Tesina Basi Di Dati

Pasquale del Moro, Dario Di Meo

**Indice**

1. Introduzione
2. Normalizzazione Master Table Province
3. Creazione Basi Di Dati e comandi DDL
4. Arrichimento base dati, alter table
5. Modello E/R
6. Analisi dei dati
   1. Query
   2. Grafici
7. Stored Procedure
8. Ottimizzazione Base di dati

**Appendice**

1. Creazione Master table
2. Fonti e siti web utili
3. **INTRODUZIONE**

La tesina consiste in un progetto di basi di dati, relativo al covid-19, nel contesto dello Stato Italiano e in un arco temporale che va dal 24/02/2020 al 03/05/2020.

Tale progetto parte da dati relativi alle regioni e alle province italiane forniti dalla protezione Civile Italiana, e ha come idea generale quella di creare una base di dati attiva con la quale poter poi interagire per analizzare dati ed eventualmente costruire grafici ed infine visualizzarli.

Dalla precedente descrizione è quindi possibile individuare i punti in cui sequenzialmente si è sviluppato il progetto:

* Conversione dei dati forniti dalla protezione civile in tabelle del **modello relazionale**;
* **Normalizzazione** di tali tabelle al fine di ricavare lo schema principale della base dati;
* Aggiunta di ulteriori dati prelevati da altre fonti;
* Estrazione di un possibile **modello ER** corrispondente allo schema di basi di dati ottenuto;
* Scelta di un **insieme di query** da effettuare sul database per la selezione e l’estrazione di dati significativi;
* Realizzazione di **trigger e stored procedure** che permettono rispettivamente di implementare politiche di reazione e parametrizzare query;
* Creazione di **grafici** sulla base dei risultati delle query e relativi alle grandezze significative della base di dati;
* Creazione di uno strumento di visualizzazione grafica, nello specifico un **istogramma animato**, realizzato nel linguaggio Python e con la libreria grafica pygame;
* Definizione di indici e viste al fine di migliorare le prestazioni di alcune query ricorrenti.

Da tale elenco si deduce la necessità di normalizzare le tabelle estratte dai dati della protezione civile, per poter poi lavorare agilmente sulla **base di dati** e renderla **attiva** con trigger e stored produre.

Alla creazione della base di dati e alla loro analisi, si è aggiunta anche una parte del progetto dedicata alla definizione, tramite un processo di **Reverse Engineering**, dello **schema concettuale** E**/**R, per rappresentare la realtà di nostro interesse.

La realizzazione fisica della base di dati è stata effettuata mediante il DBMS locale Oracle 18c XE insieme all’ambiente di sviluppo SQL-DEVELOPER.

1. **Normalizzazione Master Table Province**

Estratta la tabella dal file **Csv** riguardante il **covid-19** e relativa alle **province italiane** dal **24/02/2020** al **03/05/2020**, si vuole rendere in 3a forma normale tale tabella.

Lo **schema della tabella estratta** è il seguente:

**MASTER\_TABLE\_PROVINCE**

**( data**, stato, codice\_regione, denominazione\_regione, **codice­\_provincia**, denominazione\_provincia, sigla\_provincia, latitudine, longitudine, totale\_casi, note\_in, note\_en **)**

dove la **chiave primaria** è costituita dalla coppia di attributi *(****data, codice\_provincia****)*;

inoltre i tipi corrispondenti ai campi sono stringhe di testo o numerici, tranne per la data, che è di tipo DATE.

**PRIMA FORMA NORMALE**

La tabella estratta risulta essere in 1a forma normale in quanto banalmente il suo schema contiene solo attributi semplici, quindi né strutturati né multi-valore.

**INDIVIDUAZIONE DELE DIPENDENZE FUNZIONALI NON BANALI**

**{ codice\_provincia, data }  {** tutti gli altri attributi **}**

**{ codice\_provincia } ** stato, codice\_regione, denominazione\_regione, denominazione\_provincia, sigla\_provincia, latitudine, longitudine

**{ codice\_regione } ** stato, denominazione\_regione

**SECONDA FORMA NORMALE**

Ricerca delle eventuali dipendenze funzionali non complete dalla chiave agli attributi non primi  ricerca di dipendenze funzionali da una sotto-parte della chiave ad un altro attributo.

Attributi primi che dipendono esclusivamente da:

***codice\_provincia***  *denominazione\_provincia*, *sigla\_provincia*, *latitudine,* *longitudine*; *codice\_regione*,*denominazione\_regione, stato*.

**lo schema di relazione non risulta essere in 2a normale**.

Schemi in 2a forma normale:

**MISURE\_PROVINCE**( **data**, **codice­\_provincia**:DATI\_PROVINCE, totale\_casi, note\_in, note\_en )

**PROVINCE**( stato, codice\_regione, denominazione\_regione, **codice­\_provincia**, denominazione\_provincia, sigla\_provincia, latitudine, longitudine )

**TERZA FORMA NORMALE**

Eliminazione delle eventuali dipendenze funzionali transitive tra chiave ed attributi non primi  individuazione di tutti gli attributi non primi che dipendono da altri attributi non primi.

* **Tabella dei contagi MISURE\_PROVINCE**

Per tale schema si verifica che fra i tre attributi non primi *totale\_casi*, *note\_in*, *note\_en* non vi sono dipendenze funzionali, quindi non sussistono dipendenze transitive dalla chiave, il che implica che **la 3a forma normale è già verificata**.

* **Tabella dei dati relativi alle province *PROVINCE***

Il codice di una regione la identifica univocamente, tutti gli attributi relativi a informazioni della regione sono in dipendenza **solo** da *codice\_regione*  gli attributi *stato* e *denominazione\_regione* dipendono dall’attributo *codice\_regione*  si ha una dipendenza transitiva da *codice\_provincia* a *(stato, denominazione\_regione)*

**lo schema di relazione non risulta essere in 3a normale**.

Schema in 3a forma normale:

**PROVINCE**( codice\_regione:REGIONI, **codice­\_provincia**, denominazione\_provincia, sigla\_provincia, latitudine, longitudine ).

**REGIONI**( stato, **codice\_regione**, denominazione\_regione )

**RISULTATO DELLA NORMALIZZAZIONE**

Riordinando gli attributi in maniera più opportuna, si hanno i seguenti schemi:

**MISURE\_PROVINCE**

( **data**, **codice­\_provincia**:PROVINCE, totale\_casi, note\_in, note\_en )

**PROVINCE**

( **codice­\_provincia**, denominazione\_provincia, sigla\_provincia, codice\_regione: REGIONI, latitudine, longitudine )

**REGIONI**

( **codice\_regione**, denominazione\_regione, stato )

1. **Creazione Basi Di Dati e comandi DDL**
2. **Arrichimento base dati, alter table**
3. **Modello E/R**

PARCHI

LAVORO

FARMACIE

VARIAZIONE\_SPOSTAMENTO

SVAGO

DATA

ETC…

(1,1)

**LUOGO\_**

**MISURAZIONE**

**3**

**MISURA MOVIMENTI**

CODICE\_REGIONE

DENOMINAZIONE\_PROVINCIA

CODICE\_PROVINCIA

DENOMINAZIONE\_REGIONE

CODICE\_REGIONE

(0,N)

ETC…

TAMPONI

DECEDUTI

TERAPIA\_INTENSIVA

TOTALE\_CASI

(0,N)

DATA

LONGITUDINE

LATITUDINE

POSIZIONE

(0,N)

**LUOGO\_**

**MISURAZIONE**

**2**

CODICE\_PROVINCIA

CODICE\_REGIONE

(1,1)

EN

ITA

NOTE

(1,1)

**SITO**

**MISURA PROVINCIA**

**MISURA REGIONE**

**MISURA COVID**

**REGIONE**

**PROVINCIA**

**LUOGO\_**

**MISURAZIONE**

**1**

(1,1)

(1,N)

ETC…

ETC…

DATA

1. **Analisi dei dati**

L’analisi dei dati, si avvalora di Query, procedure e ad alcuni tool Grafici, per mostrare i dati e le relazioni tra essi.

* **Query e Grafici sulle variazioni di Spostamenti Regionali:**

**--Query parametrizzata per il report in formato CSV sui movimenti di una regione presa in INPUT**

**CREATE OR REPLACE PROCEDURE** report\_movimenti\_regione (regione\_in **IN** regioni.denominazione\_regione%TYPE)

**AS**

found\_regione **INTEGER**;

**CURSOR** cursore\_regione **IS** (

**SELECT** mov.codice\_regione, mov.data, mov.variazione\_svago\_vendita\_dettaglio, mov.variazione\_farmacie\_alimentari, mov.variazione\_parchi, mov.variazione\_stazioni\_trasporto\_pubblico, mov.variazione\_sedi\_lavoro, mov.variazione\_zone\_residenziali

**FROM** movimenti\_regioni mov **JOIN** regioni r **ON** mov.codice\_regione = r.codice\_regione

**WHERE** **LOWER**(denominazione\_regione) = **LOWER**(regione\_in)

);

riga\_movimenti movimenti\_regioni%**ROWTYPE**;

regione\_not\_found **EXCEPTION**;

**BEGIN**

**--Controllo che la regione sia presente**

**SELECT** count(distinct(codice\_regione)) **INTO** found\_regione

**FROM** regioni

**WHERE LOWER**(denominazione\_regione) = **LOWER**(regione\_in) **AND** codice\_regione **IN** (

**SELECT** distinct(codice\_regione)

**FROM** movimenti\_regioni

);

**IF** found\_regione <= 0 **THEN**

**RAISE** regione\_not\_found;

**END IF;**

**--posso stampare le regioni**

**OPEN** cursore\_regione;

**dbms\_output.put\_line**('DATI MOVIMENTI REGIONE '|| regione\_in);

**dbms\_output.put\_line**('codice\_regione,data,variazione\_svago\_vendita\_dettaglio,variazione\_farmacie\_alimentari,variazione\_parchi, variazione\_stazioni\_trasporto\_pubblico,variazione\_sedi\_lavoro,variazione\_zone\_residenziali');

**LOOP**

**FETCH** cursore\_regione **INTO** riga\_movimenti;

**EXIT WHEN** cursore\_regione**%NOTFOUND**;

**dbms\_output.put\_line**(riga\_movimenti.codice\_regione || ',' || riga\_movimenti.data || ',' || riga\_movimenti.variazione\_svago\_vendita\_dettaglio || ',' || riga\_movimenti.variazione\_farmacie\_alimentari|| ',' || riga\_movimenti.variazione\_parchi|| ',' || riga\_movimenti.variazione\_stazioni\_trasporto\_pubblico|| ',' || riga\_movimenti.variazione\_sedi\_lavoro|| ',' || riga\_movimenti.variazione\_zone\_residenziali );

**END LOOP;**

**CLOSE** cursore\_regione;

**EXCEPTION**

**WHEN** regione\_not\_found **THEN dbms\_output.put\_line**('REGIONE NON PRESENTE');

**END** report\_movimenti\_regione;

* **MEDIA MOVIMENTI REGIONALI**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

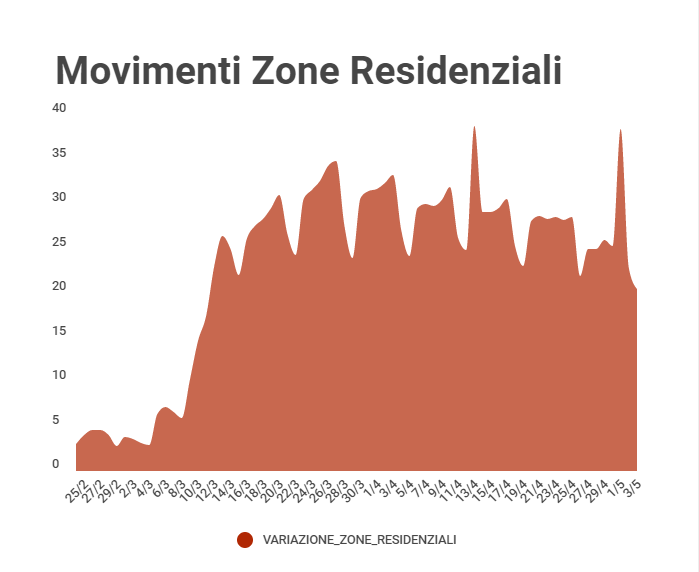
**--Media degli spostamenti degli italiani, verso una determinata categoria di luoghi**

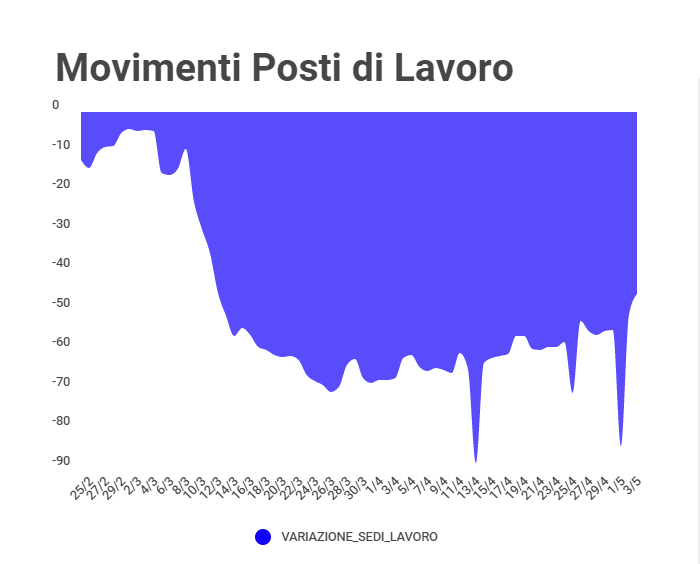
**SELECT** data**, AVG**(variazione\_\*\*\*\*)

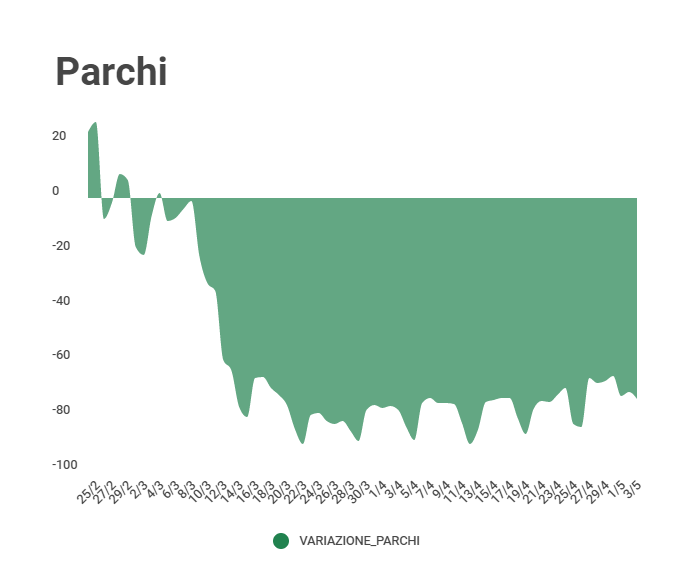
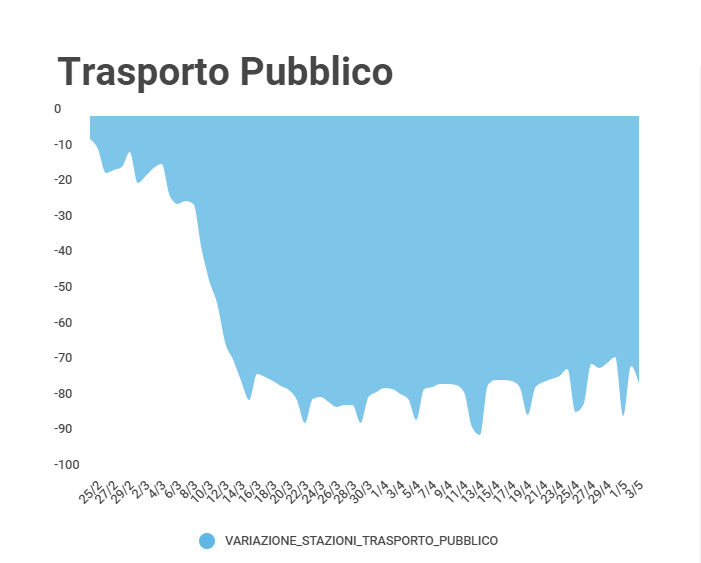
**FROM** movimenti\_regioni

**GROUP BY** data

**ORDER BY** data;

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**





* **Query che ritorna il numero di contagi e la posizione per ogni provincia, alla data più recente**

**SELECT** p.longitudine, p.latitudine, mis.totale\_casi

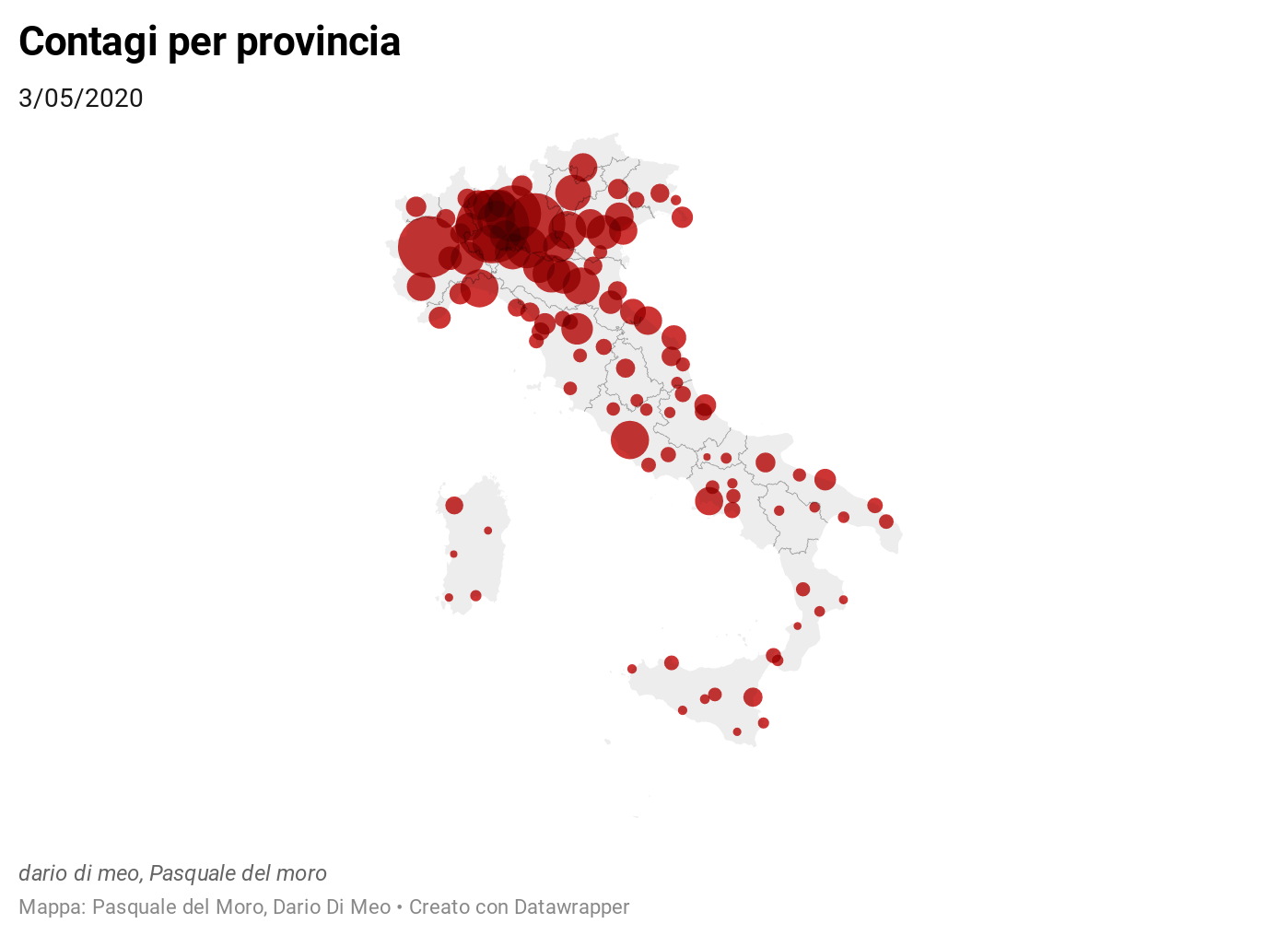
**FROM** misure\_province mis **JOIN** province p **ON** mis.codice\_provincia = p.codice\_provincia

**WHERE** data = (

**SELECT** max(data)

**FROM** misure\_province

);



1. Stored Procedure

Oltre alle Query parametrizzate usate nel paragrafo dedicato all’analisi dei dati, si è ritenuto creare alcune Procedure per ricavare dati utili.

**Procedure e Funzioni**

-----------------------------------------------------------------------------------------------------------------

**--FUNCTION CHE RITORNA IL CODICE DELLA REGIONE COL MASSIMO DEI CONTAGI DATO UNA DATA**

**CREATE OR REPLACE FUNCTION** get\_regione\_max\_contagi(data\_oss I**N** misure\_regioni.data%**TYPE**)

**RETURN** misure\_regioni.codice\_regione%**TYPE**

**AS**

cod\_max misure\_regioni.codice\_regione%**TYPE**;

**BEGIN**

**SELECT** codice\_regione **INTO** cod\_max

**FROM** misure\_regioni

**WHERE** data **=** data\_oss **AND** totale\_positivi **IN** (

**SELECT** **max(**totale\_positivi**)**

**FROM** misure\_regioni

**WHERE** data **=** data\_oss

);

**RETURN** cod\_max;

**END** get\_regione\_max\_contagi;

**--FUNCTION CHE RITORNA IL CODICE DELLA PROVINCIA COL MASSIMO DEI CONTAGI DATO UNA DATA**

**CREATE OR REPLACE FUNCTION** get\_provincia\_max\_contagi(data\_oss **IN** misure\_province.data%**TYPE**)

**RETURN** misure\_province.codice\_provincia%**TYPE**

**AS**

cod\_max misure\_province.codice\_provincia%**TYPE**;

**BEGIN**

**SELECT** codice\_provincia **INTO** cod\_max

**FROM** misure\_province

**WHERE** data **=** data\_oss **AND** totale\_casi **IN** (

**SELECT** **max(**totale\_casi**)**

**FROM** misure\_province

**WHERE** data **=** data\_oss

);

**RETURN** cod\_max;

**END** get\_provincia\_max\_contagi;

-----------------------------------------------------------------------------------------------------------------

**Trigger**

-----------------------------------------------------------------------------------------------------------------

**--TRIGGER CHE RIEMPE AUTOMATICAMENTE LA TABELLA NORMALIZZATA DAGLI INSERT SU LA**

**--MASTER TABLE PROVINCE**

**CREATE OR REPLACE TRIGGER** trigger\_insert\_master\_province

**AFTER INSERT**

**ON** master\_table\_province

**FOR EACH ROW**

**BEGIN**

**INSERT INTO** misure\_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;**

**--TRIGGER CHE RIEMPE AUTOMATICAMENTE LA TABELLE NORMALIZZATE DAGLI INSERT SU LA**

**--MASTER TABLE REGIONI**

**CREATE OR REPLACE TRIGGER** trigger\_insert\_master\_regioni

**AFTER INSERT**

**ON** master\_table\_regioni

**FOR EACH ROW**

**BEGIN**

**INSERT INTO** misure\_regioni(data,

codice\_regione,

ricoverati\_con\_sintomi,

terapia\_intensiva,

totale\_ospedalizzati,

isolamento\_domiciliare,

totale\_positivi,

variazione\_totale\_positivi,

nuovi\_positivi,

dimessi\_guariti,

deceduti,

totale\_casi,

tamponi,

casi\_testati ,

note\_it ,

note\_en )

**VALUES** (**:NEW**.data,

**:NEW**.codice\_regione,

**:NEW**.ricoverati\_con\_sintomi,

**:NEW**.terapia\_intensiva,

**:NEW**.totale\_ospedalizzati,

**:NEW**.isolamento\_domiciliare,

**:NEW**.totale\_positivi,

**:NEW**.variazione\_totale\_positivi,

**:NEW**.nuovi\_positivi,

**:NEW**.dimessi\_guariti,

**:NEW**.deceduti,

**:NEW**.totale\_casi,

**:NEW**.tamponi,

**:NEW**.casi\_testati ,

**:NEW**.note\_it ,

**:NEW**.note\_en);

**END;**

-----------------------------------------------------------------------------------------------------------------

1. **Ottimizzazione Base di dati**

L’ottimizzazione della base di dati, consiste nella creazione di **Indici** e **Viste**, per migliorare i tempi di esecuzione delle query.

Tramite le funzionalità di Cronologia SQL, su Oracle Live Sql **(**F8**)**, si può calcolare, in secondi, la durata di una query.

**Ad esempio:**

|  |
| --- |
| DURATA |
| 0.11s |

**SELECT** \*

**FROM** misure\_regioni 🡺

**WHERE** data = '3-mag-20';

**Indici**

Sono creati automaticamente sui campi chiave delle nostre tabelle. Generalmente le query sono basate su ricerche sui campi di tipo **DENOMINAZIONE**, cioè sui nomi delle regioni/province.

Per questo motivo si intorducono due **INDICI SECONDARI** sui campi non chiave, denominazione.

---------------------------------------------------------------------------------------------------------

**--Indice su denominazione regione**

**CREATE INDEX** idx\_denominazione\_regione

**ON** **regioni**(denominazione\_regione);

**--Indice su denominazione Provincia**

**CREATE INDEX** idx\_denominazione\_provincia

**ON province(**denominazione\_provincia**);**

---------------------------------------------------------------------------------------------------------

|  |  |  |
| --- | --- | --- |
| Tabella | Prima dell’indice | Dopo l’indice |
| Regioni | 0.005s | 0.002s |
| Province | 0.005s | 0.002s |

Grazie a questi due indici si nota un notevole decremento del tempo di esecuzione delle query, quantificato dallo strumento di analisi di Oracle Live Sql.

Valori ottenuti dalle seguenti Query di **Testing**:

**SELECT** \* **SELECT** \*

**FROM** regioni **FROM** province

**WHERE** denominazione\_regione = 'Campania'; **WHERE** denominazione\_provincia = 'Napoli';

**Note:**

Il calcolo del tempo di esecuzione è **Machine dipendent**, quindi è solo una stima del tempo necessario. Inoltre i tempi calcolati sono frutto di una **media aritmetica** su più prove ripetute.

**Viste**

Le viste (non materializzate), sono Query memorizzate che quando vengono chiamate vengono automaticamente eseguite per costruire una tabella temporanea su cui effettuare operazioni SQL.

Le viste vengono principalmente create per:

* Ridurre la complessità delle istruzioni sql (snellire il codice)
* Mostrare solo specifici attributi all’ utente utilizzatore della vista

Per questi motivi, si cercano **Query** frequentemente utilizzate sia per l’analisi dei dati, sia per gli eventuali utenti disposti ad utilizzare la base dati come **supporto** per una loro applicazione.

Si è pensato di creare una vista per la visualizzazione, alla **data più recente**, dei contagi per ogni **provincia**, alla cui è stata aggiunta la colonna *denominazione\_provincia* per facilitare eventuali ricerche sul campo.

**--Vista dati più recenti sui contagi per provincia**

**CREATE VIEW** last\_contagi\_province **AS**(

**SELECT** mis.data, mis.codice\_provincia, p.denominazione\_provincia, mis.totale\_casi

**FROM** misure\_province mis **JOIN** province p **ON** mis.codice\_provincia = p.codice\_provincia

**WHERE** mis.data **IN**(

**SELECT** max(data)

**FROM** misure\_province

)

);

**Viste Materializzate**

Le viste materializzate, a differenza delle viste classiche, sono uno **Snapshot** di una particolare Query. Quindi utilizzate per dati statici che mutano **molto raramente**, inoltre, sono molto più **efficienti** delle viste normali, in quanto non devono essere ricreate ad ogni utilizzo.

Una vista utile, è una sui dati riguardanti i posti letto presenti per ogni regione e il numero di positivi di ogni giorno.

**--Vista materializzata sui dati regionali mettendo in relazione i posti letto e i contagi**

**CREATE MATERIALIZED VIEW** posti\_regione\_positivi **AS**(

**SELECT** mis.data, r.codice\_regione, r.posti\_letto, r.posti\_terapia\_intensiva, mis.totale\_positivi

**FROM** misure\_regioni mis **JOIN** regioni r **ON** mis.codice\_regione = r.codice\_regione

);

**Appendice**

1. **Creazione Master Table**

Le Master Table sono tabelle non normalizzate il cui unico scopo è di essere un supporto iniziale, per la memorizzazione diretta dei dati forniti dalla Protezione Civile Italiana sul Covid.Quindi rappresentano i dati grezzi fornitici, su cui grazie alle forme normali, e a l’aggiunta di attributi si ottiene la Base Di Dati finale, cioè l’obbiettivo finale del progetto.

**CODICE SQL CREATE MASTER TABLE:**

**CREATE TABLE** master\_table\_province

(

data **DATE**,

stato **CHAR(3),**

codice\_regione **NUMBER(2),**

denominazione\_regione **VARCHAR2(200),**

codice\_provincia **NUMBER(3) NOT NULL**,

denominazione\_provincia **VARCHAR2(200),**

sigla\_provincia **CHAR(2),**

latitudine **NUMBER,**

longitudine **NUMBER,**

totale\_casi **INTEGER,**

note\_it **CLOB,**

note\_en **CLOB**

);

**ALTER TABLE** master\_table\_province **ADD CONSTRAINT** PK\_MASTER\_PROVINCE **PRIMARY KEY** (data, codice\_provincia);

**CREATE TABLE** master\_table\_regioni

( data **DATE**,

stato **CHAR(3),**

codice\_regione **NUMBER(2),**

denominazione\_regione **VARCHAR2(200),**

latitudine **NUMBER,**

longitudine **NUMBER,**

ricoverati\_con\_sintomi **INTEGER,**

terapia\_intensiva **INTEGER,**

totale\_ospedalizzati **INTEGER,**

isolamento\_domiciliare **INTEGER,**

totale\_positivi **INTEGER,**

variazione\_totale\_positivi **INTEGER,**

nuovi\_positivi **INTEGER,**

dimessi\_guariti **INTEGER**,

deceduti **INTEGER,**

totale\_casi **INTEGER,**

tamponi **INTEGER,**

casi\_testati **INTEGER,**

note\_it **CLOB,**

note\_en **CLOB**

);

**ALTER TABLE** master\_table\_regioni **ADD CONSTRAINT** PK\_MASTER\_REGIONI **PRIMARY KEY** (data, codice\_regione);

1. **Fonti e siti web utili**

**Dati:**

Dati Protezione Civile: <https://github.com/pcm-dpc/COVID-19>

Posti letto e posti terapia intensiva: <http://www.healthdata.org/covid/data-downloads>

Numero di abitanti, Densità abitativa: <http://dati.istat.it/>

Variazione spostamenti regionali: <https://www.google.com/covid19/mobility/>

Ricerche su google parola corona virus: <https://trends.google.it/trends/?geo=IT>

**Siti web utili per grafici:**

<https://app.datawrapper.de/>

<https://infogram.com/>