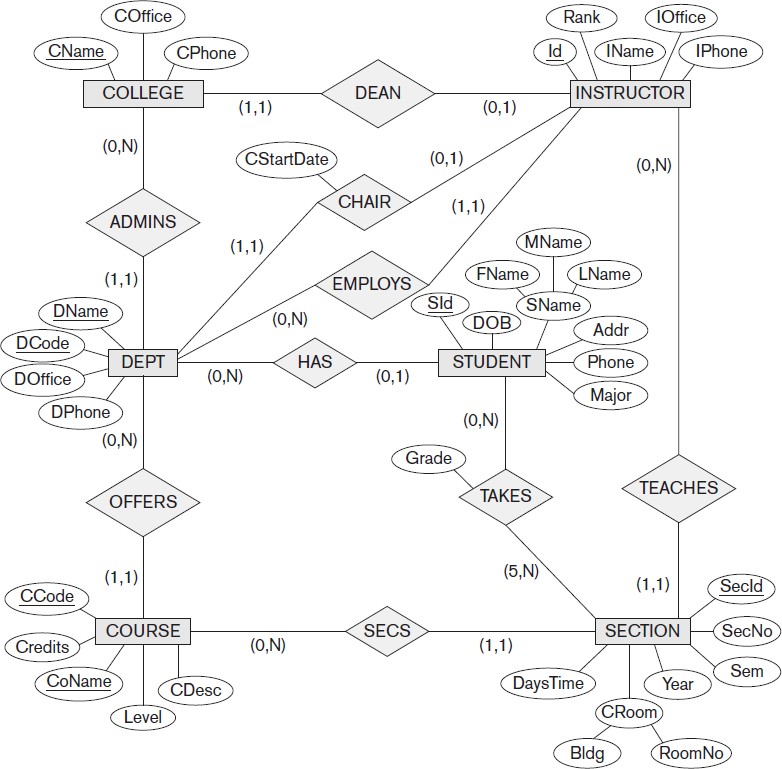
CS631 Assignment-3

1. Consider the UNIVERSITY relational database schema.



**1-a)** (15 points) Map the UNIVERSITY database schema shown above into a relational database schema.

**Solution:**

Qr code

Description automatically generated with low confidence

**1-b)** (15 points) Write appropriate SQL DDL statements for declaring the UNIVERSITY relational database schema.

Choose the appropriate action (reject, cascade, set to NULL, set to default) for each referential integrity constraint, both for the *deletion* of a referenced tuple and for the *update* of a primary key attribute value in a referenced tuple.

**Solution:**

|  |  |
| --- | --- |
| CREATE TABLE STUDENT (  SId INT NOT NULL,  DOB DATE NOT NULL,  FName VARCHAR NOT NULL,  MName VARCHAR NOT NULL,  LName VARCHAR NOT NULL,  Addr VARCHAR NOT NULL,  Phone INT NOT NULL,  Major VARCHAR NOT NULL,  PRIMARY KEY (SId) ); | CREATE TABLE SECTION (  SecID INT NOT NULL,  SecNo INT NOT NULL,  Sem VARCHAR NOT NULL,  Year INT NOT NULL,  Bldg VARCHAR NOT NULL,  RoomNo VARCHAR NOT NULL,  DaysTime DATE NOT NULL,  PRIMARY KEY (SecID) ); |
| CREATE TABLE Takes (  Grade VARCHAR NOT NULL,  SId INT NOT NULL,  SecID INT NOT NULL,  PRIMARY KEY (SId, SecID),  FOREIGN KEY (SId) REFERENCES STUDENT(SId),  FOREIGN KEY (SecID) REFERENCES SECTION(SecID) ); | CREATE TABLE COURSE (  CCode INT NOT NULL,  Credits INT NOT NULL,  CoName INT NOT NULL,  Level INT NOT NULL,  CDesc VARCHAR NOT NULL,  SecID INT NOT NULL,  PRIMARY KEY (CCode, CoName),  FOREIGN KEY (SecID) REFERENCES SECTION(SecID)); |
| CREATE TABLE COLLEGE (  CName VARCHAR NOT NULL,  COffice VARCHAR NOT NULL,  CPhone INT NOT NULL,  DName VARCHAR NOT NULL,  DCode VARCHAR NOT NULL,  PRIMARY KEY (CName),  FOREIGN KEY (DName, DCode)  REFERENCES DEPT(DName, DCode) ); | CREATE TABLE INSTRUCTOR (  InstructorID INT NOT NULL,  Rank VARCHAR NOT NULL,  IName VARCHAR NOT NULL,  IOffice VARCHAR NOT NULL,  IPhone INT NOT NULL,  SecID INT NOT NULL,  CName VARCHAR NOT NULL,  PRIMARY KEY (InstructorID),  FOREIGN KEY (SecID) REFERENCES SECTION(SecID),  FOREIGN KEY (CName) REFERENCES COLLEGE(CName) ); |
| CREATE TABLE DEPT (  DName VARCHAR NOT NULL,  DCode VARCHAR NOT NULL,  DOffice VARCHAR NOT NULL,  DPhone INT NOT NULL,  CStartChair DATE NOT NULL,  InstructorID INT NOT NULL,  SId INT NOT NULL,  CCode INT NOT NULL,  CoName INT NOT NULL,  PRIMARY KEY (DName, DCode),  FOREIGN KEY (InstructorID) REFERENCES INSTRUCTOR(InstructorID),  FOREIGN KEY (SId) REFERENCES STUDENT(SId),  FOREIGN KEY (CCode, CoName) REFERENCES COURSE(CCode, CoName) ); |  |

1. (70 points) Consider the following schema.

SUPPLIERS (SID : *integer*, SNAME : *string*, ADDRESS : *string*) PARTS (PID : *integer*, PNAME : *string*, COLOR : *string*) CATALOG (SID : *integer*, PID : *integer*, COST : *real*)

The key fields are underlined, and the domain of each field is listed after the field name. Thus, SID is the key for SUPPLIERS, PID is the key for PARTS, and SID and PID together form the key for CATALOG. The CATALOG relation lists the prices charged for parts by suppliers. CATALOG.SID is a foreign key referring to SUPPLIERS.SID and CATALOG.PID is a foreign key referring to PARTS.PID. σ Ρ ∩∪ , π ∪⨝∪

|  |  |  |
| --- | --- | --- |
| Write the following queries in | **Relational Algebra** (5 points) | **SQL** (5 points). |
|  | **Solution:** | **Solution:** |
| **2-a)** Find the names of parts supplied by suppliers who are at 1 Central Ave. | R1 🡨 σAddress=1Centrel Ave(Suppliers)  R2 🡨 R1 ⨝ R1.SID = Catalog.SID Catalog  R3 🡨 R2 ⨝ R2.PID = Parts.PID Parts  Result 🡨 πPNAME (R3) | SELECT P.pname  FROM Suppliers S , Part P, Catalog C  Where S.Address= “1 Central Ave” AND S.sid=C.sid AND C.pid=P.pid; |
| **2-b)** Find the city of the suppliers supplying a red part that costs more than $100. | R1 🡨 σColor=Red (Parts)  R2 🡨 R1 \* R1.PID = Catalog.PID Catalog  R3 🡨 σCost>100 (R2)  R4 🡨 R3 \* R3.SID = Suppliers.SID Suppliers  Result 🡨 πAddress (R3) | SELECT S.city  FROM Supplier S, Part P, Catalog C  Where S.sid=C.sid AND P.color=“Red” AND C.cost>100; |
| **2-c)** Find the SIDs of suppliers who supply a red part and a green part. | R1 🡨 σColor=Red (Parts) ∩ σColor=Green (Parts)  R2 🡨 R1 \* R1.PID = Catalog.PID Catalog  Result 🡨 πSID (R2) | SELECT S.sid  FROM Suppliers S, Parts S, Catalog C  WHERE S.sid=C.sid AND P.color=“RED” AND C.pid=P.pid  INTERSEC  SELECT S.sid  FROM Suppliers S, Parts S, Catalog C  WHERE S.sid=C.sid AND P.color=“Green” AND C.pid=P.pid; |
| **2-d)** Find the SIDs of suppliers who supply a red part or a green part. | R1 🡨 σColor=Red (Parts)  R2 🡨 σColor=Green (Parts)  R3 🡨 R1∪R2  R4 🡨 R3 \* R3.PID = Catalog.PID Catalog  Result 🡨 πSID (R4) | SELECT S.sid  FROM Suppliers S, Parts S, Catalog C  WHERE S.sid=C.sid AND P.color=“Red” AND C.pid=P.pid  UNION  SELECT S.sid  FROM Suppliers S, Parts S, Catalog C  WHERE S.sid=C.sid AND P.color=“Green” AND C.pid=P.pid; |
| **2-e)** Find pairs of PIDs such that the part with the first PID is sold at a higher price by a supplier than the part with the second PID. |  |  |
| **2-f)** Find the PIDs of parts supplied by a supplier who is at the city of Newark and by a supplier who is at the city of Trenton. | R1 🡨 σCity=Newark(Suppliers)  R1 🡨 σCity=Trenton(Suppliers)  R3 🡨 Catalog \* Catalog.SID = R1.SID R1  R4 🡨 Catalog \* Catalog.SID = R2.SID R2  Result 🡨 πPID (R3 ∩ R4) | SELECT P.pid  FROM Parts P, Suppliers S, Catalog C  WHERE S.city=“Newark” AND S.sid=C.sid AND C.pid=P.pid  INTERSECT  SELECT P.pid  FROM Parts P, Suppliers S, Catalog C  WHERE S.city=“Trenton” AND S.sid=C.sid AND C.pid=P.pid |
| **2-g)** Find the PIDs of parts supplied by each and every supplier. | Result 🡨 πCatalog.PID(Catalog) | SELECT PID  From Catalog |