***FORME NORMALI***

**Punto 2. La verifica della 3NF per lo schema della tabella precedentemente istanziata e l'eventuale decomposizione con la definizione di tutti i vincoli (mediante comandi DDL);**

Nella progettazione di un database ricopre un ruolo fondamentale l’attività di **normalizzazione**, infatti tale metodologia, permette di ottimizzare lo schema relazionale eliminando le seguenti problematiche:

* **Ridondanza;**
* **Anomalie di aggiornamento;**
* **Anomalie di cancellazione;**
* **Anomalie di inserimento.**

Iniziamo a lavorare sul nostro schema relazionale, cioè quello creato nel punto precedente, attraverso i file forniti dalla protezione civile con i dati relativi ai casi divisi per provincia. Quindi partendo dal seguente schema:

* **MASTER** (**DATA\_REG**, STATO, CODICE\_REGIONE, DENOMINAZIONE\_REGIONE, **CODICE\_PROVINCIA**, DENOMINAZIONE\_PROVINCIA, SIGLA\_PROVINCIA, LATITUDINE, LONGITUDINE, TOTALE\_CASI, NOTE\_IT, NOTE\_EN).

Definiamo le **dipendenze funzionali (DF)**, ovvero i legami che esistono tra gli attributi della relazione, questo passaggio ci aiuta a scoprire i vincoli d’integrità esistenti sullo schema di relazione, nel nostro caso abbiamo le seguenti DF:

* **DF0:** STATO 🡪 CODICE\_REGIONE;
* **DF1:** CODICE\_REGIONE 🡪 CODICE\_PROVINCIA;
* **DF2:** CODICE\_REGIONE 🡪 DENOMINAZIONE\_REGIONE;
* **DF3:** CODICE\_PROVINCIA 🡪 DENOMINAZIONE\_PROVINCIA, SIGLA\_PROVINCIA, LATITUDINE, LONGITUDINE.

(**nota:** Se indichiamo con Y 🡪 Z, la dipendenza funzionale; Leggeremo che Y determina Z, o viceversa, che Z dipende funzionalmente da Y)

Adesso possiamo passare alla verifica della **3NF**, in particolare avremo che il nostro schema è in **3NF** se rispetta le seguenti caratteristiche:

* è in **2NF**;
* ogni attributo non primo di **MASTER** non dipende in modo transitivo dalla sua chiave, cioè dipende solo dalla chiave, e non vi sono dipendenze funzionali tra attributi non primi.

Il secondo punto non viene rispettato dalle **DF0** e **DF2**, in quanto tali dipendenze funzionali riguardano due attributi non primi. Quindi occorre fare due decomposizioni una sull’attributo STATO per la **DF0** ed un’altra sull’attributo CODICE\_REGIONE per la **DF2**. Tali decomposizioni hanno lo scopo di mantenere le dipendenze funzionali, ma al contempo rispettare la **3NF** partendo quindi dal seguente schema:

* **MASTER** (**DATA\_REG**, STATO, CODICE\_REGIONE, DENOMINAZIONE\_REGIONE, **CODICE\_PROVINCIA**, DENOMINAZIONE\_PROVINCIA, SIGLA\_PROVINCIA, LATITUDINE, LONGITUDINE, TOTALE\_CASI, NOTE\_IT, NOTE\_EN).

Avremo la seguente decomposizione:

* **MASTER** (**DATA\_REG**, CODICE\_REGIONE: REGIONI, **CODICE\_PROVINCIA**, DENOMINAZIONE\_PROVINCIA, SIGLA\_PROVINCIA, LATITUDINE, LONGITUDINE, TOTALE\_CASI, NOTE\_IT, NOTE\_EN);
* **REGIONI** (**CODICE\_REGIONE**, DENOMINAZIONE\_REGIONE, STATO: STATI);
* **STATI** (STATO).

Adesso ci rimane da verificare la **2NF**, ovvero dobbiamo controllare se il nostro schema soddisfa i seguenti requisiti:

* è in **1NF**;
* ogni attributo non primo dipende completamente da ogni chiave di R(X).

Anche in questo caso il secondo punto ci crea alcuni problemi, infatti, la **DF3** ci dice che alcuni attributi non primi non dipendono completamente dalla chiave dello schema, ma solo da un sottoinsieme di essa.

Quando uno schema non soddisfa la **2NF** come in questo caso, la soluzione migliore è quella di decomporre lo schema, tuttavia, tale operazione va fatta in maniera corretta al fine di evitare problemi nella consistenza dei dati e nel rispetto dei vincoli.

In particolare, ciò a cui dobbiamo ambire in questa fase è una **decomposizione senza perdita**, per ottenere ciò bisogna effettuare la decomposizione sulla base di ‘attributi comuni’ che contengono una chiave per almeno una delle relazioni decomposte. Quindi, nel nostro caso avremo che il seguente schema:

* **MASTER** (**DATA\_REG**, CODICE\_REGIONE: REGIONI, **CODICE\_PROVINCIA**, DENOMINAZIONE\_PROVINCIA, SIGLA\_PROVINCIA, LATITUDINE, LONGITUDINE, TOTALE\_CASI, NOTE\_IT, NOTE\_EN);
* **REGIONI** (**CODICE\_REGIONE**, DENOMINAZIONE\_REGIONE, STATO: STATI);
* **STATI** (STATO).

Viene decomposto in questa maniera:

* **MASTER** (**DATA\_REG**, CODICE\_REGIONE: REGIONI, **CODICE\_PROVINCIA: PROVINCE**, TOTALE\_CASI, NOTE\_IT, NOTE\_EN);
* **PROVINCE** (**CODICE\_PROVINCIA**, DENOMINAZIONE\_PROVINCIA, SIGLA\_PROVINCIA, LATITUDINE, LONGITUDINE);
* **REGIONI** (**CODICE\_REGIONE**, DENOMINAZIONE\_REGIONE, STATO: STATI);
* **STATI** (STATO).

Infine, la verifica della **1NF** è banale in quanto ogni attributo di **MASTER** è un attributo semplice, ovvero con dominio atomico.

In definitiva il nostro schema di relazione adesso è in **3NF**, tuttavia, la **DF1** non rispetta la **forma normale di Boyce e Codd** perché abbiamo che un attributo non primo determina un attributo primo. Per eliminare tale problema, basta creare un’associazione tra le entità PROVINCE e REGIONI, in altri termini occorre fare una decomposizione su CODICE\_PROVINCIA, avremo quindi lo schema finale:

* **MASTER** (**DATA\_REG**, **CODICE\_PROVINCIA: PROVINCE**, TOTALE\_CASI, NOTE\_IT, NOTE\_EN);
* **PROVINCE** (**CODICE\_PROVINCIA**, CODICE\_REGIONE: REGIONI, DENOMINAZIONE\_PROVINCIA, SIGLA\_PROVINCIA, LATITUDINE, LONGITUDINE);
* **REGIONI** (**CODICE\_REGIONE**, DENOMINAZIONE\_REGIONE, STATO: STATI);
* **STATI** (STATO).

**COMANDI DDL**

*(DA INSERIRE)*