**Problem Set 2**

***Due Date:-***Sept 23 at 11:59pm

**Instructions:-**

a. For Problems that require explanations, create space below the question and write your answers.

b. For drawing type problems, you can use MS Word or Visio. Be sure to name your files appropriately.

Submission:- Create a zipped folder with all your assignment files - rename the folder to have the form:-***YourStudentId#\_CS60\_Lastname\_Firstname\_PS2.docx***.After completing the project, load the zipped folder using the Problem Set 2 Upload option. *Tables from Q1 Problem Set2 are also available as data files if you prefer to use MS Access to try them.*

If you need to make changes to an already uploaded assignment, change your folder name to*:- YourStudentId#\_CS60\_V***X***\_Lastname\_Firstname\_PS2.docx***,***where X indicates the version number.*

**Problem 1**:- [13 Points]

Use the following tables to answer Parts (a) to (g) below:-

Table Name:- TRUCK

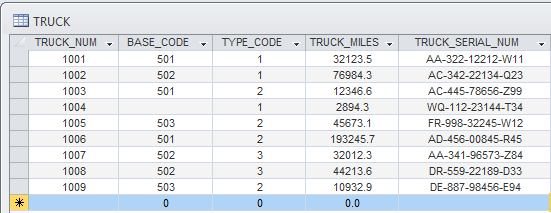


Table Name:- BASE

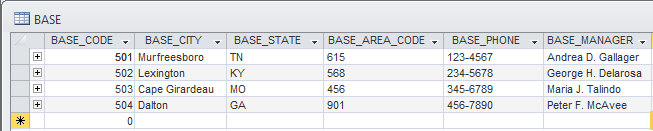
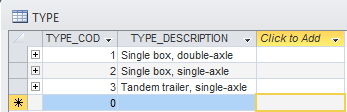


Table Name:- TYPE



1. For each table, identify the primary key and the foreign key(s). If a table does not have a foreign key, write None.

Table: TRUCK

Primary key: TRUCK\_NUM

Foreign key(s): BASE\_CODE, TYPE\_CODE

Table: BASE

Primary key: BASE\_CODE

Foreign key(s): None

Table: TYPE

Primary key: TYPE\_CODE

Foreign key(s): None

1. Do the tables exhibit entity integrity? Answer yes or no and explain your answer.

Yes, the tables exhibit entity integrity because all primary key entries in each table are unique, and no parts of any primary keys are null.

1. Do the tables exhibit referential integrity? Answer yes or no explain your answer.

Yes, the table exhibit referential integrity because all foreign keys either have a null entry or a non-null foreign key value that references an existing primary key value.

1. Identify TRUCK tables candidate key(s).

Table: TRUCK

Candidate key(s): TRUCK\_NUM, TRUCK\_SERIAL\_NUM

Note that although TRUCK\_MILES also uniquely identifies each row in the table, it has been omitted because cannot be assumed to be true. In other words, it is possible that at a given time two trucks might have the same value for the attribute TRUCK\_MILES.

1. For each table, identify a superkey and a secondary key.

Table: TRUCK

Superkey(s): TRUCK\_NUM, TRUCK\_SERIAL\_NUM, (TRUCK\_NUM, TRUCK\_SERIAL\_NUM), (TRUCK\_NUM, TYPE\_CODE), etc.

Secondary key(s): TRUCK\_SERIAL\_NUM

Table: BASE

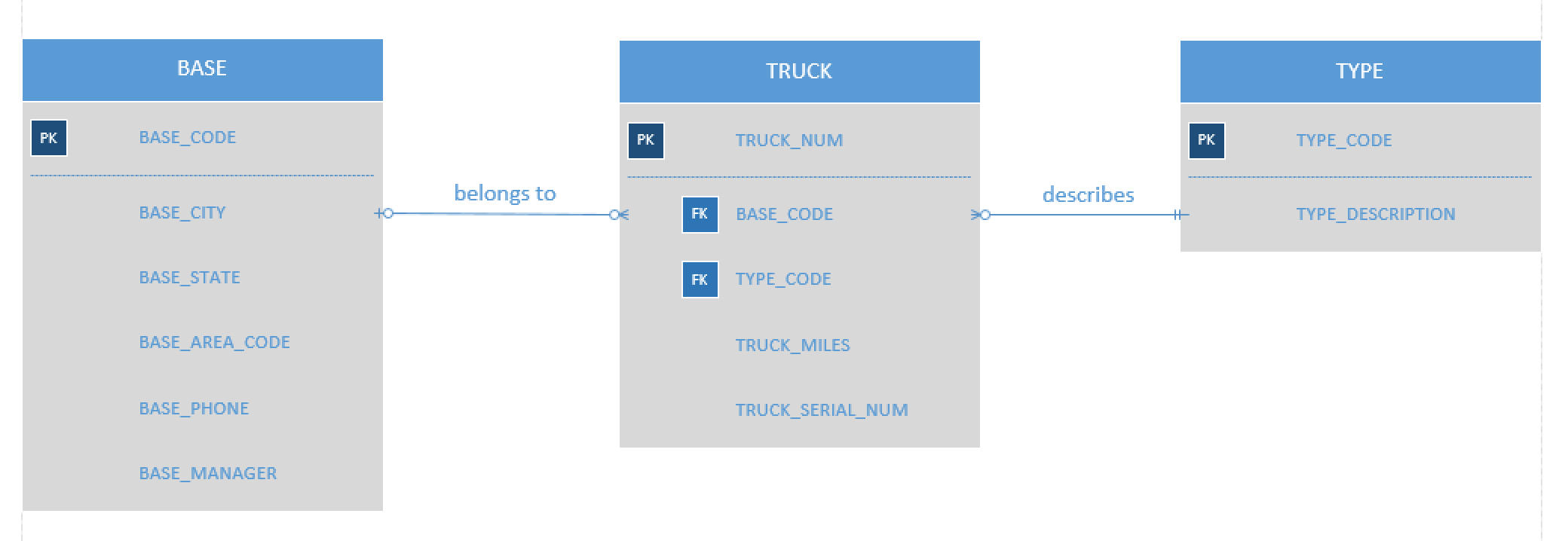
Superkey(s): BASE\_CODE, (BASE\_CODE, BASE\_MANAGER), (BASE\_CODE, BASE\_CITY, BASE\_STATE), etc.

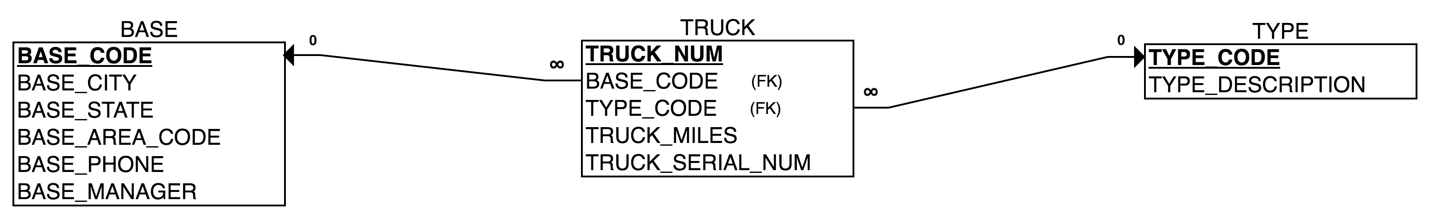
Secondary key(s): (BASE\_CITY, BASE\_MANAGER), (BASE\_CITY, BASE\_PHONE)

Table: TYPE

Superkey(s): TYPE\_CODE

Secondary key(s): TYPE\_DESCRIPTION

1. Create the ERD for this database.
2. Create the relational diagram for this database.



**Problem 2**:- [16 Points]

The Jonesburgh County Basketball Conference (JCBC) is an amateur basketball association. Each city in the county has one team as its representative. Each team has a maximum of 12 players and a minimum of 9 players. Each team also has up to three coaches (offensive, defensive, and physical training coaches). During the season, each team plays two games (home and visitor) against each of the other teams. Given those conditions, do the following:-

1. Identify the connectivity of each relationship.

Entities:

* CITY
* TEAM
* PLAYER
* COACH
* GAME

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ENTITY** | **RELATIONSHIP** | **CONNECTIVITY** | **CARDINALITY** | **RELATIONSHIP STRENGTH** | **ENTITY** |
| CITY | sponsors | 0:M | (0, N) | Weak | TEAM |
| TEAM | represents | 0:1 | (0, 1) | Weak | CITY |
| TEAM | has | 0:M | (9, 12) | Weak | PLAYER |
| PLAYER | plays for | 1:1 | (1, 1) | Weak | TEAM |
| TEAM | has | 0:M | (0, 3) | Weak | COACH |
| COACH | coaches | 1:1 | (1, 1) | Weak | TEAM |
| TEAM | plays | 0:M | (2, N) | Weak | GAME |
| TEAM | plays | 0:M | (2, N) | Weak | GAME |
| GAME | is played by | 1:1 | (1, 1) | Weak | TEAM |
| GAME | is played by | 1:1 | (1, 1) | Weak | TEAM |

Note the following:

* Although each city sponsors only one team at this point, it is likely that more teams will be organized in the future. In order to provide flexibility for this in the future, the connectivity between CITY and TEAM is defined at 0:M. The relationship participation is defined as optional because from an operational standpoint we want the ability to add a CITY to the basketball league database whose team(s) will be added a later time.
* The relationship participation between TEAM and CITY is defined to be optional so that a newly organized team can be added to the database without the requirement of determining which city they will represent first.
* Note that the relationship participation between TEAM and PLAYER is defined as optional for the practicality of creating a team and then adding players as they are found. Although the business rules define a cardinality of (9, 12) for this relationship, this requirement must be implemented at the application software level and therefore does not conflict with the optional relationship participation at the DBMS level.
* Note that the relationship between TEAM and GAME is a recursive relationship, so it appears twice in the table above. This is because each game has a home and an away team, so each row in GAME contains both the primary key of the home team and the primary key of the away team as foreign keys.
* Note that the cardinality between TEAM and GAME is (2, N) because each team plays at least two games (once as the home team and once as the away team).

1. Identify the type of dependency that exists between CITY and TEAM.

We have defined the relationship “CITY sponsors TEAM” so that the TEAM entity is existence-independent. That is, the foreign key is allowed to be null for operational purposes. We want to provide the ability to add a team before deciding which city it will represent.

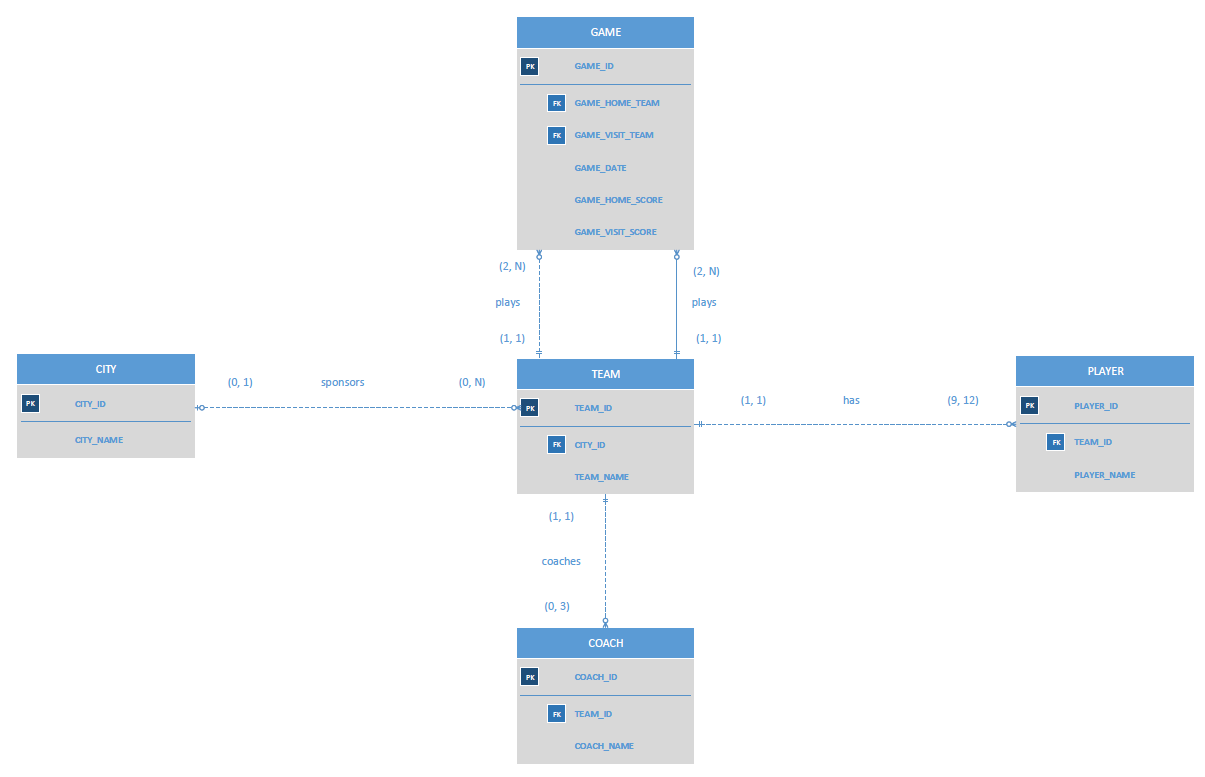
1. Identify the cardinality between teams and players and between teams and city.

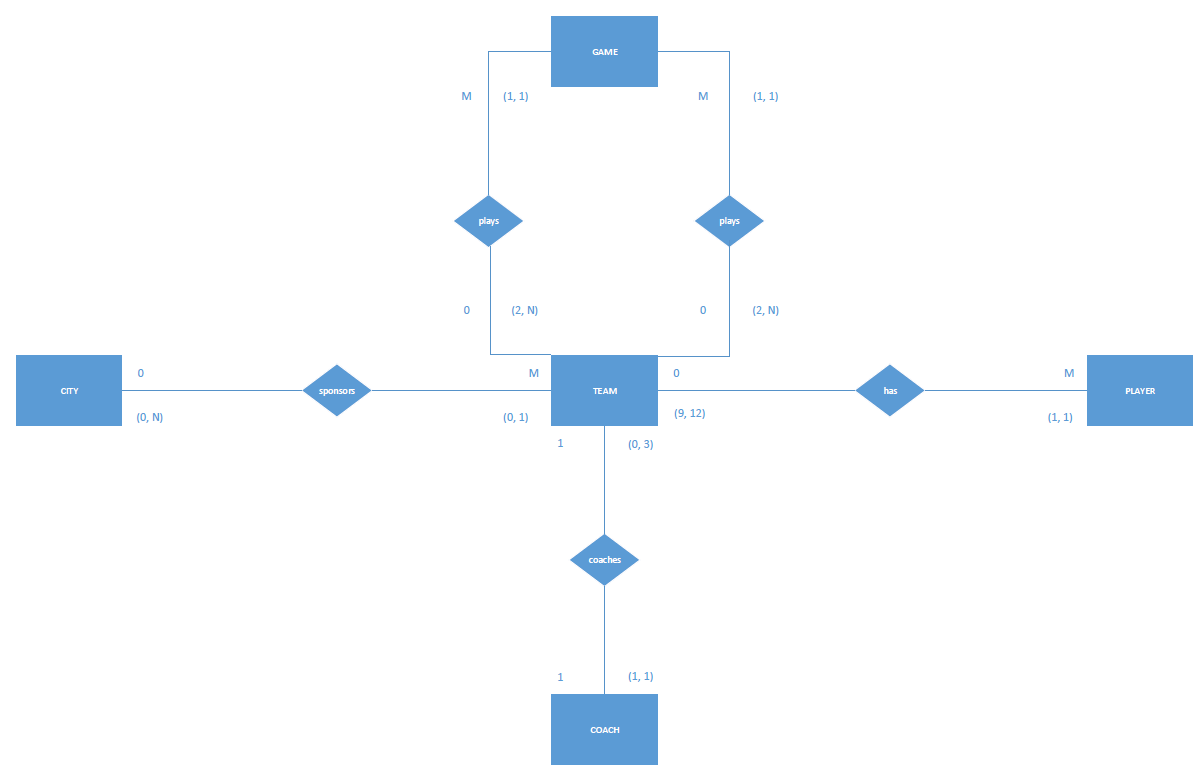
As can be seen in the table from part (a), the cardinality between TEAM and PLAYER is (9, 12) and the cardinality between TEAM and CITY is (0, 1).

1. Identify the dependency between coach and team and between team and player.

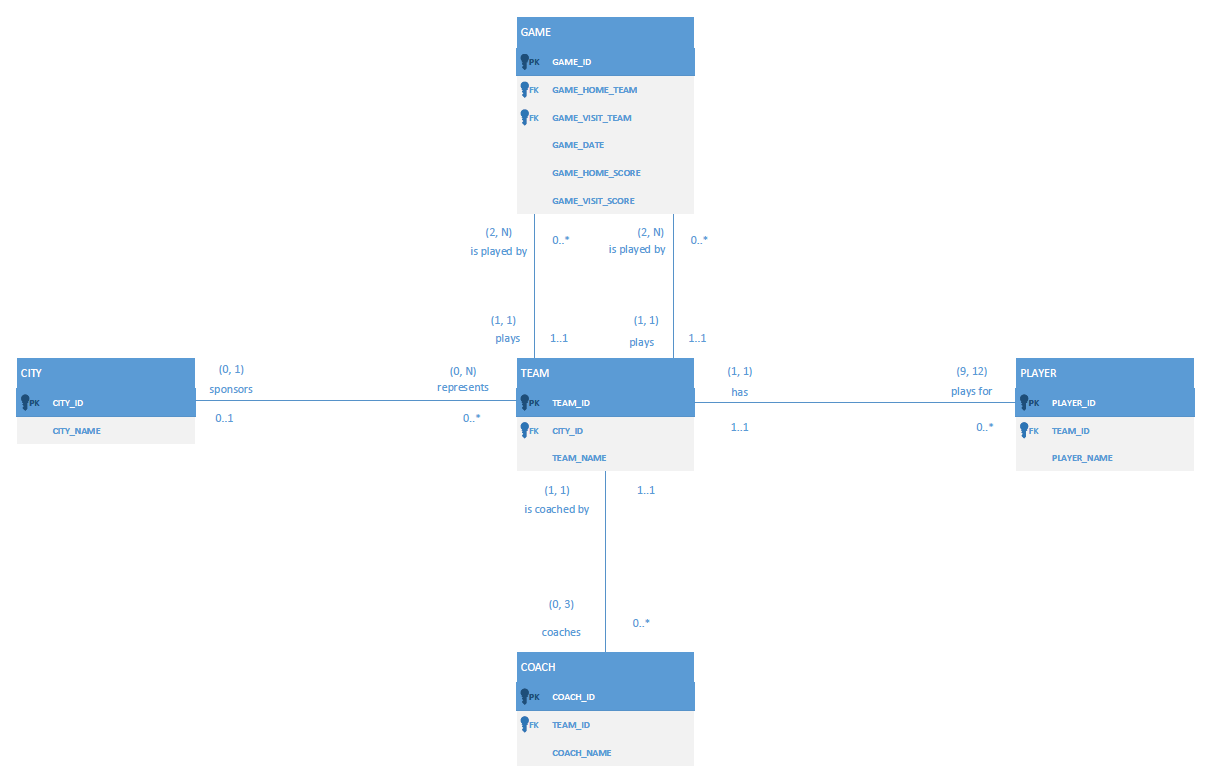
We have defined the relationship “COACH coaches TEAM” so that the COACH entity is existence-dependent. That is, the foreign key indicating the team the coach coaches is mandatory and cannot be null.

We have defined the relationship “TEAM has PLAYER” so that the PLAYER entity is existence-dependent. That is, when a PLAYER is added to the database it has to be associated with a TEAM. The foreign key indicating which team a player plays for cannot be null.

1. Draw the Chen and Crow’s Foot ERDs to represent the JCBC database.



1. Draw the UML class diagram to depict the JCBC database.



**Problem 3**:- [10 Points]

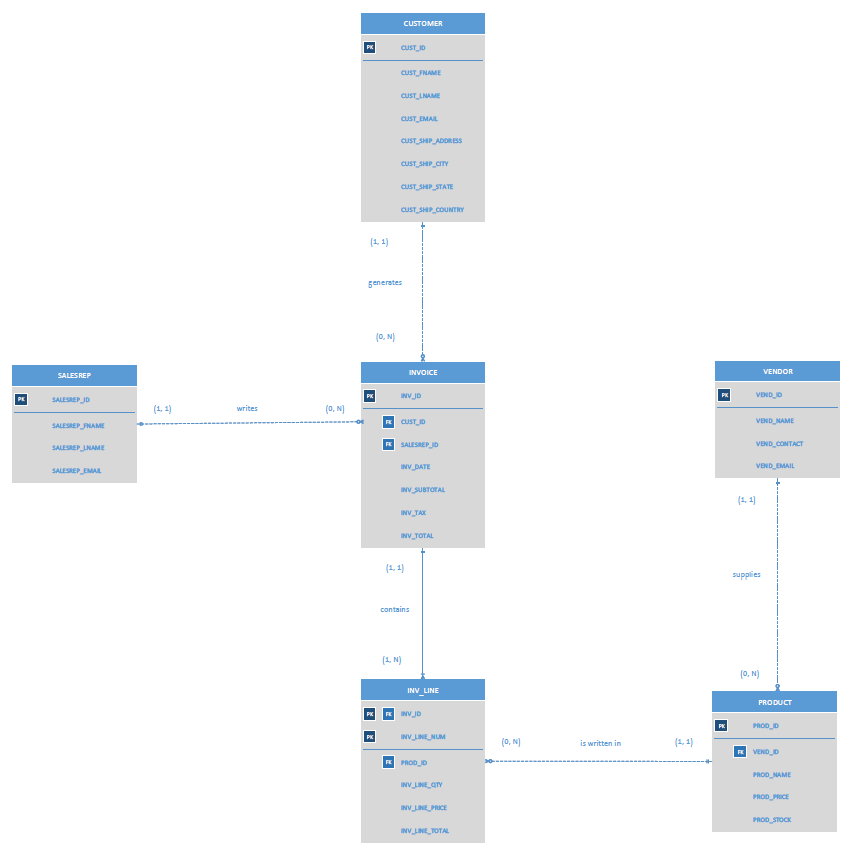
Create an ERD based on the Crow’s Foot notation, using the following requirements:-

1. An INVOICE is written by a SALESREP. Each sales representative can write many invoices, but each invoice is written by a single sales representative.
2. The INVOICE is written for a single CUSTOMER. However, each customer can have many invoices.
3. An INVOICE can include many detail lines (LINE), each of which describes one product bought by the customer.
4. The product information is stored in a PRODUCT entity.
5. The product’s vendor information is found in a VENDOR entity.

Entities:

* INVOICE
* SALESREP
* CUSTOMER
* INV\_LINE
* PRODUCT
* VENDOR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ENTITY** | **RELATIONSHIP** | **CONNECTIVITY** | **CARDINALITY** | **RELATIONSHIP STRENGTH** | **ENTITY** |
| SALESREP | writes | 0:M | (0, N) | Weak | INVOICE |
| INVOICE | is written by | 1:1 | (1, 1) | Weak | SALESREP |
| CUSTOMER | generates | 0:M | (0, N) | Weak | INVOICE |
| INVOICE | is generated by | 1:1 | (1, 1) | Weak | CUSTOMER |
| INVOICE | contains | 1:M | (1, N) | Strong | INV\_LINE |
| INV\_LINE | belongs to | 1:1 | (1, 1) | Strong | INVOICE |
| INV\_LINE | describes | 1:1 | (1, 1) | Weak | PRODUCT |
| PRODUCT | is written in | 0:M | (0, N) | Weak | INV\_LINE |
| PRODUCT | is supplied by | 1:1 | (1, 1) | Weak | VENDOR |
| VENDOR | supplies | 0:M | (0, N) | Weak | PRODUCT |

****

Note the following:

* The relationship participation between VENDOR and PRODUCT is defined to be optional with connectivity 0:M so that the database can be used to store contact information for potential new vendors, who don’t yet have products entered in the system.
* The relationship participation between PRODUCT and VENDOR is defined to be mandatory with connectivity 1:1 so that for every product that can be purchased by a customer, there is a corresponding vendor who can supply it.
* There is one strong relationship in this ERD, namely, that between INVOICE and INV\_LINE. Notice that the INV\_LINE entity inherits a primary key component from the INVOICE entity, INV\_ID.

**Problem 4:- [11 Points]**

Using the Crow’s Foot notation, create an ERD that can be implemented for a medical clinic, using the following business rules:-

* A patient can make many appointments with one or more doctors in the clinic, and a doctor can accept appointments with many patients. However, each appointment is made with only one doctor and one patient.
* Emergency cases do not require an appointment. However, for appointment management purposes, an emergency is entered in the appointment book as “unscheduled.”
* If kept, an appointment yields a visit with the doctor specified in the appointment. The visit yields a diagnosis and, when appropriate, treatment.
* With each visit, the patient’s records are updated to provide a medical history.
* Each patient visit creates a bill. Each patient visit is billed by one doctor, and each doctor can bill many patients.
* Each bill must be paid. However, a bill may be paid in many installments, and a payment may cover more than one bill.
* A patient may pay the bill directly, or the bill may be the basis for a claim submitted to an insurance

company.

* If the bill is paid by an insurance company, the deductible is submitted to the patient for payment.

Entities:

* PATIENT
* APPOINTMENT
* DOCTOR
* VISIT
* VISIT\_LINE
* BILL
* PATIENT\_PMT
* INSURANCE
* INSURANCE\_PMT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ENTITY** | **RELATION-SHIP** | **CONNECT-IVITY** | **CARDINAL-ITY** | **RELATION-SHIP STRENGTH** | **ENTITY** |
| PATIENT | requests | 0:M | (0, N) | Weak | APPOINTMENT |
| APPOINTMENT |  | 1:1 | (1, 1) | Weak | PATIENT |
| APPOINTMENT |  | 1:1 | (1, 1) | Weak | DOCTOR |
| DOCTOR | keeps | 0:M | (0, N) | Weak | APPOINTMENT |
| DOCTOR | records | 0:M | (0, N) | Weak | VISIT |
| VISIT |  | 1:1 | (1, 1) | Weak | DOCTOR |
| PATIENT | attends | 0:M | (0, N) | Weak | VISIT |
| VISIT |  | 1:1 | (1, 1) | Weak | PATIENT |
| VISIT | contains | 1:M | (1, N) | Strong | VISIT\_LINE |
| VISIT\_LINE |  | 1:1 | (1, 1) | Strong | VISIT |
| VISIT | generates | 0:1 | (0, 1) | Weak | BILL |
| BILL |  | 0:1 | (0, 1) | Weak | VISIT |
| PATIENT\_PMT | reduces | 1:1 | (1, 1) | Weak | BILL |
| BILL |  | 0:M | (0, N) | Weak | PATIENT\_PMT |
| PATIENT\_PMT |  | 1:1 | (1, 1) | Weak | PATIENT |
| PATIENT | pays | 0:M | (0, N) | Weak | PATIENT\_PMT |
| PATIENT |  | 1:1 | (1, 1) | Weak | INSURANCE |
| INSURANCE | insures | 0:M | (0, N) | Weak | PATIENT |
| INSURANCE\_PMT |  | 1:1 | (1, 1) | Weak | INSURANCE |
| INSURANCE | pays | 0:M | (0, N) | Weak | INSURANCE\_PMT |
| INSURANCE\_PMT | reduces | 1:1 | (1, 1) | Weak | BILL |
| BILL |  | 0:M | (0, N) | Weak | INSURANCE\_PMT |

Note the following:

* The relationship between VISIT and VISIT\_LINE is similar to the example of INVOICE and INV\_LINE used throughout the textbook.
* The relationship participation between VISIT and BILL is defined to be optional with connectivity 0:1 to accommodate the case when a patient visit does not generate a bill.
* The relationship participation between BILL and VISIT is defined to be optional with connectivity 0:1 to accommodate the case when a patient bill is generated that is not directly related to a visit.
* The relationship participation between PATIENT and INSURANCE is defined to be mandatory with connectivity 1:1, which makes the assumption that all patients have some form of insurance.
* The APPOINTMENT entity contains an attribute APPT\_SCHEDULED, which is designed to be implemented as a Boolean value. The value would be FALSE for an “unscheduled” emergency visit or TRUE for a regular visit made by appointment.

