**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.

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

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

Table Name:- TRUCK

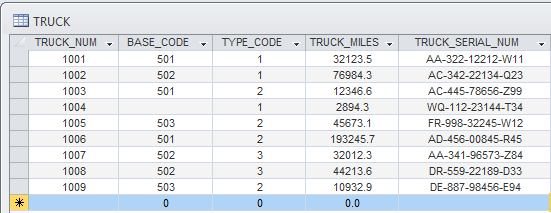


Table Name:- BASE

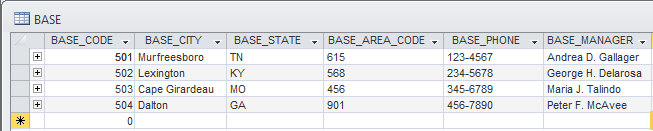
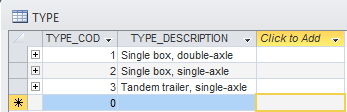


Table Name:- TYPE



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

Truck table Primary Key: TRUCK\_NUM

Base table Primary Key: BASE\_CODE

Type table Primary Key: TYPE\_CODE

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

For each table the primary keys contain ***no*** null values, and **is** unique so *yes to exhibiting* entity integrity.

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

Each table does exhibit referential integrity. The Truck table which has foreign keys, and can be checked by using the values appearing in the related table’s primary key for the same attributes which are all matching or else with one case being null (permissible). Therefore since that does not violate rules of referential integrity then the answer is that Truck table does exhibit it. The other tables do not have any foreign keys, so the term does not apply to these other tables.

1. Identify TRUCK tables candidate key(s).

Candidate Key = [TRUCK\_NUM] This is Actually the Selected Primary Key

Candidate Key = [TRUCK\_SERIAL\_NUM] Another good option since uniquely assigned

\*While TRUCK\_MILES would work for this small set of data, I would recommend against calling it a candidate key if the actual data set is much larger since that would be likely to cause problems with non-unique values. However in current state of Truck entity then I guess its candidate key.

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

*I used the logic that superkeys contain a candidate or primary key, and secondary keys are whichever attributes separate the data into a subsets, e.g. Truck type and base model*

**Trucks**

Superkey = [TRUCK\_SERIAL\_NUM, BASE\_CODE, TYPE\_CODE]

Secondary Key = [BASE\_CODE, TYPE\_CODE]

**Base**

Superkey = [BASE\_CODE, BASE\_CITY, BASE\_STATE]

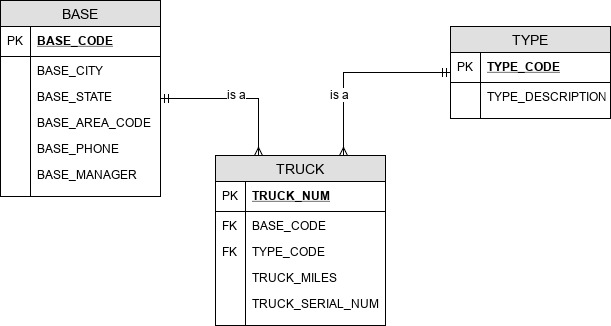
Secondary Key = [BASE\_MANAGER, BASE\_AREA\_CODE, BASE\_PHONE]

**Type**

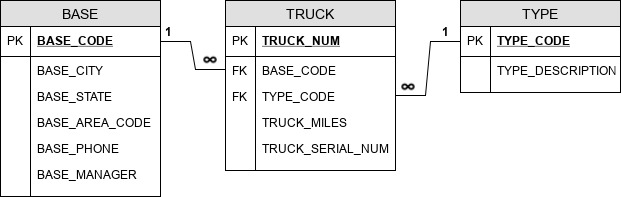
Superkey = [TYPE\_CODE, TYPE\_DESCRIPTION]

Secondary Key = [TYPE\_DESCRIPTION] or none, since two attributes total not many choices

1. Create the ERD for this database.



1. 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Relationship Entities | Relationship | Connectivity | Cardinality |
| Team, City | 1:1 |  | [Team] connector: (1,1)  [City] connector: (1,1) |
| Team, Player | 1:M |  | [Team] connector: (1,1)  [Player] connector: (9,12) |
| Team, Coaches | 1:M |  | [Team] connector: (1,1)  [Coaches] connector: (1,3) |
| County, City | 1:M |  | [County] connector: (1,1)  [City] connector: (M) |

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

Team is existence-dependent on City, since teams have to be associated directly with a city, without which there would be an absence of this important foreign key in the Team table.

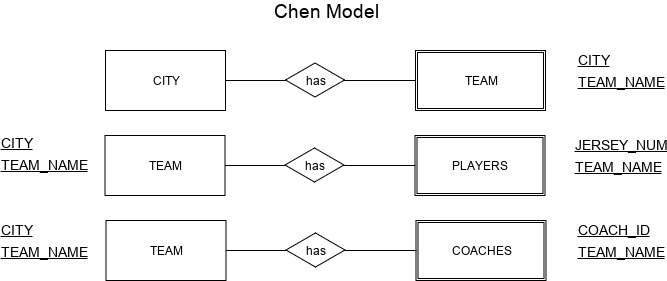
1. Identify the cardinality between teams and players and between teams and city.

Teams and players have the relationship of 1:M and thus have a specific cardinality for the players of a team which is the range of 9 to 12. The players are each assigned to a single team, which means cardinality is actually just 1 for that connector. For teams to city, since I assume that **only cities with teams are populating the database**, so a cardinality of 1 applies to both connectors and there is neither the case in which multiple teams are assigned to a city, nor is there any city that doesn’t have a team (or it would be dropped from the entity table).

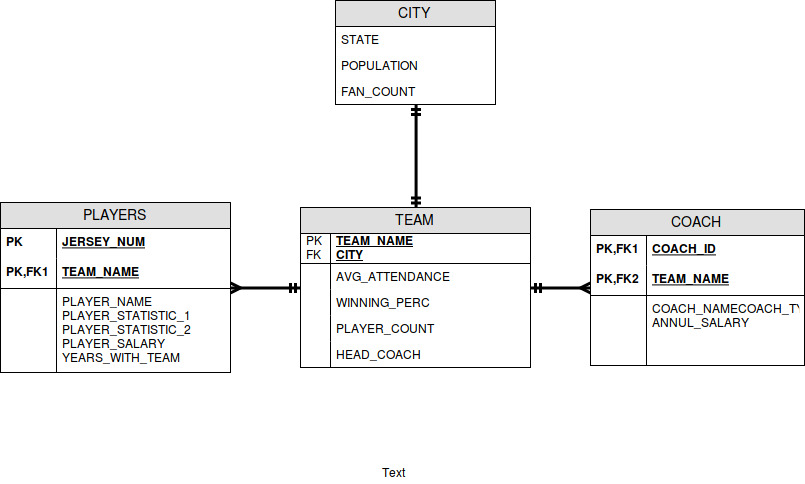
1. Identify the dependency between coach and team and between team and player.

Extending the logic mentioned in part (c) of the Team mandatory foreign key for City it makes sense that Coaches and Players are also existence-dependent on a Team. That is these entities in the database will have mandatory foreign keys of Teams. Sometimes however Players and Coaches may operate on their own, and it might occur (if for example someone is in between roles and not assigned to a specific team)...that situation would warrant a separate business rule for how to handle the question of a missing foreign key for Team in a players or coaches table. **In a real world scenario of professional sports it is the team’s owners who are the representing the Team entity and these owners along with the players and coaches are not always so clearly independent or dependent of each other.**

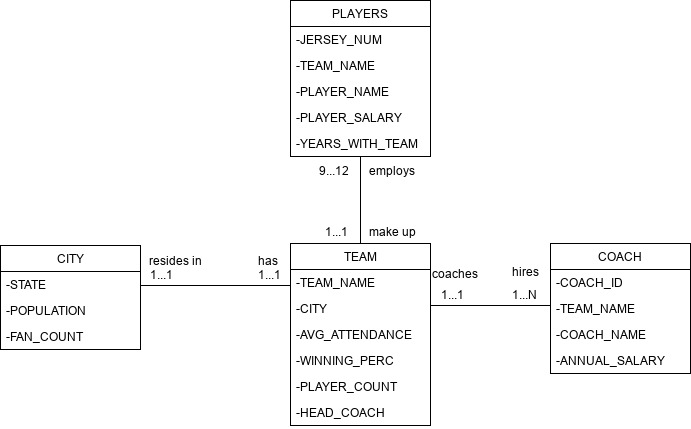
1. Draw the Chen and Crow’s Foot ERDs to represent the JCBC database.



**Crow’s Foot ERD**



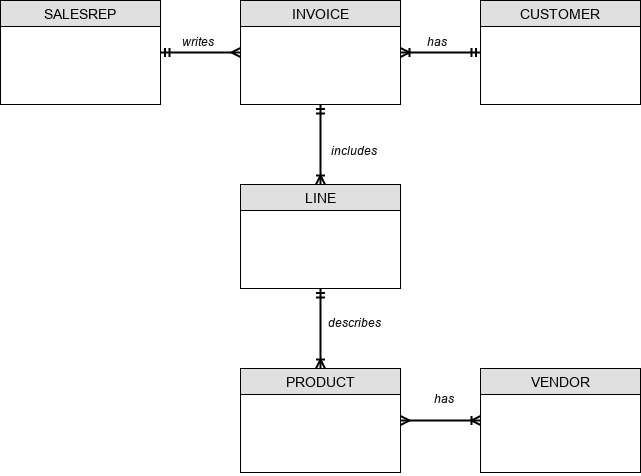
1. Draw the UML class diagram to depict the JCBC database.



**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.



**NOTE: Not sure why the Product and Vendor relationship is a many to many, since this has not been addressed in the description of the problem so not sure if there is a wrong assumption or not.**

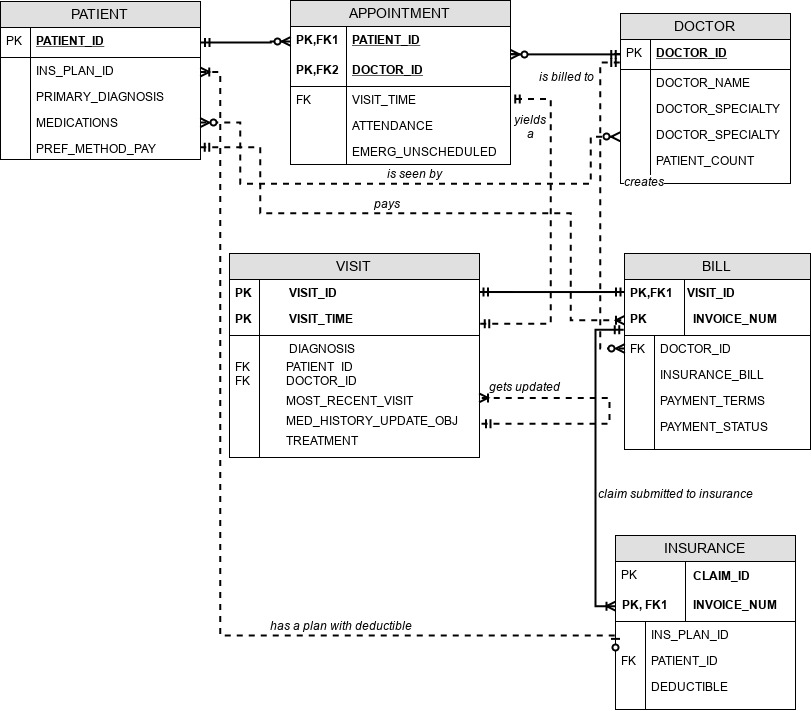
**Problem 4:- [11 Points]**

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.



Some additional information from diagram in terms of relationships/connectivity, and attributes in general according to the specifications or business rules:

DOCTOR\_ID is used not only for appointment setting, but also for billing as a foreign key to the Bill entity. (It is the primary key of course as well for Doctor table)

INVOICE is used to create a bill after a visit and goes into the Bill entity

MED\_HISTORY\_UPDATE\_OBJ is a attribute that is meant to track the medical records which are assumed to be kept in another file separate from the database (for ease of use).

PREF\_METHOD\_PAY is meant to collect some payment information at the Patient table in order to assist billing with per customer information regarding billing aspects not specific to a visit.

PAYMENT\_TERMS / PAYMENT\_STATUS are attributes that will track payment in installments and payments by patients covering more than one bill.

The APPOINTMENT entity is a composite or associative entity creating a bridge table between doctor and patient.

INVOICE\_NUMBER is a Primary key in both the Insurance and Bill entities making it a strong relationship

VISIT\_TIME is the attribute of appointment that gets recorded into the visit entity creating a particular entry.