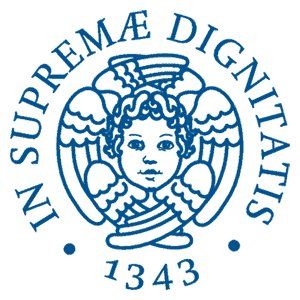
University of Pisa

Master’s degree in

Data Science & Business Informatics

Big Data Analytics



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Assignment - Part 1

Group 17

Carlo Alberto Carrucciu , Giacomo Lo Dico , Gian Maria Pandolfi

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

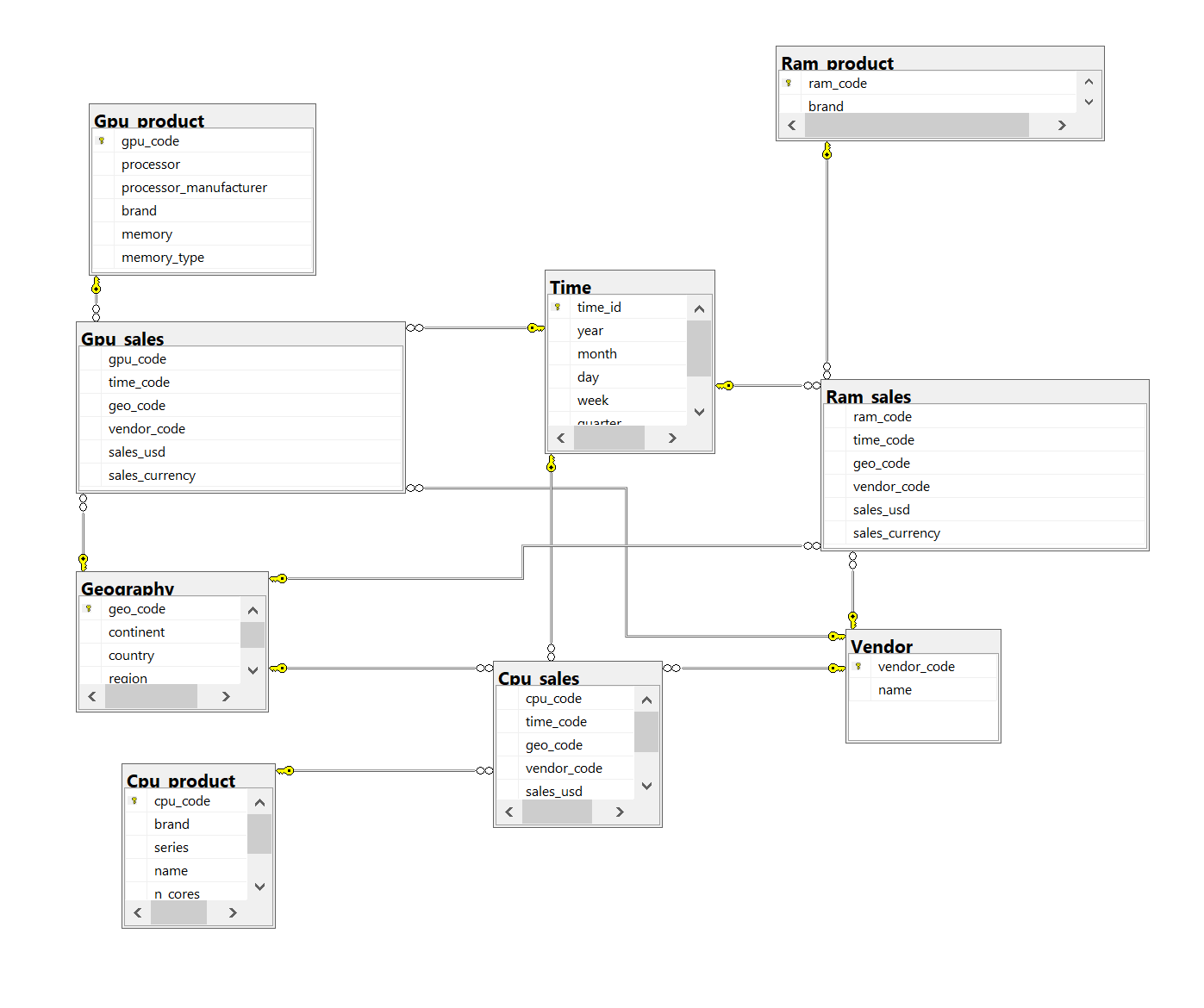
**Parte 1**

Per la parte prima del progetto, viene richiesto di creare e popolare un dataset partendo da diversi file csv (*Id, gpu\_code, cpu\_code, ram\_code, time\_code, geo\_code, vendor\_code , sales\_uds, sales\_currency*).

**Task 0:**

Durante la fase di creazione del database come richiesto nell’*assignment 0,* sono stati utilizzati esclusivamente gli strumenti forniti dal software ***windows sql management studio****.*

Nella fase di creazione, sono state definite le tabelle con “*nome tabella*”, dopodichè si è proceduto alla creazione degli attributi definendo il tipo e la lunghezza in base ai dati contenuti nei vari csv, inoltre è stata attivata opzione **Identity (1,1)** nelle tabelle che richiedevano una chiave primaria progressiva.

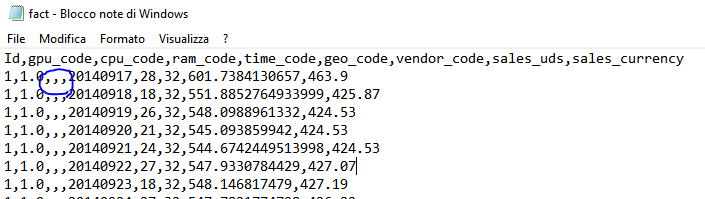
Non sono state definite le relazioni tramite chiavi perchè avrebbero potuto generare degli errori di mancata corrispondenza tra **PRIMARY KEY e FOREIGN KEY**. Tale problema è stato risolto durante la fase successiva, ovvero quella di caricamento (*assignment 2*); popolando per prime, le tabelle che contengono le chiavi primarie. 

***Struttura database dopo il popolamento e definizione chiavi.***

**Task 1:**

Per completare il primo *assignment,* era necessario dividere il file “*fact.csv*” in tre differenti file, che sono poi state denominate come le seguenti tabelle ovvero: *cpu\_sales, ram\_sales, gpu\_sales*, create per il primo *assignment*.

Il file *fact.csv* contiene una colonna per ognuno dei tre prodotti; sottolineiamo che per ogni riga solo una delle tre colonne ha valore non nullo.



Riguardo le righe, originariamente sono state disposte in un ordine tale per cui all’inizio troviamo “gpu”, terminati i record con “gpu”, susseguono quelli con “cpu”, “ram” e così fino alla fine.

Questo preciso ordine è stato sfruttato per la divisione delle righe dei tre csv descritti in precedenza, con una singola iterazione su tutte le righe del file, utilizzando la libreria *csv di python,* la quale ci ha permesso di scrivere i tre nuovi csv derivati.

**Task 2:**

Per il caricamento dei dati sul database vi era innanzitutto una richiesta particolare per quanto riguarda la tabella sul tempo. Abbiamo pertanto designato uno script specifico per la suddetta tabella, al fine di modificare il csv e aggiungere le colonne con i valori derivati per “**quarter**” e “**weekday**” .

A questo punto ci siamo dedicati alla richiesta principale del secondo assignment: caricare i dati nelle tabelle del database definito in precedenza. Per farlo è stato creato uno script unico che interasse sui nomi delle tabelle (*i file sono stati nominati allo stesso modo*) e poi su tutte le righe.

Abbiamo dovuto tener conto dei diversi tipi degli attributi e delle diverse codifiche richieste da ogni tabella al fine di generare delle query ben strutturate (con apici dove necessario, con parentesi e le virgole al posto giusto).

Per ogni riga letta dal documento è stata composta una query e lanciata. Il cursore viene anch’esso chiuso e aperto dopo ogni query lanciata.

All'apertura di un nuovo file viene lanciata una query al solo scopo di leggere le colonne. Per far questo bisogna tenere conto che non possono essere inseriti valori per le identità, quindi abbiamo escluso dalla lettura tali colonne.

La connessione viene invece aperta e chiusa sola una volta. Tuttavia abbiamo deciso di fare un commit dopo il completamento di ogni tabella.

A causa delle dimensione notevoli dei dati che riguardano le vendite, si è deciso di aggiungere un contatore per effettuare un commit ogni 1500 righe, così da tenere traccia delle righe inserite fino a quel punto ed eventualmente ricominciare da dove si era interrotto. Importante è stato caricare prima le altre tabelle e poi quelle con le *foreign keys*. In alternativa le *foreign key* possono essere aggiunte successivamente al caricamento dei dati