

OpenOffice.org 2.0 e i Database

Introduzione all'uso dei Database con OpenOffice.org 2.0

Versione 0.99 – Dicembre 2005

Soft.Com



© 2005 **Filippo Cerulo** – Soft.Com Sas

www.softcombn.com - email: filippo.cerulo@softcombn.com

OpenOffice, MySql e PostgreSQL sono Marchi Registrati dai rispettivi proprietari.

Quest'opera è rilasciata sotto la licenza *Creative Commons*

“Attribuzione - Non commerciale - Non opere derivate 2.0 Italia.”



Per visionare una copia di questa licenza visita il sito web

<http://creativecommons.org/licenses/by-nc-nd/2.0/it/> o richiedila per posta a Creative

Commons, 559 Nathan Abbott Way, Stanford, California 94305, Usa.

Tu sei libero:

- di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire o recitare l'opera

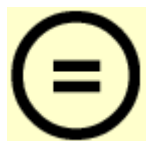
Alle seguenti condizioni:



Attribuzione. Devi riconoscere il contributo dell'autore originario.



Non commerciale. Non puoi usare quest'opera per scopi commerciali.



Non opere derivate. Non puoi alterare, trasformare o sviluppare quest'opera.

- In occasione di ogni atto di riutilizzo o distribuzione, devi chiarire agli altri i termini della licenza di quest'opera.
- Se ottieni il permesso dal titolare del diritto d'autore, è possibile rinunciare ad ognuna di queste condizioni.

Le tue utilizzazioni libere e gli altri diritti non sono in nessun modo limitati da quanto sopra

4. Motore di Database HSQL

Dalla versione 2.0, OOO ha deciso di dotarsi di un proprio motore di Db, chiamato **HSQL**. **HSQLDb** (<http://hsqldb.org/>) è un motore Db Open Source scritto in Java. Questo motore viene attivato automaticamente se, nella procedura di creazione di un Documento *Base*, non si sceglie una sorgente dati esterna. In questo caso tutte le tabelle (ed i dati in esse contenuti) non risiedono "all'esterno", ma *nel* Documento stesso.

Questo approccio limita molto l'utilizzo delle informazioni in modalità multiutente, perchè non è possibile aprire lo stesso Documento da più posti di lavoro senza avere seri problemi. Inoltre una delle regole principali della corretta gestione dei Database (quella di tenere strettamente separati *Dati ed Applicazioni*) viene ovviamente violata. HSQL usa Java, *quindi è necessario un JRE installato e funzionante*.

4.1 Tipi di Campo

4.1.1 Campi di Tipo Stringa

HSQL implementa quasi tutte le tipologie riconducibili a "stringa". Abbiamo **char** e **varchar**, oltre ad un "peculiare" **varchar_ignorecase** che dovrebbe trattare le stringhe senza badare al maiuscolo / minuscolo (utile negli ordinamenti ?). Associata alla descrizione **Memo**, troviamo anche una **longvarchar**, ma non chiedetemi quanti caratteri al massimo può contenere....

4.1.2 Campi di Tipo Numerico

I campi di tipo **numerico intero**, a seconda dell'intervallo di valori che possono contenere, si dividono in **TinyInt**, **SmallInt**, **Int**, **BigInt**. Il range dei valori disponibili non è riportato nella Guida di OOO, e non è facile da trovare neanche sul sito di HSQL, quindi non posso aiutarvi. Analogamente, i campi di tipo **numerico decimale** in HSQL possono essere definiti con un numero di decimali fisso (**Decimal**, **Numeric**) oppure variabile (**Float**, **Double**, **Real**). In particolare il tipo **Decimal** è indicato per la manipolazione di valori "valutari", quello che in altri motori di Db viene definito campo "**Money**" o "**Currency**".

TIP



Il valore realmente memorizzato nelle colonne definite **Float**, **Double** o **Real** dipende, stranamente, da come viene formattata la colonna stessa in modalità visualizzazione. OOO assume per default un numero di decimali pari a due, e, se non si specifica altrimenti, arrotonda a due cifre decimali qualsiasi valore inserito dall'utente. Questo, ovviamente, può creare grossi problemi.

4.1.3 Campi di Tipo Booleano

Definito in italiano come Si/No equivale al classico **boolean**. Nella rappresentazione interna, un valore uguale a zero dovrebbe essere considerato "falso", diverso da zero "vero".

4.1.4 Campi di Tipo Data/Ora

HSQL usa **Date** e **Time**, dal significato piuttosto intuitivo. Mi pare che il range di "Date" sia compreso tra 01-01-100 e 31-12-9999. Se si immette l'anno con sole due cifre, i valori da 01 a 29 vengono considerati successivi al 2000, da 30 a 99 precedenti (esempio 31/12/29 equivale a 31/12/2029 - 31/12/30 a 31/12/1930).

4.1.5 Campi di Tipo Binario

Abbiamo una scelta ampia: **binary**, **varbinary**, **longvarbinary**. Anche in questo caso non sono riuscito a trovare la lunghezza massima del campo.

4.1.6 Campi particolari: Intero ad incremento automatico

Possiamo usare un campo "**Integer**" con la caratteristica "**Valore Automatico**" settata.

4.1.7 Campi particolari: Timestamp

Il **Timestamp** di HSQL è definito come **Data/Ora** e non viene aggiornato automaticamente dal motore di Db.

4.2 Creazione e Modifica di Tabelle

Possiamo aggiungere o modificare Tabelle con le voci apposite del pannello *Attività* dopo aver scelto *Tabelle* nel pannello Database. La finestra di gestione della struttura delle Tabelle è concepita in modo assai semplice.

Il *Tipo di Campo* può essere scelto da una casella a discesa, e nella parte inferiore è possibile assegnare ulteriori proprietà. Per quanto sia presente un *Esempio di Formato*, non sempre poi viene rispettato nella visualizzazione. In fase di salvataggio, OOO chiede se desiderate aggiungere una chiave primaria: è importante rispondere **No** se avete già definito nella struttura un campo **Intero ad incremento automatico** (come **Mid**). Infatti in questo modo OOO utilizzerà la chiave definita da voi. La finestra di gestione degli indici si richiama con l'apposito pulsante della barra degli strumenti e non comporta difficoltà particolari. HSQL supporta anche le **viste** (**view**): la creazione di una vista è molto simile (se non uguale...) a quella di una *ricerca* (che vedremo in seguito). C'è da chiedersi, visto che l'archiviazione dei dati è interna, a cosa può servire mai una *vista*...

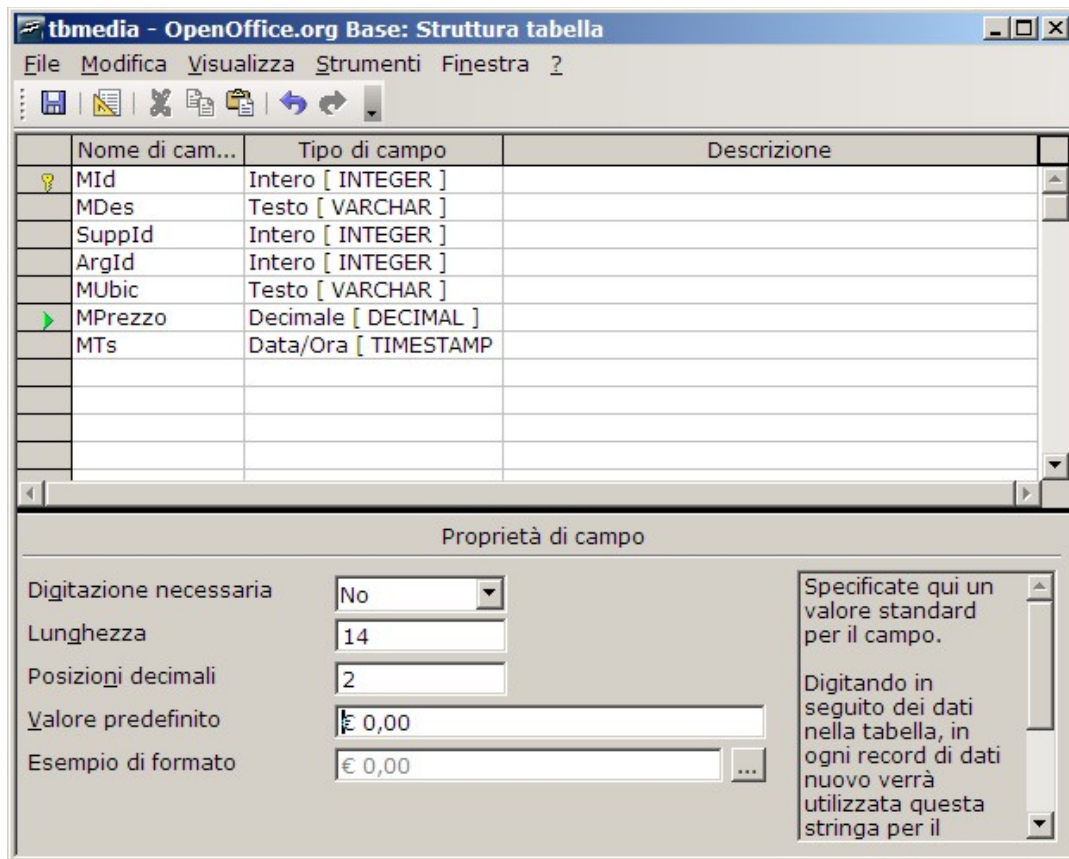


Figura 4.2.1: Modifica della struttura di una Tabella

Da questa finestra è possibile richiamare la funzionalità di manutenzione degli Indici con la voce **Strumenti -> Struttura Indice**.

TIP



Tenete presente che le variazioni agli archivi di HSQL non avvengono in maniera "fisica" ma "logica". Se ad esempio aggiungo 1000 righe ad una Tabella e quindi decido di cancellarle, la dimensione fisica del Db resta la stessa. Questo meccanismo ricorda un po' le Tabelle di Dbase che ogni tanto avevano bisogno del comando *pack*. Per ottenere la stessa cosa in HSQL è necessario scegliere *Strumenti -> SQL* dal Menu del modulo Base ed eseguire il comando **shutdown compact**.

4.3 Integrità referenziale

HSQL supporta l'utilizzo di chiavi esterne, nel modo classico dei Database Server. Per impostare le relazioni è necessario scegliere dal Menu la voce **Strumenti -> Relazioni**. La creazione di una relazione è semplice: basta aggiungere le Tabelle alla finestra, e trascinare il campo della Tabella su cui si desidera *applicare* una chiave esterna sul corrispondente campo

della Tabella che *contiene* la chiave. HSQL supporta i più comuni tipi di relazione: per una trattazione più approfondita, vedere il paragrafo relativo a MySQL. Ecco, ad esempio, le relazioni impostate per la Mediateca:

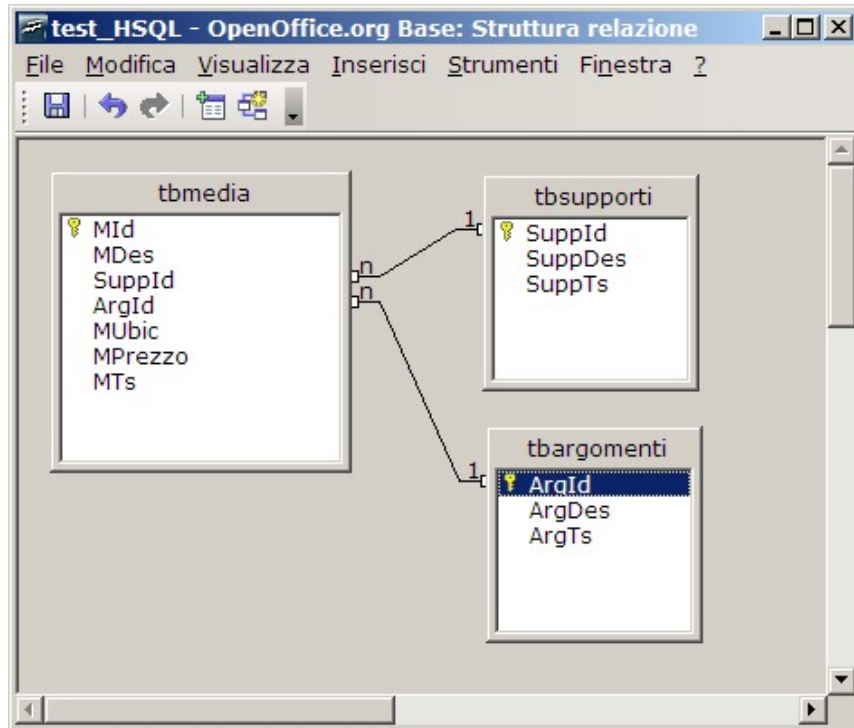


Figura 4.3.1: Relazioni in HSQL per la Mediateca

4.4 Importazione di Archivi esterni

Non credo che esista un sistema semplice per importare archivi esistenti in HSQL. Infatti non disponiamo di un Driver ODBC per HSQL, e non ho trovato nessuna procedura di importazione per altri formati (magari almeno in fixed text, che diamine...). Dopo un po' di prove, la soluzione più semplice è questa:

- creare una Database OOo collegato alla sorgente dati da importare (ad es. MySQL)
- creare un Database OOo HSQL
- aprire i due file
- trascinare le tabelle da trasferire dal primo Database al secondo, oppure usare copia/incolla

A questo punto si apre una finestra di auto composizione che aiuta nel processo. Attenzione: non sempre è possibile ottenere risultati accettabili; per esempio non c'è verso di importare Tabelle di Access, mentre MySQL funziona quasi sempre.

4.5 I Limiti di HSQL

La versione usata in OOO Base, l'abbiamo detto, è "embedded", cioè non si appoggia ad un motore esterno, e questo è un grosso problema. Vediamo perché.

- **Formato di File non manipolabile.** Nonostante il formato di OOO Base (.odb) sia uno standard OASIS, i dati vengono "inclusi" nel file XML in forma binaria e non in XML. Questo comporta che non siano accessibili dall'esterno in modo semplice.
- **Assenza della multiutenza.** HSQL in versione OOO non è multiutente, quindi non è permesso l'accesso contemporaneo ai dati da parte di più utenti.
- **Assenza di qualunque Driver.** I dati di un file .odb HSQL possono essere letti solo direttamente da OOO. Non esiste infatti alcun Driver per l'accesso esterno. Quindi niente ODBC, oppure JDBC oppure OLEDB.
- **Difficoltà di importazione da altre fonti.** Importare Database da altri formati non è semplice. Non esiste una procedura guidata, i formati di Data e Ora sono difficilmente manipolabili, ed anche un semplice file .csv crea problemi ad un utente non tecnico.
- **Dipendenza da JAVA.** HSQL non funziona se non è installata e funzionante una macchina virtuale JAVA, che, lo ricordo, non è un prodotto Open Source.
- **Prestazioni.** Il motore è lento, già con relativamente pochi record mostra la corda. Con Archivi di un certo "spessore" è inusabile.

Basta questo per definire, almeno per ora, HSQL un prodotto non adatto in ambito professionale.