

Cognome:

Nome:

Matricola:

Basi di Dati – Corso di Laurea in Informatica Umanistica

Appello del 21.04.2006

Parte 1 – modello relazionale, SQL (A)

Docente: Dino Pedreschi

Si consideri il seguente schema di base di dati universitaria:

TABLE Studenti
{*Matricola* integer **PRIMARY KEY**,
Cognome char(30) **NOT NULL**,
Nome char(20) **NOT NULL**,
Sesso char(1),
CorsoStudi char(20),
AnnoCorso integer,
Relatore char(4) **REFERENCES**
Professori(*Codice*) }

TABLE Esami
{*Studente* integer **REFERENCES**
Studenti(*Matricola*),
Corso char(3) **REFERENCES** *Corsi*(*Codice*),
Data date **NOT NULL**,
Voto integer **NOT NULL**,
Lode bool **NOT NULL**,
CHECK (*voto* >= 18 AND *voto* <= 30),
CHECK (*not lode* OR *voto* = 30),
PRIMARY KEY(*Studente*, *Corso*) }

TABLE Professori
{*Codice* char(4) **PRIMARY KEY**,
Cognome char(30) **NOT NULL**,
Nome char(20) **NOT NULL**,
Qualifica char(15),
Facoltà char(20) }

TABLE Corsi
{*Codice* char(3) **PRIMARY KEY**,
Titolo varchar(20) **NOT NULL**,
CorsoStudi char(20),
Docente char(4) **REFERENCES** *Professori*(*Codice*) }

1) Elencare i corsi che sono tenuti da un professore associato della facoltà di lettere e filosofia (algebra oppure SQL, punti 5)

2) Elencare ogni professore che è relatore di uno studente che ha sostenuto un esame con il professore medesimo con un voto maggiore di 28 (algebra oppure SQL, punti 5)

Cognome:

Nome:

Matricola:

3) Elencare i professori ordinari che tengono un corso alla laurea in Informatica Umanistica ed uno anche alla laurea in Lettere (algebra o SQL, punti 7)

4) Elencare gli studenti di Informatica Umanistica che hanno sostenuto l'esame di "italiano" e, successivamente, quello di "basi di dati"(algebra o SQL, punti 8)

5) Elencare gli studenti che hanno preso almeno un trenta e lode e nessun voto al di sotto del 28 (algebra o SQL, punti 7)

Cognome:

Nome:

Matricola:

Informatica per le Scienze Umane – Corso di Laurea in Lettere

Appello del 21.04.2006

Parte I – modello relazionale, SQL

Docente: Dino Pedreschi

Si consideri il seguente schema di base di dati universitaria:

TABLE Studenti
{*Matricola* integer **PRIMARY KEY**,
Cognome char(30) **NOT NULL**,
Nome char(20) **NOT NULL**,
Sesso char(1),
CorsoStudi char(20),
AnnoCorso integer,
Relatore char(4) **REFERENCES**
Professori(Codice) }

TABLE Esami
{*Studente* integer **REFERENCES**
Studenti(Matricola),
Corso char(3) **REFERENCES** *Corsi(Codice)*,
Data date **NOT NULL**,
Voto integer **NOT NULL**,
Lode bool **NOT NULL**,
CHECK (*voto* >= 18 AND *voto* <= 30),
CHECK (*not lode* OR *voto* = 30),
PRIMARY KEY(*Studente*, *Corso*) }

TABLE Professori
{*Codice* char(4) **PRIMARY KEY**,
Cognome char(30) **NOT NULL**,
Nome char(20) **NOT NULL**,
Qualifica char(15),
Facoltà char(20) }

TABLE Corsi
{*Codice* char(3) **PRIMARY KEY**,
Titolo varchar(20) **NOT NULL**,
CorsoStudi char(20),
Docente char(4) **REFERENCES** *Professori(Codice)* }

1) Elencare i corsi che sono tenuti da un professore associato della facoltà di lettere e filosofia (algebra oppure SQL, punti 5)

2) Elencare ogni professore che è relatore di uno studente che ha sostenuto un esame con il professore medesimo (algebra oppure SQL, punti 5)

Cognome:

Nome:

Matricola:

3) Elencare i professori che tengono un corso alla laurea in Informatica Umanistica ed uno anche alla laurea in Lettere (algebra o SQL, punti 7)

4) Elencare gli studenti che hanno sostenuto l'esame di "italiano" e, successivamente, quello di "informatica per le scienze umane"(algebra o SQL, punti 8)

5) Elencare gli studenti che hanno preso almeno un trenta e lode e nessun voto al di sotto del 28 (algebra o SQL, punti 7)