School of Engineering and Computer Science

SWEN 304 **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, 28 March, 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.

Answer:

First, we can check the individual attributes to see if any of them are eligible to be a candidate key. Performance is unique at first, but as supercomputers can change in performance overtime (technological upgrades, system updates, degradation, etc.) we can deduct that performance cannot be part of the candidate key.

Instead, we have the name, which is unique in this data table. As names will not change over time and are unique for the lifetime of the supercomputer, we can deduce that the name could be the candidate key.

Manufacturer cannot be the candidate key as it is not unique. We can see in the table that IBM has two systems.

Country is ineligible to be the candidate key as multiple systems come from the same country, five from the US and two from China.

Year cannot be the candidate key as there are multiple supercomputers from the same year.

Second, we can try combinations of two or more attributes like manufacturer & country. This combination is particularly useless as we can see there are two IBM systems coming from the US. Manufacturer & year is also redundant as there are two IBM systems from the year 2018. Likewise, country + year is ineligible as there are two systems from the US in 2018.

We could rather potentially look at combinations which include the name (which is already a unique identifier) with another attribute such as manufacturer, country or year. However, this is redundant as the name attribute itself is already unique, so we can rule those out.

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:

In the given instance, each supercomputer has a unique name. We can see that not only are none repeated, but {name} is also a minimal subset as we cannot remove any attributes from it. Semantically, supercomputer names are already intended to be unique identifiers in the real world (e.g., model names).

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

Answer:

I would choose the candidate key consisting solely of {name} as the primary key because it uniquely identifies each tuple in the relation, is minimal, and simple to use.

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:

Performance:	13499
Name:	Phlop
Manufacturer:	PCBang
Country:	NewZealand
Year:	2025

Tuple: (13499, Phlop, PCBang, New Zealand, 2025)

To check that the primary key {name} is still valid, I would verify that the new supercomputer's name does not already exist in the current data. If the name ("Phlop") is unique, then the primary key constraint remains valid (which it does).

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? Justify your answer.

Answer:

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

The relation {Country, Capital} will contain five records, as there are five unique countries in the table: China, Germany, Italy, Japan, and the US. Each is associated with its capital city in the new relation.

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:

We are given the relation R(Manufacturer, City) with ten distinct values each in both attributes. Assuming that we have no constraints such as primary keys or uniqueness and the relation is allowed to contain every possible pairing of {Manufacturer} and {City}, we have a cartesian product situation.

Therefore in relational theory, we have 10 values of {Manufacturer} and {City}, meaning that the maximum number of possible pairings is 100 (10*10).

$|r(R)| = |DOM(Manufacturer)| \times |DOM(City)| = 10 \times 10 = 100$

We can see that with two attributes with domains of ten values each, we have a total number of one hundred records in an instance of this particular relation schema.

Question 2 [10 marks]

Your university is using a relational database to manage its data on students and their study performance. Suppose the underlying database schema includes the following two relation schemas:

- STUDENT (SID: STRING, Name: STRING, DoB: DATE, GPA: REAL) with primary key {SID}
- RESULTS (CourId: STRING, StudId: STRING, Grade: STRING) with primary key {StudId, CourID} and foreign key StudID $\subseteq$ STUDENT[SID]

Below you find instances of these two relation schemas:

RESULTS

CourID	StudID	Grade
SWEN102	53666	A
SWEN221	53688	B
SWEN301	53832	B
SWEN224	53650	C

STUDENT

SID	Name	DoB	GPA
50000	Dave	22/01/1985	3.3
53666	Jones	11/02/1986	3.4
53688	Smith	22/01/1985	3.2
53650	Smith	15/05/1986	3.8
53831	Mathew	16/06/1984	1.8
53832	Jack	25/11/1983	3.0

Your tasks are as follows. **Justify your answers**!

a) [5 marks] Decide which of the following tuples can be inserted into the given instances.

1. Insert tuple (‘53688’, ‘Mike’, ‘16/06/1985’, 3.4) into STUDENT

   Answer:

   This insert would be rejected as it violates the primary key constraint on SID (53688 SID is already attached to Smith).

2. Insert tuple (*null*, ‘Mike’, ‘16/06/1985’, 3.4) into STUDENT

   Answer:

   This insert would also be rejected as SID is the primary key and it cannot have a value of null. This violates the not-null constraint on a primary key.

3. Insert tuple (‘SWEN224’, ‘53650’, ‘B’) into RESULTS

   Answer:

   This tuple cannot be inserted as the RESULTS instance already contains the primary key {53650, SWEN224}.

4. Insert tuple (‘SWEN102’, ‘50505’, ‘B’) into RESULTS

   Answer:

   This tuple would also be rejected as the SID 50505 does not exist within the STUDENT instance.

5. Insert tuple (‘SWEN222’, *null*, ‘B’) into RESULTS

   Answer:

   Lastly this would also be rejected as it has a null value in place of the {StudID}, which doesn’t meet the foreign key constraints.

b) [5 marks] Decide which of the following tuples can be deleted from the given instances.

1. Delete tuple (‘53831’, ‘Matt’, ‘16/06/1984’, 1.8) from STUDENT

   Answer:

   This tuple cannot be deleted from the instance as it does not exist. The name given in the tuple here is ‘Matt’, which does not match the name in the instance (‘Mathew’). The deletion would fail due to the mismatch.

2. Delete tuple (‘53688’, ‘Smith’, ‘22/01/1985’, 3.2) from STUDENT

   Answer:

   This would also fail as the SID ‘53688’ has an entry in the RESULTS instance. This would be a foreign key violation, as deletion would cause the corresponding RESULTS entry to point to a non-existent StudID.

3. Delete tuple (‘50000’, ‘Dave’, ‘22/01/1985’, 3.3) from STUDENT

   Answer:

   This deletion would be allowed as SID ‘50000’ is not referenced in RESULTS, and the tuple is the exact same as in the STUDENT instance.

4. Delete tuple (‘SWEN301’, ‘53832’, ‘B’) from RESULTS

   Answer:

   This tuple can be deleted from RESULTS as it is a valid row in RESULTS and deleting it does not affect any constraints in the STUDENT instance.

5. Delete tuple (‘SWEN301’, ‘53832’, *null*) from RESULTS

   Answer:

   This tuple cannot be deleted from RESULTS as it does not exist in the first place. There is a row (‘SWEN301’, ‘53832’, ‘B’), but as the grade value does not match, it cannot be deleted.

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:

1. Each translator has a unique IRD number, a unique phone and a unique name.
2. For each translator, the IRDNumber must be specified, while Field may be left blank (if not known).
3. Translators can be experts in up to four languages.
4. 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:

Here we have the relation schema Translator(Name, Phone, Field, IRDNumber). Given that each translator has a unique IRD number, phone, and name, we can consider those to be suitable candidate keys. As field can be left blank if unknown, we can rule field out from being the primary key.

As for which key I would choose as the primary key, I must choose IRD number, as it is the most stable, system-assigned, and very unlikely to change, making it the most suitable candidate number.

As more translators are hired in the future, there could be multiple translators with the same name, and although phone numbers are generally unique, people do change phone numbers due to many potential reasons (e.g., lost sim, changing provider, etc.).

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

Answer:

TranslationAgency(AgencyNumber, Name):

I can assume that it is a top-level entity as it does not depend on any other relation schema and does not reference any other schema via a foreign key. As such I can identify no suitable foreign keys here, as there is nothing it needs to reference from other instances.

Translator(Name, Phone, Field, IRDNumber):

We have chosen IRDNumber as the primary key, and as translators are individual entities in this instance, they are not dependent on others, therefore there are no foreign keys in this schema.

IsExpert(Name, Language):

Seems to reference `Name` from Translator, and as we know that we have the current constraint of names being unique, we can reference it as a valid foreign key.

Foreign Key: Name ⊆ Translator[Name]

TranslationOrder(OrderNumber, OrderDate, PageNumber, Budget, FromLanguage, ToLanguage, Deadline):

TranslationOrder also seems to be a top-level entity, with no dependencies on other relation schemas and no references to foreign keys as it shares no attributes with other relations. No suitable foreign keys here.

Assignment(Agent, OrderNumber, Part, Language, Name):

Likely references OrderNumber from TranslationOrder. Name and Language could be referenced from IsExpert, however we do not know if assignments are assigned strictly to translators who are experts in the relevant language (e.g., can an assignment for an English to Chinese assignment be assigned to a translator who knows both English and Chinese, but is only an expert in English, or must they be an expert in both languages.) It is not explicitly stated that there is a constraint that only expert translators may be assigned translation assignments in their relevant languages.

It is possible that Assignment rather uses Translator as a reference for Name, as we know that they must assign an assignment to a translator. It is more likely that it uses Translator[Name] as the foreign key for the Name attribute.

Foreign Keys: OrderNumber ⊆ TranslationOrder[OrderNumber], OrderNumber ⊆ Translator[Name]

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

Answer:

TranslationAgency(AgencyNumber, Name):

As the primary key here is {AgencyNumber}, it must not be null. There is no mention of whether Name can be null, but following logic, we should assume that an agency name is required, as agencies would normally have a name in real life.

Translator(Name, Phone, Field, IRDNumber):

As answered before, {IRDNumber} is the most suitable primary key and thus cannot be null. We know that there is a constraint that makes Translator `Name` and `Phone` unique, so those must be declared as not null too. As `Field` may be left blank, we can safely assume that it can contain null values.

IsExpert(Name, Language):

We know that the primary key for this is {Name, Language}, and as such this schema cannot have null attributes.

TranslationOrder(OrderNumber, OrderDate, PageNumber, Budget, FromLanguage, ToLanguage, Deadline):

Here, the primary key is known to be {OrderNumber}. We can reasonably assume that `OrderDate` cannot be null as a date should be required to process and store orders for easy searching and referencing.

`PageNumber` likely refers to the number of a page or a range of page numbers in a document that requires translation and therefore must also not be null.

`Budget` is important within business and likely must be declared as not null.

`FromLanguage` and `ToLanguage` are obvious attributes which must be declared as not null, as how can a translator know what their job is if they do not know what language to translate to.

`Deadline` is also important and as such should not be left null.

Assignment(Agent, OrderNumber, Part, Language, Name):

As the primary key here is {OrderNumber, Part}, both attributes cannot be null.

As mentioned before, `Name` likely references to Translator, meaning it should not be null.

`Language` should not be null as it is logically required for knowing what translator to assign this to.

`Name` refers to Translator and therefore should not be null.

`Agent` is a tricky one, but I think it could be argued that it can be left null, as I am unable to point out as to where `Agent` may refer to. It is not explicitly stated that it is referenced from TranslationAgency.

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:

The issue here is that as Translator is referenced by Assignment[Name] as a foreign key, if we delete ‘Peter Pan’ from Translator, then the foreign key in Assignment would refer to a non-existent translator, violating referential integrity. Therefore, we need to define a rule to handle the problem. As assignments should not be lost, then we cannot cascade delete the assignments, and instead preserve them.

First, we could remove the foreign key constraint and not use `Name` as a foreign key. This would allow us to keep the name attribute filled with a non-null value (in this case, even after deleting ‘Peter Pan’, all previous assignments they worked on would keep the name attribute as ‘Peter Pan’).

However, if we want to keep `Name` as a foreign key, then we must use `ON DELETE SET NULL`. This would allow us to keep `Name` as a foreign key and delete Translators, while keeping Assignments intact. The only downside to this solution is that now Assignments will have a null value for the `Name` attribute but allows the rest of the assignment data (agent, order, part, language) to remain intact.

FOREIGN KEY (Name) ⊆ Translator(Name) ON DELETE SET NULL

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:

As we want all related assignments to the TranslationOrder to be deleted automatically, we should use `ON DELETE CASCADE`. This means that when a row in TranslationOrder is deleted (here it’s 42), all rows in Assignment with the matching `OrderNumber` are automatically deleted by the DBMS. This will prevent orphaned assignments, data consistency, and matches business logic (if a translation order is cancelled, its parts become irrelevant and should be removed).

FOREIGN KEY (OrderNumber) ⊆ TranslationOrder(Order Number) ON DELETE CASCADE

Question 4 [30 marks]

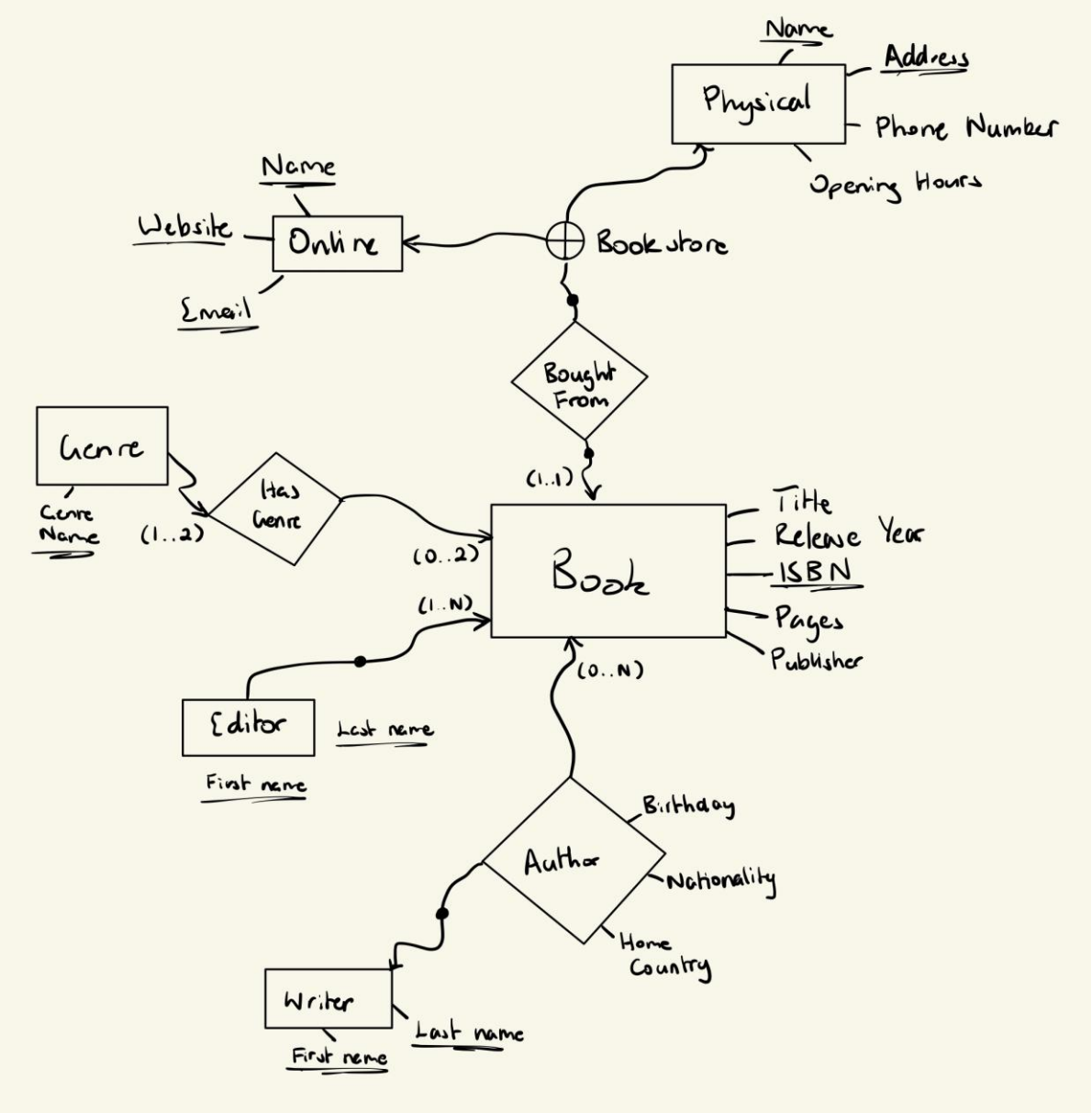
You are asked to design a new database for your grandma's collection of books. A book has a title, a release year, a unique international standard book number, a number of pages and was published by a certain publishing house. For each book, your grandma only bought one copy.

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).

Answer:


Name
Address
Physical
Phone Number
Opening Hours
Name
Website
Online
Bookstore
Email
Bought From
Genre
Genre Name
(1..2)
Has Genre
(1..1)
(0..2)
Book
Title
Release Year
ISBN
Pages
Publisher
(1..N)
(0..N)
Editor
Last name
First name
Birthday
Author
Nationality
Home Country
Writer
Last name
First name

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.

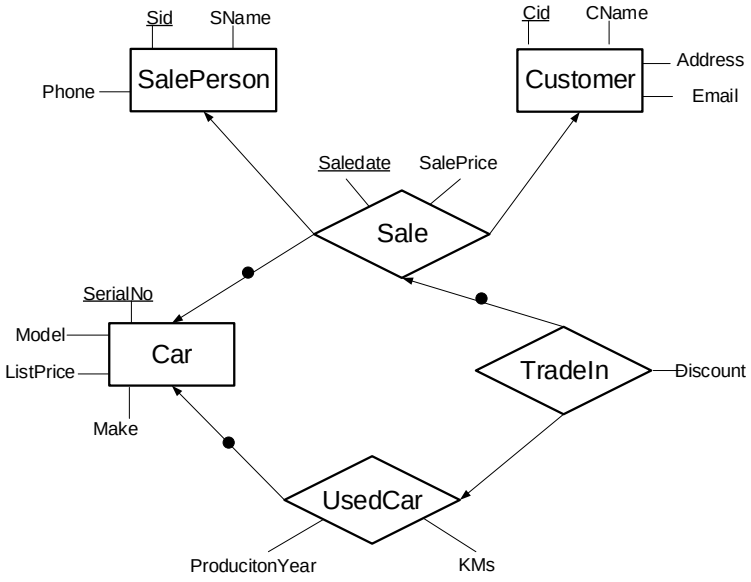
Answer:
In my diagram, I am unable to represent the relationship between editors and authors. The constraint specifies that a book without authors has at least one editor, but I can only represent that there is at least one editor (1..N) regardless of the number of authors (0..N). We cannot see in the diagram that if a book has a null author, it must then have at least one editor.

Looking at this diagram, we have to assume that a writer cannot be both an author and an editor at the same time. While it is possible for an author to also edit their own book in the real world, we cannot assume this possibility at all when looking at the diagram.

Finally, the constraint states that grandma only bought one copy of each book. Although this is represented with a (1..1) cardinality, the diagram as a whole does not show whether a book is sold at multiple stores.

Question 5 [20 marks]

Consider the following extended ER diagram:

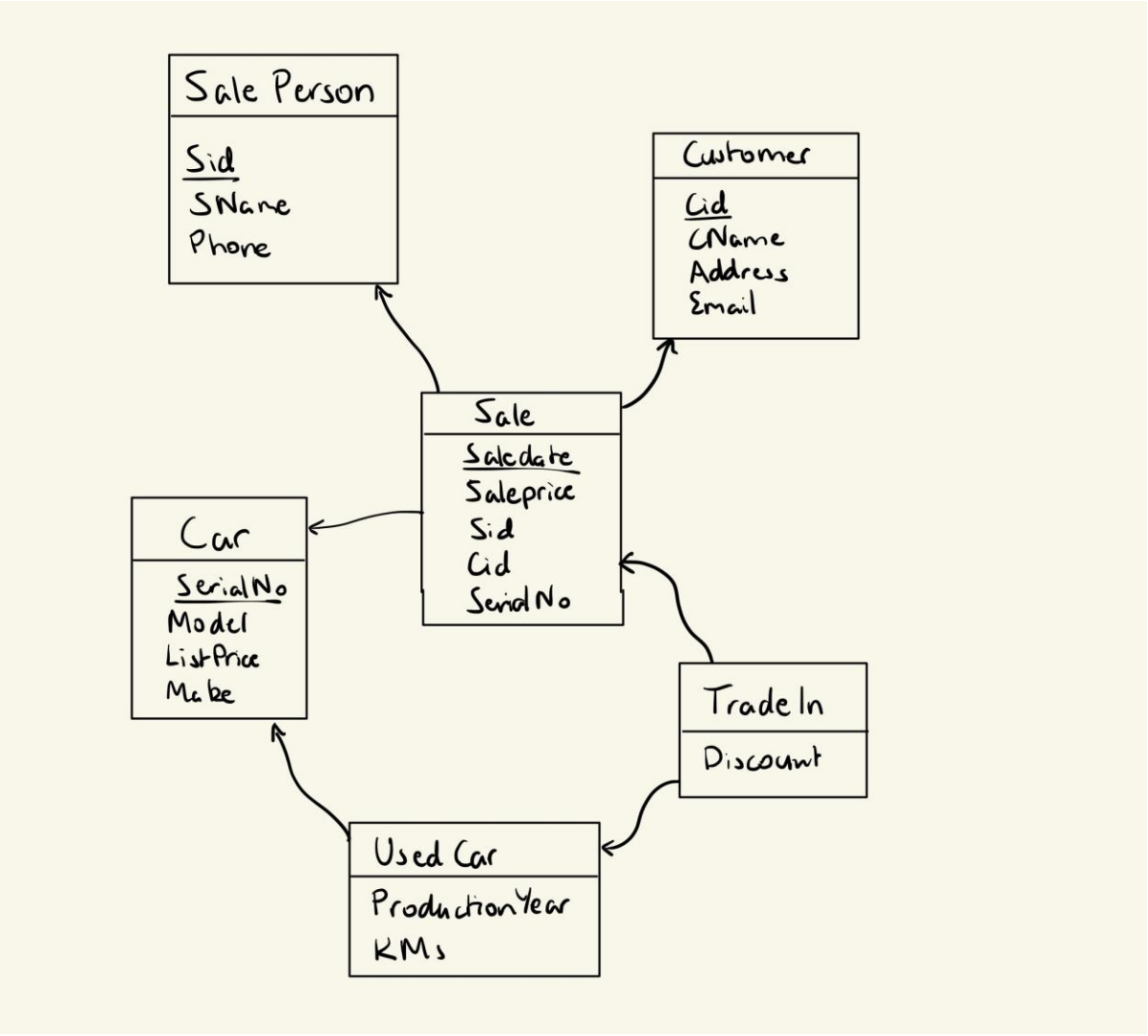


a) [5 marks] Present the extended ER schema of the extended ER diagram above.

Answer:
SalePerson(Sid, SName, Phone) with primary key {Sid}
Customer(Cid, CName, Address, Email) with primary key {Cid}
Car(SerialNo, Make, Model, ListPrice) with primary key {SerialNo}
UsedCar(ProductionYear, KMs)
TradeIn(Discount)
Sale(SaleDate, SalePrice, Sid, Cid, SerialNo) with primary key {Saledate}

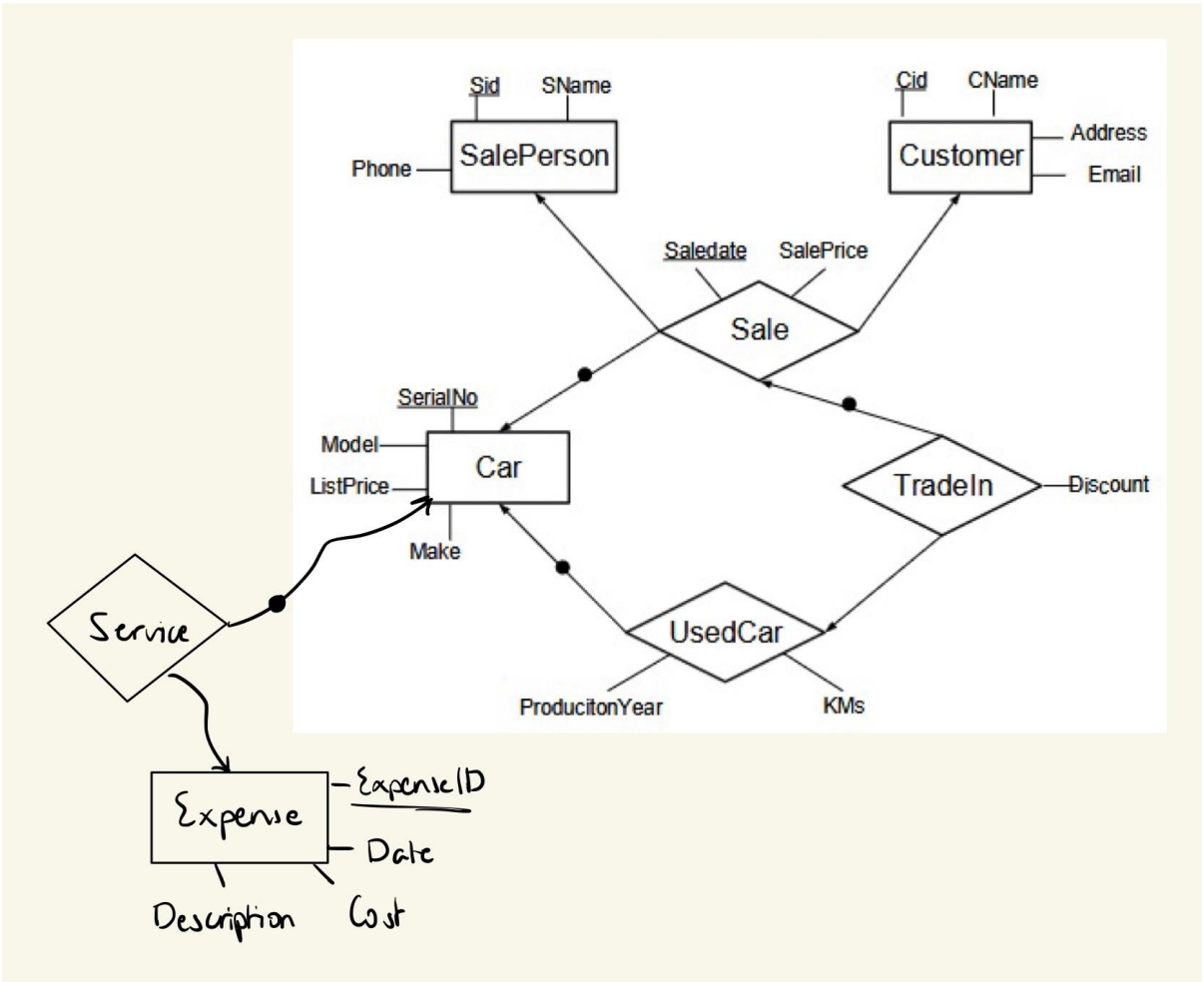
b) [10 marks] Transform your extended ER schema into a relational database schema. In particular, list all the relation schemas in your relational database schema. For each relation schema, list all attributes, the primary key, the NOT NULL constraints, and the foreign keys.

Answer:



c) [5 marks] We also want to record information about expenses related to services for all the cars sold by the company. Each related expense has a date, a cost, and a 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:



Here I have added Expense(ExpenseID, Date, Cost, Description) as its own entity, with ExpenseID as the primary key to uniquely identify each recorded expense. Since service expenses vary by type, timing, cost, and the specific car model involved, it is appropriate to model them as a separate entity.

I connected Expense to Car using a new relationship Service, which captures the association between cars and their related service expenses. This relationship shows that each expense must be associated with exactly one car, while a car may incur multiple expenses over time.

****