



# Basi di Dati



Esercizi sulle interrogazioni in SQL

# Videonoleggio DVD

---

- ▶ Si supponga di voler costruire un database per la gestione dei noleggi di DVD di un negozio. Tale database, in una versione del tutto semplicistica ed irrealistica, potrebbe essere rappresentato dalle tabelle clienti, dvd e noleggi:
  - ▶ tabella clienti, relativa ai clienti del videonoleggio e contenente informazioni quali il numero di tessera, il nome, il cognome (tutti obbligatori) ed il numero di telefono.
  - ▶ tabella dvd, relativa ai dvd in possesso del videonoleggio (si assuma per semplicità che le copie di ogni dvd siano illimitate, e che quindi più clienti possano prendere a noleggio copie diverse di uno stesso dvd contemporaneamente). Per ogni dvd la tabella tiene traccia del codice (obbligatorio), del titolo, del regista, del voto della critica (da 1 a 10) e della trama.
  - ▶ tabella noleggi, relativa ai noleggi di dvd in atto in quel momento. In particolare, la tabella tiene traccia del codice del dvd noleggiato, del numero di tessera del cliente che sta effettuando il noleggio e della data in cui il cliente ha iniziato il noleggio.

## Videonoleggio DVD

---

- ▶ Traducendo la realtà relativa al videonoleggio con i costrutti del modello relazionale si ottiene:
  - ▶ CLIENTI (TESSERA, NOME, COGNOME, TELEFONO)
  - ▶ DVD (CODICE, TITOLO, REGISTA, VOTO, TRAMA)
  - ▶ NOLEGGI (CODICE\_DVD, TESSERA, DATA\_NOLEGGIO)
    - ▶ FK: CODICE\_DVD REFERENCES DVD (CODICE)
    - ▶ FK: TESSERA REFERENCES CLIENTI (TESSERA)
- ▶ Gli attributi sottolineati rappresentano la chiave primaria di ciascuna tabella

## Videonoleggio DVD

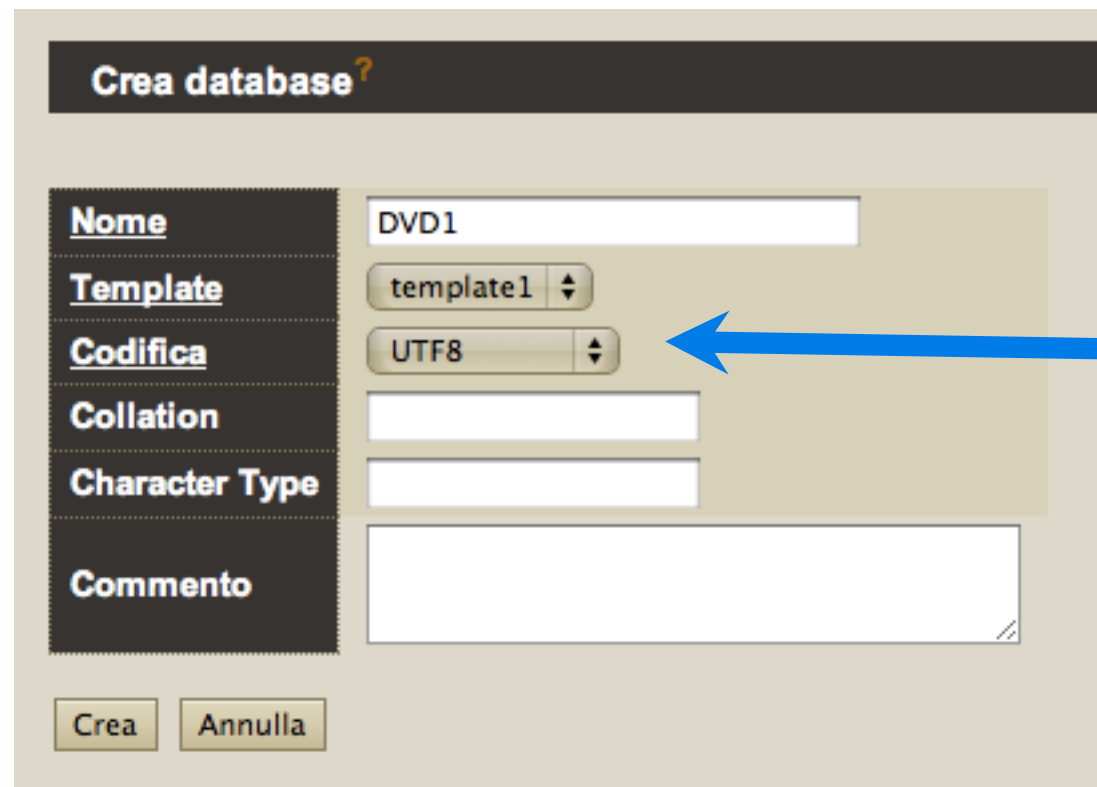
---

- ▶ Apriamo il browser, accediamo a **phpPgAdmin** tramite l'indirizzo:  
**<http://www.isgroup.unimo.it/postgres>**
- ▶ ed eseguiamo il login.
  
- ▶ Creiamo un nuovo database (**DVDxx**) e lo impostiamo come db corrente (**xx** rappresenta il numero del vostro gruppo)
  
- ▶ Possiamo ora definire mediante statement SQL gli schemi relativi al database "DVDxx" specificando gli opportuni vincoli.



## Creazione Nuovo Database

- ▶ **Attenzione:** affinché la creazione di un nuovo database vada a buon fine, è necessario specificare come codifica “UTF-8”!



The screenshot shows a dialog box titled "Crea database?". It has a dark header bar with the title. Below the header, there are several fields and buttons:

- Nome:** A text input field containing "DVD1".
- Template:** A dropdown menu showing "template1".
- Codifica:** A dropdown menu showing "UTF8". A blue arrow points to this field from the right.
- Collation:** An empty text input field.
- Character Type:** An empty text input field.
- Commento:** A large empty text area.

At the bottom of the dialog, there are two buttons: "Crea" and "Annulla".

# Videonoleggio DVD

---

```
CREATE TABLE CLIENTI (  
    TESSERA CHAR(5) NOT NULL,  
    NOME VARCHAR(60) NOT NULL,  
    COGNOME VARCHAR(60) NOT NULL,  
    TELEFONO VARCHAR(20),  
    PRIMARY KEY (TESSERA)  
);
```

```
CREATE TABLE DVD (  
    CODICE CHAR(5) NOT NULL,  
    TITOLO VARCHAR(60),  
    REGISTA VARCHAR(60),  
    VOTO INT CHECK (VOTO BETWEEN 1 AND 10),  
    TRAMA VARCHAR(200),  
    PRIMARY KEY (CODICE)  
);
```

# Videonoleggio DVD

---

```
CREATE TABLE NOLEGGI (  
    CODICE_DVD CHAR(5) NOT NULL REFERENCES DVD(CODICE),  
    TESSERA CHAR(5) NOT NULL REFERENCES CLIENTI(TESSERA),  
    DATA_NOLEGGIO DATE,  
    PRIMARY KEY (CODICE_DVD,TESSERA)  
);
```

## Videonoleggio DVD

---

- ▶ Popolare le tabelle create seguendo la traccia indicata di seguito:
  - ▶ Provare ad inserire almeno un record “non valido” (che va contro la definizione del vincolo espresso in fase di definizione della tabella) per i vincoli di ciascuna tabella
  - ▶ Inserire alcuni record “validi” e consistenti per ogni tabella (prevedendo anche la possibilità di valori nulli)
  - ▶ Aggiornare, aumentando di 3 unità il voto, il record della tabella DVD relativo al codice ‘12345’



# Videonoleggio DVD

---

INSERT INTO CLIENTI

VALUES ('00001','Carlo', 'Rossi', '+390596606633')

...

INSERT INTO DVD

VALUES ('00001','Braveheart','M.Gibson',8, 'William Wallace unites the 13th Century Scots in their battle to overthrow English rule')

...

UPDATE DVD

SET VOTO = VOTO + 3

WHERE CODICE = '12345'

# Videonoleggio DVD

---

- ▶ Formulare le seguenti interrogazioni:
  - ▶ “Elenco dei DVD (distinti) che hanno un voto maggiore di 5”
  - ▶ “Telefono dei clienti di nome ‘Carlo’”
  - ▶ “Codici DVD noleggiati da clienti con tessere (‘00001’, ‘00002’, ‘00003’, ‘00004’)”

# Videonoleggio DVD

---

```
SELECT CODICE  
FROM DVD  
WHERE VOTO>5
```

```
SELECT TELEFONO  
FROM CLIENTI  
WHERE NOME='Carlo'
```

```
SELECT CODICE_DVD  
FROM NOLEGGI  
WHERE TESSERA ='00001' OR TESSERA ='000002'  
OR TESSERA ='000003' OR TESSERA ='000004'
```

## In algebra relazionale...

---

- ▶ `PROJECT [CODICE] (SELECT [VOTO>5] DVD)`
- ▶ `PROJECT [TELEFONO]  
(SELECT [NOME=Carlo] CLIENTI)`
- ▶ `PROJECT [CODICE_DVD]  
(SELECT [TESSERA=00001  
OR TESSERA=00002  
OR TESSERA=00003  
OR TESSERA=00004]  
NOLEGGI )`



oppure...

---

- ▶ PROJECT [CODICE\_DVD]  
    (SELECT [TESSERA='00001'] NOLEGGI)  
        U  
PROJECT [CODICE\_DVD]  
    (SELECT [TESSERA='00002'] NOLEGGI)  
        U  
PROJECT [CODICE\_DVD]  
    (SELECT [TESSERA='00003'] NOLEGGI)  
        U  
PROJECT [CODICE\_DVD]  
    (SELECT [TESSERA='00004'] NOLEGGI)



# Database Esami

---

- ▶ Il Database **Esami** rappresenta la situazione di Studenti, Corsi, Docenti ed Esami:

S (Matr, SNome, Citta, ACorso)

C (CC, CNome, CD)

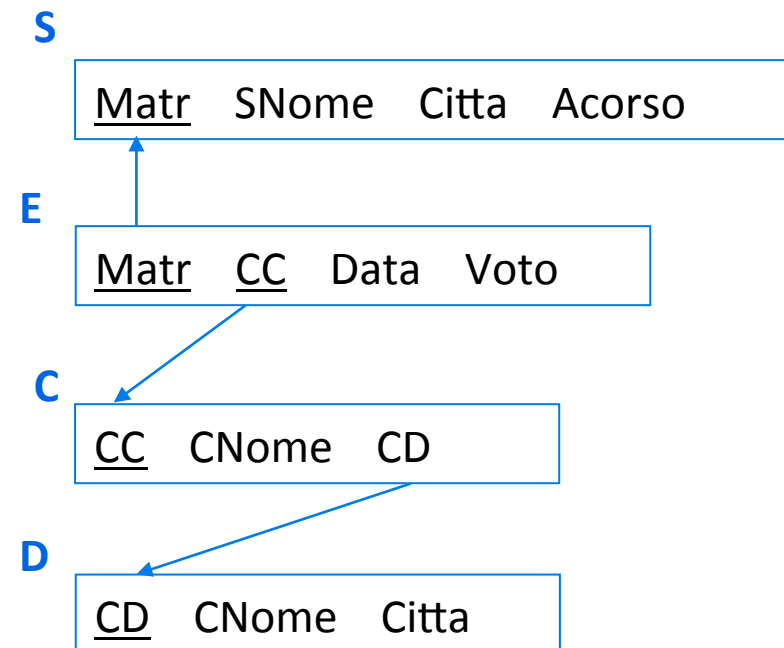
FK: CD REFERENCES D

D (CD, CNome, Citta)

E (Matr, CC, Data, Voto)

FK: Matr REFERENCES S

FK: CC REFERENCES C



# Database Esami – interrogazioni semplici

---

- ▶ Selezionare ...
  - ▶ Studenti del secondo anno di corso

```
SELECT *  
FROM S  
WHERE ACorso=2
```

# Database Esami – interrogazioni semplici

---

- ▶ Esami con voto compreso tra 24 e 28

```
SELECT *  
FROM E  
WHERE Voto >= 24  
AND Voto <= 28
```

```
SELECT *  
FROM E  
WHERE Voto BETWEEN 24 AND 28
```



# Database Esami – interrogazioni semplici

---

- ▶ Studenti il cui nome inizia con G

```
SELECT *  
FROM S  
WHERE SNome LIKE 'G%'
```

# Database Esami – interrogazioni semplici

---

- ▶ Studenti con l'attributo città non specificato

```
SELECT *  
FROM S  
WHERE Città IS NULL
```

## Database Esami – interrogazioni semplici

---

- ▶ Coppie di studenti residenti nella stessa città

```
SELECT S1.Matr,S2.Matr  
FROM S S1, S S2  
WHERE S1.Citta = S2.Citta  
AND S1.Matr < S2.Matr
```

## Database Esami – interrogazioni semplici

---

- ▶ Per ogni esame con voto superiore a 24 riportare il nome dello studente e il codice del docente del corso

```
SELECT S.SNome, C.CD  
FROM S,E,C  
WHERE S.Matr=E.Matr  
AND E.CC=C.CC  
AND Voto > 24
```

## Database Esami – interrogazioni semplici

---

- ▶ Matricole degli studenti che hanno sostenuto almeno uno degli esami sostenuti dallo studente di nome 'PAOLA'

```
SELECT E1.Matr  
FROM S, E E1, E E2  
WHERE E2.Matr = S.Matr  
AND E1.CC = E2.CC  
AND S.SNome= 'PAOLA'
```