

VISUALIZZATORE PIANI SQL

ANALISI DEI REQUISITI

INFORMAZIONI DOCUMENTO

Titolo documento:	Analisi dei requisiti
Data creazione:	2011/10/19
Versione attuale:	1.0.0
Stato del documento:	Formale a uso esterno
Nome file:	Analisi_dei_requisiti_1.0.0.pdf
Redazione:	Luca Fongaro

DATA	VERSIONE	REDATTORI	MOTIVO DELLA MODIFICA
2011/10/20	1.0.0	Luca Fongaro	RILASCIO VERSIONE Rilascio della versione 1.0
2011/10/19	0.1.0	Luca Fongaro	PRIMA STESURA. Prima stesura del documento

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Riferimenti	1
2	Descrizione Generale	2
2.1	Contesto d'uso del prodotto	2
2.2	Tecnologie utilizzate e ambiente di esecuzione	2
2.3	Funzioni dell'interfaccia	2
2.4	Tipologie di utenti	3
2.5	Vincoli generali	3
3	Diagrammi Use Case	4
3.1	Diagramma Use Case Utente	4
3.2	Diagramma Use Case Utente - Manipolazione Struttura	5
4	Requisiti	6
4.1	Requisiti funzionali	7
4.1.1	Obbligatori	7
4.1.2	Desiderabili	7
4.1.3	Opzionali	7
4.2	Requisiti prestazionali	7
4.2.1	Obbligatori	7
4.3	Requisiti di qualità	7
4.3.1	Obbligatori	7
4.4	Requisiti di vincolo	7
4.4.1	Obbligatori	7
5	Tabella tracciamento requisiti funzionali - Diagramma Use Case	8

Elenco delle figure

1	Diagramma use case 1	4
2	Diagramma use case 1.2	5

Elenco delle tabelle

1	Tracciamento dei requisiti	8
---	--------------------------------------	---

1 Introduzione

1.1 Scopo del documento

Il presente documento contiene una descrizione dei requisiti e dei casi d'uso emersi dall'attività di analisi eseguita, sul progetto di stage *Visualizzatore piani SQL*. Lo scopo finale è quello di fornire al committente una visuale dettagliata sulle tecnologie implementative che verranno utilizzate e sulle funzionalità implementate per svolgere questo progetto.

1.2 Scopo del prodotto

Il visualizzatore di piani SQL ha come obiettivo il fornire una visione semplice e completa di un piano di esecuzione SQL. Creando quest'interfaccia sarà possibile avere una visione più chiara di come una *query* verrà eseguita da un determinato database potendo quindi ottimizzarla nel caso si accerti la presenza di operazione onerose o da evitare. L'interfaccia sarà parte integrante del tool *Visual Query* che è una componente del prodotto software *SITEPAINTER Portal Studio*, di proprietà dell'azienda *Zucchetti SpA*.

1.3 Riferimenti

- JavaScript Documentation <https://developer.mozilla.org/en/JavaScript>
- Raphael Reference <http://dmitrybaranovskiy.github.io/raphael/reference.html>
- XHTML Reference http://www.w3schools.com/html/html_xhtml.asp
- HTML5 http://www.w3schools.com/html5/html5_reference.asp

2 Descrizione Generale

2.1 Contesto d'uso del prodotto

Il risultato finale di questo progetto è parte integrante di *Visual Query* un tool il cui scopo è aiutare a gestire un database. Compito specifico di quest'interfaccia è visualizzare il costo delle singole operazioni che compongono una *query* specifica. Per permettere questo verrà disegnato un grafico con struttura ad albero, la quale sarà composta da nodi che rappresentano i singoli passi che il database opererebbe se la *query* fosse effettivamente eseguita. Ogni nodo avrà un nome, che descrive l'operazione, un colore, che rappresenta un giudizio sull'operazione stessa stabilito dall'applicazione lato server, e altre informazioni. Ogni nodo potrà avere zero o più figli associati che rappresentano altre operazioni da eseguire per completare l'operazione del padre.

2.2 Tecnologie utilizzate e ambiente di esecuzione

L'interfaccia è creata basandosi sulle tecnologie *javascript* e *XHTML* e *HTML 5* e l'utilizzo della libreria *Raphael*, l'interazione sarà possibile tramite i browser più recenti. Il prodotto intero, ovvero *SITEPAINTER Portal Studio* risiederà, invece, su un server di proprietà di un'azienda che ha acquistato il prodotto.

2.3 Funzioni dell'interfaccia

L'interfaccia permetterà all'utente di avere tutte le informazioni necessarie per analizzare i singoli passi della *query*, visualizzando le informazioni più importanti ovvero nome dell'operazione e il giudizio ad esso associata, ovvero il colore, che sarà di cinque tipologie:

- Rosso: indica le operazioni che sono da evitare, in quanto fonte di un eccessivo costo;
- Giallo: indica le operazioni che possono essere comunque fonte di un eccessivo carico sul database;
- Verde: operazioni il cui risultato è direttamente proporzionale al numero di record ritornati (es. *INDEX SCAN*);
- Blu: operazioni il cui risultato non è direttamente collegato al numero di record ritornati e che indicano che il giudizio assegnato va valutato in base ad altre operazioni ad essa associate (es. *DISTINCT*, *GROUP BY*);
- Grigio scuro: operazioni per le quali non è stato possibile stabilire un giudizio in quanto non sono riconosciute.

Sarà possibile anche esaminare delle informazioni aggiuntive sull'operazione.

Tutte le operazioni saranno organizzate in un grafico con struttura ad albero che permetterà di comprendere l'ordine di esecuzione dei singoli passi. All'utente sarà anche offerta la possibilità di manipolare questa struttura potendo spostare singoli elementi o interi sotto-alberi e comprimendo i sotto alberi di un certo nodo.

2.4 Tipologie di utenti

Per quest'interfaccia è prevista un'unica tipologia di utente, ovvero una persona con competenze di *SQL* che sarà agevolata nel compito di ottimizzare le *query* create per consultare un determinato database.

2.5 Vincoli generali

L'interfaccia dovrà essere sviluppata con tecnologia *javascript* e dovrà essere correttamente fruibile nei browser:

- Firefox versione 7.0 o superiore
- Internet Explorer versione 8.0 o superiore
- Opera versione 11.0 o superiore
- Chrome versione 14.0 o superiore
- Safari versione 5.0 o superiore

L'interfaccia dovrà essere sviluppata basandosi sulle tecnologie *javascript*, *XHTML* e *HTML5* e sulla libreria *Raphael*.

3 Diagrammi Use Case

3.1 Diagramma Use Case Utente

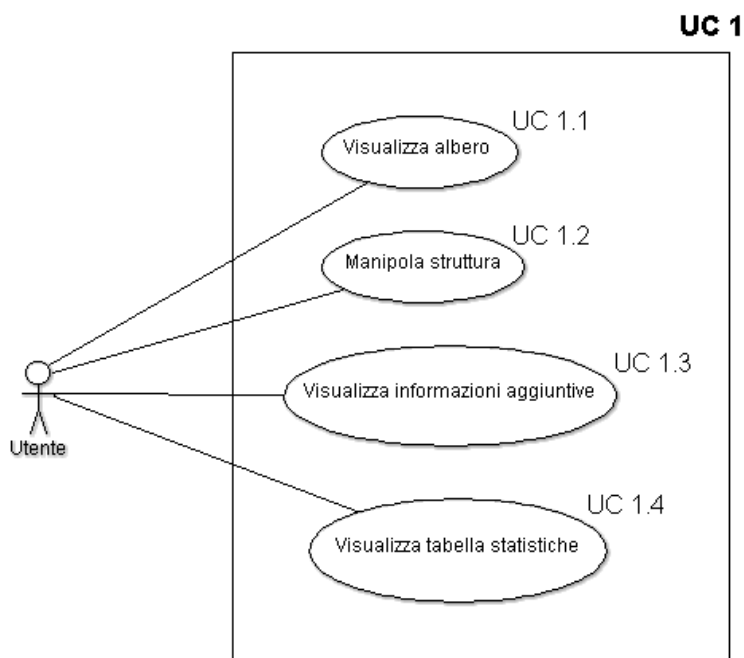


Figura 1: Diagramma use case 1

- Attori principali: Utente
- Precondizioni:
 - La struttura ad albero è stata disegnata correttamente
- Scenario Principale:
 - Visualizza Albero (UC1.1): l'utente è in grado di visionare ed analizzare l'output grafico per valutare l'efficienza di una particolare *query* in maniera chiara;
 - Manipola Struttura (UC1.2): si offre la possibilità all'utente di manipolare la struttura per adattarla alle sue esigenze;
 - Visualizza informazioni aggiuntive (UC1.3): l'utente è in grado di visualizzare informazioni aggiuntive sulla singola operazione;
 - Visualizza tabella statistiche (UC1.4): l'utente è in grado di visualizzare una tabella dove sono presenti delle statistiche inerenti il numero di operazioni per ogni categoria di giudizio.
- Post-Condizione per successo: l'utente ha potuto svolgere una delle operazioni permesse.

3.2 Diagramma Use Case Utente - Manipolazione Struttura

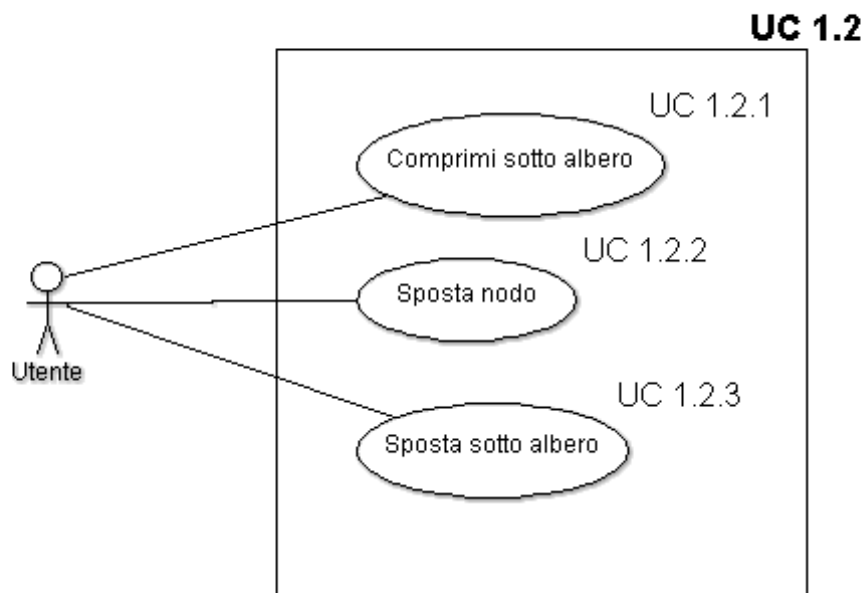


Figura 2: Diagramma use case 1.2

- Attori principali: Utente
- Precondizioni:
 - La struttura ad albero è stata disegnata correttamente
- Scenario Principale:
 - Comprimi sotto albero (UC1.2.1): l'utente può nascondere tutti i nodi figli di un certo nodo;
 - Sposta nodo (UC1.2.2): l'utente può spostare un singolo nodo;
 - Sposta sotto-albero (UC1.2.3): l'utente può spostare un nodo e tutti i suoi figli.
- Post-Condizione per successo: l'utente ha potuto svolgere una delle operazioni permesse.

4 Requisiti

I requisiti di questo progetto vanno suddivisi in:

- **Funzionali:** descrivono le interazioni tra l'interfaccia e l'ambiente esterno, indipendentemente dall'implementazione, ovvero si interessano delle funzionalità fornite dal sistema.
- **Non funzionali:** descrivono aspetti dell'interfaccia che non sono legati direttamente alle funzionalità, ma che coinvolgono i dettagli implementativi dello stesso e si suddividono in:
 - **Prestazionali:** descrivono requisiti del prodotto che aumentano l'efficienza globale del software fornito;
 - **Di qualità:** descrivono requisiti del prodotto che possono aumentare la qualità del software fornito;
 - **Di vincolo:** descrivono le condizioni da applicare all'intero prodotto e sono definiti prima che inizi l'effettiva raccolta dei requisiti.

Che sia funzionale o non funzionale, ogni requisito deve essere catalogato in una delle seguenti categorie in base alla rilevanza strategica:

- **Obbligatorio:** l'interfaccia deve soddisfare necessariamente tale vincolo;
- **Desiderabile:** l'interfaccia può non soddisfare necessariamente tale vincolo, ma la sua presenza nel risultato finale attribuisce valore aggiunto all'interfaccia;
- **Opzionale:** il soddisfacimento di tale vincolo è facoltativo e la sua presenza nel risultato finale non attribuisce valore aggiunto all'interfaccia.

Si è deciso di attribuire un codice univoco a ogni singolo requisito, al fine di identificarli meglio in fase di tracciamento. Ogni codice di requisito sarà composto da tre lettere maiuscole e un numero (es. "FOB01"). La prima lettera serve a identificare la tipologia del requisito (funzionale o non funzionale) e sono qui descritte:

- **F:** requisito funzionale;
- **P:** requisito prestazionale (non funzionale);
- **Q:** requisito di qualità (non funzionale);
- **V:** requisito di vincolo (non funzionale).

La seconda e la terza lettera indicano la priorità del requisito:

- **OB:** requisito obbligatorio;
- **DE:** requisito desiderabile;
- **OP:** requisito opzionale.

Il numero è stato introdotto al fine di organizzare i requisiti in maniera gerarchica. In caso di requisito principale si avrà un solo numero, mentre in caso di sotto-requisiti si avranno più numeri, assegnati in ordine crescente, separati da un punto. Il numero sarà sempre composto da due cifre, per cui, ad esempio, il numero 1 sarà indicato come 01. Di seguito sono riportati gli elenchi, per ogni requisito individuato, riportano le seguenti informazioni:

- codice di identificazione univoco;
- breve descrizione.

4.1 Requisiti funzionali

4.1.1 Obbligatori

- **FOB01**: l'utente è in grado di visualizzare il grafico che permetterà l'analisi dell'efficienza di una query e della struttura del database;
- **FOB02**: l'utente è in grado di visualizzare informazioni aggiuntive sulla singola operazione svolta dal database.

4.1.2 Desiderabili

- **FDE01**: l'utente è in grado di manipolare la struttura, in dettaglio:
 - **FDE01.1**: l'utente può spostare un nodo dell'albero;
 - **FDE01.2**: l'utente può spostare un sotto-albero;
 - **FDE01.3**: l'utente può comprimere un sotto-albero.

4.1.3 Opzionali

- **FOP01**: l'utente è in grado di visualizzare una tabella con le statistiche sul numero di operazioni catalogate per giudizio.

4.2 Requisiti prestazionali

4.2.1 Obbligatori

- **POB01**: l'interfaccia risulta efficiente e non richiede al browser un elevato carico.

4.3 Requisiti di qualità

4.3.1 Obbligatori

- **QOB01**: l'interfaccia è chiara e facile da utilizzare;
- **QOB02**: Ogni funzione è chiaramente identificata.

4.4 Requisiti di vincolo

4.4.1 Obbligatori

- **VOB01**: l'interfaccia sarà creata con le tecnologie *javascript*, *XHTML* e *HTML5* e l'utilizzo della libreria *Raphael*
- **VOB02**: l'interfaccia sarà utilizzabile dai browser:
 - Firefox versione 7.0 o superiore;
 - Internet Explorer versione 8.0 o superiore;
 - Opera versione 11.0 o superiore;
 - Chrome versione 14.0 o superiore;
 - Safari versione 5.0 o superiore.

5 Tabella tracciamento requisiti funzionali - Diagramma Use Case

Requisito funzionale software	Use Case
FOB01, FOB02, FDE01, FOP01	UC1
FDE01.1, FDE01.2, FDE01.3	UC1.2

Tabella 1: Tracciamento dei requisiti