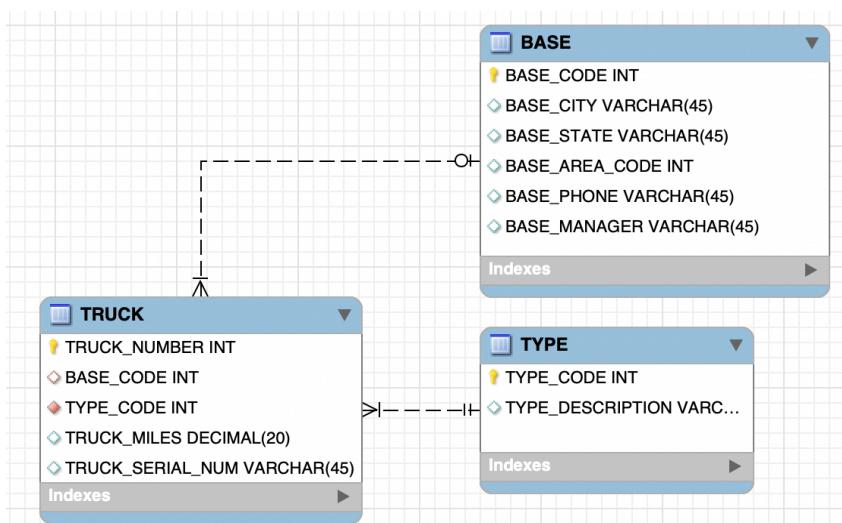


Chapter Three:

1. What is the difference between a database and a table?
Ans: Table belongs to the database. Tables are normally a representation of actual world entities. Database is a collection of different tables that are constructed following business rules. Database always support convenient manipulation of tables and access management.
2. Why are entity integrity and referential integrity important in a database?
Ans: Because database are designed to minimize the data redundancy and maximize data consistency. Entity integrity and referential integrity is the key approach to achieve this.
3. Using the STUDENT and PROFESSOR tables, illustrate the difference between a natural join, an equijoin and an outer join.
 - a. Natural join only returns students and professors that has same PROF_CODE and it will eliminate duplicate results.
 - b. Equijoin will also return students and professors that has same PROF_CODE but it will not eliminate duplicate results.
 - c. Outer join: If we do STUDENT left outer join PROFESSOR, it will return not only the common records but also all the rest record in STUDENT and for these non-common records, it will leave column features from PROFESSOR table as NULL.
4. For each table, identify the primary key and the foreign key(s). If a table does not have a foreign key, write None.
 - a. EMPLOYEE: Primary key (EMP_CODE), Foreign key (STORE_CODE)
 - b. STORE: Primary key (STORE_CODE), Foreign key (EMP_CODE, STORE_CODE)
 - c. REGION: Primary key (REGION_CODE), Foreign key ()
5. Describe the type(s) of relationship(s) between STORE and REGION.
Ans: A region will have many stores, so the relationship between region and store is one-to-many.
6. Create the ERD for this database.



Chapter Four:

1. What two conditions must be met before an entity can be classified as a weak entity?
Give an example of a weak entity.

Ans: The entity cannot exist without the entity with which it has a relationship. And the entity has a primary key that is partially or totally derived from parent entity in the relationship. For example, an employee may have multiple dependents where the dependents cannot exist without an employee and it will have the employee's primary key in its primary key.

2. What is a strong (or identifying) relationship, and how is it depicted in a Crow's Foot ERD?

Ans: Strong relationship means the two entities in this relationship are existence-dependent. It is represented as solid line in Crow's Foot ERD.

3. What is a composite entity, and when is it used?

The composite entity is used to transform a M:N relationship to two 1:M relationships. It will at least contain the two primary keys as its foreign keys from the two entities that is linked.

4. Write the business rules that are reflected in it.

- a. Customer can own many or zero cars, a car needed to be owned by one and only one customer.
- b. A car get many or zero maintenances, a maintenance is done on one and only one car.
- c. A maintenance contains many but at least one maint-line, a maint-line is included in one and only one maintenance.
- d. A part can be recorded in many or zero maint-lines, and maint-line is written in one and only one part.

5. What is a derived attribute? Give an example. What are the advantages or disadvantages of storing or not storing a derived attribute?

A derived attribute is an attribute that calculated from another attribute.

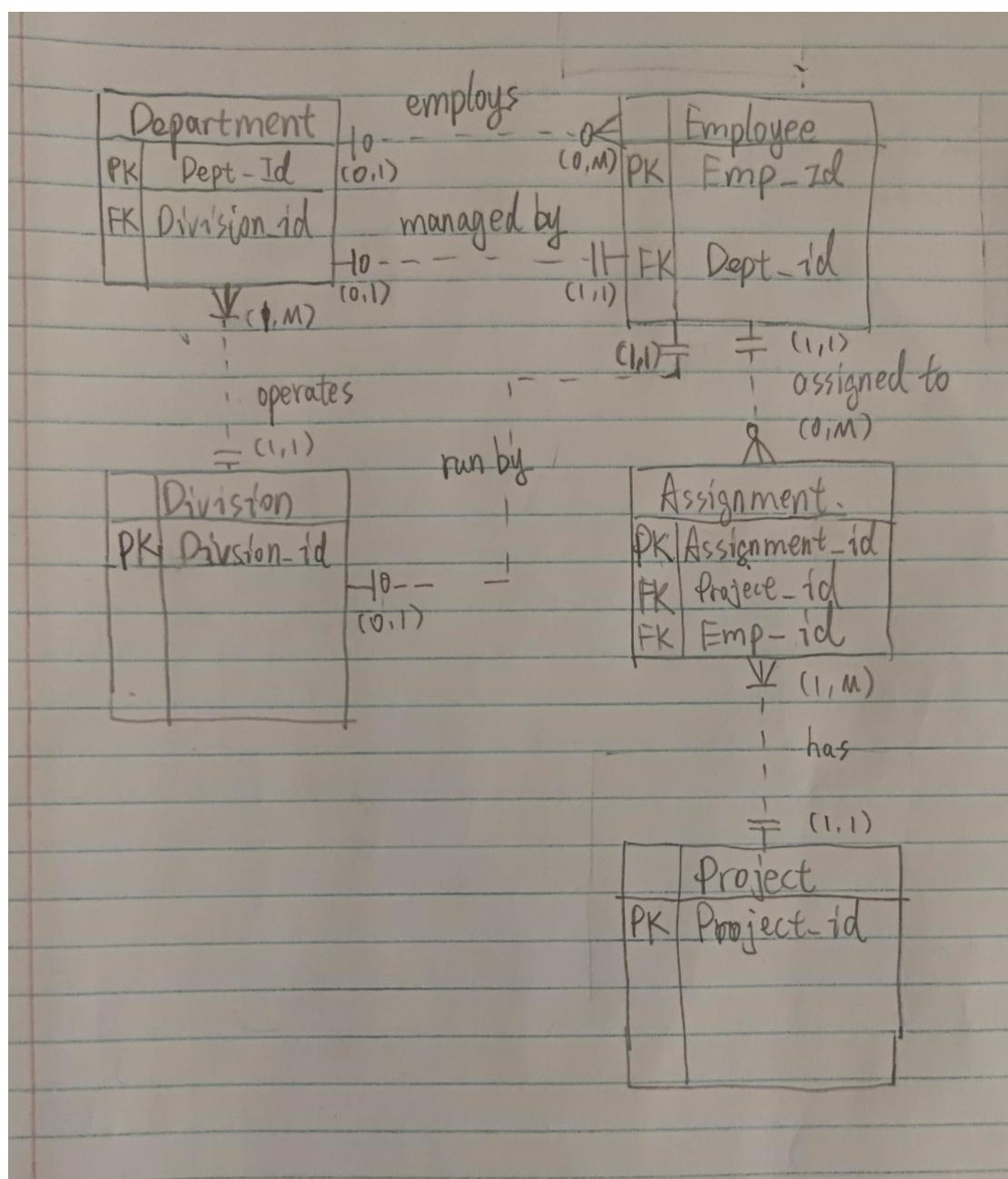
By storing the derived attribute, you save the time from reading and calculating the attribute every time it gets queried, and the results can be used to keep track of historical data. But it will require extra maintenance to check the value of derived attribute whenever there is an update for other attributes.

By not storing the derived attribute, you save the space and you can make sure every time you calculate the derived attribute, you will get a updated number. But doing calculation for every query will cost more time.

- What are multivalued attributes, and how can they be handled within the database design?

Multivalued attributes are attributes that has many values like an array or a list. They are usually handled within the database by creating a new table that one record will hold one value for the multivalued attributes. The new table will also contain a foreign key for the origin entity.

7.



8.

