

SQL_MANIFESTI_STUDI

Iscrizioni universitarie

CORSI_LAUREA(codice, nome, descrizione)
INSEGNAMENTI(codice, nome, crediti, ssd)
MANIFESTI(laurea(fk), insegnamento(fk), fondamentale(boolean), anno_corso)
STUDENTI(matricola, nome, cognome, data_nascita)
ISCRIZIONI(studente(fk), anno_iscrizione, laurea(fk), data_iscrizione, anno_corso)

```
CREATE TABLE corsi_laurea (  
    codice integer NOT NULL PRIMARY KEY,  
    nome VARCHAR(200) NOT NULL,  
    descrizione VARCHAR(200) NOT NULL  
  
    -- alternative  
    -- PRIMARY KEY (codice)  
    -- creiamo i nostri vincoli  
    CONSTRAINT codice_non_nullo CHECK(codice IS NOT NULL)  
    CONSTRAINT codice_pk PRIMARY KEY (codice)  
);
```

La tabella puo' essere modificata con DML `INSERT INTO`:

```
INSERT INTO corsi_laurea (codice, nome, descrizione)  
VALUES (1, 'Informatica', 'Corso di Laurea triennale Informatica');  
  
INSERT INTO corsi_laurea (codice, nome, descrizione)  
VALUES (2, 'Matematica', 'Corso di Laurea triennale Matematica');
```

Possiamo aggiungere specifiche alla tabella, alterandola con `ALTER TABLE`:

```
ALTER TABLE corsi_laurea  
ADD UNIQUE(nome);
```

```
-- codice e nome degli insegnamenti disattivati in ordine  
-- alfabetico  
SELECT codice, nome  
FROM insegnamenti  
EXCEPT  
SELECT codice, nome  
FROM insegnamenti I, manifesti M  
WHERE I.codice = M.insegnamento  
ORDER BY nome ASC;
```

oppure

```
SELECT codice, nome  
FROM insegnamenti  
WHERE codice NOT IN (SELECT insegnamento  
                     FROM manifesti)  
  
ORDER BY nome;
```

```
-- codice e nome degli insegnamenti obbligatori  
SELECT DISTINCT I.codice, I.nome  
FROM insegnamenti I, manifesti M  
WHERE I.codice = M.insegnamento
```

```

        AND M.fondamentale
-- posizionale sugli attributi selezionati
-- ORDER BY I.nome
ORDER BY 2;

```

```

-- codice e nome degli insegnamenti solo a scelta
SELECT DISTINCT I.codice, I.nome
FROM insegnamenti I, manifesti M
WHERE I.codice = M.insegnamento
      AND NOT M.fondamentale
      AND NOT EXISTS (SELECT *
                      FROM manifesti M2
                      WHERE M2.insegnamento = I.codice
                        AND M2.fondamentale)
ORDER BY I.nome;

```

```

-- iscrizioni "proseguimento" nel 2022
-- creiamo una vista virtuale da usare sotto
CREATE OR REPLACE VIEW proseguimento (codice, cognome, nome) AS
SELECT S.codice, S.cognome, S.nome
FROM studenti S, iscrizione I22, iscrizione I21
WHERE I22.studente = I21.studente
      AND I22.anno_iscrizione = 2022
      AND I21.anno_iscrizione = 2021
      AND I22.laurea = I21.laurea
      AND I22.anno_corso = I21.anno_corso + 1
      AND S.matricola = I22.studente

SELECT *
FROM proseguimento
ORDER BY cognome, nome;

```

```

-- iscrizioni "naturali" nel 2022
SELECT *
FROM proseguimenti

UNION

select S.*
FROM studenti S, iscrizione I22
WHERE I22.anno_iscrizione = 2022
      AND I22.anno_corso = 1
      AND S.matricola = I22.studente
      AND NOT EXISTS (SELECT *
                      FROM iscrizione I_OLD
                      WHERE I_OLD.studente = S.matricola
                        AND I_OLD.anno_iscrizione < 2022)

ORDER BY cognome, nome;

```

```

-- estrarre i nomi dei corsi di laurea il cui manifesto comprende un insegnamento di
informatica (ssd uguale a INF/01 oppure ING-INF/05) come corso fondamentale
SELECT DISTINCT c.nome
FROM corsi_laurea c, insegnamenti i, manifesti m
WHERE c.codice = m.laurea
      AND m.insegnamento = i.codice
      AND i.fondamentale
      AND i.ssd IN ('INF/01', 'ING-INF/05')

```

```

-- per ogni corso di laurea, estrarre le date di nascita dello studente piu giovane e dello
studente pi'u vecchio iscritti a tale corso nell'anno 2012
SELECT C.nome,
       MIN(S.data_nascita) AS data_nascita_di_matusa,
       MAX(S.data_nascita) AS data_nascita_di_beniamino
FROM iscrizioni I, studenti S, corsi_laurea C
WHERE I.studente = S.matricola
      AND I.anno_iscrizione = 2012
      AND C.codice = I.laurea
GROUP BY C.nome

```

```

-- estrarre l'elenco degli insegnamenti che compaiono come fondamentali in almeno tre corsi di
laurea
SELECT DISTINCT I.codice, I.nome
FROM insegnamenti I, manifesti M1, manifesti M2, manifesti M3
WHERE I.codice = M1.insegnamento
      AND I.codice = M2.insegnamento
      AND I.codice = M3.insegnamento
      AND M1.fondamentale, M2.fondamentale, M3.fondamentale
      AND M1.laurea < M2.laurea AND M2.laurea < M3.laurea

```

oppure scritto meglio

```

SELECT I.codice, I.nome
FROM insegnamenti I, manifesti M
WHERE I.codice = M.insegnamenti
      AND M.fondamentale
GROUP BY I.codice, I.nome
HAVING COUNT(*) ≥ 3

```

```

-- per ogni insegnamento, calcolare il numero (presunto) di studenti iscritti nell'anno 2012
che frequentano l'insegnamento; uno studente `e frequentante se l'insegnamento compare nel
piano degli studi del corso al quale `e iscritto ed `e erogato nello stesso anno di corso dello
studente
SELECT INS.codice, INS.nome, COUNT(*) AS numero_frequentanti
FROM iscrizioni ISC, manifesti MAN, insegnamenti INS
WHERE ISC.laurea = MAN.laurea
      AND INS.codice = MAN.insegnamento
      AND ISC.anno_iscrizione = 2012
      AND MAN.anno_corso = ISC.anno_corso
GROUP BY INS.codice, INS.nome

```

```

-- per ogni studente iscritto nel 2021 al K-esimo anno di corso
-- di un corso di laurea con N anni di corso, dove K < N
-- iscriverlo nel 2022 allo stesso corso di laurea all'anno di corso K + 1
INSERT INTO iscrizioni (studente, anno_iscrizione, laurea, data_iscrizione, anno_corso)
  SELECT i.studente, 2022, i.laurea, current_date, i.anno_corso + 1
  FROM iscrizioni i, corsi_laurea c
  WHERE i.laurea = c.codice
        -- nello schema manca l'attributo anni_corso
        AND i.anno_corso < c.anni_corso
        AND i.anno_iscrizione = 2021

```