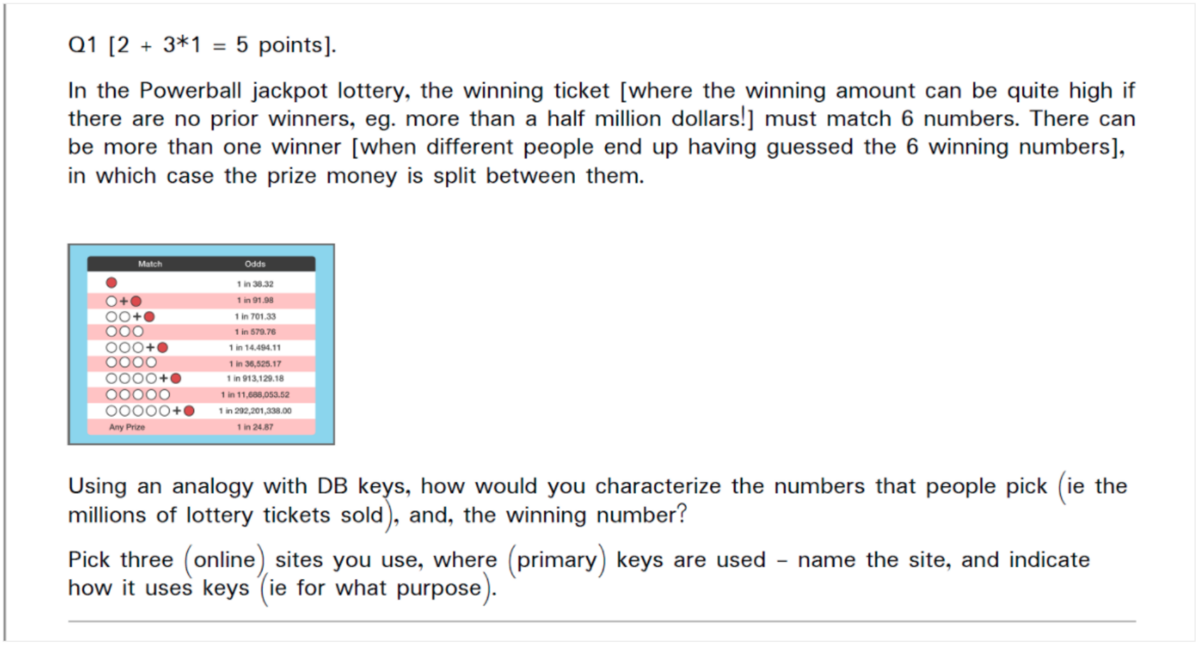
Midterm Rubrics:-

Q1 -

Q1 (5 points)



Q1 [2 + 3*1 = 5 points].

In the Powerball jackpot lottery, the winning ticket [where the winning amount can be quite high if there are no prior winners, eg. more than a half million dollars!] must match 6 numbers. There can be more than one winner [when different people end up having guessed the 6 winning numbers], in which case the prize money is split between them.

Match	Odds
●	1 in 38.32
○+●	1 in 91.98
○○+●	1 in 701.33
○○○	1 in 579.76
○○○+●	1 in 14,494.11
○○○○	1 in 36,525.17
○○○○+●	1 in 913,129.18
○○○○○	1 in 11,688,053.52
○○○○○+●	1 in 292,201,338.00
Any Prize	1 in 24.87

Using an analogy with DB keys, how would you characterize the numbers that people pick (ie the millions of lottery tickets sold), and, the winning number?

Pick three (online) sites you use, where (primary) keys are used - name the site, and indicate how it uses keys (ie for what purpose).

Answer part 1: any reasonable answer can be correct, we show an example interpretation as below

(1 point)
The number people pick will be stored in 6 different columns [**secondary keys - user table]** to query 6 **primary keys [lottery table]** to form 6 matched numbers as columns [**secondary keys**] in an information/winning table.
E.g. the tb_usr_input table as below (no need for students to write a table).

(1 point)
The winning number can be characterized as 6 values queried by 6 positional keys.
E.g. the value in the lottery number column of the tb_lottery table (no need for students to write a table).

Example: assume the lottery number is '037625'
Table 1: tb_usr_input

UID	Digit_1	Digit_2	...	Digit_6
6789	9	0	...	6
56830	2	9	...	5

Table 2: tb_lottery

Nth_Digit	Lottery_Number
0	0
1	3
2	7
3	6
4	2
5	5

Queried results:

PlayerID	Match1	Match2	Match3	Match4	Match5	Match6	Total	Match_Winner
6789	1	0	0	0	1	1	2	no
56830	1	1	1	1	1	1	6	yes

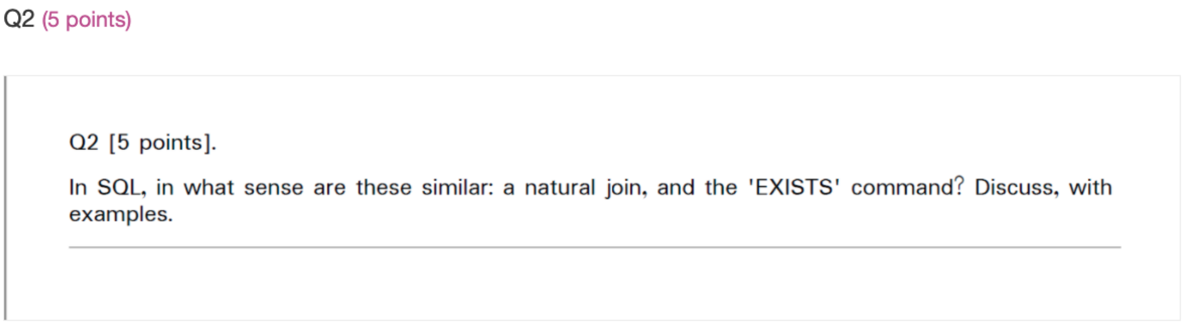
……

[Saty] This simpler answer is fine, too:
All the numbers form a set of **candidate keys**, loosely speaking (even if multiple people pick a number!). The winning number (even if multiple people pick it) is eqvt to a **primary key** - because it would be in the list of candidate keys if someone or a group of people picked it.

Answer part 2: Any reasonable answers work.
For example:
(1 point)
Amazon eCommerce website: primary keys are used to manage the account, create orders, track orders
(1 point)
Facebook: primary keys are used to define the user feed based on likes and dislikes, build a friend network - mutual friends
(1 point)
GSuite account: primary keys are used to identify the user and allow them to use/manage Google services like Gmail, Google Drive, Google Photos

[Saty] More examples: eBay auctions, LinkedIn profiles, bit.ly etc URL shorteners, YouTube video IDs, Medium.com article IDs, etc.

Q2:-



Q2 (5 points)

Q2 [5 points].

In SQL, in what sense are these similar: a natural join, and the 'EXISTS' command? Discuss, with examples.

1.(2 points) use English sentence to describe the similarity
Both natural join and “EXISTS” operator filter the common attributes on two tables.

2 (2 points) provide an example of tables,
Consider, the two tables Customer and Agent

Customer Table

CUS_CODE	CUS_NAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1231242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	37134	231

Agent Table

AGENT_CODE	AGENT_PHONE

125	615223211
167	615223299
231	615223200
333	615223288

3 (1 point) write correct SQL to show the similarity

Using natural JOIN,
SELECT * FROM Customer NATURAL JOIN Agent

Using EXISTS
SELECT * FROM Customer
WHERE EXISTS (SELECT * FROM Agent WHERE Customer.AGENT_CODE =
Agent.AGENT_CODE)

Outcome: Both the queries will produce the same result.

Q3:-

Q3 (5 points)

Q3 [5 points].

Between two entities, a **strong** relationship, paradoxically, can lead to a **weak** entity :) Explain the meaning of 'strong' and 'weak' in this context.

Between two entities, a strong relationship can lead to a weak entity.

An entity is considered weak when its existence is dependent on another entity. In other words, it is weak when its primary key is completely or partially derived from another entity's primary key.

A strong relationship implies that the child entity has a primary key involving the parent's primary key.

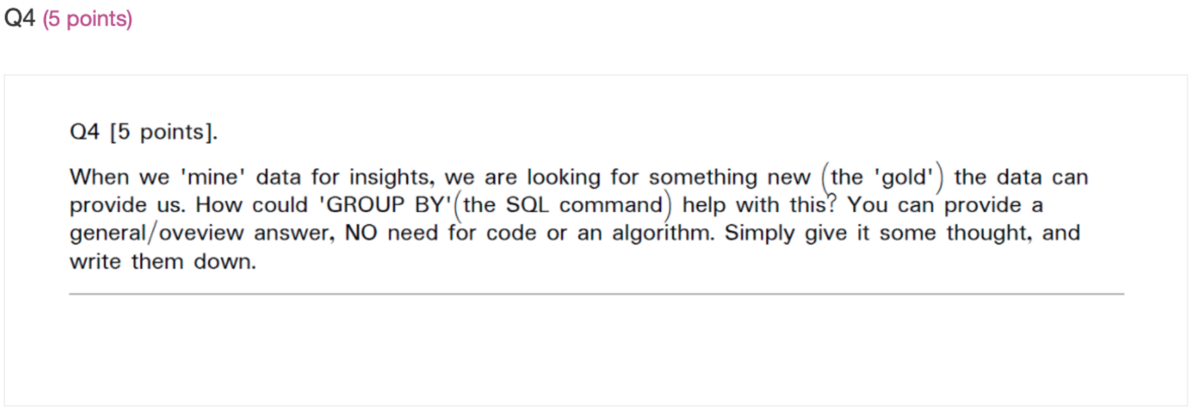
Hence, Strong relationship between two entities would lead to a weak entity.

Rubric:

- -3 If only one part of the answer is correct.

- -5 if the explanation of both Strong relationship and weak entity is wrong.

Q4: -



Q4 (5 points)

Q4 [5 points].

When we 'mine' data for insights, we are looking for something new (the 'gold') the data can provide us. How could 'GROUP BY'(the SQL command) help with this? You can provide a general/oveview answer, NO need for code or an algorithm. Simply give it some thought, and write them down.

The GROUP BY Statement in SQL is used to aggregate identical data into groups and calculate simple descriptive statistics. It provides a way for classification of data based on particular attributes. It is very good at summarising, transforming, filtering, and a few other very essential data analysis tasks.

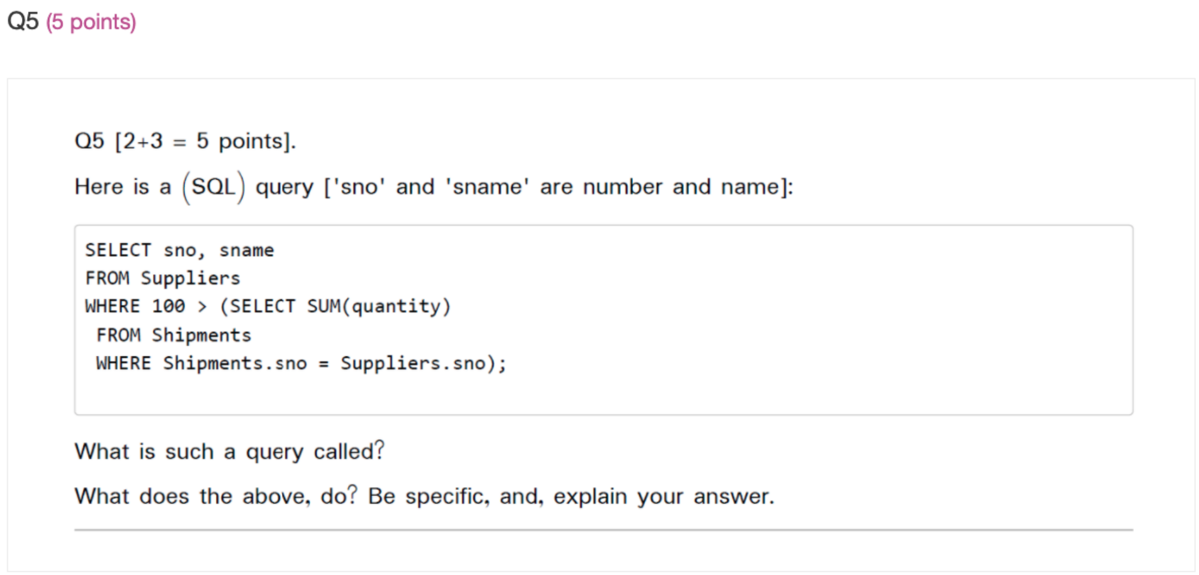
Also include any real world example that helps reaffirm the idea.

Rubric:

- -5 if GROUP BY description is wrong
- -3 if the relationship between GROUP BY and data mining is missing.
- -2 if only an example is described instead of general answer

[Saty] Mentioning that GROUP BY is in essence, ‘itemizing’ will also be a correct answer - by itemizing, we are able to ‘drill down’ into data.

Q5:-



Q5 (5 points)

Q5 [2+3 = 5 points].

Here is a (SQL) query ['sno' and 'sname' are number and name]:

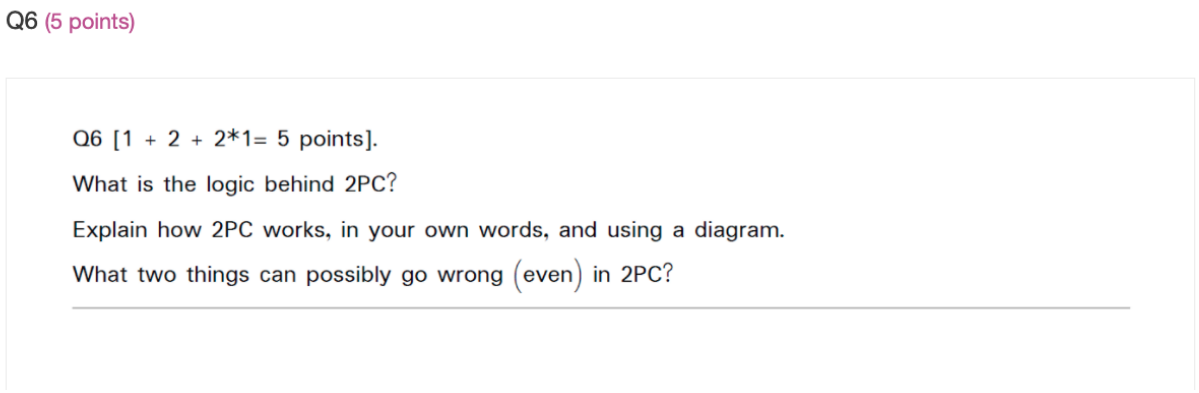
```
SELECT sno, sname
FROM Suppliers
WHERE 100 > (SELECT SUM(quantity)
 FROM Shipments
 WHERE Shipments.sno = Suppliers.sno);
```

What is such a query called?

What does the above, do? Be specific, and, explain your answer.

1. Query with subquery [Saty: even just mentioning ‘subquery’ or ‘inner query’ is fine]
2. - Get the sum of quantity we call it as “Sum” under the condition that the number in shipments and in suppliers are equal.
   - Then we get the number and name from the Supplier table under the condition that “Sum” < 100 from shipments table.
   [Saty: in other words, the query finds the suppliers from whom we have ordered 100 units or less]

Q6:-



Q6 (5 points)

Q6 [1 + 2 + 2*1= 5 points].

What is the logic behind 2PC?

Explain how 2PC works, in your own words, and using a diagram.

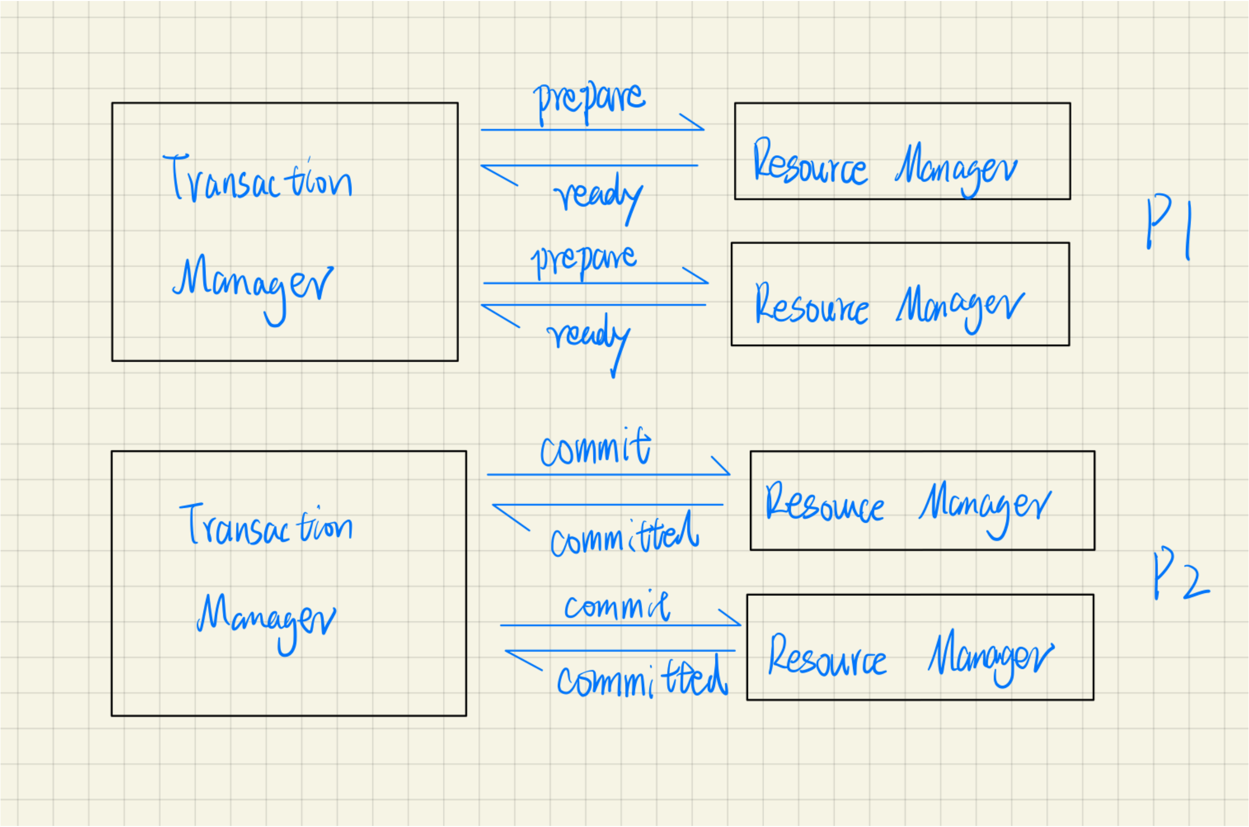
What two things can possibly go wrong (even) in 2PC?

The logic is to make the commit as 2 phases:

1. Preparation stage(voting)

2. Commit stage(execution)

A request comes through the coordinator, sends a ‘Prepare’ message to each participant, executes the local script but does not commit the transaction. If the coordinator receives the participant’s failure message or timeout, it directly sends a rollback message to each participant; otherwise, sends a commit message; the participant performs the commit or rollback operation according to the coordinator’s instructions, releasing all the resources occupied in the process of transaction processing. 2PC has achieved all operations either successfully or completely failed.



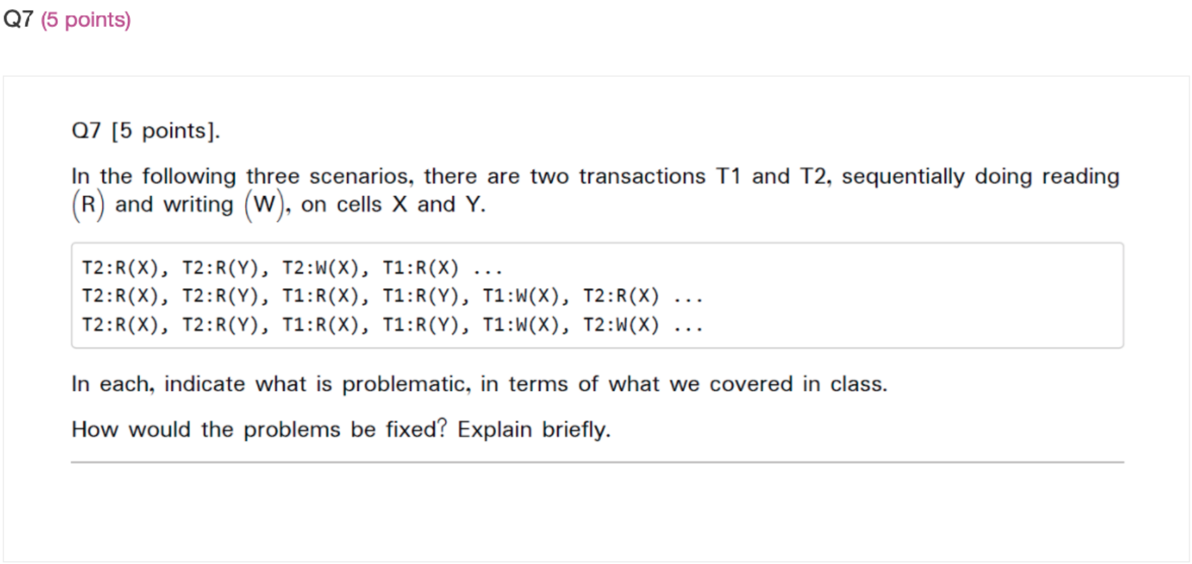
Shortcomings:

- The greatest disadvantage of the two-phase commit protocol is that it is a blocking protocol. If the coordinator fails permanently, some participants will never resolve their transaction. After a participant has sent an agreement message to the coordinator, it will block until a commit or rollback is received.
- There is inconsistency among data when there is network delay(jitter)

[Saty] The logic is that we minimize transaction failure by making sure first that a (distributed) transaction can indeed be carried out for real.

As for failures - the coordinator can fail, after sending out the first set of broadcast messages (query phase) but before, or while, sending out the commit messages. Also, a node can fail after responding with a ‘yes’, but before actually committing.

Q7:-



Q7 (5 points)

Q7 [5 points].

In the following three scenarios, there are two transactions T1 and T2, sequentially doing reading (R) and writing (W), on cells X and Y.

```
T2:R(X), T2:R(Y), T2:W(X), T1:R(X) ...
T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), T2:R(X) ...
T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), T2:W(X) ...
```

In each, indicate what is problematic, in terms of what we covered in class.

How would the problems be fixed? Explain briefly.

7)

1st Scenario (write read conflict) - 1 pt
Dirty Read – If T2 for any reason fails to commit & is rolled back, T1 that has already read the value of X will have inconsistent data.

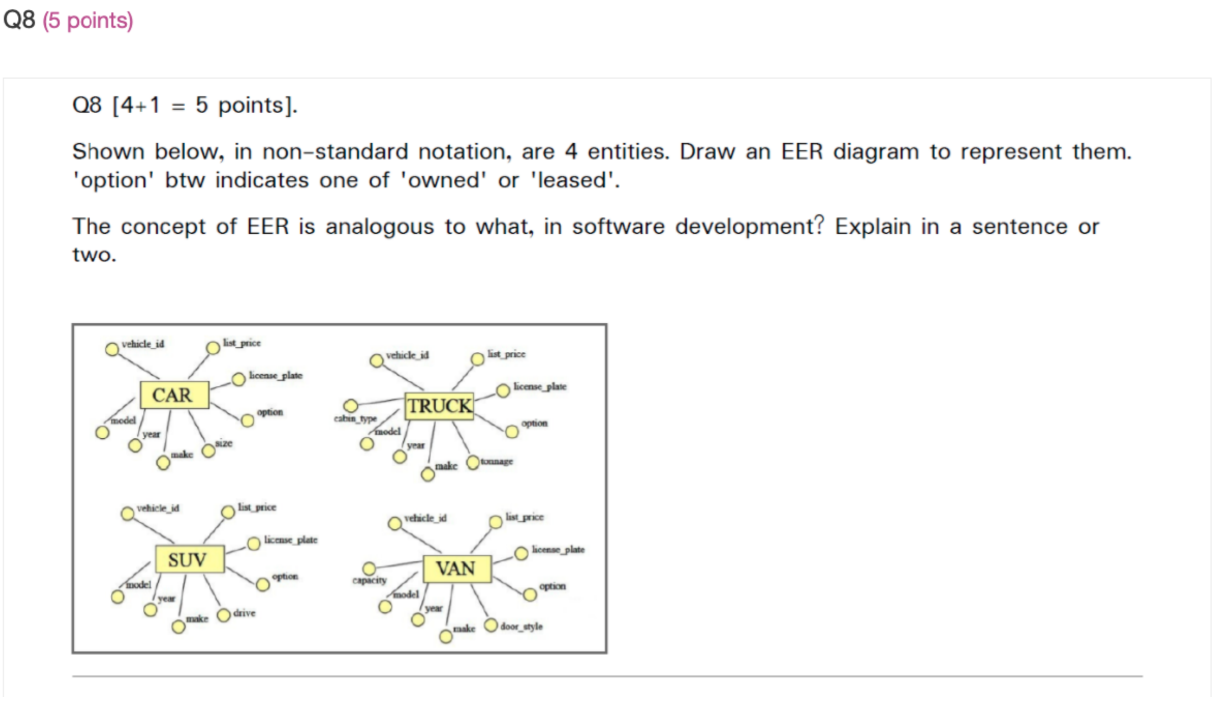
2nd Scenario (read write conflict) – 1 pt
Uncommitted Data – After T1 writes X, T2 has read the updated value. If T1 fails to commit and gets rolled back, T2 has read the wrong value and would be in an inconsistent state.

3rd Scenario (write write conflict) – 1 pt

Lost Update – Both T1 and T2 are trying to write a value of X, it is possible that one transaction overwrites the value of other as a result of which the updated value is lost.

Problems can be fixed by locking mechanisms such as 2PL locking protocol, exclusive locks, shared locks and acquiring all the locks before writing the data. – 2pt

Q8:-



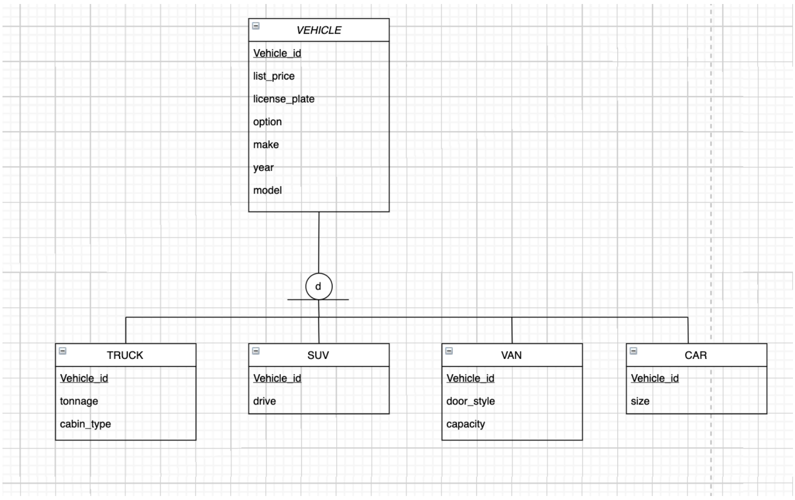
Q8 (5 points)

Q8 [4+1 = 5 points].

Shown below, in non-standard notation, are 4 entities. Draw an EER diagram to represent them. 'option' btw indicates one of 'owned' or 'leased'.

The concept of EER is analogous to what, in software development? Explain in a sentence or two.

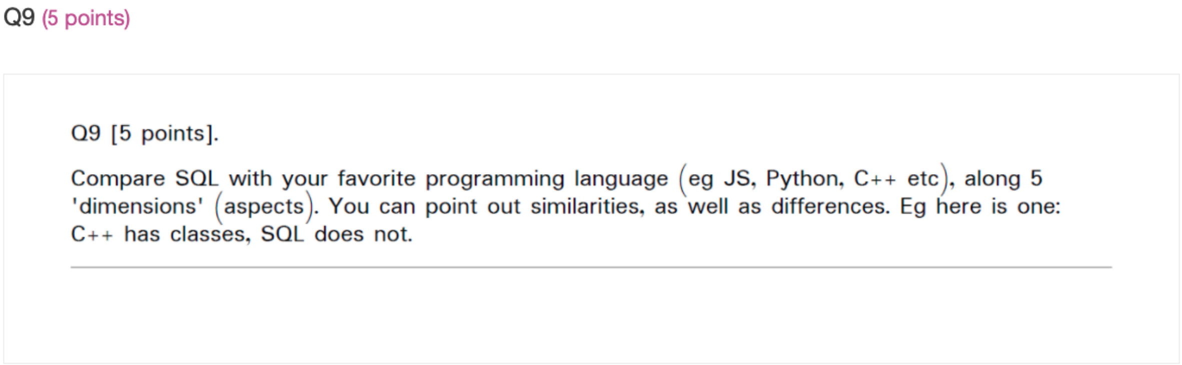
A 'VEHICLE' (or transportation device etc) superclass, with all the common properties, with CAR, TRUCK, SUV, VAN being subclasses, each with its specific attributes. Vehicle_id is the PK.



EER diagrams are basically an expanded upon version of ER diagrams. EER models are helpful tools for designing databases with hierarchy relationships.

Rubrics: -1 if superclass is not correct.
-1 if missing subclass tables
-1 if attributes in tables are not correct.
-1 if no mention of PK.
-1 if the EER concept part is wrong.

Q9:-



Q9 (5 points)

Q9 [5 points].

Compare SQL with your favorite programming language (eg JS, Python, C++ etc), along 5 'dimensions' (aspects). You can point out similarities, as well as differences. Eg here is one: C++ has classes, SQL does not.

- 1 point for each similarity/difference in 5 unique dimensions/aspects
- No point for any similarity/difference that is essentially the same aspect (e.g. the following are all related to language elements, so 1 point should be awarded, not 3)
    - (Java has classes, SQL does not; Java has interfaces, SQL does not; Java has annotations; SQL does not)

- Use same language when comparing with SQL for all 5 points (if compared with different languages, take the maximum points for one language)
- No point if any similarity/difference is inaccurate in terms of SQL features e.g. SQL has threads.
- SQL features across flavours (e.g. MySQL, PostgreSQL) are OK.
- 0.5 points for PL/SQL specific features (e.g. Variables, For loops, Triggers) rather than standard SQL.

Possible similarities/differences:

1. Language Type
    - Lang is imperative, SQL is declarative / Lang is procedural, SQL is non-procedural
2. Language Features
    - Lang has operators, SQL has operators
3. Language Domain
    - Lang is generic domain, SQL is domain specific
4. Language Compilation
    - Lang is compiled, SQL is interpreted
5. Language Execution
    - Lang is executed independently, SQL is run on DB server