

LOST IN ASSETS

A DEEP LOOK INTO FINANCE

UN PROGETTO DI
MATTEO STROMIERI,
MATTEO CONCUTELLI E
LUCA ZHOU



SOMMARIO

| | |
|--|----|
| Intro al progetto | 3 |
| Descrizione | 3 |
| Tabella Termini..... | 4 |
| Tabella Entità..... | 7 |
| Tabella Relazioni..... | 9 |
| Schemi | 11 |
| Normalizzazione | 11 |
| Schema Fisico | 12 |
| Generalizzazione | 12 |
| Implementazione Database - MySQL | 13 |
| Creazione Tabelle | 13 |
| Inserimenti Manuali | 18 |
| INDICE AZIONARIO | 18 |
| SETTORE | 18 |
| VALUTA..... | 19 |
| NAZIONE..... | 20 |
| CRIPTOVALUTA | 20 |
| AZIENDA | 21 |
| OPERA..... | 22 |
| IMPIEGATO | 23 |
| BANCA..... | 23 |
| MERCATO DELLE COMMODITY..... | 24 |
| ASSET | 25 |
| TITOLO DI STATO | 28 |
| COMMODITY..... | 28 |
| CLASSIFICAZIONE AZIONE | 29 |
| AZIONE | 29 |
| OBBLIGAZIONE..... | 30 |
| STRUMENTO DERIVATO..... | 31 |
| BORSA..... | 31 |
| AZIENDA POSSIEDE CRIPTOVALUTA | 32 |
| IMPIEGATO POSSIEDE CRIPTOVALUTA | 33 |
| AZIENDA POSSIEDE ASSET | 34 |
| CONTIENE | 35 |
| AZIONE QUOTATA IN | 35 |
| OBBLIGAZIONE QUOTATA IN..... | 36 |

| | |
|---|-----|
| NAZIONE POSSIEDE ASSET..... | 37 |
| Query | 38 |
| MySQL..... | 38 |
| Algebra Relazionale | 64 |
| Calcolo Relazionale | 65 |
| Generazione casuale di tuple in Python: MySQL | 66 |
| Ottimizzazione | 75 |
| Trigger..... | 85 |
| Sicurezza | 88 |
| Views | 88 |
| Creazione Utenti:..... | 89 |
| Stored Procedures | 91 |
| MongoDB | 93 |
| Insert Mongo | 93 |
| Azienda..... | 93 |
| Azioni..... | 99 |
| Impiegato | 103 |
| Query..... | 105 |
| Azienda..... | 105 |
| Azioni..... | 106 |
| Impiegato | 108 |
| MongoDB vs MySQL..... | 112 |
| Impiegato | 112 |
| Azienda..... | 114 |
| Azione..... | 116 |

INTRO AL PROGETTO

La Banca d'Italia ha svolto due rilevazioni sull'Alfabetizzazione e le competenze finanziarie degli Italiani (IACOFI), nel 2017 e nel 2020, sulla base di una metodologia sviluppata dall'OCSE. Secondo tale approccio metodologico, il punteggio di alfabetizzazione finanziaria varia tra 1 e 21 ed è ottenuto come somma di tre componenti: conoscenze finanziarie (da 0 a 7), comportamenti finanziari (da 0 a 9) e attitudini finanziarie (da 1 a 5). I rispondenti che contribuiscono al punteggio OCSE hanno un'età compresa tra i 18 e i 79 anni; le interviste raccolte si riferiscono anche a individui dagli 80 anni in su.

Da queste indagini è facile osservare che il livello di educazione finanziaria in Italia è un grande punto debole dei risparmiatori, che si trovano da tempo agli ultimi posti fra i Paesi OCSE in termini di competenze. Nel 2020 il nostro Paese è 25-esimo sui 26 degli stati più avanzati al mondo.

Dunque, per lo svolgimento del progetto ci siamo posti in una situazione in cui sia necessario rappresentare quella che è la struttura del sistema economico – finanziario globale; ovvero quella in cui si debba produrre un database di questo tipo per una scuola superiore. Ovviamente, per rendere il lavoro più semplice è stato opportuno modellizzare appositamente la realtà tramite alcune semplificazioni per rendere i macro-aspetti chiari e semplici, mascherando i dettagli esoterici.

DESCRIZIONE

Ci è stato commissionato da un istituto tecnico ad indirizzo commerciale un database didattico in grado di raccogliere informazioni sulle maggiori transazioni, dati e agenti economici così che studenti e professori, dotati di un'opportuna applicazione con interfaccia grafica, possano fargli domande di natura economico - finanziaria.
In particolare, si vuole sapere, per le maggiori aziende internazionali, il settore in cui operano, le maggiori borse in cui è quotata, i nomi delle persone che ci lavorano e le azioni e/o obbligazioni emesse e/o acquistate. Si vuole sapere inoltre in quale paese opera e se ha emesso o detenuto una qualche tipologia di criptovaluta e il tipo di società.

Per quanto riguarda le azioni vogliamo sapere quale azienda le ha emesse e da quali sono detenute, i macro-attributi finanziari, se è sottostante a qualche strumento derivato (Se sì si vuole sapere quale) e le borse in cui è scambiata. Per ogni impiegato si vorrebbe sapere l'azienda per la quale lavora, il ruolo ricoperto. Si vuole inoltre sapere se qualche impiegato detiene delle criptovalute. Riguardo obbligazioni o titoli di stato si vuole sapere l'ente che li ha emessi e le nazioni che lo possiedono, oltre ai maggiori dati finanziari. Dunque, per i titoli di stato si vuole anche sapere la nazione emittente, per la quale è richiesto poter fare domande al database in base ai dati macroeconomici, quali borse vi operano, la valuta di riferimento, le obbligazioni emesse o le azioni detenute.

Inoltre, si vogliono rappresentare i principali mercati delle materie prime a livello mondiale e locale, per ognuno di essi vogliamo sapere le principali commodity scambiate con i relativi attributi finanziari principali, si vogliono inoltre specificare le varie localizzazioni dei vari mercati delle materie prime con relative quotazioni e i principali possessori. Per ogni azione inoltre vogliamo sapere se è contenuta in qualche indice azionario, in tal caso vogliamo sapere oltre a questa quali altre azioni sono contenute, la borsa di riferimento e le quotazioni con i maggiori dati finanziari.

Per ogni valuta si vuole tenere traccia dei listini di cambio rispetto all'euro. Si vuole anche tenere traccia dell'andamento della criptovaluta.

TABELLA TERMINI

| Entità | Descrizione | Sinonimo |
|---------------------------|--|--------------|
| Azienda | Organismo composto di persone e beni, diretto al raggiungimento di un fine economico, d'interesse sia pubblico sia privato. | Società |
| Asset | Entità materiale o immateriale suscettibile di valutazione economica per un certo soggetto. | |
| Settore | Suddivisione formale del sistema economico sulla base delle caratteristiche delle attività economiche. Un settore economico comprende e accoppa tutte le attività economiche con una medesima caratteristica comune. | |
| Borsa | Luogo virtuale dove vengono negoziati asset di vario genere attraverso diversi strumenti finanziari. | Borsa valori |
| Impiegato | Membro di un board amministrativo in un'azienda | Membro |
| Azione | Unità minima di partecipazione di un socio al capitale sociale di una "società per azioni". | Quota, stock |
| Obbligazione | Titolo che conferisce all'investitore che lo compra il diritto a ricevere, alla scadenza definita nel titolo, il rimborso della somma versata e una remunerazione a titolo di interesse (chiamata cedola). | |
| Strumento derivato | Titolo finanziario che deriva il proprio valore da un altro asset finanziario oppure da un indice (ad esempio, azioni, indici finanziari, valute, tassi d'interesse o anche materie prime), detto sottostante. | Derivato |
| Titolo di stato | Obbligazione emessa periodicamente dal Ministero dell'Economia e delle Finanze per conto dello Stato con lo scopo di finanziare (coprire) il proprio debito pubblico o direttamente il deficit pubblico. | |
| Nazione | Nazione in termini di finanza ed economia. | Paese |
| Valuta | Moneta di circolazione in un determinato luogo. | |

| | | |
|--------------------------------|--|-----------------|
| Paniere azionario | Un paniere è costituito da un insieme di attività non necessariamente omogenee tra loro. Esso può rappresentare il sottostante di uno o più strumenti derivati. | |
| Indice azionario | Sintetizza il valore di un paniere di azioni e le sue variazioni nel tempo. Il valore di un indice è dato dalla media ponderata dei prezzi dei titoli compresi nel portafoglio. | Indice di borsa |
| Banca | Istituto che compie operazioni monetarie e di credito, impiegando il capitale proprio e quello depositato dai clienti. | |
| Classificazione azione | <ol style="list-style-type: none"> 1. Azioni ordinarie 2. Azioni privilegiate 3. Azioni di risparmio 4. Azioni a voto limitato 5. Azioni correlate 6. Azioni postergate 7. Azioni di godimento 8. Azioni a favore dei prestatori di lavoro | |
| Criptovaluta | Valuta virtuale che costituisce una rappresentazione digitale di valore ed è utilizzata come mezzo di scambio o detenuta a scopo di investimento. Le criptovalute possono essere trasferite, conservate o negoziate elettronicamente. | |
| Commodity | Materie prime. | |
| Mercato delle commodity | Mercato delle materie prime. | |
| Utile | Risultato economico o guadagno ottenuto in un certo periodo. Per un'azienda è dato dai ricavi di vendita meno i costi. | |
| Fatturato | Indicatore economico che misura l'ammontare complessivo dei ricavi. Registrati durante l'esercizio contabile da un'impresa, e derivanti dalla vendita dei beni prodotti e/o da prestazione di servizi. | |
| Capitalizzazione | Valore di mercato di una società, calcolato moltiplicando il numero delle azioni per il prezzo corrente in borsa. | |
| Dividendo | Parte di utile che viene distribuito da una società ai suoi azionisti. | |

| | | |
|-------------------------------------|---|--|
| Tasso di povertà | Percentuale di persone che vivono in un'economia domestica il cui reddito disponibile è inferiore alla soglia di povertà. | |
| Tasso di occupazione | Il tasso di occupazione, in economia e statistica, è un indicatore statistico del mercato del lavoro che quantifica l'incidenza della popolazione che ha un'occupazione sul totale della popolazione e si calcola come rapporto percentuale tra il numero di persone occupate e la popolazione. | |
| Emitente | Ente che beneficia del finanziamento ricavato dalla vendita di una certa Obbligazione e/o Azione. | |
| Scadenza (Obbligazione) | La scadenza individua il termine del prestito obbligazionario, il momento in cui l'emittente dovrà restituire le somme ricevute. | |
| Interesse (Obbligazione) | L'interesse è il prezzo che l'investitore pretende per prestare il proprio denaro. | |
| Rendimento (Obbligazione) | Il rendimento dell'obbligazione, oltre che dall'interesse, è composto anche dal guadagno in conto capitale (capital gain), che si ha se il titolo è acquistato a un prezzo minore di quello a cui è rimborsato o venduto. | |
| Possessore di un asset | Persona o azienda che possiede un determinato asset finanziario in un certo istante. | |
| PIL (Prodotto Interno Lordo) | Somma dei beni e dei servizi finali prodotti da un paese in un dato periodo di tempo. | |
| PIL pro-capite | Rapporto tra il PIL espresso ai prezzi di mercato alla popolazione residente nella regione. | |
| Ticker | Codice utilizzato per identificare in modo univoco le azioni di un determinata azienda quotata in borsa. | |
| Flottante | Per un generico asset, rappresenta la quantità in circolazione. | |

TABELLA ENTITÀ

| Nome Entità | Descrizione | Attributi | Relazioni Coinvolte |
|-------------------------------|---|--|--|
| Azienda | Organismo composto di persone e beni, diretto al raggiungimento di un fine economico o d'interesse pubblico. | ID , nome, utile, fatturato, numeroDipendenti | Opera, EmetteAzione, AziendaPossiedeCriptovaluta, EmetteObbligazione, Lavora, EmetteStrumentoDerivato, AziendaRisiede, AziendaPossiedeAsset |
| Banca | Istituto che compie operazioni monetarie e di credito, impiegando il capitale proprio e quello depositato dai clienti. | ID , nome, fatturato, capitalizzazione, numeroDipendenti | BancaRisiede |
| Borsa | Luogo virtuale dove vengono negoziati asset di vario genere attraverso diversi strumenti finanziari. | nome , volumeDiScambi | AzioneQuotataIn, ObbligazioneQuotataIn, BorsaRisiede |
| Classificazione Azione | Tipologia di azione | tipologia , nominatività, dirittoDiVotoOrdinario, dirittoDiVotoStraordinario, dirittoAlDividendo, ripartizioneUtili, maggiorazionePercentuale | Tipologia |
| Criptovaluta | Valuta virtuale che costituisce una rappresentazione digitale di valore ed è utilizzata come mezzo di scambio o detenuta a scopo di investimento. | ticker , flottante | AziendaPossiedeCriptovaluta, ImpiegatoPossiedeCriptovaluta |
| Indice Azionario | Sintetizza il valore di un paniere di azioni e le sue variazioni nel tempo | ticker , valoreNominale, prezzoEmissione | Contiene |

| | | | |
|--------------------------------|---|--|---|
| Mercato delle Commodity | Mercato delle materie prime | nome , volumeDiScambi | MercatoRisiede, Quotataln |
| Nazione | Nazione in termini di finanza ed economia | nome , popolazione, pil, tassoPoverta, tassoOccupazione | AziendaRisiede, ImpiegatoRisiede, BorsaRisiede, BancaRisiede, EmetteValuta, NazionePossiedeAsset |
| Impiegato | Persona che presta continuativamente la propria attività professionale, esclusa quella di semplice mano d'opera, alle dipendenze di un ente pubblico o privato | CF, nazione , nome, cognome | ImpiegatoPossiedeCriptovaluta, ImpiegatoRisiede, Lavora |
| Settore | Tipologia di ruolo che svolte una azienda | nome , ricavatoComplessivo | Opera |
| Valuta | Moneta di circolazione in un determinato luogo | codice_ISO , nome, circolante, tassoDiCambioEuro | EmetteValuta |
| Azione | Unità minima di partecipazione di un socio al capitale sociale di una "società per azioni" | ticker | EmetteAzione, Tipologia, Sovrastante, Contiene, AzioneQuotataln |
| Commodity | Materie prime | ticker , volumeTotale | Quotataln |
| Obbligazione | Titolo che conferisce all'investitore che lo compra il diritto a ricevere, alla scadenza definita nel titolo, il rimborso della somma versata e una remunerazione a titolo di interesse | ticker , scadenza, interesse, rendimento | ObbligazioneQuotataln, EmetteObbligazione |
| Strumento Derivato | Titolo finanziario che deriva il proprio valore da un altro asset finanziario oppure da un indice | ticker , aziendaEmittente, azione | EmetteStrumentoDerivato |
| Titolo di Stato | Obbligazione emessa periodicamente dal | ticker , scadenza, interesse, rendimento | EmetteTitoloDiStato |

| | | | |
|--------------|--|--|---|
| | Ministero dell'Economia e delle Finanze per conto dello Stato | | |
| Asset | Entità materiale o immateriale suscettibile di valutazione economica per un certo soggetto | ticker , valoreNominale, prezzoEmissione, flottante | AziendaPossiedeAsset, NazionePossiedeAsset |

TABELLA RELAZIONI

| Relazione | Descrizione | Entità |
|----------------------------------|--|--|
| Contiene | Ogni indice contiene un certo numero di azioni | Indice Azionario(1,n), Azione(o,n) |
| Azione Quotata In | La borsa è il luogo dove le azioni vengono scambiate, su una borsa vengono scambiate più azioni e una azione viene scambiata in almeno una borsa | Azione(1,n), Borsa(1,n) |
| Obbligazione Quotata In | La borsa è il luogo dove le obbligazioni vengono scambiate, su una borsa vengono scambiate più obbligazioni e una obbligazione viene scambiata in almeno una borsa | Obbligazione(1,n), Borsa(1,n) |
| Tipologia | Ogni azione è di un certo tipo che le conferisce alcune proprietà | Azione(1,1), Classificazione Azione(1,n) |
| Sovrastante | Ogni strumento derivato è sottostante ad una azione che ne influenza definitivamente il valore di mercato | Azione(1,n), Strumento Derivato(1,1) |
| Quotata In | Ogni commodity/materia prima è venduta in un Mercato delle Commodity | Commodity(1,1), Mercato delle Commodity(1,n) |
| Azienda Possiede Asset | Una azienda può possedere un qualsiasi asset | Azienda(o,n), Azienda Possiede Asset(o,n) |
| Emette Azione | Alcune aziende hanno il diritto di emettere azioni, dunque questa relazione associa ad ogni azione la sua azienda emittente | Azione(1,1), Azienda(o,n) |
| Emette obbligazione | Alcune aziende hanno il diritto di emettere obbligazioni, dunque questa relazione associa ad ogni obbligazione la sua azienda emittente | Azienda(o,n), Obbligazione(1,1) |
| Emette Strumento Derivato | Alcune aziende hanno il diritto di emettere strumenti derivati, dunque questa relazione associa ad ogni strumento derivato la sua azienda emittente | Strumento Derivato(1,1), Azienda(o,n) |

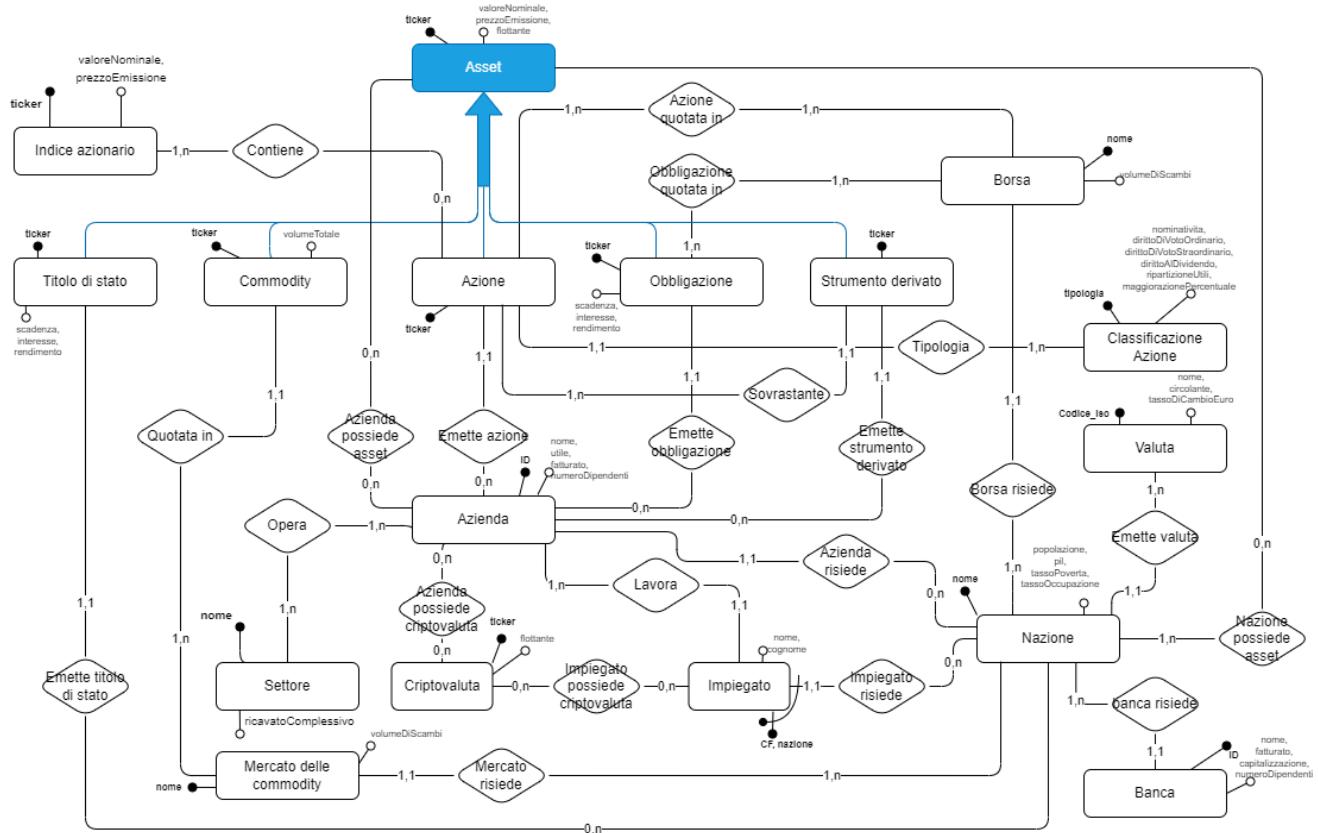
| | | |
|--|---|--|
| Opera | Ogni azienda opera in almeno un settore, per esempio la azienda 'Amazon' opera nel settore eCommerce | Settore(1,n), Azienda(1,n) |
| Borsa Risiede | Ogni borsa risiede legalmente in un certo paese, per esempio la Borsa di Milano è, ovviamente, Italiana. | Borsa(1,1), Nazione(1,n) |
| Emette Valuta | Ogni nazione è associata alla sua Banca Centrale che emette una certa valuta. | Valuta(1,n), Nazione(1,1) |
| Azienda Risiede | Ogni azienda risiede legalmente in uno e un solo paese | Azienda(1,1), Nazione(o,n) |
| Azienda Possiede Criptovaluta | Una azienda può possedere una qualsiasi criptovaluta | Azienda(o,n), Criptovaluta(o,n) |
| Lavora | Ogni impiegato rappresentato nel database in questione lavora in una azienda presente nella entità del database | Azienda(1,n), Impiegato(1,1) |
| Emette Titolo di Stato | Ogni nazione può emettere dei titoli di stato per finanziare le proprie politiche | Titolo di Stato(1,1), Nazione(o,n) |
| Impiegato Possiede Criptovaluta | Ogni impiegato ha il diritto ma non il dovere di possedere delle criptovaluta | Impiegato(o,n), Criptovaluta(o,n) |
| Impiegato Risiede | Ogni impiegato risiede in una e una sola nazione(Semplificazione per rendere lo schema fisico di migliore comprensione) | Impiegato(1,1), Nazione(o,n) |
| Nazione Possiede Asset | Ogni nazione può possedere Asset emessi da qualsiasi nazione o azienda | Nazione(o,n), Asset(o,n) |
| Mercato Risiede | Ogni mercato delle Commodity ovviamente risiede legalmente in una e una sola nazione | Mercato delle Commodity(1,1), Nazione(1,n) |
| Banca Risiede | Ogni banca ovviamente risiede legalmente in una e una sola nazione | Nazione(1,n), Banca(1,1) |

SCHEMI

Modello entità-relazione è un modello teorico per la rappresentazione concettuale e grafica dei dati a un alto livello di astrazione. Uno schema E/R ben fatto deve avere quattro caratteristiche:

- **Completezza**
- **Correttezza**
- **Leggibilità**
- **Minimalità e non-ridondanza**

Il seguente è lo schema E/R logico del database che stiamo progettando, esso conta più di 20 entità e relazioni.

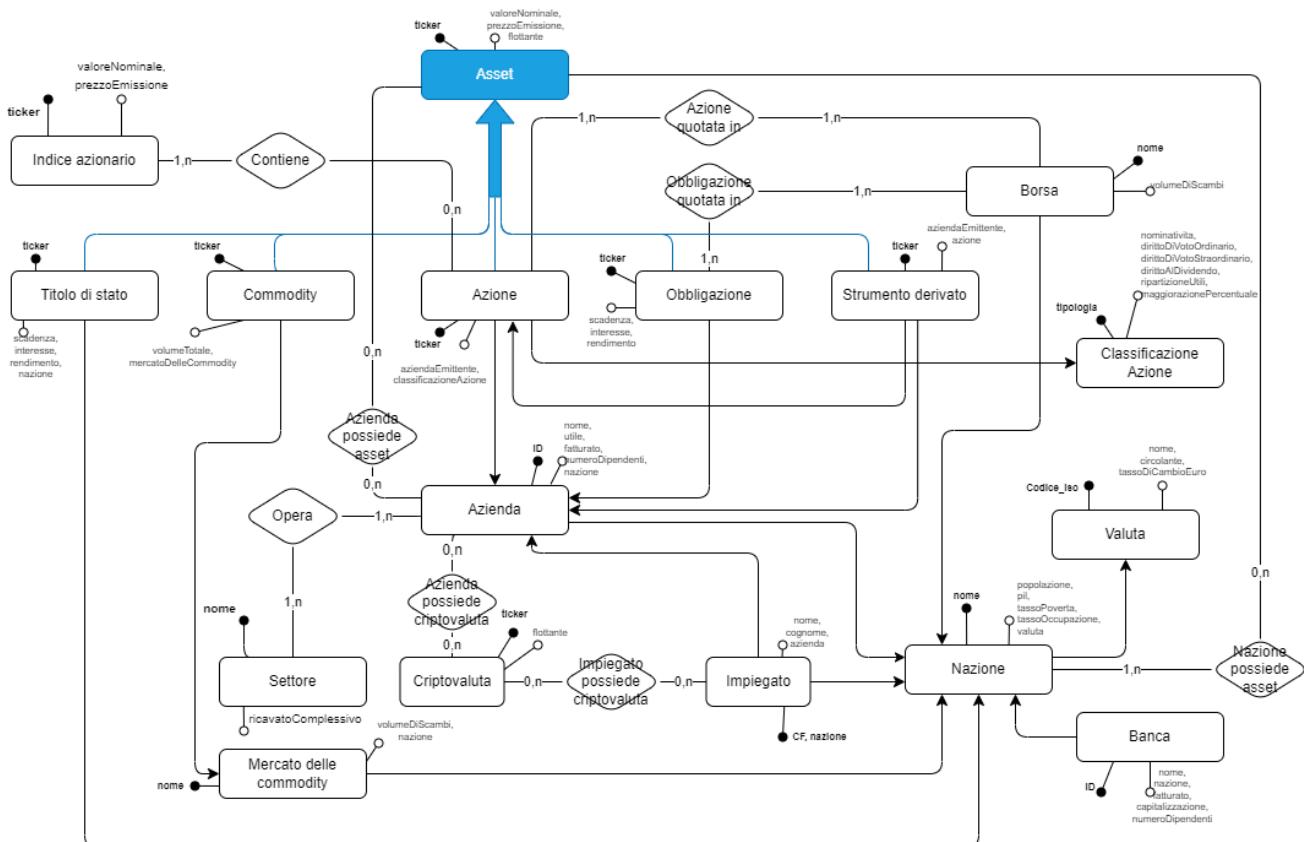


NORMALIZZAZIONE

De seguito si discutono le forme normali dello schema logico:

- **1NF**: tutti gli schemi di relazione nello schema logico sopra riportato sono in 1NF, poiché tutti gli attributi sono semplici, ovvero contengono soltanto valori atomici indivisibili.
- **2NF**: tutti gli schemi di relazione dello schema logico sono anche già in 2NF, poiché sono già in 1NF e nessun attributo presenta alcuna dipendenza parziale. Tutti gli attributi dipendono funzionalmente solo dalla chiave primaria della stessa tabella.
- **3NF**: tutti gli schemi di relazione sono anche in 3NF perché già in 2NF, ed inoltre, tutti gli attributi delle tabelle dipendono funzionalmente e direttamente dalla chiave primaria, senza transitività.

SCHEMA FISICO



GENERALIZZAZIONE

Una generalizzazione rappresenta un legame logico tra un'entità genitore e una o più entità figlie, in questo caso l'entità genitore è "Asset" e le entità figlie sono: "Titolo di stato", "Commodity", "Azione", "Obbligazione", "Strumento derivato". Abbiamo tre metodi per rappresentare una generalizzazione a livello fisico:

- Accorpamento del padre nelle entità figlie
- Accorpamento delle entità figlie nel padre
- Sostituzione della generalizzazione con relazioni

Tra questi metodi abbiamo scelto il terzo in quanto da noi considerato il più adeguato. Infatti, il primo metodo avrebbe portato ad una ridondanza di relazioni del tipo: "Azienda possiede" e "Nazione possiede" ("Azienda possiede obbligazione/azione/titolo di stato..."), "Nazione possiede obbligazione/azione/titolo di stato..."). Il secondo metodo necessita dell'aggiunta di un attributo nell'entità "Asset", con il compito di specificare il tipo di asset (Obbligazione o Azione o Titolo di stato...), in più si sarebbe dovuto scegliere se perdere informazioni (attributi) dei figli o inserire le informazioni nel padre, quindi aggiungere attributi dei figli al padre. La seconda scelta avrebbe portato ad una quantità non indifferente di valori NULL. Inoltre, c'è anche la questione delle relazioni delle entità figlie che devono essere collegate al padre, così facendo sicuramente ci sarà cardinalità (0,1) o (0,n) verso l'entità collegata, e quindi l'impossibilità di, eventualmente, normalizzare la relazione.

IMPLEMENTAZIONE DATABASE - MYSQL

CREAZIONE TABELLE

```
CREATE TABLE IndiceAzionario (
    ticker varchar(20) NOT NULL,
    valoreNominali decimal(10,2) NOT NULL,
    prezzoEmissione decimal(10,2) NOT NULL,
    PRIMARY KEY(ticker)
);

CREATE TABLE Settore (
    nome varchar(20) NOT NULL,
    ricavatoComplessivo decimal(20,2) NOT NULL,
    PRIMARY KEY(nome)
);

CREATE TABLE Valuta (
    Codice_ISO varchar(5) NOT NULL,
    nome varchar(30) NOT NULL,
    circolante decimal(20,2) NOT NULL,
    tassoDiCambioEuro decimal(20,2) NOT NULL,
    PRIMARY KEY(Codice_ISO)
);

CREATE TABLE Nazione (
    nome varchar(25) NOT NULL,
    popolazione bigint NOT NULL,
    pil bigint NOT NULL,
    tassoPoverta decimal(5,2) NOT NULL,
    tassoOccupazione decimal(5,2) NOT NULL,
    valuta varchar(20) NOT NULL,
    PRIMARY KEY(nome),
    FOREIGN KEY(valuta) REFERENCES Valuta(Codice_ISO)
);

CREATE TABLE Criptovaluta (
    ticker varchar(20) NOT NULL,
    flottante bigint NOT NULL,
    PRIMARY KEY(ticker)
);

CREATE TABLE Azienda (
    ID int NOT NULL,
    nome varchar(20) NOT NULL,
    nazione varchar(20) NOT NULL,
    utile decimal(20,2) NOT NULL,
```

```

fatturato decimal(20,2) NOT NULL,
numeroDipendenti int NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY(nazione) REFERENCES Nazione(nome)
);

CREATE TABLE Opera (
    azienda int NOT NULL,
    settore varchar(20) NOT NULL,
    PRIMARY KEY(azienda,settore),
    FOREIGN KEY(azienda) REFERENCES Azienda(ID),
    FOREIGN KEY(settore) REFERENCES Settore(nome)
);

CREATE TABLE Impiegato (
    CF varchar(20) NOT NULL,
    nazione varchar(20) NOT NULL,
    nome varchar(20) NOT NULL,
    cognome varchar(20) NOT NULL,
    azienda int NOT NULL,
    PRIMARY KEY(CF, nazione),
    FOREIGN KEY(nazione) REFERENCES Nazione(nome),
    FOREIGN KEY(azienda) REFERENCES Azienda(ID)
);

CREATE TABLE Banca (
    ID int NOT NULL,
    nome varchar(100) NOT NULL,
    nazione varchar(20) NOT NULL,
    fatturato int NOT NULL,
    capitalizzazione int NOT NULL,
    numeroDipendenti int NOT NULL,
    PRIMARY KEY (ID),
    FOREIGN KEY(nazione) REFERENCES Nazione(nome)
);

CREATE TABLE MercatoDelleCommodity (
    nome varchar(60) NOT NULL,
    volumeDiScambi int NOT NULL,
    nazione varchar(20) NOT NULL,
    PRIMARY KEY(nome),
    FOREIGN KEY(nazione) REFERENCES Nazione(nome)
);

CREATE TABLE Asset (
    ticker varchar(20) NOT NULL,
    valoreNominali decimal(7,2) NOT NULL,
    prezzoEmissione decimal(7,2) NOT NULL,
    flottante decimal(30,2) NOT NULL,
    PRIMARY KEY(ticker)
);

```

```

);

CREATE TABLE TitoloDiStato (
    ticker varchar(20) NOT NULL,
    scadenza date NOT NULL,
    nazione varchar(20) NOT NULL,
    interesse decimal(5,2) NOT NULL,
    rendimento decimal(5,2) NOT NULL,
    PRIMARY KEY(ticker),
    FOREIGN KEY(nazione) REFERENCES Nazione(nome),
    FOREIGN KEY(ticker) REFERENCES Asset(ticker)
);

CREATE TABLE Commodity (
    ticker varchar(20) NOT NULL,
    volumeTotale int NOT NULL,
    mercatoDelleCommodity varchar(60) NOT NULL,
    PRIMARY KEY(ticker),
    FOREIGN KEY(ticker) REFERENCES Asset(ticker),
    FOREIGN KEY(mercatoDelleCommodity) REFERENCES MercatoDelleCommodity (nome)
);

CREATE TABLE ClassificazioneAzione (
    tipologia varchar(20) NOT NULL,
    nominativita ENUM('True','False') NOT NULL,
    dirittoDiVotoOrdinario ENUM('True','False') NOT NULL,
    dirittoDiVotoStraordinario ENUM('True','False') NOT NULL,
    dirittoAlDividendo ENUM('True','False') NOT NULL,
    ripartizioneUtili ENUM('True','False') NOT NULL,
    maggiorazionePercentuale ENUM('True','False') NOT NULL,
    PRIMARY KEY(tipologia)
);

CREATE TABLE Azione (
    ticker varchar(20) NOT NULL,
    aziendaEmittente int NOT NULL,
    classificazioneAzione varchar(20) NOT NULL,
    PRIMARY KEY(ticker),
    FOREIGN KEY(ticker) REFERENCES Asset(ticker),
    FOREIGN KEY(aziendaEmittente) REFERENCES Azienda(ID),
    FOREIGN KEY(classificazioneAzione) REFERENCES ClassificazioneAzione(tipologia)
);

CREATE TABLE Obbligazione (
    ticker varchar(20) NOT NULL,
    azienda int NOT NULL,
    scadenza date NOT NULL,
    interesse decimal(5,2) NOT NULL,
    rendimento decimal(5,2) NOT NULL,
    PRIMARY KEY(ticker),

```

```

FOREIGN KEY(ticker) REFERENCES Asset(ticker),
FOREIGN KEY(azienda) REFERENCES Azienda(ID)
);

CREATE TABLE StrumentoDerivato (
    ticker varchar(20) NOT NULL,
    aziendaEmittente int NOT NULL,
    azione varchar(20) NOT NULL,
    PRIMARY KEY(ticker),
    FOREIGN KEY(ticker) REFERENCES Asset(ticker),
    FOREIGN KEY(aziendaEmittente) REFERENCES Azienda(ID),
    FOREIGN KEY(azione) REFERENCES Azione(ticker)
);

CREATE TABLE Borsa (
    nome varchar(60) NOT NULL,
    volumeDiScambi int NOT NULL,
    nazione varchar(20) NOT NULL,
    PRIMARY KEY (nome),
    FOREIGN KEY(nazione) REFERENCES Nazione(nome)
);

CREATE TABLE AziendaPossiedeCriptovaluta (
    ID int NOT NULL,
    ticker varchar(20) NOT NULL,
    PRIMARY KEY (ID,ticker),
    FOREIGN KEY(ID) REFERENCES Azienda(ID),
    FOREIGN KEY(ticker) REFERENCES Criptovaluta(ticker)
);

CREATE TABLE ImpiegatoPossiedeCriptovaluta (
    CF varchar(20) NOT NULL,
    nazione varchar(20) NOT NULL,
    ticker varchar(20) NOT NULL,
    PRIMARY KEY (CF,nazione,ticker),
    FOREIGN KEY(CF) REFERENCES Impiegato(CF),
    FOREIGN KEY(nazione) REFERENCES Impiegato(nazione),
    FOREIGN KEY(ticker) REFERENCES Criptovaluta(ticker)
);

CREATE TABLE AziendaPossiedeAsset (
    ID int NOT NULL,
    ticker varchar(20) NOT NULL,
    PRIMARY KEY (ID, ticker),
    FOREIGN KEY(ticker) REFERENCES Asset(ticker),
    FOREIGN KEY(ID) REFERENCES Azienda(ID)
);

CREATE TABLE Contiene (
    tickerIndice varchar(20) NOT NULL,

```

```
tickerAzione varchar(20) NOT NULL,  
PRIMARY KEY (tickerIndice,tickerAzione),  
FOREIGN KEY(tickerIndice) REFERENCES IndiceAzionario(ticker),  
FOREIGN KEY(tickerAzione) REFERENCES Azione(ticker)  
);  
  
CREATE TABLE AzioneQuotataIn (  
    ticker varchar(20) NOT NULL,  
    borsa varchar(60) NOT NULL,  
    PRIMARY KEY (ticker,borsa),  
    FOREIGN KEY(ticker) REFERENCES Azione(ticker),  
    FOREIGN KEY(borsa) REFERENCES Borsa(nome)  
);  
  
CREATE TABLE ObbligazioneQuotataIn (  
    ticker varchar(20) NOT NULL,  
    borsa varchar(60) NOT NULL,  
    PRIMARY KEY(ticker,borsa),  
    FOREIGN KEY(ticker) REFERENCES Obbligazione(ticker),  
    FOREIGN KEY(borsa) REFERENCES Borsa(nome)  
);  
  
CREATE TABLE NazionePossiedeAsset (  
    nazione varchar(20) NOT NULL,  
    ticker varchar(20) NOT NULL,  
    PRIMARY KEY (nazione,ticker),  
    FOREIGN KEY(nazione) REFERENCES Nazione(nome),  
    FOREIGN KEY(ticker) REFERENCES Asset(ticker)  
);
```

INSERIMENTI MANUALI

INDICE AZIONARIO

```
INSERT INTO IndiceAzione VALUES('SP500',10,12);
INSERT INTO IndiceAzione VALUES('DJI',2,5);
INSERT INTO IndiceAzione VALUES('NAC',3,6);
INSERT INTO IndiceAzione VALUES('TASK',5,6);
INSERT INTO IndiceAzione VALUES('PAC',6,8);
INSERT INTO IndiceAzione VALUES('STOF',1,3);
INSERT INTO IndiceAzione VALUES('LOAD',8,11);
INSERT INTO IndiceAzione VALUES('KAYT',10,12);
INSERT INTO IndiceAzione VALUES('KATY',17,18);
INSERT INTO IndiceAzione VALUES('FOAM',11,15);
INSERT INTO IndiceAzione VALUES('LOTA',13,19);
INSERT INTO IndiceAzione VALUES('DATTO',12,12);
INSERT INTO IndiceAzione VALUES('TORA',8,12);
INSERT INTO IndiceAzione VALUES('TTORA',3,10);
INSERT INTO IndiceAzione VALUES('KAPPA',7,8);
INSERT INTO IndiceAzione VALUES('VAMO',11,16);
INSERT INTO IndiceAzione VALUES('ESPN',15,19);
INSERT INTO IndiceAzione VALUES('COCA',11,17);
INSERT INTO IndiceAzione VALUES('COLA',10,12);
INSERT INTO IndiceAzione VALUES('ELLE',6,12);
INSERT INTO IndiceAzione VALUES('HAP',7,14);
INSERT INTO IndiceAzione VALUES('SAMS',8,16);
INSERT INTO IndiceAzione VALUES('WAOM',9,18);
INSERT INTO IndiceAzione VALUES('MARIO',4,8);
INSERT INTO IndiceAzione VALUES('GATTO',8,9);
INSERT INTO IndiceAzione VALUES('RACE',14,18);
INSERT INTO IndiceAzione VALUES('FAP',13,16);
INSERT INTO IndiceAzione VALUES('MICRO',18,19);
```

SETTORE

```
INSERT INTO Settore VALUES ('agriculture', 870265234984.65);
INSERT INTO Settore VALUES ('automotive',450265234984.53);
INSERT INTO Settore VALUES ('technology',670265234984.19);
INSERT INTO Settore VALUES ('petrol',980265234984.34);
INSERT INTO Settore VALUES ('renewable energy',210265234984.74);
INSERT INTO Settore VALUES ('building',600265234984.52);
INSERT INTO Settore VALUES ('livestock',450265234984.41);
INSERT INTO Settore VALUES ('medicine',150265234984.98);
INSERT INTO Settore VALUES ('science',110265234984.18);
INSERT INTO Settore VALUES ('catering',2190265234984.72);
INSERT INTO Settore VALUES ('food',2990265234984.29);
INSERT INTO Settore VALUES ('weapon',1090265234984.48);
INSERT INTO Settore VALUES ('astronomy',590265234984.35);
INSERT INTO Settore VALUES ('education',1990265234984.11);
```

```
INSERT INTO Settore VALUES ('software',290265234984.30);
INSERT INTO Settore VALUES ('biology',230265234984.73);
INSERT INTO Settore VALUES ('tourism',270265234984.66);
INSERT INTO Settore VALUES ('material',370265234984.50);
INSERT INTO Settore VALUES ('e-commerce',770265234984.17);
INSERT INTO Settore VALUES ('entertainment',475865234484.63);
INSERT INTO Settore VALUES ('clothing',735865234484.63);
```

VALUTA

```
INSERT INTO Valuta VALUES ('AFN','Afghani', 1774473417, 94.66);
INSERT INTO Valuta VALUES ('ALL','Albanian Lek', 1154618276, 119.79);
INSERT INTO Valuta VALUES ('DZD','Algerian Dinaro', 685057811, 154.04);
INSERT INTO Valuta VALUES ('EUR','Euro', 51541722, 1);
INSERT INTO Valuta VALUES ('AOA','Angolian Kwanza', 956072931, 452.91);
INSERT INTO Valuta VALUES ('XCD','Oriental Caribbean Dollar', 99500, 2.86);
INSERT INTO Valuta VALUES ('SAR','Saudi Riyal', 179845370, 3.98);
INSERT INTO Valuta VALUES ('ARS','Argentinian Peso', 501254302, 132.14 );
INSERT INTO Valuta VALUES ('AMD','Armenian Dram', 23958439, 434.16);
INSERT INTO Valuta VALUES ('AUD','Australian Dollar',94441814, 1.53);
INSERT INTO Valuta VALUES ('AZN','Azerbaijan Manat',74385157, 1.80);
INSERT INTO Valuta VALUES ('BSD','Bahamas Dollar',2694949, 1.06);
INSERT INTO Valuta VALUES ('BHD','Bahrein Dinaro', 15897483, 0.40);
INSERT INTO Valuta VALUES ('BDT','Bengal Taka', 1488094342, 98.54);
INSERT INTO Valuta VALUES ('BBD','Barbados Dollar',2090362, 2.14);
INSERT INTO Valuta VALUES ('BZD','Belize Dinaro', 2982606, 2.14);
INSERT INTO Valuta VALUES ('BYN','Belarusian Rublo', 51008246, 3.58);
INSERT INTO Valuta VALUES ('MMK','Birmanian Kyat', 216969025, 1965.40);
INSERT INTO Valuta VALUES ('BOB','Boliviano', 90005002, 7.30);
INSERT INTO Valuta VALUES ('BAM','Bosnian Mark', 4829357, 1.96);
INSERT INTO Valuta VALUES ('BRL','Brazilian Real', 540930421, 5.53);
INSERT INTO Valuta VALUES ('BGN','Bulgarian Lev',50480155, 1.96);
INSERT INTO Valuta VALUES ('HUF','Hungarian Forint',43571752, 403.64);
INSERT INTO Valuta VALUES ('JPY','Yen',93782069, 143.43);
INSERT INTO Valuta VALUES ('CHF','Swiss franc',16677508, 1.01);
INSERT INTO Valuta VALUES ('PLN','Polish zloty',224612607, 4.69);
INSERT INTO Valuta VALUES ('CAD','Canadian dollar', 38015623, 1.37);
INSERT INTO Valuta VALUES ('GBP','British pound', 67229874, 0.86);
INSERT INTO Valuta VALUES ('RON','Romanian leu',35086216, 4.94);
INSERT INTO Valuta VALUES ('RUB','Russian ruble', 1171108874, 56.10);
INSERT INTO Valuta VALUES ('CNY','Yuan', 1402456022, 7.09 );
INSERT INTO Valuta VALUES ('USD','U.S. dollar', 31233756, 1.06);
INSERT INTO Valuta VALUES ('KWD','Kuwaiti dinar', 4271741, 1.06);
INSERT INTO Valuta VALUES ('DKK','Danish krone', 5831414, 7.44);
INSERT INTO Valuta VALUES ('TWD','New Taiwan dollar', 23570551, 31.38);
INSERT INTO Valuta VALUES ('IRR','Riyal iraniano', 83995338, 44806.09);
INSERT INTO Valuta VALUES ('ZAR','South African Rand', 59315864, 16.78);
INSERT INTO Valuta VALUES ('EGP','Egyptian pound', 102306583, 19.87);
INSERT INTO Valuta VALUES ('KRW','South Korean Won', 302456022, 1358.6);
```

NAZIONE

```
INSERT INTO Nazione VALUES ('Italy', 59550000, 8000, 9.1, 73.5, 'EUR');
INSERT INTO Nazione VALUES ('Afghanistan', 38930000, 210, 11, 53.9, 'AFN');
INSERT INTO Nazione VALUES ('Albany', 2838000, 14, 13, 75.9, 'ALL');
INSERT INTO Nazione VALUES ('Algeria', 43850000, 145, 21, 68.2, 'DZD');
INSERT INTO Nazione VALUES ('Andorra', 77265, 3, 12, 43.6, 'EUR');
INSERT INTO Nazione VALUES ('Angola', 32870000, 62, 28, 66.4, 'AOA');
INSERT INTO Nazione VALUES ('Saudi Arabia', 34810000, 700, 38.9, 88.5, 'SAR');
INSERT INTO Nazione VALUES ('Argentine', 45380000, 383, 12.8, 68.3, 'ARS');
INSERT INTO Nazione VALUES ('Armenia', 2963000, 12, 17.1, 71.2, 'AMD');
INSERT INTO Nazione VALUES ('Australia', 25690000, 1300, 7.8, 90.4, 'AUD');
INSERT INTO Nazione VALUES ('Azerbaijan', 10110000, 42, 14, 77.3, 'AZN');
INSERT INTO Nazione VALUES ('Bahamas', 393248, 11, 5.9, 74.1, 'BSD');
INSERT INTO Nazione VALUES ('Bahrein', 1702000, 2, 68.9, 80.1, 'BHD');
INSERT INTO Nazione VALUES ('Bangladesh', 164700000, 324, 28.3, 69.96, 'BDT');
INSERT INTO Nazione VALUES ('Barbados', 287371, 40, 21.9, 77.1, 'BBD');
INSERT INTO Nazione VALUES ('Belize', 397621, 12, 1.2, 55.1, 'BZD');
INSERT INTO Nazione VALUES ('Bielorussia', 9399000, 56, 27.9, 45.8, 'BYN');
INSERT INTO Nazione VALUES ('Birmania', 54410000, 12, 19.3, 78.78, 'MMK');
INSERT INTO Nazione VALUES ('Bolivia', 11670000, 4, 7.1, 66.4, 'BOB');
INSERT INTO Nazione VALUES ('Bosnia and Erzegovina', 3281000, 5, 12.9, 91.3, 'BAM');
INSERT INTO Nazione VALUES ('Brazil', 212600000, 1400, 3.4, 78.4, 'BRL');
INSERT INTO Nazione VALUES ('Bulgary', 6927000, 120, 8.2, 78.1, 'BGN');
INSERT INTO Nazione VALUES ('Hungary', 9750000, 80, 17.9, 67.9, 'HUF');
INSERT INTO Nazione VALUES ('Japan', 125800000, 5000, 12.0, 88.4, 'JPY');
INSERT INTO Nazione VALUES ('Switzerland', 8637000, 748, 8.5, 96.9, 'CHF');
INSERT INTO Nazione VALUES ('Poland', 37950000, 500, 9.9, 91.2, 'PLN');
INSERT INTO Nazione VALUES ('Canada', 38010000, 1821.44, 9.4, 70.0, 'CAD');
INSERT INTO Nazione VALUES ('UK', 67220000, 27000, 3, 68.9, 'GBP');
INSERT INTO Nazione VALUES ('Romania', 19290000, 200, 19.6, 34.6, 'RON');
INSERT INTO Nazione VALUES ('Russia', 144100000, 1400, 12.6, 100.00, 'RUB');
INSERT INTO Nazione VALUES ('China', 1402000000, 14720, 0.3, 87.0, 'CNY');
INSERT INTO Nazione VALUES ('USA', 329500000, 20940, 15.1, 82.0, 'USD');
INSERT INTO Nazione VALUES ('Germany', 83240000, 3800, 3.5, 88.1, 'EUR');
INSERT INTO Nazione VALUES ('Kuwait', 4271000, 340, 28.9, 52.3, 'KWD');
INSERT INTO Nazione VALUES ('Spain', 47350000, 2000, 12.9, 71.5, 'EUR');
INSERT INTO Nazione VALUES ('Netherlands', 17440000, 1500, 3.5, 84.1, 'EUR');
INSERT INTO Nazione VALUES ('France', 67390000, 2600, 9.7, 87.99, 'EUR');
INSERT INTO Nazione VALUES ('Denmark', 5831000, 355.2, 16.3, 74.0, 'DKK');
INSERT INTO Nazione VALUES ('Taiwan', 23570000, 0.759, 1.5, 96.32, 'TWD');
INSERT INTO Nazione VALUES ('Iran', 83990000, 191.7, 18.7, 75.0, 'IRR');
INSERT INTO Nazione VALUES ('South Africa', 59310000, 301.9, 47.0, 65.5, 'ZAR');
INSERT INTO Nazione VALUES ('South Korea', 51780000, 4300, 37.0, 55.5, 'KRW');
INSERT INTO Nazione VALUES ('Egypt', 102300000, 363.1, 27.8, 91.9, 'EGP');
```

CRIPTOVALUTA

```
INSERT INTO Criptovaluta VALUES ('BTC', 19079000);
INSERT INTO Criptovaluta VALUES ('ETH', 121321000);
```

```

INSERT INTO Criptovaluta VALUES ('USDT', 668670000000);
INSERT INTO Criptovaluta VALUES ('USDC', 558030000000);
INSERT INTO Criptovaluta VALUES ('BNB', 163277000);
INSERT INTO Criptovaluta VALUES ('BUSD', 173890000000);
INSERT INTO Criptovaluta VALUES ('XRP', 483430000000);
INSERT INTO Criptovaluta VALUES ('ADA', 337390000000);
INSERT INTO Criptovaluta VALUES ('SOL', 342768000);
INSERT INTO Criptovaluta VALUES ('DOGE', 1326710000000);
INSERT INTO Criptovaluta VALUES ('DOT', 987579000);
INSERT INTO Criptovaluta VALUES ('ALT', 4638829101);
INSERT INTO Criptovaluta VALUES ('FAT', 463281003);
INSERT INTO Criptovaluta VALUES ('GUS', 56292073900);
INSERT INTO Criptovaluta VALUES ('LAT', 103847294739);
INSERT INTO Criptovaluta VALUES ('KNT', 0009865433578);
INSERT INTO Criptovaluta VALUES ('NNN', 15402866391);
INSERT INTO Criptovaluta VALUES ('3RAT', 249274017552);
INSERT INTO Criptovaluta VALUES ('FAG', 971648950000);
INSERT INTO Criptovaluta VALUES ('GET', 7200373562);
INSERT INTO Criptovaluta VALUES ('DOPE', 0192745820);
INSERT INTO Criptovaluta VALUES ('DAT', 0183746272991);
INSERT INTO Criptovaluta VALUES ('AGE', 9009865433578);
INSERT INTO Criptovaluta VALUES ('QWE', 95402866391);
INSERT INTO Criptovaluta VALUES ('RTY', 949274017552);
INSERT INTO Criptovaluta VALUES ('UIO', 171648950000);
INSERT INTO Criptovaluta VALUES ('OPA', 9200373562);
INSERT INTO Criptovaluta VALUES ('SDF', 9192745820);
INSERT INTO Criptovaluta VALUES ('GHJ', 9183746272991);

```

AZIENDA

```

INSERT INTO Azienda VALUES (1, 'Amazon', 'USA', 30560700000, 70035600000, 30500 );
INSERT INTO Azienda VALUES (2, 'Sony', 'Japan', 3012578000, 10560000, 73000 );
INSERT INTO Azienda VALUES (3, 'Asus', 'Taiwan', 700500900, 1300500000, 12000 );
INSERT INTO Azienda VALUES (4, 'Hawlet Packard', 'USA', 1433700500, 7800531000, 1500 );
INSERT INTO Azienda VALUES (5, 'Huawei', 'China', 7600400000, 21912600000, 17300 );
INSERT INTO Azienda VALUES (6, 'Apple', 'USA', 70500321000, 119344021000, 48000);
INSERT INTO Azienda VALUES (7, 'Samsung', 'South Korea', 3234550000, 16732001000, 49546
);
INSERT INTO Azienda VALUES (8, 'DreamTeam Company', 'Italy', -6001, 1200500, 3 );
INSERT INTO Azienda VALUES (9, 'Bytedance', 'China', 37560700000, 6035600000, 8560 );
INSERT INTO Azienda VALUES (10, 'Coursera', 'USA', 534550000, 732001000, 1000 );
INSERT INTO Azienda VALUES (11, 'Pfizer', 'USA', 8600400000, 9035600000, 30000 );
INSERT INTO Azienda VALUES (12, 'KFC', 'USA', 16560700000, 11960700000, 70000 );
INSERT INTO Azienda VALUES (13, 'Audi', 'Germany', 10960700000, 12160700000, 50000 );
INSERT INTO Azienda VALUES (14, 'Fiat', 'Italy', 3930200000, 2591600000, 30000 );
INSERT INTO Azienda VALUES (15, 'Mulino Bianco', 'Italy', 530200000, 491600000, 10000 );
INSERT INTO Azienda VALUES (16, 'eDreams', 'Spain', 1151000000, 1480000000, 3000 );
INSERT INTO Azienda VALUES (17, 'Booking', 'Netherlands', 2321000000, 2062000000, 5000
);

```

```
INSERT INTO Azienda VALUES (18, 'Tesla', 'USA', 90500321000, 70500321000, 100000 );
INSERT INTO Azienda VALUES (19, 'WWE', 'USA', 10500321000, 10500321000, 10000 );
INSERT INTO Azienda VALUES (20, 'Yamaha', 'Japan', 8500321000, 9500321000, 30000 );
INSERT INTO Azienda VALUES (21, 'Q8', 'Kuwait', 13600321000, 11200321000, 50000 );
INSERT INTO Azienda VALUES (22, 'Esso', 'USA', 20500321000, 18900321000, 60000 );
INSERT INTO Azienda VALUES (23, 'Immobiliare', 'Italy', 40321000, 36721000, 10000 );
INSERT INTO Azienda VALUES (24, 'BBC', 'UK', 4700321000, 3900348000, 10000 );
INSERT INTO Azienda VALUES (25, 'Alibaba', 'China', 18360700000, 41405600000, 20400 );
INSERT INTO Azienda VALUES (26, 'NASA', 'USA', 1300321000, 9700321000, 30000 );
INSERT INTO Azienda VALUES (27, 'WWF', 'switzerland', 4321000, 2321000, 60000 );
INSERT INTO Azienda VALUES (28, 'Warner Bros', 'USA', 5800331000, 7201840000, 10000 );
INSERT INTO Azienda VALUES (29, 'Universal', 'USA', 4903381000, 4171191000, 12000 );
INSERT INTO Azienda VALUES (30, 'Ubisoft', 'Canada', 3670321000, 3392880000, 80000 );
INSERT INTO Azienda VALUES (31, 'Nike', 'USA', 6503831000, 5592380000, 800000 );
INSERT INTO Azienda VALUES (32, 'Adidas', 'Germany', 4903831000, 4573840000, 730000 );
```

OPERA

```
INSERT INTO Opera VALUES (1, 'technology');
INSERT INTO Opera VALUES (1, 'e-commerce');
INSERT INTO Opera VALUES (2, 'technology');
INSERT INTO Opera VALUES (3, 'technology');
INSERT INTO Opera VALUES (4, 'technology');
INSERT INTO Opera VALUES (5, 'technology');
INSERT INTO Opera VALUES (6, 'technology');
INSERT INTO Opera VALUES (7, 'technology');
INSERT INTO Opera VALUES (8, 'technology');
INSERT INTO Opera VALUES (8, 'education');
INSERT INTO Opera VALUES (8, 'software');
INSERT INTO Opera VALUES (9, 'software');
INSERT INTO Opera VALUES (10, 'education');
INSERT INTO Opera VALUES (11, 'medicine');
INSERT INTO Opera VALUES (12, 'catering');
INSERT INTO Opera VALUES (13, 'automotive');
INSERT INTO Opera VALUES (14, 'automotive');
INSERT INTO Opera VALUES (15, 'food');
INSERT INTO Opera VALUES (16, 'tourism');
INSERT INTO Opera VALUES (17, 'tourism');
INSERT INTO Opera VALUES (18, 'automotive');
INSERT INTO Opera VALUES (18, 'technology');
INSERT INTO Opera VALUES (18, 'renewable energy');
INSERT INTO Opera VALUES (19, 'entertainment');
INSERT INTO Opera VALUES (20, 'automotive');
INSERT INTO Opera VALUES (21, 'petrol');
INSERT INTO Opera VALUES (22, 'petrol');
INSERT INTO Opera VALUES (23, 'building');
INSERT INTO Opera VALUES (24, 'science');
INSERT INTO Opera VALUES (24, 'entertainment');
INSERT INTO Opera VALUES (24, 'education');
```

```
INSERT INTO Opera VALUES (25, 'e-commerce');
INSERT INTO Opera VALUES (26, 'astronomy');
INSERT INTO Opera VALUES (27, 'biology');
INSERT INTO Opera VALUES (28, 'entertainment');
INSERT INTO Opera VALUES (29, 'entertainment');
INSERT INTO Opera VALUES (30, 'technology');
INSERT INTO Opera VALUES (30, 'software');
INSERT INTO Opera VALUES (30, 'entertainment');
INSERT INTO Opera VALUES (31, 'clothing');
INSERT INTO Opera VALUES (32, 'clothing');
```

IMPIEGATO

```
INSERT INTO Impiegato VALUES ('CNCTLMT2000', 'Italy', 'Matteo', 'Concutelli', 8);
INSERT INTO Impiegato VALUES ('STRMRMT2001', 'Italy', 'Matteo', 'Stromieri', 8);
INSERT INTO Impiegato VALUES ('ZHOULC2000', 'China', 'Luca', 'Zhou', 8);
INSERT INTO Impiegato VALUES ('MUSKELN1971', 'Canada', 'Elon', 'Musk', 18);
INSERT INTO Impiegato VALUES ('GPTRNGL2001', 'Italy', 'Gaia', 'Petrangeli', 12);
INSERT INTO Impiegato VALUES ('JHNCN1979', 'USA', 'John', 'Cena', 19);
INSERT INTO Impiegato VALUES ('THRCK1972', 'USA', 'Dwayne', 'Johnson', 28);
INSERT INTO Impiegato VALUES ('SMNMNC', 'Italy', 'Simone', 'Mancia', 1);
INSERT INTO Impiegato VALUES ('PNCPLN2003', 'Germany', 'Pinco', 'Pallino', 5);
INSERT INTO Impiegato VALUES ('MRRSS1990', 'UK', 'Mario', 'Rossi', 15);
INSERT INTO Impiegato VALUES ('ALCVRD2002', 'France', 'Alice', 'Verdi', 31);
INSERT INTO Impiegato VALUES ('SMNZHNG2001', 'China', 'Simone', 'Zheng', 22);
INSERT INTO Impiegato VALUES ('FRNCX2001', 'China', 'Francesco', 'Xu', 32);
INSERT INTO Impiegato VALUES ('FRDMRCR1945', 'UK', 'Freddie', 'Mercury', 20);
INSERT INTO Impiegato VALUES ('CHRSPRT1978', 'USA', 'Chris', 'Pratt', 28);
INSERT INTO Impiegato VALUES ('LNGLNG1980', 'China', 'Lang', 'Lang', 20);
INSERT INTO Impiegato VALUES ('DNLWS1994', 'Australia', 'Dean', 'Lewis', 20);
INSERT INTO Impiegato VALUES ('MRKBRN1999', 'Spain', 'Mark', 'Brown', 1);
INSERT INTO Impiegato VALUES ('SHWNL1985', 'Canada', 'Shawn', 'Lee', 2);
INSERT INTO Impiegato VALUES ('MRVNVT1994', 'Italy', 'Marvin', 'Vettori', 6);
INSERT INTO Impiegato VALUES ('NDRNG1970', 'UK', 'Andrew', 'Ng', 10);
INSERT INTO Impiegato VALUES ('RSTXVL2000', 'Germany', 'Orest', 'Xhuveli', 15);
INSERT INTO Impiegato VALUES ('LDVCND1948', 'Italy', 'Ludovico', 'Einaudi', 20);
INSERT INTO Impiegato VALUES ('JSTNWT1998', 'Canada', 'Justin', 'White', 17);
INSERT INTO Impiegato VALUES ('RMNPRZ1998', 'Spain', 'Ramon', 'Perez', 26);
INSERT INTO Impiegato VALUES ('LRNJKS1990', 'USA', 'Lauren', 'Jackson', 4);
INSERT INTO Impiegato VALUES ('JMHNG1992', 'South Korea', 'Jim', 'Heung', 11);
INSERT INTO Impiegato VALUES ('YNGZNG2000', 'China', 'Ying', 'Zhang', 26);
INSERT INTO Impiegato VALUES ('BRKSNC1995', 'Russia', 'Brock', 'Sinclair', 9);
INSERT INTO Impiegato VALUES ('JLJNS2002', 'USA', 'Julie', 'Jones', 23);
```

BANCA

```
INSERT INTO Banca VALUES (3723, 'Unicredit', 'Italy', 63, 921, 7000 );
```

```

INSERT INTO Banca VALUES (7842,'Industrial & Commercial Bank of China Limited', 'China', 1933, 38890, 22000 );
INSERT INTO Banca VALUES (6476,'Credito Agricole SA', 'France', 5.1, 419, 6000);
INSERT INTO Banca VALUES (2598,'JPMorgan Chase Bank', 'USA', 124.5, 6922, 9000);
INSERT INTO Banca VALUES (1694,'Japan Post Bank', 'Japan', 31, 1230, 4000 );
INSERT INTO Banca VALUES (3199,'Bank of Algeria', 'Algeria', 2.9, 198, 3400 );
INSERT INTO Banca VALUES (4629,'Intesa San Paolo', 'Italy', 4.67, 300, 6000);
INSERT INTO Banca VALUES (2056,'Bank of America', 'USA', 89, 3200, 20000);
INSERT INTO Banca VALUES (6879,'Wells Fargo', 'USA', 76, 2190, 19000);
INSERT INTO Banca VALUES (5707,'Bank of Germany', 'Germany', 23.3, 2560, 12394);
INSERT INTO Banca VALUES (2140,'Bank of Argentine', 'Argentine', 13.3, 2345, 49243);
INSERT INTO Banca VALUES (7924,'Canadian Bank', 'Canada', 7.3, 1568, 14853);
INSERT INTO Banca VALUES (1372,'Spanish Central Bank', 'Spain', 33.2, 1234, 2561);
INSERT INTO Banca VALUES (4467,'Bank of Romania', 'Romania', 12.1, 1323, 79842);
INSERT INTO Banca VALUES (5006,'Bank of Taiwan', 'Taiwan', 6.7, 8753, 15561);
INSERT INTO Banca VALUES (7437,'Bank of China', 'China', 14.5, 1235, 11231);
INSERT INTO Banca VALUES (8698,'Bank of Japan', 'Japan', 8.8, 2956, 54846);
INSERT INTO Banca VALUES (8451,'Bank of France', 'France', 9.0, 4861, 95453);
INSERT INTO Banca VALUES (3982,'Bank of UK', 'UK', 8.7, 4645, 15654);
INSERT INTO Banca VALUES (4943,'Bank of Italy', 'Italy', 2.3, 8653, 15651);
INSERT INTO Banca VALUES (3861,'Bank of Russia', 'Russia', 2.4, 7492, 56456);
INSERT INTO Banca VALUES (8518,'Bank of Switzerland', 'Switzerland', 4.9, 7563, 15651);
INSERT INTO Banca VALUES (7279,'Bank of Brazil', 'Brazil', 7.7, 846, 35168);
INSERT INTO Banca VALUES (3005,'Bank of Shanghai', 'China', 8.7, 1458, 65226);
INSERT INTO Banca VALUES (3459,'Bank of Los Angeles', 'USA', 12.8, 3426, 65535);
INSERT INTO Banca VALUES (9998,'Bank of New York', 'USA', 17.3, 7513, 95522);
INSERT INTO Banca VALUES (7334,'Bank of London', 'UK', 13.6, 7432, 5623);
INSERT INTO Banca VALUES (5997,'Bank of Madrid', 'Spain', 10.4, 1432, 51653);
INSERT INTO Banca VALUES (6673,'Bank of Sao Paolo', 'Brazil', 8.2, 732, 4845);
INSERT INTO Banca VALUES (8652,'Central Bank of Tokyo', 'Japan', 3.5, 2432, 64531);
INSERT INTO Banca VALUES (4808,'Central Bank of Australia', 'Australia', 14.6, 2358, 61143);
INSERT INTO Banca VALUES (2429,'Toronto Credit', 'Canada', 8.3, 2324, 15626);
INSERT INTO Banca VALUES (9782,'Banca Centrale di Ferentino', 'Italy', 2918, 762232, 100);

```

MERCATO DELLE COMMODITY

```

INSERT INTO MercatoDelleCommodity VALUES('Commodity Market of New York', 10000,'USA');
INSERT INTO MercatoDelleCommodity VALUES('Commodity Market of Toronto', 3000,'Canada');
INSERT INTO MercatoDelleCommodity VALUES('Commodity Market of Hong Kong', 10000,'China');
INSERT INTO MercatoDelleCommodity VALUES('Central Commodity Market of China', 14000,'China');
INSERT INTO MercatoDelleCommodity VALUES('Commodity Market of Rome', 900,'Italy');
INSERT INTO MercatoDelleCommodity VALUES('Trading Center - Los Angeles', 8000,'USA');
INSERT INTO MercatoDelleCommodity VALUES('London Trading Center', 70000,'UK');
INSERT INTO MercatoDelleCommodity VALUES('Swiss Commodity Exchange', 4000,'Switzerland');

```

```
INSERT INTO MercatoDelleCommodity VALUES('Beijing Commodity Market', 11000, 'China');
INSERT INTO MercatoDelleCommodity VALUES('Japanese Central Trading Center',
8000, 'Japan');
INSERT INTO MercatoDelleCommodity VALUES('French Commodity Exchange', 3000, 'France');
INSERT INTO MercatoDelleCommodity VALUES('Turin Exchange', 600, 'Italy');
```

ASSET

```
INSERT INTO Asset VALUES('I101', 1900, 1800, 4222000);
INSERT INTO Asset VALUES('I102', 1200, 1400, 3622000);
INSERT INTO Asset VALUES('I103', 900, 1100, 2932000);
INSERT INTO Asset VALUES('U101', 2900, 2800, 10222000);
INSERT INTO Asset VALUES('U102', 3500, 4000, 24292000);
INSERT INTO Asset VALUES('U103', 3100, 2700, 15252000);
INSERT INTO Asset VALUES('U104', 2500, 2400, 8712000);
INSERT INTO Asset VALUES('U105', 4800, 5500, 24333000);
INSERT INTO Asset VALUES('C101', 2900, 3300, 20881000);
INSERT INTO Asset VALUES('C102', 3100, 2700, 19622000);
INSERT INTO Asset VALUES('C103', 4100, 4400, 29258000);
INSERT INTO Asset VALUES('C104', 3900, 4200, 30992000);
INSERT INTO Asset VALUES('C105', 1900, 1500, 7812000);
INSERT INTO Asset VALUES('C106', 2700, 2200, 10222000);
INSERT INTO Asset VALUES('R101', 2300, 1500, 7632000);
INSERT INTO Asset VALUES('R102', 1700, 2000, 3132000);
INSERT INTO Asset VALUES('E101', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('E102', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('F101', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('F102', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('C201', 1900, 1800, 4222000);
INSERT INTO Asset VALUES('C202', 1200, 1400, 3622000);
INSERT INTO Asset VALUES('C203', 900, 1100, 2932000);
INSERT INTO Asset VALUES('C204', 2900, 2800, 10222000);
INSERT INTO Asset VALUES('C205', 3500, 4000, 24292000);
INSERT INTO Asset VALUES('C206', 3100, 2700, 15252000);
INSERT INTO Asset VALUES('C207', 2500, 2400, 8712000);
INSERT INTO Asset VALUES('C208', 4800, 5500, 24333000);
INSERT INTO Asset VALUES('C209', 2900, 3300, 20881000);
INSERT INTO Asset VALUES('C210', 3100, 2700, 19622000);
INSERT INTO Asset VALUES('C211', 4100, 4400, 29258000);
INSERT INTO Asset VALUES('C212', 3900, 4200, 30992000);
INSERT INTO Asset VALUES('C213', 1900, 1500, 7812000);
INSERT INTO Asset VALUES('C214', 2700, 2200, 10222000);
INSERT INTO Asset VALUES('C215', 2300, 1500, 7632000);
INSERT INTO Asset VALUES('C216', 1700, 2000, 3132000);
INSERT INTO Asset VALUES('C217', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('C218', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('C219', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('C220', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('C221', 2400, 2500, 8185000);
```

```
INSERT INTO Asset VALUES('C222', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('C223', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('AMZN', 2900, 2800, 24222000);
INSERT INTO Asset VALUES('SNY', 2300, 1900, 10222000);
INSERT INTO Asset VALUES('ASUS', 1500, 2000, 6222000);
INSERT INTO Asset VALUES('HP', 1100, 1200, 4222000);
INSERT INTO Asset VALUES('HAWI', 1900, 1800, 12222000);
INSERT INTO Asset VALUES('APL', 1900, 1800, 4222000);
INSERT INTO Asset VALUES('SMSG', 1200, 1400, 3622000);
INSERT INTO Asset VALUES('DTC', 900, 1100, 2932000);
INSERT INTO Asset VALUES('BTD', 2900, 2800, 10222000);
INSERT INTO Asset VALUES('CSR', 3500, 4000, 24292000);
INSERT INTO Asset VALUES('PZR', 3100, 2700, 15252000);
INSERT INTO Asset VALUES('KFC', 2500, 2400, 8712000);
INSERT INTO Asset VALUES('RACE', 4800, 5500, 24333000);
INSERT INTO Asset VALUES('FIAT', 2900, 3300, 20881000);
INSERT INTO Asset VALUES('MBCO', 3100, 2700, 19622000);
INSERT INTO Asset VALUES('DRM', 4100, 4400, 29258000);
INSERT INTO Asset VALUES('BKNG', 3900, 4200, 30992000);
INSERT INTO Asset VALUES('TSL', 1900, 1500, 7812000);
INSERT INTO Asset VALUES('WWE', 2700, 2200, 10222000);
INSERT INTO Asset VALUES('YMH', 2300, 1500, 7632000);
INSERT INTO Asset VALUES('Q8', 1700, 2000, 3132000);
INSERT INTO Asset VALUES('ESSO', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('IMBL', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('BBC', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('BABA', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('NASA', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('WWF', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('WBR', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('UBI', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('UNI', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('WIN', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('NIKE', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('ADS', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('0001', 1900, 1800, 4222000);
INSERT INTO Asset VALUES('0002', 1200, 1400, 3622000);
INSERT INTO Asset VALUES('0003', 900, 1100, 2932000);
INSERT INTO Asset VALUES('0004', 2900, 2800, 10222000);
INSERT INTO Asset VALUES('0005', 3500, 4000, 24292000);
INSERT INTO Asset VALUES('0006', 3100, 2700, 15252000);
INSERT INTO Asset VALUES('0007', 2500, 2400, 8712000);
INSERT INTO Asset VALUES('0008', 4800, 5500, 24333000);
INSERT INTO Asset VALUES('0009', 2900, 3300, 20881000);
INSERT INTO Asset VALUES('0010', 3100, 2700, 19622000);
INSERT INTO Asset VALUES('0011', 4100, 4400, 29258000);
INSERT INTO Asset VALUES('0012', 3900, 4200, 30992000);
INSERT INTO Asset VALUES('0013', 1900, 1500, 7812000);
INSERT INTO Asset VALUES('0014', 2700, 2200, 10222000);
INSERT INTO Asset VALUES('0015', 2300, 1500, 7632000);
```

```
INSERT INTO Asset VALUES('0016', 1700, 2000, 3132000);
INSERT INTO Asset VALUES('0017', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('0018', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('0019', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('0020', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('0021', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('0022', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('0023', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('0024', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('0025', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('0026', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('0027', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('0028', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('0029', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('0030', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('0031', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('D001', 1900, 1800, 4222000);
INSERT INTO Asset VALUES('D002', 1200, 1400, 3622000);
INSERT INTO Asset VALUES('D003', 900, 1100, 2932000);
INSERT INTO Asset VALUES('D004', 2900, 2800, 10222000);
INSERT INTO Asset VALUES('D005', 3500, 4000, 24292000);
INSERT INTO Asset VALUES('D006', 3100, 2700, 15252000);
INSERT INTO Asset VALUES('D007', 2500, 2400, 8712000);
INSERT INTO Asset VALUES('D008', 4800, 5500, 24333000);
INSERT INTO Asset VALUES('D009', 2900, 3300, 20881000);
INSERT INTO Asset VALUES('D010', 3100, 2700, 19622000);
INSERT INTO Asset VALUES('D011', 4100, 4400, 29258000);
INSERT INTO Asset VALUES('D012', 3900, 4200, 30992000);
INSERT INTO Asset VALUES('D013', 1900, 1500, 7812000);
INSERT INTO Asset VALUES('D014', 2700, 2200, 10222000);
INSERT INTO Asset VALUES('D015', 2300, 1500, 7632000);
INSERT INTO Asset VALUES('D016', 1700, 2000, 3132000);
INSERT INTO Asset VALUES('D017', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('D018', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('D019', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('D020', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('D021', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('D022', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('D023', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('D024', 3100, 3600, 17552000);
INSERT INTO Asset VALUES('D025', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('D026', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('D027', 2000, 2100, 5890000);
INSERT INTO Asset VALUES('D028', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('D029', 2400, 2500, 8185000);
INSERT INTO Asset VALUES('D030', 1900, 2500, 6771000);
INSERT INTO Asset VALUES('D031', 2000, 2100, 5890000);
```

TITOLO DI STATO

```
INSERT INTO TitoloDiStato VALUES('I101', '2025-10-25','Italy', 3.3, 3.6);
INSERT INTO TitoloDiStato VALUES('I102', '2026-01-12','Italy', 5.9, 4.9);
INSERT INTO TitoloDiStato VALUES('I103', '2028-05-08','Italy', 2.2, 2.7);
INSERT INTO TitoloDiStato VALUES('U101', '2025-12-22','USA', 3.2, 3.8);
INSERT INTO TitoloDiStato VALUES('U102', '2024-09-27','USA', 1.5, 4.1);
INSERT INTO TitoloDiStato VALUES('U103', '2027-04-11','USA', 5.3, 5);
INSERT INTO TitoloDiStato VALUES('U104', '2030-02-10','USA', 4.2, 7.1);
INSERT INTO TitoloDiStato VALUES('C101', '2028-10-25','China', 5.5, 8.1);
INSERT INTO TitoloDiStato VALUES('C102', '2032-11-01','China', 6.5, 8.2);
INSERT INTO TitoloDiStato VALUES('C103', '2024-07-09','China', 4.3, 5.9);
INSERT INTO TitoloDiStato VALUES('R101', '2028-11-07','Russia', 6.4, 6.7);
INSERT INTO TitoloDiStato VALUES('E101', '2032-11-01','UK', 4.3, 4.7);
INSERT INTO TitoloDiStato VALUES('F101', '2024-07-09','France', 3.2, 5.1);
INSERT INTO TitoloDiStato VALUES('U105', '2031-11-10','USA', 4.4, 7.6);
INSERT INTO TitoloDiStato VALUES('C104', '2026-06-23','China', 3.3, 4.1);
INSERT INTO TitoloDiStato VALUES('C105', '2025-07-03','China', 5.5, 3.2);
INSERT INTO TitoloDiStato VALUES('C106', '2024-07-09','China', 4.8, 2.2);
INSERT INTO TitoloDiStato VALUES('R102', '2028-11-07','Russia', 6.4, 6.7);
INSERT INTO TitoloDiStato VALUES('E102', '2026-12-15','UK', 7.3, 3.7);
INSERT INTO TitoloDiStato VALUES('F102', '2028-03-19','France', 4.2, 2.1);
```

COMMODITY

```
INSERT INTO Commodity VALUES('C201',50000 , 'Turin Exchange');
INSERT INTO Commodity VALUES('C202',60000 , 'Commodity Market of Rome');
INSERT INTO Commodity VALUES('C203',220000 , 'Commodity Market of New York');
INSERT INTO Commodity VALUES('C204',200000 , 'Trading Center - Los Angeles');
INSERT INTO Commodity VALUES('C205',400000 , 'Central Commodity Market of China');
INSERT INTO Commodity VALUES('C206',450000 , 'Beijing Commodity Market');
INSERT INTO Commodity VALUES('C207',65000 , 'French Commodity Exchange');
INSERT INTO Commodity VALUES('C208',190000 , 'Japanese Central Trading Center');
INSERT INTO Commodity VALUES('C209',220000 , 'Commodity Market of Toronto');
INSERT INTO Commodity VALUES('C210',300000 , 'Swiss Commodity Exchange');
INSERT INTO Commodity VALUES('C211',220000 , 'Commodity Market of Toronto');
INSERT INTO Commodity VALUES('C212',300000 , 'Swiss Commodity Exchange');
INSERT INTO Commodity VALUES('C213',260000 , 'Commodity Market of New York');
INSERT INTO Commodity VALUES('C214',150000 , 'Commodity Market of New York');
INSERT INTO Commodity VALUES('C215',260000 , 'London Trading Center');
INSERT INTO Commodity VALUES('C216',280000 , 'London Trading Center');
INSERT INTO Commodity VALUES('C217',330000 , 'London Trading Center');
INSERT INTO Commodity VALUES('C218',290000 , 'Commodity Market of Hong Kong');
INSERT INTO Commodity VALUES('C219',540000 , 'Commodity Market of Hong Kong');
INSERT INTO Commodity VALUES('C220',240000 , 'Commodity Market of Hong Kong');
INSERT INTO Commodity VALUES('C221',290000 , 'Central Commodity Market of China');
INSERT INTO Commodity VALUES('C222',670000 , 'Commodity Market of Hong Kong');
INSERT INTO Commodity VALUES('C223',430000 , 'Beijing Commodity Market');
```

CLASSIFICAZIONE AZIONE

```
INSERT INTO ClassificazioneAzione VALUES
('Common', 'True', 'False', 'True', 'True', 'False', 'False');
INSERT INTO ClassificazioneAzione VALUES
('Preference', 'False', 'False', 'True', 'True', 'False', 'True');
INSERT INTO ClassificazioneAzione VALUES
('Growth', 'True', 'True', 'True', 'False', 'False', 'False');
INSERT INTO ClassificazioneAzione VALUES
('Value', 'True', 'False', 'True', 'False', 'False', 'True');
INSERT INTO ClassificazioneAzione VALUES
('Income', 'False', 'False', 'False', 'False', 'False', 'True');
INSERT INTO ClassificazioneAzione VALUES ('Blue
Chip', 'False', 'True', 'True', 'False', 'True', 'False');
INSERT INTO ClassificazioneAzione VALUES
('Cyclic', 'True', 'True', 'True', 'True', 'True', 'True');
INSERT INTO ClassificazioneAzione VALUES
('Acyclic', 'False', 'False', 'False', 'False', 'False', 'False');
INSERT INTO ClassificazioneAzione VALUES
('Defensive', 'True', 'False', 'True', 'False', 'True', 'False');
INSERT INTO ClassificazioneAzione VALUES
('IPO', 'False', 'True', 'False', 'True', 'False', 'True');
INSERT INTO ClassificazioneAzione VALUES ('Penny
Stock', 'False', 'False', 'True', 'True', 'False', 'True');
INSERT INTO ClassificazioneAzione VALUES
('ESG', 'True', 'False', 'False', 'True', 'True', 'False');
```

AZIONE

```
INSERT INTO Azione VALUES('AMZN',1, 'Common');
INSERT INTO Azione VALUES('SNY',2 , 'Preference');
INSERT INTO Azione VALUES('ASUS',3, 'Growth');
INSERT INTO Azione VALUES('HP',4 , 'Income');
INSERT INTO Azione VALUES('HAWI',5, 'IPO');
INSERT INTO Azione VALUES('APL',6 , 'Common');
INSERT INTO Azione VALUES('SMSG',7, 'Common');
INSERT INTO Azione VALUES('DTC',8 , 'Penny Stock');
INSERT INTO Azione VALUES('BTD',9 , 'Preference');
INSERT INTO Azione VALUES('CSR',10 , 'Growth');
INSERT INTO Azione VALUES('PZR',11 , 'Acyclic');
INSERT INTO Azione VALUES('KFC',12, 'Defensive');
INSERT INTO Azione VALUES('RACE',13, 'Growth');
INSERT INTO Azione VALUES('FIAT',14 , 'Common');
INSERT INTO Azione VALUES('MBCO',15 , 'Preference');
INSERT INTO Azione VALUES('DRM',16 , 'Value');
INSERT INTO Azione VALUES('BKNG',17, 'Blue Chip');
INSERT INTO Azione VALUES('TSL',18, 'Cyclic');
INSERT INTO Azione VALUES('WWE',19, 'Preference');
```

```
INSERT INTO Azione VALUES('YMH',20, 'Income');
INSERT INTO Azione VALUES('Q8',21, 'Growth');
INSERT INTO Azione VALUES('ESSO',22, 'Common');
INSERT INTO Azione VALUES('IMBL',23, 'Preference');
INSERT INTO Azione VALUES('BBC',24 , 'Common');
INSERT INTO Azione VALUES('BABA',25, 'Preference');
INSERT INTO Azione VALUES('NASA',26, 'Penny Stock');
INSERT INTO Azione VALUES('WWF',27, 'Growth');
INSERT INTO Azione VALUES('WBR',28 , 'Common');
INSERT INTO Azione VALUES('UBI',30 , 'Growth');
INSERT INTO Azione VALUES('UNI',29, 'Income');
INSERT INTO Azione VALUES('WIN',31, 'Preference');
INSERT INTO Azione VALUES('NIKE',31, 'Income');
INSERT INTO Azione VALUES('ADS',32, 'Growth');
```

OBBLIGAZIONE

```
INSERT INTO Obbligazione VALUES('0001',5,'2025-08-29', 3.5, 9);
INSERT INTO Obbligazione VALUES('0002',5,'2023-03-23', 3.3, 5);
INSERT INTO Obbligazione VALUES('0003',6,'2024-02-19', 5.5, 5);
INSERT INTO Obbligazione VALUES('0004',2,'2026-07-09', 4.5, 9.2);
INSERT INTO Obbligazione VALUES('0005',18,'2028-12-28', 6.1, 6.1);
INSERT INTO Obbligazione VALUES('0006',18,'2023-10-14', 2.2, 5.6);
INSERT INTO Obbligazione VALUES('0007',24,'2029-01-21', 1.9, 2.1);
INSERT INTO Obbligazione VALUES('0008',7,'2030-01-10', 3.6, 4.5);
INSERT INTO Obbligazione VALUES('0009',7,'2026-02-21', 7.5, 6.5);
INSERT INTO Obbligazione VALUES('0010',20,'2026-05-19', 5.3, 6.7);
INSERT INTO Obbligazione VALUES('0011',31,'2028-05-31', 8, 10);
INSERT INTO Obbligazione VALUES('0012',32,'2024-05-09', 2.8, 2.8);
INSERT INTO Obbligazione VALUES('0013',1,'2023-07-25', 5.8, 3.6);
INSERT INTO Obbligazione VALUES('0014',2,'2023-09-18', 5.9, 3.4);
INSERT INTO Obbligazione VALUES('0015',1,'2026-03-08', 8.6, 9.2);
INSERT INTO Obbligazione VALUES('0016',19,'2027-02-21', 1.2, 1);
INSERT INTO Obbligazione VALUES('0017',29,'2030-11-27', 1.7, 2.2);
INSERT INTO Obbligazione VALUES('0018',9,'2031-10-22', 7, 8.8);
INSERT INTO Obbligazione VALUES('0019',12,'2023-11-07', 3.9, 2.5);
INSERT INTO Obbligazione VALUES('0020',25,'2030-11-12', 7.4, 6.1);
INSERT INTO Obbligazione VALUES('0021',17,'2031-04-27', 3.7, 4.5);
INSERT INTO Obbligazione VALUES('0022',14,'2024-04-18', 5.9, 7.3);
INSERT INTO Obbligazione VALUES('0023',13,'2029-07-17', 4.5, 4.5);
INSERT INTO Obbligazione VALUES('0024',11,'2025-08-13', 6.5, 8.2);
INSERT INTO Obbligazione VALUES('0025',30,'2025-08-13', 6.6, 7.7);
INSERT INTO Obbligazione VALUES('0026',32,'2027-05-10', 8.5, 7.6);
INSERT INTO Obbligazione VALUES('0027',25,'2026-10-16', 9.5, 11);
INSERT INTO Obbligazione VALUES('0028',1,'2027-03-24', 3.9, 6.5);
INSERT INTO Obbligazione VALUES('0029',13,'2029-05-15', 2.9, 7.9);
INSERT INTO Obbligazione VALUES('0030',3,'2023-07-21', 1.6, 4.7);
INSERT INTO Obbligazione VALUES('0031',13,'2029-01-12', 5.6, 3.7);
```

STRUMENTO DERIVATO

```
INSERT INTO StrumentoDerivato VALUES('D001', 5, 'HAWI');
INSERT INTO StrumentoDerivato VALUES('D002', 5, 'HAWI');
INSERT INTO StrumentoDerivato VALUES('D003', 6, 'APL');
INSERT INTO StrumentoDerivato VALUES('D004', 6, 'APL');
INSERT INTO StrumentoDerivato VALUES('D005', 2, 'SNY');
INSERT INTO StrumentoDerivato VALUES('D006', 18, 'TSL');
INSERT INTO StrumentoDerivato VALUES('D007', 8, 'DTC');
INSERT INTO StrumentoDerivato VALUES('D008', 18, 'TSL');
INSERT INTO StrumentoDerivato VALUES('D009', 1, 'AMZN');
INSERT INTO StrumentoDerivato VALUES('D010', 25, 'BABA');
INSERT INTO StrumentoDerivato VALUES('D011', 25, 'BABA');
INSERT INTO StrumentoDerivato VALUES('D012', 7, 'SMSG');
INSERT INTO StrumentoDerivato VALUES('D013', 3, 'ASUS');
INSERT INTO StrumentoDerivato VALUES('D014', 11, 'PZR');
INSERT INTO StrumentoDerivato VALUES('D015', 11, 'PZR');
INSERT INTO StrumentoDerivato VALUES('D016', 31, 'WIN');
INSERT INTO StrumentoDerivato VALUES('D017', 32, 'HAWI');
INSERT INTO StrumentoDerivato VALUES('D018', 30, 'UBI');
INSERT INTO StrumentoDerivato VALUES('D019', 12, 'KFC');
INSERT INTO StrumentoDerivato VALUES('D020', 13, 'RACE');
INSERT INTO StrumentoDerivato VALUES('D021', 7, 'SMSG');
INSERT INTO StrumentoDerivato VALUES('D022', 9, 'BTD');
INSERT INTO StrumentoDerivato VALUES('D023', 5, 'HAWI');
INSERT INTO StrumentoDerivato VALUES('D024', 25, 'BABA');
INSERT INTO StrumentoDerivato VALUES('D025', 6, 'APL');
INSERT INTO StrumentoDerivato VALUES('D026', 24, 'BBC');
INSERT INTO StrumentoDerivato VALUES('D027', 29, 'UNI');
INSERT INTO StrumentoDerivato VALUES('D028', 28, 'WBR');
INSERT INTO StrumentoDerivato VALUES('D029', 9, 'BTD');
INSERT INTO StrumentoDerivato VALUES('D030', 13, 'RACE');
INSERT INTO StrumentoDerivato VALUES('D031', 19, 'WWE');
```

BORSA

```
INSERT INTO Borsa VALUES ('Euronext Securities Milan', 520200400, 'Italy');
INSERT INTO Borsa VALUES ('New York Stock Exchange', 30201000, 'USA');
INSERT INTO Borsa VALUES ('Tokyo Stock Exchange', 400300400, 'Japan');
INSERT INTO Borsa VALUES ('Hong Kong Stock Exchange', 560320000, 'China');
INSERT INTO Borsa VALUES ('Shanghai Stock Exchange', 870901000, 'China');
INSERT INTO Borsa VALUES ('Amsterdam Stock Exchange', 320459000, 'Netherlands');
INSERT INTO Borsa VALUES ('NASDAQ', 900230900, 'USA');
INSERT INTO Borsa VALUES ('Shenzhen Stock Exchange', 1782309908, 'China');
INSERT INTO Borsa VALUES ('Toronto Stock Exchange', 123810939, 'Canada');
INSERT INTO Borsa VALUES ('Deutsche Boerse', 274828100, 'Germany');
INSERT INTO Borsa VALUES ('London Stock Exchange', 83741029, 'UK');
INSERT INTO Borsa VALUES ('Saudi Stock Exchange', 283741209, 'Saudi Arabia');
```

```
INSERT INTO Borsa VALUES ('Copenhagen Stock Exchange', 327419220, 'Denmark');
INSERT INTO Borsa VALUES ('Euronext Paris Exchange', 293791820, 'France');
INSERT INTO Borsa VALUES ('Frankfurt Stock Exchange', 928379120, 'Germany');
INSERT INTO Borsa VALUES ('Swiss Stock Exchange', 236109288, 'switzerland');
INSERT INTO Borsa VALUES ('South Korea Stock Exchange', 923782190, 'South Korea');
INSERT INTO Borsa VALUES ('Australia Stock Exchange', 341928109, 'Australia');
INSERT INTO Borsa VALUES ('Taiwan Stock Exchange', 32871901, 'Taiwan');
INSERT INTO Borsa VALUES ('Tehran Stock Exchange', 127392102, 'Iran');
INSERT INTO Borsa VALUES ('Johannesburg Stock Exchange', 209302128, 'South Africa');
INSERT INTO Borsa VALUES ('Algiers Stock Exchange', 837320310, 'Algeria');
INSERT INTO Borsa VALUES ('Angola Debt and Stock Exchange', 982374102, 'Angola');
INSERT INTO Borsa VALUES ('Egyptian Exchange', 938102939, 'Egypt');
INSERT INTO Borsa VALUES ('China Financial Futures Exchange', 2371820, 'China');
INSERT INTO Borsa VALUES ('JASDAQ', 2031830100, 'Japan');
INSERT INTO Borsa VALUES ('JASDAQ NEO', 900230100, 'Japan');
INSERT INTO Borsa VALUES ('Mothers', 817091810, 'Japan');
```

AZIENDA POSSIEDE CRIPTOVALUTA

```
INSERT INTO AziendaPossiedeCriptovaluta VALUES(1, 'ETH');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(2, 'ETH');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(6, 'ETH');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(7, 'ETH');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(1, 'BTC');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(2, 'BNB');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(6, 'BTC');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(7, 'NNN');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(13, 'ETH');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(18, 'QWE');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(29, 'UIO');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(13, 'OPA');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(18, 'FAT');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(29, 'SOL');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(13, 'ADA');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(15, 'DOPE');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(22, 'DOPE');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(13, 'AGE');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(11, 'FAG');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(22, 'GET');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(1, 'SDF');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(2, 'RTY');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(2, 'BTC');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(6, 'ADA');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(4, 'DAT');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(4, 'XRP');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(11, 'GHJ');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(19, 'DAT');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(18, 'XRP');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(18, 'GHJ');
```

```
INSERT INTO AziendaPossiedeCriptovaluta VALUES(30, 'OPA');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(31, 'ETH');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(32, 'BTC');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(30, 'BTC');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(28, 'USDC');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(25, 'USDC');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(25, 'DOPE');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(11, '3RAT');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(9, 'BUSD');
INSERT INTO AziendaPossiedeCriptovaluta VALUES(9, '3RAT');
```

IMPIEGATO POSSIEDE CRIPTOVALUTA

```
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'ETH');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('JHNCN1979', 'USA', 'ETH');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('GPTRNGL2001', 'Italy', 'ETH');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('ZHOULC2000', 'China', 'ETH');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'BTC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'BNB');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('JHNCN1979', 'USA', 'BTC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'NNN');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'QWE');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('JHNCN1979', 'USA', 'UIO');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'OPA');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('GPTRNGL2001', 'Italy', 'FAT');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('ZHOULC2000', 'China', 'SOL');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'ADA');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('ZHOULC2000', 'China', 'DOPE');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'DOPE');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('CNCTLMT2000', 'Italy', 'AGE');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('THRCK1972', 'USA', 'FAG');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('CNCTLMT2000', 'Italy', 'GET');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('STRMRMT2001', 'Italy', 'SDF');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'RTY');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('CNCTLMT2000', 'Italy', 'BTC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('STRMRMT2001', 'Italy', 'BTC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('STRMRMT2001', 'Italy', 'DAT');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('ZHOULC2000', 'China', 'XRP');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('THRCK1972', 'USA', 'GHJ');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'DAT');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('MUSKELN1971', 'Canada', 'XRP');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('GPTRNGL2001', 'Italy', 'GHJ');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('GPTRNGL2001', 'Italy', 'OPA');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('THRCK1972', 'USA', 'ETH');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('ZHOULC2000', 'China', 'BTC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('THRCK1972', 'USA', 'BTC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('ZHOULC2000', 'China', 'USDC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('THRCK1972', 'USA', 'USDC');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('STRMRMT2001', 'Italy', 'DOPE');
```

```
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('THRCK1972', 'USA', '3RAT');
INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('STRMRMT2001', 'Italy', 'BUSD');
```

AZIENDA POSSIEDE ASSET

```
INSERT INTO AziendaPossiedeAsset VALUES(1, '0001');
INSERT INTO AziendaPossiedeAsset VALUES(1, '0002');
INSERT INTO AziendaPossiedeAsset VALUES(1, '0005');
INSERT INTO AziendaPossiedeAsset VALUES(2, 'D009');
INSERT INTO AziendaPossiedeAsset VALUES(2, 'AMZN');
INSERT INTO AziendaPossiedeAsset VALUES(3, 'BABA');
INSERT INTO AziendaPossiedeAsset VALUES(4, '0020');
INSERT INTO AziendaPossiedeAsset VALUES(5, 'PZR');
INSERT INTO AziendaPossiedeAsset VALUES(5, 'I101');
INSERT INTO AziendaPossiedeAsset VALUES(6, 'C105');
INSERT INTO AziendaPossiedeAsset VALUES(6, 'C102');
INSERT INTO AziendaPossiedeAsset VALUES(7, 'D011');
INSERT INTO AziendaPossiedeAsset VALUES(7, 'APL');
INSERT INTO AziendaPossiedeAsset VALUES(11, 'APL');
INSERT INTO AziendaPossiedeAsset VALUES(12, 'BTD');
INSERT INTO AziendaPossiedeAsset VALUES(12, 'C205');
INSERT INTO AziendaPossiedeAsset VALUES(13, 'F101');
INSERT INTO AziendaPossiedeAsset VALUES(14, 'R101');
INSERT INTO AziendaPossiedeAsset VALUES(15, 'C103');
INSERT INTO AziendaPossiedeAsset VALUES(17, 'U101');
INSERT INTO AziendaPossiedeAsset VALUES(17, '0021');
INSERT INTO AziendaPossiedeAsset VALUES(18, '0019');
INSERT INTO AziendaPossiedeAsset VALUES(19, '0019');
INSERT INTO AziendaPossiedeAsset VALUES(21, 'D015');
INSERT INTO AziendaPossiedeAsset VALUES(21, '0003');
INSERT INTO AziendaPossiedeAsset VALUES(22, 'C103');
INSERT INTO AziendaPossiedeAsset VALUES(22, 'UBI');
INSERT INTO AziendaPossiedeAsset VALUES(22, 'NIKE');
INSERT INTO AziendaPossiedeAsset VALUES(23, 'SNY');
INSERT INTO AziendaPossiedeAsset VALUES(24, 'HAWI');
INSERT INTO AziendaPossiedeAsset VALUES(25, 'HAWI');
INSERT INTO AziendaPossiedeAsset VALUES(26, 'AMZN');
INSERT INTO AziendaPossiedeAsset VALUES(26, 'BABA');
INSERT INTO AziendaPossiedeAsset VALUES(27, 'BABA');
INSERT INTO AziendaPossiedeAsset VALUES(27, '0017');
INSERT INTO AziendaPossiedeAsset VALUES(28, 'D013');
INSERT INTO AziendaPossiedeAsset VALUES(29, 'C209');
INSERT INTO AziendaPossiedeAsset VALUES(29, 'C211');
INSERT INTO AziendaPossiedeAsset VALUES(29, 'ADS');
INSERT INTO AziendaPossiedeAsset VALUES(31, 'U105');
INSERT INTO AziendaPossiedeAsset VALUES(32, 'YMH');
```

CONTIENE

```
INSERT INTO Contiene VALUES ('SP500','AMZN');
INSERT INTO Contiene VALUES ('DJI','SNY');
INSERT INTO Contiene VALUES ('NAC', 'ASUS');
INSERT INTO Contiene VALUES ('TASK','HP');
INSERT INTO Contiene VALUES ('PAC','HAWI');
INSERT INTO Contiene VALUES ('STOF','APL');
INSERT INTO Contiene VALUES ('LOAD','SMSG');
INSERT INTO Contiene VALUES ('KAYT','DTC');
INSERT INTO Contiene VALUES ('KATY','BTD');
INSERT INTO Contiene VALUES ('FOAM','CSR');
INSERT INTO Contiene VALUES ('LOTA','PZR');
INSERT INTO Contiene VALUES ('DATTO','KFC');
INSERT INTO Contiene VALUES ('TORA','RACE');
INSERT INTO Contiene VALUES ('TTORA','FIAT');
INSERT INTO Contiene VALUES ('KAPPA','MBCO');
INSERT INTO Contiene VALUES ('VAMO','DRM');
INSERT INTO Contiene VALUES ('ESPN', 'BKNG');
INSERT INTO Contiene VALUES ('COCA','TSL');
INSERT INTO Contiene VALUES ('COLA','WWE');
INSERT INTO Contiene VALUES ('ELLE','YMH');
INSERT INTO Contiene VALUES ('HAP','Q8');
INSERT INTO Contiene VALUES ('SAMS','ESSO');
INSERT INTO Contiene VALUES ('WAOM','IMBL');
INSERT INTO Contiene VALUES ('MARIO','BBC');
INSERT INTO Contiene VALUES ('GATTO','BABA');
INSERT INTO Contiene VALUES ('RACE','NASA');
INSERT INTO Contiene VALUES ('FAP','WWF');
INSERT INTO Contiene VALUES ('MICRO','WBR');
```

AZIONE QUOTATA IN

```
INSERT INTO AzioneQuotataIn VALUES ('AMZN','Toronto Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('AMZN','Swiss Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('APL','London Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('RACE','Taiwan Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('WIN','Shenzhen Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('HAWI','China Financial Futures Exchange');
INSERT INTO AzioneQuotataIn VALUES ('HAWI','Saudi Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('FIAT','Saudi Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('WWE','Hong Kong Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('Q8','Amsterdam Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('ESSO','Australia Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('BABA','Mothers');
INSERT INTO AzioneQuotataIn VALUES ('BABA','Copenhagen Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('BBC','Taiwan Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('WBR','Frankfurt Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('TSL','South Korea Stock Exchange');
```

```
INSERT INTO AzioneQuotataIn VALUES ('TSL', 'Tehran Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('TSL', 'Egyptian Exchange');
INSERT INTO AzioneQuotataIn VALUES ('TSL', 'Deutsche Boerse');
INSERT INTO AzioneQuotataIn VALUES ('DTC', 'New York Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('BKNG', 'New York Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('PZR', 'New York Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('DTC', 'Euronext Paris Exchange');
INSERT INTO AzioneQuotataIn VALUES ('DTC', 'NASDAQ');
INSERT INTO AzioneQuotataIn VALUES ('SNY', 'Johannesburg Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('YMH', 'Saudi Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('ASUS', 'Saudi Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('AMZN', 'Hong Kong Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('NIKE', 'Amsterdam Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('UBI', 'Australia Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('UNI', 'Mothers');
INSERT INTO AzioneQuotataIn VALUES ('NASA', 'Copenhagen Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('DRM', 'Toronto Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('WIN', 'Swiss Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('RACE', 'London Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('HAWI', 'Taiwan Stock Exchange');
INSERT INTO AzioneQuotataIn VALUES ('KFC', 'Shenzhen Stock Exchange');
```

OBBLIGAZIONE QUOTATA IN

```
INSERT INTO ObbligazioneQuotataIn VALUES ('0001', 'New York Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0003', 'Swiss Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0013', 'NASDAQ');
INSERT INTO ObbligazioneQuotataIn VALUES ('0005', 'South Korea Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0006', 'New York Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0022', 'Euronext Paris Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0008', 'New York Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0017', 'Frankfurt Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0010', 'Euronext Paris Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0011', 'New York Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0012', 'NASDAQ');
INSERT INTO ObbligazioneQuotataIn VALUES ('0017', 'New York Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0014', 'Toronto Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0015', 'Euronext Paris Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0016', 'NASDAQ');
INSERT INTO ObbligazioneQuotataIn VALUES ('0022', 'Hong Kong Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0019', 'Amsterdam Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0020', 'Euronext Paris Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0021', 'Saudi Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0022', 'New York Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0008', 'South Korea Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0010', 'Deutsche Boerse');
INSERT INTO ObbligazioneQuotataIn VALUES ('0026', 'London Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0003', 'Frankfurt Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0028', 'South Korea Stock Exchange');
```

```
INSERT INTO ObbligazioneQuotataIn VALUES ('0029', 'Australia Stock Exchange');
INSERT INTO ObbligazioneQuotataIn VALUES ('0015', 'Johannesburg Stock Exchange');
```

NAZIONE POSSIEDE ASSET

```
INSERT INTO NazionePossiedeAsset VALUES ('Italy', 'APL');
INSERT INTO NazionePossiedeAsset VALUES ('Italy', 'SMSG');
INSERT INTO NazionePossiedeAsset VALUES ('USA', '0001');
INSERT INTO NazionePossiedeAsset VALUES ('USA', '0004');
INSERT INTO NazionePossiedeAsset VALUES ('USA', 'D018');
INSERT INTO NazionePossiedeAsset VALUES ('USA', 'SMSG');
INSERT INTO NazionePossiedeAsset VALUES ('China', 'F102');
INSERT INTO NazionePossiedeAsset VALUES ('China', 'C211');
INSERT INTO NazionePossiedeAsset VALUES ('China', 'C212');
INSERT INTO NazionePossiedeAsset VALUES ('China', 'D001');
INSERT INTO NazionePossiedeAsset VALUES ('Germany', 'D001');
INSERT INTO NazionePossiedeAsset VALUES ('Germany', 'D002');
INSERT INTO NazionePossiedeAsset VALUES ('UK', 'FIAT');
INSERT INTO NazionePossiedeAsset VALUES ('UK', 'AMZN');
INSERT INTO NazionePossiedeAsset VALUES ('Russia', 'BKNG');
INSERT INTO NazionePossiedeAsset VALUES ('Russia', 'PZR');
INSERT INTO NazionePossiedeAsset VALUES ('France', 'KFC');
INSERT INTO NazionePossiedeAsset VALUES ('Spain', 'KFC');
INSERT INTO NazionePossiedeAsset VALUES ('Japan', 'WWE');
INSERT INTO NazionePossiedeAsset VALUES ('Japan', 'HAWI');
INSERT INTO NazionePossiedeAsset VALUES ('South Korea', 'YMH');
INSERT INTO NazionePossiedeAsset VALUES ('Saudi Arabia', 'TSL');
INSERT INTO NazionePossiedeAsset VALUES ('Canada', 'HAWI');
INSERT INTO NazionePossiedeAsset VALUES ('Canada', 'UNI');
INSERT INTO NazionePossiedeAsset VALUES ('Canada', 'RACE');
INSERT INTO NazionePossiedeAsset VALUES ('Australia', 'BABA');
INSERT INTO NazionePossiedeAsset VALUES ('Australia', 'AMZN');
INSERT INTO NazionePossiedeAsset VALUES ('Argentine', 'BABA');
INSERT INTO NazionePossiedeAsset VALUES ('Brazil', 'BBC');
INSERT INTO NazionePossiedeAsset VALUES ('Switzerland', 'TSL');
INSERT INTO NazionePossiedeAsset VALUES ('Switzerland', 'WBR');
INSERT INTO NazionePossiedeAsset VALUES ('Switzerland', 'APL');
INSERT INTO NazionePossiedeAsset VALUES ('Netherlands', 'AMZN');
INSERT INTO NazionePossiedeAsset VALUES ('USA', 'TSL');
INSERT INTO NazionePossiedeAsset VALUES ('China', 'U105');
INSERT INTO NazionePossiedeAsset VALUES ('China', 'BABA');
INSERT INTO NazionePossiedeAsset VALUES ('USA', 'E102');
INSERT INTO NazionePossiedeAsset VALUES ('Russia', 'D008');
```

QUERY

MYSQL

1. Per ogni azienda si vogliono sapere gli asset posseduti, con nome azienda e ticker asset posseduto:

```
SELECT az.nome, azpa.ticker AS asset  
FROM Azienda az, AziendaPossiedeAsset azpa  
WHERE az.ID = azpa.ID;
```

| nome | asset |
|----------------|-------|
| Amazon | 0001 |
| Amazon | 0002 |
| Amazon | 0005 |
| Sony | AMZN |
| Sony | D009 |
| Asus | BABA |
| Hawlet Packard | O020 |
| Huawei | I101 |
| Huawei | PZR |
| Apple | C102 |
| Apple | C105 |
| Samsung | APL |
| Samsung | D011 |
| Pfizer | APL |
| KFC | BTD |
| KFC | C205 |
| Audi | F101 |
| Fiat | R101 |
| Mulino Bianco | C103 |
| Booking | O021 |
| Booking | U101 |
| Tesla | 0019 |
| WWF | 0019 |
| Q8 | D015 |
| Q8 | 0003 |
| Esso | C103 |
| Esso | NIKE |
| Esso | UBI |
| Immobiliare | SNY |
| BBC | HAWI |
| Alibaba | HAWI |
| NASA | AMZN |
| NASA | BABA |
| WWF | BABA |
| WWF | O017 |
| Warner Bros | D013 |
| Universal | ADS |
| Universal | C209 |
| Universal | C211 |
| Nike | U105 |
| Adidas | YMH |

41 rows in set (0,00 sec)

2. Si vogliono sapere nome e cognome degli impiegati italiani che lavorano nella 'DreamTeam Company' :

```
SELECT Impiegato.nome, cognome  
FROM Impiegato, Azienda  
WHERE  
Impiegato.nazione = 'Italy' AND  
Azienda.nome = 'DreamTeam Company' AND  
Azienda.ID = Impiegato.azienda;
```

| nome | cognome |
|--------|------------|
| Matteo | Concutelli |
| Matteo | Stromieri |
| Luca | Zhou |

3. Si vuole sapere il numero di impiegati per ogni nazione:

```
SELECT Nazione.nome, sum(Azienda.numeroDipendenti) as numeroImpiegati
FROM Azienda, Nazione
WHERE
Azienda.nazione = Nazione.nome GROUP BY Azienda.nazione;
```

| nome | numeroImpiegati |
|-------------|-----------------|
| USA | 1203000 |
| Japan | 103000 |
| Taiwan | 12000 |
| China | 46260 |
| South Korea | 49546 |
| Italy | 50003 |
| Germany | 780000 |
| Spain | 3000 |
| Netherlands | 5000 |
| Kuwait | 50000 |
| UK | 10000 |
| Switzerland | 60000 |
| Canada | 80000 |

4. Si vuole sapere nome e cognome degli impiegati che lavorano per una azienda che emette l'azione 'NIKE', e nome dell'azienda :

```
SELECT Impiegato.nome, cognome, Azienda.nome as azienda
FROM Impiegato, Azienda, Azione
WHERE
Azione.ticker = 'NIKE' AND
Azienda.ID = Azione.aziendaEmittente AND
Impiegato.azienda = Azienda.ID;
```

| nome | cognome | azienda |
|-------------------------|---------|---------|
| Alice | Verdi | Nike |
| 1 row in set (0,00 sec) | | |

5. Si vuole sapere la percentuale di occupazione che un'azienda offre, con nome e nazione dell'azienda:

```
SELECT Azienda.nome, Azienda.nazione, numeroDipendenti/popolazione as occupazione
FROM Azienda, Nazione
WHERE Azienda.nazione = Nazione.nome;
```

| nome | nazione | occupazione |
|-------------------|-------------|-------------|
| Amazon | USA | 0.0001 |
| Sony | Japan | 0.0006 |
| Asus | Taiwan | 0.0005 |
| Hawlet Packard | USA | 0.0000 |
| Huawei | China | 0.0000 |
| Apple | USA | 0.0001 |
| Samsung | South Korea | 0.0010 |
| DreamTeam Company | Italy | 0.0000 |
| Bytedance | China | 0.0000 |
| Coursera | USA | 0.0000 |
| Pfizer | USA | 0.0001 |
| KFC | USA | 0.0002 |
| Audi | Germany | 0.0006 |
| Fiat | Italy | 0.0005 |
| Mulino Bianco | italy | 0.0002 |
| eDreams | Spain | 0.0001 |
| Booking | Netherlands | 0.0003 |
| Tesla | USA | 0.0003 |
| WWE | USA | 0.0000 |
| Yamaha | Japan | 0.0002 |
| Q8 | Kuwait | 0.0117 |
| Esso | USA | 0.0002 |
| Immobiliare | Italy | 0.0002 |
| BBC | UK | 0.0001 |
| Alibaba | China | 0.0000 |
| NASA | USA | 0.0001 |
| WWF | switzerland | 0.0069 |
| Warner Bros | USA | 0.0000 |
| Universal | USA | 0.0000 |
| Ubisoft | Canada | 0.0021 |
| Nike | USA | 0.0024 |
| Adidas | Germany | 0.0088 |

6. Si vogliono conoscere quali aziende italiane sono in perdita (utile negativo):

```
SELECT nome, utile, fatturato
FROM Azienda
WHERE Azienda.nazione = 'Italy' AND utile < 0;
```

| nome | utile | fatturato |
|-------------------|----------|------------|
| DreamTeam Company | -6001.00 | 1200500.00 |

7. Elencare le criptovalute possedute dall'impiegato "Luca Zhou" e quelle possedute da Amazon:

```
SELECT CONCAT(Impiegato.nome, " ", cognome) as nome ipc.ticker
FROM ImpiegatoPossiedeCriptovaluta ipc, Impiegato
WHERE
    ipc.CF = Impiegato.CF AND
    Impiegato.nome = 'Luca' AND
    Impiegato.cognome = 'Zhou'
UNION
SELECT nome, apc.ticker FROM Azienda, AziendaPossiedeCriptovaluta apc
WHERE
    apc.ID = Azienda.ID AND
    nome = 'Amazon';
```

| nome | ticker |
|----------|--------|
| LucaZhou | BTC |
| LucaZhou | DOPE |
| LucaZhou | ETH |
| LucaZhou | SOL |
| LucaZhou | USDC |
| LucaZhou | XRP |
| Amazon | BTC |
| Amazon | ETH |
| Amazon | SDF |

8. Per ogni impiegato si vuole sapere la nazione e l'azienda per cui lavora:

```
SELECT Impiegato.nome, cognome, Impiegato.nazione, Azienda.nome as azienda
FROM Impiegato,Azienda
WHERE Impiegato.azienda = Azienda.ID;
```

| nome | cognome | nazione | azienda |
|-----------|------------|-------------|-------------------|
| Alice | Verdi | France | Nike |
| Brock | Sinclair | Russia | Bytedance |
| Chris | Pratt | USA | Warner Bros |
| Matteo | Concutelli | Italy | DreamTeam Company |
| Dean | Lewis | Australia | Yamaha |
| Freddie | Mercury | UK | Yamaha |
| Francesco | Xu | China | Adidas |
| Gaia | Petrangeli | Italy | KFC |
| John | Cena | USA | WWE |
| Julie | Jones | USA | Immobiliare |
| Jim | Heung | South Korea | Pfizer |
| Justin | White | Canada | Booking |
| Ludovico | Einaudi | Italy | Yamaha |
| Lang | Lang | China | Yamaha |
| Lauren | Jackson | USA | Hawlett Packard |
| Mark | Brown | Spain | Amazon |
| Mario | Rossi | UK | Mulino Bianco |
| Marvin | Vettori | Italy | Apple |
| Elon | Musk | Canada | Tesla |
| Andrew | Ng | UK | Coursera |
| Pinco | Pallino | Germany | Huawei |
| Ramon | Perez | Spain | NASA |
| Orest | Xhuveli | Germany | Mulino Bianco |
| Shawn | Lee | Canada | Sony |
| Simone | Mancia | Italy | Amazon |
| Simone | Zheng | China | Esso |
| Matteo | Stromieri | Italy | DreamTeam Company |
| Dwayne | Johnson | USA | Warner Bros |
| Ying | Zhang | China | NASA |
| Luca | Zhou | China | DreamTeam Company |

9. Si vogliono sapere tutte le azioni e obbligazioni quotate nella stessa borsa in cui è quotata Amazon (AMZN), con nome azione, nome obbligazione e nome borsa:

```
SELECT AzioneQuotataIn.ticker, ObbligazioneQuotataIn.ticker, Borsa.nome
FROM Borsa, AzioneQuotataIn, AzioneQuotataIn x, ObbligazioneQuotataIn
WHERE
x.ticker = 'AMZN' AND
AzioneQuotataIn.borsa = x.borsa AND
ObbligazioneQuotataIn.borsa = x.borsa AND
AzioneQuotataIn.borsa = ObbligazioneQuotataIn.borsa AND
Borsa.nome = AzioneQuotataIn.borsa AND
AzioneQuotataIn.ticker != 'AMZN';
```

| ticker | ticker | nome |
|--------|--------|--------------------------|
| WWE | 0022 | Hong Kong Stock Exchange |
| WIN | 0003 | Swiss Stock Exchange |
| DRM | 0014 | Toronto Stock Exchange |

10. Per ogni settore si vuole sapere il numero di aziende che opera in questo ed il ricavato complessivo:

```
SELECT Settore.nome, count(*) as numAziende, ricavatoComplessivo
FROM Opera, Settore
WHERE Opera.settore = Settore.nome GROUP BY Settore.nome;
```

| nome | numAziende | ricavatoComplessivo |
|------------------|------------|---------------------|
| astronomy | 1 | 590265234984.35 |
| automotive | 4 | 450265234984.53 |
| biology | 1 | 230265234984.73 |
| building | 1 | 600265234984.52 |
| catering | 1 | 2190265234984.72 |
| clothing | 2 | 735865234484.63 |
| e-commerce | 2 | 770265234984.17 |
| education | 3 | 1990265234984.11 |
| entertainment | 5 | 475865234484.63 |
| food | 1 | 2990265234984.29 |
| medicine | 1 | 150265234984.98 |
| petrol | 2 | 980265234984.34 |
| renewable energy | 1 | 210265234984.74 |
| science | 1 | 110265234984.18 |
| software | 3 | 290265234984.30 |
| technology | 10 | 670265234984.19 |
| tourism | 2 | 270265234984.66 |

11. Si vuole sapere il nome, nazione, capitalizzazione e numero dipendenti delle banche situate dove è situata la banca 'Unicredit' :

```
SELECT y.nome, y.capitalizzazione, y.numerodipendenti
FROM Banca x, Banca y
Where
x.nome = 'Unicredit' AND
y.nazione = x.nazione;
```

| nome | capitalizzazione | numerodipendenti |
|-----------------------------|------------------|------------------|
| Unicredit | 921 | 7000 |
| Intesa San Paolo | 300 | 6000 |
| Bank of Italy | 8653 | 15651 |
| Banca Centrale di Ferentino | 762232 | 100 |

12. Per ogni impiegato si vuole conoscere nome, cognome e settore in cui lavora:

```
SELECT x.nome, x.cognome, settore
FROM Impiegato x, Azienda, Opera
WHERE
Azienda.ID = x.azienda AND
Azienda.ID = Opera.azienda;
```

| nome | cognome | settore |
|-----------|------------|------------------|
| Alice | Verdi | clothing |
| Brock | Sinclair | software |
| Chris | Pratt | entertainment |
| Matteo | Concutelli | education |
| Matteo | Concutelli | software |
| Matteo | Concutelli | technology |
| Dean | Lewis | automotive |
| Freddie | Mercury | automotive |
| Francesco | Xu | clothing |
| Gaia | Petrangeli | catering |
| John | Cena | entertainment |
| Julie | Jones | building |
| Jim | Heung | medicine |
| Justin | White | tourism |
| Ludovico | Einaudi | automotive |
| Lang | Lang | automotive |
| Lauren | Jackson | technology |
| Mark | Brown | e-commerce |
| Mark | Brown | technology |
| Mario | Rossi | food |
| Marvin | Vettori | technology |
| Elon | Musk | automotive |
| Elon | Musk | renewable energy |
| Elon | Musk | technology |
| Andrew | Ng | education |
| Pinco | Pallino | technology |
| Ramon | Perez | astronomy |
| Orest | Xhuveli | food |
| Shawn | Lee | technology |
| Simone | Mancia | e-commerce |
| Simone | Mancia | technology |
| Simone | Zheng | petrol |
| Matteo | Stromieri | education |
| Matteo | Stromieri | software |
| Matteo | Stromieri | technology |
| Dwayne | Johnson | entertainment |
| Ying | Zhang | astronomy |
| Luca | Zhou | education |
| Luca | Zhou | software |
| Luca | Zhou | technology |

13. Quante aziende operano nel settore "technology":

```
SELECT settore, count(*) as numAziende
FROM Opera
WHERE
settore = 'technology' GROUP BY settore;
```

| settore | numAziende |
|------------|------------|
| technology | 10 |

14. Si vogliono sapere le tipologie di azioni possedute dal Canada:

```
SELECT tipologia, nazione
FROM NazionePossiedeAsset x, Azione, ClassificazioneAzione y
WHERE
x.nazione = 'Canada' AND
Azione.ticker = x.ticker AND
y.tipologia = Azione.classificazioneAzione;
```

| tipologia | nazione |
|-----------|---------|
| IPO | Canada |
| Growth | Canada |
| Income | Canada |

15. Si vogliono sapere quali nazioni hanno un tasso di povertà minore di quello della nazionalità di "Elon Musk":

```
SELECT Nazione.nome, Nazione.tassoPoverta
FROM Nazione, (select tassoPoverta from Impiegato, Nazione where Impiegato.nome = 'Elon' and cognome =
'Musk' and Impiegato.nazione = Nazione.nome) as NazioneMusk
WHERE Nazione.tassoPoverta < NazioneMusk.tassoPoverta;
```

| nome | tassoPoverta |
|-------------|--------------|
| Australia | 7.80 |
| Bahamas | 5.90 |
| Belize | 1.20 |
| Bolivia | 7.10 |
| Brazil | 3.40 |
| Bulgary | 8.20 |
| China | 0.30 |
| Germany | 3.50 |
| Italy | 9.10 |
| Netherlands | 3.50 |
| Switzerland | 8.50 |
| Taiwan | 1.50 |
| UK | 3.00 |

16. Si vuole sapere quale borsa quota più azioni di aziende residenti negli USA:

```
SELECT borsa, numAzioni
FROM
(select max(numAzioni) as max from (select count(*) numAzioni from Azione, Azienda, AzioneQuotataIn
where nazione = 'USA' and aziendaEmittente = ID and AzioneQuotataIn.ticker = Azione.ticker group by borsa)
X,
(select borsa, count(*) numAzioni
from Azione, Azienda, AzioneQuotataIn
where nazione = 'USA' and aziendaEmittente = ID and AzioneQuotataIn.ticker = Azione.ticker group by borsa)
Y
WHERE
max = numAzioni;
```

| borsa | numAzioni |
|--------------------------|-----------|
| Hong Kong Stock Exchange | 2 |
| Swiss Stock Exchange | 2 |
| Shenzhen Stock Exchange | 2 |

17. Si vogliono conoscere tutti gli asset che non sono azioni e che non sono posseduti da aziende:

```
SELECT Asset.ticker
FROM Asset LEFT JOIN Azione ON Asset.ticker = Azione.ticker
LEFT JOIN AziendaPossiedeAsset ON AziendaPossiedeAsset.ticker = Asset.ticker
WHERE
Azione.ticker IS NULL AND
AziendaPossiedeAsset.ticker IS NULL;
```

| ticker |
|--------|
| C101 |
| C104 |
| C106 |
| C201 |
| C202 |
| C203 |
| C204 |
| C206 |
| C207 |
| C208 |
| C210 |
| C212 |
| C213 |
| C214 |
| C215 |
| C216 |
| C217 |
| C218 |
| C219 |
| C220 |
| C221 |
| C222 |
| C223 |
| D001 |
| D002 |
| D003 |

18. Si vogliono conoscere nome e cognome degli impiegati che lavorano in un'azienda USA nel settore tecnologico:

```
SELECT Impiegato.nome, cognome
FROM Impiegato, Azienda, Opera
WHERE
    Opera.settore = 'Canada' AND
    Azienda.ID = Opera.azienda AND
    Impiegato.azienda = Azienda.ID AND
    Azienda.nazione = 'USA';
```

| nome | cognome |
|--------|---------|
| Mark | Brown |
| Simone | Mancia |
| Lauren | Jackson |
| Marvin | Vettori |
| Elon | Musk |

19. Quali criptovalute non sono possedute da nessun impiegato:

```
SELECT Criptovaluta.ticker
FROM Criptovaluta LEFT JOIN ImpiegatoPossiedeCriptovaluta ON Criptovaluta.ticker =
ImpiegatoPossiedeCriptovaluta.ticker
WHERE
    ImpiegatoPossiedeCriptovaluta.CF IS NULL;
```

| ticker |
|--------|
| ALT |
| DOGE |
| DOT |
| GUS |
| KNT |
| LAT |
| USDT |

20. Lista dei settori in cui lavorano degli italiani in ordine decrescente di numero di lavoratori italiani:

```
SELECT Opera.settore, count(*) as numImp
FROM Opera, Azienda, Impiegato
WHERE
    Opera.azienda = Azienda.ID AND
    Azienda.ID = Impiegato.azienda AND
    Impiegato.nazione = 'Italy' GROUP BY Opera.settore
ORDER BY numImp DESC;
```

| settore | numImp |
|------------|--------|
| technology | 4 |

21. Nazioni con un Pil superiore a quello dell'Italia, con il loro Pil:

```
SELECT X.nome, X.pil
FROM Nazione X, Nazione Y
WHERE Y.nome = 'Italy'
AND X.pil >= Y.pil
AND X.nome != Y.nome;
```

| nome | pil |
|-------|-------|
| China | 14720 |
| UK | 27000 |
| USA | 20940 |

22. Top 3 dipendenti con il maggior numero di tipi di criptovalute, che non sono canadesi:

```
SELECT ImpiegatoPossiedeCriptovaluta .CF, COUNT(*) AS num_cripto
FROM ImpiegatoPossiedeCriptovaluta, Impiegato
WHERE ImpiegatoPossiedeCriptovaluta .CF = Impiegato.CF
AND Impiegato.nazione != 'Canada'
GROUP BY ImpiegatoPossiedeCriptovaluta .CF
```

```

ORDER BY num_cripto DESC
LIMIT 3;

```

| CF | num_cripto |
|-------------|------------|
| ZHOULC2000 | 6 |
| THRCK1972 | 6 |
| STRMRMT2001 | 5 |

23. Numero di asset che ogni nazione ha in ordine decrescente:

```

SELECT Nazione.nome, COUNT(NazionePossiedeAsset.ticker) as num_asset
FROM Nazione JOIN NazionePossiedeAsset ON Nazione.nome = NazionePossiedeAsset.nazione
GROUP BY Nazione.nome
ORDER BY num_asset DESC
LIMIT 10;

```

| nome | num_asset |
|-------------|-----------|
| China | 6 |
| USA | 6 |
| Russia | 3 |
| Canada | 3 |
| Switzerland | 3 |
| UK | 2 |
| Italy | 2 |
| Japan | 2 |
| Australia | 2 |
| Germany | 2 |

24. Obbligazioni con scadenza oltre il 2028 che non sono né degli USA né cinesi:

```

SELECT Obbligazione.ticker, Obbligazione.scadenza, Azienda.nome
FROM Obbligazione join Azienda on Obbligazione.azienda = Azienda.ID
AND DATE(Obbligazione.scadenza) >= "2028-01-01";

```

| ticker | scadenza | nome |
|--------|------------|-----------|
| 0005 | 2028-12-28 | Tesla |
| 0007 | 2029-01-21 | BBC |
| 0008 | 2030-01-10 | Samsung |
| 0011 | 2028-05-31 | Nike |
| 0017 | 2030-11-27 | Universal |
| 0018 | 2031-10-22 | Bytedance |
| 0020 | 2030-11-12 | Alibaba |
| 0021 | 2031-04-27 | Booking |
| 0023 | 2029-07-17 | Audi |
| 0029 | 2029-05-15 | Audi |
| 0031 | 2029-01-12 | Audi |

25. Società avente sede nella stessa nazione della società “Nike”, in ordine crescente di numero degli asset posseduti:

```
SELECT X.ID, X.nome, COUNT(XX.ticker) as num_asset
FROM Azienda X, AziendaPossiedeAsset XX, Azienda Y, AziendaPossiedeAsset YY
WHERE X.ID = XX.ID
AND Y.ID = YY.ID
AND Y.nome = 'Nike'
AND X.nazione = Y.nazione
AND X.nome != 'Nike'
GROUP BY XX.ID
ORDER BY num_asset DESC;
```

| ID | nome | num_asset |
|----|----------------|-----------|
| 1 | Amazon | 3 |
| 22 | Esso | 3 |
| 29 | Universal | 3 |
| 6 | Apple | 2 |
| 12 | KFC | 2 |
| 26 | NASA | 2 |
| 4 | Hawlet Packard | 1 |
| 11 | Pfizer | 1 |
| 18 | Tesla | 1 |
| 19 | WWE | 1 |
| 28 | Warner Bros | 1 |

26. I 5 paesi più popolosi che usano l'euro:

```
SELECT Nazione.nome, Nazione.popolazione
FROM Nazione JOIN Valuta ON Nazione.valuta = Valuta.Codice_ISO
WHERE Valuta.nome = 'Euro'
ORDER BY Nazione.popolazione DESC
LIMIT 5;
```

| nome | popolazione |
|-------------|-------------|
| Germany | 83240000 |
| France | 67390000 |
| Italy | 59550000 |
| Spain | 47350000 |
| Netherlands | 17440000 |

27. Aziende con i maggiori fatturati (se presenti) per ogni settore, in ordine decrescente di fatturato:

```
SELECT Opera.settore, Azienda.ID, Azienda.nome, Azienda.fatturato
FROM Opera JOIN Azienda ON Opera.azienda = Azienda.ID
WHERE (Opera.settore, Azienda.fatturato) IN
    (SELECT Opera.settore, MAX(Azienda.fatturato)
     FROM Azienda, Opera
     WHERE Azienda.ID = Opera.azienda
     GROUP BY Opera.settore
    )
ORDER BY Azienda.fatturato DESC;
```

| settore | ID | nome | fatturato |
|------------------|----|---------------|-----------------|
| technology | 6 | Apple | 119344021504.00 |
| automotive | 18 | Tesla | 70500319232.00 |
| renewable energy | 18 | Tesla | 70500319232.00 |
| e-commerce | 1 | Amazon | 70035603456.00 |
| petrol | 22 | Esso | 18900320256.00 |
| catering | 12 | KFC | 11960699904.00 |
| entertainment | 19 | WWE | 10500321280.00 |
| astronomy | 26 | NASA | 9700321280.00 |
| medicine | 11 | Pfizer | 9035599872.00 |
| software | 9 | Bytedance | 6035599872.00 |
| clothing | 31 | Nike | 5592379904.00 |
| education | 24 | BBC | 3900347904.00 |
| science | 24 | BBC | 3900347904.00 |
| tourism | 17 | Booking | 2862000000.00 |
| food | 15 | Mulino Bianco | 491600000.00 |
| building | 23 | Immobiliare | 36721000.00 |
| biology | 27 | WWF | 2321000.00 |

28. Top 5 settori con il più alto fatturato totale delle sue aziende in ordine crescente:

```
SELECT Opera.settore, SUM(Azienda.fatturato) as tot_revenue
FROM Opera JOIN Azienda ON Opera.azienda = Azienda.ID
GROUP BY Opera.settore
ORDER BY tot_revenue DESC
LIMIT 5;
```

| settore | tot_revenue |
|------------------|-----------------|
| technology | 311030217588.00 |
| e-commerce | 111441203200.00 |
| automotive | 94752940544.00 |
| renewable energy | 70500319232.00 |
| petrol | 30100640768.00 |

29. Dipendenti residenti nella stessa nazione in cui risiede azienda "Alibaba":

```
SELECT Impiegato.CF, Impiegato.nome, Impiegato.cognome, Impiegato.nazione
FROM Impiegato JOIN Azienda ON Impiegato.nazione = Azienda.nazione
WHERE Azienda.nome = 'Alibaba';
```

| CF | nome | cognome | nazione |
|-------------|-----------|---------|---------|
| FRNCX2001 | Francesco | Xu | China |
| LNGLNG1980 | Lang | Lang | China |
| SMNZHNG2001 | Simone | Zheng | China |
| YNGZNG2000 | Ying | Zhang | China |
| ZHOULC2000 | Luca | Zhou | China |

30. Top 3 paesi più popolosi che hanno almeno un'azienda che possiede la criptovaluta "ETH":

```
SELECT DISTINCT Nazione.nome, Nazione.popolazione
FROM Nazione, Azienda, AziendaPossiedeCriptovaluta
WHERE Nazione.nome = Azienda.nazione
AND Azienda.ID = AziendaPossiedeCriptovaluta.ID
AND AziendaPossiedeCriptovaluta.ticker = 'ETH'
ORDER BY Nazione.popolazione DESC
LIMIT 3;
```

| nome | popolazione |
|---------|-------------|
| USA | 329500000 |
| Japan | 125800000 |
| Germany | 83240000 |

31. Classificazioni delle azioni, con il loro numero di azioni negli USA:

- tutte le classificazioni vengono visualizzate:

```
SELECT ClassificazioneAzione.tipologia, Azienda.nazione, COUNT(*) AS num
FROM ClassificazioneAzione, Azione, Azienda
WHERE ClassificazioneAzione.tipologia = Azione.classificazioneAzione
AND Azione.aziendaEmittente = Azienda.ID
AND Azienda.nazione = 'USA'
GROUP BY ClassificazioneAzione.tipologia
ORDER BY num DESC;
```

- solo le classificazioni con almeno un'azione statunitense vengono visualizzate:

```
SELECT Azione.classificazioneAzione, Azienda.nazione, COUNT(*) AS num
FROM Azione, Azienda
WHERE Azione.aziendaEmittente = Azienda.ID
AND Azione.aziendaEmittente = Azienda.ID
AND Azienda.nazione = 'USA'
GROUP BY Azione.classificazioneAzione
ORDER BY num DESC;
```

| tipologia | nazione | num |
|-------------|---------|-----|
| Common | USA | 4 |
| Income | USA | 3 |
| Preference | USA | 2 |
| Growth | USA | 1 |
| Defensive | USA | 1 |
| Penny Stock | USA | 1 |
| Ayclic | USA | 1 |
| Cyclic | USA | 1 |

32. Ricavi totali nel settore technology dalle aziende in Canada:

```
SELECT Nazione.nome, SUM(Azienda.fatturato) as TOT_REVENUETECH
FROM Nazione JOIN Azienda ON Nazione.nome = Azienda.nazione
JOIN Opera ON Azienda.ID = Opera.azienda
AND Opera.settore = 'technology'
AND Nazione.nome = 'Canada'
GROUP BY Nazione.nome;
```

| nome | TOT_REVENUETECH |
|--------|-----------------|
| Canada | 3392880128.00 |

33. Criptovalute che non sono di proprietà di alcun impiegato americano o società americana:

```
SELECT Criptovaluta.ticker
FROM Criptovaluta LEFT JOIN
(SELECT DISTINCT Criptovaluta.ticker as tick
FROM Criptovaluta, AziendaPossiedeCriptovaluta, ImpiegatoPossiedeCriptovaluta, Azienda
WHERE Criptovaluta.ticker = AziendaPossiedeCriptovaluta.ticker
AND Criptovaluta.ticker = ImpiegatoPossiedeCriptovaluta.ticker
AND AziendaPossiedeCriptovaluta.ID = Azienda.ID
AND Azienda.nazione = 'USA'
AND ImpiegatoPossiedeCriptovaluta.nazione = 'USA') AS USA_cripto
ON Criptovaluta.ticker = USA_cripto.tick
WHERE USA_cripto.tick IS NULL;
```

Faccio un LEFT JOIN tra Criptovaluta ed un'altra tabella Criptovaluta ma con le sole criptovalute possedute da impiegati o aziende statunitensi. Usando il LEFT JOIN, si generano valori NULL per le criptovalute non possedute da impiegati o aziende statunitensi. Quindi selezionando ove capita NULL equivale a selezionare le criptovalute non possedute da essi.

| ticker |
|--------|
| ADA |
| AGE |
| ALT |
| BNB |
| BUSD |
| DAT |
| DOGE |
| DOPE |
| DOT |
| FAT |
| GET |
| GUS |
| KNT |
| LAT |
| NNN |
| OPA |
| QWE |
| RTY |
| SDF |
| SOL |
| USDT |
| XRP |

34. Nazioni (diverse da USA e Cina) con il numero di asset possedute in ordine decrescente:

```
SELECT Nazione.nome, COUNT(*) AS num_assets
FROM Nazione JOIN NazionePossiedeAsset
ON Nazione.nome = NazionePossiedeAsset.nazione
WHERE Nazione.nome != 'USA'
AND Nazione.nome != 'China'
GROUP BY NazionePossiedeAsset.nazione
ORDER BY num_assets DESC;
```

| nome | num_assets |
|--------------|------------|
| Canada | 3 |
| Russia | 3 |
| Switzerland | 3 |
| Australia | 2 |
| Germany | 2 |
| Italy | 2 |
| Japan | 2 |
| UK | 2 |
| Argentine | 1 |
| Brazil | 1 |
| France | 1 |
| Netherlands | 1 |
| Saudi Arabia | 1 |
| South Korea | 1 |
| Spain | 1 |

35. Titoli di stato ordinati per rapporto (rendimento / tasso di interesse) in ordine decrescente:

```
SELECT ticker, rendimento/interesse AS ratio  
FROM TitoloDiStato  
ORDER BY ratio DESC;
```

| ticker | ratio |
|--------|----------|
| U102 | 2.733333 |
| U105 | 1.727273 |
| U104 | 1.690476 |
| F101 | 1.593750 |
| C101 | 1.472727 |
| C103 | 1.372093 |
| C102 | 1.261538 |
| C104 | 1.242424 |
| I103 | 1.227273 |
| U101 | 1.187500 |
| E101 | 1.093023 |
| I101 | 1.090909 |
| R101 | 1.046875 |
| R102 | 1.046875 |
| U103 | 0.943396 |
| I102 | 0.830508 |
| C105 | 0.581818 |
| E102 | 0.506849 |
| F102 | 0.500000 |
| C106 | 0.458333 |

36. Tutte le azioni emesse dalle società cinesi, con la società:

```
SELECT Azione.ticker, Azione.aziendaEmittente AS ID, Azienda.nome AS Azienda, Azienda.nazione  
FROM Azione, Azienda  
WHERE Azione.aziendaEmittente = Azienda.ID  
AND Azienda.nazione = 'China';
```

| ticker | ID | Azienda | nazione |
|--------|----|-----------|---------|
| HAWI | 5 | Huawei | China |
| BTD | 9 | Bytedance | China |
| BABA | 25 | Alibaba | China |

37. Aziende nello stesso settore dell'azienda "Samsung" ordinate per fatturato :

```
SELECT X.ID, X.nome, X.fatturato, XX.settore  
FROM Azienda X, Opera XX, Azienda Y, Opera YY  
WHERE X.ID = XX.azienda  
AND Y.ID = YY.azienda  
AND Y.nome = 'Samsung'  
AND XX.settore = YY.settore  
ORDER BY X.fatturato DESC;
```

| ID | nome | fatturato | settore |
|----|-------------------|-----------------|------------|
| 6 | Apple | 119344021504.00 | technology |
| 18 | Tesla | 70500319232.00 | technology |
| 1 | Amazon | 70035603456.00 | technology |
| 5 | Huawei | 21912600576.00 | technology |
| 7 | Samsung | 16732001280.00 | technology |
| 4 | Hawlet Packard | 7800530944.00 | technology |
| 30 | Ubisoft | 3392880128.00 | technology |
| 3 | Asus | 1300499968.00 | technology |
| 2 | Sony | 10560000.00 | technology |
| 8 | DreamTeam Company | 1200500.00 | technology |

38. Dipendenti che lavorano nello stesso settore di "Luca Zhou":

```

SELECT X.CF, X.nome, X.cognome, AX.nome
FROM Impiegato X, Impiegato Y, Azienda AX, Azienda AY, Opera OX, Opera OY
WHERE X.azienda = AX.ID
AND Y.azienda = AY.ID
AND AX.ID = OX.azienda
AND AY.ID = OY.azienda
AND Y.nome = 'Luca'
AND Y.cognome = 'Zhou'
AND OX.settore = OY.settore;
    
```

| CF | nome | cognome | nome |
|--------------|--------|-----------|-------------------|
| MRKBRN1999 | Mark | Brown | Amazon |
| SMNNMC | Simone | Mancia | Amazon |
| ZHOULC2000 | Luca | Zhou | Amazon |
| MRKBRN1999 | Mark | Brown | Amazon |
| SMNNMC | Simone | Mancia | Amazon |
| ZHOULC2000 | Luca | Zhou | Amazon |
| SHWNL1985 | Shawn | Lee | Sony |
| LRNJKS1990 | Lauren | Jackson | Hawlet Packard |
| PNCPLN2003 | Pinco | Pallino | Huawei |
| STRMRRMT2001 | Matteo | Stromieri | Huawei |
| MRVNVT1994 | Marvin | Vettori | Apple |
| RSTXVL2000 | Orest | Xhuveli | DreamTeam Company |
| MUSKELN1971 | Elon | Musk | Tesla |

39. Borsa con il più alto volume di scambio in un paese che utilizza l'euro:

```

SELECT Borsa.nome, Borsa.volumeDiScambi, Borsa.nazione
FROM Borsa, (SELECT Nazione.nome as nation, Nazione.valuta
              FROM Nazione
              WHERE Nazione.valuta = 'EUR') AS eur_nations
WHERE Borsa.nazione = eur_nations.nation
ORDER BY Borsa.volumeDiScambi DESC
LIMIT 1;
    
```

| nome | volumeDiScambi | nazione |
|--------------------------|----------------|---------|
| Frankfurt Stock Exchange | 928379120 | Germany |

40. Numero di aziende, per nazione, con il fatturato totale in ordine decrescente:

```
SELECT Nazione.nome, COUNT(*) AS num_companies, SUM(Azienda.fatturato) AS tot_revenue
FROM Nazione JOIN Azienda ON Nazione.nome = Azienda.nazione
GROUP BY Nazione.nome
ORDER BY tot_revenue DESC;
```

| nome | num_companies | tot_revenue |
|-------------|---------------|-----------------|
| USA | 13 | 345475149824.00 |
| China | 3 | 69353800192.00 |
| Germany | 2 | 16734540288.00 |
| South Korea | 1 | 16732001280.00 |
| Kuwait | 1 | 11200320512.00 |
| Japan | 2 | 9510880768.00 |
| UK | 1 | 3900347904.00 |
| Canada | 1 | 3392880128.00 |
| Italy | 4 | 3121121628.00 |
| Netherlands | 1 | 2062000000.00 |
| Spain | 1 | 1480000000.00 |
| Taiwan | 1 | 1300499968.00 |
| Switzerland | 1 | 2321000.00 |

41. Numero di tipi di criptovaluta delle aziende che sono in paesi con almeno 50 milioni di persone:

```
SELECT Azienda.nome, COUNT(*) AS num
FROM Nazione JOIN Azienda ON Nazione.nome = Azienda.nazione
JOIN AziendaPossiedeCriptovaluta ON Azienda.ID = AziendaPossiedeCriptovaluta.ID
WHERE Nazione.popolazione > 50000000
GROUP BY Azienda.nome;
```

| nome | num |
|----------------|-----|
| Amazon | 3 |
| Sony | 4 |
| Hawlet Packard | 2 |
| Apple | 3 |
| Bytedance | 2 |
| Pfizer | 3 |
| Tesla | 4 |
| WWE | 1 |
| Esso | 2 |
| Alibaba | 2 |
| Warner Bros | 1 |
| Universal | 2 |
| Nike | 1 |

42. Borsa con il più alto volume di scambio nella nazione di cittadinanza di "Luca Zhou":

```
SELECT Borsa.nome, Borsa.VolumeDiScambi, Borsa.Nazione
FROM Borsa, Impiegato, (SELECT MAX(Borsa.VolumeDiScambi) AS max_vol
                        FROM Borsa
                        GROUP BY Borsa.nazione ) AS max_vol_naz
WHERE Borsa.VolumeDiScambi = max_vol_naz.max_vol
AND Borsa.nazione = Impiegato.nazione
AND Impiegato.cognome = 'Zhou'
AND Impiegato.nome = 'Luca';
```

| nome | VolumeDiScambi | Nazione |
|-------------------------|----------------|---------|
| Shenzhen Stock Exchange | 1782309908 | China |

43. Tutte le azioni di tipo 'Common' di aziende che operano nel settore tecnologico:

```
SELECT Azione.ticker
FROM Azione,Azienda,Opera
WHERE Azienda.ID = Opera.azienda
AND Azione.ClassificazioneAzione = 'Common'
AND Azione.AziendaEmittente = Azienda.ID
AND Opera.settore = 'technology';
```

| ticker |
|--------|
| AMZN |
| APL |
| SMSG |

44. L'azienda per la quale lavora l'impiegato che possiede la criptovaluta con il flottante più grande:

```
SELECT Impiegato.nome, Impiegato.cognome, Impiegato.CF, Impiegato.Azienda, Azienda.nome
FROM Impiegato, ImpiegatoPossiedeCriptovaluta, Azienda
WHERE azienda.ID = Impiegato.Azienda
AND ImpiegatoPossiedeCriptovaluta.ticker = (SELECT Criptovaluta.ticker FROM criptovaluta WHERE
Criptovaluta.flottante = (SELECT MAX(flottante) FROM criptovaluta))
AND ImpiegatoPossiedeCriptovaluta.CF = Impiegato.CF;
```

| nome | cognome | CF | Azienda | nome |
|--------|------------|-------------|---------|-------------|
| Gaia | Petrangeli | GPTRNGL2001 | 12 | KFC |
| Dwayne | Johnson | THRCK1972 | 28 | Warner Bros |

45. Dire, per ogni banca, la valuta usata nel paese in cui è collocata:

```
SELECT Banca.nome, Valuta.nome  
FROM Banca,Nazione, Valuta  
WHERE Banca.nazione = Nazione.nome  
AND Nazione.valuta = Valuta.codice_iso;
```

| nome | nome |
|---|-------------------|
| Spanish Central Bank | Euro |
| Japan Post Bank | Yen |
| Bank of America | U.S. dollar |
| Bank of Argentine | Argentinian Peso |
| Toronto Credit | Canadian dollar |
| JPMorgan Chase Bank | U.S. dollar |
| Bank of Shanghai | Yuan |
| Bank of Algeria | Algerian Dinar |
| Bank of Los Angeles | U.S. dollar |
| Unicredit | Euro |
| Bank of Russia | Russian ruble |
| Bank of UK | British pound |
| Bank of Romania | Romanian leu |
| Intesa San Paolo | Euro |
| Central Bank of Australia | Australian Dollar |
| Bank of Italy | Euro |
| Bank of Taiwan | New Taiwan dollar |
| Bank of Germany | Euro |
| Bank of Madrid | Euro |
| Credito Agricole SA | Euro |
| Bank of Sao Paolo | Brazilian Real |
| Wells Fargo | U.S. dollar |
| Bank of Brazil | Brazilian Real |
| Bank of London | British pound |
| Bank of China | Yuan |
| Industrial & Commercial Bank of China Limited | Yuan |
| Canadian Bank | Canadian dollar |
| Bank of France | Euro |
| Bank of Switzerland | Swiss franc |
| Central Bank of Tokyo | Yen |
| Bank of Japan | Yen |
| Banca Centrale di Ferentino | Euro |
| Bank of New York | U.S. dollar |

46. Dire quale Titolo di stato massimizza il valore nominale:

```
SELECT DISTINCT TitoloDiStato.ticker  
FROM TitoloDiStato,Asset  
WHERE TitoloDiStato.ticker = Asset.ticker  
AND Asset.ValoreNominale = (SELECT max(Asset.valoreNominale) FROM TitoloDiStato,Asset WHERE  
TitoloDiStato.Ticker = Asset.Ticker);
```

| ticker |
|--------|
| U105 |

47. Per ogni nazione e per ogni settore si vuole sapere la somma dei fatturati delle aziende che vi operano:

```
SELECT azienda.nazione, opera.settore, SUM(azienda.fatturato) FatturatoTotale  
FROM opera,azienda  
WHERE Azienda.ID = Opera.Azienda GROUP BY Azienda.nazione, Opera.settore;
```

| nazione | settore | FatturatoTotale |
|-------------|------------------|-----------------|
| USA | e-commerce | 70035603456.00 |
| USA | technology | 267680475136.00 |
| Japan | technology | 10560000.00 |
| Taiwan | technology | 1300499968.00 |
| China | technology | 21912600576.00 |
| South Korea | technology | 16732001280.00 |
| Italy | education | 1200500.00 |
| Italy | software | 1200500.00 |
| Italy | technology | 1200500.00 |
| China | software | 6035599872.00 |
| USA | education | 732001024.00 |
| USA | medicine | 9035599872.00 |
| USA | catering | 11960699904.00 |
| Germany | automotive | 12160700416.00 |
| Italy | automotive | 2591600128.00 |
| Italy | food | 491600000.00 |
| Spain | tourism | 1480000000.00 |
| Netherlands | tourism | 2062000000.00 |
| USA | automotive | 70500319232.00 |
| USA | renewable energy | 70500319232.00 |
| USA | entertainment | 21873352448.00 |
| Japan | automotive | 9500320768.00 |
| Kuwait | petrol | 11200320512.00 |
| USA | petrol | 18900320256.00 |
| Italy | building | 36721000.00 |
| UK | education | 3900347904.00 |
| UK | entertainment | 3900347904.00 |
| UK | science | 3900347904.00 |
| China | e-commerce | 41405599744.00 |
| USA | astronomy | 9700321280.00 |
| switzerland | biology | 2321000.00 |
| Canada | entertainment | 3392880128.00 |
| Canada | software | 3392880128.00 |
| Canada | technology | 3392880128.00 |
| USA | clothing | 5592379904.00 |
| Germany | clothing | 4573839872.00 |

48. Si vuole sapere la nazione che ha emesso i titoli di stato a più basso rendimento:

```
SELECT TitoloDiStato.nazione
FROM TitoloDiStato, (SELECT MIN(TitoloDiStato.rendimento) minimo FROM TitoloDiStato ) C
WHERE TitoloDiStato.rendimento = C.minimo;
```

| nazione |
|---------|
| France |

49. Si vogliono sapere quali criptovalute sono possedute dalle aziende che detengono le azioni emesse dall'azienda "Amazon":

```
SELECT DISTINCT AziendaPossiedeCriptovaluta.Ticker
FROM Azione, AziendaPossiedeCriptovaluta, AziendaPossiedeAsset, Asset, Azienda
WHERE Azienda.nome = 'Amazon'
AND Azione.AziendaEmittente = Azienda.ID
AND Asset.ticker = Azione.ticker
AND asset.ticker = aziendaPossiedeAsset.ticker
AND AziendaPossiedeAsset.ID = AziendaPossiedeCriptovaluta.ID;
```

| Ticker |
|--------|
| BNB |
| BTC |
| ETH |
| RTY |

50. Si vogliono sapere i mercati delle commodity nelle quali sono scambiate le commodity che hanno un nome che inizia con la lettera "C":

```
SELECT Commodity.MercatoDelleCommodity FROM Commodity WHERE Commodity.ticker like "C%";
```

| MercatoDelleCommodity |
|-----------------------------------|
| Turin Exchange |
| Commodity Market of Rome |
| Commodity Market of New York |
| Trading Center - Los Angeles |
| Central Commodity Market of China |
| Beijing Commodity Market |
| French Commodity Exchange |
| Japanese Central Trading Center |
| Commodity Market of Toronto |
| Swiss Commodity Exchange |
| Commodity Market of Toronto |
| Swiss Commodity Exchange |
| Commodity Market of New York |
| Commodity Market of New York |
| London Trading Center |
| London Trading Center |
| London Trading Center |
| Commodity Market of Hong Kong |
| Commodity Market of Hong Kong |
| Commodity Market of Hong Kong |
| Central Commodity Market of China |
| Commodity Market of Hong Kong |
| Beijing Commodity Market |

51. si vuole sapere, per ogni borsa, il numero di dipendenti medio delle aziende le cui azioni sono scambiate su quella borsa:

```
SELECT AzioneQuotataIn.borsa, AVG(Azienda.numeroDipendenti) mediaDipendenti
FROM AzioneQuotataIn, Azione, Azienda
WHERE Azione.aziendaEmittente = Azienda.ID
and Azione.ticker = azioneQuotataIn.ticker GROUP BY AzioneQuotataIn.borsa;
```

| borsa | mediaDipendenti |
|----------------------------------|-----------------|
| Amsterdam Stock Exchange | 425000.0000 |
| Australia Stock Exchange | 70000.0000 |
| China Financial Futures Exchange | 173000.0000 |
| Copenhagen Stock Exchange | 25200.0000 |
| Deutsche Boerse | 100000.0000 |
| Egyptian Exchange | 100000.0000 |
| Euronext Paris Exchange | 3.0000 |
| Frankfurt Stock Exchange | 10000.0000 |
| Hong Kong Stock Exchange | 20250.0000 |
| Johannesburg Stock Exchange | 73000.0000 |
| London Stock Exchange | 49000.0000 |
| Mothers | 16200.0000 |
| NASDAQ | 3.0000 |
| New York Stock Exchange | 11667.6667 |
| Saudi Stock Exchange | 22325.0000 |
| Shenzhen Stock Exchange | 435000.0000 |
| South Korea Stock Exchange | 100000.0000 |
| Swiss Stock Exchange | 415250.0000 |
| Taiwan Stock Exchange | 25766.6667 |
| Tehran Stock Exchange | 100000.0000 |
| Toronto Stock Exchange | 16750.0000 |

52. tutti i titoli di stato che hanno un interesse pari a quello massimo:

```
SELECT TitoloDiStato.Ticker, TitoloDiStato.interesse
FROM TitoloDiStato, (SELECT MAX(TitoloDiStato.interesse) interesse FROM TitoloDiStato ) C
WHERE TitoloDiStato.interesse = C.interesse;
```

| Ticker | interesse |
|--------|-----------|
| E102 | 7.30 |

53. Si vuole sapere la quantità di borse in ogni paese che ne ha almeno una, insieme al volume di scambi medio per ogni nazione:

```
SELECT COUNT(*) nBorse, Borsa.Nazione, AVG(Borsa.VolumeDiScambi) VolumeDiScambi
FROM Borsa
GROUP BY Borsa.Nazione;
```

| nBorse | Nazione | VolumeDiScambi |
|--------|--------------|-----------------|
| 1 | Algeria | 837320310.0000 |
| 1 | Angola | 982374102.0000 |
| 1 | Australia | 341928109.0000 |
| 1 | Canada | 123810939.0000 |
| 4 | China | 803975682.0000 |
| 1 | Denmark | 327419220.0000 |
| 1 | Egypt | 938102939.0000 |
| 1 | France | 293791820.0000 |
| 2 | Germany | 601603610.0000 |
| 1 | Iran | 127392102.0000 |
| 1 | Italy | 520200400.0000 |
| 4 | Japan | 1037363102.5000 |
| 1 | Netherlands | 320459000.0000 |
| 1 | Saudi Arabia | 283741209.0000 |
| 1 | South Africa | 209302128.0000 |
| 1 | South Korea | 923782100.0000 |
| 1 | switzerland | 236109288.0000 |
| 1 | Taiwan | 32871901.0000 |
| 1 | UK | 83741029.0000 |
| 2 | USA | 465215950.0000 |

54. Tutti gli strumenti derivati sovrastanti ad azioni di aziende che hanno almeno un dipendente che possiede la criptovaluta con ticker "ETH":

```
SELECT DISTINCT StrumentoDerivato.Ticker
FROM StrumentoDerivato,
(SELECT azione.ticker azioniDer FROM azione,azienda,impiegato,impiegatoPossiedeCriptovaluta
WHERE impiegatoPossiedeCriptovaluta.Ticker = 'ETH'
AND impiegato.CF = impiegatoPossiedeCriptovaluta.CF
AND Impiegato.Azienda = Azienda.ID
AND Azione.aziendaEmittente = Azienda.ID) C
WHERE StrumentoDerivato.azione = C.azioniDer;
```

| Ticker |
|--------|
| D019 |
| D031 |
| D006 |
| D008 |
| D028 |
| D009 |

55. si vogliono sapere le 10 commodity con il volume totale più alto ordinate per volume totale:

```
SELECT Commodity.Ticker, Commodity.VolumeTotale  
FROM Commodity  
ORDER BY Commodity.VolumeTotale;
```

| Ticker | VolumeTotale |
|--------|--------------|
| C222 | 670000 |
| C219 | 540000 |
| C206 | 450000 |
| C223 | 430000 |
| C205 | 400000 |
| C217 | 330000 |
| C210 | 300000 |
| C212 | 300000 |
| C218 | 290000 |
| C221 | 290000 |

56. Dire quali criptovalute hanno un flottante almeno pari alla media delle criptovalute:

```
SELECT criptovaluta.ticker, criptovaluta.flottante  
FROM criptovaluta, (SELECT VG(criptovaluta.flottante) valore FROM Criptovaluta) C  
WHERE C.valore <= criptovaluta.flottante;
```

| ticker | flottante |
|--------|---------------|
| AGE | 9009865433578 |
| DOGE | 1326710000000 |
| FAG | 971648950000 |
| GHJ | 9183746272991 |
| RTY | 949274017552 |

57. Quali sono i titoli di stato a più alto rendimento:

```
SELECT TitoloDiStato.ticker  
FROM titoloDiStato, ( SELECT MAX(TitoloDiStato.rendimento) rendMax FROM TitoloDiStato ) C  
WHERE titoloDiStato.rendimento = C.rendMax;
```

| ticker |
|--------|
| C102 |

58. Per ogni settore dire quante aziende, che vi operano, quotano le proprie azioni nell'indice azionario "SP500":

```
SELECT Opera.settore, COUNT(*) nAziende
FROM Opera, Azienda,
(SELECT Azione.AziendaEmittente Emittente
 FROM Contiene,Azione
 WHERE Contiene.TickerIndice = 'SP500'
 AND Contiene.TickerAzione = Azione.Ticker) C
WHERE Azienda.ID = C.Emittente
AND azienda.ID = Opera.Azienda
GROUP BY Opera.settore;
```

| settore | count(*) |
|------------|----------|
| e-commerce | 1 |
| technology | 1 |

59. Si vuole conoscere il mercato delle commodity in cui vengono scambiate più commodity:

```
SELECT Commodity.mercatoDelleCommodity, COUNT(*)
FROM commodity
GROUP BY Commodity.mercatoDelleCommodity
ORDER BY COUNT(*) DESC
LIMIT 1;
```

| mercatoDelleCommodity | count(*) |
|-------------------------------|----------|
| Commodity Market of Hong Kong | 4 |

60. Per ogni indice dire quali sono le borse nelle quali sono quotate le azioni che esso contiene:

```
SELECT Contiene.TickerIndice, AzioneQuotataIn.borsa
FROM AzioneQuotataIn, Azione, Contiene
WHERE AzioneQuotataIn.ticker = Azione.ticker
AND Azione.ticker = Contiene.tickerAzione
GROUP BY Contiene.tickerIndice, AzioneQuotataIn.borsa;
```

| TickerIndice | borsa |
|--------------|----------------------------------|
| SP500 | Hong Kong Stock Exchange |
| SP500 | Swiss Stock Exchange |
| SP500 | Toronto Stock Exchange |
| STOF | London Stock Exchange |
| NAC | Saudi Stock Exchange |
| GATTO | Copenhagen Stock Exchange |
| GATTO | Mother's |
| MARIO | Taiwan Stock Exchange |
| ESPN | New York Stock Exchange |
| VAMO | Toronto Stock Exchange |
| KAYT | Euronext Paris Exchange |
| KAYT | NASDAQ |
| KAYT | New York Stock Exchange |
| SAMS | Australia Stock Exchange |
| TTORA | Saudi Stock Exchange |
| PAC | China Financial Futures Exchange |
| PAC | Saudi Stock Exchange |
| PAC | Taiwan Stock Exchange |
| DATTO | Shenzhen Stock Exchange |
| RACE | Copenhagen Stock Exchange |
| LOTA | New York Stock Exchange |
| HAP | Amsterdam Stock Exchange |
| TORA | London Stock Exchange |
| TORA | Taiwan Stock Exchange |
| DJI | Johannesburg Stock Exchange |
| COCA | Deutsche Boerse |
| COCA | Egyptian Exchange |
| COCA | South Korea Stock Exchange |
| COCA | Tehran Stock Exchange |
| MICRO | Frankfurt Stock Exchange |
| COLA | Hong Kong Stock Exchange |
| ELLE | Saudi Stock Exchange |

61. Nazione la cui media dei fatturati delle sue aziende è massima:

```
SELECT Azienda.nazione, AVG(Azienda.fatturato) mediaFatturatoAziende
FROM Azienda GROUP BY Azienda.nazione
ORDER BY mediaFatturatoAziende DESC
LIMIT 1;
```

| nazione | mediaFatturatoAziende |
|---------|-----------------------|
| USA | 26575011384.615385 |

62. Si vogliono conoscere gli impiegati che lavorano in un'azienda di nazione diversa da quella della loro di residenza:

```
SELECT Impiegato.CF,Impiegato.nome, Impiegato.cognome, Impiegato.nazione nato, Azienda.nazione
lavora
FROM Impiegato,Azienda
WHERE Impiegato.azienda = Azienda.id
AND Impiegato.nazione != Azienda.nazione;
```

| CF | nome | cognome | nato | lavora |
|-------------|-----------|------------|-------------|-------------|
| ALCVRD2002 | Alice | Verdi | France | USA |
| BRKSN1995 | Brock | Sinclair | Russia | China |
| DNLWS1994 | Dean | Lewis | Australia | Japan |
| FRDMRRC1945 | Freddie | Mercury | UK | Japan |
| FRNCX2001 | Francesco | Xu | China | Germany |
| GPTRNGL2001 | Gaia | Petrangeli | Italy | USA |
| JLNJS2002 | Julie | Jones | USA | Italy |
| JMHNG1992 | Jim | Heung | South Korea | USA |
| JSTNWT1998 | Justin | White | Canada | Netherlands |
| LDVCND1948 | Ludovico | Einaudi | Italy | Japan |
| LNGLNG1980 | Lang | Lang | China | Japan |
| MRKBKN1999 | Mark | Brown | Spain | USA |
| MRRSS1990 | Mario | Rossi | UK | Italy |
| MRVNVT1994 | Marvin | Vettori | Italy | USA |
| MUSKELN1971 | Elon | Musk | Canada | USA |
| NDRNG1970 | Andrew | Ng | UK | USA |
| PNCPNL2003 | Pinco | Pallino | Germany | China |
| RMNNPRZ1998 | Ramon | Perez | Spain | USA |
| RSTXVL2000 | Orest | Xhaveli | Germany | Italy |
| SHWNL1985 | Shawn | Lee | Canada | Japan |
| SMNNMNC | Simone | Mancia | Italy | USA |
| SMNZHNG2001 | Simone | Zheng | China | USA |
| YNGZNG2000 | Ying | Zhang | China | USA |
| ZHOULC2000 | Luca | Zhou | China | Italy |

ALGEBRA RELAZIONALE

L'algebra relazionale è un linguaggio query procedurale in notazione algebrica. In una query, si applicano sequenzialmente le operazioni alle relazioni. Ogni operazione (unaria o binaria) riceve in input una relazione e ne produce un'altra in output.

Le operazioni primitive sono:

- Selezione (σ)
- Proiezione (π)
- Unione (U)
- Differenza Insiemistica (-)
- Prodotto Cartesiano (X)
- Ridenominazione (ρ)

Esistono altre operazioni da esse derivabili, tra cui l'intersezione insiemistica (\cap).

Di seguito troviamo alcune query sul nostro database scritte in Algebra Relazionale:

1. PER OGNI AZIENDA SI VOGLIONO SAPERE GLI ASSET POSSEDUTI, CON NOME AZIENDA E TICKER ASSET POSSEDUTO:

$$\rho_{asset<-ticker} \{ \pi_{nome,ticker} [\rho_{ID2 <- ID(Azienda)} | x | ID = ID2(AziendaPossiedeAsset)] \}$$

2. SI VOGLIONO SAPERE NOME E COGNOME DEGLI IMPIEGATI ITALIANI CHE LAVORANO NELLA "DREAM TEAM COMPANY":

$$\pi_{nome_imp, cognome} \{ \rho_{nome_imp <- nome} [\rho_{nazione='italy'} (Impiegato)] | x | Azienda = ID \rho_{nome='DreamTeam Company'}(Azienda) \}$$

3. NAZIONI CON UN PIL SUPERIORE A QUELLO DELL'ITALIA, CON IL LORO PIL:

$$\pi_{nome} \{ Nazione | x | pil > itapil [\rho_{nome='Italy'} \rho_{itapil = pil}(Nazione)] \}$$

4. SOCIETÀ AVENTE SEDE NELLA STESSA NAZIONE DELLA SOCIETÀ "NIKE", IN ORDINE CRESCENTE DI NUMERO DEGLI ASSET POSSEDUTI:

$$\pi_{ID, nome} \{ Azienda | x | nazione = nazioneNike [\rho_{nome='Nike'} \rho_{nazioneNike = nazione}(Azienda)] \}$$

5. PER OGNI BANCA SU VUOLE SAPERE LA VALUTA DEL PAESE IN CUI OPERA:

$$\pi_{nomeBanca, nome} \{ Nazione | x | Nome=nazione \pi_{ID, nome, nazione} (Banca) \}$$

6. PER OGNI STRUMENTO DERIVATO SI VUOLE SAPERE L'AZIENDA CHE HA EMESSO L'AZIONE SOTTOSTANTE:

$$\pi_{aziendaEmittente, tickerStrumento} \{ Azione | x | tickerAzione = ticker [\rho_{ticker = tickerStrumento, azione = tickerAzione} (Nazione) \pi_{ticker, azione} (StrumentoDerivato) \}$$

7. TUTTE LE AZIONI DI TIPO 'COMMON' CHE OPERANO NEL SETTORE TECHNOLOGY:

$$\rho_{classificazioneAzione='Common' \text{ and } Settore='Technology'} \{ Azione | x | id = aziendaEmittente \{ Azienda | x | id = aziendaOpera \}$$

CALCOLO RELAZIONALE

Il calcolo relazionale è un linguaggio query non procedurale ma dichiarativo. Invece dell'algebra, utilizza il calcolo dei predicati matematici del primo ordine in notazione logica. L'output di una query è una relazione che contiene solo tuple che soddisfano le formule logiche espresse. Il potere espressivo del calcolo relazionale è dunque equivalente a quello dell'algebra relazionale.

Versioni:

1. Calcolo relazionale sui domini
2. Calcolo relazionale sulle tuple con dichiarazione di range

Di seguito sono alcune query espresse tramite il calcolo relazionale sulle tuple con dichiarazione di range:

1. PER OGNI AZIENDA SI VOGLIONO SAPERE GLI ASSET POSSEDEDUTI, CON NOME AZIENDA E TICKER ASSET POSSEDUTO:

{ az.(nome), azpa.(ticker) | az(Azienda), azpa(AziendaPossiedeAsset) | az.ID = azpa.ID }

2. SI VOGLIONO SAPERE NOME E COGNOME DEGLI IMPIEGATI ITALIANI CHE LAVORANO NELLA 'DREAMTEAM COMPANY':

{ imp.(nome,cognome) | imp(Impiegato), az(Azienda) | imp.nazione="Italy" \wedge az.nome = "DreamTeam Company" \wedge az.ID = imp.azienda }

3. SOCIETÀ AVENTE SEDE NELLA STESSA NAZIONE DELLA SOCIETÀ "NIKE", IN ORDINE CRESCENTE DI NUMERO DEGLI ASSET POSSEDEDUTI:

{ i.(CF, azienda) | i(Impiegato), a(Azienda) | a.nome = 'Alibaba' \wedge i.nazione = a.nazione }

4. DIPENDENTI RESIDENTI NELLA STESSA NAZIONE IN CUI RISIEDE AZIENDA "ALIBABA" :

{ i.(ID, nome) | i(Azienda), i'(Azienda) | i.nazione = i'.nazione \wedge i'.nome = 'Nike' }

5. TUTTI GLI IMPIEGATI CHE LAVORANO IN ITALIA:

{i.* | i(Impiegato),j(Azienda) | i.azienda = j.ID \wedge j.Nazione = Italia'}

6. TUTTE LE AZIONI DI TIPO "COMMON"CHE OPERANO NEL SETTORE "TECHNOLOGY" :

{i.ticker | i(Azione), j(Azienda), k(Opera) | j.ID = k.Azienda \wedge i.ClassificaAzione = 'Common' \wedge i.AziendaEmittente = j.ID, k.Settore = 'Technology'}

Generazione casuale di tuple in Python: MySQL

Per rendere paragonabili i tempi di esecuzione delle query non ottimizzate con quelle ottimizzate, è stato necessario introdurre, nel database, un gran numero di record. Per velocizzare questo processo, è stato scritto uno script in Python che scrive tutti gli inserimenti generati casualmente, in un semplice file di testo.

```
import random
import string
import decimal

print("esecuzione iniziata")

f= open("1.txt","w+")
unique_IndiceAzione = [""]
symbols = 'ABCDEFGHIJKLMNPQRSTUVWXYZ1234567890'
randName = ''

print("indice Azionario")

for i in range(6000):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(3,8)))

    unique_IndiceAzione.append(randName)

    f.write("INSERT INTO IndiceAzione VALUES('"
           + randName + "','" + str(random.randint(1, 100000)) + "," +
           str(random.randint(1, 100000)) + ");")

    f.write("\n")
print("Indice Done ")

unique_Setore = [""]
symbols = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNPQRSTUVWXYZ1234567890'
randName = ''

print("Settore")

for i in range(100):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(4,12)))

    unique_Setore.append(randName)

    f.write("INSERT INTO Settore VALUES('"
           + randName + "','" + str(random.randint(1000000, 100000000000)) + ");")
```

```

f.write("\n")

print("Valuta")

unique_Valuta= ["""]
upper_symbols = 'ABCDEFGHIJKLMNPQRSTUVWXYZ'
lower_symbols = 'abcdefghijklmnopqrstuvwxyz'
randName = ''

for i in range(100):
    randName = ''.join(random.choice(upper_symbols) for i in
range(random.randint(2,4)))
    unique_Valuta.append(randName)
    randLongName = ''.join(random.choice(lower_symbols) for i in
range(random.randint(3,12)))

    f.write("INSERT INTO Valuta VALUES('"
        + randName + "," + """
        + randLongName +",'"
        +str(random.randint(1, 100000))
        + "," + str(random.randint(1, 100000))
        + ");")

f.write("\n")

print("Nazione")

unique_Nazione = ["""]
symbols = ' ABCDEFGHIJKLMNOPQRSTUVWXYZ'
randName = ''

for i in range(195):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(3,12)))
    unique_Nazione.append(randName)

    f.write("INSERT INTO Nazione VALUES('"
        + randName +",'"
        + str(random.randint(100000, 1000000000))+ "," #population
        + str(random.randint(1, 1000000))+ "," #pil
        + str(random.randint(1, 100))+ ","
        + str(random.randint(1, 100))+ ","
        + random.choice(unique_Valuta)
        +");")

f.write("\n")

```

```

print("Criptovaluta")

unique_Criptovaluta= ['']
symbols = 'ABCDEFGHIJKLMNPQRSTUVWXYZ1234567890'
randName = ''

for i in range(20000):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(2,7)))
    unique_Criptovaluta.append(randName)

    f.write("INSERT INTO Criptovaluta VALUES('"
            + randName + "','" + str(random.randint(1, 1000000)) + ");")

    f.write("\n")

f.close()
f= open("1.5.txt", "w+")
#####
#



print("Azienda")

symbols = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNPQRSTUVWXYZ1234567890'
randName = ''

for i in range(33, 20001):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(2,12)))
    utile = random.randint(1000, 1000000000)
    fatturato = utile + random.randint(1, 1000000000)
    f.write("INSERT INTO Azienda VALUES(" + str(i) + ","
            + "" + randName+ "','" + "" + random.choice(unique_Nazione) + "','" +
            str(utile) + ","
            + str(fatturato) + ","
            + str(random.randint(1, 1000000)) +
            ");")

    f.write("\n")
print("Azienda Done ")

f.close()
f= open("2.txt", "w+")
#####
#



print("Opera")

```

```

for i in range(33, 200001):
    f.write("INSERT INTO Opera VALUES(" + str(i) + ","
            + "''" + random.choice(unique_Settore) + "''"
            + ");")
    f.write("\n")
print("Opera Done ")

print("Impiegato")

unique_Impiegato= ["""]
upper_symbols = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890'
lower_symbols = 'abcdefghijklmnopqrstuvwxyz'
randName = ''

for i in range(250000):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(5,20)))
    unique_Impiegato.append(randName)

    f.write("INSERT INTO Impiegato VALUES('"
            + randName + "','"
            + "''" + random.choice(unique_Nazione) + "','""
            + ''.join(random.choice(lower_symbols) for i in
range(random.randint(2,12))) + "','""
            + ''.join(random.choice(lower_symbols) for i in
range(random.randint(2,12))) + "','""
            + str(random.randint(1, 200000))
            + ");")

    f.write("\n")
print("Impiegato Done ")

f.close()
f= open("3.txt","w+")
#####
#####

print("Banca")

symbols = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890& '
randName = ''

for i in range(10000, 60000):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(3,20)))

```

```

f.write("INSERT INTO Banca VALUES("
    + str(i)
    + "','" + randName + "','""
    + random.choice(unique_Nazione) + "','""
    + str(random.randint(0, 800000000)) + "," "#fatturato
    + str(random.randint(0, 800000000)) + "," "#capitalizzazione
    + str(random.randint(3, 60000))#dipendenti
    + ");")

f.write("\n")

print("MercatoCommodity")

unique_MercatoDelleCommodity= ["""]
symbols = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890 &
randName = ''

for i in range(1000):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(3,50)))
    unique_MercatoDelleCommodity.append(randName)

    f.write("INSERT INTO MercatoDelleCommodity VALUES('"
        + randName + "','"
        + str(random.randint(10,10000)) + ","
        + "" + random.choice(unique_Nazione)
        + "');")

f.write("\n")

print("Asset")

unique_Asset= ["""]
symbols = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890'
randName = ""

for i in range(110000):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(4,5)))
    unique_Asset.append(randName)

    f.write("INSERT INTO Asset VALUES('"
        + randName + "','"
        + str(random.randint(10,99999)) + ","
        + str(random.randint(10,99999)) + ","
        + str(random.randint(1000,2000000000))
        + "');")

    f.write("\n")
print("Asset Done ")

```

```

print("titolodistato")

unique_TitoloDiStato= ["""]
randName = ''

for i in range(20000):
    randName = ''.join(unique_Asset[i])
    unique_TitoloDiStato.append(randName)

    f.write("INSERT INTO TitoloDiStato VALUES('"
        + randName + "','""
        + str(random.randint(2023, 2040))+ "-" + str(random.randint(1, 12)) + "-"
        + str(random.randint(1, 31))
        + "','"' + random.choice(unique_Nazione) + "','"'
        + str(float(decimal.Decimal(random.randrange(0, 999))/100)) + ","
        + str(float(decimal.Decimal(random.randrange(0, 999))/100))
        + ");")

f.write("\n")

print("commodity")

unique_Commodity= ["""]
randName = random.choice(unique_Asset)

for i in range(20000):
    randName = ''.join(unique_Asset[20000 + i])
    unique_Commodity.append(randName)

    f.write("INSERT INTO Commodity VALUES('"
        + randName + "','"'
        + str(random.randint(1, 500000)) + "','"'
        + random.choice(unique_MercatoDelleCommodity)
        + "');")

f.write("\n")

unique_ClassificazioneAzione = ['Common',
'Preference','Growth','Value','Income','Blue Chip','Cyclic','Acyclic',
'Defensive','IPO','Penny Stock','ESG']

print("Azione")

```

```

unique_Azione= ["""]
randName = random.choice(unique_Asset)

for i in range(20000):
    randName = ''.join(unique_Asset[40000 + i])
    unique_Azione.append(randName)

    f.write("INSERT INTO Azione VALUES('"
        + randName + "','"
        + str(random.randint(1, 200000)) + ",\""
        + random.choice(unique_ClassificazioneAzione)
        + "')")

f.write("\n")

print("Obbligazione")


unique_Obbligazione = ["""]
randName = random.choice(unique_Asset)

for i in range(20000):
    randName = ''.join(unique_Asset[60000 + i])
    unique_Obbligazione.append(randName)

    f.write("INSERT INTO Obbligazione VALUES('"
        + randName + "','"
        + str(random.randint(1, 200000)) + ","
        + str(random.randint(2023, 2040))+ "-" + str(random.randint(1, 12)) +
"- " + str(random.randint(1, 31))
        + "','" + str(float(decimal.Decimal(random.randrange(0, 999))/100)) + ","
        + str(float(decimal.Decimal(random.randrange(0, 999))/100)))
        + ");")

f.write("\n")

print("strumento derivato")

unique_StrumentoDerivato= ["""]
randName = random.choice(unique_Asset)

for i in range(20000):
    randName = ''.join(unique_Asset[80000 + i])
    unique_StrumentoDerivato.append(randName)

    f.write("INSERT INTO StrumentoDerivato VALUES('"
        + randName + "','"
        + str(random.randint(1, 200000)) + ","
        + random.choice(unique_Azione)

```

```

+ "');")
f.write("\n")

f.close()
f= open("4.txt","w+")
#####
print("borsa")

unique_Borsa= ["""]
symbols = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890 &
randName = ''

for i in range(1010):
    randName = ''.join(random.choice(symbols) for i in range(random.randint(3,50)))
    unique_Borsa.append(randName)

    f.write("INSERT INTO Borsa VALUES('"
        + randName + "','"
        + str(random.randint(1, 200000000)) + ","
        + random.choice(unique_Nazione)
        + "')");

f.write("\n")

for i in range(20000):
    f.write("INSERT INTO AziendaPossiedeCriptovaluta VALUES("
        + str(random.randint(1, 200000)) + ","
        + random.choice(unique_Criptovaluta)
        + "')");

f.write("\n")

print("impoiegato possiede cripto")

for i in range(50000):
    f.write("INSERT INTO ImpiegatoPossiedeCriptovaluta VALUES('"
        + random.choice(unique_Impiegato) + "','"
        + random.choice(unique_Nazione) + "','"
        + random.choice(unique_Criptovaluta)
        + "')");

f.write("\n")

print("azienda possiede asset")

```

```
for i in range(150000):
    f.write("INSERT INTO AziendaPossiedeAsset VALUES("
        + str(random.randint(1, 200000)) + ",'"
        + random.choice(unique_Asset)
        + "');")

    f.write("\n")
print("AziendaPossiedeAsset Done ")

print("contiene")

for i in range(20000):
    f.write("INSERT INTO Contiene VALUES('"
        + random.choice(unique_IndiceAzionario) + "','"
        + random.choice(unique_Azione)
        + "');")

    f.write("\n")

print("azione quotata in")

for i in range(50000):
    f.write("INSERT INTO AzioneQuotataIn VALUES('"
        + random.choice(unique_Azione) + "','"
        + random.choice(unique_Borsa)
        + "');")

    f.write("\n")

print("obbligazione quotata in")

for i in range(50000):
    f.write("INSERT INTO ObbligazioneQuotataIn VALUES('"
        + random.choice(unique_Obbligazione) + "','"
        + random.choice(unique_Borsa)
        + "');")

    f.write("\n")

print("nazione possiede asset")

for i in range(5000):
    f.write("INSERT INTO NazionePossiedeAsset VALUES('"
        + random.choice(unique_Nazione) + "','"
        + random.choice(unique_Asset)
```

```
+ "');")  
  
f.write("\n")  
  
f.close()  
  
print("esecuzione terminata")
```

OTTIMIZZAZIONE

Di seguito abbiamo selezionato degli attributi su cui creare indici secondari per velocizzare l'esecuzione delle query. Ovviamente, non abbiamo creato troppi indici per una questione di costi di memoria. Gli indici occupano memoria e quindi abbiamo trovato un compromesso, creando indici solo per gli attributi più richiesti.

Creazione Indici Secondari in MySQL:

```
CREATE INDEX index_fatturato  
ON Azienda (fatturato);
```

```
CREATE INDEX index_volumeDiScambi  
ON Borsa (volumeDiScambi);
```

```
CREATE INDEX index_flottante  
ON Criptovaluta (flottante);
```

```
CREATE INDEX index_volumeDiScambi  
ON MercatoDelleCommodity (volumeDiScambi);
```

```
CREATE INDEX index_ricavatoComplessivo  
ON Settore (ricavatoComplessivo);
```

```
CREATE INDEX index_pil  
ON Nazione (pil);
```

```
CREATE INDEX index_popolazione  
ON Nazione (popolazione);
```

```
CREATE INDEX index_tassoPoverta  
ON Nazione (tassoPoverta);
```

```
CREATE INDEX index_rendimento  
ON TitoloDiStato (rendimento);
```

Utilizzando questi indici secondari, abbiamo la versione ottimizzata di alcune delle query descritte in precedenza, che vengono eseguite sui dati casuali generati dal programma Python. Riportiamo inoltre, la frazione di miglioramento temporale delle ottimizzazioni.

Formula del miglioramento: $100 * (\text{originale} - \text{nuovo}) / \text{originale}$

1. Per ogni azienda si vogliono sapere gli asset posseduti, con nome azienda e ticker asset posseduto:

```
SELECT SQL_NO_CACHE az.nome, azpa.ticker as asset FROM Azienda az, AziendaPossiedeAsset azpa WHERE az.ID = azpa.ID;
```

```
146172 rows in set, 1 warning (0,28 sec)

mysql> explain SELECT SQL_NO_CACHE az.nome, azpa.ticker as asset FROM Azienda az, AziendaPossiedeAsset azpa WHERE az.ID = azpa.ID;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | azpa | NULL | index | PRIMARY | ticker | 82 | NULL | 149302 | 100.00 | Using index |
| 1 | SIMPLE | az | NULL | eq_ref | PRIMARY | PRIMARY | 4 | DTP.azpa.ID | 1 | 100.00 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Dall'explain vediamo che l'ottimizzatore sceglie come tabella esterna "AziendaPossiedeAsset" e come tabella interna "Azienda". Questa è la scelta giusta, infatti, con l'algoritmo di nested loop join, va scelta come tabella esterna quella più selettiva, che in questo caso è proprio "AziendaPossiedeAsset" :

```
mysql> select count(*) from Azienda
    -> ;
+-----+
| count(*) |
+-----+
| 197943 |
+-----+
1 row in set (0,01 sec)

mysql> select count(*) from AziendaPossiedeAsset;
+-----+
| count(*) |
+-----+
| 148561 |
+-----+
1 row in set (0,12 sec)
```

Per quanto riguarda l'uso degli indici, con l'uso dell'algoritmo nested loop, per avere un join efficiente è necessario utilizzare un indice sulla tabella interna, sull'attributo di join che in questo caso è "Azienda.ID", cosa che l'ottimizzatore fa. Notiamo però che utilizza l'indice 'ticker' su AziendaPossiedeAsset, scelta che risulta sbagliata, al massimo si poteva scegliere di utilizzare l'indice su "AziendaPossiedeAsset.ID". Così facendo il tempo di esecuzione risulta questo:

```
SELECT SQL_NO_CACHE /*! STRAIGHT_JOIN */ az.nome, azpa.ticker as asset
FROM AziendaPossiedeAsset azpa FORCE INDEX(PRIMARY), Azienda az FORCE INDEX(PRIMARY)
WHERE az.ID = azpa.ID;
```

```
146172 rows in set, 1 warning (0,20 sec)

mysql> explain SELECT SQL_NO_CACHE /*! STRAIGHT_JOIN */ az.nome, azpa.ticker as asset FROM AziendaPossiedeAsset azpa FORCE INDEX(PRIMARY), Azienda az FORCE INDEX(PRIMARY) WHERE az.ID = azpa.ID;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | azpa | NULL | index | PRIMARY | PRIMARY | 86 | NULL | 149302 | 100.00 | Using index |
| 1 | SIMPLE | az | NULL | eq_ref | PRIMARY | PRIMARY | 4 | DTP.azpa.ID | 1 | 100.00 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

- Miglioramento = $100 * (0.28 - 0.20) / 0.28 = 28.5\%$

15. Si vogliono sapere quali nazioni hanno un tasso di povertà minore di quello della nazionalità di 'Elon Musk':

```
SELECT SQL_NO_CACHE Nazione.nome, Nazione.tassoPoverta
FROM Nazione, (select SQL_NO_CACHE tassoPoverta from Impiegato, Nazione where Impiegato.nome = 'Elon' and cognome = 'Musk' and Impiegato.nazione = Nazione.nome) as NazioneMusk
WHERE
Nazione.tassoPoverta < NazioneMusk.tassoPoverta;
```

```
24 rows in set, 1 warning (0,09 sec)

mysql> explain
-> SELECT SQL_NO_CACHE Nazione.nome, Nazione.tassoPoverta FROM Nazione, (select tassoPoverta from Impiegato, Nazione where Impiegato.nome = 'Elon' and cognome = 'Musk' and Impiegato.nazione = Nazione.nome) as NazioneMusk WHERE Nazione.tassoPoverta < NazioneMusk.tassoPoverta;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | Nazione | NULL | ALL | nazione | NULL | NULL | NULL | 234 | 100.00 | NULL |
| 1 | SIMPLE | Impiegato | NULL | ALL | PRIMARY | PRIMARY | 102 | DTP.Impiegato.nazione | 228307 | 1.00 | Using where; Using join buffer (hash join) |
| 1 | SIMPLE | Nazione | NULL | eq_ref | PRIMARY | PRIMARY | 4 | DTP.Nazione | 1 | 33.33 | Using where |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Ottimizzata:

Notiamo che abbiamo una selezione su Impiegato.nome e un'altra su Impiegato.cognome. In questo caso possiamo pesare all'aggiunta di un indice multicolumn sulla tabella Impiegato, sugli attributi 'nome' e 'cognome':

```
mysql> create index index_nome_cognome on Impiegato(nome,cognome);
Query OK, 0 rows affected (2,13 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Rieseguendo la query avremo questo risultato:

```
24 rows in set, 2 warnings (0,00 sec)

mysql> explain
   → SELECT SQL_NO_CACHE Nazione.nome, Nazione.tassoPoverta FROM Nazione, (select SQL_NO_CACHE tassoPoverta from Impiegato, Nazione where Impiegato.nome = 'Elon' and cognome = 'Musk' and Impiegato.nazione = Nazione.nome) as NazioneMusk
 WHERE Nazione.tassoPoverta < NazioneMusk.tassoPoverta;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | Impiegato | NULL | ref | nazione_index_nome_cognome | index_nome_cognome | 164 | const,const | 1 | 100.00 |
| 1 | SIMPLE | Nazione | NULL | eq_ref | PRIMARY,index_tassoPoverta | PRIMARY | 102 | DTP.Impiegato.nazione | 1 | 100.00 |
| 1 | SIMPLE | Nazione | NULL | ALL | index_tassoPoverta | NULL | NULL | NULL | 234 | 33.33 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Range checked for each record (index map: 0x4)
```

7. Elencare le criptovalute possedute dall'impiegato "Luca Zhou" e quelle possedute da Amazon :

```
SELECT SQL_NO_CACHE CONCAT(Impiegato.nome, " ", cognome) as nome, ipc.ticker
FROM ImpiegatoPossiedeCriptovaluta ipc, Impiegato
WHERE ipc.CF = Impiegato.CF AND Impiegato.nome = 'Luca' AND Impiegato.cognome = 'Zhou'
UNION
SELECT nome, apc.ticker FROM Azienda, AziendaPossiedeCriptovaluta apc
WHERE apc.ID = Azienda.ID AND nome = 'Amazon';
```

```
10 rows in set, 1 warning (0,04 sec)
```

```
mysql> explain
   → SELECT SQL_NO_CACHE CONCAT(Impiegato.nome, " ", cognome) as nome, ipc.ticker FROM ImpiegatoPossiedeCriptovaluta ipc, Impiegato WHERE ipc.CF = Impiegato.CF AND Impiegato.nome = 'Luca' AND Impiegato.cognome = 'Zhou' UNION SELECT
nome, apc.ticker FROM Azienda, AziendaPossiedeCriptovaluta apc WHERE apc.ID = Azienda.ID AND nome = 'Amazon';
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | ipc | NULL | ref | PRIMARY,index_nome_cognome | index_nome_cognome | 164 | const,const | 1 | 100.00 |
| 1 | PRIMARY | ipc | NULL | ref | PRIMARY | PRIMARY | 82 | DTP.Impiegato.CF | 1 | 100.00 |
| 2 | UNION | apc | NULL | ref | PRIMARY | PRIMARY | 82 | NULL | 18863 | 100.00 |
| 2 | UNION | Azienda | NULL | eq_ref | PRIMARY | PRIMARY | 4 | DTP.apc.ID | 1 | 10.00 |
| NULL | UNION RESULT | <union1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Ottimizzazione:

Notiamo che nella seconda select l'ottimizzatore sceglie come outer table la tabella Azienda scelta che risulta non ottimale considerando che AziendaPossiedeCriptovaluta è più selettiva. Da notare l'uso degli indici : index_nome_cognome, index_nome.

```
SELECT CONCAT(Impiegato.nome, " ", cognome) as nome, ipc.ticker
FROM ImpiegatoPossiedeCriptovaluta ipc, Impiegato USE INDEX (index_nome_cognome)
WHERE ipc.CF = Impiegato.CF AND Impiegato.nome = 'Luca' AND Impiegato.cognome = 'Zhou'
UNION
SELECT /*! STRAIGHT_JOIN */ nome, apc.ticker
FROM Azienda, AziendaPossiedeCriptovaluta apc
WHERE apc.ID = Azienda.ID AND nome = 'Amazon';
```

```
10 rows in set (0,00 sec)

mysql> explain
   → SELECT CONCAT(Impiegato.nome, " ", cognome) as nome, ipc.ticker FROM ImpiegatoPossiedeCriptovaluta ipc, Impiegato USE INDEX (index_nome_cognome) WHERE ipc.CF = Impiegato.CF AND Impiegato.nome = 'Luca' AND Impiegato.cognome = 'Zhou' UNION SELECT /*! STRAIGHT_JOIN */ nome, apc.ticker FROM Azienda FORCE INDEX (index_nome), AziendaPossiedeCriptovaluta apc WHERE apc.ID = Azienda.ID AND nome = 'Amazon';
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | Impiegato | NULL | ref | index_nome_cognome | index_nome_cognome | 164 | const,const | 1 | 100.00 |
| 1 | PRIMARY | ipc | NULL | ref | PRIMARY | PRIMARY | 82 | DTP.Impiegato.CF | 1 | 100.00 |
| 2 | UNION | Azienda | NULL | ref | index_nome | index_nome | 82 | const | 1 | 100.00 |
| 2 | UNION | apc | NULL | ref | PRIMARY | PRIMARY | 4 | DTP.Azienda.ID | 1 | 10.00 |
| NULL | UNION RESULT | <union1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

57. Elencare i titoli di stato a più alto rendimento:

```
SELECT SQL_NO_CACHE TitoloDiStato.ticker FROM TitoloDiStato, ( SELECT max(TitoloDiStato.rendimento) rendMax FROM TitoloDiStato ) C WHERE TitoloDiStato.rendimento = C.rendMax;
```

```
26 rows in set, 1 warning (0,13 sec)
```

```

mysql> explain
→ SELECT SQL_NO_CACHE TitoloDiStato.ticker FROM TitoloDiStato, ( SELECT max(TitoloDiStato.rendimento) rendMax FROM TitoloDiStato ) C WHERE TitoloDiStato.rendimento = C.rendMax;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | <derived2> | NULL | system | NULL | NULL | NULL | NULL | 1 | 100.00 |
| 1 | PRIMARY | TitoloDiStato | NULL | ALL | NULL | NULL | NULL | NULL | 19456 | 10.00 |
| 2 | DERIVED | TitoloDiStato | NULL | ALL | NULL | NULL | NULL | NULL | 19456 | 100.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Notiamo che per prendere il massimo rendimento non si utilizza alcun indice, questo comporta ad un full table scan.
Possiamo quindi forzare l'indice nella query derivata. Possiamo usare questo indice anche per la inner table.

```

SELECT SQL_NO_CACHE TitoloDiStato.ticker FROM TitoloDiStato FORCE INDEX(index_rendimento), ( SELECT max(TitoloDiStato.rendimento) rendMax FROM TitoloDiStato FORCE INDEX(index_rendimento) ) C WHERE C.rendMax = TitoloDiStato.rendimento;

```

```

26 rows in set, 1 warning (0.00 sec)
mysql> explain
→ SELECT SQL_NO_CACHE TitoloDiStato.ticker FROM TitoloDiStato FORCE INDEX(index_rendimento), ( SELECT max(TitoloDiStato.rendimento) rendMax FROM TitoloDiStato FORCE INDEX(index_rendimento) ) C WHERE C.rendMax = TitoloDiStato.rendimento;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | <derived2> | NULL | system | NULL | NULL | NULL | NULL | 1 | 100.00 |
| 1 | PRIMARY | TitoloDiStato | NULL | ref | index_rendimento | index_rendimento | 3 | const | 26 | 100.00 |
| 2 | DERIVED | TitoloDiStato | NULL | Select tables optimized away |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

25. Visualizzare le aziende con sede nella stessa nazione di "Nike" in ordine decrescente di asset posseduti:

```

mysql> SELECT X.ID, X.nome, COUNT(XX.ticker) as num_asset
   -> FROM Azienda X, AziendaPossiedeAsset XX, Azienda Y, AziendaPossiedeAsset YY
   -> WHERE X.ID = XX.ID
   -> AND Y.ID = YY.ID
   -> AND Y.nome = 'Nike'
   -> AND X.nazione = Y.nazione
   -> AND X.nome != 'Nike'
   -> GROUP BY XX.ID
   -> ORDER BY num_asset DESC;
+-----+-----+
| ID | nome      | num_asset |
+-----+-----+
| 10 | Coursera  | 156 |
| 11 | Pfizer    | 143 |
| 28 | Warner Bros | 143 |
| 6  | Apple      | 117 |
| 18 | Tesla      | 117 |
| 4  | Hawlet Packard | 104 |
| 12 | KFC        | 104 |
| 29 | Universal  | 104 |
| 1  | Amazon     | 91  |
| 22 | Esso       | 91  |
| 19 | WWE        | 52  |
| 26 | NASA       | 52  |
+-----+-----+
12 rows in set (0.31 sec)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | YY | NULL | index | PRIMARY | ticker | 82 | NULL | 133115 | 100.00 |
| 1 | SIMPLE     | Y  | NULL | eq_ref | PRIMARY,nazione | PRIMAY | 4 | dtp.YY.ID | 1 | 10.00 |
| 1 | SIMPLE     | X  | NULL | ref   | PRIMARY,nazione | nazione | 82 | dtp.Y.nazione | 988 | 90.00 |
| 1 | SIMPLE     | XX | NULL | ref   | PRIMARY,ticker | PRIMARY | 4 | dtp.X.ID | 6 | 100.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Ottimizzata: uso la join ansi-style e forzo X ad essere la tabella esterna nella join:

```

mysql> SELECT X.ID, X.nome, COUNT(XX.ticker) as num_asset
   -> FROM Azienda X STRAIGHT_JOIN AziendaPossiedeAsset XX ON X.ID = XX.ID,
   -> Azienda Y STRAIGHT_JOIN AziendaPossiedeAsset YY
   -> ON Y.ID = YY.ID
   -> WHERE Y.nome = 'Nike'
   -> AND X.nazione = Y.nazione
   -> AND X.nome != 'Nike'
   -> GROUP BY XX.ID
   -> ORDER BY num_asset DESC;
+-----+-----+
| ID | nome      | num_asset |
+-----+-----+
| 10 | Coursera  | 156 |
| 11 | Pfizer    | 143 |
| 28 | Warner Bros | 143 |
| 6  | Apple      | 117 |
| 18 | Tesla      | 117 |
| 4  | Hawlet Packard | 104 |
| 12 | KFC        | 104 |
| 29 | Universal  | 104 |
| 1  | Amazon     | 91  |
| 22 | Esso       | 91  |
| 19 | WWE        | 52  |
| 26 | NASA       | 52  |
+-----+-----+
12 rows in set (0.12 sec)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | Y  | NULL | ALL | PRIMARY,nazione | nazione | 82 | dtp.Y.nazione | 988 | 90.00 |
| 1 | SIMPLE     | X  | NULL | ref  | PRIMARY,ticker | PRIMARY | 4 | dtp.X.ID | 6 | 100.00 |
| 1 | SIMPLE     | YY | NULL | ref  | PRIMARY | PRIMARY | 4 | dtp.Y.ID | 6 | 100.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

- Miglioramento = $100 * (0.31 - 0.12) / 0.31 = 61.3\%$

27. Visualizzare l'azienda con maggior fatturato per ogni settore, riportando anche il settore e il fatturato:

```
mysql> SELECT Opera.settore, Azienda.ID, Azienda.nome, Azienda.fatturato
-> FROM Opera JOIN Azienda ON Opera.azienda = Azienda.ID
-> WHERE (Opera.settore, Azienda.fatturato) IN
->     (SELECT Opera.settore, MAX(Azienda.fatturato)
->      FROM Azienda, Opera
->      WHERE Azienda.ID = Opera.azienda
->      GROUP BY Opera.settore
->      )
-> ORDER BY Azienda.fatturato DESC;
```

```
117 rows in set (10.96 sec)
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|----|-------------|---------|------------|--------|-----------------|---------|-------------------|------|--------|-------------|--|
| 1 | PRIMARY | Opera | NULL | index | PRIMARY | settore | 82 | NULL | 195480 | 100.00 | Using index; Using temporary; Using filesort |
| 1 | PRIMARY | Azienda | NULL | eq_ref | PRIMARY | 4 | dtp.Opera.azienda | 1 | 100.00 | Using where | |
| 2 | SUBQUERY | Opera | NULL | index | PRIMARY,settore | settore | 82 | NULL | 195480 | 100.00 | Using index |
| 2 | SUBQUERY | Azienda | NULL | eq_ref | PRIMARY | 4 | dtp.Opera.azienda | 1 | 100.00 | NULL | |

Ottimizzata: trasformo la query nidificata in una equivalente ma derivata

```
mysql> SELECT Opera.settore, Azienda.ID, Azienda.nome, Azienda.fatturato
-> FROM Opera, Azienda FORCE INDEX (index_fatturato),
->      (SELECT Opera.settore as SETT, MAX(Azienda.fatturato) as MX
->       FROM Azienda, Opera
->       WHERE Azienda.ID = Opera.azienda
->       GROUP BY Opera.settore
->      ) AS X
-> WHERE Opera.azienda= Azienda.ID
-> AND Opera.settore = X.SETT
-> AND Azienda.fatturato = X.MX
-> ORDER BY Azienda.fatturato DESC;
```

```
117 rows in set (4.07 sec)
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|----|-------------|------------|------------|--------|-----------------|-----------------|---------|-----------------------|--------|----------|--|
| 1 | PRIMARY | <derived2> | NULL | ALL | NULL | NULL | NULL | NULL | 197621 | 100.00 | Using where; Using temporary; Using filesort |
| 1 | PRIMARY | Azienda | NULL | ref | index_fatturato | index_fatturato | 9 | X.MX | 1 | 100.00 | NULL |
| 1 | PRIMARY | Opera | NULL | eq_ref | PRIMARY,settore | PRIMARY | 86 | dtp.Azienda.ID,X.SETT | 1 | 100.00 | Using index |
| 2 | DERIVED | Azienda | NULL | index | PRIMARY | index_fatturato | 9 | NULL | 197621 | 100.00 | Using index; Using temporary |
| 2 | DERIVED | Opera | NULL | ref | PRIMARY,settore | PRIMARY | 4 | dtp.Azienda.ID | 1 | 100.00 | Using index |

- Miglioramento = $100 * (10.96 - 4.07) / 10.96 = 62.9\%$

28. Visualizzare i settori in ordine decrescente di somma di fatturato delle sue aziende:

```
mysql> SELECT Opera.settore, SUM(Azienda.fatturato) as tot_revenue
-> FROM Opera JOIN Azienda ON Opera.azienda = Azienda.ID
-> GROUP BY Opera.settore
-> ORDER BY tot_revenue DESC;
```

```
117 rows in set (4.92 sec)
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|----|-------------|---------|------------|--------|-----------------|---------|-------------------|------|--------|----------|--|
| 1 | SIMPLE | Opera | NULL | index | PRIMARY,settore | settore | 82 | NULL | 195480 | 100.00 | Using index; Using temporary; Using filesort |
| 1 | SIMPLE | Azienda | NULL | eq_ref | PRIMARY | 4 | dtp.Opera.azienda | 1 | 100.00 | NULL | |

Ottimizzata? Forzo Opera ad essere la tabella esterna nella join

```
mysql> SELECT Opera.settore, SUM(Azienda.fatturato) as tot_revenue
-> FROM Opera STRAIGHT_JOIN Azienda ON Opera.azienda = Azienda.ID
-> GROUP BY Opera.settore
-> ORDER BY tot_revenue DESC
-> ;
```

```
117 rows in set (4.91 sec)
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|----|-------------|---------|------------|--------|-----------------|---------|---------|-------------------|--------|----------|--|
| 1 | SIMPLE | Opera | NULL | index | PRIMARY,settore | settore | 82 | NULL | 195480 | 100.00 | Using index; Using temporary; Using filesort |
| 1 | SIMPLE | Azienda | NULL | eq_ref | PRIMARY | PRIMARY | 4 | dtp.Opera.azienda | 1 | 100.00 | NULL |

In questo caso, vediamo che l'ottimizzatore aveva già pianificato tale ordine di join, quindi il piano di esecuzione è rimasto invariato. Non c'è stato alcun miglioramento.

37. Visualizzare le aziende nel settore di "Samsung" ordinate per fatturato, riportando anche il settore:

```
mysql> SELECT X.ID, X.nome, X.fatturato, XX.settore
-> FROM Azienda X, Opera XX, Azienda Y, Opera YY
-> WHERE X.ID = XX.azienda
-> AND Y.ID = YY.azienda
-> AND Y.nome = 'Samsung'
-> AND XX.settore = YY.settore
-> ORDER BY X.fatturato DESC;
```

| ID | nome | fatturato | settore |
|----|-------------------|-----------------|------------|
| 6 | Apple | 119344021000.00 | technology |
| 18 | Tesla | 70000321000.00 | technology |
| 1 | Amazon | 70035600000.00 | technology |
| 5 | Huawei | 21912600000.00 | technology |
| 7 | Samsung | 16732001000.00 | technology |
| 4 | Hawlet Packard | 7800531000.00 | technology |
| 30 | Ubisoft | 3392880000.00 | technology |
| 3 | Asus | 1300500000.00 | technology |
| 2 | Sony | 10560000.00 | technology |
| 8 | DreamTeam Company | 1200500.00 | technology |

```
10 rows in set (4.90 sec)
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|----|-------------|-------|------------|--------|-----------------|---------|---------|----------------|--------|----------|--|
| 1 | SIMPLE | YY | NULL | index | PRIMARY,settore | settore | 82 | NULL | 195480 | 100.00 | Using index; Using temporary; Using filesort |
| 1 | SIMPLE | Y | NULL | eq_ref | PRIMARY | PRIMARY | 4 | dtp.YY.azienda | 1 | 10.00 | Using where |
| 1 | SIMPLE | XX | NULL | ref | PRIMARY,settore | settore | 82 | dtp.YY.settore | 1761 | 100.00 | Using index |
| 1 | SIMPLE | X | NULL | eq_ref | PRIMARY | PRIMARY | 4 | dtp.XX.azienda | 1 | 100.00 | NULL |

Ottimizzata: forzo Y ad essere la tabella esterna dato che Y.nome = "Samsung" è estremamente selettiva

```

mysql> SELECT X.ID, X.nome, X.fatturato, XX.settore
    -> FROM Azienda X JOIN Opera XX ON X.ID = XX.azienda,
    -> Azienda Y STRAIGHT_JOIN Opera YY ON Y.ID = YY.azienda
    -> WHERE Y.nome = 'Samsung'
    -> AND XX.settore = YY.settore
    -> ORDER BY X.fatturato DESC;
+----+-----+-----+
| ID | nome      | fatturato | settore |
+----+-----+-----+
| 6  | Apple     | 119344021000.00 | technology |
| 18 | Tesla     | 70500321000.00 | technology |
| 1  | Amazon    | 70035600000.00 | technology |
| 5  | Huawei    | 21912600000.00 | technology |
| 7  | Samsung   | 16732001000.00 | technology |
| 4  | Hawlet Packard | 7800531000.00 | technology |
| 30 | Ubisoft   | 3392880000.00 | technology |
| 3  | Asus       | 13005000000.00 | technology |
| 2  | Sony       | 10560000.00    | technology |
| 8  | DreamTeam Company | 1200500.00    | technology |
+----+-----+-----+
10 rows in set (0.18 sec)

```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|----|-------------|-------|------------|--------|-----------------|---------|---------|----------------|--------|----------|--|
| 1 | SIMPLE | Y | NULL | ALL | PRIMARY | NULL | NULL | NULL | 197621 | 10.00 | Using where; Using temporary; Using filesort |
| 1 | SIMPLE | YY | NULL | ref | PRIMARY,settore | PRIMARY | 4 | dtp.Y.ID | 1 | 100.00 | Using index |
| 1 | SIMPLE | XX | NULL | ref | PRIMARY,settore | settore | 82 | dtp.YY.settore | 1761 | 100.00 | Using index |
| 1 | SIMPLE | X | NULL | eq_ref | PRIMARY | PRIMARY | 4 | dtp.XX.azienda | 1 | 100.00 | NULL |

- Miglioramento = $100 * (4.90 - 0.18) / 4.90 = 96.3\%$

50. Si vogliono sapere i mercati delle commodity nelle quali sono scambiate le commodity che hanno un nome che inizia con la lettera "C":

```

SELECT Commodity.MercatoDelleCommodity
FROM Commodity
WHERE Commodity.ticker like "C%";
```

```

SELECT Commodity.MercatoDelleCommodity
FROM Commodity FORCE INDEX(Primary)
WHERE Commodity.ticker like "C%";
```

```

mysql> show profiles;
+----+-----+-----+
| Query_ID | Duration | Query
+----+-----+-----+
| 1 | 0.00216375 | SELECT Commodity.MercatoDelleCommodity FROM Commodity WHERE Commodity.ticker like "C%" |
| 2 | 0.00157200 | SELECT Commodity.MercatoDelleCommodity FROM Commodity FORCE INDEX(Primary) WHERE Commodity.ticker like "C%" |
+----+-----+-----+
```

- Miglioramento = $100 * (0.00216 - 0.00157) / 0.00216 = 27.3\%$

51. si vuole sapere, per ogni borsa, il numero di dipendenti medio delle aziende le cui azioni sono scambiate su quella borsa:

```

SELECT AzioneQuotataIn.borsa, avg(Azienda.numeroDipendenti) mediaDipendenti
FROM AzioneQuotataIn, Azione, Azienda
WHERE azione.AziendaEmittente = Azienda.ID and Azione.ticker = azioneQuotataIn.ticker
GROUP BY AzioneQuotataIn.borsa;
```

```

SELECT /*! STRAIGHT_JOIN */ AzioneQuotataIn.borsa, avg(Azienda.numeroDipendenti) mediaDipendenti
FROM AzioneQuotataIn USE INDEX(), Azione USE INDEX(), Azienda FORCE INDEX(Primary)
WHERE azioneQuotataIn.ticker = Azione.ticker and azione.AziendaEmittente = Azienda.ID
GROUP BY AzioneQuotataIn.borsa;
```

```

Query_ID: 7
Duration: 0.32962500
  Query: SELECT AzioneQuotataIn.borsa, avg(Azienda.numeroDipendenti) mediaDipendenti
FROM AzioneQuotataIn, Azione, Azienda
WHERE azione.AziendaEmissente = Azienda.ID and Azione.ticker = azioneQuotataIn.ticker GROUP BY AzioneQuotataIn.borsa
***** 5. row *****
Query_ID: 8
Duration: 0.28790650
  Query: SELECT /*! STRAIGHT_JOIN */ AzioneQuotataIn.borsa, avg(Azienda.numeroDipendenti) mediaDipendenti
FROM AzioneQuotataIn USE INDEX(), Azione USE INDEX(), Azienda FORCE INDEX(Primary)
WHERE azioneQuotataIn.ticker = Azione.ticker and azione.AziendaEmissente = Azienda.ID GROUP BY AzioneQuotataIn.borsa

```

- Miglioramento = $100 * (0.3296 - 0.2879) / 0.3296 = 12.6\%$

52. tutti i titoli di stato che hanno un interesse pari a quello massimo:

```

SELECT TitoloDiStato.Ticker, TitoloDiStato.interesse
FROM TitoloDiStato, (SELECT max(TitoloDiStato.interesse) interesse FROM TitoloDiStato ) C WHERE
TitoloDiStato.interesse = C.interesse;

```

```

SELECT TitoloDiStato.Ticker, TitoloDiStato.interesse FROM TitoloDiStato FORCE INDEX(index_interesse), (SELECT
max(TitoloDiStato.interesse) interesse FROM TitoloDiStato FORCE INDEX(index_interesse) ) C
WHERE TitoloDiStato.interesse = C.interesse;

```

```

Query_ID: 9
Duration: 0.00391700
  Query: SELECT TitoloDiStato.Ticker, TitoloDiStato.interesse FROM TitoloDiStato, (SELECT max(TitoloDiStato.interesse)
interesse FROM TitoloDiStato ) C WHERE TitoloDiStato.interesse = C.interesse
***** 7. row *****
Query_ID: 10
Duration: 0.00059025
  Query: SELECT TitoloDiStato.Ticker, TitoloDiStato.interesse FROM TitoloDiStato FORCE INDEX(index_interesse), (SELECT
max(TitoloDiStato.interesse) interesse FROM TitoloDiStato FORCE INDEX(index_interesse) ) C WHERE TitoloDiStato.interesse
= C.interesse

```

- Miglioramento = $100 * (0.0039 - 0.00059) / 0.0039 = 84.9\%$

53. Si vuole sapere la quantità di borse in ogni paese che ne ha almeno una, insieme al volume di scambi medio per ogni paese:

```

SELECT count(*) nBorse, Borsa.Nazione, avg(Borsa.VolumeDiScambi) VolumeDiScambi
FROM Borsa
GROUP BY Borsa.Nazione;

```

```

SELECT count(*) nBorse, Borsa.Nazione, avg(Borsa.VolumeDiScambi) VolumeDiScambi
FROM Borsa USE INDEX()
GROUP BY Borsa.Nazione;

```

```

Query_ID: 11
Duration: 0.01678700
  Query: SELECT count(*) nBorse, Borsa.Nazione, avg(Borsa.VolumeDiScambi) VolumeDiScambi FROM Borsa GROUP BY Borsa.Nazi
one
***** 9. row *****
Query_ID: 81
Duration: 0.00271600
  Query: SELECT count(*) nBorse, Borsa.Nazione, avg(Borsa.VolumeDiScambi) VolumeDiScambi
FROM Borsa USE INDEX()
GROUP BY Borsa.Nazione
15 rows in set, 1 warning (0.00 sec)

```

- Miglioramento = $100 * (0.0168 - 0.00271) / 0.0168 = 83,86\%$

60. per ogni indice dire quali sono le borse nelle quali sono quotate le azioni che esso contiene:

```
SELECT contiene.tickerindice,azionequotatain.borsa
FROM azionequotatain, azione, contiene
WHERE azionequotatain.ticker = azione.ticker and azione.ticker = contiene.tickerazione
GROUP BY contiene.tickerindice,azionequotatain.borsa;
```

```
SELECT /*! STRAIGHT_JOIN*/ Contiene.TickerIndice, AzioneQuotataln.borsa
FROM azione use index(), contiene use index(), azionequotatain use index()
WHERE AzioneQuotataln.ticker = Azione.ticker
AND Azione.ticker = Contiene.tickerAzione
GROUP BY Contiene.tickerIndice, AzioneQuotataln.borsa;
```

```
***** 1. row *****
Query_ID: 1
Duration: 0.62269775
    Query: SELECT contiene.tickerindice,azionequotatain.borsa
FROM azionequotatain, azione, contiene
WHERE azionequotatain.ticker = azione.ticker and azione.ticker = contiene.tickerazione
GROUP BY contiene.tickerindice,azionequotatain.borsa
... 15 rows in set, 1 warning (0.00 sec)

Query_ID: 60
Duration: 0.39284175
    Query: SELECT SQL_NO_CACHE /*!STRAIGHT_JOIN*/ contiene.tickerindice,azionequotatain.borsa
FROM contiene FORCE INDEX(index_tickerAzione_tickerIndice), azionequotatain FORCE INDEX(primary)
WHERE azionequotatain.ticker = contiene.tickerazione
GROUP BY contiene.tickerindice,azionequotatain.borsa
15 rows in set, 1 warning (0.00 sec)
```

- Miglioramento = $100 * (0.622 - 0.392) / 0.622 = 23\%$

61. nazione il cui media dei fatturati delle sue aziende è massima:

```
SELECT azienda.nazione, avg(azienda.fatturato) mediaFatturatoAziende
FROM azienda
GROUP BY azienda.nazione
ORDER BY mediaFatturatoAziende DESC
LIMIT 1;
```

```
SELECT azienda.nazione, avg(azienda.fatturato) mediaFatturatoAziende
FROM azienda FORCE INDEX (nazione)
GROUP BY azienda.nazione
ORDER BY mediaFatturatoAziende DESC
LIMIT 1;
```

```
Query_ID: 15
Duration: 0.89563725
    Query: SELECT azienda.nazione, avg(azienda.fatturato) mediaFatturatoAziende FROM azienda GROUP BY azienda.nazione ORDER BY mediaFatturatoAziende DESC LIMIT 1
***** 13. row *****
Query_ID: 16
Duration: 0.48147575
    Query: SELECT azienda.nazione, avg(azienda.fatturato) mediaFatturatoAziende FROM azienda FORCE INDEX (nazione) GROUP BY azienda.nazione ORDER BY mediaFatturatoAziende DESC LIMIT 1
```

- Miglioramento = $100 * (0.896 - 0.481) / 0.896 = 46.3\%$

62. tutti gli impiegati che lavorano in un'azienda di nazionalità diversa da quella loro di residenza:

```
SELECT Impiegato.CF,Impiegato.nome, Impiegato.cognome, Impiegato.nazione nato, Azienda.nazione lavora
FROM Impiegato,Azienda
WHERE Impiegato.azienda = Azienda.id and Impiegato.nazione != Azienda.nazione;
```

```
SELECT Impiegato.CF,Impiegato.nome, Impiegato.cognome, Impiegato.nazione nato, Azienda.nazione lavora
FROM Impiegato STRAIGHT_JOIN Azienda
WHERE Impiegato.azienda = Azienda.id AND Impiegato.nazione != Azienda.nazione;
```

```
14. row
Query_ID: 17
Duration: 1.24599000
  Query: SELECT Impiegato.CF,Impiegato.nome, Impiegato.cognome, Impiegato.nazione nato, Azienda.nazione lavora FROM Impiegato,Azienda WHERE Impiegato.azienda = Azienda.id and Impiegato.nazione != Azienda.nazione
***** 15. row *****
Query_ID: 18
Duration: 0.84110800
  Query: SELECT Impiegato.CF,Impiegato.nome, Impiegato.cognome, Impiegato.nazione nato, Azienda.nazione lavora FROM Impiegato STRAIGHT_JOIN Azienda WHERE Impiegato.azienda = Azienda.id AND Impiegato.nazione != Azienda.nazione
```

- Miglioramento = $100 * (1.246 - 0.841) / 1.246 = 32.5\%$

49. Si vogliono sapere quali criptovalute sono possedute dalle aziende che detengono le azioni emesse dall'azienda "Amazon":

```
SELECT DISTINCT AziendaPossiedeCriptovaluta.Ticker
FROM Azione, AziendaPossiedeCriptovaluta, AziendaPossiedeAsset, Asset, Azienda
WHERE Azienda.nome = 'Amazon'
AND Azione.AziendaEmittente = Azienda.ID
AND Asset.ticker = Azione.ticker
AND asset.ticker = aziendaPossiedeAsset.ticker
AND AziendaPossiedeAsset.ID = AziendaPossiedeCriptovaluta.ID;
```

```
SELECT /*! STRAIGHT_JOIN*/ AziendaPossiedeCriptovaluta.Ticker
FROM Azienda FORCE INDEX(nomeAzienda), Azione FORCE INDEX(aziendaEmittente), Asset FORCE
INDEX(primary), AziendaPossiedeAsset FORCE INDEX(ticker), AziendaPossiedeCriptovaluta FORCE INDEX(primary)
WHERE Azienda.nome = 'Amazon'
AND Azienda.ID = Azione.AziendaEmittente
AND Azione.ticker = Asset.ticker
AND asset.ticker = aziendaPossiedeAsset.ticker
AND AziendaPossiedeAsset.ID = AziendaPossiedeCriptovaluta.ID;
***** 1. row *****
Query_ID: 1
Duration: 0.04474600
  Query: SELECT DISTINCT AziendaPossiedeCriptovaluta.Ticker
FROM Azione, AziendaPossiedeCriptovaluta, AziendaPossiedeAsset, Asset, Azienda
WHERE Azienda.nome = 'Amazon'
AND Azione.AziendaEmittente = Azienda.ID
AND Asset.ticker = Azione.ticker
AND asset.ticker = aziendaPossiedeAsset.ticker
AND AziendaPo
***** 2. row *****
Query_ID: 2
Duration: 0.00061875
  Query: SELECT /*! STRAIGHT_JOIN*/ AziendaPossiedeCriptovaluta.Ticker
FROM Azienda FORCE INDEX(nomeAzienda), Azione FORCE INDEX(aziendaEmittente), Asset FORCE INDEX(primary), AziendaPossiedeAsset FORCE INDEX(ticker), AziendaPossiedeCriptovaluta FORCE INDEX(primary)
WHERE Azienda.nome = 'Amazon'
AND Azienda
2 rows in set, 1 warning (0.00 sec)
```

- Miglioramento = $100 * (0.04475 - 0.00062) / 0.04475 = 98.6\%$

Per alcune query l'ottimizzazione consiste nell'obbligare MySQL a non usare indici, questo perché se si accede ad un grande numero di righe di una certa tabella o la tabella è molto piccola è conveniente fare una scansione lineare di questa, accedendovi quindi in Full Table Scan.

TRIGGER

I trigger fanno parte del DDL (Data Definition Language), essi seguono il principio ECA, ovvero Event-Condition-Action. Solitamente, un trigger si può attivare prima o dopo un inserimento e hanno 2 livelli di granularità:

1. attivarsi per ogni tupla
2. attivarsi per ogni istruzione DML

Nello specifico in MySQL i trigger operano a livello di riga e si ammette un solo trigger per tabella. Osserviamo inoltre che questi vengono usati per mantenere constraint di ogni tipo, in primis il vincolo di integrità referenziale.

Quelli di seguito sono una serie di trigger di esempio necessari per mantenere una serie di vincoli nel nostro database

1. Il tasso di povertà di ogni nazione è un valore decimale tra 0 e 100 che indica una percentuale.

```
DELIMITER //
Create Trigger tassoPovertaTrigger BEFORE INSERT ON Nazione FOR EACH ROW
BEGIN
    IF NEW.tassoPoverta > 100 THEN SET NEW.tassoPoverta = 100;
    END IF;
    IF NEW.tassoPoverta < 0 THEN SET NEW.tassoPoverta = 0;
    END IF;
END //
```

```
mysql> Create Trigger tassoPovertaTrigger BEFORE INSERT ON Nazione FOR EACH ROW
-> BEGIN
->     IF NEW.tassoPoverta > 100 THEN SET NEW.tassoPoverta = 100;
->     END IF;
->     IF NEW.tassoPoverta < 0 THEN SET NEW.tassoPoverta = 0;
->     END IF;
-> END //
```

In azione: aggiungo un indice azionario con valore nominale negativo:

```
INSERT INTO IndiceAzione VALUES('IXP1000', -9, 16);
mysql> INSERT INTO IndiceAzione VALUES('IXP1000', -9, 16);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT* FROM indiceAzione WHERE ticker = 'IXP1000';
+-----+-----+-----+
| ticker | valoreNominal | prezzoEmissione |
+-----+-----+-----+
| IXP1000 | 0.00 | 16.00 |
+-----+-----+-----+
```

Lo stesso tipo di trigger va creato per tutti gli altri campi che rappresentano una percentuale.

- Nazione.tassoOccupazione
- TitoloDiStato.interesse
- TitoloDiStato.rendimento
- Obbligazione.interesse
- Obbligazione.rendimento

2. Il valore nominale di un indice azionario non può assumere un valore negativo.

```
DELIMITER //
Create Trigger valoreNominaleTrigger BEFORE INSERT ON IndiceAzione FOR EACH ROW
BEGIN
    IF NEW. valoreNominale < 0 THEN SET NEW. valoreNominale = 0;
    END IF;
END //
```

```
mysql> DELIMITER //
mysql> Create Trigger valoreNominaleTrigger BEFORE INSERT ON IndiceAzione FOR EACH ROW
-> BEGIN
-> IF NEW. valoreNominale < 0 THEN SET NEW. valoreNominale = 0;
-> END IF;
-> END //
```

In azione: inserisco una nazione con tasso di povertà maggiore di 1 (ossia > 100%):

```
INSERT INTO Nazione VALUES ('Vatican City', 1000, 3100, 101, 92, 'EUR');
mysql> INSERT INTO Nazione VALUES ('Vatican City', 1000, 3100, 101, 92, 'EUR');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT* FROM Nazione WHERE nome = 'Vatican City';
+-----+-----+-----+-----+-----+
| nome | popolazione | pil | tassoPoverta | tassoOccupazione | valuta |
+-----+-----+-----+-----+-----+
| Vatican City | 1000 | 3100 | 100.00 | 1.00 | EUR |
+-----+-----+-----+-----+-----+
```

Lo stesso tipo di trigger è definito per tutti gli altri campi del database che devono assumere valori non negativi:

- IndiceAzione.prezzoEmissione
- Settore.ricavatoComplessivo
- Valuta.circolante
- Valuta.tassoDiCambioEuro
- Nazione.popolazione
- Nazione.pil
- Criptovaluta.flottante
- Azienda.utile
- Azienda.fatturato
- Azienda.numeroDipendenti
- Banca.fatturato
- Banca.capitalizzazione
- Banca.numeroDipendenti
- MercatoDelleCommodity.volumeDiScambi
- Asset.valoreNominale
- Asset.prezzoEmissione
- Asset.flottante
- Commodity.volumeTotale
- Azione.valoreNominale
- Azione.prezzoEmissione
- Azione.numeroAzioni
- Borsa.volumeDiScambi

3. Se il numero dei possessori di una criptovaluta supera il flottante della stessa criptovaluta, l'insert deve essere annullato.

```
DELIMITER //
Create Trigger possessoriTrigger BEFORE INSERT ON AziendaPossiedeCriptovaluta
FOR EACH ROW
BEGIN
    IF
        (SELECT COUNT(*) FROM AziendaPossiedeCriptovaluta X WHERE X.ticker = NEW.ticker) +
        (SELECT COUNT(*) FROM ImpiegatoPossiedeCriptovaluta X WHERE X.ticker = NEW.ticker)
    >=
        (SELECT flottante FROM Criptovaluta WHERE Criptovaluta.ticker = NEW.ticker)
    THEN signal sqlstate '45000';
    END IF;
END //
```

```
mysql> DELIMITER //
mysql> Create Trigger possessoriTrigger BEFORE INSERT ON AziendaPossiedeCriptovaluta
-> FOR EACH ROW
-> BEGIN
->     IF (SELECT COUNT(*) FROM AziendaPossiedeCriptovaluta X WHERE X.ticker = NEW.ticker) +
->         (SELECT COUNT(*) FROM ImpiegatoPossiedeCriptovaluta X WHERE X.ticker = NEW.ticker) >=
->             (SELECT flottante FROM Criptovaluta WHERE Criptovaluta.ticker = NEW.ticker)
->     THEN signal sqlstate '45000';
->     END IF;
-> END //
```

Lo stesso tipo di trigger va creato anche per tutte le altre relazioni di possessione di qualche asset se l'asset ha un numero limitato di disponibilità.

4. L'utile di un'azienda deve essere al più pari al suo fatturato.

```
DELIMITER //
Create Trigger utileTrigger BEFORE INSERT ON azienda FOR EACH ROW
BEGIN
    IF NEW.utile > NEW.fatturato THEN SET NEW.utile = NEW.fatturato;
    END IF;
END //
```

```
mysql> DELIMITER //
mysql> Create Trigger utileTrigger BEFORE INSERT ON azienda FOR EACH ROW
-> BEGIN
->     IF NEW.utile > NEW.fatturato THEN SET NEW.utile = NEW.fatturato;
->     END IF;
-> END //
```

In azione: inserisco una azienda che ha fatturato minore dell'utile:

```
INSERT INTO Azienda VALUES (35,'Leonardo','Italy',10000000,1000000,30500);
```

```
mysql> INSERT INTO Azienda VALUES (35, 'Leonardo', 'Italy', 10000000,1000000, 30500 );
```

```
mysql> SELECT * FROM Azienda WHERE nome = 'Leonardo';
+----+-----+-----+-----+
| ID | nome | nazione | utile | fatturato | numeroDipendenti |
+----+-----+-----+-----+
| 35 | Leonardo | Italy | 1000000.00 | 1000000.00 | 30500 |
+----+-----+-----+-----+
```

SICUREZZA

Ovviamente in un database scolastico devono essere presenti diverse tipologie di utenti con diversi diritti, nella nostra modellizzazione della realtà, infatti, abbiamo definito tre classi di utenti: un amministratore che ha tutti i diritti, l'insegnante che può aggiungere righe e fare query e, infine, lo studente che può solamente fare query. Inoltre, si è definito un quarto utente che ha accesso solamente a delle view in modalità lettura, questo perché non gli si vuole dare accesso alle tabelle originali per questioni di sicurezza. Ovviamente la creazione di questo ultimo utente ha il solo fine dimostrativo e non sarebbe effettivamente inserito in un progetto reale.

Le view sono tabelle che non memorizzano dati, esse condividono lo stesso spazio delle tabelle originali. Spesso vengono assegnate ad altri utenti con specifici campi oscurati anche se il loro utilizzo inappropriato può portare all'inconsistenza del database.

VIEWS

1. Nazioni in ordine descrescente di numero di asset posseduti

```
CREATE VIEW AssetsPerCountry AS
    SELECT Nazione.nome, COUNT(*) AS num_assets
    FROM Nazione STRAIGHT_JOIN NazionePossiedeAsset
    ON Nazione.nome = NazionePossiedeAsset.nazione
    GROUP BY NazionePossiedeAsset.nazione
    ORDER BY num_assets DESC;
```

```
mysql> CREATE VIEW AssetsPerCountry AS
    -> SELECT Nazione.nome, COUNT(*) AS num_assets
    -> FROM Nazione STRAIGHT_JOIN NazionePossiedeAsset
    -> ON Nazione.nome = NazionePossiedeAsset.nazione
    -> GROUP BY NazionePossiedeAsset.nazione
    -> ORDER BY num_assets DESC;
```

```
mysql> SELECT* FROM AssetsPerCountry;
+-----+-----+
| nome | num_assets |
+-----+-----+
| China |       6 |
| USA   |       6 |
| Switzerland |   3 |
| Russia |   3 |
| Canada |   3 |
| Australia |  2 |
| Germany |  2 |
| Japan  |  2 |
| Italy   |  2 |
| UK     |  2 |
| Argentine | 1 |
| Saudi Arabia | 1 |
| Brazil | 1 |
| Netherlands | 1 |
| Spain  | 1 |
| France | 1 |
| South Korea | 1 |
+-----+-----+
```

2. Si vuole sapere, per ogni borsa, il numero di dipendenti medio delle aziende le cui azioni sono scambiate su quella borsa

```
CREATE VIEW borsaNumeroDipendentiAziendaQuotata AS
    SELECT AzioneQuotataIn.borsa, avg(Azienda.numeroDipendenti) mediaDipendenti
    FROM AzioneQuotataIn, Azione FORCE INDEX(Primary), Azienda
    WHERE azione.AziendaEmittente = Azienda.ID and Azione.ticker = azioneQuotataIn.ticker GROUP
    BY AzioneQuotataIn.borsa;
```

```
mysql> CREATE VIEW borsaNumeroDipendentiAziendaQuotata AS
-> SELECT AzioneQuotataIn.borsa, avg(Azienda.numeroDipendenti) mediaDipendenti
-> FROM AzioneQuotataIn, Azione FORCE INDEX(Primary), Azienda
-> WHERE azione.AziendaEmittente = Azienda.ID AND Azione.ticker = azioneQuotataIn.ticker GROUP BY AzioneQuotataIn.borsa;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from borsanumerodipendentiaziendaquotata;
+-----+-----+
| borsa | mediadipendenti |
+-----+-----+
| Amsterdam Stock Exchange | 425000.0000 |
| Australia Stock Exchange | 70000.0000 |
| China Financial Futures Exchange | 17300.0000 |
| Copenhagen Stock Exchange | 25200.0000 |
| Deutsche Boerse | 100000.0000 |
| Egyptian Exchange | 100000.0000 |
| Euronext Paris Exchange | 3.0000 |
| Frankfurt Stock Exchange | 10000.0000 |
| Hong Kong Stock Exchange | 20250.0000 |
| Johannesburg Stock Exchange | 73000.0000 |
| London Stock Exchange | 49000.0000 |
| Mothers | 16200.0000 |
| NASDAQ | 3.0000 |
| New York Stock Exchange | 11667.6667 |
| Saudi Stock Exchange | 22325.0000 |
| Shenzhen Stock Exchange | 435000.0000 |
| South Korea Stock Exchange | 100000.0000 |
| Swiss Stock Exchange | 415250.0000 |
| Taiwan Stock Exchange | 25766.6667 |
| Tehran Stock Exchange | 100000.0000 |
| Toronto Stock Exchange | 16750.0000 |
+-----+-----+
```

3. Si vuole sapere quale borsa quota più azioni di aziende USA

```
CREATE VIEW borsaConMaggioriAzioniUSA AS
SELECT borsa, numAzioni
FROM
(SELECT max(numAzioni) max
     FROM (SELECT count(*) numAzioni
           FROM Azione, Azienda, AzioneQuotataln
          WHERE
                nazione = 'USA' AND
                aziendaEmittente = ID AND
                AzioneQuotataln.ticker = Azione.ticker GROUP BY borsa) a
    ) x,
(SELECT borsa, count(*) numAzioni
  FROM Azione, Azienda, AzioneQuotataln
  WHERE
        nazione = 'USA' AND
        aziendaEmittente = ID AND
        AzioneQuotataln.ticker = Azione.ticker GROUP BY borsa
) y
WHERE
max = numAzioni;
```

| borsa | numAzioni |
|--------------------------|-----------|
| Hong Kong Stock Exchange | 2 |
| Swiss Stock Exchange | 2 |
| Shenzhen Stock Exchange | 2 |

CREAZIONE UTENTI:

Poiché il progetto rappresenta una realtà istruttiva di una scuola superiore, abbiamo creato 3 classi di utenti in ordine decrescente di grado di privilegi. Un amministratore è colui che gestisce il database e quindi ha tutti i diritti. I professori hanno diritto di inserire nuove tuple e di effettuare query ai fini educativi. Gli studenti sono solo autorizzati ad effettuare query, in quanto usufruiscono del database solo ai fini di apprendimento finanziario.

Infine, abbiamo creato anche un generico utente autorizzato solo ad interrogare le view esistenti.

1. **Amministratore**: ha tutti i diritti

```
CREATE USER 'administrator'@'localhost' IDENTIFIED BY 'adminpassword';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'administrator'@'localhost';
GRANT ALL ON DTP.* TO 'administrator'@'localhost';
```

2. **Professore**: ha diritto di inserire tuple e fare query

```
CREATE USER 'professor'@'localhost' IDENTIFIED BY 'profpassword';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'professor'@'localhost';
GRANT SELECT ON DTP.* TO 'professor'@'localhost';
GRANT INSERT ON DTP.* TO 'professor'@'localhost';
```

3. **Studente**: ha solo diritto di fare query

```
CREATE USER 'student'@'localhost' IDENTIFIED BY 'studpassword';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'student'@'localhost';
GRANT SELECT ON DTP.* TO 'student'@'localhost';
```

4. Creazione di un utente che ha il solo diritto di accedere alle view precedentemente create

```
CREATE USER 'matteostromieri'@'localhost' IDENTIFIED BY 'mattpassword';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'student'@'localhost';
GRANT SELECT ON AssetsPerCountry TO 'matteostromieri'@'localhost';
GRANT SELECT ON BorsaNumeroDipendentiAziendaQuotataIn TO 'matteostromieri'@'localhost';
GRANT SELECT ON borsaConMaggioriAzioniUsa TO 'matteostromieri'@'localhost';
```

STORED PROCEDURES

1. Stampa tutte le informazioni relative a un impiegato

```
Delimiter //  
Create Procedure infoEmp(IN CF varchar(20), IN nazione varchar(20))  
Begin  
    SELECT *  
    FROM Impiegato  
    WHERE Impiegato.CF = CF AND Impiegato.nazione = Nazione;  
    SELECT ticker  
    FROM ImpiegatoPossiedeCriptovaluta  
    WHERE ImpiegatoPossiedeCriptovaluta.CF = CF  
    AND ImpiegatoPossiedeCriptovaluta.nazione = Nazione;  
End //  
Delimiter ;
```

```
mysql> Delimiter //  
mysql> Create Procedure infoEmp(IN CF varchar(20), IN nazione varchar(20))  
    -> Begin  
    -> SELECT *  
    -> FROM Impiegato  
    -> WHERE Impiegato.CF = CF AND Impiegato.nazione = Nazione;  
    ->  
    -> SELECT ticker  
    -> FROM ImpiegatoPossiedeCriptovaluta  
    -> WHERE ImpiegatoPossiedeCriptovaluta.CF = CF  
    -> AND ImpiegatoPossiedeCriptovaluta.nazione = Nazione;  
    -> End //
```

Chiamata:

```
mysql> call infoEmp('ZHOULC2000', 'China');  
+-----+-----+-----+-----+  
| CF      | nazione | nome   | cognome | azienda |  
+-----+-----+-----+-----+  
| ZHOULC2000 | China   | Luca   | Zhou    |       1 |  
+-----+-----+-----+-----+  
1 row in set (0.01 sec)  
  
+-----+  
| ticker |  
+-----+  
| BTC    |  
| DOPE   |  
| ETH    |  
| SOL    |  
| USDC   |  
| XRP    |  
+-----+
```

2. Stampa il numero di aziende e totale ricavato per un certo settore

```
Delimiter //  
Create Procedure InfoSector(IN settore varchar(20))  
Begin  
SELECT Settore.nome, COUNT(*) as Aziende, Settore.ricavatoComplessivo as Ricavato  
FROM Settore STRAIGHT_JOIN Opera ON Settore.nome = Opera.settore  
WHERE Settore.nome = settore;  
End //  
Delimiter ;
```

```
mysql> Create Procedure InfoSector(IN settore varchar(20))  
-> Begin  
-> SELECT Settore.nome, COUNT(*) as Aziende, Settore.ricavatoComplessivo as Ricavato  
-> FROM Settore STRAIGHT_JOIN Opera ON Settore.nome = Opera.settore  
-> WHERE Settore.nome = settore;  
->  
-> End //
```

Chiamata:

```
mysql> call InfoSector('technology');  
+-----+-----+-----+  
| nome | Aziende | Ricavato |  
+-----+-----+-----+  
| technology | 10 | 670265234984.19 |  
+-----+-----+-----+
```

3. Mostra tutte le azioni contenute in un certo indice

```
Delimiter //  
Create Procedure infoIndiceAzione(IN IndiceAzione varchar(20))  
Begin  
SELECT Contiene.tickerIndice  
FROM Contiene  
WHERE tickerIndice = IndiceAzione;  
End //  
Delimiter ;
```

```
mysql> Delimiter //  
mysql> Create Procedure infoIndiceAzione(IN IndiceAzione varchar(20))  
-> Begin  
-> SELECT *  
-> FROM Contiene  
-> WHERE tickerIndice = IndiceAzione;  
-> End //
```

```
mysql> CALL infoIndiceAzione('SP500');  
+-----+-----+  
| tickerIndice | tickerAzione |  
+-----+-----+  
| SP500 | AMZN |  
+-----+-----+
```

MONGODB

Si è scelto un tipo di rappresentazione dei dati sottoforma di denormalized document. La ragione principale è che le query a cui il sistema risponde sono particolarmente articolate e realizzare una versione normalizzata ci avrebbe obbligato a gestire solo alcune di esse tramite \$lookup, ovvero quelle che coinvolgono non più di due tabelle e le altre a livello applicativo, rendendo le operazioni che emulano le join particolarmente dispendiose, anche più di quanto avviene con MySQL. La realizzazione normalizzata a partire da un database relazione è la più semplice, comoda e diretta ma meno funzionale. Possiamo ricorrere ad un modello denormalizzato anche perché i dati interessati sono abbastanza statici e si prevede possano cambiare difficilmente e in misura molto ridotta nel tempo. Il lavoro ha visto, alla luce della struttura, un ripensamento dell'organizzazione dei dati, al fine di sfruttare al massimo le potenzialità del NoSQL.

Il database è stato diviso in tre sezioni indipendenti, una per ogni partecipante al progetto. Di seguito ci sono le strutture dei vari documenti e gli script in Python scritti per generare inserimenti di documenti casuali.

INSERT MONGO

AZIENDA

STRUTTURA DOCUMENTO

```
{  
    "_id" : ObjectId("630b7ccb234dc6ae8181d0d6"),  
    "nome" : "zjccqtoukdqs",  
    "numeroDipendenti" : 71490,  
    "utile" : 888690,  
    "fatturato" : 250942,  
    "assetPosseduti" : [  
        {  
            "ticker" : "N770"  
        },  
        {  
            "ticker" : "R397"  
        },  
        {  
            "ticker" : "U310"  
        }  
    ],  
    "nazione" : {  
        "nome" : "rjjeijqtvt",  
        "popolazione" : 714959505,  
        "pil" : 333,  
        "tassoPoverta" : 59,  
        "tassoOccupazione" : 35  
    },  
    "settore" : {  
        "nome" : "yumnraqgoul",  
        "ricavatoComplessivo" : 510907479  
    },  
    "azioniEmesse" : [  
        {  
            "titolo" : "azionisti",  
            "percentuale" : 50,  
            "anno" : 2023  
        },  
        {  
            "titolo" : "investitori",  
            "percentuale" : 30,  
            "anno" : 2023  
        },  
        {  
            "titolo" : "partner",  
            "percentuale" : 20,  
            "anno" : 2023  
        }  
    ]  
}
```

```
        "ticker" : "ILVHNUA",
        "valore" : 2625,
        "prezzoEmissione" : 922,
        "flottante" : 56427194
    },
    {
        "ticker" : "SAXJOOS",
        "valore" : 6590,
        "prezzoEmissione" : 594,
        "flottante" : 31104997
    },
    {
        "ticker" : "GKDGLTD",
        "valore" : 9980,
        "prezzoEmissione" : 8296,
        "flottante" : 2647226
    }
],
"obbligazioniEmesse" : [
    {
        "ticker" : "FWSUYXY",
        "scadenza" : "2030-11-14",
        "interesse" : 36184260,
        "rendimento" : 100
    },
    {
        "ticker" : "GJOURTJ",
        "scadenza" : "2035-3-13",
        "interesse" : 48346213,
        "rendimento" : 47
    },
    {
        "ticker" : "ETUCWFP",
        "scadenza" : "2027-9-10",
        "interesse" : 79985696,
        "rendimento" : 29
    },
    {
        "ticker" : "PQIMTHN",
        "scadenza" : "2027-4-13",
        "interesse" : 78140941,
        "rendimento" : 96
    },
    {
        "ticker" : "QNERBFC",
        "scadenza" : "2031-10-22",
        "interesse" : 48863920,
        "rendimento" : 75
    }
],
```

```

"criptovalutePossedute" : [
    {
        "ticker" : "GLUGNHV",
        "flottante" : 93277464
    },
    {
        "ticker" : "MDGENSO",
        "flottante" : 19157858
    },
    {
        "ticker" : "FQRVMQM",
        "flottante" : 74239684
    },
    {
        "ticker" : "JGGVGTO",
        "flottante" : 31420334
    },
    {
        "ticker" : "XTTJTAD",
        "flottante" : 54259751
    },
    {
        "ticker" : "HYNXFCO",
        "flottante" : 41063537
    }
]
}

```

PYTHON

```

import random

alfabetoMaiuscolo = "ABCDEFGHIJKLMNPQRSTUVWXYZ"
alfabetoMinuscolo = "abcdefghijklmnoprstuvwxyz"

n = 200000
# generazione nazione

Nazioni = []
nazione = ""

nNazioni = 200
for i in range(nNazioni):
    nome = "\\" + "".join(random.choice(alfabetoMinuscolo) for i in range(10)) +
    "\\"
    popolazione = str(random.randint(1000000, 100000000))
    pil = str(random.randint(1,500))
    tassoPoverta = str(random.randint(1,100))
    tassoOccupazione = str(random.randint(1,100))

```

```

    valuta = "" + "".join(random.choice(alfabetoMaiuscolo) for i in range(4)) +
    "\n"
    nazione = "{" + "\n" + "\"nome\":" + nome + ",\n" + "\"popolazione\":" +
popolazione + ",\n" + "\"pil\":" + pil + ",\n" + "\"tassoPoverta\":" + tassoPoverta +
",\n" + "\"tassoOccupazione\":" + tassoOccupazione + "\n}"
    Nazioni.append(nazione)

print(" nazioni create")

# generazione settore

Settori = []
settore = ""

nSettori = 50
for i in range(nSettori):
    nome = "" + "".join(random.choice(alfabetoMinuscolo) for i in range(10)) +
"\n"
    ricavatoComplessivo = str(random.randint(10000, 1000000000))
    settore = "{\n" + "\"nome\":" + nome + ",\n" + "\"ricavatoComplessivo\":" +
ricavatoComplessivo + "\n}"
    Settori.append(settore)

print(" settori creati")

# generazione AzioneEmessa (una o piu azioni) [{azione1}, {azione2} ...]

SetAzioniEmesse = []

for i in range(n):
    azioniEmesse = "[\n"
    for j in range(random.randint(0,10)):
        if j != 0 :
            azioniEmesse = azioniEmesse + ","

        ticker = "" + "".join(random.choice(alfabetoMaiuscolo) for i in range(7)) +
"\n"
        valoreNominal = str(random.randint(0,10000))
        prezzoEmissione = str(random.randint(0,10000))
        flottante = str(random.randint(1,100000000))

        azioniEmesse = azioniEmesse + "{\n" + "\"ticker\":" + ticker + ",\n" +
"\n" + "\"valore\":" + valoreNominal + ",\n" + "\"prezzoEmissione\":" + prezzoEmissione +
",\n" + "\"flottante\":" + flottante + "\n}"

    azioniEmesse = azioniEmesse + "]"
    SetAzioniEmesse.append(azioniEmesse)

print(" azioni create")

```

```

# generazione ObbligazioneEmessa (array di obbligazioni)

SetObbligazioniEmesse = []

for i in range(n):
    obbligazioniEmesse = "[\n"
    for j in range(random.randint(0,10)):
        if j != 0 :
            obbligazioniEmesse = obbligazioniEmesse + ","

            ticker = "\\" + "".join(random.choice(alfabetoMaiuscolo) for i in range(7))
+ \""
            scadenza = "\"" + str(random.randint(2025, 2035)) + "-" +
str(random.randint(1, 12)) + "-" + str(random.randint(1, 28)) + "\\""
            interesse = str(random.randint(1,100000000))
            rendimento = str(random.randint(1,100))

            obbligazioniEmesse = obbligazioniEmesse + "{\n" + "\"ticker\":" + ticker +
",\n" + "\"scadenza\":" + scadenza + ",\n" + "\"interesse\":" + interesse + ",\n" +
"\"rendimento\":" + rendimento + "\n}"

    obbligazioniEmesse = obbligazioniEmesse + "]"
    SetObbligazioniEmesse.append(obbligazioniEmesse)

print(" obbligazioni create")

# generazione CriptovalutaPosseduta

SetCriptoPossedute = []

nCripto = 25
for i in range(nCripto):

    criptoPossedute = "[\n"
    for j in range(random.randint(0,10)):
        if j != 0 :
            criptoPossedute = criptoPossedute + ","

            ticker = "\\" + "".join(random.choice(alfabetoMaiuscolo) for i in range(7))
+ \""
            flottante = str(random.randint(1,100000000))

            criptoPossedute = criptoPossedute + "{\n" + "\"ticker\":" + ticker + ",\n" +
"\"flottante\":" + flottante + "\n}"

    criptoPossedute = criptoPossedute + "]"
    SetCriptoPossedute.append(criptoPossedute)

print(" cripto create")
# generazione attributi azienda

```

```

file = open("InserimentiMongo.txt", "w+")

Aziende = "" + "\n"
for i in range(n) :
    print(i)
    if (i != 0):
        Aziende = Aziende + ",\n"

    nome = "\"" + "".join(random.choice(alfabetoMinuscolo) for i in range(12)) +
"\\""

    numeroDipendenti = str(random.randint(1, 100000))
    utile = str(random.randint(1, 1000000))
    fatturato = str(random.randint(1,1000000))
    assetPosseduti = "[\n"
    for j in range(random.randint(0, 10)):
        if j != 0:
            assetPosseduti = assetPosseduti + ",\n"
        assetPosseduti = assetPosseduti + "{\n\"ticker\":" + "\"" +
"".join(random.choice(alfabetoMaiuscolo)) + str(random.randint(100,999)) + "\\"\n}"

    assetPosseduti = assetPosseduti + "]\n"

    Aziende = Aziende + "{\n" + "\"nome\":" + nome + ",\n" + "\"numeroDipendenti\":" +
+ numeroDipendenti + ",\n" + "\"utile\":" + utile + ",\n" + "\"fatturato\":" +
fatturato + ",\n" + "\"assetPosseduti\":" + assetPosseduti + ",\n"
    # aggiunta altre strutture create
    Aziende = Aziende + "\"nazione\":" + Nazioni[random.randint(0,nNazioni-1)] +
",\n" + "\"settore\":" + Settori[random.randint(0,nSettori-1)] + ",\n" +
"\\"azioniEmesse\":" + SetAzioniEmesse[i] + ",\n" + "\"obbligazioniEmesse\":" +
SetObbligazioniEmesse[i] + ",\n" + "\"criptovalutePossedute\":" +
SetCriptoPossedute[random.randint(0,nCripto-1)] + "\n }"

    if (i%5000 == 0) :
        file.write(Aziende)
        Aziende = "\n"

Aziende = Aziende + ","

file.write(Aziende)
file.close()

print("fine")

```

AZIONI

STRUTTURA DOCUMENTO

```
[{
    "ticker": "IUATAEX",
    "valoreNominal": 1807,
    "prezzoEmissione": 4497,
    "flottante": 89457200,
    "aziendaEmittente": {
        "nome": "s",
        "numeroDipendenti": 2178257,
        "fatturato": 16442902,
        "settore": "yzbqbr",
        "nazione": "skir"
    },
    "tipologia": {
        "nome": "tipologia8",
        "nominativita": true,
        "dirittoDiVoto": true,
        "dirittoDiVotoStraordinario": true,
        "dirittoAlDividendo": false,
        "ripartizioneUtile": false,
        "maggiorazionePercentuale": false
    },
    "quotaIn": [
        {
            "nome": "qjyc",
            "volumeScambi": 963819,
            "nazione": "ymvu"
        },
        {
            "nome": "abaq",
            "volumeScambi": 4145055,
            "nazione": "gfjs"
        },
        {
            "nome": "fxac",
            "volumeScambi": 2248996,
            "nazione": "ebty"
        },
        {
            "nome": "fxda",
            "volumeScambi": 1576523,
            "nazione": "buak"
        },
        {
            "nome": "pirn",
            "volumeScambi": 4135014,
            "nazione": "msnq"
        }
    ],
    "indiceAzionario": [
        {
            "titolo": "Indice Azionario Italiano"
        }
    ]
}]
```

```

        "ticker":"YYYJTQ",
        "valoreNominale":3184,
        "prezzoEmissione":1265
    },
    {
        "ticker":"SUZHLO",
        "valoreNominale":5532,
        "prezzoEmissione":1028
    },
    {
        "ticker":"OODEDW",
        "valoreNominale":9686,
        "prezzoEmissione":1927
    },
    {
        "ticker":"YGNRNX",
        "valoreNominale":7740,
        "prezzoEmissione":5283
    },
    {
        "ticker":"EDRCVD",
        "valoreNominale":5605,
        "prezzoEmissione":1027
    }
]
}

```

PYTHON

```

import random

alfabetoMaiuscolo = "ABCDEFGHIJKLMNPQRSTUVWXYZ"
alfabetoMinuscolo = "abcdefghijklmnopqrstuvwxyz"

# prima generiamo delle possibili "tipologie"

tipologia = []
attributi = ["true", "false"]

i = 1
for nominativita in (0,1):
    for dirittoDiVoto in (0,1):
        for dirittoDiVotoStraordinario in (0,1):
            for dirittoAlDividendo in (0,1):
                for ripartizioneUtile in (0,1):
                    for maggiorazionePercentuale in (0,1):
                        nome = "\"tipologia"+str(i)+"\""
                        i = i + 1
                        stringaTipologia = "\"tipologia\": {" + "\n" + "\"nome\":\"" +
nome + "," + "\n" + "\"nominativita\":\"" + attributi[nominativita] + "," + "\n" +

```

```

\"dirittoDiVoto\":" + attributi[dirittoDiVoto] + "," + "\n" +
\"dirittoDiVotoStraordinario\":" + attributi[dirittoDiVotoStraordinario] + "," + "\n" +
"\n" + \"dirittoAlDividendo\":" + attributi[dirittoAlDividendo] + "," + "\n" +
\"ripartizioneUtile\":" + attributi[ripartizioneUtile] + "," + "\n" +
\"maggiorazionePercentuale\":" + attributi[maggiorazionePercentuale] + "\n" + "}"
tipologia.append(stringaTipologia)

# generiamo le nazioni

nazioni = []
i = 0
for i in range(200):
    nazione = "".join(random.choice(alfabetoMinuscolo) for i in range(4))
    nazioni.append(nazione)

# generiamo 1000 borse

borse = []
borsa = ""

for i in range(1000):
    nome = "\"" + "".join(random.choice(alfabetoMinuscolo) for i in range(4)) + "\""
    nazione = "\"" + nazioni[random.randint(0,len(nazioni)-1)] + "\""
    volumeScambi = str(random.randint(0,10000000))
    borsa = "{" + "\n" + "\"nome\":" + nome + "," + "\n" + "\"volumeScambi\":" +
volumeScambi + "," + "\n" + "\"nazione\":" + nazione + "\n" + "}"
    borse.append(borsa)

# generiamo 1000 indici azionari

indici = []
indice = ""

for i in range(1000):
    ticker = "\"" + "".join(random.choice(alfabetoMaiuscolo) for i in range(6)) +
"\n"
    valoreNominale = str(random.randint(0,10000))
    prezzoEmissione = str(random.randint(0,10000))
    indice = "{" + "\n" + "\"ticker\":" + ticker + "," + "\n" +
"\n" + "\"valoreNominale\":" + valoreNominale + "," + "\n" + "\"prezzoEmissione\":" +
prezzoEmissione + "\n" + "}"
    indici.append(indice)

# generiamo 100 settori

settori = []

for i in range(100):
    settori.append("".join(random.choice(alfabetoMinuscolo) for i in range(6)))

# generiamo 10000 azioni

```

```

azioni = "[" + "\n"

n = 50000
for i in range(n):
    azione = ""
    ticker = "\\" + "".join(random.choice(alfabetoMaiuscolo) for i in range(7)) +
"\\""
    valoreNominale = valoreNominale = str(random.randint(0,10000))
    prezzoEmissione = str(random.randint(0,10000))
    flottante = str(random.randint(1,100000000))
    nomeAzienda = "\\" + "".join(random.choice(alfabetoMinuscolo) for i in
range(random.randint(1,10))) + "\\"
    settore = "\\" + random.choice(settori) + "\\"
    nazione = "\\" + random.choice(nazioni) + "\\"
    numeroDipendenti = str(random.randint(1,3000000))
    fatturato = str(random.randint(0,1000000000))

    azione = azione + "{" + "\n" + "\"ticker\":" + ticker + "," + "\n" +
"\\"valoreNominale\":" + valoreNominale + "," + "\n" + "\"prezzoEmissione\":" +
prezzoEmissione + "," + "\n" + "\"flottante\":" + flottante + "," + "\n"
    azione = azione + "\"aziendaEmittente\":" + "{" + "\n" + "\"nome\":" +
nomeAzienda + "," + "\n" + "\"numeroDipendenti\":" + numeroDipendenti + "," + "\n" +
"\\"fatturato\":" + fatturato + "," + "\n" + "\"settore\":" + settore + "," + "\n" +
"\\"nazione\":" + nazione + "\n" + "}"
    azione = azione + "," + "\n"
    azione = azione + random.choice(tipologia) + "," + "\n"
    azione = azione + "\"quotaIn\":["
    for j in range(random.randint(0,5)):
        azione = azione + random.choice(borse) + "," + "\n"
    azione = azione + random.choice(borse) + "\n"
    azione = azione + "]" + "," + "\n"

    azione = azione + "\"indiceAzionario\":["
    for k in range(random.randint(0,5)):
        azione = azione + random.choice(indici) + "," + "\n"
    azione = azione + random.choice(indici) + "\n"
    azione = azione + "]" + "\n"
    azione = azione + "}"
    if i == n-1:
        azioni = azioni + azione + "\n"
    else:
        azioni = azioni + azione + "," + "\n"
azioni = azioni + "]"

# copia delle azioni su file json

file = open("inserimenti.json","w+")
file.write(azioni)
file.close()

```

IMPIEGATO

STRUTTURA DOCUMENTO

```
{  
    "_id" : ObjectId("630a39f70eb59ecbe853efe2"),  
    "CF" : "EDWTK090595",  
    "nazione" : "Australia",  
    "nome" : "Edward",  
    "cognome" : "Whittaker",  
    "criptovalute" : [  
        {  
            "ticker" : "AMZN",  
            "flottante" : 4143474  
        },  
        {  
            "ticker" : "RACE",  
            "flottante" : 3786694  
        }  
    ],  
    "azienda" : {  
        "ID" : 15950,  
        "nome" : "Moonlight srl",  
        "nazione" : "Italy",  
        "utile" : 21921431,  
        "fatturato" : 60260526,  
        "numeroDipendenti" : 531  
    }  
}
```

PYTHON

```
import random  
  
symbols = 'ABCDEFGHIJKLMNPQRSTUVWXYZ1234567890'  
lower = 'abcdefghijklmnopqrstuvwxyz'  
  
criptos = []  
cripto = ""  
for i in range(1007):  
    ticker = '"' + ''.join(random.choice(symbols)) for i in range(6)) + "  
    flottante = str(random.randint(10,10000000))  
    cripto = "{\n" + "\"ticker\":" + ticker + ",\n" + "\"flottante\":" + flottante  
+ "\n}"  
    criptos.append(cripto)  
  
  
aziende = []  
azienda = ""  
for i in range(197836):
```

```

azienda = ""
azienda = azienda + "{\"ID\":"
azienda = azienda + str(i)
azienda = azienda + ",\n \"nome\": " + "\\""
azienda = azienda + ''.join(random.choice(lower) for i in
range(random.randint(3,13)))
azienda = azienda +"\\""
azienda = azienda + ",\n \"nazione\":"
azienda = azienda + "\\""
azienda = azienda + ''.join(random.choice(lower) for i in
range(random.randint(3,13)))
azienda = azienda +"\\""
azienda = azienda + ",\n \"utile\":"
azienda = azienda + str(random.randint(1000, 900000000))
azienda = azienda + ",\n \"fatturato\":"
azienda = azienda + str(random.randint(1000, 900000000))
azienda = azienda + ",\n \"numeroDipendenti\": " + str(random.randint(6, 70000))
+ "}\n\n"
aziende.append(azienda)
f= open("Impiegato.json","w+")

f.write("[")
CF = "ZHOULC2000"
nazione = "China"
unique_CF_nazione = set()
unique_CF_nazione.add((CF, nazione))

CF = ''.join(random.choice(symbols) for i in range(10))
nazione = ''.join(random.choice(lower) for i in range(random.randint(3,10)))

for i in range(1, 245310):
    while((CF, nazione) in unique_CF_nazione):
        CF = ''.join(random.choice(symbols) for i in range(10))
        nazione = ''.join(random.choice(lower) for i in range(random.randint(3,10)))
    unique_CF_nazione.add((CF, nazione))

    criptovalute = ""
    times = random.randint(0, 5)
    for i in range(times):
        if (i == times - 1):
            criptovalute = criptovalute + random.choice(criptos)
        else:
            criptovalute = criptovalute + random.choice(criptos) + ",\n"

    f.write("{\"CF\": " + "\"" + CF + "\"" +
            ",\n \"nazione\": " + "\"" + nazione + "\"" + "

```

```

        ",\n \"nome\": " + "\"" + ''.join(random.choice(lower)) for i in
range(random.randint(3,10))) +"\n" +
        ",\n \"cognome\": " + "\"" + ''.join(random.choice(lower)) for i in
range(random.randint(3,10))) +"\n" +
        ",\n \"criptovalute\":[+" + criptovalute +"]" +
        ",\n \"azienda\": " + str(random.choice(aziende)) + "},\n\n"
    )

f.write("{\"CF\": " + "\"" + CF + "\" +\n
        ",\n \"nazione\": " + "\"" + nazione +"\n" +
        ",\n \"nome\": " + "\"" + ''.join(random.choice(lower)) for i in
range(random.randint(3,10))) +"\n" +
        ",\n \"cognome\": " + "\"" + ''.join(random.choice(lower)) for i in
range(random.randint(3,10))) +"\n" +
        ",\n \"criptovalute\":[+" + criptovalute +"]" +
        ",\n \"azienda\": " + str(random.choice(aziende)) + "}]}\n\n")

f.close()

```

QUERY

AZIENDA

- 1. Si vuole sapere la percentuale di occupazione che un'azienda offre, con nome e nazione dell'azienda:**

```
db.azienda.aggregate([
    {$project:{nome:1, "nazione.nome":1,occupazione:{$divide:[{"$numeroDipendenti","$nazione.popolazione"]}}}
])

```

- 2. Per ogni settore si vuole sapere il numero di aziende che opera in questo ed il ricavato complessivo:**

```
db.azienda.aggregate([
    {"$group":{"_id":{"settore":"$settore.nome", "ricavato":"$settore.ricavatoComplessivo"}, 
    "numeroAziende":{$sum:1}},
    {"$project":{settore:1, numeroAziende:1}}
])

```

- 3. Top 3 dipendenti con il maggior numero di tipi di criptovalute, che non sono americani:**

```
db.azienda.aggregate([
    {"$group":{"_id":{"nazione":"$nazione.nome"}, "numeroAsset":{"$sum":{$size:"$assetPosseduti"}}, 
    "$project":{"nazione.nome":1, "numeroAsset":1},
    {$sort:{numeroAsset:-1}}
])

```

4. Numero di aziende, per paese, con il fatturato totale in ordine decrescente:

```
db.azienda.aggregate([
    {$group:{"_id": {"nazione": "$nazione.nome"}, numAziende:{$sum:1}, fatturatoTotale:{$sum:"$fatturato"}}, 
    {$project:{ "nazione.nome":1, numAziende:1, fatturatoTotale:1}}, 
    {$sort:{fatturatoTotale:-1}}
])
```

5. Numero criptovalute possedute delle aziende che sono in paesi con almeno 50 milioni di persone:

```
db.azienda.aggregate([
    {$match:{ "nazione.popolazione":{$gt:50000000}}},
    {$project:{ nome:1, numCriptovalute:{$size:"criptovalutePossedute"}, "nazione.popolazione":1}}
])
```

AZIONI

1. Dire, per ogni tipologia di azione, il valore nominale totale delle azioni che vi appartengono:

```
db.azione.aggregate([
    {"$group": {"_id": {"tipologia": "$tipologia.nome"}, "valoreNominaleTotale":{$sum:"$valoreNominale"}}, 
    {"$project": {"_id":0,"tipologia":1,"valoreNominaleTotale":1}}
])
```

2. Si vogliono sapere quali indici hanno valore nominale maggiore di 1000:

```
db.azione.find( {
    indiceAzione: { "$elemMatch": { valoreNominali: { $gte : 1000 } } },
    {indiceAzione:1}
}.pretty()
```

3. si vuole sapere il numero di azioni che sono di tipologia 1:

```
db.azione.aggregate([
    {"$match": {"tipologia.nome": "tipologia1"}}, 
    {"$count": "Totale"}
])
```

4. azioni quotate in indici dal prezzo di emissione di almeno 3000 che sono quotate in borse che iniziano per 'a':

```
db.azione.find({
    "indiceAzione": { "$elemMatch": {"prezzoEmissione": {"$gte": 3000}}}, 
    "quotaln": { "$elemMatch": {"nome": /a/i}}}.pretty()
})
```

5. si vogliono sapere le azioni che sono quotate in più borse e sono di tipo 23:

```
var j = db.azione.aggregate([
  { "$project": {"borse": { "$size": "$quotaln" } }, { "$sort": { "borse": -1 } },
  { $limit:1},
  { $project:{"_id":0,"borse":1}}])
```

```
var i = j.toArray()
```

```
db.azione.aggregate([
  {"$match":{ "tipologia.nome":"tipologia23","quotaln":{"$size":i[0].borse}}}
])
```

6. si vuole sapere il numero di azioni che hanno diritto di voto e diritto di voto straordinario e sono quotate nello stesso indice:

```
db.azione.aggregate( [
  { $match: { "tipologia.dirittoDiVoto": true, "tipologia.dirittoDiVotoStraordinario":true } },
  { $unwind : "$indiceAкционario" },
  { $group: { _id:"$indiceAкционario.ticker", totale:{ $sum: 1 } } },
])
```

7. si vogliono mostrare i ticker delle azioni di aziende italiane quotate in borse americane:

```
db.azione.find(
  {"aziendaEmittente.nazione":"skir", quotaln:{$elemMatch:{nazione:"gfjs"}}, 
  {ticker:1}
)
```

8. mostrare le aziende che hanno emesso azioni che vengono quotate in paesi diversi dai loro:

```
db.azione.aggregate([
  {$match:{quotaln:{$elemMatch:{nazione:$ne:"aziendaEmittente.nazione"}}}}
])
```

9. mostrare le aziende che hanno emesso azioni che vengono quotate nei loro stessi paesi:

```
db.azione.aggregate([
  {$unwind:"$quotaln"}, 
  {$match:{$expr:{$eq:["$quotaln.nazione","$aziendaEmittente.nazione"]}}},
  {$project:{aziendaEmittente:1,quotaln:1}}
])
```

10. mostrare la media dei flottanti delle azioni che hanno annesso il diritto alla maggioranzaPercentuale, quotate e emesse in USA e contenute in indici azionari con ticker che iniziano che la A:

```
db.azione.aggregate([
    {$match:{"tipologia.maggiorazionePercentuale":true,"aziendaEmittente.nazione":"skir",
        quotaInn:$elemMatch:{nazione:"skir"},indiceAкционario:$elemMatch:{ticker:
            {$regex:/^A/}}}},
    {$group:{_id:null,mediaFlottante:{$avg:"$flottante}}},
    {$project:{_id:o}}
])
```

IMPIEGATO

1. visualizzare CF, nome, cognome e azienda di tutti gli impiegati che lavorano in una azienda con un fatturato maggiore di 80000000 e numero di dipendenti minore di 15:

```
db.Impiegato.find(
    { "azienda.fatturato": { "$gt": 80000000 }, "azienda.numeroDipendenti": { "$lt": 15 } },
    { _id:o, CF:1, nome:1, cognome:1, azienda:1 }
).pretty()
```

```
> db.Impiegato.find(
... { "azienda.fatturato": { "$gt": 80000000 }, "azienda.numeroDipendenti": { "$lt": 15 } }, { _id:0, CF:1, nome:1, cognome:1, azienda:1 }
... ).pretty()
{
    "CF" : "RRNX2QI4DJ",
    "nome" : "nhnum",
    "cognome" : "qhn",
    "azienda" : {
        "ID" : 68104,
        "nome" : "hinc",
        "nazione" : "wlxvuijzhf1",
        "utile" : 43882183,
        "fatturato" : 833556656,
        "numeroDipendenti" : 7
    }
}
{
    "CF" : "I70VHNT80A",
    "nome" : "xuitearg",
    "cognome" : "hkw",
    "azienda" : {
        "ID" : 68104,
        "nome" : "hinc",
        "nazione" : "wlxvuijzhf1",
        "utile" : 43882183,
        "fatturato" : 833556656,
        "numeroDipendenti" : 7
    }
}
>
```

2. visualizzare il numero di impiegati che hanno almeno 4 criptovalute:

```
db.Impiegato.find( { "criptovalute.4": { $exists: true } } ).count()
```

```
> db.Impiegato.find( { "criptovalute.4": { $exists: true } } ).count()
40944
```

3. visualizzare il numero di impiegati (nel Database) che lavorano nelle aziende con ID < 8, raggruppato per azienda e ordinato in modo non decrescente:

```
db.Impiegato.aggregate([
  { $match: { "azienda.ID": { $lt: 8 } } },
  { $group: { _id: "$azienda.ID", count: { $sum: 1 } } },
  { $sort: {count: -1} }
])
```

```
> db.Impiegato.aggregate([
... { $match: { "azienda.ID": { $lt: 8 } } },
... { $group: { _id: "$azienda.ID", count: { $sum: 1 } } },
... { $sort: {count: -1} }
... ])
{ "_id" : 4, "count" : 4 }
{ "_id" : 7, "count" : 2 }
{ "_id" : 1, "count" : 2 }
{ "_id" : 2, "count" : 1 }
{ "_id" : 3, "count" : 1 }
>
```

4. visualizzare CF e nazione degli impiegati che possiedono almeno una criptovaluta che abbia un flottante di almeno 9990000 e la criptovaluta stessa:

```
db.Impiegato.aggregate([
  { $unwind: "$criptovalute" },
  { $project: { _id: 0, CF: 1, nazione: 1, criptovalute: 1 } },
  { $match: { "criptovalute.flottante": { $gte: 9990000 } } },
  { $limit: 3 }
]).pretty()
```

```
> db.Impiegato.aggregate([
... { $unwind: "$criptovalute" },
... { $project: { _id: 0, CF: 1, nazione: 1, criptovalute: 1 } },
... { $match: { "criptovalute.flottante": { $gte: 9990000 } } },
... { $limit: 3 }
... ]).pretty()
{
  "CF" : "RSXOLOE9QA",
  "nazione" : "vmmglsa",
  "criptovalute" : {
    "ticker" : "UK2ZV0",
    "flottante" : 9990579
  }
}
{
  "CF" : "ENWVGQFQ902",
  "nazione" : "jyemhnc",
  "criptovalute" : {
    "ticker" : "UK2ZV0",
    "flottante" : 9990579
  }
}
{
  "CF" : "8E4N601GAU",
  "nazione" : "jq1",
  "criptovalute" : {
    "ticker" : "UK2ZV0",
    "flottante" : 9990579
  }
}>
```

(Si limitano i documenti a 3 per una questione di visualizzazione)

5. visualizza l'azienda con il rapporto (fatturato / numeroDipendenti) massimo:

```
db.Impiegato.aggregate([
  { $addFields: { "azienda.rapporto": { $divide: [ "$azienda.fatturato", "$azienda.numeroDipendenti" ] } } },
  { $sort: { "azienda.rapporto": -1 } },
  { $limit: 1 },
  { $project: { _id: 0, azienda: 1 } },
]).pretty()
```

```
> db.Impiegato.aggregate( [
... { $addFields: { "azienda.rapporto": { $divide: [ "$azienda.fatturato", "$azienda.numeroDipendenti" ] } } },
... { $sort: { "azienda.rapporto": -1 } },
... { $limit: 1 },
... { $project: { _id: 0, azienda: 1 } },
... ] ).pretty()
{
  "azienda" : {
    "_id" : 68104,
    "nome" : "hinc",
    "nazione" : "wlxvujzhfl",
    "utile" : 43882183,
    "fatturato" : 833556656,
    "numeroDipendenti" : 7,
    "rapporto" : 119079522.28571428
  }
}
>
```

6. visualizza le 5 criptovalute che sono maggiormente possedute:

```
db.Impiegato.aggregate([
  { $unwind: "$criptovalute" },
  { $project: { _id: 0, criptovalute: 1 } },
  { $group: { _id: "$criptovalute.ticker", count: { $sum: 1 } } },
  { $sort: { count: -1 } },
  { $limit: 5 }
]).pretty()
```

```
> db.Impiegato.aggregate( [
... { $unwind: "$criptovalute" },
... { $project: { _id: 0, criptovalute: 1 } },
... { $group: { _id: "$criptovalute.ticker", count: { $sum: 1 } } },
... { $sort: { count: -1 } },
... { $limit: 5 }
... ] ).pretty()
{ "_id" : "88N4TU", "count" : 696 },
{ "_id" : "DDTWU8", "count" : 690 },
{ "_id" : "1AT9ZI", "count" : 682 },
{ "_id" : "XHBA4K", "count" : 680 },
{ "_id" : "EU9EXZ", "count" : 679 }
>
```

7. visualizza il flottante medio delle 5 criptovalute maggiormente possedute:

```
var flottanti = db.Impiegato.aggregate([
  { $unwind: "$criptovalute" },
  { $group: { _id: "$criptovalute.ticker", count: { $sum: 1 }, flottante: { $last: "$criptovalute.flottante" } } },
  { $sort: { count: -1 } },
  { $project: { _id: 0, flottante: 1 } },
  { $limit: 5 }
]).pretty()

var flottanti_array = flottanti.toArray()
var sum = 0
for (i = 0; i < 5; i++) {
  sum += flottanti_array[i].flottante * (1/5)
}

> var flottanti = db.Impiegato.aggregate([
... { $unwind: "$criptovalute" },
... { $group: { _id: "$criptovalute.ticker", count: { $sum: 1 }, flottante: { $last: "$criptovalute.flottante" } } },
... { $sort: { count: -1 } },
... { $project: { _id: 0, flottante: 1 } },
... { $limit: 5 }
... ]).pretty()
>
> var flottanti_array = flottanti.toArray()
> var sum = 0
> for ( i = 0; i < 5; i++ ) {
...   sum += flottanti_array[i].flottante * (1/5)
...
4760473.8
>
```

8. visualizza tutte le aziende che hanno almeno un impiegato con esattamente 5 criptovalute e i CF degli impiegati stessi:

```
db.Impiegato.find(
  { criptovalute: { $size: 5 } },
  { _id: 0, azienda: 1, CF: 1 }

).limit(2).pretty()
```

```
> db.Impiegato.find(
... { criptovalute: { $size: 5 } },
... { _id: 0, azienda: 1, CF: 1 }
... ).limit(2).pretty()
{
  "CF" : "AZXSLCLAJ5",
  "azienda" : {
    "ID" : 45635,
    "nome" : "wtqzydgmozrok",
    "nazione" : "oukr",
    "utile" : 724573049,
    "fatturato" : 42880195,
    "numeroDipendenti" : 41249
  }
}
{
  "CF" : "1Twx6QL3JE",
  "azienda" : {
    "ID" : 175445,
    "nome" : "qpdanlimvlhc",
    "nazione" : "plsjtlpxj",
    "utile" : 241978400,
    "fatturato" : 282690653,
    "numeroDipendenti" : 24052
}
>
```

9. visualizza il numero delle aziende che hanno esclusivamente impiegati con esattamente 3 criptovalute:

```
db.Impiegato.aggregate([
  { $unwind: "$criptovalute" },
  { $group: { _id: {CF: "$CF", nazione: "$nazione"}, count: { $sum: 1 }, azienda: { $last: "$azienda" } } },
  { $match : { count: 3 } },
  { $group: { _id: "$azienda" } },
  { $count: "count" }
]).pretty()
```

```
> db.Impiegato.aggregate([
... { $unwind: "$criptovalute" },
... { $group: { _id: {CF: "$CF", nazione: "$nazione"}, count: { $sum: 1 }, azienda: { $last: "$azienda" } } },
... { $match : { count: 3 } },
... { $group: { _id: "$azienda" } },
... { $count: "count" }
... ]).pretty()
{ "count" : 36527 }
```

MONGODB VS MYSQL

IMPIEGATO

3. visualizzare il numero di impiegati (nel Database) che lavorano nelle aziende con ID < 8, raggruppato per azienda e ordinato in modo non decrescente:

Mongo:

```
db.Impiegato.aggregate([
  { $match: { "azienda.ID": { $lt: 8 } } },
  { $group: { _id: "$azienda.ID", count: { $sum: 1 } } },
  { $sort: {count: -1} }
])
```

```
"executionTimeMillis" : 236,
```

MySQL:

```
SELECT Impiegato.azienda, COUNT(*) as count
FROM Impiegato
WHERE Impiegato.azienda < 8
GROUP BY Impiegato.azienda
ORDER BY count DESC;
```

```
7 rows in set (0.00 sec)
```

6. visualizza le 5 criptovalute che sono maggiormente possedute:

Mongo:

```
db.Impiegato.aggregate([
  { $unwind: "$criptovalute" },
  { $project: { _id: 0, criptovalute: 1 } },
  { $group: { _id: "$criptovalute.ticker", count: { $sum: 1 } } },
  { $sort: { count: -1 } },
  { $limit: 5 }
]).pretty()
```

```
"executionTimeMillis" : 1084,
```

MySQL:

```
SELECT ticker, Count(*) as count
FROM ImpiegatoPossiedeCriptovaluta
GROUP BY ticker
ORDER BY count DESC
LIMIT 5;
```

```
5 rows in set (0.01 sec)
```

9. Visualizza il numero delle aziende che hanno esclusivamente impiegati con esattamente tre criptovalute:

Mongo:

```
db.Impiegato.aggregate([
  { $unwind: "$criptovalute" },
  { $group: { _id: { CF: "$CF", nazione: "$nazione" }, count: { $sum: 1 }, azienda: { $last: "$azienda" } } },
  { $match: { count: 3 } },
  { $group: { _id: "$azienda" },
    { $count: "count" }
  }
]).pretty()
```

```
"executionTimeMillis" : 1721,
```

MySQL:

```
SELECT COUNT(*) AS count
FROM (
  SELECT X.CF AS CF, X.nazione AS nazione , COUNT(*) as numCripto
```

```
FROM ImpiegatoPossiedeCriptovaluta X
GROUP BY X.CF, X.nazione
) AS subTab
WHERE subTab.numCripto = 3;
```

```
1 row in set (0.03 sec)
```

AZIENDA

1. Si vuole sapere la percentuale di occupazioe che un'azienda offre, con nome e nazione dell'azienda:

```
SELECT Azienda.nome, Azienda.nazione, numeroDipendenti/popolazione as occupazione FROM Azienda, Nazione
WHERE Azienda.nazione = Nazione.nome;
```

```
195954 rows in set (0,47 sec)
```

```
db.azienda.aggregate({$project:{nome:1, "nazione.nome":1,
occupazione:{$divide:["$numeroDipendenti","$nazione.popolazione"]}}}).pretty()
```

```
"executionTimeMillis" : 625,
"totalKeysExamined" : 0,
"totalDocsExamined" : 200000,
```

2. Per ogni settore si vuole sapere il numero di aziende che opera in questo ed il ricavato complessivo:

```
SELECT Settore.nome, count(*) as numAziende, ricavatoComplessivo
FROM Opera, Settore
WHERE Opera.settore = Settore.nome GROUP BY Settore.nome;
```

```
117 rows in set (0,27 sec)
```

```
db.azienda.aggregate({"$group":{"_id":{"settore":"$settore.nome", "ricavato":"$settore.ricavatoComplessivo"}, "numeroAziende":{$sum:1}}, {"$project":{settore:1, numeroAziende:1}})
```

```
"executionTimeMillis" : 320,
"totalKeysExamined" : 0,
"totalDocsExamined" : 200000,
```

3. Numero di asset che ogni nazione ha in ordine decrescente (comprese tutte le nazioni):

```
SELECT Nazione.nome, COUNT(NazionePossiedeAsset.ticker) as num_asset
FROM Nazione JOIN NazionePossiedeAsset on Nazione.nome = NazionePossiedeAsset.nazione
GROUP BY Nazione.nome
ORDER BY num_asset DESC;
```

```
209 rows in set (0,01 sec)
```

```
db.azienda.aggregate({$group:{"_id":{"settore":"$settore.nome", "ricavato":"$settore.ricavatoComplessivo"}, "numeroAziende":{$sum:1}}, {"$project":{"settore:1, numeroAziende:1}})
```

```
"executionTimeMillis" : 332,  
"totalKeysExamined" : 0,  
"totalDocsExamined" : 200000,
```

4. Numero di aziende, per paese, con il fatturato totale in ordine decrescente:

```
SELECT Nazione.nome, COUNT(*) as num_companies, SUM(Azienda.fatturato) as tot_revenue  
FROM Nazione JOIN Azienda ON Nazione.nome = Azienda.nazione  
GROUP BY Nazione.nome  
ORDER BY tot_revenue DESC;
```

```
205 rows in set (0,47 sec)
```

```
db.azienda.aggregate({$group:{"_id":{"nazione":"$nazione.nome"}, numAziende:{$sum:1},  
fatturatoTotale:{$sum:"$fatturato"}}, {$project:{nazione.nome:1, numAziende:1, fatturatoTotale:1}},  
{$sort:{fatturatoTotale:-1}})
```

```
"executionTimeMillis" : 307,  
"totalKeysExamined" : 0,  
"totalDocsExamined" : 200000,
```

5. Numero criptovalute possedute delle aziende che sono in paesi con almeno 50 milioni di persone:

```
SELECT Azienda.nome, COUNT(*) AS num  
FROM Nazione JOIN Azienda ON Nazione.nome = Azienda.nazione  
JOIN AziendaPossiedeCriptovaluta ON Azienda.ID = AziendaPossiedeCriptovaluta.ID  
WHERE Nazione.popolazione > 50000000  
GROUP BY Azienda.nome;
```

```
11350 rows in set (0,12 sec)
```

```
db.azienda.aggregate({$match:{nazione.popolazione:{$gt:50000000}}}, {$project:{nome:1, numCriptovalute:{$size:"criptovalutePossedute"}, "nazione.popolazione":1}})
```

```
"executionTimeMillis" : 1069,  
"totalKeysExamined" : 0,  
"totalDocsExamined" : 200000,
```

AZIONE

1. Dire, per ogni tipologia di azione, il valore nominale totale delle azioni che vi appartengono:

```
SELECT azione.classificazioneAzione, sum(valoreNominal)
FROM Azione, Asset
WHERE Azione.ticker = Asset.ticker GROUP BY azione.classificazioneAzione;
```

```
12 rows in set (0.14 sec)
```

```
db.azione.aggregate([
    {"$group": {
        "_id": {"tipologia": "$tipologia.nome"},
        "valoreNominalTotale": {$sum: "$valoreNominal"}
    }},
    {"$project": {
        "_id": 0, "tipologia": 1, "valoreNominalTotale": 1
    }
})
])
```

```
"executionTimeMillis" : 19,
```

2. Si vogliono sapere quali indici hanno valore nominale maggiore di 1000:

```
db.azione.find(
    { indiceAzione: { $elemMatch: { valoreNominal: { $gte: 1000 } } } },
    {indiceAzione:1}
).pretty()
```

```
"executionTimeMillis" : 0,
```

```
SELECT indiceAzione.ticker
FROM IndiceAzione
WHERE IndiceAzione.valoreNominal > 1000;
```

```
5931 rows in set (0.02 sec)
```

6. si vuole sapere il numero di azioni che hanno diritto di voto e diritto di voto straordinario e sono quotate nello stesso indice:

```
db.azione.aggregate( [
    { $match: { "tipologia.dirittoDiVoto": true, "tipologia.dirittoDiVotoStraordinario":true } },
    { $unwind: "$indiceAzione" },
    { $group: { _id: "$indiceAzione.ticker", totale:{ $sum: 1 } } },
])
])
```

```
"executionTimeMillis" : 3,
```

```
SELECT count(*)  
FROM ClassificazioneAzione, Azione, Contiene  
WHERE ClassificazioneAzione.dirittoDiVotoOrdinario = True and ClassificazioneAzione.dirittoDiVotoStraordinario =  
True and ClassificazioneAzione.tipologia = azione.classificazioneAzione and azione.ticker = contiene.tickerAzione  
GROUP BY contiene.tickerIndice;
```

```
3346 rows in set (0.10 sec)
```

7. si vogliono mostrare i ticker delle azioni di aziende italiane quotate in borse americane:

```
db.azione.find(  
{"aziendaEmittente.nazione":"skir", quotaIn:$elemMatch:{nazione:"gfjs"}},  
{ticker:1}  
).pretty()
```

```
"executionTimeMillis" : 0,
```

```
SELECT azione.ticker  
FROM Azione, Azienda, Quotataln, Borsa  
WHERE Azione.aziendaEmittente = azienda.ID and azienda.nazione = "Italy" and Azione.ticker =  
AzioneQuotataln.ticker and Quotataln.borsa = Borsa.nome and Borsa.nazione = "USA" ;
```

```
2 rows in set (0.01 sec)
```

8. mostrare le aziende che hanno emesso azioni che vengono quotate in paesi diversi dai loro:

```
db.azione.aggregate([  
    {$match:{quotataln:$elemMatch:{nazione:$ne:"aziendaEmittente.nazione"}}}  
]).pretty()
```

```
"executionTimeMillis" : 0,
```

```
SELECT azienda.*  
FROM azienda, azione, azioneQuotataln, borsa  
WHERE azione.aziendaEmittente = azienda.ID and azione.ticker = azioneQuotataln.ticker and azioneQuotataln.borsa  
= borsa.nome and borsa.nazione != azienda.nazione;
```

```
46924 rows in set (0.62 sec)
```