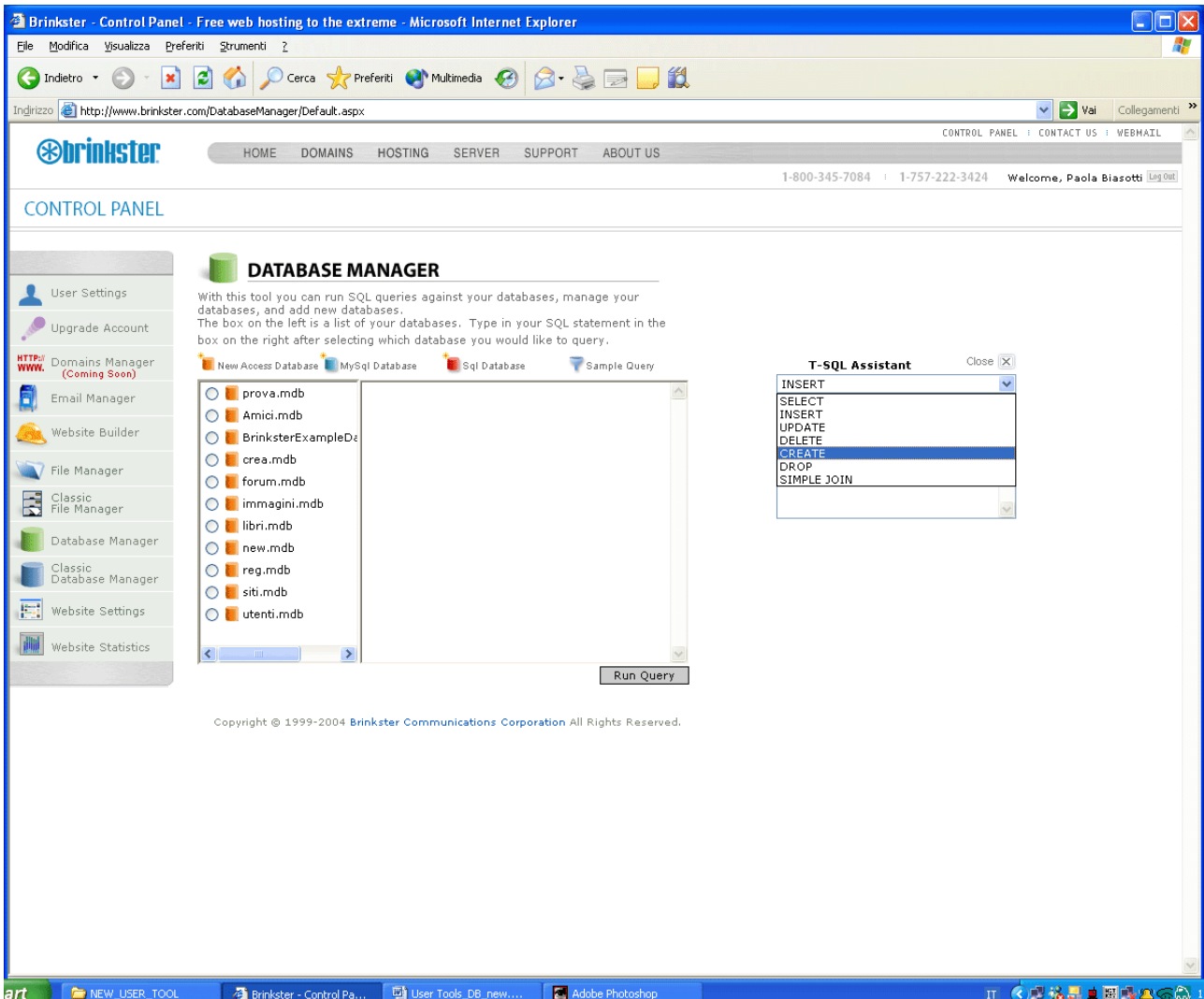


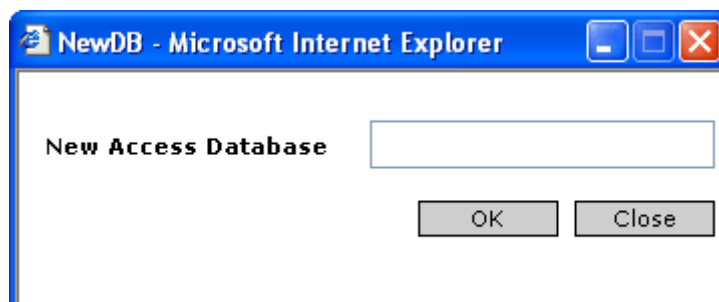
User Tools: DataBase Manager

Spazio di lavoro

Per usare “*T-SQL Assistant*” selezionare il link “*Simple Query*” e spostare a piacere la piccola finestra dove un menu a tendina mostra i diversi comandi SQL selezionabili, il pulsante *Preview* mostra la sintassi del comando selezionato e *Insert* inserirà nel box il comando personalizzato che si potrà eseguire premendo il pulsante *Run Query*, dopo aver selezionato un dato db.



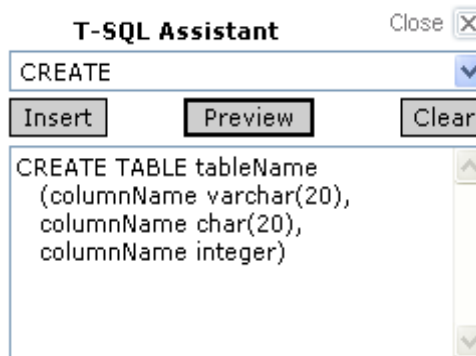
Per creare un **DataBase relazionale**: selezionare il link “*New Access Data base*” ed impostare il nome (estensione *.mdb* per default) nella finestra modale tipo “prompt” che si apre automaticamente; quindi confermare col bottone OK. Ogni DB sarà salvato nella cartella “database” di default ma solo la cartella “db” ha *permessi di lettura e scrittura*.



User Tools: DataBase Manager

Per **creare una tabella** in un DB:

- Selezionare il db selezionando un radio button (ad es. relativo a “prova.mdb”)
- Inserire il comando SQL desiderato nel box di destra:
 - direttamente con comando previsto **CREATE**¹
 - oppure con uso di *T_SQL Assistant*: selezionare il comando nel menu e premere *Preview*



➤ personalizzare la sintassi mostrata premendo *Preview* ed inserire con *Insert*

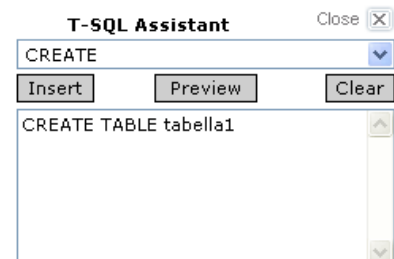
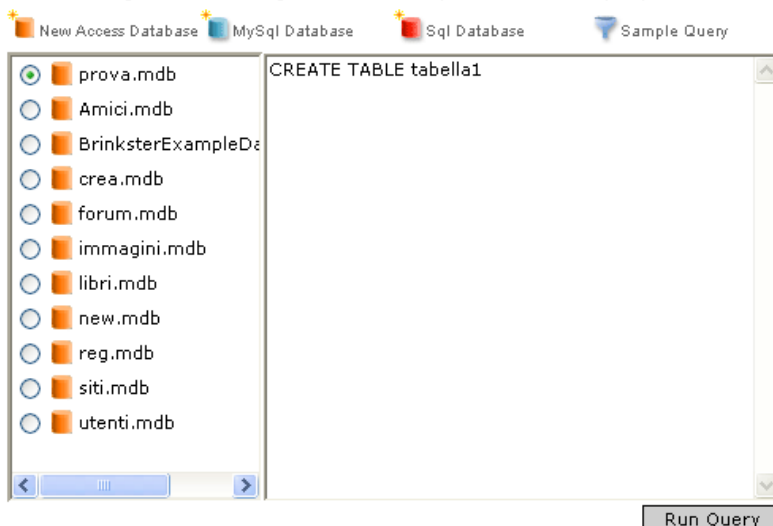
Esempio: CREATE TABLE *nomeTabella* (id INTEGER **PRIMARY KEY**, nome CHAR (20) , classe CHAR (20))
Con possibili **tipi**:

- CHAR (n) stringa di lunghezza massima “n”,
- VARCHAR (n) stringa corrispondente alla lunghezza effettiva del dato non superiore a “n”,
- SMALLINT intero (tra – 32.768 e +32.767),
- INTEGER numero intero (tra – 2.147.483.648 e + 2.147.483.647),
- DECIMAL (n, m) numero razionale con massimo “n” cifre di cui “m” decimali,
- FLOAT numero reale in notazione esponenziale.

In alcuni dialetti anche DATE (in formato standard MM/GG/AA o nel più recente formato ISO: AAAA/MM/GG)

DATABASE MANAGER

With this tool you can run SQL queries against your databases, manage your databases, and add new databases.
The box on the left is a list of your databases. Type in your SQL statement in the box on the right after selecting which database you would like to query.



Copyright © 1999-2004 Brinkster Communications Corporation All Rights Reserved.

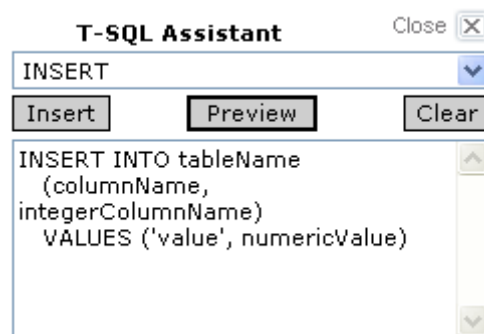
- eseguire il comando SQL premendo il bottone *Run Query*

¹ Anche CREATE VIEW per viste logiche, CREATE INDEX per indici ON tabella, CREATE UNIQUE INDEX se indice associato a eandidata o esterna e CREATE DATABASE.

User Tools: DataBase Manager

Per **inserire** informazioni in **una tabella** in un DB:

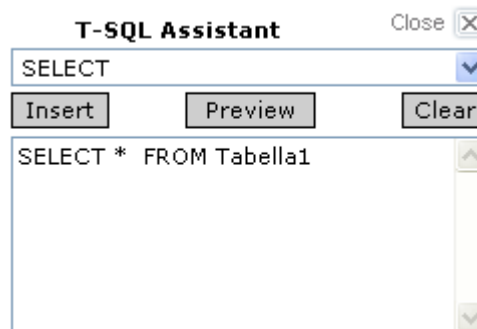
- Selezionare il db selezionando un radio button (ad es. relativo a “prova.mdb”)
- Inserire il comando SQL desiderato nel box di destra:
 - direttamente
 - oppure con uso di *T_SQL Assistano* seguendo i passi seguenti:
 - selezionare il comando nel menu,
 - premere *Preview*



- personalizzare la sintassi mostrata premendo *Preview* ed inserire con *Insert*
- eseguire il comando SQL premendo il bottone *Run Query*

User Tools: DataBase Manager

Query : estrazione di **informazioni** (dati e significato) con *criteri* e report



Uso di oggetto **RS** (Record Set) per creare al volo la tabella logica che contiene i risultati

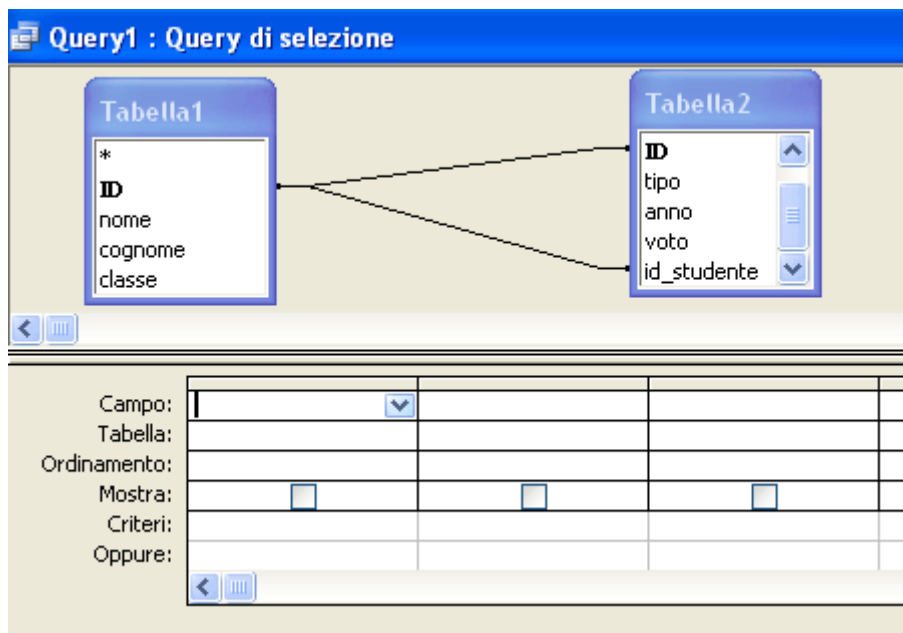
			Run Query
IDLink	descrizione	link preferiti	
1	motore di ricerca	www.google.com	
2	motore di ricerca	www.libero.it	
3	indice strutturato	www.yahoo.com	
4	sito account	www.brinkster.com	
5	libro on line	www.laterza.it	
6	libro on line	www.laterza.it	
7	libro on line	www.laterza.it	
8	libro on line	www.laterza.it	
9	libro on line	www.laterza.it	
10	libro on line	www.laterza.it	
11	libro on line	www.laterza.it	
12	libro on line	www.laterza.it	
13	libro on line	www.laterza.it	

User Tools: DataBase Manager

In ambiente *Access* automatico INNER JOIN² se si *aggiungono* più tabelle nel creare una query, con sintassi:

```
SELECT FROM Tabella1 INNER JOIN Tabella2 ON Tabella1.ID = Tabella2.ID;
```

oppure creando connessione tra ID della Tabella1 e id_studente della Tabella2:



```
SELECT  
FROM Tabella1 INNER JOIN Tabella2 ON (Tabella1.ID = Tabella2.id_studente) AND  
(Tabella1.ID = Tabella2.ID);
```

che, selezionando solo alcuni campi, diventa:

```
SELECT Tabella1.nome, Tabella1.cognome, Tabella2.voto  
FROM Tabella1 INNER JOIN Tabella2 ON (Tabella1.ID = Tabella2.id_studente) AND  
(Tabella1.ID = Tabella2.ID);
```

² La clausola **JOIN** indica che la tabella va “collegata” con altra tabella, mentre la parola chiave **ON** indica su quali colonne ciò deve avvenire. (join interno o **INNER JOIN**). Si parla di join esterno (**OUTER JOIN**) quando l’operatore restituisce comunque tutti i records (anche se non ci sono corrispondenti in alcuna tabella di join). Tale join può essere di tre tipi:

- join esterno sinistro (**LEFT OUTER JOIN**)
- join esterno destro (**RIGHT OUTER JOIN**)
- join esterno completo (**FULL OUTER JOIN**)

Se si vogliono i nominativi di tutti gli *studenti*, con o senza superamento *esame*:

```
SELECT Tabella1.nome, Tabella2.*  
FROM Tabella1 LEFT OUTER JOIN Tabella2 ON Tabella1.id = Tabella2.id_studente
```

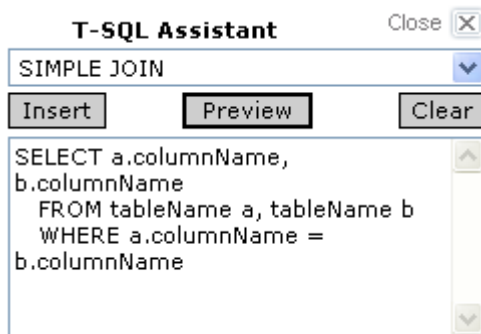
Se invece si vuole estendere l’esempio per ottenere l’elenco di tutti gli esami con o senza studenti:

```
SELECT Tabella1.nome, Tabella2.*  
FROM Tabella1 RIGHT OUTER JOIN Tabella2 ON Tabella1.id = Tabella2.id_studente
```

Se infine si vogliono ottenere i nominativi e l’elenco esami al completo:

```
SELECT Tabella1.nome, Tabella2.*  
FROM Tabella1 FULL OUTER JOIN Tabella2 ON Tabella1.id = Tabella2.id_studente
```

In Brinkster solo **Simple Join**:



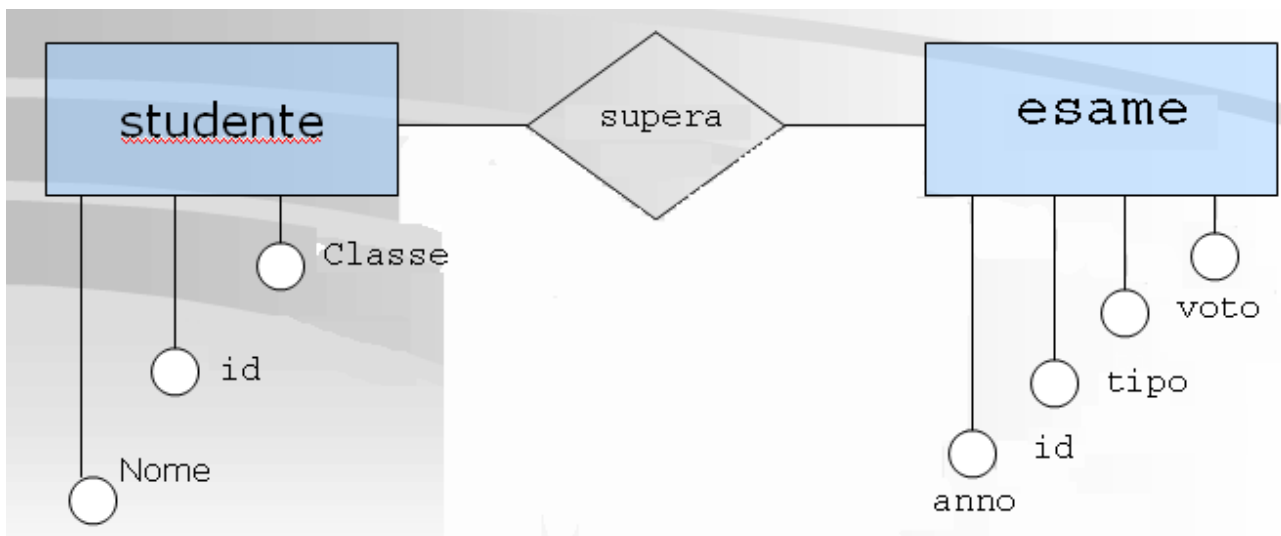
Un'istruzione di selezione si dice **JOIN** se nella clausola FROM dell'istruzione SELECT *intervengono due o più tabelle* e nella clausola WHERE *si confrontano colonne di tabelle diverse*.

La clausola WHERE e', al solito, opzionale.

Si usa la notazione puntata per il trattamento dei sottocampi di un record per evitare ambiguità. Per *default* ogni dato è interpretato in relazione all'ultima tabella presente nell'istruzione.

Ad esempio:

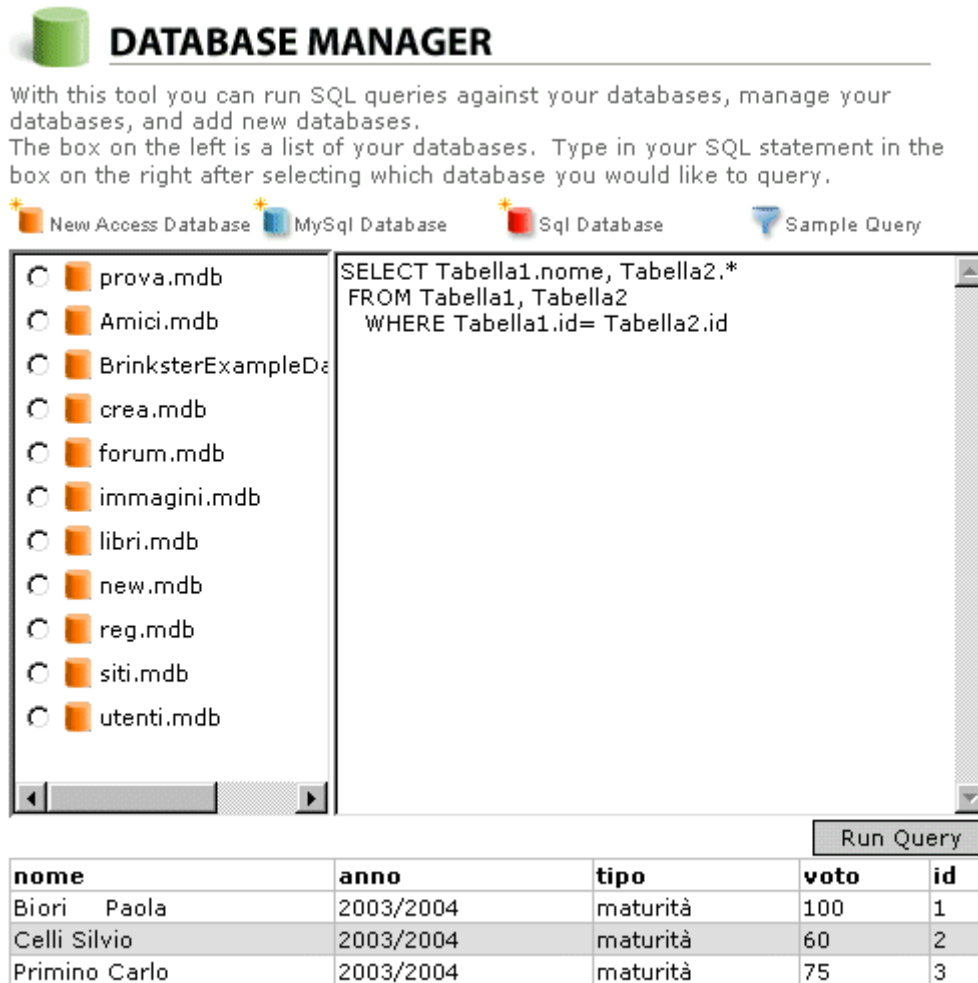
- create due tabelle "Tabella1" e "Tabella2" nel db memorizzato in "prova.mdb"
 - la prima per implementare l'entità *studente* con attributi: nome, classe ed id
 - la seconda per implementare l'entità *esame* con attributi: anno, tipo, voto ed id



- si vuole impostare una **query** che estragga il nome dello studente ed i dati relativi al superamento dell'esame, selezionando uguale identificativo
- si vuole impostare una **query** che estragga i dati dello studente ed i dati relativi al superamento dell'esame ad eccezione della replica dell'identificativo, selezionando uguale identificativo ed una classe particolare

1. Impostando la *query* (simple join) che estre il nome dello studente ed i dati relativi al superamento dell'esame, selezionando uguale identificativo

```
Select tabella1.nome, tabella2.*  
from tabella1, tabella2  
Where Tabella1.id = Tabella2.id
```



DATABASE MANAGER

With this tool you can run SQL queries against your databases, manage your databases, and add new databases.
The box on the left is a list of your databases. Type in your SQL statement in the box on the right after selecting which database you would like to query.

New Access Database MySQL Database Sql Database Sample Query

prova.mdb
 Amici.mdb
 BrinksterExampleDa
 crea.mdb
 forum.mdb
 immagini.mdb
 libri.mdb
 new.mdb
 reg.mdb
 siti.mdb
 utenti.mdb

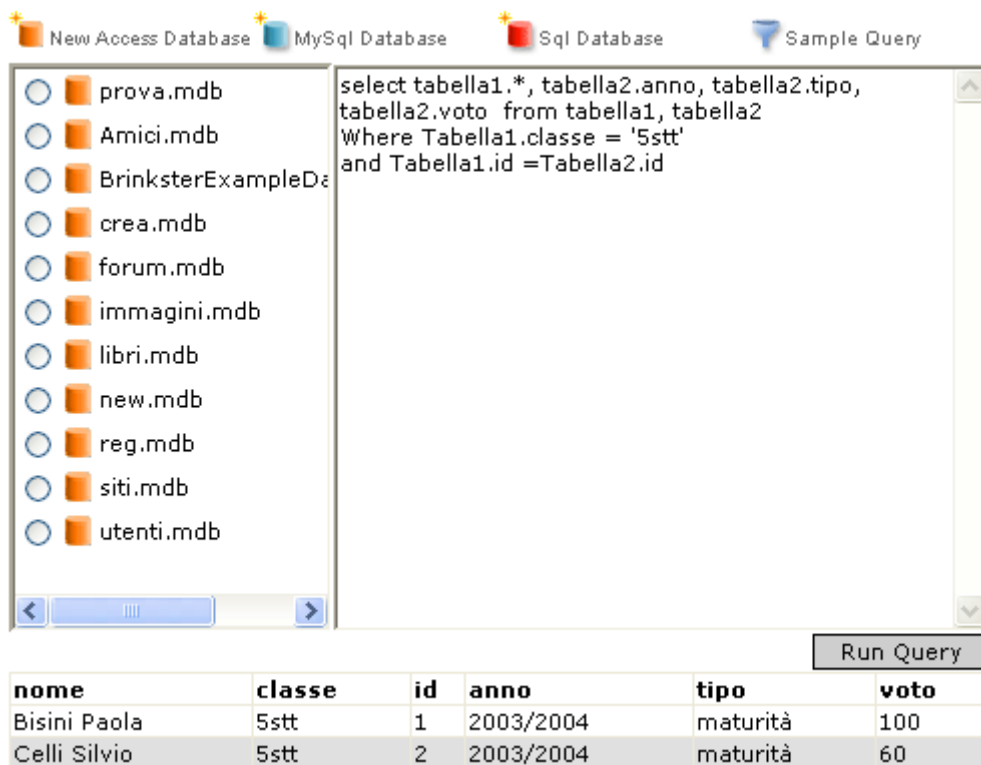
```
SELECT Tabella1.nome, Tabella2.*  
FROM Tabella1, Tabella2  
WHERE Tabella1.id= Tabella2.id
```

Run Query

nome	anno	tipo	voto	id
Biori Paola	2003/2004	maturità	100	1
Celli Silvio	2003/2004	maturità	60	2
Primino Carlo	2003/2004	maturità	75	3

2. Impostando la **query** (simple join) che estrae i dati dello studente ed i dati relativi al superamento dell'esame ad eccezione della replica dell'identificativo, selezionando uguale identificativo ed una classe particolare

Select tabella1.*, tabella2.anno, tabella2.tipo, tabella2.voto
from tabella1, tabella2
Where Tabella1.classe = '5stt ' **And** Tabella1.id = Tabella2.id



The screenshot shows a database management interface with a list of databases on the left and a query editor on the right. The query editor contains the following SQL query:

```
select tabella1.*, tabella2.anno, tabella2.tipo,
tabella2.voto from tabella1, tabella2
Where Tabella1.classe = '5stt'
and Tabella1.id =Tabella2.id
```

Below the query editor is a "Run Query" button. The results of the query are displayed in a table below:

nome	classe	id	anno	tipo	voto
Bisini Paola	5stt	1	2003/2004	maturità	100
Celli Silvio	5stt	2	2003/2004	maturità	60

Si veda l'[esempio](#) di query tipo simple join nell'elaborazione dati inoltrati da form e salvati in DB remoto.