Basi di Dati

Interazione sulle Interrogazioni

a.a. 2019-2020 versione 2

Esercizio: Segreteria Studenti Università

Le tabelle di riferimento sono:

Corsi (NomeCorso, CognomeDocente, Crediti, Anno)

Studenti(Matricola, Cognome, Nome, DataNascita)

Esami(<u>Studente</u>, <u>Corso</u>, Datae, Voto, Lode)

In pratica non ammettiamo che uno studente può sostenere l'esame più volte, ovvero registriamo il voto solo di un esame.

Arricchire lo schema Segreteria Studenti Università al fine di consentire per l'esame di Basi di Dati la rappresentazione di prove intracorso (scritta, orale e laboratorio) e per l'esame di Lingua (writing-reading test, e conversation test), ciascuna con una propria valutazione.

In pratica si vogliono rappresentare tutte le prove intermedie in ogni esame, con il nome della prova e il voto conseguito.

Attenzione: non è noto a priori il numero e la tipologia delle prove intermedie di un esame.

Esercizio: Segreteria Studenti di Università

Le tuple della nuova relazione PROVE dovranno contenere: Il nome della prova; La votazione; Il riferimento all'esame a cui appartiene la prova rappresentato tramite una chiave esterna verso la relazione ESAMI.

Inoltre poiché la chiave primaria della relazione ESAMI è costituita dalla coppia (Studente, Corso) anche la chiave esterna di PROVE dovrà essere costituita da una coppia che chiameremo (StudenteEsame, CorsoEsame)

PROVE (NomeProva, Voto, StudenteEsame, CorsoEsame)

Esercizio: Segreteria Studenti di Università

NomeCorso varchar(50) primary key, CognomeDocente varchar(60), Crediti Real, Anno integer);

Esercizio: Segreteria Studenti di Università

Drop table if exists Prove CASCADE; Create table PROVE (NomeProva varchar(15), Voto integer, Vincoli di Integrità Referenziale

StudenteEsame varchar(16),

CorsoEsame varchar(50), primary key (NomeProva, StudenteEsame, CorsoEsame),

CONSTRAINT fk_Prove_StudentePCorsoP_Esami_StudenteECorsoE FOREIGN KEY (StudenteEsame, CorsoEsame) REFERENCES Esami (Studente, Corso)

ON UPDATE RESTRICT ON DELETE RESTRICT,

CONSTRAINT CHECK_NomeProva CHECK (((CorsoEsame='Basi di Dati') AND (NomeProva='Scritta' OR NomeProva='Orale' OR NomeProva='Laboratorio')) OR

((CorsoEsame LIKE 'Lingua%') AND NomeProva='Writing-Reading' OR NomeProva='Conversation')));

Esercizio: Segreteria Studenti di Università

/* Carico TABELLA Studenti */
insert into Studenti (Matricola, Cognome, Nome, DataNascita)
values('001', 'Primo', 'Mario', '1993-12-29');
insert into Studenti (Matricola, Cognome, Nome, DataNascita)
values('002', 'Secondo', 'Franco', '1993-07-23');
insert into Studenti (Matricola, Cognome, Nome, DataNascita)
values('003', 'Terzo', 'Michele', '1993-07-14');
insert into Studenti (Matricola, Cognome, Nome, DataNascita)
values('004', 'Quarto', 'Mauro', '1993-08-11');
/* Carico TABELLA Corsi */
insert into corsi (NomeCorso, CognomeDocente, Crediti, Anno)
values ('Fondamenti di Informatica', 'Saggese', 6, 1);
insert into corsi (NomeCorso, CognomeDocente, Crediti, Anno)
values ('Tecnologie per il Web', 'D'Aniello', 12, 2);
insert into corsi (NomeCorso, CognomeDocente, Crediti, Anno)
values ('Sistemi Informativi', 'Gaeta', 6, 3);
insert into corsi (NomeCorso, CognomeDocente, Crediti, Anno)
values ('Lingua Inglese', 'Jones', 12, 2);
insert into corsi (NomeCorso, CognomeDocente, Crediti, Anno)
values ('Lingua Tedesca', 'Muller', 12, 3);
insert into corsi (NomeCorso, CognomeDocente, Crediti, Anno)
values ('Basi di Dati', 'D''Acierno', 6, 3);
/* Carico TABELLA Esami */
insert into Esami (studente, corso, datae, voto, lode)
values ('001', 'Tecnologie per il Web', '2014-06-24', 30, false);
insert into Esami (studente, corso, datae, voto, lode)
values ('001', 'Iigua Inglese', '2015-01-13', 30, true);
insert into Esami (studente, corso, datae, voto, lode)
values ('001', 'Iigua Inglese', '2015-02-18', 29, false);
insert into Esami (studente, corso, datae, voto, lode)
values ('001', 'Iigua Inglese', '2015-02-18', 29, false);
insert into Esami (studente, corso, datae, voto, lode)
values ('001', 'Iigua Inglese', '2015-01-13', 30, true);
insert into Esami (studente, corso, datae, voto, lode)
values ('001', 'Iigua Inglese', '2015-01-13', 30, true);

Esercizio: Segreteria Studenti di Università

/* Carico TABELLA Prove */

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Scritta', 29, '001', 'Basi di Dati');

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Orale', 30, '001', 'Basi di Dati');

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Laboratorio', 30, '001', 'Basi di Dati');

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Conversation', 27, '001', 'Lingua Inglese');

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Writing-Reading', 30, '001', 'Lingua Inglese');

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Scritta', 19, '003', 'Basi di Dati');

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Orale', 15, '003', 'Basi di Dati');

insert into Prove (NomeProva, Voto, StudenteEsame, CorsoEsame) values('Laboratorio', 23, '003', 'Basi di Dati');

Select* from Prove;

	CONTENUTO DI	LLA TABE	LLA Prove DOPO gli INSE	RIMENTI	
Dat	a Output Explain Messages	s Notificat	ions		
4	nomeprova [PK] character varying (15)	voto integer	studenteesame [PK] character varying (16)	corsoesame [PK] character varying (50)	
1	Scritta	29	001	Basi di Dati	
2	Orale	30	001	Basi di Dati	
3	Laboratorio	30 001		Basi di Dati	
4	Conversation	27	001	Lingua Inglese	
5	Writing-Reading	30	001	Lingua Inglese	
6	Scritta	19	003	Basi di Dati	
7	Orale	15	003	Basi di Dati	
8	Laboratorio	23	003	Basi di Dati	

Determinare i voti degli Esami superati dallo Studente Primo Mario

Esercizio: Studenti

Determinare i voti degli Esami superati dallo Studente Mario Primo

SELECT *

FROM Studenti S JOIN ESAMI E ON (S.Matricola=E.Studente)
WHERE((S.Nome='Mario') AND (S.Cognome='Primo'));

SELECT *

FROM Studenti S LEFT JOIN ESAMI E ON (S.Matricola=E.Studente) WHERE ((S.Nome='Mario') AND (S.Cognome='Primo'));

SELECT*

FROM Studenti S RIGHT JOIN ESAMI E ON (S.Matricola=E.Studente) WHERE ((S.Nome='Mario') AND (S.Cognome='Primo'));

SELECT *

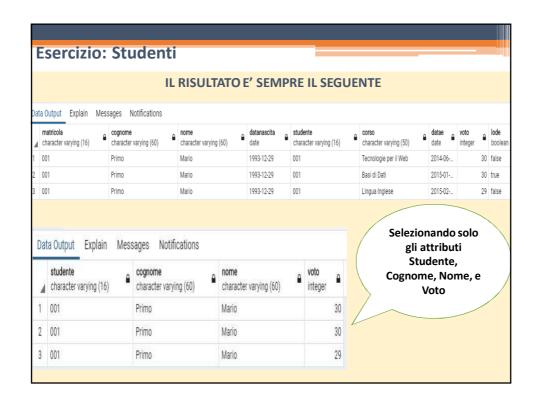
FROM Studenti S NATURAL JOIN ESAMI E

WHERE (S.Matricola=E.Studente) AND ((S.Nome='Mario') AND (S.Cognome='Primo'));

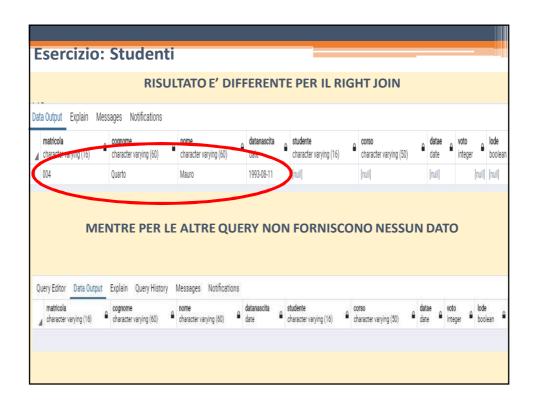
SELECT ³

FROM Studenti S, Esami E

WHERE (S.Matricola=E.Studente) AND ((S.Nome='Mario') AND (S.Cognome='Primo'));



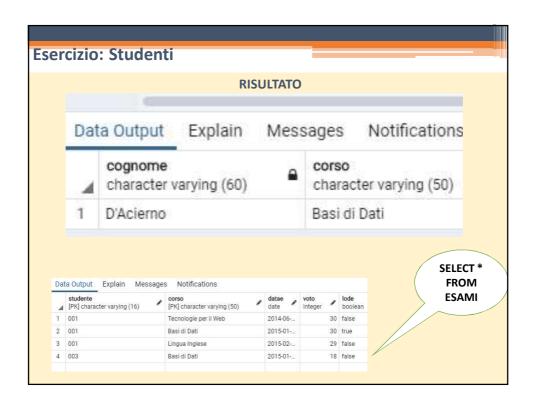
Esercizio: Studenti Osserviamo cosa accade facendo le stesse QUERY determinando i voti degli Esami superati dallo Studente Mauro Quarto **SELECT * FROM Studenti S JOIN ESAMI E ON** (S.Matricola=E.Studente) WHERE((S.Nome='Mauro') AND (S.Cognome='Quarto')); **SELECT * FROM Studenti S LEFT JOIN ESAMI E ON** (S.Matricola=E.Studente) WHERE ((S.Nome='Mauro') AND (S.Cognome='Quarto')); **SELECT * FROM Studenti S RIGHT JOIN ESAMI E ON** (S.Matricola=E.Studente) WHERE ((S.Nome='Mauro') AND (S.Cognome='Quarto')); **SELECT * FROM Studenti S NATURAL JOIN ESAMI E** WHERE (S.Matricola=E.Studente) AND ((S.Nome='Mauro') AND (S.Cognome='Quarto')); **SELECT * FROM Studenti S, Esami E** WHERE (S.Matricola=E.Studente) AND ((S.Nome='Mauro') AND (S.Cognome='Quarto'));



Determinare il Cognome dei Docenti che hanno assegnato Voto Inferiori al 20

Determinare il Cognome dei Docenti che hanno assegnato Voto Inferiori al 20 SELECT C.CognomeDocente AS COGNOME, C.NomeCorso AS CORSO FROM Corsi C JOIN ESAMI E ON (C.NomeCorso=E.Corso) WHERE(E.Voto<20);

SELECT C.CognomeDocente AS COGNOME, C.NomeCorso AS CORSO FROM Corsi C, ESAMI E WHERE (C.NomeCorso=E.Corso AND E.VOTO<20);



Determinare il Cognome dei Docenti dei Corsi per i quali Mario Primo ha sostenuto un esame

Esercizio: Studenti Determinare il Cognome dei Docenti dei Corsi dei quali Mario Primo ha sostenuto un esame **SELECT C.CognomeDocente** FROM CORSI C JOIN ESAMI E ON (C.NomeCorso=E.Corso) JOIN STUDENTI S ON (S.Matricola=E.Studente) WHERE(S.Nome='Mario' AND S.Cognome='Primo'); **SELECT C.CognomeDocente** character varying (60) FROM STUDENTI S JOIN ESAMI E ON 1 D'Aniello 2 D'Acierno (S.Matricola=E.Studente) JOIN CORSI C ON (C.NomeCorso=E.Corso) WHERE(S.Nome='Mario' AND S.Cognome='Primo');

Riportare per ogni studente presente in tabella Studenti gli eventuali esami sostenuti, in assenza di esami la Query ritorna i campi di interesse relativi alla tabella ESAMI con il valore NULL

Esercizio: Studenti

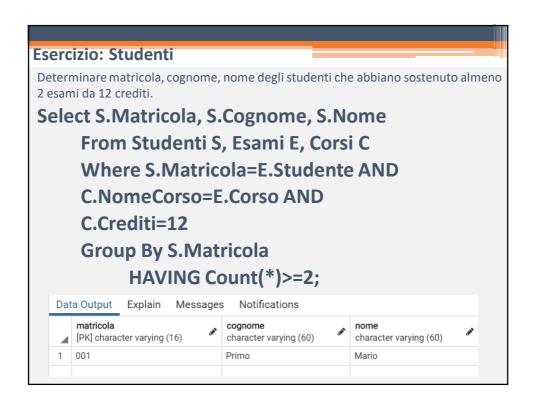
Riportare per ogni studente presente in tabella Studenti gli eventuali esami sostenuti, in assenza di esami la Query ritorna i campi di interesse relativi alla tabella ESAMI con il valore NULL

SELECT *

FROM STUDENTI S LEFT JOIN ESAMI E ON (S.Matricola=E.Studente);

Da	Data Output Explain Messages Notifications											
4	matricola character varying (16)	cognome character varying (60)	nome character varying (60)	datanascita date	studente character varying (16)	corso character varying (50)	datae date	voto integer	lode boolean			
1	001	Primo	Mario	1993-12-29	001	Tecnologie per il Web	2014-06	30	false			
2	001	Primo	Mario	1993-12-29	001	Basi di Dati	2015-01	30	true			
3	001	Primo	Mario	1993-12-29	001	Lingua Inglese	2015-02	29	false			
4	003	Terzo	Michele	1993-07-14	003	Basi di Dati	2015-01	18	false			
5	002	Secondo	Franco	1993-07-23	[null]	[null]	[null]	[null]	[null]			
6	004	Quarto	Mauro	1993-08-11	[null]	[null]	[null]	[null]	[null]			

Determinare matricola, cognome, nome degli studenti che abbiano sostenuto almeno 2 esami da 12 crediti.



Determinare il numero di studenti la cui media è superiore a 25.

Suggerimento

Iniziamo ad individuare gli studenti che hanno la media superiore a 25. Questa Query diverrà la Sottoquery della Query che conta gli Studenti

Select E.Studente

From Esami E

Group By E.Studente

Having AVG(E.Voto)>=25;

Esercizio: Studenti

Determinare il numero di studenti la cui media è superiore a 25.

Anche in questo caso la precedente Query diviene una sottoquery di una Query che conta.

```
Select Count (*)
From
(
Select E.Studente
From Esami E
Group By E.Studente
Having AVG(E.Voto)>=25
)
AS MATRICOLASTUDENTI;
```