

Esercitazione su SQL

Giuseppe De Giacomo

Domenico Lembo

OBIETTIVI DELL'ESERCITAZIONE

- Creare schema
- Creare e cancellare utenti
- Assegnare privilegi agli utenti
- Interrogare il catalogo
- Creare e cancellare tabelle
- Popolare uno schema
- Effettuare interrogazioni
- Dimostrazione di utilizzo del DBMS Oracle10g Lite e dell'ambiente MSQl

ORACLE 10g LITE

Informazioni su come

1. Scaricare Oracle 10g Lite,
2. Installarlo
3. Utilizzare I comandi principali di MSQL

disponibili alla pagina

<http://www.dis.uniroma1.it/~lembo/didattica/ProgettoBasiDati/Oracle10g-lite-intro/index.html>

Slide 3

1. CREARE UNA BASE DI DATI

- Per creare una nuova base di dati operate dal prompt dei comandi e posizionatevi in una cartella di lavoro (ad es., **c:\MusicistiDir**). Se la base di dati da creare è **Musicisti**, con DSN **MusicistiDSN** eseguite il comando

```
createdb MusicistiDSN Musicisti
```

- In questo modo viene creato il database **Musicisti** con DSN **MusicistiDSN** (in pratica viene creato il file **musicisti.odb** nella cartella da cui avete eseguito il comando, cioè **c:\MusicistiDir**)
- Registrate il DSN **MusicistiDSN** (vedi la guida sull'utilizzo di Oracle 10g)
- Connettetevi al database **Musicisti** tramite il comando

```
msql system/manager@jdbc:polite:MusicistiDSN
```
- A questo punto potete eseguire comandi SQL sul database **Musicisti**.

Slide 4

2. CREARE UNO SCHEMA

- Il comando che usiamo per creare uno schema è

```
CREATE SCHEMA <nome-schema>
```

- Per creare tabelle (o altre risorse) in uno schema si deve far precedere il nome della tabella dal nome dello schema al momento della definizione della tabella. Ad esempio, se lo schema è `SCH_Music`, il comando sarà

```
CREATE TABLE SCH_Music.Strumenti(  
    Nome          varchar(20) ,  
    Categoria     varchar(20)  
)
```

Slide 5

3. CREARE UTENTI

- Il comando che usiamo per creare utenti è

```
CREATE USER <nome-utente> IDENTIFIED BY <password>;
```

- Il comando che usiamo per garantire privilegi è

```
GRANT <privilegio> ON <risorsa> TO <nome-utente>;
```

Dove <privilegio> è `SELECT, DELETE, UPDATE, INSERT, ALL`

- Il comando che usiamo per assegnare ruoli è

```
GRANT <ruolo> TO <nome-utente>;
```

Dove <ruolo> è `ADMIN, DBA (o DDL), RESOURCE`

- Il comando che usiamo per revocare privilegi e\o ruoli è

```
REVOKE <ruolo>|<privilegio> [on <risorsa>] FROM <nome-utente>;
```

Slide 6

DATABASE 'Musicisti'

Sia dato il seguente schema relazionale.

- **musicisti(codice, nome, cognome, datanascita, sesso)**
rappresenta i dati anagrafici dei musicisti.
- **strumenti(nome, categoria)**
descrive gli strumenti. Categoria è il tipo di strumento;
- **abilita(musicista, strumento)**
descrive gli strumenti suonati dai musicisti.

Slide 7

CREARE TABELLE (a)

- Nota: per semplicità nel seguito non si farà riferimento ad alcuno schema. (quanto segue può comunque essere riferito ad uno schema, ad esempio **SCH_Music**, facendo precedere tutti i nomi di tabella usati nel seguito dal prefisso **SCH_Music.**)

- Fornire i comandi SQL per creare la tabella:

strumenti(nome, categoria)

- nome è una stringa di massimo 20 caratteri (chiave primaria)
- categoria è una stringa di 20 caratteri, che assume di default il valore 'NULL';

```
create table strumenti (  
    nome varchar(20) primary key,  
    categoria varchar(20) default null  
)
```

Slide 8

CREARE TABELLE (b)

Fornire i comandi SQL per creare la tabella:

- **musicisti(codice, nome, cognome, datanascita, sesso)**

- il codice è una stringa di 7 caratteri che inizia sempre con le lettere CM; (chiave primaria)
- l'attributo sesso assume i valori 'M' o 'F';
- non ci sono due musicisti con lo stesso nome, cognome e data di nascita.
- Nome cognome e data di nascita non possono assumere valori nulli

create table musicisti (

```
codice      character(7) check (codice like 'CM_____') primary key,  
nome        varchar(20) not null,  
cognome     varchar(20) not null,  
datanascita date not null,  
sesso       character check (sesso = 'M' or sesso = 'F'),  
unique (nome, cognome, datanascita)  
)
```

Slide 9

CREARE TABELLE (c)

Fornire i comandi SQL per creare la tabella:

- **abilita(musicista, strumento)**

Dove:

Ciascun musicista può suonare più strumenti e viceversa. L'attributo musicista è il codice di un musicista, l'attributo strumento è il nome di uno strumento.

create table abilita (

```
musicista   character(7) not null check (musicista like 'CM_____'),  
strumento  varchar(20) not null,  
primary key (musicista, strumento),  
foreign key (musicista) references musicisti,  
foreign key (strumento) references strumenti  
)
```

Slide 10

EFFETTUARE INTERROGAZIONI (1)

musicisti	codice	nome	cognome	datanascita	sex
strumenti	nome	categoria			
abilita	musicista	strumento			

Fornire le interrogazioni SQL per trovare

1. cognome e data di nascita dei musicisti di nome 'John'
2. nome di tutti gli strumenti della categoria 'Ottoni'
3. nome e cognome di tutti i musicisti che suonano il 'Sax Tenore'
4. numero di musicisti per ciascun sesso;
5. per ogni musicista che suoni strumenti della categoria 'Ottoni' fornire nome, cognome, data di nascita e strumenti della categoria 'Ottoni' suonati
6. per ogni musicista che suoni strumenti della categoria 'Ottoni' fornire nome, cognome, data di nascita e strumenti suonati

Slide 11

EFFETTUARE INTERROGAZIONI (2)

musicisti	codice	nome	cognome	datanascita	sex
strumenti	nome	categoria			
abilita	musicista	strumento			

Fornire le interrogazioni SQL per trovare

7. numero di musicisti per ciascuna categoria di strumenti;
8. strumenti suonati esclusivamente da musicisti di sesso maschile;
9. nome, cognome e data di nascita dei musicisti che suonano almeno 5 strumenti.
10. nome e cognome del musicista più vecchio (o dei musicisti più vecchi se ci sono musicisti nati nello stesso giorno) e strumenti da lui (da loro) suonati.
11. Categoria(e) di strumenti suonati dal massimo numero di musicisti.

Slide 12