOTALE OSPEDALIZZATI
--IN UN ARCO TEMPORALE
SELECT A.DATA , R.Denominazione_Regione ,A.totale_ospedalizzati  , A.terapia_intensiva,
       A.totale_casi ,A.deceduti ,A.dimessi_guariti , A.tamponi
FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
WHERE A.DATA  BETWEEN '1-APR-2020' AND '20-APR-2020'
    AND R.Denominazione_Regione='Campania' and  A.TOTALE_OSPEDALIZZATI=(
    SELECT MAX(a.totale_ospedalizzati)
    FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
    WHERE A.DATA  BETWEEN '1-APR-2020' AND '20-APR-2020' AND
          R.Denominazione_Regione='Campania'
    );
```

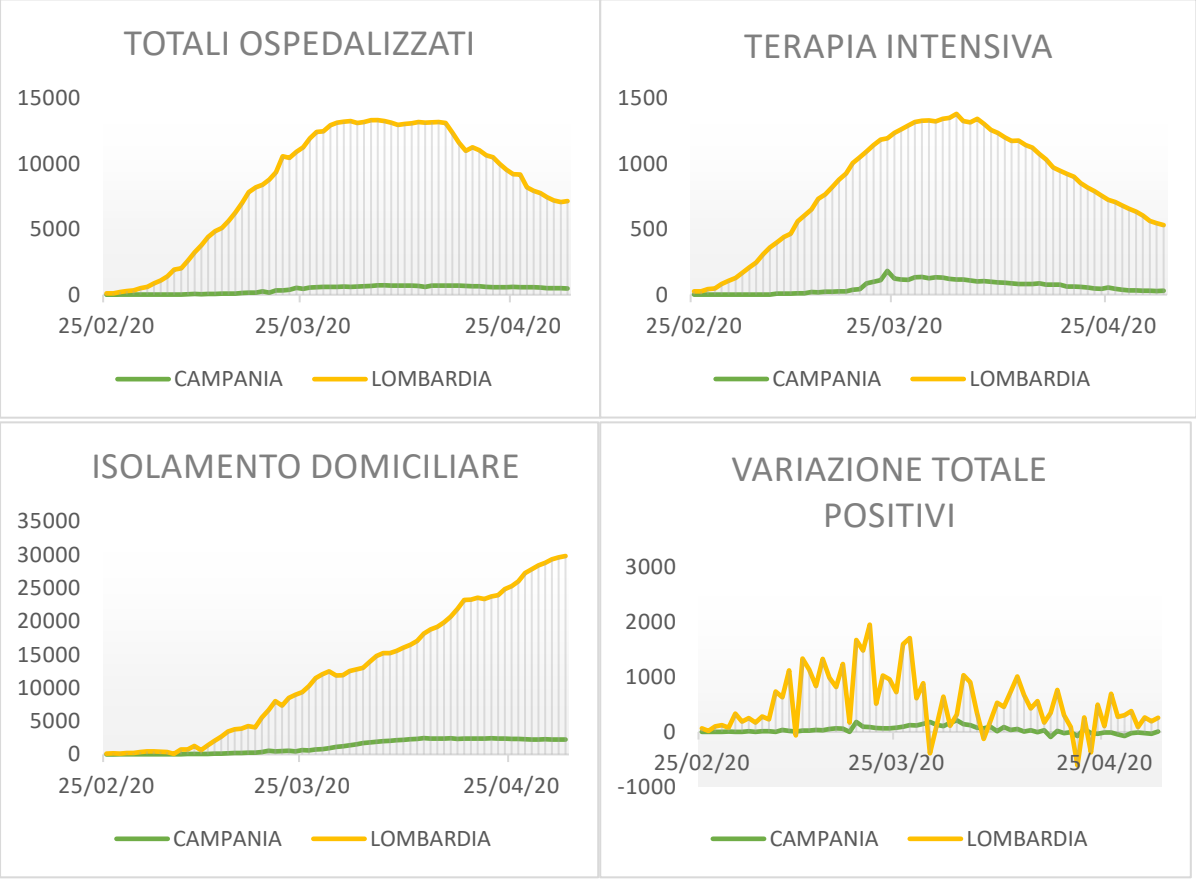
Risultato: QUERY\QUERY8.xlsx

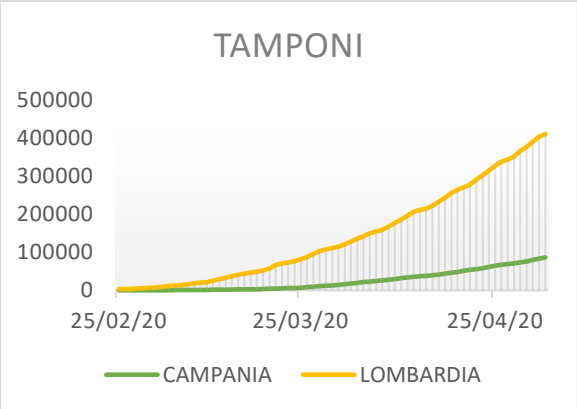
```sql
--QUERY9
--SELEZIONA DATA,REGIONE,CONTAGI E DECEDUTI DELLA DATA CON IL PICCO DEI CONTAGIATI
--DI UNA SPECIFICA REGIONE
select ANDAMENTO_REGIONI.data as Data_Del_Picco,  REGIONI.Denominazione_Regione ,
       ANDAMENTO_REGIONI.nuovi_positivi as Contagiati,
       ANDAMENTO_REGIONI.deceduti as Deceduti
from ANDAMENTO_REGIONI join REGIONI
    on ANDAMENTO_REGIONI.codice_regione = REGIONI.codice_regione
where REGIONI.denominazione_regione='Lombardia' and data = (
    select ANDAMENTO_REGIONI.data
    from ANDAMENTO_REGIONI
    where ANDAMENTO_REGIONI.nuovi_positivi >= all (
        select ANDAMENTO_REGIONI.nuovi_positivi
        from ANDAMENTO_REGIONI
        )
    );
```

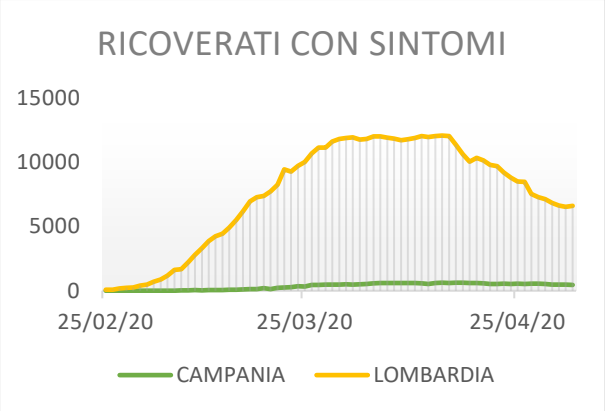
Risultato: QUERY\QUERY9.xlsx

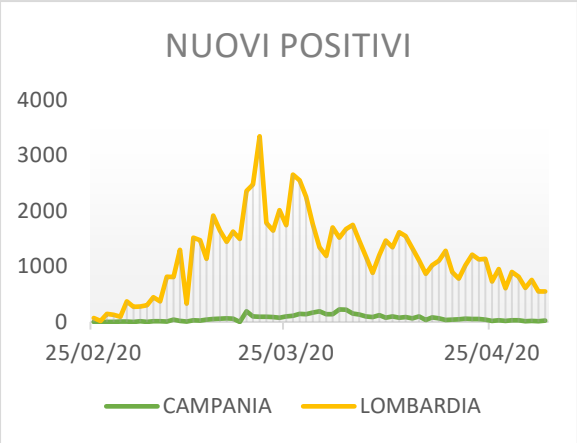
6.2 GRAFICI

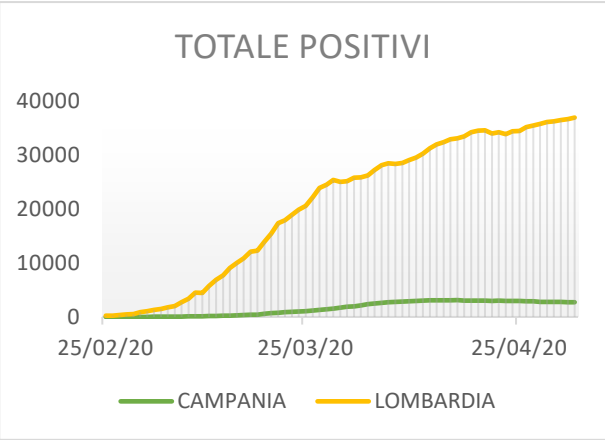
Di seguito riportiamo dei grafici comparati tra la regione Campania e la regione Lombardia riguardanti diversi dati.











7 SPECIFICHE IN PL/SQL

Per una ulteriore comprensione del contagio e per eventuali aggiornamenti automatici della base di dati, possono essere utili le procedure e i trigger, con annesse spiegazioni e risultati, che riportiamo di seguito.

7.1 PROCEDURA_1

```
--Stampa per ogni giorno  la regione e il numero di persone in terapia intensiva in una finestra temporale
create or replace PROCEDURE  procedura_1 (DAT1 IN ANDAMENTO_REGIONI.DATA%TYPE, DAT2 IN
ANDAMENTO_REGIONI.DATA%TYPE,
                            REG IN REGIONI.DENOMINAZIONE_REGIONE%TYPE) IS
CURSOR CURS_1
   IS SELECT A.DATA, A.CODICE_REGIONE,A.TERAPIA_INTENSIVA
   FROM ANDAMENTO_REGIONI A JOIN REGIONI R ON A.codice_regione = R.codice_regione
   WHERE REG=R.DENOMINAZIONE_REGIONE AND A.DATA>=DAT1 AND A.DATA<=DAT2; --CAMBIAMENTO
ANDAM CURS_1%ROWTYPE;
   BEGIN
   OPEN CURS_1;
   LOOP
      FETCH
      CURS_1 INTO ANDAM;
      EXIT WHEN CURS_1%NOTFOUND;
```

```
    DBMS_OUTPUT.ENABLE();
    DBMS_OUTPUT.PUT_LINE(  'Data:  ' ||RPAD(ANDAM.data,15) || 'Codice Regione:  '
||rpad(ANDAM.codice_regione,15)
                    || 'Terapia intensiva:  ' || rpad(ANDAM.terapia_intensiva,15));
  end loop;
  CLOSE CURS_1;
end;

CALL procedura_1('21-APR-2020', '30-APR-2020', 'Campania');
```

RISULTATO:

```
Data:  21-APR-20      Codice Regione:  15           Terapia intensiva:  58
Data:  22-APR-20      Codice Regione:  15           Terapia intensiva:  53
Data:  23-APR-20      Codice Regione:  15           Terapia intensiva:  47
Data:  24-APR-20      Codice Regione:  15           Terapia intensiva:  44
Data:  25-APR-20      Codice Regione:  15           Terapia intensiva:  55
Data:  26-APR-20      Codice Regione:  15           Terapia intensiva:  45
Data:  27-APR-20      Codice Regione:  15           Terapia intensiva:  37
Data:  28-APR-20      Codice Regione:  15           Terapia intensiva:  33
Data:  29-APR-20      Codice Regione:  15           Terapia intensiva:  31
Data:  30-APR-20      Codice Regione:  15           Terapia intensiva:  29
```

7.2 PROCEDURA_2

```
--In un arco temporale, calcola la regione con la minor media di guariti e poi calcola
-- la media guariti, media nuovi positivi, media tamponi e media terapia intensiva di
-- tutte le regioni
create OR REPLACE PROCEDURE procedura_2 (DAT1 IN ANDAMENTO_REGIONI.DATA%TYPE ,
    DAT2 IN ANDAMENTO_REGIONI.DATA%TYPE) IS
CURSOR CURSO_1 IS
  SELECT R.Denominazione_Regione , CAST(AVG(A.RICOVERATI_CON_SINTOMI)AS INTEGER) AS A1
,CAST(AVG(A.NUOVI_POSITIVI)AS INTEGER) AS A2,CAST(AVG(A.TOTALE_OSPEDALIZZATI)AS INTEGER) AS
A3,CAST(AVG(A.TERAPIA_INTENSIVA)AS INTEGER) AS A4
  FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
  WHERE data >=DAT1 AND data<=DAT2
  GROUP BY Denominazione_Regione
  ORDER BY AVG(A.DIMESSI_GUARITI) ;
VAR CURSO_1%ROWTYPE;
ERRORE_DATA_INVERTITA EXCEPTION;
ERRORE_DAT1_ASSENTE EXCEPTION;
ERRORE_DAT2_ASSENTE EXCEPTION;
CONT1 INTEGER;
CONT2 INTEGER;
BEGIN
  SELECT COUNT(*) INTO CONT1 FROM ANDAMENTO_REGIONI A JOIN REGIONI R
    on A.codice_regione = R.CODICE_REGIONE WHERE A.DATA=DAT1;
  SELECT COUNT(*) INTO CONT2 FROM ANDAMENTO_REGIONI A JOIN REGIONI R
```

```
    on A.codice_regione = R.CODICE_REGIONE WHERE A.DATA=DAT2;
  IF (CONT1=0) THEN
    RAISE ERRORE_DAT1_ASSENTE;
  ELSE IF (CONT2=0) THEN
    RAISE ERRORE_DAT2_ASSENTE;
  ELSE IF (DAT1>DAT2) THEN
    RAISE ERRORE_DATA_INVERTITA;
  ELSE
    OPEN CURSO_1;
    FETCH CURSO_1 INTO VAR;
    DBMS_OUTPUT.ENABLE();
    DBMS_OUTPUT.PUT_LINE('   ');
    DBMS_OUTPUT.PUT_LINE('Regione con media minore ricoverati con sintomi:' );
    DBMS_OUTPUT.PUT_LINE(RPAD(VAR.Denominazione_Regione,25) || RPAD(VAR.A1,30)
                    ||RPAD(VAR.A2,30) || RPAD(VAR.A3,30)||RPAD(VAR.A4,30));
    DBMS_OUTPUT.PUT_LINE('   ');
    DBMS_OUTPUT.PUT(RPAD('REGIONE   ',25) || RPAD('MEDIA RICOVERATI CON SINTOMI '
      , 30)||RPAD('MEDIA NUOVI POSITIVI ',30) ||
      RPAD('MEDIA TOTALE OSPEDALIZZATI',30)||RPAD('MEDIA TERAPIA INTENSIVA',30));
    DBMS_OUTPUT.PUT_LINE('   ');
  LOOP
    FETCH CURSO_1 INTO VAR;
    EXIT WHEN CURSO_1%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(RPAD(VAR.Denominazione_Regione,25) ||
                RPAD(VAR.A1,30) ||RPAD(VAR.A2,30) ||
                RPAD(VAR.A3,30)||RPAD(VAR.A4,30));
  END LOOP;
  CLOSE CURSO_1;
  end if;
  end if;
  end if;

  EXCEPTION
    WHEN ERRORE_DAT1_ASSENTE THEN DBMS_OUTPUT.PUT_LINE
      ('La prima data selezionata non è presente nel database...');
    WHEN ERRORE_DAT2_ASSENTE THEN DBMS_OUTPUT.PUT_LINE
      ('La seconda data selezionata non è presente nel database...');
    WHEN ERRORE_DATA_INVERTITA THEN DBMS_OUTPUT.PUT_LINE
      ('La prima data è maggiore della seconda. Invertire le date');
END;

call procedura_2('21-APR-2020', '30-APR-2020');
call procedura_2('30-APR-2020', '21-APR-2020');
call procedura_2('21-gen-2020', '30-APR-2020');
call procedura_2('21-APR-2020', '30-mag-2020');
```

RISULTATO:

```
[2020-05-26 12:17:20] Regione con media minore ricoverati con sintomi:
[2020-05-26 12:17:20] Molise                  20                          2                           21                          1

[2020-05-26 12:17:20] REGIONE                 MEDIA RICOVERATI CON SINTOMI  MEDIA NUOVI POSITIVI         MEDIA TOTALE OSPEDALIZZATI    MEDIA TERAPIA INTENSIVA
[2020-05-26 12:17:20] Basilicata              58                          3                           64                          6
[2020-05-26 12:17:20] Calabria                120                         7                           126                         7
[2020-05-26 12:17:20] Sardegna                95                          7                           114                         19
[2020-05-26 12:17:20] Abruzzo                 318                         32                          344                         26
[2020-05-26 12:17:20] Sicilia                 454                         41                          488                         34
[2020-05-26 12:17:20] Puglia                  488                         51                          540                         52
[2020-05-26 12:17:20] Valle d'Aosta           86                          4                           92                          7
[2020-05-26 12:17:20] Umbria                  90                          4                           108                         18
[2020-05-26 12:17:20] Campania                528                         35                          572                         43
[2020-05-26 12:17:20] Lazio                   1420                        80                          1581                        161
[2020-05-26 12:17:20] Friuli Venezia Giulia   131                         25                          146                         15
[2020-05-26 12:17:20] Marche                  675                         42                          736                         62
[2020-05-26 12:17:20] Toscana                 696                         85                          852                         156
[2020-05-26 12:17:20] Liguria                 767                         132                         850                         83
[2020-05-26 12:17:20] Trentino Alto Adige     348                         65                          391                         43
[2020-05-26 12:17:20] Piemonte                2888                        494                         3123                        235
[2020-05-26 12:17:20] Veneto                  1121                        183                         1254                        133
[2020-05-26 12:17:20] Emilia-Romagna          2701                        257                         2950                        249
[2020-05-26 12:17:20] Lombardia               8321                        876                         9043                        722
C##GIULIANO> call procedura_2('30-APR-2020', '21-APR-2020')
[2020-05-26 12:18:17] completed in 5 ms
[2020-05-26 12:18:17] La prima data è maggiore della seconda. Invertire le date
C##GIULIANO> call procedura_2('21-gen-2020', '30-APR-2020')
[2020-05-26 12:18:51] completed in 9 ms
[2020-05-26 12:18:51] La prima data selezionata non è presente nel database...
C##GIULIANO> call procedura_2('21-APR-2020', '30-mag-2020')
[2020-05-26 12:18:55] completed in 11 ms
[2020-05-26 12:18:55] La seconda data selezionata non è presente nel database...
```

7.3 PROCEDURA_3

```
--Densita abitativa delle province
create OR REPLACE PROCEDURE procedura_3 IS
CURSOR CURSO_1 IS SELECT denominazione_provincia, superficie, PROVINCE.residenti
FROM PROVINCE ;
VAR CURSO_1%ROWTYPE;
DENSITA  NUMERIC(10,3);
BEGIN
   OPEN CURSO_1;
   DBMS_OUTPUT.ENABLE();
   DBMS_OUTPUT.PUT_LINE('  ');
   DBMS_OUTPUT.PUT(RPAD('PROVINCIA   ',25) || RPAD('ABITANTI',20)||
            RPAD('SUPERFICIE',20) || RPAD('DENSITA ' ,20) );
   DBMS_OUTPUT.PUT_LINE('   ');
LOOP
   FETCH CURSO_1 INTO VAR;
   EXIT WHEN CURSO_1%NOTFOUND;
   DENSITA:=VAR.residenti/VAR.superficie;
   DBMS_OUTPUT.PUT_LINE( RPAD(VAR.denominazione_provincia ,25) ||
                RPAD(VAR.residenti ,20) || RPAD(VAR.superficie,20) ||
                RPAD(DENSITA,20) );
END LOOP;
CLOSE CURSO_1;
END;
```

RISULTATO:

```
C##GIULIANO> call PROCEDURA_3()
[2020-05-26 12:24:28] completed in 19 ms

[2020-05-26 12:24:28] PROVINCIA              ABITANTI          SUPERFICIE          DENSITA
[2020-05-26 12:24:28] Perugia                657873            6334,09             103,862
[2020-05-26 12:24:28] Venezia                847983            2466,49             343,802
[2020-05-26 12:24:28] Verona                 907352            3120,97             290,728
[2020-05-26 12:24:28] Chieti                 389053            2588,35             150,309
[2020-05-26 12:24:28] Matera                 200012            3446,12             58,04
[2020-05-26 12:24:28] Crotone                171666            1716,58             100,005
[2020-05-26 12:24:28] Vibo Valentia          162252            1139,47             142,393
[2020-05-26 12:24:28] Viterbo                315623            3611,53             87,393
```

7.4 PROCEDURA_4

```
--PERCENTUALE CASI/POPOLAZIONE (tutti i dati tranne nuovi positivi,variazione positivi) FINO A UNA
DETERMINATA GIORNATA
--PER OGNI REGIONE ,TASSO DI MORTALITA' CONTAGIATI (DECEDUTI/TOTALICASI)
CREATE OR REPLACE PROCEDURE procedura_4 (DAT IN ANDAMENTO_REGIONI.DATA%TYPE) IS
CURSOR CURSO_1 IS
   SELECT R. denominazione_regione, R.numero_abitanti, A.totale_casi,
        A.ricoverati_con_sintomi,
        A.terapia_intensiva, A.totale_ospedalizzati, A.dimessi_guariti, A.deceduti
   FROM REGIONI R JOIN ANDAMENTO_REGIONI A ON R.codice_regione = A.codice_regione
   WHERE A.data=DAT;
   VAR CURSO_1%ROWTYPE;
   Perc_casi DECIMAL(4,2);
   Perc_ricoverati_con_sintomi DECIMAL(4,2);
   Perc_terapia_intensiva DECIMAL(4,2);
   Perc_totale_ospedalizzati DECIMAL(4,2);
   Perc_dimessi_guariti DECIMAL(4,2);
   Tasso_mortalita DECIMAL(4,2);
   errore_data exception;
   cont integer;
BEGIN
   SELECT count(*) INTO cont FROM REGIONI R JOIN ANDAMENTO_REGIONI A  ON R.codice_regione =
A.codice_regione WHERE A.data=DAT;
   IF (cont=0) THEN
      RAISE errore_data;
   ELSE
      OPEN CURSO_1;
      DBMS_OUTPUT.ENABLE();
      DBMS_OUTPUT.PUT_LINE('   ');
      DBMS_OUTPUT.PUT(RPAD('REGIONE   ',25) || RPAD('PERCENTUALE CASI TOTALI ', 40)||
                RPAD('PERCENTUALE RICOVERATI CON SINTOMI',40) ||
                RPAD('PERCENTUALE TERAPIA INTENSIVA ',40) ||
                RPAD('PERCENTUALE TOTALE OSPEDALIZZATI',40)||
                RPAD('PERCENTUALE DIMESSI GUARITI',40)||
                RPAD('TASSO DI MORTALITA', 40));
      DBMS_OUTPUT.PUT_LINE('   ');
```

```
    LOOP
      FETCH CURSO_1 INTO VAR;
      EXIT WHEN CURSO_1%NOTFOUND;
      Perc_casi:=(VAR.totale_casi/VAR.numero_abitanti)*100;
      Perc_ricoverati_con_sintomi:=(VAR.ricoverati_con_sintomi/VAR.numero_abitanti)*100;
      Perc_terapia_intensiva:=(VAR.terapia_intensiva/VAR.numero_abitanti)*100;
      Perc_totale_ospedalizzati:=(VAR.totale_ospedalizzati/VAR.numero_abitanti)*100;
      Perc_dimessi_guariti:=(VAR.dimessi_guariti/VAR.numero_abitanti)*100;
      Tasso_mortalita:=(VAR.deceduti/VAR.totale_casi)*100;
      DBMS_OUTPUT.PUT_LINE(RPAD(VAR.Denominazione_Regione,25)  ||
                  RPAD('0'||Perc_casi || '%',40)
                  ||RPAD('0'||Perc_ricoverati_con_sintomi || '%',40) ||
                  RPAD('0'||Perc_terapia_intensiva || '%',40)
                  || RPAD('0'||Perc_totale_ospedalizzati || '%',40)  ||
                  RPAD('0'||Perc_dimessi_guariti || '%',40)
                  || RPAD(Tasso_mortalita || '%',40));
      END LOOP;
  CLOSE CURSO_1;
  end if;
  EXCEPTION
      WHEN errore_data THEN DBMS_OUTPUT.PUT_LINE
        ('La data selezionata non è presente nel database...');
END;

  CALL procedura_4('13-apr-2020');
```

RISULTATO:

```
C##GIULIANO> CALL procedura_4('13-apr-2020')
[2020-05-26 12:27:51] completed in 10 ms

[2020-05-26 12:27:51] REGIONE                 PERCENTUALE CASI TOTALI          PERCENTUALE RICOVERATI CON SINTOMI      PERCENTUALE TERAPIA INTENSIVA
[2020-05-26 12:27:51] Abruzzo                 0,17%                            0,03%                                   00%
[2020-05-26 12:27:51] Basilicata              0,06%                            0,01%                                   00%
[2020-05-26 12:27:51] Calabria                0,05%                            0,01%                                   00%
[2020-05-26 12:27:51] Campania                0,06%                            0,01%                                   00%
[2020-05-26 12:27:51] Emilia-Romagna          0,46%                            0,08%                                   0,01%
[2020-05-26 12:27:51] Friuli Venezia Giulia   0,2%                             0,01%                                   00%
[2020-05-26 12:27:51] Lazio                   0,08%                            0,02%                                   00%
[2020-05-26 12:27:51] Liguria                 0,36%                            0,07%                                   0,01%
[2020-05-26 12:27:51] Lombardia               0,6%                             0,12%                                   0,01%
[2020-05-26 12:27:51] Marche                  0,35%                            0,06%                                   0,01%
[2020-05-26 12:27:51] Molise                  0,08%                            0,01%                                   00%
[2020-05-26 12:27:51] Trentino Alto Adige     0,5%                             0,05%                                   0,01%
[2020-05-26 12:27:51] Piemonte                0,39%                            0,08%                                   0,01%
[2020-05-26 12:27:51] Puglia                  0,08%                            0,01%                                   00%
[2020-05-26 12:27:51] Sardegna                0,07%                            0,01%                                   00%
[2020-05-26 12:27:51] Sicilia                 0,05%                            0,01%                                   00%
[2020-05-26 12:27:51] Toscana                 0,2%                             0,03%                                   0,01%
[2020-05-26 12:27:51] Umbria                  0,15%                            0,01%                                   00%
[2020-05-26 12:27:51] Valle d'Aosta           0,73%                            0,09%                                   0,01%
[2020-05-26 12:27:51] Veneto                  0,29%                            0,03%                                   00%
```

PERCENTUALE TOTALE OSPEDALIZZATI	PERCENTUALE DIMESSI GUARITI	TASSO DI MORTALITA
0,03%	0,02%	10,12%
0,01%	0,01%	5,64%
0,01%	00%	7,22%
0,01%	0,01%	6,76%
0,09%	0,09%	12,79%
0,02%	0,08%	8,14%
0,03%	0,01%	5,72%
0,08%	0,09%	13,58%
0,13%	0,17%	18,07%
0,07%	0,1%	13,25%
0,01%	0,01%	5,84%
0,06%	0,11%	9,71%
0,08%	0,06%	10,66%
0,02%	0,01%	8,71%
0,01%	0,01%	6,65%
0,01%	00%	6,96%
0,03%	0,02%	7,01%
0,02%	0,07%	3,94%
0,1%	0,18%	12,41%
0,03%	0,05%	6,19%

7.5 PROCEDURA_5

```
--Percentuale della somma dei nuovi casi in un arco temporale
create OR REPLACE PROCEDURE procedura_5 (DAT1 IN ANDAMENTO_REGIONI.DATA%TYPE , DAT2 IN
ANDAMENTO_REGIONI.DATA%TYPE ) IS
CURSOR CURSO_1  IS
   SELECT  B1.Denominazione_Regione , B1.A1 AS C1, B2.A5 AS C5
   FROM (SELECT R.Denominazione_Regione , SUM(A.NUOVI_POSITIVI) AS A1
   FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
   WHERE A.DATA>=DAT1 AND A.DATA<=DAT2
   GROUP BY R.Denominazione_Regione
   ORDER BY R.Denominazione_Regione) B1
   JOIN
   (SELECT R.Denominazione_Regione ,R.numero_abitanti AS A5
   FROM REGIONI R
   ORDER BY R.Denominazione_Regione) B2 ON B1.Denominazione_Regione=B2.Denominazione_Regione
;
VAR CURSO_1%ROWTYPE;
PER_A1 NUMBER(6,2);
BEGIN
   OPEN CURSO_1;
   DBMS_OUTPUT.ENABLE();
   DBMS_OUTPUT.PUT_LINE( RPAD('REGIONE',25) || RPAD('PERCENTUALE NUOVI POSITIVI',30) ||
RPAD('SOMMA NUOVI POSITIVI',30)  );
   DBMS_OUTPUT.PUT_LINE('   ');

LOOP
   FETCH CURSO_1 INTO VAR;
   EXIT WHEN CURSO_1%NOTFOUND;
   PER_A1:=VAR.C1/VAR.C5*100;
DBMS_OUTPUT.PUT_LINE( RPAD(VAR.Denominazione_Regione,25) || '0' || RPAD(PER_A1 || '%',30) ||
RPAD(VAR.C1,30) || RPAD('DAL :',4) || RPAD(DAT1,11)||RPAD('AL',4)|| RPAD(DAT2,11)  );
   END LOOP;
CLOSE CURSO_1;
END;

CALL PROCEDURA_5('2-APR-2020','20-APR-2020');
```

RISULTATO:

```
[2020-05-26 12:52:15] REGIONE                 PERCENTUALE NUOVI POSITIVI    SOMMA NUOVI POSITIVI

[2020-05-26 12:52:15] Abruzzo                 0,09%                          1176                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Basilicata              0,02%                          105                           DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Calabria                0,02%                          369                           DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Campania                0,03%                          1843                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Emilia-Romagna          0,18%                          8080                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Friuli Venezia Giulia   0,09%                          1090                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Lazio                   0,04%                          2551                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Liguria                 0,19%                          3009                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Lombardia               0,22%                          22198                         DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Marche                  0,12%                          1864                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Molise                  0,04%                          121                           DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Piemonte                0,26%                          11554                         DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Puglia                  0,04%                          1621                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Sardegna                0,03%                          483                           DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Sicilia                 0,02%                          1041                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Toscana                 0,1%                           3640                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Trentino Alto Adige     0,25%                          2696                          DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Umbria                  0,03%                          254                           DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Valle d'Aosta           0,36%                          457                           DAL 02-APR-20  AL  20-APR-20
[2020-05-26 12:52:15] Veneto                  0,13%                          6502                          DAL 02-APR-20  AL  20-APR-20
```

7.6 PROCEDURA_6

```
--Calcola in numero di casi , deceduti guariti e tamponi in un arco temporale per ogni regione
create OR REPLACE PROCEDURE procedura_6 (DAT1 IN ANDAMENTO_REGIONI.DATA%TYPE , DAT2 IN
ANDAMENTO_REGIONI.DATA%TYPE) IS

CURSOR CURSO_1 IS
   SELECT R.Denominazione_Regione , A.totale_casi , A.deceduti ,A.dimessi_guariti , A.tamponi
   FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
   WHERE data=DAT1
   ORDER BY Denominazione_Regione DESC;
CURSOR CURSO_2 IS
   SELECT R.Denominazione_Regione , A.totale_casi , A.deceduti ,A.dimessi_guariti , A.tamponi
   FROM ANDAMENTO_REGIONI A JOIN REGIONI R on A.codice_regione = R.CODICE_REGIONE
   WHERE data=DAT2
   ORDER BY Denominazione_Regione DESC;
VAR1 CURSO_1%ROWTYPE;
VAR2 CURSO_2%ROWTYPE;
A1 INTEGER;
A2 INTEGER;
A3 INTEGER;
A4 INTEGER;
BEGIN
   OPEN CURSO_1;
   OPEN CURSO_2;
   DBMS_OUTPUT.ENABLE();
   DBMS_OUTPUT.PUT_LINE('   ');
   DBMS_OUTPUT.PUT(RPAD('REGIONE   ',25) || RPAD('NUMERO CASI ', 20)|| RPAD('NUMERO DECEDUTI
', 20)|| RPAD('NUMERO GUARITI ', 20)|| RPAD('NUMERO TAMPONI ', 20));
   DBMS_OUTPUT.PUT_LINE('   ');
LOOP
   FETCH CURSO_1 INTO VAR1;
   FETCH CURSO_2 INTO VAR2;
   A1:=VAR2.totale_casi-VAR1.totale_casi;
   A2:=VAR2.deceduti-VAR1.deceduti;
   A3:=VAR2.dimessi_guariti-VAR1.dimessi_guariti;
   A4:=VAR2.tamponi-VAR1.tamponi;

   EXIT WHEN CURSO_1%NOTFOUND ;
   DBMS_OUTPUT.PUT_LINE(RPAD(VAR1.Denominazione_Regione,25) || RPAD(A1,20) || RPAD(A2,20)||
RPAD(A3,20) || RPAD(A4,20)|| RPAD('DAL :',4) || RPAD(DAT1,11)||RPAD('AL',4)|| RPAD(DAT2,11));
END LOOP;
CLOSE CURSO_2;
CLOSE CURSO_1;
END;
   CALL PROCEDURA_6('1-APR-2020' ,'30-APR-2020' );
```

RISULTATO:

```
[2020-05-26 12:53:25] REGIONE                NUMERO CASI        NUMERO DECEDUTI    NUMERO GUARITI     NUMERO TAMPONI
[2020-05-26 12:53:25] Veneto                 8335               960                7452               236481             DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Valle d'Aosta          497                78                 870                5914               DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Umbria                 297                30                 898                27379              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Trentino Alto Adige    3346               404                3365               58205              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Toscana                4485               589                2744               105274             DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Sicilia                1448               147                688                62833              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Sardegna               550                82                 399                19253              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Puglia                 2126               286                647                47251              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Piemonte               16494              2180               7291               126474             DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Molise                 138                11                 68                 5289               DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Marche                 2285               429                2102               47192              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Lombardia              30959              6179               14334              255494             DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Liguria                4333               707                2720               37428              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Lazio                  3352               272                1370               101786             DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Friuli Venezia Giulia  1340               167                1209               51943              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Emilia-Romagna         10649              1819               10756              124400             DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Campania               2192               211                1184               60380              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Calabria               439                48                 261                25920              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Basilicata             130                16                 147                10512              DAL 01-APR-20  AL  30-APR-20
[2020-05-26 12:53:25] Abruzzo                1494               197                593                28386              DAL 01-APR-20  AL  30-APR-20
```

7.7 TRIGGER

I trigger sono gli strumenti atti ad implementare automatismi nella basi di dati, ad esempio, nel nostro caso, quando viene fatto un inserimento nella tabella MASTER, allora il trigger controlla se ogni valora della nuova tupla rispetti i vincoli di dominio e, se ciò accade, inserisce opportunamente i dati nelle tabelle.

```
CREATE OR REPLACE trigger TRIGGER1  before insert on MASTER  for each row
declare
   errore exception ;
   cont integer;
BEGIN
select count(*) into cont
from MASTER
   where :new.codice_provincia=CODICE_PROVINCIA and
denominazione_provincia=:new.denominazione_provincia and
       sigla_provincia=:new.sigla_provincia and stato='ITA'AND CODICE_REGIONE=:new.CODICE_REGIONE
AND
       DENOMINAZIONE_REGIONE=:new.DENOMINAZIONE_REGIONE AND
LONGITUDINE=:new.LONGITUDINE AND LATITUDINE=:new.LATITUDINE;

IF (CONT=0) THEN
   RAISE ERRORE;
ELSE
INSERT INTO ANDAMENTO_PROVINCE(DATA, CODICE_PROVINCIA, TOTALE_CASI, NOTE_IT, NOTE_EN)
VALUES (:new.DATA, :new.CODICE_PROVINCIA, :new.TOTALE_CASI, :new.NOTE_IT, :new.NOTE_EN);
END IF;
   EXCEPTION
      WHEN ERRORE THEN DBMS_OUTPUT.PUT_LINE ('CONTROLLARE CHE A OGNI OGNI TUPLA E STATO
INSERITO IL VALORE GIUSTO');
      raise_application_error(-20001,'CORRISPONDENZA DEI DATI INSERITI NON VALIDA');
   END;
```

Esempio inserimento corretto:

```
INSERT INTO
MASTER(data,stato,codice_regione,denominazione_regione,codice_provincia,denominazione_provincia,sigla_
provincia,latitudine, longitudine,totale_casi,note_it,note_en)
VALUES ('25-APR-2033','ITA',13,'Abruzzo',069,'Chieti','CH',42.35103167,14.16754574,0,NULL,NULL);
```

La transazione è stato conclusa con successo

```
[2020-05-27 16:10:46] 1 row affected in 24 ms
[2020-05-27 16:11:02] transaction committed: @console [TesinaCovid]
```

Controlliamo se sono stati inseriti nelle relative tabelle:

Esempio inserimento non corretto:

```
INSERT INTO
MASTER(data,stato,codice_regione,denominazione_regione,codice_provincia,denominazione_provincia,sigla_
provincia,latitudine, longitudine,totale_casi,note_it,note_en)
VALUES ('25-APR-2033','ITA',13,'Abruzzo',069,'Chieti','CT',42.35103167,14.16754574,0,NULL,NULL);
```

La transazione ha dato il seguente errore che abbiamo definito nel trigger:

```
[2020-05-27 16:13:57] [72000][20001] ORA-20001: CORRISPONDENZA DEI DATI INSERITI NON VALIDA
[2020-05-27 16:13:57] ORA-06512: a "C##GIULIANO.TRIGGER1", line 18
[2020-05-27 16:13:57] ORA-04088: errore durante esecuzione del trigger 'C##GIULIANO.TRIGGER1'
[2020-05-27 16:13:57] Position: 12
[2020-05-27 16:13:57] CONTROLLARE CHE A OGNI OGNI ATTRIBUTO E STATO INSERITO IL VALORE CORRISPONDENTE DELLA PROVINCIA
```

8 VISTE

Alcune possibili viste per migliorare i tempi dell'esecuzione delle query sono le seguenti:

8.1 ANDAMENTO_NAZIONALE

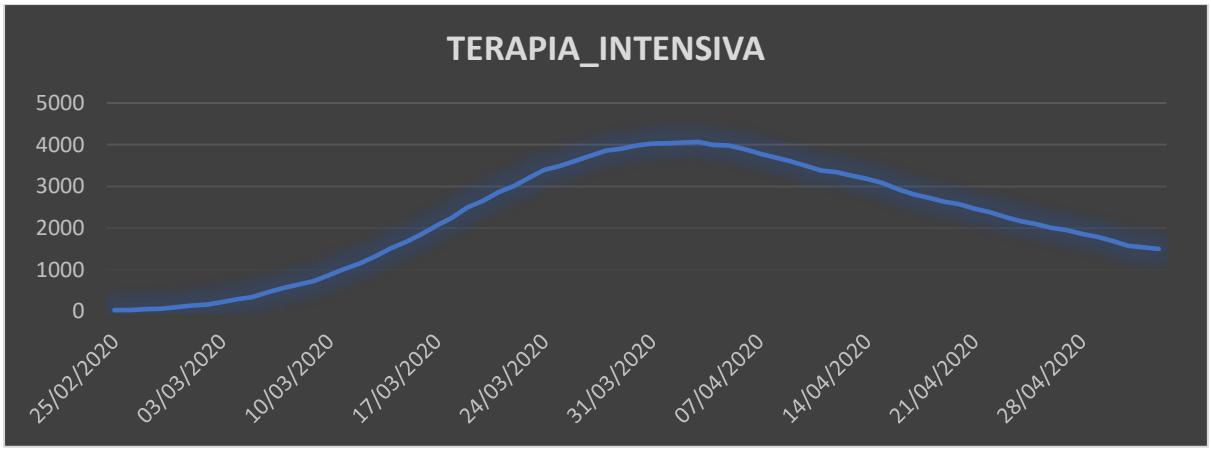
```
create MATERIALIZED VIEW ANDAMENTO_NAZIONALE  as
select data ,SUM(ricoverati_con_sintomi) AS RICOVERATI_CON_SINTOMI, SUM(terapia_intensiva) AS TERAPIA_INTENSIVA,
       SUM(totale_ospedalizzati) AS TOTALE_OSPEDALIZZATI, SUM(isolamento_domiciliare) AS ISOLAMENTO_DOMICILIARE,
       SUM(totale_positivi) AS TOTALE_POSITIVI, SUM(variazione_totale_positivi) AS VARIAZIONE_TOTALE_POSITIVI,
       SUM(nuovi_positivi) AS NUOVI_POSITIVI,  SUM(dimessi_guariti) AS DIMESSI_GUARITI, SUM(deceduti) AS DECEDUTI,
       SUM(totale_casi) AS TOTALE_CASI, SUM(tamponi) AS TAMPONI, SUM(casi_testati) AS CASI_TESTATI
from ANDAMENTO_REGIONI
group by data
ORDER BY data;
```

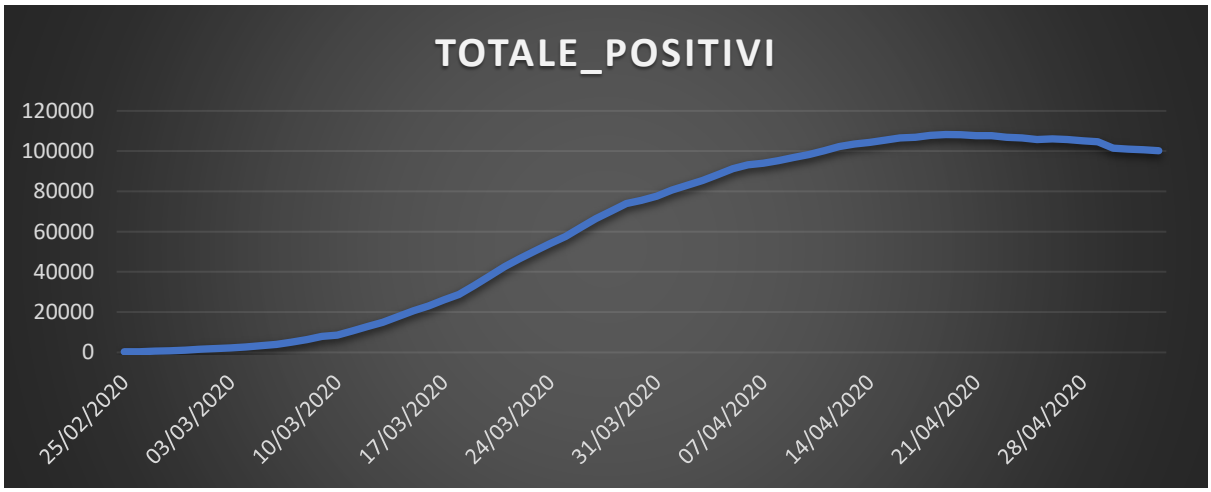
La vista ANDAMENTO_NAZIONALE somma tutti gli attributi della tabella ANDAMENTO_REGIONI per ogni data disponibile. Questa view è utile per ottimizzare i tempi di esecuzione delle query che richiedono i dati riguardanti tutta la nazione.

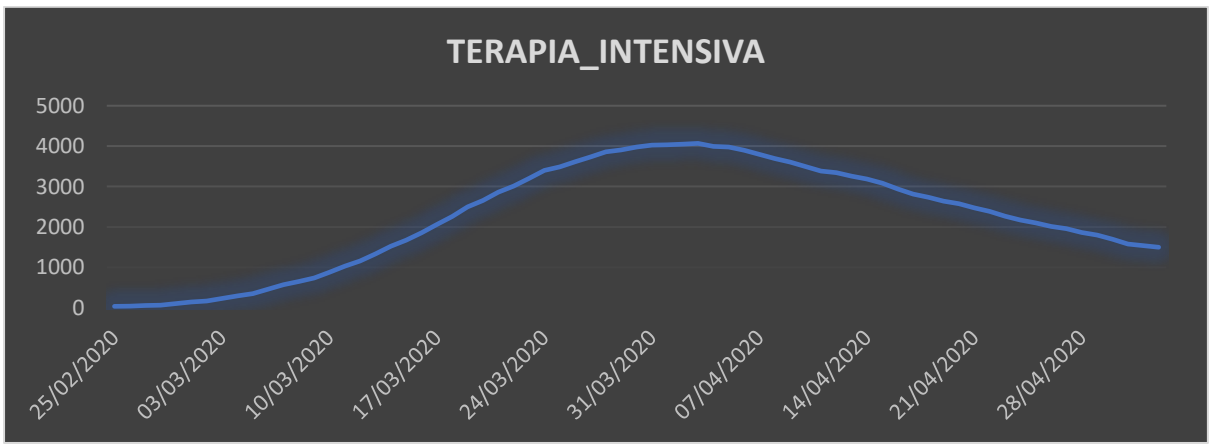
8.2 GRAFICI ANDAMENTO_NAZIONALE

Di seguito sono riportati i grafici relativi all'andamento del contagio in tutta l'Italia dal 25/02/2020 al 03/05/2020.


RICOVERATI_CON_SINTOMI
35000
30000
25000
20000
15000
10000
5000
0
25/02/2020
03/03/2020
10/03/2020
17/03/2020
24/03/2020
31/03/2020
07/04/2020
14/04/2020
21/04/2020
28/04/2020


TERAPIA_INTENSIVA
5000
4000
3000
2000
1000
0
25/02/2020
03/03/2020
10/03/2020
17/03/2020
24/03/2020
31/03/2020
07/04/2020
14/04/2020
21/04/2020
28/04/2020


TOTALE_POSITIVI
120000
100000
80000
60000
40000
20000
0
25/02/2020
03/03/2020
10/03/2020
17/03/2020
24/03/2020
31/03/2020
07/04/2020
14/04/2020
21/04/2020
28/04/2020


TERAPIA_INTENSIVA
5000
4000
3000
2000
1000
0
25/02/2020
03/03/2020
10/03/2020
17/03/2020
24/03/2020
31/03/2020
07/04/2020
14/04/2020
21/04/2020
28/04/2020

8.3 VIEW1

```
--VIEW PER QUERY 1
create view VIEW1 as
    select A.data, P.denominazione_provincia, A.totale_casi
    from ANDAMENTO_PROVINCE A join PROVINCE P on A.codice_provincia = P.codice_provincia;
--LA QUERY1 DIVENTA:
select * from VIEW1 where data='10-apr-2020';
```

La VIEW1 sostituisce l'attributo *codice_provincia* nella tabella ANDAMENTO_PROVINCE con l'attributo *denominazione_provincia*, così da permettere un'ottimizzazione delle query che richiedono il numero di casi totali data la *denominazione_provincia*.

8.4 VIEW6

```
--VIEW PER QUERY 6
create view VIEW6 as
    select A.data, R.denominazione_regione, A.totale_casi, A.deceduti, A.dimessi_guariti, A.tamponi
    from ANDAMENTO_REGIONI A join REGIONI R on A.codice_regione = R.codice_regione
    order by A.tamponi desc;
--LA QUERY6 DIVENTA
select * from VIEW6 where data ='20-apr-2020';
```

La VIEW6 sostituisce l'attributo *codice_regione* nella tabella ANDAMENTO_REGIONE con l'attributo *denominazione_regione* ed è composta solo da alcuni degli attributi presenti in ANDAMENTO_REGIONE. Così da permettere un'ottimizzazione delle query e procedure che richiedono il esclusivamente quegli attributi data la *denominazione_regione*.