Yun Zhou 300442776

School of Engineering and Computer Science

SWEN 304/439 **Database System Engineering**

Assignment 1

The objective of this assignment is to test your understanding of database foundations, basic terms, and the relational data model the entity relational model. It is worth 15% of your final grade. The assignment is marked out of 100.

The assignment is due on **Friday, 1 April, 23:59 pm**. Please submit your assignment in **pdf** via the submission system.

Question 1 [20 marks]

The TOP500 project lists the 500 most powerful non-distributed computer systems in the world (also called supercomputers). Suppose we use a relational database to manage the current and future data of this project. For this purpose, we use a relation schema with the attribute set {Performance, Name, Manufacturer, Country, Year}.

The following table shows a portion of the current instance of the SUPERCOMPUTERS relation schema that stores data for some supercomputers. Note that the performance in the table is measured in petaFLOPS.

SUPERCOMPUTERS

Performance	Name	Manufacturer	Country	Year
442010	Fugaku	Fujitsu	Japan	2020
148600	Summit	IBM	United States	2018
94640	Sierra	IBM	United States	2018
93015	Sunway	NRCPC	China	2016
64590	Perlmutter	HPE	United States	2021
63460	Selene	Nvidia	United States	2020
61445	Tianhe-2A	NUDT	China	2013
44120	JUWELS	Atos	Germany	2020
35450	HPC5	Dell EMC	Italy	2020
30050	Voyager-EUS2	Microsoft	United States	2021

a) **[8 marks]** For every set of attributes (that is, for every subset of the attribute set) decide whether you can deduce that it is *not* a candidate key, assuming the shown instance is legal. Justify your answer.

The Candidate key can contain the null value unless the attribute requires non-null, it has the ability to uniquely identifies a row in the table. I'll deduce each attribute below.

Answer:

{Performance}It is the candidate key since it is unique and minimal that has the ability to uniquely identify records in the table.

{Name} It is the candidate key since it is unique and minimal that has the ability to uniquely identify records in the table.

{Manufacturer} It is **NOT** a candidate key since it does not have the ability to uniquely identify a row due to there are duplicate values.

{Country} It is **NOT** a candidate key since it does not have the ability to uniquely identify a row due to there are duplicate values.

{Year} It is **NOT** a candidate key since it does not have the ability to uniquely identify a row due to there are duplicate values.

{Manufacturer, Country} can **NOT** be seen as the candidate key. It's because there are 2 rows contain the same combination of IBM and United States, so it can't uniquely identify a row.

{Manufacturer, Year} can **NOT** be seen as the candidate key. It's because there are 2 rows that contain the same combination of IBM and 2018, so it can't uniquely identify a row.

{Country, Year} can **NOT** be seen as the candidate key. It's because there are 2 rows that contain the same combination of United States and 2018, so it can't uniquely identify a row.

{Performance, Name} can be seen as the candidate key. It is because both Performance and Name are valid candidate keys that can identify a row uniquely.

For other combinations of attributes like {Performance,Year},{Name, Year}, etc, since they all have the candidate key involved, which is {Performance}or {Name}, so they all have the ability to uniquely identify records in the table which means they're all candidate keys.

b) **[4 marks]** For every remaining set of attributes (that is, for every set not ruled out as a candidate key in part **a)**), discuss whether you consider it a suitable candidate key? Justify your answer.

Answer:

{Performance} I do **NOT** think it can be considered as a suitable candidate key. It is because although the performance for each supercomputer row within the given list is unique, we can not guarantee that other supercomputers will also have different performance. If we have the same performance value, then it can not uniquely identify the specified row.

{Name}: I think it can be considered as the suitable candidate key. It is because when a new supercomputer is coming, the country will name it differently to avoid the confusion.

For other candidate keys that is formed as the collection, they are not suitable. A single attribute is always better then a collection of attributes since it removes the redundancy, so these collections of attributes should not be deduced as the suitable candidate key.

c) **[2 marks]** Which of the candidate keys identified in part **b)** would you choose as the primary key? Justify your answer

Answer: I will choose {Name} as the primary key. It is because it will always have unique and non-null values..

d) **[2 mark]** Add a new tuple for a computer into the SUPERCOMPUTERS relation. How would you check that the primary key identified in part **c)** is still valid?

Answer: tuple = {(Performance, 442010), (Name, Tianhe-250), (Manufacturer, NUDT), (Country, China), (Year, 2022)}

We can see that the Name attribute is unique and not null, which means through the Name, we can still uniquely identify the row. However, for other attribute, they all have the duplicate value, such as the performance of my tuple is the same as Fugaku, etc.

e) **[2 mark]** Create a relation that shows for each country in the table above the country and the capital, i.e., use a relation schema with attribute set {Country, Capital}. How many records are in your relation?

Answer:There will be **5** records in my relation. The following table shows all these 5 records/rows of the relation schema {Country, Capital}.

Country	**Capital**
Japan	Tokyo
United States	Washington, D.C.
China	Beijing
Germany	Berlin
Italy	Rome

f) **[2 mark]**. Consider a relation schema with attribute set {Manufacturer, City} and assume that both attributes have a domain with ten values each. What would be the maximum number of records in an instance of this relation schema?

Answer: 10^2 = 100

The maximum number of records in an instance of this relation schema is **100.**

Question 2 [10 marks]

Suppose your software company has developed a relational database for the grocery store "Fruits and more". The underlying database schema contains the following relation schemas:

COMPANY (Cid: STRING, Name: STRING, Location: STRING) with primary key {Cid}

FRUITS (Fid: STRING, Name: STRING, Cid: STRING, InStock: INTEGER, Price: INTEGER) with primary key {Fid, Cid} and foreign key Cid ⊆ COMPANY[Cid]

Below you find instances of these two relation schemas:

FRUITS

COMPANY

Fid	Name	Cid	InStock	Price
557	Apple	23XY	50	21
85520	Pear	A15F	0	78
63311	Pear	FVT35	211	49
36773	Kiwi	23XY	50	21
36773	Kiwi	FVT35	29	22

Cid	Name	Location
23XY	GreatFruits	Wellington
FVT35	Yummy	Wellington
F15A	GreatFruits	Levin
A15F	BetterFruits	Lower Hutt
5AB32	NiceFruits	*null*

Your tasks are as follows.

A. **[5 marks]** Decide which of the following tuples can be added or removed, respectively. *Justify your answers!*

1. Insert tuple (null, 'Tasty', 'Wellington') into COMPANY

Answer: Reject. Cid is the primary key which cannot be null, it violate the not null constraints.

2. Insert tuple ('FVT35', 'SweetFruits', 'Porirua') into COMPANY

Answer: Reject. As the primary key, the value must be different for each record, but 'FVT35' has already exists. It violate the Uniqueness constraints.

3. Delete tuple ('A15F', 'BetterFruits', *null*) from COMPANY

Answer: Reject. This tuple does not match any existing record in the COMPANY table due to the name of Location should be 'Lower Hutt'. Even if the name is correct, it will also be reject since it violate the referential integrity constraint that will let the corresponding FRUITS tuple refer to nothing.

4. Delete tuple ('23XY', 'GreatFruits', 'Wellington') from COMPANY

Answer: Reject. This Cid '23XY' is both exist in COMPANY and FRUITS tables. It is referred to by foregin keys in the FRUITS tuple, so delete it from the COMPANY database will bring the corresponding FRUITS tuple refer to nothing. It violate the referential integrity constraint.

5. Insert tuple (‘XYZ4’, ‘Wellington’, ‘Yummy’) into COMPANY

Answer: Accept. It does not violate any constraints.

b) [5 marks] Decide which of the following tuples can be added or removed, respectively. *Justify your answers!*

1. Delete tuple (‘36773’, ‘Kiwi’, ‘23XY’, 50, 21) from FRUITS

Answer: Accept. It does not violate any constraints.

2. Insert tuple (‘55555’, *null*, ‘F15A’, 2, 99) into FRUITS

Answer:Accept. Name can be set as null since it does not belongs to the primary key, so it does not violate the not null constraints. Also, for the foreign key value ‘F15A’, it match the tuple in the COMPANY relation, so it does not violate the Referential Integrity constraints. In conclusion, It does not violate any constraints.

3. Insert tuple (‘54556’, ‘Lemon’, ‘FV35’, 20, 43) into FRUITS

Answer: Reject. It violate the Referential Integrity constraints in which the value of a foreign key ‘FV35’ do NOT match any tuple in the COMPANY relation.

4. Insert tuple (‘53557’, ‘Apple’, ‘5AB32’, 500, 1) into FRUITS

Answer: Accept. For the foreign key value 5AB32’, it match the tuple in the COMPANY relation, so it does not violate the Referential Integrity constraints. Therefore, it does not violate any constraints.

5. Delete tuple (‘46557’, ‘Apple’, ‘23XY’, 1, 21) from FRUITS

Answer: Reject. This tuple does not match any existing instances in the FRUITS table.

Question 3 [20 marks]

The Wellington Foreign Trade Office needs to translate hundreds of documents every day. To ensure professional translation in a timely manner the office cooperates with several translation agencies and expert translators in New Zealand. The processing of the data about translations as well as the checking of deadlines and quality requirements is time consuming and error prone if this is done manually on paper.

Therefore, the office wants to build a new database to record all relevant data that is needed for processing and checking translations. Suppose the following relation schemas have been proposed to be belong to the database schema for the new database.

TranslationAgency (AgencyNumber, Name) with primary key {AgencyNumber}

Translator (Name, Phone, Field, IRDNumber) with unknown primary key

IsExpert (Name, Language) with primary key {Name, Language}

TranslationOrder (OrderNumber, OrderDate, PageNumber, Budget, FromLanguage, ToLanguage, Deadline) with primary key {OrderNumber}

Assignment (Agent, OrderNumber, Part, Language, Name) with primary key {OrderNumber, Part}

The following additional constraints are known:

Each translator has a unique IRD number, a unique phone and a unique name.

For each translator, the IRDNumber must be specified, while Field may be left blank (if not known).

Translators can be experts in up to four languages.

An agent can assign a translation order to multiple translators who can be distinguished by the assigned part of the order.

Your tasks are as follows:

a) **[3 marks]** For the relation schema Translator, identify all suitable candidate keys. Explain your answer. Which candidate key would you choose as the primary key? Justify your answer.

Answer: {IRDNumber} {Name}{Phone}, these 3 attributes are suitable candidate keys since they are all unique that can be used to identify a specified translator.

I will choose {IRDNumber} as the primary key. It is because the primary key requires unique and not null constraints, for the {IRDNumber}, it is unique and it must be specified for each translator which means not null is guaranteed. However, for {Name} and {Phone}, although they are unique as well, it does not have the additional constraints that they must be specified which means they might be null as well. Even we do not have this constraints, the {IRDNumber} is more credible then others.

So, {IRDNumber} is the chosen only as the primary key.

b) **[5 marks]** For each of the relation schemas, identify all suitable foreign keys (if there are any). Explain your answer.

Answer: For TranslationAgency, there will be no foreign key. It is because Name is only related to TranslationAgency, there is no relationship to Translator and IsExpert. Also, no matter which attribute is used, it will not point to the unique primary key of other relation schemas.

For Translator , there will be no foreign key.It is because {Name} is not a unique candidate/primary key in IsExpert schema due to Translators can be experts in up to four languages. Foreign key should point to the unique key of other relation schema.

For IsExpert , the foreign key is {Name}. It points to the Translator database schema. Although {Name} in Translator is not the primary key, it is a unique candidate key which means {Name} is a suitable foreign key in IsExpert. (P.S. I find in the web that the reference of foreign key can be *primary key or a unique key*, so unique candidate key is acceptable.)

For TranslationOrder, there will be no foreign key.

For Assignment, the suitable foreign key is {OrderNumber} , {Name} and {Agent}. {OrderNumber} points to the primary key of TranslationOrder, {Name} points to the Translator database schema and it is a unique candidate key in Translator, so they are suitable foreign keys. For the {Agent}, by asking the tutor, I obtain that it points to the primary key{AgencyNumber} in the TranslationAgency relation, so it is the foreign key as well.

c) **[2 marks]** For each of the relation schemas, decide which attributes must be declared as not null. Explain your answer.

Answer:

TranslationAgency: {AgencyNumber(NOTNULL), Name}{AgencyNumber} is the primary key which must be declared as **not null**. For {Name}, it is not a primary or foreign key which means the null value is accepted although there should be a name for each TranslationAgency.

Translator : {Name(NOTNULL), Phone, Field, IRDNumber(NOTNULL)}. {IRDNumber} is the primary key which be declared as **not null**. For {Name}, it can **not be null** due to it is both the foreign key and primary key of IsExpert relation schema. It means in IsExpert, Name is always NOTNULL, if Name is null somewhere in Translator, then it will violate the Referential Integrity constraints in which the value of a foreign key in IsName do not match any instance in Translator.

IsExpert: {Name(NOTNULL), Language(NOTNULL)}. As a attribute set, they are primary key which means they must be declared as not null.

TranslationOrder:{OrderNumber(NOTNULL), OrderDate, PageNumber, Budget, FromLanguage, ToLanguage, Deadline}. OrderNumber is the primary key which must be declared as not null.

Assignment:{Agent, OrderNumber(NOTNULL), Part(NOTNULL), Language, Name). The primary key is {OrderNumber, Part}, which means they must be declared as not null and unique.For {Name}, although it is foreign key, the null is acceptable, so it can have null.

d) **[5 marks]** Assume, the translator with name ‘Peter Pan’ in the Translator relation retires. When deleting the record of this translator from the Translator relation, all the assignments made to him should not be lost. How would you ensure this requirement? Explain your answer.

Answer: According to the slide**,** we can use DELETE SET NULL or DELETE SET DEFAULT to ensure this requirement. It will automatically insert null or a default value in the foreign key attributes of tuples in other relation(s) that refer to t. It means whenever delete a translator from Translator relation schema, it will automatically set the corresponding foreign key(i.e. Name with 'Peter Pan' value) in Assignment relation to null that refer to Translator.

e) **[5 marks]** Assume, a translation order with order number '42' in the TranslationOrder relation is cancelled. Suppose, however, that already some assignments have been made to translate parts of this translation order. When deleting the record of the translation order from the TranslationOrder relation, then all the assignments should be deleted, too. How can this requirement be ensured? Explain your answer.

Answer:Like previous question, according to the slide,we can use DELETE CASCADE to ensure this requirement. It delete tuples in other relation(s) that refer to t, which means that when declare the attribute constraints for the foreign key OrderNumber in relation Assignment, and add constraints of on DELETE CASCADE. Thus, when delete the record from the TranslationOrder relation, it will automatically delete all the corresponding records with the same foreign key value (i.e. orderNumber: '42') in Assignment.

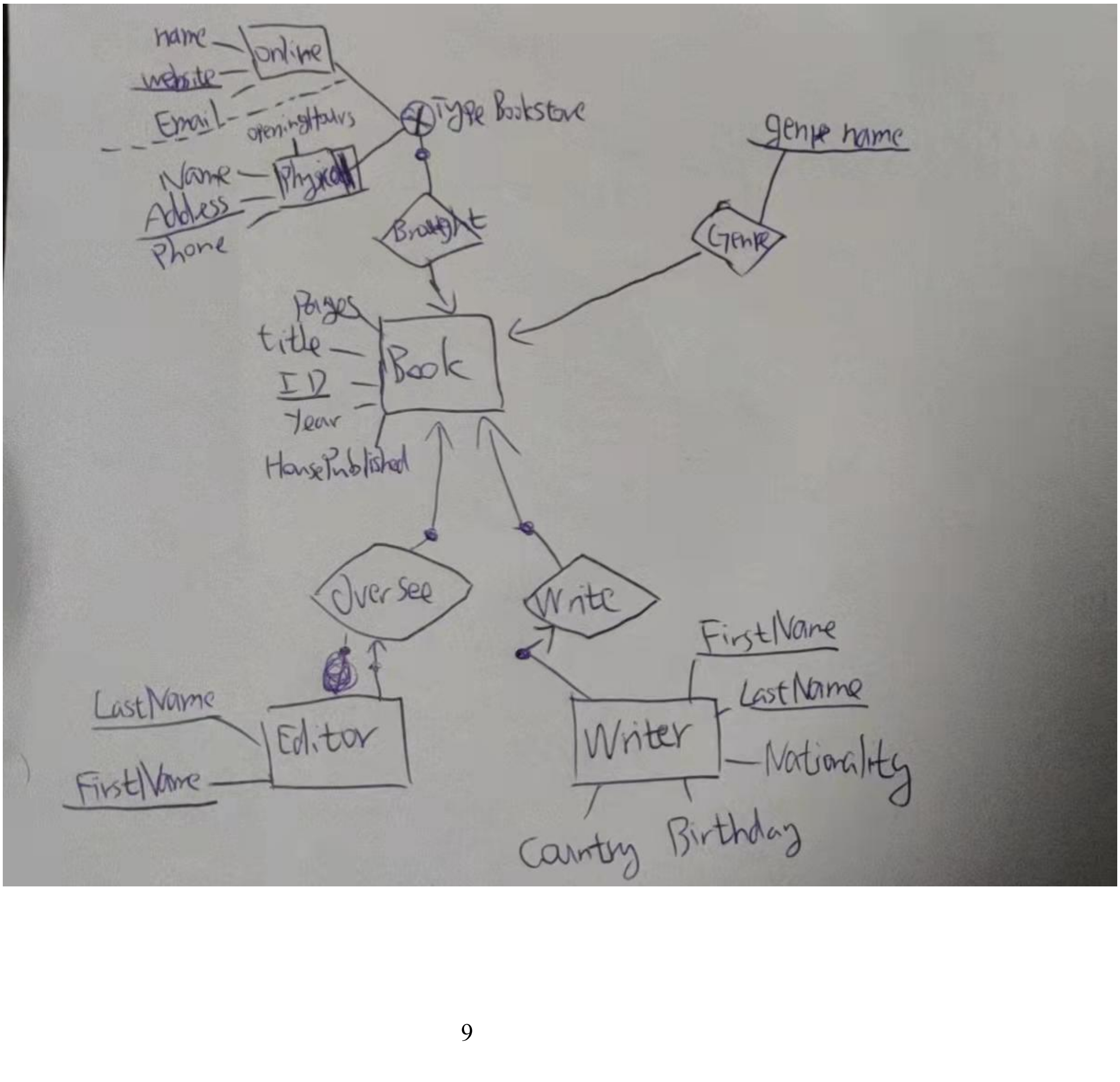
Question 4 [30 marks]

You are asked to design a new database for your grandma's collection of books. A book has a title, a release year, a unique international standard book number, a number of pages and was published by a certain publishing house.

A book can have one or more authors. The authors of a book are writers. A writer has a first name, a last name, a birthday, a nationality, and a home country. A book can have one or more editors. An editor has a first name, a last name. Editors oversee the emergence of a book from the first manuscript to the print-ready form. A book without authors has at least one editor.

Furthermore, your grandma buys books at certain bookstores which are either physical ones or online ones. Physical bookstores have a name, an address, a contact phone number and opening hours, while online bookstores have a name, a website and a contact email address. There are different genres such as adventure, comedy, crime, mystery, fantasy or science fiction. For every book at most two genres should be recorded in the database.

a) **[24 marks]** Draw an extended ER diagram for the database above. Write down the corresponding extended ER schema, including declarations of all the entity types (showing attributes and keys) and relationship types (showing components, attributes and keys).



Entity type: level 0. Written in E = (attr(E), id(E)) with attr(E) the set of attributes and id(E) the primary key of E

- Book = ({title, year, id, pages, HousePublished },{id})
- Writer=({first name, last name, birthday, country},{first name, last name})
- Editor = ({first name, last name},{first name, last name})
- physical bookstore = ({name, address,phone, openingHours},{name})
- online bookstore = ({name, website, email,genre},{name,genre})

Relation type: level 1. Written in R = (comp(R), attr(R),id(R)}with attr(R) the set of attributes and id(R) the primary key of R and comp(R) the component of R

- OverSee =({Editor, Book, Writer},{emergency},{Editor, Book})
- Write = {(Writer, Book),{},{Writer, Book}}
- Brought = ({Book, Oversee,Write}{}{Book, Oversee,Write})

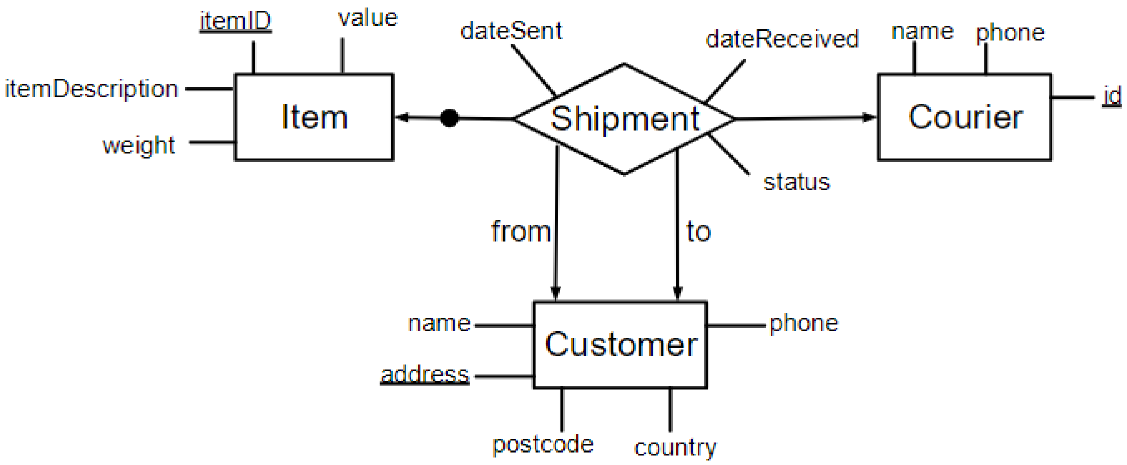
b) **[6 marks]** There may be information, requirements or integrity constraints that you are not able to represent in your diagram. Give three examples of integrity constraints that have not been represented in your diagram.

Remark**:** Whenever you feel that information is missing in the problem description above, add an assumption and make your assumption explicit. In practice you would consult the domain experts or potential users for clarification.

Answer:

Question 5 [20 marks]

Consider the following extended ER diagram:



a) **[5 marks]** Present the extended ER schema of the extended ER diagram above.

Entity type: level 0

- Item
  - attributes: {itemID, value, itemDescription, weight}
  - primary key: itemID
- Customer
  - attributes: {name, phone, address, postcode, country}
  - primary key: address
- Courier
  - attributes: {name, phone, id}
  - primary key: id

Relationship Types: level 1

Relationship type **Shipment**

- with component set{Item, Courier, Customer},
- attribute set{dataSent, dataReceived, status},
- primary key{Item }

b) **[10 marks]** Transform your extended ER schema into a relational database schema. In particular, list all the relation schemas in your relational database schema. For each relation schema, list all attributes, the primary key, the NOT NULL constraints, and the foreign keys.

- Item:
  - attributes: {itemID, value, itemDescription, weight}
  - primary key: {itemID}
  - foreign key: no foreign key
  - NOT NULL constraints: itemID
- Customer:
  - attributes: {name, phone, address, postcode, country}
  - primary key: {address}

- foreign key:no foreign key
- NOT NULL constraints: address

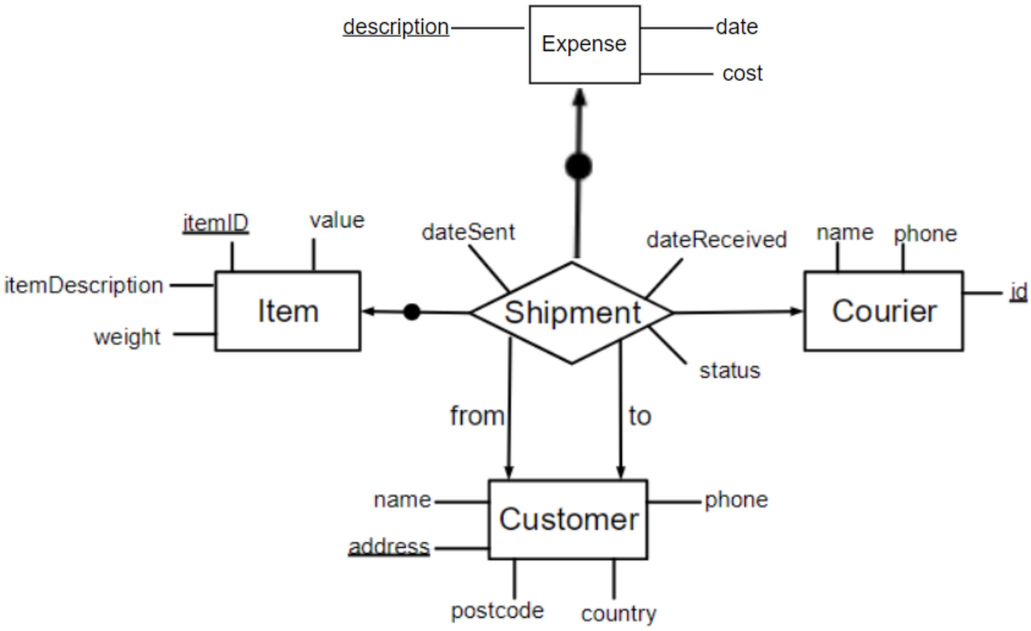
- Courier:
  - attributes: {name, phone, <u>id</u>}
  - primary key: {id}
  - foreign key: no foreign key
  - NOTNULL constraints: id
- Shipment
  - attribute set{dataSent, dataReceived, status},
  - primary key:{Item}
  - Foreign keys:
    - {itemID ⊆ Item[itemID], Courier_id⊆Courier[id], from_Customer_address ⊆ Customer[address], to_customer_address⊆ Customer[address] }
  - NOTNULL constraints: {itemID⊆ Item}

c) **[5 marks]** We also want to record information about expenses related to shipments. Each related expense has a date, a cost, and a description. The related expenses of a given shipment can be uniquely identified by the description.

Enhance the given extended ER diagram to reflect this additional information. Present the extended ER diagram with your proposed enhancements. Justify your answer.

Answer:

Expense = ({date, cost <u>description</u>},{description}).
An entity type Expense with a set of attributes{date, cost <u>description</u>} and a primary key {description}



****