Partendo da un caso studio di suo interesse si esegua la progettazione di una base dati con almeno 5 relazioni dello schemo concetuale.

Caso studio: DOCENTE E STUDENTE DI UN LICEO

1. La progettazione consiste in un doccumento con
2. Descrizione caso di studio scelto
3. Modello ER
4. Modello logico
5. Progettazione fisica codice DDL, DML per la creazione del data base delle tabelle, definizione della chiave primaria ed esterna e degli attributi
6. Scrivere e/o visualizzare almeno 10 query con almeno due tra group by e order by, join e date

Risoluzione

Descrizione del caso di studio scelto:

Il liceo ha docenti e studenti. Ogni docente può insegnare una o più materie, e ogni studente può essere iscritto a uno o più corsi. I docenti registrano i voti degli studenti nei loro corsi. Inoltre, il liceo tiene traccia dei corsi offerti.

Modello ER:

Di seguito è uno schema ER per il caso di studio:

Entità Studente

Attributi: StudenteID (PK), Nome, Cognome, DataNascita

Entità Docente

Attributi: DocenteID (PK), Nome, Cognome, Materia,

Entità Materia

Attributi: MateriaID (PK), NomeMateria, ...

Entità Corso

Attributi: CorsoID (PK), NomeCorso, Anno, MateriaID (FK), DocenteID (FK), ...

Entità Voti

Attributi: VotoID (PK), StudenteID (FK), CorsoID (FK), DataVoto, Voto, ...

Modello Logico:

Basato sul modello ER, il modello logico definisce le tabelle e le relazioni:

Tabella Studente

StudenteID (PK)

Nome

Cognome

DataNascita

Tabella Docente

DocenteID (PK)

Nome

Cognome

Materia

Tabella Materia

MateriaID (PK)

NomeMateria

Tabella Corso

CorsoID (PK)

NomeCorso

MateriaID (FK)

DocenteID (FK)

Tabella Voti

StudenteID (FK)

CorsoID (FK)

DataVoto

Voto

Progettazione Fisica:

Di seguito, i codici per creare le tabelle nel database:

Creazione della tabella Studente

CREATE TABLE Studente (

StudenteID INT PRIMARY KEY not null,

Nome VARCHAR(50) NOT NULL,

Cognome VARCHAR(50) NOT NULL,

DataNascita DATE

);

Creazione della tabella Docente

CREATE TABLE Docente (

DocenteID INT PRIMARY KEY not null,

Nome VARCHAR(50) NOT NULL,

Cognome VARCHAR(50) NOT NULL,

Materia VARCHAR(100) NOT NULL

);

CREATE TABLE Materia (

MateriaID INT PRIMARY KEY NOT NULL,

NomeMateria VARCHAR(100) NOT NULL

);

CREATE TABLE Corso (

CorsoID INT PRIMARY KEY NOT NULL,

NomeCorso VARCHAR(100) NOT NULL,

MateriaID INT,

DocenteID INT,

FOREIGN KEY (MateriaID) REFERENCES Materia(MateriaID),

FOREIGN KEY (DocenteID) REFERENCES Docente(DocenteID)

);

CREATE TABLE Voti (

StudenteID INT NOT NULL,

CorsoID INT NOT NULL,

DataVoto DATE,

Voto DECIMAL(3, 1),

FOREIGN KEY (StudenteID) REFERENCES Studente(StudenteID),

FOREIGN KEY (CorsoID) REFERENCES Corso(CorsoID)

);

-- Tabella Studente

INSERT INTO Studente (StudenteID, Nome, Cognome, DataNascita)

VALUES

(1, 'Mario', 'Rossi', '2000-05-15'),

(2, 'Luca', 'Bianchi', '1999-08-22'),

(3, 'Anna', 'Verdi', '2001-03-10'),

(4, 'Giulia', 'Neri', '2002-11-05'),

(5, 'Marco', 'Gialli', '1998-07-18'),

(6, 'Elena', 'Rosa', '2003-01-25'),

(7, 'Paolo', 'Celeste', '1997-12-03'),

(8, 'Sara', 'Marroni', '2000-09-14');

-- Tabella Docente

INSERT INTO Docente (DocenteID, Nome, Cognome, Materia)

VALUES

(1, 'Maria', 'Bianchi', 'Matematica'),

(2, 'Luigi', 'Verdi', 'Storia'),

(3, 'Giovanni', 'Rossi', 'Fisica'),

(4, 'Laura', 'Neri', 'Chimica');

-- Tabella Materia

INSERT INTO Materia (MateriaID, NomeMateria)

VALUES

(1, 'Matematica'),

(2, 'Storia'),

(3, 'Fisica'),

(4, 'Chimica');

-- Tabella Corso

INSERT INTO Corso (CorsoID, NomeCorso, MateriaID, DocenteID)

VALUES

(1, 'Matematica',1, 1),

(2, 'Storia ', 2, 3),

(3, 'Fisica ', 4, 2),

(4, 'Chimica', 3, 4);

-- Tabella Voti

INSERT INTO Voti (StudenteID, CorsoID, DataVoto, Voto)

VALUES

(1, 1, '2023-06-15', 8.5),

(2, 2, '2023-06-20', 7.0),

(3, 3, '2023-06-18', 9.0),

(4, 4, '2023-06-17', 7.5),

(5, 1, '2023-06-15', 9.0),

(6, 2, '2023-06-20', 8.0),

(7, 3, '2023-06-18', 8.5),

(8, 4, '2023-06-17', 6.5);

QUERY

1. SELECT\*

FROM corso;

1. SELECT DISTINCT Nome, Cognome

FROM docente

WHERE Materia='Storia';

1. SELECT StudenteID

FROM Voti

ORDER BY Voto;

1. SELECT \*

FROM studente

WHERE nome LIKE 'M%';

1. SELECT COUNT(StudenteID)

FROM voti

WHERE voto> 7;

1. SELECT AVG(Voto)

FROM voti;

1. Per ogni studente, visualizzare i nomi dei studenti con voto con voto minore 7

SELECT nome

FROM studente

INNER JOIN voti

ON studente.StudenteID = voti.StudenteID

where Voto <7;

1. Visualizzare le materie dove gli studenti hanno avuto un voto uguale a 9

SELECT NomeMateria

FROM voti V, materia M, corso C

where V.CorsoID=C.corsoID and C.MateriaID=M.MateriaID and V.voto=9;

1. SELECT voto

FROM voti

GROUP BY Voto;

1. SELECT DataNascita, StudenteID

FROM studente

GROUP BY DataNascita, StudenteID;