

SQL Data Manipulation Language

corso di basi di dati e laboratorio

Prof. Alfio Ferrara

Anno Accademico 2020/2021

Indice

| | | |
|----------|---|-----------|
| 1 | Linguaggio di manipolazione dei dati | 1 |
| 1.1 | Introduzione e esempio | 1 |
| 2 | Interrogazioni | 3 |
| 2.1 | Introduzione alle interrogazioni | 3 |
| 2.2 | Join | 6 |
| 2.3 | Altre condizioni di selezione | 8 |
| 2.4 | Join interni e esterni | 10 |
| 3 | Interrogazioni complesse | 12 |
| 3.1 | Operatori aggregati | 12 |
| 3.2 | Interrogazioni di tipo insiemistico | 16 |
| 3.3 | Interrogazioni nidificate | 17 |
| 3.4 | Subquery correlate | 19 |
| 4 | Piano di esecuzione delle interrogazioni | 22 |
| 4.1 | Explain | 22 |

1 Linguaggio di manipolazione dei dati

1.1 Introduzione e esempio

Schema relazionale di esempio

country(iso3, government, founding_year, min_lon, max_lon, min_lat, max_lat, currency_type)

borders(country, country1)

measure(id, year, value, ref_unit, percentage, label, country, coding_code, coding_category)

name(id, script, language, abbreviated, gazeteer, short, formal, country)

web(url_website, iso3_country, official)

website(url, title, description)

Inserimento

```
INSERT INTO NomeTabella [ ListaAttributi ]  
VALUES (ListadiValori) | SELECTSQL
```

- Inserimento di singole righe:

```
INSERT INTO country(iso3, government, founding_year, currency_type) VALUES  
( 'ITA', 'republic', 1861, € );
```

```
INSERT INTO measure(year, value, label, country) VALUES(2013, 59831093,  
'Population', 'ITA');
```

```
INSERT INTO borders VALUES('ITA', 'FRA');
```

Inserimento di insiemi di righe da dati della DB

```
CREATE TABLE statistics_2013(  
  id INTEGER PRIMARY KEY,  
  value DOUBLE PRECISION,  
  label VARCHAR,  
  country CHAR(3) REFERENCES country(iso3)  
)
```

```
INSERT INTO statistics_2013(id, value, label, country)  
(SELECT id, value, label, country  
FROM measure  
WHERE year=2013)
```

Modifica

```
UPDATE NomeTabella SET
  Attributo = { Espressione | SELECTSQL | NULL | DEFAULT }
  { , Attributo = ... } [ WHERE Condizione ]
```

Aggiornamento di uno o più attributi delle tuple di NomeTabella che verificano (se specificata) la condizione in WHERE. Senza WHERE si esegue la modifica su tutte le righe.

Esempio

Singola riga:

```
UPDATE measure SET value = value*1.05
WHERE label='GDP' AND country='ITA';
```

Il prodotto interno lordo italiano è aumentato del 5 per cento.

Più righe:

```
UPDATE web SET official='f'
```

Stabilisce che ogni sito non è quello ufficiale.

Cancellazione

```
DELETE FROM NomeTabella [ WHERE Condizione ]
```

Senza WHERE: cancellazione di **tutte** le righe della tabella.

con WHERE: cancellazione selettiva di righe della tabella.

Esempio: `DELETE FROM measure WHERE label='GDP' AND country='ITA'`
Cancella il prodotto interno lordo italiano. In WHERE ci può essere una query nidificata.

2 Interrogazioni

2.1 Introduzione alle interrogazioni

Linguaggio SQL di interrogazione

- **Dichiarativo (non procedurale)**, ovvero basato sulla descrizione dell'obiettivo dell'operazione.
- L'interprete SQL del DBMS analizza l'istruzione e formula una interrogazione equivalente nel linguaggio procedurale interno (nascosto all'utente).

- SQL è incorporabile in un programma in linguaggio ospite.
- Le query SQL sono tradotte in modo ottimizzato (**query optimizer**).
- Gli utenti di SQL sono sia gli utenti che interagiscono direttamente con l'interprete sia i programmatori di applicazioni.

L'istruzione SELECT

SELECT ListaAttributi FROM ListaTabelle [WHERE Condizione]

- SELECT: nomi di attributo degli attributi target i cui valori devono essere reperiti con la query.
- FROM: nomi di relazioni su cui la query deve essere valutata.
- WHERE: espressione (booleana) di ricerca che identifica le ennuple che dovranno far parte del risultato della query.

Istanza della base di dati di esempio

COUNTRY

| iso3 | government | founding_year | currency_type |
|------|-------------------------|---------------|---------------|
| FRA | republic | 1792 | € |
| USA | federal republic | 1776 | \$ |
| GBR | constitutional monarchy | 1707 | NULL |
| ITA | republic | 1861 | € |
| ... | ... | ... | ... |

MEASURE

| id | year | value | label | country |
|------|------|------------------|------------|---------|
| 952 | 2013 | 66028467.0 | Population | FRA |
| 957 | 2013 | 2.7390000001e+13 | GDP | FRA |
| 1345 | 2013 | 59831093.0 | Population | ITA |
| 1350 | 2013 | 2.0680000001e+13 | GDP | ITA |
| 2848 | 2013 | 64097085.0 | Population | GBR |
| ... | ... | ... | ... | ... |

Clausola WHERE

- Argomento: espressioni booleane costruite combinando predicati semplici mediante operatori logici AND, OR, NOT.

- Uso di operatori di confronto (=, <>, <, >, <=, >=) fra espressione a sinistra costruita a partire dai valori degli attributi per la riga e un valore costante o un'altra espressione a destra.
- **AND**: selezione delle righe che rendono veri tutti i predicati.
- **OR**: selezione delle righe per cui almeno un predicato è vero.
- **NOT**: unario, inverte il valore di verità del predicato.

Altri operatori in WHERE

- [not] BETWEEN a AND b
- [not] IN (lista valori)
- IS [not] NULL
- [not] LIKE 'pattern'
- [NOT] EXISTS

Esempio.

Reperire codice e forma di governo delle nazioni fondate dopo il 1860.

```
SELECT iso3, government
FROM country
WHERE founding_year > 1860;
```

COUNTRY

| iso3 | government |
|------|------------|
| ITA | republic |

$$\Pi_{iso3, government}(\sigma_{founding_year > 1860}(country))$$

Si noti che le nazioni con founding_year NULL non sono restituite dall'interrogazione.

Pattern matching di stringhe

[NOT] LIKE pattern

Nella specifica del pattern: _ rappresenta un carattere arbitrario % rappresenta una stringa di lunghezza arbitraria.

Esempio.

Reperire tutti i dati delle repubbliche.

```
SELECT * FROM country WHERE government LIKE '%republic%';
```

COUNTRY

| iso3 | government | founding_year | currency_type |
|------|------------------|---------------|---------------|
| AFG | islamic republic | NULL | Af |
| DZA | republic | NULL | DA |
| AGO | republic | NULL | Kz |
| ARG | republic | NULL | \$Arg |
| ARM | republic | NULL | dram |
| AUT | federal republic | NULL | € |
| ... | ... | ... | ... |

Ridenominazione degli attributi

Per le nazioni con prodotto interno lordo superiore a 30.000.000.000.000, mostrare il prodotto interno medio aumentato del 20 per cento.

```
SELECT country, value*1.2 AS new_gdp
FROM measure
WHERE value > 30000000000000 AND label = 'GDP';
```

MEASURE

| country | new_gdp |
|---------|-------------------|
| ASM | 5.54640000012e+13 |
| ARG | 5.81520000012e+13 |
| AUT | 5.01480000012e+13 |
| AZE | 9.12120000012e+13 |
| BHS | 1.00476000001e+14 |
| BRB | 5.11440000012e+13 |
| ... | ... |

2.2 Join**Join**

Reperire codice, governo e popolazione delle nazioni.

I dati sul codice e il governo delle nazioni si trovano nella tabella `country`.

I dati sulla popolazione delle nazioni si trovano nella tabella `measure`.

- Le tuple di *measure* sono collegate a quelle di *country* tramite chiave esterna (*measure.country*).
- Per ricostruire l'informazione cercata, ogni tupla di *measure* va concatenata con la tupla della **corrispondente** *country*.
- La condizione da imporre è l'uguaglianza dei valori di **chiave esterna** di *measure* e corrispondente **chiave primaria** di *country* (**equijoin**).

Esempio.

```
SELECT iso3, government, value AS population
FROM country, measure
WHERE measure.country = country.iso3
AND measure.label = 'Population'
```

| iso3 | government | population |
|------|-------------------------|-------------|
| FRA | republic | 66028467.0 |
| ITA | republic | 59831093.0 |
| GBR | constitutional monarchy | 64097085.0 |
| USA | federal republic | 316128839.0 |
| ... | ... | ... |

Join

- Tutte le tabelle coinvolte nel **JOIN** sono esplicitate nella clausola **FROM**.
- Sul prodotto cartesiano di tali tabelle si applicano le condizioni specificate in **WHERE**.
- Le condizioni del **JOIN** sono scritte in modo esplicito nella clausola **WHERE** (uso dei valori di chiave esterna per collegare tuple di tabelle diverse e combinare i loro dati nel risultato).

Interpretazione algebrica.

```
SELECT iso3, government, value AS population
FROM country, measure
WHERE measure.country = country.iso3
AND measure.label = 'Population'
```

$$\rho_{value \rightarrow population}$$

$$(\Pi_{iso3, government, value}(\sigma_{label='Population'}(\sigma_{measure.country=country.iso3}(measure \times country))))$$

Assenza della selezione

`SELECT * FROM country, measure;` Questa interrogazione produce il prodotto cartesiano delle tabelle `country` e `measure`.

Operatore di Join

La stessa interrogazione vista in precedenza si può esprimere direttamente tramite l'operatore `JOIN` come segue:

```
SELECT iso3, government, value AS population
FROM country JOIN measure ON measure.country = country.iso3
WHERE measure.label = 'Population'
```

Che ha un diretto corrispettivo algebrico nell'uso del join:

$\rho_{value \rightarrow population}$

$$(\Pi_{iso3, government, value}(\sigma_{label='Population'}(measure \bowtie_{country=iso3} country)))$$

Uso di alias e 'self join'

Trovare le nazioni che utilizzano la stessa moneta usata in Francia.

```
SELECT c.iso3
FROM country AS c, country AS m
WHERE c.currency_type = m.currency_type
      AND m.iso3 = 'FRA'
      AND c.iso3 <> m.iso3;
```

Alias di relazione: `c` rappresenta le nazioni, `m` rappresenta la nazione con codice 'FRA'.

2.3 Altre condizioni di selezione

Valori nulli

- Introduzione di condizioni atomiche per verificare se un valore è specificato oppure è nullo.
- Predicato **is null**: selezione delle righe con valori nulli:

`attributo is [not] null`

- Il predicato risulta vero se l'attributo ha valore nullo (`is not null` è la sua negazione).

Eliminazione dei duplicati

- Possibile presenza di righe uguali all'interno di una tabella risultato di un'interrogazione.
- Eliminazione dei duplicati a carico dell'utente: `SELECT DISTINCT`.

Reperire l'elenco delle tipologie di misura disponibili.

```
SELECT DISTINCT label FROM measure;
```

Ordinamento del risultato

Ordinamento delle ennuple risultato secondo uno o più attributi specificati.

```
ORDER BY attributo [ASC | DESC] {, attributo [ASC | DESC]}
```

```
SELECT country, year, value AS population FROM measure  
WHERE label = 'Population'  
ORDER BY value DESC, year;
```

| country | year | population |
|---------|------|--------------|
| CHN | 2013 | 1357380000.0 |
| IND | 2013 | 1252139596.0 |
| USA | 2013 | 316128839.0 |
| IDN | 2013 | 249865631.0 |
| BRA | 2013 | 200361925.0 |
| PAK | 2013 | 182142594.0 |
| NGA | 2013 | 173615345.0 |
| BGD | 2013 | 156594962.0 |
| ... | ... | ... |

Ordinamento con valori nulli

```
UPDATE statistics_2013 SET value = NULL WHERE id = 580;  
SELECT * FROM statistics_2013  
WHERE label = 'Population' ORDER BY value DESC;
```

| id | value | label | country |
|------|--------------|------------|---------|
| 580 | NULL | Population | CHN |
| 1251 | 1252139596.0 | Population | IND |
| 2861 | 316128839.0 | Population | USA |
| 1265 | 249865631.0 | Population | IDN |
| 375 | 200361925.0 | Population | BRA |
| 2052 | 182142594.0 | Population | PAK |
| 1987 | 173615345.0 | Population | NGA |
| 219 | 156594962.0 | Population | BGD |
| ... | ... | ... | ... |

2.4 Join interni e esterni

L'operatore JOIN

- Sintassi per la specifica di join introdotta in SQL-2 che permette di distinguere le condizioni di join dalle altre condizioni.
- Le condizioni di join appaiono nella clausola FROM associate alle tabelle che partecipano al join.
- Tipi di join:

(INNER) JOIN (interno, default coincide con il theta-join).

NATURAL JOIN (join naturale).

RIGHT, LEFT, FULL OUTER JOIN (join esterni).

Inner join

Senza operatore di join:

```
SELECT country.government, name.gazeteer, name.formal
FROM country, name
WHERE country.iso3=name.country AND name.script = 'en-us';
```

Con operatore di join:

```
SELECT country.government, name.gazeteer, name.formal
FROM country JOIN name ON country.iso3=name.country
WHERE name.script = 'en-us';
```

Natural join

- Nell'operazione **NATURAL JOIN** di due relazioni non viene specificata alcuna condizione di join.
- Viene applicata una condizione implicita di equijoin per ciascuna coppia di attributi con lo stesso nome nelle due relazioni.
- Ogni coppia di attributi di questo tipo è inclusa una sola volta nel risultato.
- Se i nomi degli attributi non sono gli stessi nelle due relazioni, si possono ridenominare per farli corrispondere e poi applicare **NATURAL JOIN**.

Join esterni

- (OUTER) JOIN: Esegue un join mantenendo nel risultato tutte le righe che fanno parte di una o di entrambe le relazioni coinvolte.
- LEFT (OUTER) JOIN: Fornisce come risultato il join interno esteso con le tuple della relazione che compare a sinistra del join per le quali non esiste una corrispondente tupla nella relazione di destra.
- RIGHT (OUTER) JOIN: Fornisce come risultato il join interno esteso con le tuple della relazione che compare a destra del join per le quali non esiste una corrispondente tupla nella relazione di sinistra.
- FULL (OUTER) JOIN: Fornisce come risultato il join interno esteso con le tuple di ciascuna relazione per le quali non esiste una corrispondente tupla dall'altra parte.

Esempio.

```
SELECT website.url, web.iso3_country AS country
FROM website LEFT JOIN web ON website.url = web.url.website;
```

| url | country |
|-------------------|---------|
| google.com | NULL |
| gta.gov.zw | ZWE |
| zampost.com.zm | ZMB |
| zambia-mining.com | ZMB |
| statehouse.gov.zm | ZMB |
| zamstats.gov.zm | ZMB |
| ... | ... |

3 Interrogazioni complesse

3.1 Operatori aggregati

Operatori aggregati

SQL mette a disposizione un insieme di **operatori aggregati** per derivare valori aggregati a partire da insiemi di tuple di relazioni. Ad esempio, la popolazione totale dei delle nazioni della base di

dati non è una caratteristica di una singola tupla della relazione *measure*, ma deriva dalla somma delle popolazioni di ogni *measure*, ovvero da un calcolo effettuato su più tuple.

Operatori aggregati

I principali operatori aggregati in SQL sono: COUNT, SUM, MAX, MIN, AVG

- Prima si esegue l'interrogazione considerando le clausole FROM e WHERE della query.
- L'operatore aggregato viene applicato al risultato ottenuto.

L'operatore COUNT

- COUNT (*): restituisce il numero di righe di una relazione.
- COUNT ALL ListaAttributi: restituisce il numero di righe della relazione che possiedono valori diversi da NULL per gli attributi specificati.
- COUNT DISTINCT ListaAttributi: restituisce il numero di valori diversi all'interno della relazione per gli attributi specificati.

Esempi.

```
SELECT COUNT(*) FROM country;
```

Restituisce il numero di nazioni.

```
SELECT COUNT(government) FROM country;
```

Restituisce il numero di nazioni di cui è presente la forma di governo.

```
SELECT COUNT(DISTINCT government) FROM country;
```

Restituisce il numero di forme di governo.

Esempi.

Determinare il massimo, il minimo e la media delle popolazioni nell'anno 2013.

```
SELECT MAX(value), MIN(value), AVG(value)
FROM measure
WHERE year = 2013 AND label = 'Population';
```

Determinare la somma delle popolazioni di Italia e Francia.

```
SELECT SUM(value) FROM measure
WHERE (country = 'ITA' OR country = 'FRA') AND label = 'Population';
```

Determinare la popolazione massima fra le repubbliche.

```
SELECT MAX(value)
FROM country JOIN measure ON country.iso3 = measure.country
WHERE government LIKE '%republic%'
AND label='Population';
```

Errori.

Trovare l'errore nella seguente interrogazione:

```
SELECT country.iso3, MAX(value)
FROM country JOIN measure ON country.iso3 = measure.country
WHERE government LIKE '%republic%'
AND label='Population';
```

L'interrogazione chiede di restituire un valore aggregato insieme a valori specifici della singola tupla. Tale approccio è concettualmente (e di fatto) scorretto. **L'operatore aggregato necessita di lavorare su dati aggregati.**

Raggruppamento.

- Necessità di applicare operatori aggregati a sottogruppi di tuple di una relazione in base al valore di uno o più attributi.
- Clausola `GROUP BY ListaAttributi` per eseguire il raggruppamento.
- Prima si effettua il raggruppamento e poi si applica l'operatore aggregato a **ciascun sottogruppo individuato**.

Esempio.

Selezionare la popolazione massima e media per tipologia di governo.

```
SELECT country.government, MAX(value), AVG(value)
FROM country JOIN measure ON country.iso3 = measure.country
WHERE label='Population'
GROUP BY government;
```

| government | max | avg |
|------------------|--------------|-------------|
| communist state | 1357380000.0 | 366281064.0 |
| federal republic | 1252139596.0 | 246032223.5 |
| federation | 143499861.0 | 143499861.0 |
| ... | ... | ... |

Condizioni di aggregazione

- Esigenza di applicare gli operatori aggregati solo su raggruppamenti che verificano un insieme di condizioni date.
- Clausola **HAVING** che specifica una **condizione** su un gruppo di tuple **associata al valore degli attributi di raggruppamento**.
- Solo i raggruppamenti che soddisfano la condizione specificata nella clausola **HAVING** sono inclusi nel risultato dell'interrogazione

Esempio.

```
SELECT country.government, MAX(value) AS maxp, AVG(value) AS avgp
FROM country JOIN measure ON country.iso3 = measure.country
WHERE label='Population'
```

```
GROUP BY government
HAVING MAX(value) < 60000000;
```

| government | maxp | avgp |
|---------------------------|------------|------------|
| parliamentary monarchy | 46647421.0 | 46647421.0 |
| a parliamentary democracy | 35158304.0 | 35158304.0 |
| islamic republic | 30551674.0 | 30551674.0 |
| ... | ... | ... |

Predicati sui raggruppamenti

Per le nazioni di cui siano noti almeno due siti web, selezionare il codice e il numero di siti.

```
SELECT iso3_country, COUNT(*)
FROM web
GROUP BY iso3_country
HAVING COUNT(*) > 1;
```

Comportamento con valori nulli

- COUNT(*) conta le righe, mentre COUNT(attributo) conta i valori di 'attributo'.
- Ciò implica un comportamento diverso in presenza di valori nulli.
- SELECT COUNT(*) FROM country → 249.
- SELECT COUNT(government) FROM country → 217.
- Ciò significa che ci sono 32 nazioni in cui il valore di 'government' è NULL.

Altri operatori con valori nulli

```
SELECT * FROM statistics_2013
WHERE (country = 'CHN' OR country = 'ITA')
AND label = 'Population';
```

| id | value | label | country |
|------|------------|------------|---------|
| 1345 | 59831093.0 | Population | ITA |
| 580 | NULL | Population | CHN |

```
SELECT COUNT(*), COUNT(value),  
SUM(value), MIN(value), MAX(value), AVG(value)  
FROM statistics_2013  
WHERE (country = 'CHN' OR country = 'ITA')  
AND label = 'Population';
```

| count | count | sum | min | max | avg |
|-------|-------|------------|------------|------------|------------|
| 2 | 1 | 59831093.0 | 59831093.0 | 59831093.0 | 59831093.0 |

3.2 Interrogazioni di tipo insiemistico

Operatori insiemistici

- Le operazioni insiemistiche UNION, INTERSECT, EXCEPT sono state incorporate direttamente in SQL.
- **Default**: eliminazione dei duplicati.
- **ALL**: mantenimento dei duplicati nel risultato.

Proprietà degli operatori

- **UNION**: arricchisce la potenza espressiva di SQL e permette di scrivere interrogazioni altrimenti non formulabili.
- Interrogazioni con **INTERSECT** e **EXCEPT** possono essere espresse utilizzando altri costrutti del linguaggio (cfr. *interrogazioni nidificate*).

Unione

Determinare l'insieme dei nomi brevi e formali delle nazioni.

```
SELECT short FROM name  
UNION  
SELECT formal FROM name;
```

Nota: UNION ALL preserva i duplicati, quindi se ci sono stringhe che sono sia nomi brevi sia formali compariranno più volte.

Intersezione

Determinare i nomi sono sia brevi sia formali.

```
SELECT short FROM name
INTERSECT
SELECT formal FROM name;
```

Sottrazione

Trovare il codice delle nazioni per le quali non abbiamo alcuna misura.

```
SELECT iso3 FROM country
EXCEPT
SELECT country FROM measure;
```

3.3 Interrogazioni nidificate

Interrogazioni nidificate

- Argomento della clausola WHERE è un predicato in cui si confronta un valore (risultato di una espressione valutata su una singola riga) con il risultato dell'esecuzione di una query SQL interna.
- **Operatori:** ALL, ANY
- Per estendere i normali operatori di confronto (=,<>, >, <, >=, <=) al problema di confrontare un singolo valore con l'insieme dei valori risultato della query interna.

Esempio.

Trovare la popolazione delle monarchie.

```
SELECT country, value AS population
FROM country JOIN measure ON iso3 = country
WHERE government LIKE '%monarchy%'
AND label = 'Population';
```

La stessa query si può esprimere in forma nidificata.

```
SELECT country, value AS population
FROM measure
WHERE label = 'Population' AND country =ANY (
```

```
SELECT iso3 FROM country WHERE government LIKE '%monarchy%');
```

N.B. Operatori **IN** e **NOT IN** per testare l'appartenenza ad e l'esclusione da un insieme (identici a '= ANY' e '<> ALL').

Esempio.

Trovare le nazioni che usano la stessa moneta della Spagna.

```
SELECT iso3
FROM country
WHERE currency_type IN (
    SELECT currency_type FROM country WHERE iso3='ESP'
);
```

Esempio.

Trovare le nazioni di cui non abbiamo il nome in inglese americano.

```
SELECT iso3 FROM country WHERE iso3 NOT IN(
    SELECT country FROM name
    WHERE script = 'en-us')
```

La stessa query si può esprimere mediante sottrazione.

```
SELECT iso3 FROM country
EXCEPT
SELECT country FROM name WHERE script = 'en-us';
```

In algebra relazionale:

$$\Pi_{iso3}(country) - \Pi_{country}(\sigma_{script='en-us'}(name))$$

Esempio.

Le interrogazioni con MIN e MAX si possono esprimere mediante query nidificate.

Trovare la nazione più popolata.

```
SELECT country FROM measure
WHERE label='Population' AND
value = (SELECT MAX(value) FROM measure WHERE label='Population');

SELECT country FROM measure
```

```
WHERE label='Population' AND
AND value >=ALL ( SELECT DISTINCT value FROM measure
WHERE label='Population' AND value IS NOT NULL);
```

3.4 Subquery correlate

Query correlate

- Negli esempi visti ogni subquery viene eseguita una sola volta e l'insieme dei valori risultante è usato per la valutazione della clausola WHERE della query esterna.
- E' possibile definire subquery che sono eseguite ripetutamente **per ogni tupla candidata** considerata nella valutazione della query esterna.

Esempio

Determinare le nazioni (e il corrispondente tipo di governo) il cui prodotto lordo superiore alla media del prodotto delle nazioni che hanno la stessa forma di governo.

E' necessaria una query esterna per la selezione delle nazioni in base a un predicato sul prodotto interno lordo; tale query avrà la forma:

```
SELECT M.country, C.government
FROM country AS C JOIN measure AS M ON C.iso3 = M.country
WHERE M.label='GDP' AND M.value >= (
Media del prodotto lordo per il tipo di governo della nazione considerata
)
```

Esempio

La subquery dovrà calcolare la media del prodotto interno lordo relativo al tipo di governo della tupla correlata; tale subquery avrà la forma:

```
SELECT AVG(M1.value)
FROM country AS C1 JOIN measure AS M1 ON C1.iso3 = M1.country
WHERE M1.label='GDP' AND C1.government =
(governo della nazione in esame nella query esterna)
```

Query correlate

- Ogni volta che la query esterna considera una tupla candidata, viene invocata la subquery e viene passato il governo della nazione in esame.
- La subquery calcola quindi la media del prodotto interno lordo delle nazioni che hanno quella forma di governo e restituisce tale valore alla query esterna.
- La query esterna può quindi confrontare il prodotto interno lordo della nazione in esame con il valore restituito dalla subquery.

Query correlate

- Questo tipo di query è chiamato correlato, perché ogni esecuzione della subquery è correlata al valore di uno o più attributi delle tuple candidate nella query esterna.
- Per poter far riferimento alle colonne delle tuple candidate nella query esterna si fa uso degli alias di relazione.
- L'alias di relazione è definito nella query esterna e riferito nella query interna.

Esempio

Determinare le nazioni (e il corrispondente tipo di governo) il cui prodotto lordo superiore alla media del prodotto delle nazioni che hanno la stessa forma di governo.

```
SELECT M.country, C.government
FROM country AS C JOIN measure AS M ON C.iso3 = M.country
WHERE M.label='GDP' AND M.value >= (
    SELECT AVG(M1.value)
    FROM country AS C1 JOIN measure AS M1 ON C1.iso3 = M1.country
    WHERE M1.label='GDP' AND C1.government = C.government
)
```

Predicato EXISTS

- Il predicato `EXISTS(sq)` restituisce il valore booleano `TRUE` se la subquery restituisce almeno un tupla; restituisce il valore booleano `FALSE` altrimenti.
- Il predicato `NOT EXISTS(sq)` restituisce il valore booleano `TRUE` se la subquery non restituisce alcuna tupla; restituisce il valore booleano `FALSE` altrimenti.

Esempio

Trovare le nazioni con lo stesso nome in lingue diverse.

```
SELECT country, gazeteer, script FROM name N
WHERE EXISTS (
  SELECT * FROM name X
  WHERE N.gazeteer = X.gazeteer AND N.script <> X.script );
```

Divisione

Le subquery correlate e l'operatore di NOT EXISTS sono particolarmente utili per implementare l'operazione di divisione in SQL.

Si consideri lo schema seguente:

COUNTRY(iso3, government)

PERSON(code, name, surname)

TRAVEL(country, person, date)

Trovare le persone che hanno visitato tutte le repubbliche.

$$\Pi_{person, country}(TRAVEL) \div \rho_{iso3 \rightarrow country}(\Pi_{iso3}(\sigma_{government='republic'}(COUNTRY)))$$

Divisione

- La specifica della divisione in SQL fa uso dell'operatore NOT EXISTS e richiede di ragionare in base al concetto di **controesempio**.
- Una persona A verifica l'interrogazione se **non esiste** una repubblica che A **non** abbia visitato.
- Ovvero, data una tupla A di persona non deve esistere una tupla C di country tale che C sia una repubblica e che non esista una relazione TRAVEL fra A e C .

Esempio

```
SELECT code, name, surname FROM person A
WHERE NOT EXISTS (
  SELECT * FROM country C
```

```
WHERE government LIKE '%republic%'
AND NOT EXISTS (
  SELECT * FROM travel T
  WHERE T.person=A.code AND T.country=C.iso3
));
```

4 Piano di esecuzione delle interrogazioni

4.1 Explain

Piano di esecuzione

- Le interrogazioni SQL sono trasformate in una successione di operazioni atomiche sui dati denominata ‘piano di esecuzione’.
- E’ possibile accedere al piano di esecuzione attraverso la clausola EXPLAIN.
- A titolo di esempio, confrontiamo di seguito tre interrogazioni fra loro equivalenti.

Esempio 1

```
EXPLAIN
SELECT iso3, government
FROM country, measure
WHERE measure.country = country.iso3
AND measure.label = 'Population'
AND measure.value > 50000000;

Hash Join (cost=11.60..84.88 rows=18 width=28)
  Hash Cond: (measure.country = country.iso3)
    -> Seq Scan on measure (cost=0.00..73.03 rows=18 width=4)
        Filter: ((value > 50000000::double precision) AND
            ((label)::text = 'Population'::text))
    -> Hash (cost=8.49..8.49 rows=249 width=28)
        -> Seq Scan on country (cost=0.00..8.49 rows=249 width=28)
```

Esempio 2

```
EXPLAIN
SELECT iso3, government
```

```
FROM country JOIN measure
ON measure.country = country.iso3
WHERE measure.label = 'Population'
AND measure.value > 500000000;
```

```
Hash Join (cost=11.60..84.88 rows=18 width=28)
  Hash Cond: (measure.country = country.iso3)
  -> Seq Scan on measure (cost=0.00..73.03 rows=18 width=4)
    Filter: ((value > 500000000::double precision) AND
      ((label)::text = 'Population'::text))
  -> Hash (cost=8.49..8.49 rows=249 width=28)
    -> Seq Scan on country (cost=0.00..8.49 rows=249 width=28)
```

Esempio 3

```
EXPLAIN
SELECT iso3, government
FROM country
WHERE iso3 IN (
  SELECT country FROM measure
  WHERE measure.label = 'Population'
  AND measure.value > 500000000);
```

```
Nested Loop (cost=73.08..77.21 rows=18 width=28)
  -> HashAggregate (cost=73.08..73.09 rows=2 width=4)
    -> Seq Scan on measure (cost=0.00..73.03 rows=18 width=4)
      Filter: ((value > 500000000::double precision) AND
        ((label)::text = 'Population'::text))
  -> Index Scan using country_pk on country (cost=0.00..2.05 rows=1 width=28)
    Index Cond: (country.iso3 = measure.country)
```

Esempio sui costi delle interrogazioni correlate

```
SELECT M.country, C.government
FROM country AS C JOIN measure AS M ON C.iso3 = M.country
WHERE M.label='GDP' AND C.government LIKE '%republic%'
AND M.value >= ALL(
  SELECT AVG(M1.value)
  FROM country AS C1 JOIN measure AS M1 ON C1.iso3 = M1.country
```

```
WHERE M1.label='GDP' AND C1.government = C.government  
) ;
```

Tempo esecuzione indicativo: 0.0891170501709

Esempio sui costi delle interrogazioni correlate

```
SELECT M.country, C.government  
FROM country AS C JOIN measure AS M ON C.iso3 = M.country  
WHERE M.label='GDP' AND C.government LIKE '%republic%'  
AND M.value >= [costo pre-calcolato];
```

A cui aggiungere la subquery:

```
SELECT AVG(M1.value)  
FROM country AS C1 JOIN measure AS M1 ON C1.iso3 = M1.country  
WHERE M1.label='GDP' AND C1.government LIKE '%republic%'
```

Tempo cumulativo esecuzione (indicativo): 0.00635099411011

Esercizio: utilizzare la clausola EXPLAIN per verificare il piano di esecuzione.