

Bredariol Francesco SM3201379

Esame Base di Dati anno 2024

# IL SISTEMA TRIBUTARIO DELL'IMPERO GALATTICO DI AGOSTINO MDI

*Siamo nell'anno 3249 dR.*

*L'imperatore Agostino MDI chiede al consiglio di progettare una base di dati per poter visualizzare l'attuale situazione del sistema tributario dell'Impero Galattico. Il suo consiglio, poco furbamente, affida il compito ad un mio {for i in range (1000) "pro-"}nipote, il quale, non essendo particolarmente capace, decide di affidarsi ad una telefonata spazio-temporale per chiamarmi e chiedermi aiuto. Per sua fortuna la tecnologia MySQL persiste ancora nella sua epoca e dunque, dovendo io preparare l'esame di base di dati e non avendo idea di cosa costruire, decido di dargli una mano. Allego le indicazioni imperiali su ciò che occorre preparare.*

# INDICAZIONI IMPERIALI

Buongiorno, come già spiegato in previa sede imperiale, occorre che lei ci aiuti a sviluppare una base di dati in appoggio al sistema tributario imperiale (**STI**).

Le ricordiamo brevemente la suddivisione fiscale del nostro impero: alla base troviamo i **Mondi**, i quali hanno tutti un nome ed una data di ingresso nell'Impero, oltre che ad un prodotto interno lordo (indicato in milioni di agostiniani), un numero di abitanti ed un indice di superficie produttiva. Ogni Mondo rientra in una fascia fiscale, la quale serve al STI per calcolare i tributi pro Mondo. Faccia riferimento alla **TUFF** (Tabella Unica delle Fasce Fiscali) per il calcolo dei tributi dovuti. I Mondi appartengono ai **Distretti Imperiali**, un ente fiscale più ampio identificato dal codice Imperiale di Distretto. I **Consoli** sono coloro i quali governano i Distretti ed, oltre ad avere un Identificativo Imperiale, hanno un contatto telefonico. Hanno inoltre un nome, un cognome, una data di nascita ed una data di elezione. I Consoli sono coloro i quali, in nome del Distretto Imperiale che governano, versano i tributi. I Consoli vengono eletti ogni 3 anni e durante la loro carriera possono essere Consoli solo presso un unico Distretto ed una ed una sola volta. Una volta che viene eletto un nuovo Console il vecchio Console dovrà aggiornare la sua data di fine mandato. Ogni Distretto Imperiale infine appartiene ad una **Regione Galattica**. Oltre ad avere una data di ingresso nell'Impero ed un nome, quest'ultima è importante poiché, nonostante la valuta ufficiale sia l'Agostiniano, ogni regione può avere la propria valuta indipendente. Le valute sono identificate da un tasso di conversione che alle volte può essere richiesto per visualizzare i tributi nella moneta regionale. Per la conversione si riferisca alla **TUCG** (Tabella Unica della Conversione Galattica). I **Tributi**, identificati da un opportuno codice, data emissione, valore, Distretto a cui vengono richiesti e status (Pagato o Non Pagato), vengono emessi annualmente e vengono poi registrati tutti i **Pagamenti**. Per ogni pagamento bisogna registrarne la data, il Tributo di Riferimento, il codice imperiale del Console del Distretto interessato e quello di altri due Consoli che vengono chiamati a fare da Testimoni. I Testimoni non possono appartenere alla stessa Regione del Console che versa il tributo. Le alleghiamo le operazioni più importanti indicate direttamente da sua magnificenza l'imperatore Agostino MDI. Infine le alleghiamo la TUFF e la TUCG che siamo sicuri lei potrà integrare tranquillamente nel sistema.

# OPERAZIONI FONDAMENTALI

- A. visualizzare i tributi dovuti da ogni Distretto in tempo reale, nonché da ogni Regione per poter comprendere in quali zone esercitare maggiore pressione (anche una volta al giorno se necessario)
- B. visualizzare i 5 Distretti che hanno contribuito maggiormente in un determinato periodo (operazione una tantum)
- C. dato un determinato Distretto visualizzare i tributi che deve nella sua moneta Regionale (una tantum)
- D. visualizzare i 5 paesi su cui incidono maggiormente i tributi così da poter dare la possibilità di cambiarne la fascia fiscale di appartenenza per abbassare la pressione fiscale (la pressione è vista come rapporto tra tributi e pil) (una volta all'anno)
- E. visualizzare per un dato Distretto lo storico dei Consoli che l'hanno governato (operazione una tantum)

**AGOSTINO MDI**

## TUCG (Tabella Unica della Conversione Galattica)

VALUTA	TASSO CONVERSIONE
Agostiniano	1
Bersko	6.3
Cherubo	1.2
Curmino	8.2
Gerano	14.5
Jordano	3549.1
Kilo	1000
Korolo	2.3
Nippo	1320.2
Nuevo	46.4

La formula di conversione è la seguente:

$$1 \text{ Agostiniano} = \text{Tasso conversione} * \text{Valuta}$$

## TUFF (Tabella Unica delle Fasce Fiscali)

FASCIA FISCALE	INDICE PRO CAPITALE	INDICE PRO METRO
1	0.05	0.01
2	0.06	0.02
3	0.075	0.03
4	0.1	0.05
5	0.12	0.06
6	0.15	0.1
7	0.2	0.2

La formula per il calcolo tributo pro mondo in Agostiniani è:

**popolazione x indice pro capite + superficie produttiva x indice pro metro**

## GLOSSARIO DEI TERMINI

TERMINE	DEFINIZIONE	COLLEGAMENTO
Mondo	Elemento di base su cui il STI calcola i tributi dovuti da ogni Distretto. ogni Mondo appartiene ad un Distretto	Distretto, Fascia Fiscale
Fascia Fiscale	Indice relativo ad un mondo utilizzato dalla STI per il calcolo dei Tributi del distretto.	Mondo
Distretti Imperiali	Elemento fiscale più complesso a capo dei Mondì. Sono i diretti interessati del sistema tributario poiché sono i loro Consoli imperiali a pagare i tributi.	Mondì, Regione Galattica, Console Imperiale, Tributo
Regioni Galattiche	Elemento fiscale più ampio che ingloba i vari Distretti. Serve al STI per poter calcolare i tassi di conversione dei tributi da richiedere poiché ogni Regione Galattica, per vari motivi, può avere valute diverse.	Distretto Imperiale, Valuta
Valuta	Può essere di vario tipo, esiste una tabella che serve alla STI per convertirla in Agostiniani, la moneta dell'impero.	Regione Galattica
Tributo	Emesso dalla STI annualmente, calcolato sui dati dei Distretti. Viene spedito ad un Console e da lui deve essere pagato.	Distretto Imperiale, Console
Pagamento	E' il processo mediante il quale un Console versa un tributo al STI e altri due	Console, Tributo

TERMINE	DEFINIZIONE	COLLEGAMENTO
	Consoli fanno da testimoni. Fa riferimento ad un Tributo preciso.	
Testimonianza	Nel processo di pagamento è la presenza di due Consoli testimoni per confermare l'avvenuto pagamento.	Console, Tributo

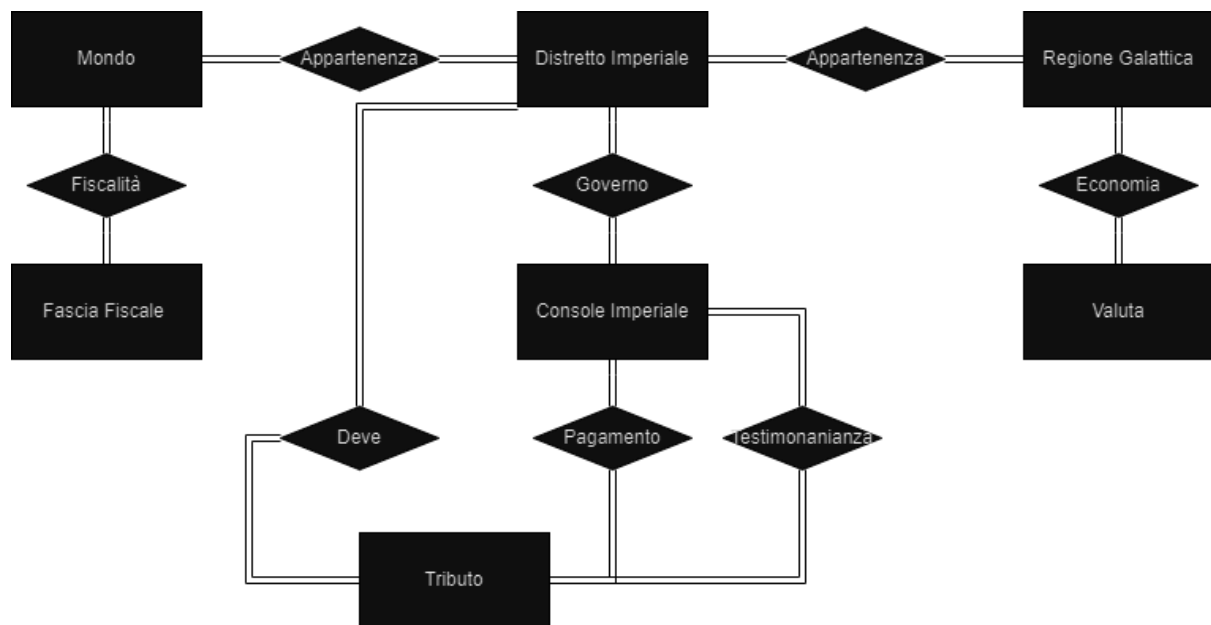
## VINCOLI NON ESPRIMIBILI

Dall'analisi del testo e grazie all'analisi del glossario dei termini riusciamo ad individuare i seguenti vincoli non esprimibili:

- durante il Pagamento i Consoli Testimoni non possono venire dalla stessa Regione Galattica del Console Pagante
- quando un nuovo Console viene eletto si deve aggiornare automaticamente la data di fine mandato del Console uscente e anche codice Console del Distretto di riferimento
- quando viene effettuato un Pagamento dobbiamo aggiornare lo status del tributo di riferimento

**NOTA TEMPORALE:** l'anno 3249 dR equivale all'incirca al 2000000 dC. Per comodità nella fase di programmazione le date verranno convertite in anni dC e scalati di circa 2000000 di anni (cioè l'anno 3249 dR sarà visualizzato come 2029 dC)

# PRIMO SCHEMA RELAZIONALE



Dall'analisi del glossario dei termini questo è il primo schema relazionale che deduciamo. A prima vista si nota un ciclo tra le entità Distretto - Console - Tributo, tuttavia esso è necessario e non crea problemi: il Distretto Imperiale è l'entità a dovere il Tributo nei confronti del Sistema Tributario, ma è il Console governante ad effettuare il pagamento. Se volessimo evitarlo potremmo attribuire al Console il dovere del Tributo, ma sarebbe scorretto poiché, per come funziona il sistema, il Console ha il solo compito di pagare (e testimoniare) il Tributo.

Studiamo ora le varie relazioni ed entità per migliorare lo schema relazionale e per poter introdurre le cardinalità delle relazioni tramite il dizionario dei dati. Eseguiamo infine lo studio dei volumi e l'analisi delle ridondanze per ristrutturare al meglio il tutto e per prepararci a costruire lo schema logico.

# DIZIONARIO DEI DATI

ENTITA'	ATTRIBUTI	CHIAVE
<b>Mondo</b>	nome, data di ingresso nell'Impero, prodotto interno lordo, numero di abitanti, superficie produttiva, fascia fiscale	<b>pk</b> : nome <b>fk</b> : fascia fiscale, codice Distretto Imperiale
<b>Fascia fiscale</b>	codice fascia, indice pro capite, indice pro metro	<b>pk</b> : codice fascia
<b>Distretto</b>	codice Distretto Imperiale, codice Regione Galattica, codice Console Imperiale	<b>pk</b> : codice Distretto Imperiale <b>fk</b> : codice Regione Galattica, codice Console Imperiale
<b>Console</b>	data di Nascita, data di Elezione, nome, cognome, codice Console Imperiale, codice Distretto Imperiale, contatto telefonico, data fine Mandato	<b>pk</b> : codice Console Imperiale <b>fk</b> : codice Distretto Imperiale
<b>Regione Galattica</b>	codice Regione Galattica, nome, valuta, data ingresso nell'Impero	<b>pk</b> : codice Regione Galattica <b>fk</b> : valuta
<b>Valuta</b>	nome, tasso di conversione	<b>pk</b> : nome
<b>Pagamento</b>	numero di pagamento, numero di Tributo di riferimento, codice Console di riferimento, codice Console di testimonianza 1, codice Console di testimonianza 2, data pagamento	<b>pk</b> : numero di pagamento <b>fk</b> : numero di Tributo di riferimento, codice Console di riferimento, codice Console di testimonianza 1, codice Console di testimonianza 2
<b>Tributo</b>	numero di Tributo, codice Distretto Imperiale di riferimento, data di emissione, valore, status	<b>pk</b> : numero di Tributo <b>fk</b> : codice Distretto Imperiale di riferimento

Notiamo che il pagamento è sicuramente un'entità a sé e non una semplice relazione. La testimonianza la possiamo inglobare nel pagamento. Inoltre un Distretto avrà una



cardinalità 1:N nei confronti dei Consoli, non perché ci siano più Consoli a governare contemporaneamente, ma perché in questo modo andiamo a descrivere lo storico dei Consoli che hanno governato un certo Distretto.

## STUDIO DEI VOLUMI

E' noto che l'Impero Galattico ingloba all'incirca un milione di mondi al suo interno, un qualche migliaio di Distretti e 79 regioni Galattiche. Supponendo che la base di dati venga integrata con dati risalenti al più di un secolo fa, sappiamo che i consoli sono all'incirca  $3 \cdot 10^4$ . Invece per quanto riguarda le relazioni le studiamo nel periodo di tempo "annuale" che è ciò che ci interessa visto che l'STI emette annualmente i tributi.

CONCETTO	TIPO	VOLUME
MONDO	E	$10^6$
FASCIA FISCALE	E	7
DISTRETTO	E	$10^3$
REGIONE	E	79
CONSOLI	E	$3 \cdot 10^4$
PAGAMENTI	E	$10^5$
TRIBUTI	E	$10^5$
FISCALITA'	R	$10^6$
APPARTENENZA M	R	$10^6$
APPARTENENZA D	R	$10^3$
GOVERNO	R	$10^3$
PARTECIPAZIONE	R	$3 \cdot 10^3$
DEFINIZIONE	R	$10^3$
DOVERE	R	$10^3$

Se la base di dati dovesse persistere per migliaia di anni quasi sicuramente dovremmo pensare di migrare parte dei tributi e dei pagamenti su una memoria esterna poiché potrebbe cominciare ad appesantire l'interrogazione del database.

Analizziamo ora le operazioni di interesse per capire se le performance si potrebbero ottimizzare o meno in base alle relative frequenze.

## OPERAZIONI DI INTERESSE

Facendo riferimento alle operazioni tramite le lettere indicate al punto "OPERAZIONI FONDAMENTALI". Tutte le operazioni indicate "una tantum" vanno assunte con frequenza tra il semestrale ed il lustrale.

OPERAZIONE	TIPO	FREQUENZA
Emissione tributi	Batch	Annuale
A	Interattiva	Giornaliera
B	Interattiva	Tantum
C	Interattiva	Tantum
D	Interattiva	Annuale
E	Interattiva	Tantum

Dunque notiamo che l'ottimizzazione delle performance sarà sicuramente da rivolgersi verso l'operazione A. L'operazione A fortunatamente è già pressoché immediata poiché, grazie al mantenimento dell'entità tributi calcolata annualmente in cui ci sono quasi tutte le informazioni richieste l'interrogazione della base di dati per compiere le operazioni risulta molto veloce. Solo se dovessimo effettuare un raggruppamento per Regione l'operazione potrebbe subire un leggero rallentamento, tuttavia stiamo parlando semplicemente di esplicitare una relazione diretta tra il Distretto e la Regione dunque nulla che rallenti eccessivamente il processo. Potremmo considerare di inserire nell'entità Tributo il codice della Regione ma questo porterebbe a Ridondanza evitabile. Per quanto riguarda le altre operazioni possiamo dire che hanno tutte una frequenza relativamente bassa. Bisogna osservare che ci sono delle operazioni, gestite dai tecnici del STI sotto il cofano, che sono gli aggiornamenti della TUFF, della TUCG e dei consoli alle elezioni: queste operazioni sono in realtà molto leggere perché implicano, le prime due, il semplice aggiornamento di un valore in una tabella, la terza, l'aggiunta al più di un migliaio di righe ogni 3 anni in una singola tabella. Infine c'è il processo di pagamento che avviene qualche migliaio di volte ogni anno in una singola sede (al più una decina di terminali presenti) ed implica l'aggiunta di righe nella tabella Pagamenti, processo pressoché immediato e senza problemi.

## ANALISI RIDONDANZE

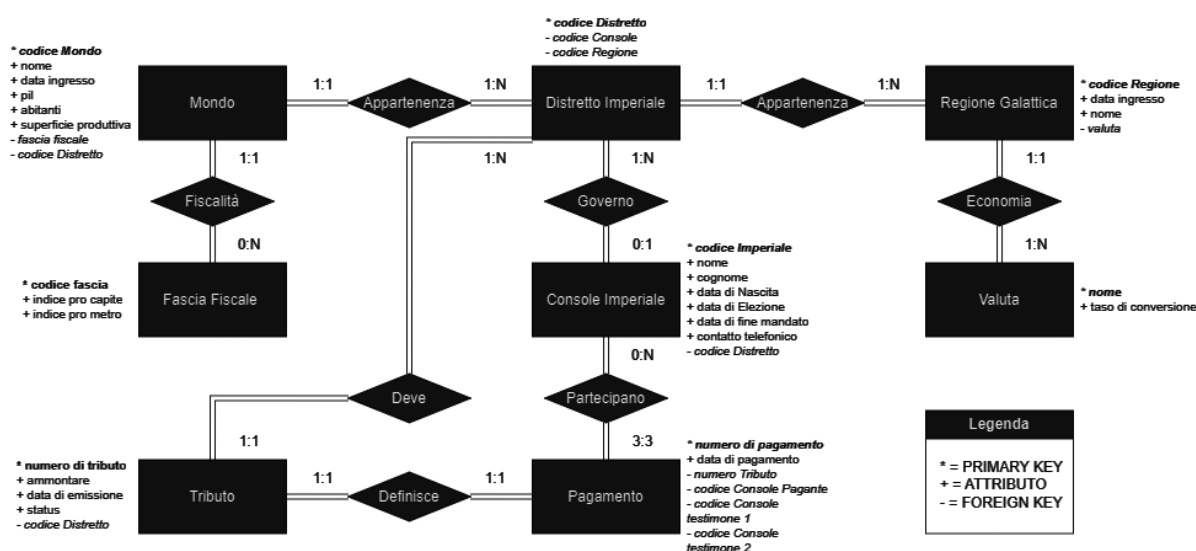
Non notiamo attributi composti da cui possono nascere ridondanze, tuttavia sia gli attributi Fascia Fiscale riferito a Mondo che Valuta riferita a Regione Galattica potrebbero inglobare completamente le relative entità Fascia Fiscale e Valuta. Ma le entità Fascia Fiscale e Valuta vengono definite da tabelle esterne, la TUFF e la TUCG, che ci è stato esplicitamente chiesto di integrare all'interno della base di dati. Inoltre mantenerle relazioni esterne ci permette di risparmiare una quantità di aggiornamenti enormi: se una fascia fiscale cambia (es) l'indice pro capite, ci è sufficiente lavorare sulla TUFF (una  $7 \times 3$ ) anziché lavorare su Mondi ( $10^6 \times 8$ ). Discorso equivalente, ma con cardinalità minori, per la TUCG. Decidiamo dunque di mantenere inalterata la struttura.

Per quanto riguarda il ciclo (di cui abbiamo già anticipatamente parlato) potremmo notare che l'entità Tributo ha campi completamente calcolabili (l'ammontare). In questo caso tuttavia dobbiamo considerare che vengono calcolati una ed una sola volta all'anno, mentre vengono richiesti anche più e più volte. Dunque averli salvati altrove ci conviene in termini di risparmio di calcolo poiché una sola volta li calcoliamo e molte volte li visualizziamo. Inoltre il campo "ammontare" deve mantenersi invariato nel tempo, poiché dipende strettamente dallo stato dell'Impero alla data di emissione del tributo stesso.

# SCHEMA RELAZIONALE

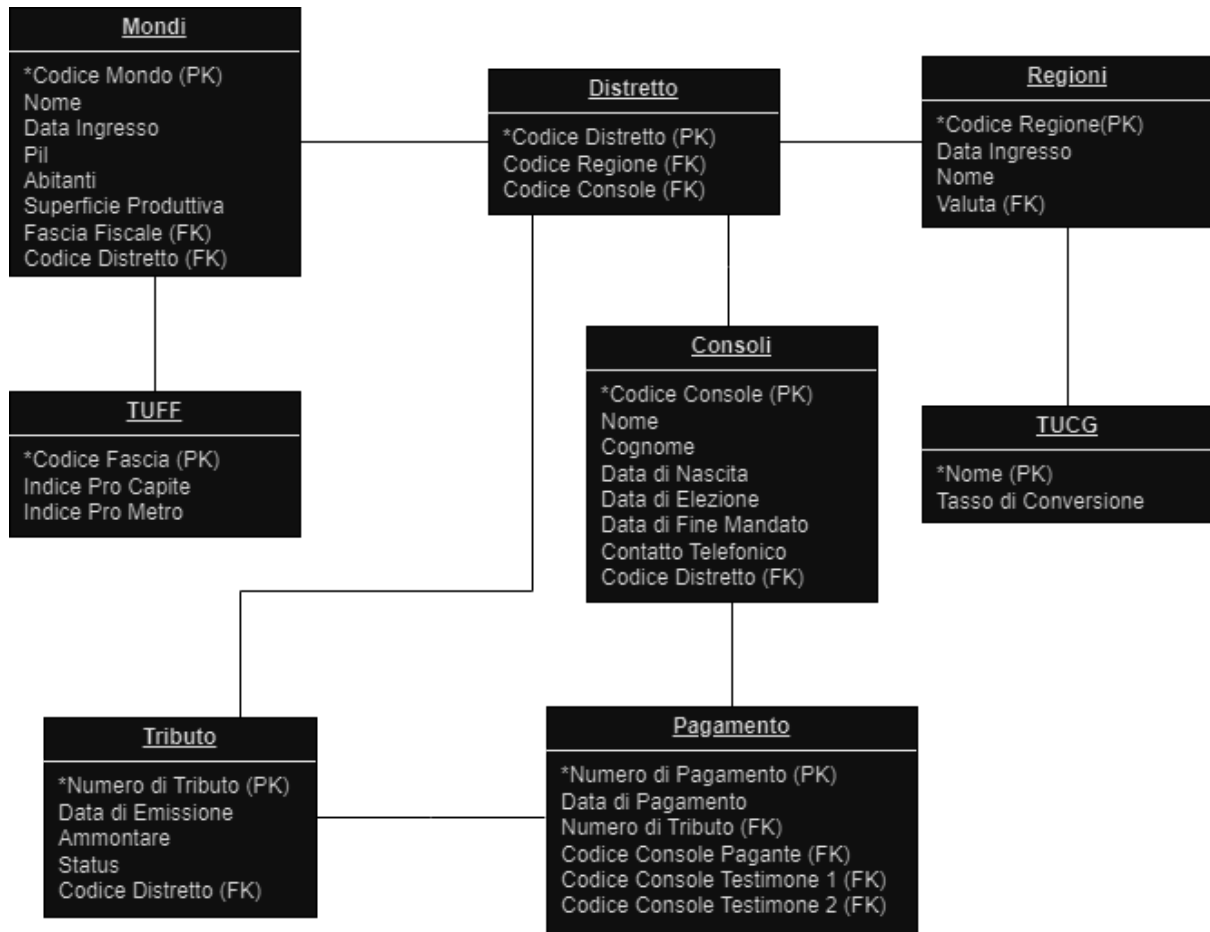
Dopo tutta la serie di analisi eseguite siamo giunti a definire il seguente schema relazionale. Inseriamo sotto anche la lista di identificatori scelti per ogni Entità.

- **FASCIA FISCALE** : *codice fascia*
- **MONDO** : *codice Mondo*
- **DISTRETTO IMPERIALE** : *codice Distretto*
- **REGIONE GALATTICA** : *codice Regione*
- **VALUTA** : *nome*
- **CONSOLE IMPERIALE** : *codice Imperiale*
- **PAGAMENTO** : *numero di pagamento*
- **TRIBUTO** : *numero di tributo*



Notiamo che la relazione Pagamento nei confronti del Console Imperiale è definita 3:3, che non dovrebbe essere propriamente un tipo di relazione. Tuttavia nel nostro caso è ben definita poiché sappiamo che in un Pagamento partecipano **sempre e solo** tre Consoli diversi per volta. Una relazione N:N sarebbe stata un po' ambigua poiché avrebbe lasciato il dubbio sulla possibile interpretazione. Tutto il resto è stato già ampiamente affrontato e visto, possiamo dunque passare allo schema Logico.

# SCHEMA LOGICO



## NORMALIZZAZIONE

Notiamo che la base di dati è già in 1FN.

Notiamo che la base di dati è già in 2FN (dopotutto tutte le PK sono atomiche).

Tutte le tabelle sono in 3FN ad eccezione della tabella Tributo che ha un campo calcolato.

Notiamo che in realtà non è un vero campo calcolato poiché non cambia col cambiare dello stato del DB e dunque abbiamo anche la 3FN.

# CODICE SQL RILEVANTE

Per quanto riguarda l'inserimento dei Consoli, poiché all'inserimento di uno deve avvenire l'aggiornamento dell'altro, e MySQL non supporta i trigger che agiscono sulla stessa tabella da cui vengono invocati, dobbiamo andare a definire noi la funzione per inserire i consoli da zero.

```
139 • CREATE PROCEDURE inserisciConsole(  
140     IN CI int,  
141     IN N varchar(100),  
142     IN C varchar(100),  
143     IN DN Date,  
144     IN DE Date,  
145     IN CT varchar (100),  
146     in CD int  
147 )  
148 BEGIN  
149     insert into Consoli  
150     (CodiceImperiale, Nome, Cognome, DataDiNascita, DataDiElezione, ContattoTelefonico, CodiceDistretto)  
151     values (CI, N, C, DN, DE, CT, CD);  
152     update Consoli  
153     set DataDiFineMandato = DE  
154     where CD = CodiceDistretto  
155     and DataDiFineMandato is null  
156     and CI != CodiceImperiale;  
157 END;
```

Pensiamo poi al trigger per quanto riguarda la tabella Distretti e la tabella Tributi

```
159 • CREATE TRIGGER CambioDiGoverno  
160 AFTER INSERT ON Consoli  
161 FOR EACH ROW  
162 BEGIN  
163     CALL updateDistretti(NEW.CodiceImperiale, NEW.CodiceDistretto);  
164 END; //  
165  
166 • CREATE TRIGGER aggiornaStatus  
167 AFTER INSERT ON Pagamenti  
168 FOR EACH ROW  
169 BEGIN  
170     UPDATE Tributi  
171     SET status = "Pagato"  
172     WHERE NumeroTributo = NEW.NumeroTributo;  
173 END;
```

Per poi rendere il codice più leggibile costruiamo ad hoc delle procedure che useremo nel trigger per verificare la legittimità di un pagamento.

```
175 • CREATE PROCEDURE regioneConsole(  
176     IN ci int,  
177     OUT cr int  
178 )  
179 BEGIN  
180     SELECT codiceRegione INTO cr FROM Distretti where codiceConsole = ci;  
181 END; //  
182  
183 • CREATE PROCEDURE distrettoPagamento(  
184     IN nt int,  
185     OUT dp int  
186 )  
187 BEGIN  
188     SELECT codiceDistretto INTO dp FROM Tributi where numeroTributo = nt;  
189 END; //  
190  
191 • CREATE PROCEDURE distrettoConsole(  
192     IN ci int,  
193     OUT cd int  
194 )  
195 BEGIN  
196     SELECT codiceDistretto INTO cd from Consoli where codiceImperiale = ci;  
197 END; //  
198  
199 • CREATE PROCEDURE statusTributo(  
200     IN nt int,  
201     OUT s varchar(100)  
202 )  
203 BEGIN  
204     SELECT status INTO s from tributi where numeroTributo = nt;  
205 END; //
```

Creiamo quindi ora il trigger sui Pagamenti

```
208 • CREATE TRIGGER controllaPagamenti
209 BEFORE INSERT ON Pagamenti
210 FOR EACH ROW
211 BEGIN
212     DECLARE ccp int; -- codice regione console pagante
213     DECLARE cct1 int; -- codice regione console testimone 1
214     DECLARE cct2 int; -- codice regione console testimone 2
215     DECLARE cdc int; -- codice distretto console pagante
216     DECLARE cdp int; -- codice distretto pagamento
217     DECLARE s varchar(100); -- status tributo di riferimento
218     CALL regioneConsole(NEW.CodiceConsolePagante, ccp);
219     CALL regioneConsole(NEW.CodiceConsoleTestimone1, cct1);
220     CALL regioneConsole(NEW.CodiceConsoleTestimone2, cct2);
221     CALL distrettoConsole(NEW.CodiceConsolePagante, cdc);
222     CALL distrettoPagamento(NEW.numeroPagamento, cdp);
223     CALL statusTributo(NEW.numeroTributo, s);
224
225     IF NEW.CodiceConsoleTestimone1 = NEW.CodiceConsoleTestimone2 THEN
226         SIGNAL SQLSTATE '45000'
227         SET MESSAGE_TEXT = 'il console pagante non governa il distretto interessato dal tributo. pagamento non autorizzato';
228     END IF;
229
230     IF s = "Pagato" THEN
231         SIGNAL SQLSTATE '45000'
232         SET MESSAGE_TEXT = 'pagamento già avvenuto.';
233     END IF;
234
235     IF cdc != cdp THEN
236         SIGNAL SQLSTATE '45000'
237         SET MESSAGE_TEXT = 'il console pagante non governa il distretto interessato dal tributo. pagamento non autorizzato';
238     END IF ;
239
240     IF ccp IS NULL or cct1 IS NULL or cct2 IS NULL THEN
241         SIGNAL SQLSTATE '45000'
242         SET MESSAGE_TEXT = 'uno dei consoli non è più in carica. pagamento non autorizzato';
243     END IF;
244
245     IF ccp = cct1 or ccp = cct2 THEN
246         SIGNAL SQLSTATE '45000'
247         SET MESSAGE_TEXT = 'testimoni provenienti dalla stessa regione. pagamento non autorizzato';
248     END IF;
249 END;
```



Per quanto riguarda l'operazione di batch per l'emissione dei tributi creiamo una stored procedure. Usiamo un cursore per ottimizzare l'operazione.

```
328 DELIMITER //
329 ● CREATE PROCEDURE batchTributi(
330     IN DataAttuale Date
331 )
332 BEGIN
333     DECLARE t float;
334     DECLARE cd int;
335     DECLARE finished int default 0;
336
337     DECLARE tributi_cursor CURSOR for
338     select round(sum(indiceprometro * superficieproduttiva + indiceprocapite * abitanti)) as tributo,
339     codiceDistretto from mondi
340     join TUFF on mondi.fasciaFiscale = TUFF.codiceFascia
341     group by codiceDIstretto;
342
343     DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;
344
345     OPEN tributi_cursor;
346     WHILE (finished = 0) DO
347         FETCH tributi_cursor into t, cd;
348         INSERT INTO tributi (DataDiEmissione, Ammontare, CodiceDistretto) values (DataAttuale, t, cd);
349     END WHILE;
350     CLOSE tributi_cursor;
351 END;
```

Per le operazioni fondamentali vediamo il codice sql come semplici query con variabili di esempio, tuttavia nel programma Python le operazioni verranno eseguite tramite prepared statement / stored procedures.

```
384 • SET @DataBegin = '2024-01-01';
385 • SET @DataEnd = '2030-01-01';
386 • SET @DistrettoProva = 19;

388 -- operazione A
389 • select sum(if(status = 'non pagato', ammontare, 0)) as debito, codiceDistretto
390 from tributi
391 group by codiceDistretto
392 order by debito;
393
394 • select sum(if(status = 'non pagato', ammontare, 0)) as debito, distretti.codiceRegione
395 from tributi join Distretti on tributi.codiceDistretto = distretti.CodiceDistretto
396 group by distretti.codiceRegione
397 order by debito;
400 -- operazione B
401 • select sum(if(status = 'pagato', ammontare, 0)) as contributo, codiceDistretto
402 from tributi
403 join pagamenti
404 on tributi.numeroTributo = pagamenti.numeroTributo
405 where dataDiPagamento between @dataBegin and @dataEnd
406 group by codiceDistretto
407 order by contributo desc limit 5;
409 -- operazione C
410 • select sum(if(status = 'non pagato', ammontare, 0))*tassodiconversione as Debito,
411 TUCG.Valuta, tributi.codiceDistretto as Distretto
412 from tributi
413 join distretti
414 on tributi.codiceDistretto = distretti.codiceDistretto
415 join regioni
416 on distretti.codiceRegione = regioni.codiceRegione
417 join TUCG
418 on regioni.valuta = TUCG.valuta
419 where tributi.codiceDistretto = @DistrettoProva
420 group by tributi.codiceDistretto;
422 -- operazione D
423 • select (abitanti*indiceprocapite + superficieproduttiva*indiceprometro)/pil as pressione, nome
424 from mondi
425 join TUFF on fasciaFiscale = codiceFascia
426 order by pressione desc limit 5;
428 -- operazione E
429 • select nome, cognome, codiceImperiale
430 from consoli
431 where codiceDistretto = @DistrettoProva
432 order by DataDiElezione;
```

# CODICE PYTHON

Il codice python è presente come file .ipynb. Il codice ha bisogno di aver istanziato il database su MySQL sotto il nome STI. Per questo è presente anche uno script MySQL completamente funzionante che setta tutto il necessario affinché si possa usare il notebook. Chiaramente per velocizzare i processi la base di dati ha una dimensione molto ridotta rispetto al problema teorico (si vedano i volumi teorici alla sezione "STUDIO DEI VOLUMI") presentato nel progetto. Il notebook è completo di documentazione markdown. Aggiungo anche una cartella contenente il codice usato per generare i dati iniziali del database.

## CONCLUSIONE

*Caro {for i in range (1000) "pro-"} nipote,  
spero che questo progetto possa esserti d'aiuto e possa accontentare l'Imperatore  
Galattico Agostino MDI.  
Ti lascio con questo augurio, mentre auguro a me stesso fortuna con il professor De  
Lorenzo ed il mio esame di Base di Dati.  
Con affetto, tuo  
{for i in range (1000) "pro-"} zio,*

**FRANCESCO BREDARIOL**