Progetto per il corso di Basi di Dati

Consegna ed Istruzioni

Il progetto può essere svolto da soli o in gruppo. Un gruppo deve essere formato preferibilmente da 2 persone, 3 al massimo.

Il progetto prevede lo studio e la realizzazione e l'ottimizzazione di un database, la conversione e il caricamento di dati nel database progettato e l'analisi dei dati attraverso la realizzazione di query.

Il progetto è quindi diviso in 3 sezioni:

1. Creazione del database
2. Inserimento dei dati (e loro eventuale normalizzazione e conversione di formato)
3. Progettazione delle Query SQL per l'analisi dei dati

I criteri di valutazione terranno conto di parametri quali:

1. La corretta importazione dei dati, tale da non causare una perdita di dati o un'inserimento di dati erronei.
2. Il tempo necessario al caricamento e alla formattazione dei dati
3. Lo spazio occupato su disco dal database considerando sia i dati che le strutture ausiliare (e.s. indici)
4. La correttezza delle risposte delle query richieste
5. La velocità di esecuzione delle query

Chiaramente tutti i dati originari devono essere presenti nel database una volta importati (seppure in formato o con schema diverso). Esempio: se nei file di dati è presente un certo numero di categorie distinte, od un certo numero di commenti distinti, lo stesso numero di categorie e commenti deve essere presente nel database importato (come risultato di una opportuna query).

Il progetto finale deve:

1. essere consegnato in un file compresso `tar.gz` con nome `db2015_nomeGruppo.tar.gz`
2. contenere un file `creazione_db.sql` che assume l'esistenza di un database con nome `db2015` ed esegue la creazione degli schemi e delle tabelle necessarie.
3. contenere un file `importazione_dati.sql` che assume l'esistenza dei file `.csv` nella cartella `/tmp/dati` ed esegue tutte le operazioni necessarie per importare i dati dai file alle tabelle del punto precedente, ed esegue ogni necessaria operazione di pulizia e formattazione delle tabelle e dei dati.
4. contenere un file `query_XX.sql` per ogni query assegnata (dove `XX` è il numero della query)
5. contenere un file `report.pdf` - di massimo una pagina - contenente eventuale descrizione di qualsiasi

assunzione o scelta tecnica ritenuta rilevante e non ovvia

> Il progetto deve essere inviato a ml@disi.unitn.eu con oggetto: [Progetto DB] NomeGruppo
> In Cc. all'email devono esserci tutti i componenti del gruppo.
>
> NB.1 L'archivio contenente i file del progetto consegnato non deve contenere file di dati
>
> NB.2 I file verrano eseguiti attraverso il comando `psql < nome_file.sql`, oppure all'interno di postgres con il comando `\i nome_file.sql`

I file SQL verrano eseguiti su due macchine ed i tempi di esecuzione misurati su entrambe.

Clotho

```
PostgreSQL 9.3.6
compiled by gcc (Ubuntu 4.8.2-19ubuntu1) 4.8.2, 64-bit
on Intel(R) Pentium(R) D CPU 3.40GHz — No of Cores:2
with 8GB of RAM
```

Snowwhite

```
PostgreSQL 9.3.6
compiled by gcc (Ubuntu 4.8.2-19ubuntu1) 4.8.2, 64-bit
on Intel(R) Xeon(R) CPU E5-2440 0 @ 2.40GHz — No of Cores:24
with 198GB of RAM
```

Contesto & Dati

Si assuma di dover rispondere alle esigenze di una agenzia di marketing che vuole condurre un'analisi di mercato sulle aziende, imprese e attività locali. A questo scopo ha acquistato dei dati da un rivenditore. Ha così ottenuto un set di dati che contiene informazioni su attività e imprese locali, recensioni delle stesse e informazioni sugli utenti per un campione di 10 città in 4 paesi diversi.

In particolari i dati raccolti sono stati suddivisi nei seguenti file:

1. `review-votes.csv` : ogni linea contiene una recensione di un utente per una attività locale e il giudizio degli utenti per quella recensione. Una recensione può ricevere più di un giudizio in diverse categorie. Contiene i seguenti campi:
    1. `record_type` : il tipo di record è `review` ,
    2. `business_id` : il codice identificativo anonimizzato dell'attività recensita,

3. `user_id` : l'identificativo anonimizzato dell'utente che ha scritto la recensione,
4. `stars` : il numero di stelle attribuite dalla recensione all'attività recensita,
5. `text` : il testo della recensione,
6. `date` : la data della recensione,
7. `vote_type` il tipo di voto attribuito alla recensione,
8. `count` il numero di persone che hanno espresso il voto

2. `business-categories.csv` : ogni linea contiene i dettagli di una attività e una delle categorie a cui appartiene. Un'attività può appartenere a multiple categorie.

    1. `record_type` : il tipo di record è `category` ,
    2. `business_id` : il codice identificativo anonimizzato dell'attività categorizzata,
    3. `name` : il nome dell'attività,
    4. `full_address` : l'indirizzo completo della sede dell'attività,
    5. `city` : la città in cui l'attività è situata,
    6. `state` : lo stato in cui l'attività è situata,
    7. `stars` : il numero medio di stelle date in tutte le recensioni per questa attività,
    8. `review_count` il numero totale di recensioni presenti per questa attività,
    9. `open` : indica se l'attività è ancora attiva o se ha cessato,
    10. `category` : una delle categorie a cui è associata questa attività.

3. `business-neighborhoods.csv` : ogni linea contiene la locazione geografica dell'attività e il nome del quartiere in cui è situata, alcune attività possono essere situate all'intersezione di più di un quartiere

    1. `record_type` : il tipo di record è `location` ,
    2. `business_id` : il codice identificativo anonimizzato dell'attività,
    3. `name` : il nome dell'attività,
    4. `city` : la città in cui l'attività è situata,
    5. `state` : lo stato in cui l'attività è situata,
    6. `latitude` : la coordinata relativa alla latitudine,
    7. `longitude` : la coordinata relativa alla longitudine,
    8. `neighborhood` : il nome di uno dei quartieri in cui si situa l'attività.

4. `business-openhours.csv` : ogni linea contiene gli orari di apertura di una attività in un singolo giorno

    1. `record_type` : il tipo di record è `open-hours` ,
    2. `business_id` : il codice identificativo anonimizzato dell'attività,
    3. `name` : il nome dell'attività,
    4. `full_address` : l'indirizzo completo della sede dell'attività,
    5. `city` : la città in cui l'attività è situata,
    6. `state` : lo stato in cui l'attività è situata,

7. `open` : indica se l'attività è ancora attiva o se ha cessato,
8. `day` : uno dei giorni di apertura dell'attività
9. `opens` : l'orario a cui l'attività apre nel giorno
10. `closes` : l'orario a cui l'attività chiude nel giorno.

5. `user-profiles.csv` : ogni linea contiene informazioni generiche su di un singolo utente

    1. `record_type` : il tipo di record è `user` ,
    2. `user_id` : il codice identificativo anonimizzato dell'utente,
    3. `name` : il nome dell'utente,
    4. `review_count` : il numero di recensioni scritte dall'utente,
    5. `average_stars` : il numero medio di stelle date dall'utente,
    6. `registered_on` : la data di iscrizione dell'utente,
    7. `fans_count` : il numero di utenti `fans` ,
    8. `elite_years_count` : il numero di anni in cui l'utente è stato categorizzato `elite` .

6. `user-compliments.csv` : ogni linea contiene un complimento fatto all'utente da altri utenti e il numero di utenti che condividono il complimento

    1. `record_type` : il tipo di record è `compliment` ,
    2. `user_id` : il codice identificativo anonimizzato dell'utente,
    3. `name` : il nome dell'utente,
    4. `compliment_type` : il tipo di complimento ricevuto dall'utente,
    5. `num_compliments_of_this_type` : il numero di utenti che hanno espresso lo stesso tipo di complimento.

7. `user-friends.csv` : ogni linea contiene una relazione di amicizia tra un utente e un amico

    1. `record_type` : il tipo di record è `friend` ,
    2. `user_id` : il codice identificativo anonimizzato dell'utente,
    3. `name` : il nome dell'utente,
    4. `friend_id` : il codice identificativo anonimizzato dell'utente amico.

8. `user-votes.csv` : ogni linea contiene un voto fatto dall'utente ad altre recensioni ed il numero di tali voti espressi

    1. `record_type` : il tipo di record è `user-vote` ,
    2. `user_id` : il codice identificativo anonimizzato dell'utente,
    3. `name` : il nome dell'utente,
    4. `vote_type` : il tipo di voto espresso dall'utente,
    5. `count` : il numero di voti di questo tipo espressi dall'utente.

Snapshot dei Dati

Per svolgere il progetto e le query, viene fornito un campione di circa `1%` dei dati totali disponibili.

Il campione di dati è scaricabile al link: j.mp/Db2015DatiProgetto

Il progetto verrà poi testato e valutato su due dataset distinti:

1. Un campione del 35% dei dati su `Clotho`
2. Il set completo dei dati su `Snowwhite`

Query da Realizzare

- [query_00] Ricostruire i file `.csv` utilizzati in input ordinandoli su `business_id`, `user_id` o entrambi (a seconda dei campi contenuti).

> Il resto delle query verrà mandato via email agli studenti registrati