# Corso di Basi di Dati 11. Esercitazioni in SQL: Altri esercizi

Guido Pezzini

A.A. 2016–2017

# Esercizi

Esempio: consideriamo le seguenti tabelle

ARTICOLI(Id, Nome, Colore)
FORNITORI(Id, Nome, Indirizzo)
CATALOGO(Articolo, Fornitore, Prezzo)

Catalogo.Articolo si riferisce ad Articoli.Id, Catalogo.Fornitore si riferisce a Fornitori.Id, ma può essere `NULL` se un articolo non ha un fornitore.

Esercizio: trovare i nomi dei pezzi (eliminando le ripetizioni) che hanno un fornitore.

# Esercizi

Soluzione:

```
select distinct A.Nome
from Articoli A, Catalogo C
where A.Id = C.Articolo
and C.Fornitore is not NULL;
```

# Esercizi

ARTICOLI(Id, Nome, Colore)
FORNITORI(Id, Nome, Indirizzo)
CATALOGO(Articolo, Fornitore, Prezzo)

Esercizio: trovare i nomi dei fornitori che hanno tutti gli articoli, sempre eliminando le ripetizioni.

Soluzione:

```
select distinct F.Nome
from Fornitori F
where not exists (select *
                  from Articoli A
                  where A.Id not in (select C.Articolo
                                     from Catalogo C
                                     where
                                     C.Fornitore = F.Id))
```

# Esercizi

Esempio:

GENITORI(Figlio, Genitore)

Esercizio: estrarre le coppie di fratelli (cioè coppie di persone con almeno un genitore in comune), eliminando le ripetizioni. Il risultato dovrà avere due colonne chiamate "Nome1" e "Nome2".

# Esercizi

Soluzione:

```
select distinct G1.Figlio AS Nome1,
G2.Figlio AS Nome2
from Genitori G1, Genitori G2
where G1.Genitore = G2.Genitore
and G1.Figlio <> G2.Figlio
```

# Esercizi

Esempio:

LIBRI(Id, Titolo, IdAutore, Anno)
AUTORI(Id, Nome)

Esercizio: estrarre i nomi degli autori che hanno scritto più di dieci libri.

# Esercizi

Soluzione 1:

```
select A.Nome
from Autori A
where 10 < (select count(*)
            from Libri L
            where A.Id = L.IdAutore)
```

# Esercizi

Soluzione 1:

```
select A.Nome
from Autori A
where 10 < (select count(*)
            from Libri L
            where A.Id = L.IdAutore)
```

Soluzione 2:

```
select distinct A.Nome
from Autori A, Libri L
where A.Id = L.IdAutore
group by A.Id
having count(*) > 10
```

# Altri esercizi

PRODOTTI(Id, Nome, Marca)
VENDITE(IdProdotto, Anno, Quantità)

In queste tabelle, VENDITE.IdProdotto si riferisce a PRODOTTI.Id.

Per ogni (nome di) prodotto che abbia almeno una vendita negli anni compresi fra il 2012 e il 2015 (estremi inclusi), calcolare la quantità totale venduta in tali anni.

Dare il risultato in una tabella con due colonne: "NomeProdotto" e "QuantitàTot".

# Altri esercizi

Soluzione:

```
select P.Nome as NomeProdotto,
       sum(V.Quantità) as QuantitàTot
from Vendite V, Prodotti P
where V.IdProdotto = P.Id
      and V.Anno >= 2012
      and V.Anno <= 2015
group by V.Prodotto
```

# Altri esercizi

Esercizio: Dare le definizioni SQL delle tabelle:

FARMACIE(Id, Indirizzo),
FARMACI(Id, Nome),
CATALOGO(IdFarmacia, IdFarmaco, Prezzo).

1. Le colonne tipo "Id..." e "Prezzo" devono essere numeri interi, le altre devono essere stringhe di lunghezza massima 30 caratteri.
2. "Prezzo" deve avere valore di default 0.
3. Specificare in FARMACIE e in FARMACI la colonna `Id` come chiave primaria,
4. Inserire un vincolo di `foreign key` su CATALOGHI.`IdFarmacia`, che si deve riferire a FARMACIE.`Id`, e un vincolo di `foreign key` su CATALOGHI.`IdFarmaco`, che si deve riferire a FARMACI.`Id`,
5. Specificare su entrambi i vincoli `no action` (cioè impedire l'operazione) in caso di modifiche o cancellazioni di dati.

# Altri esercizi

Soluzione (prime due tabelle):

```
create table Farmacie (
Id int primary key,
Indirizzo varchar(30) );

create table Farmaci (
Id int primary key,
Nome varchar(30) );
```

# Altri esercizi

Soluzione (terza tabella):

```
create table Catalogo (
IdFarmacia int,
IdFarmaco int,
Prezzo int default 0,
foreign key (IdFarmacia) references Farmacie(Id)
on update no action
on delete no action,
foreign key (IdFarmaco) references Farmaci(Id)
on update no action
on delete no action );
```