

Esercizio fine modulo 3- Database e linguaggio SQL

Guido Marconi

Caso di studio

Si desidera analizzare la struttura di una università, creando una base di dati che includerà diversa informazione relativa alla organizzazione dello staff, studenti e sedi.

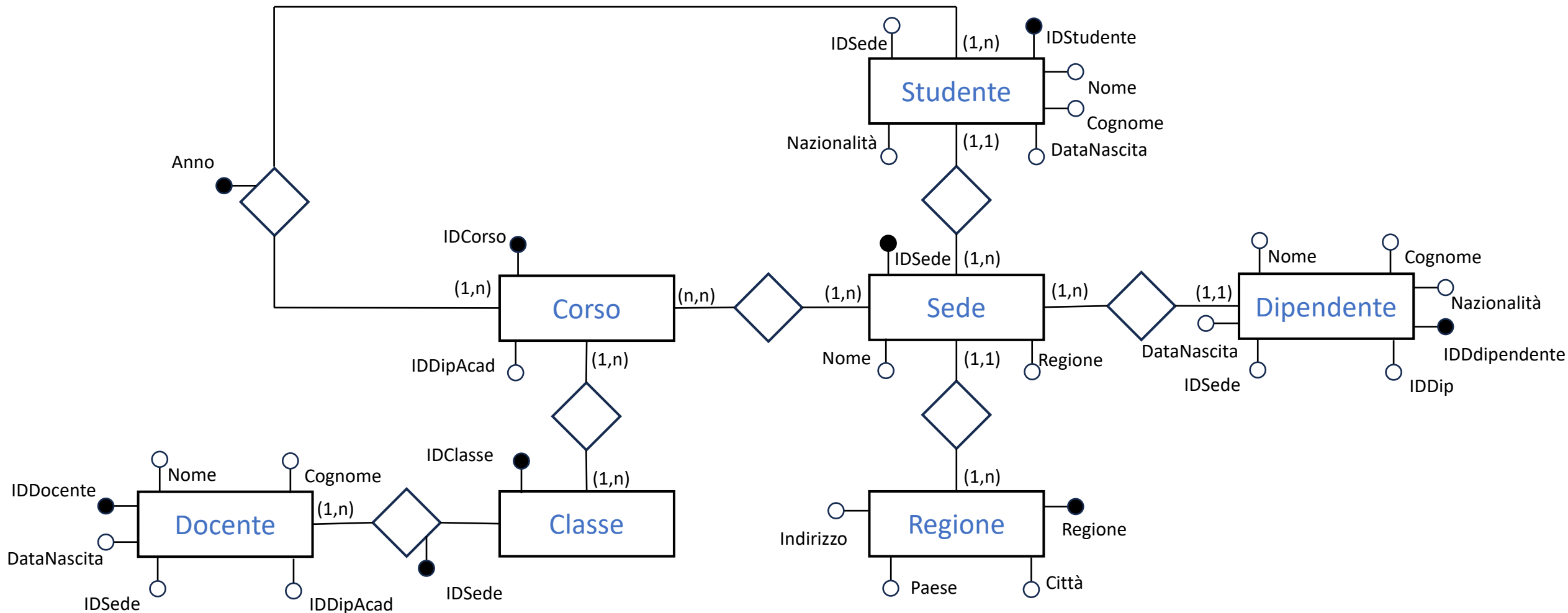
Progettazione concettuale

L'università si compone di una sede principale a Bologna e tre altre sedi in diversi paesi al mondo. I corsi possibili da frequentare sono gli stessi per tutte le sedi e gli studenti possono frequentare solo un corso nello stesso anno accademico. Il programma accademico è lo stesso per tutte le sedi. Ogni corso si compone di un gruppo di classi, e per ciascuna classe viene assegnato un docente. I docenti possono essere responsabili di più di una classe. Ogni sede ha il suo staff che si compone di diversi dipartimenti secondo la loro funzione.

Definizione delle entità e attributi:

- Studente: **IDstudente (PK)**, Nome, Cognome, DataNascita, Nazionalità, **IDSede (FK)**
- Docente: **IDDocente (PK)**, Nome, Cognome, DataNascita, **IDSede (FK)**, DipartimentoAcad
- Dipendente: **IDDipendente (PK)**, Nome, Cognome, DataNascita, **IDSede**, Dipartimento
- Sede: **IDSede (PK)**, Nome, **Regione (FK)**.
- Regione: **Nome (PK)**, Città, Paese, Indirizzo.
- Corso: **IDCorso (PK)**, DipartimentoAcad,
- Classe: **[IDClasse, IDSede (FK)] (PK)**, Docente, Crediti
- Periodo: **[Anno, IDStudente (FK)] (PK)**, IDCorso (FK)

Progettazione concettuale



Progettazione Logica

Studente					
IdStudente	Nome	Cognome	DataNascita	Nazionalità	IDSede
S-1	Stephen	Jiang	17/05/1951	Italiana	1
S-2	Brian	Welcker	04/01/1977	Argentina	3
S-3	Michael	Blythe	25/07/1968	Spagnola	2

Periodo			
Anno	IdStudente	Idcorso	IsCurrent
2011	S-1	SA-1	True
2023	S-3	I-11	False
2018	S-25	PS-3	True

Regione			
Regione	Città	Paese	Indirizzo
Europa	Bologna	Italia	Via Belvemoro 23
America	Washington DC	Stati Uniti	555 Pennsylvania Avenue
Asia	Tokio	Giapone	7 Chome-3-1 Hongo

Sede		
IdSede	Nome	Regione
1	UniBologna	Europa
2	UniWashington	America
3	UniTokio	Asia

Corso		
IdCorso	Nome	DipAcad
SA-1	Direzione aziendale	Scienze Aziendali
I-11	Ingegneria meccanica	Ingegneria
PS-3	Neuroscienze	Psicologia

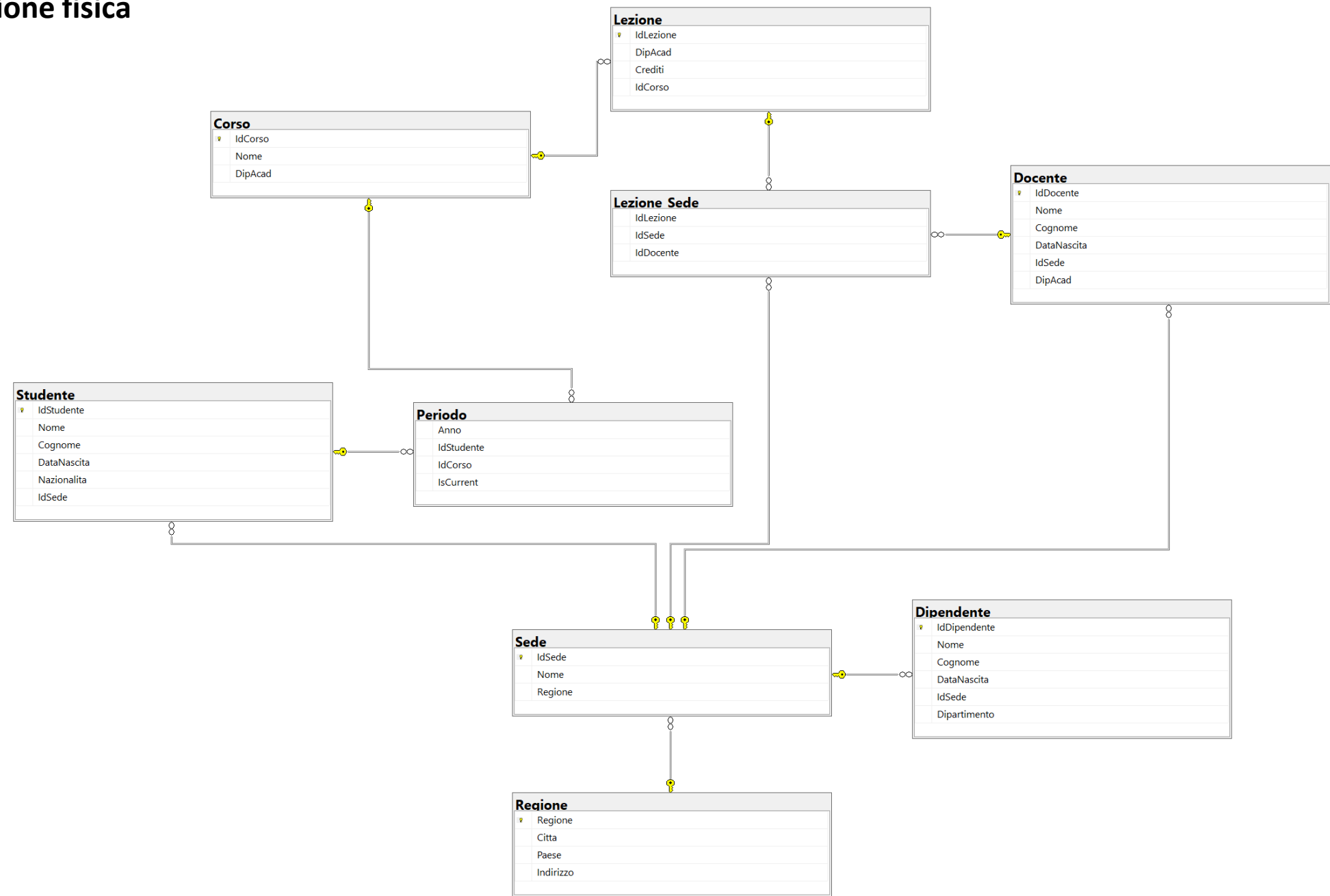
Dipendente					
IdDipend	Nome	Cognome	DataNascita	IdSede	Dipartimento
DI-1	Tsvi	Reiter	18/08/1973	1	Amministrazione
DI-2	Pamela	Ansman-Wolfe	06/07/1974	2	Alumni
DI-3	Shu	Ito	08/10/1943	2	Didattica

Lezione			
IdLezione	Nome	Crediti	IdCorso
SA-DA-1	Diritto Societario	10	SA-1
I-IM-12	Fisica I	8	I-11
PS-N-3	Psicologia Clinica	14	PS-3

Lezione_Sede		
IdLezione	IdSede	IdDocente
SA-DA-1	1	DO-1
I-IM-12	2	DO-2
PS-N-3	3	DO-3

Docente					
IdDocente	Nome	Cognome	DataNascita	IdSede	DipAcad
DO-1	Syed	Abbas	11/08/1974	2	Scienze Aziendali
DO-2	Rachel	Valdez	06/02/1975	1	Ingenieria
DO-3	Lynn	Tsoflias	16/10/1970	3	Psicologia

Progettazione fisica



--Creazione di tabelle

```
create table Regione(  
  Regione varchar (20),  
  Citta varchar (15),  
  Paese varchar (15),  
  Indirizzo varchar (25),  
  constraint PK_Regione_Regione primary key (Regione))
```

```
create table Sede(  
  IdSede int,  
  Nome varchar(15),  
  Regione varchar(20),  
  constraint PK_Sede_IDSede primary key (IdSede),  
  constraint FK_Sede_Regione_Regione foreign key (Regione) references Regione (Regione))
```

```
create table Dipendente(  
  IdDipendente varchar(10),  
  Nome varchar(15),  
  Cognome varchar (15),  
  DataNascita date,  
  IdSede int,  
  Dipartimento varchar(15),  
  constraint PK_Dipendente_IdDipendente primary key (IdDipendente),  
  constraint FK_Dipendente_Sede_IdSede foreign key (IdSede) references Sede (IdSede))
```

```
create table Studente(  
  IdStudente varchar(15),  
  Nome varchar(15),  
  Cognome varchar (15),  
  DataNascita date,  
  Nazionalita varchar(15),  
  IdSede int,  
  constraint PK_Studente_IdStudente primary key (IdStudente),  
  constraint FK_Studente_Sede_IdSede foreign key (IdSede) references Sede (IdSede))
```

```
create table Corso(  
  IdCorso varchar (15),  
  Nome varchar (30),  
  DipAcad varchar(30),  
  constraint PK_Corso_IdCorso primary key (IdCorso))
```

```
create table Periodo(  
  Anno date,  
  IdStudente varchar(15),  
  IdCorso varchar(15),  
  IsCurrent varchar (5),  
  constraint FK_Periodo_Studente_IdStudente foreign key (IdStudente) references Studente (IdStudente),  
  constraint FK_Periodo_Corso_IdCorso foreign key (IdCorso) references Corso (IdCorso))
```

```
create table Lezione(  
  IdLezione varchar(20),  
  Nome varchar(25),  
  Crediti int,  
  IdCorso varchar(15),  
  constraint PK_Lezione_IdLezione primary key (IdLezione),  
  constraint FK_Lezione_Corso_IdCorso foreign key (IdCorso) references Corso (IdCorso))
```

```
create table Docente(  
  IdDocente varchar(15),  
  Nome varchar(15),  
  Cognome varchar (15),
```

```
DataNascita date,  
IdSede int,  
DipAcad varchar(25),  
constraint PK_Docente_IdDocente primary key (IdDocente),  
constraint FK_Docente_Sede_IdSede foreign key (IdSede) references Sede (IdSede))  
  
create table Lezione_Sede(  
IdLezione varchar(20),  
IdSede int,  
IdDocente varchar(15),  
constraint FK_Lezione_Sede_Lezione_IdLezione foreign key (IdLezione) references Lezione (IdLezione),  
constraint FK_Lezione_Sede_IdSede foreign key (IdSede) references Sede (IdSede),  
constraint FK_Lezione_Docente_IdDocente foreign key (IdDocente) references Docente (IdDocente))
```

--Creazione di tabelle

```
create table Regione(  
  Regione varchar (20),  
  Citta varchar (15),  
  Paese varchar (15),  
  Indirizzo varchar (25),  
  constraint PK_Regione_Regione primary key (Regione))
```

```
create table Sede(  
  IdSede int,  
  Nome varchar(15),  
  Regione varchar(20),  
  constraint PK_Sede_IDSede primary key (IdSede),  
  constraint FK_Sede_Regione_Regione foreign key (Regione) references Regione (Regione))
```

```
create table Dipendente(  
  IdDipendente varchar(10),  
  Nome varchar(15),  
  Cognome varchar (15),  
  DataNascita date,  
  IdSede int,  
  Dipartimento varchar(15),  
  constraint PK_Dipendente_IdDipendente primary key (IdDipendente),  
  constraint FK_Dipendente_Sede_IdSede foreign key (IdSede) references Sede (IdSede))
```

```
create table Studente(  
  IdStudente varchar(15),  
  Nome varchar(15),  
  Cognome varchar (15),  
  DataNascita date,  
  Nazionalita varchar(15),  
  IdSede int,  
  constraint PK_Studente_IdStudente primary key (IdStudente),  
  constraint FK_Studente_Sede_IdSede foreign key (IdSede) references Sede (IdSede))
```

```
create table Corso(  
  IdCorso varchar (15),  
  Nome varchar (30),  
  DipAcad varchar(30),  
  constraint PK_Corso_IdCorso primary key (IdCorso))
```

```
create table Periodo(  
  Anno date,  
  IdStudente varchar(15),  
  IdCorso varchar(15),  
  IsCurrent varchar (5),  
  constraint FK_Periodo_Studente_IdStudente foreign key (IdStudente) references Studente (IdStudente),  
  constraint FK_Periodo_Corso_IdCorso foreign key (IdCorso) references Corso (IdCorso))
```

```
create table Lezione(  
  IdLezione varchar(20),  
  Nome varchar(25),  
  Crediti int,  
  IdCorso varchar(15),  
  constraint PK_Lezione_IdLezione primary key (IdLezione),  
  constraint FK_Lezione_Corso_IdCorso foreign key (IdCorso) references Corso (IdCorso))
```

```
create table Docente(  
  IdDocente varchar(15),  
  Nome varchar(15),  
  Cognome varchar (15),
```

```
DataNascita date,  
IdSede int,  
DipAcad varchar(25),  
constraint PK_Docente_IdDocente primary key (IdDocente),  
constraint FK_Docente_Sede_IdSede foreign key (IdSede) references Sede (IdSede))  
  
create table Lezione_Sede(  
IdLezione varchar(20),  
IdSede int,  
IdDocente varchar(15),  
constraint FK_Lezione_Sede_Lezione_IdLezione foreign key (IdLezione) references Lezione (IdLezione),  
constraint FK_Lezione_Sede_IdSede foreign key (IdSede) references Sede (IdSede),  
constraint FK_Lezione_Docente_IdDocente foreign key (IdDocente) references Docente (IdDocente))
```


-- Interrogazioni del DB

--1) Quantità di studenti per regione e promedio di età

```
SELECT S.Regione, COUNT(*) AS Studenti, AVG(DATEDIFF(yy,St.DataNascita,GETDATE())) AS Età_Media
FROM Studente St
LEFT JOIN Sede S
ON St.IdSede = S.IdSede
GROUP BY S.Regione
```

--2) Elenco di lavoratori siano docenti che dipendenti

```
SELECT Di.Nome, Di.Cognome, Di.IdDipendente, Di.IdSede, Di.Dipartimento AS Settore, 0 AS
Dip_Didattico
FROM Dipendente Di
UNION ALL
SELECT Do.Nome, Do.Cognome, Do.IdDocente, Do.IdSede, Do.DipAcad AS Settore, 1 AS Dip_Didattico
FROM Docente Do
ORDER BY Nome
```

--3) Elenco di lavoratori della regione europea che hanno più di 60 anni

```
SELECT lavoratori.Nome, lavoratori.Cognome, lavoratori.Settore,
DATEDIFF(yy,lavoratori.DataNascita,GETDATE()) AS Età
FROM (SELECT Di.Nome, Di.Cognome, Di.IdDipendente,Di.DataNascita, Di.IdSede, Di.Dipartimento AS
Settore, 0 AS Dip_Didattico
FROM Dipendente Di
UNION ALL
SELECT Do.Nome, Do.Cognome, Do.IdDocente,Do.DataNascita, Do.IdSede, Do.DipAcad AS
Settore, 1 AS Dip_Didattico
FROM Docente Do) lavoratori
WHERE (DATEDIFF(yy,lavoratori.DataNascita,GETDATE()) - 60 ) > 0 and lavoratori.IdSede = 1
```

--4) Quantità di studenti per corso

```
SELECT P.IdCorso, COUNT(S.IdStudente) AS Qty_Studenti
FROM Periodo P
JOIN Studente S
ON S.IdStudente = P.IdStudente
WHERE P.IsCurrent = 'True'
GROUP BY P.IdCorso
```

--5)Elenco di studenti che frequentano un corso del dipartimento academico "Scienze Aziendali"

```
SELECT S.IdStudente, S.Nome, S.Cognome, C.IdCorso, C.Nome, S.IdSede
FROM Periodo P
JOIN Corso C
ON P.IdCorso = C.IdCorso
JOIN Studente S
ON S.IdStudente = P.IdStudente
WHERE P.IsCurrent = 'True' AND C.DipAcad = 'Scienze Aziendali'
```

--6) Elenco di lezioni per sede

```
SELECT S.Nome, LS.IdLezione, L.Nome, C.DipAcad
FROM Lezione_Sede LS
LEFT JOIN Lezione L
ON LS.IdLezione = L.IdLezione
JOIN Sede S
ON LS.IdSede = S.IdSede
JOIN Corso C
ON C.IdCorso = L.IdCorso
ORDER BY S.Nome, C.DipAcad
```

--7) Corsi che hanno una quantità di studenti superiore alla media

```
SELECT P.IdCorso, COUNT(S.IdStudente) AS
```

```

FROM Periodo P
JOIN Studente S
ON S.IdStudente = P.IdStudente
WHERE P.IsCurrent = 'True'
GROUP BY P.IdCorso
HAVING COUNT(S.IdStudente) > (SELECT count(S.IdStudente)/count(P.IdCorso)
                                FROM Periodo P
                                JOIN Studente S
                                ON S.IdStudente = P.IdStudente
                                WHERE P.IsCurrent = 'True')

ORDER BY p.IdCorso

```

--8) Quantità di docenti per sede

```

SELECT D.IdSede, S.Nome, COUNT(D.IdDocente) AS Qty_Docenti
FROM Docente D
JOIN Sede S
ON D.IdSede = S.IdSede
GROUP BY D.IdSede, S.Nome

```

--9) Elenco di lezione per regione ordinate secondo i crediti

```

SELECT LS.IdSede, L.IdLezione, L.Nome, L.Crediti
FROM Lezione L
JOIN Lezione_Sede LS
ON L.IdLezione = LS.IdLezione
ORDER BY LS.IdSede, L.Crediti DESC

```

--10) Ultima volta che ogni studente si è iscritto in un corso

```

SELECT CONCAT(S.Nome, ' ', S.Cognome) AS Studente, MAX(YEAR(P.Anno)) AS Ultima_Iscrizione
FROM Periodo P
JOIN Studente S
ON P.IdStudente = S.IdStudente
GROUP BY CONCAT(S.Nome, ' ', S.Cognome)

```