

Schema fisico

workload

Per stimare un workload, pensando alle operazioni che possono essere fatte piu' spesso sulla base di dati, abbiamo ragionato principalmente sui prodotti, sui donatori (e quindi le donazioni) e sui turni di lavoro.

Principalmente, sui prodotti verranno effettuate molte operazioni di selezione e inserimento, molto raramente aggiornamenti e cancellazioni. Le selezioni verranno effettuate in generale per visualizzare tutti i prodotti, per controllare gli acquisti di un dato cliente, per vedere quali prodotti hanno superato la data di scadenza massima (e quindi sono da scaricare) ecc ecc.

Una condizione che abbiamo notato potrebbe dare problemi a livello di efficienza e' cercare, per esempio, tutti i prodotti scaricati fino alla data di oggi

```
-- Selezione di tutti i prodotti scaricati
SELECT *
FROM prodotti
WHERE data_scarico <= CURRENT_DATE;
```

lo schema fisico per questa interrogazione e' la seguente:

```
Seq Scan on prodotti (cost=0.00..153.52 rows=54 width=33) (actual time=0.047..0.541 rows=54 loops=1)
  Filter: (data_scarico <= CURRENT_DATE)
  Rows Removed by Filter: 6514
Planning Time: 0.056 ms
Execution Time: 0.551 ms
```

Viene ovviamente eseguita una scansione sequenziale, perche' non e' presente nessun indice $I_{data_scarico}(prodotti)$. Eventualmente, sarebbe un indice ad albero, siccome la maggior parte delle selezioni di questo tipo sarebbero in un range.

Sempre per la stessa colonna in prodotti, abbiamo pensato ad una query che controlla quale volontario ha effettuato lo scarico di un dato prodotto (poniamo per esempio il prodotto 1)

```
SELECT *
FROM volontario v
JOIN scarichi s ON s.volontario = v.id
JOIN prodotti p ON p.data_scarico = s.data AND p.ora_scarico = s.ora
WHERE p.id = 1
```

NOTE:

- l'indice $I_{CF}(clienti)$ potrebbe essere tranquillamente ad hash

```
-- Per visualizzare gli indici
SELECT C.oid, relname, indexrelid, relam as tipo_indice, indnatts, indisunique, indisprimary, indiscluster
FROM (pg_namespace N JOIN pg_class C ON N.oid = C.relnamespace) JOIN pg_index ON C.oid = indexrelid
WHERE N.nspname = 'social_market'
```

- Inserire indice $I_{data_scarico}(prodotti)$ ad albero

```
Seq Scan on prodotti (cost=0.00..169.94 rows=6568 width=33) (actual time=0.014..4.023 rows=6568 loops=1)
  Filter: (scadenza <= CURRENT_DATE)
Planning Time: 0.117 ms
Execution Time: 4.160 ms
```

```
-- f_selettivita: 1/6568
```

tabella	n_pagine
clienti	215
familiari	942
telefoni	192
email	288
fasce_orarie	0
appuntamenti	146
volontari	210
scarichi	96
scorte	0
prodotti	59