QUESTION ONE

1. Find the drivers who have participated in at least one accident where the amount of the damage is at least 1000. Show the ids of those drivers.

$\Pi_{driverId}(\sigma_{damageAmount \geq 1000}(Report))$

2. Find the names of drivers who did not have an accident.

$\Pi_{name} (Driver \bowtie (\Pi_{driverId} (Driver) - \Pi_{driverId} (Report)))$

3. Find the drivers who have the same name as some other driver. Show the names of the drivers.

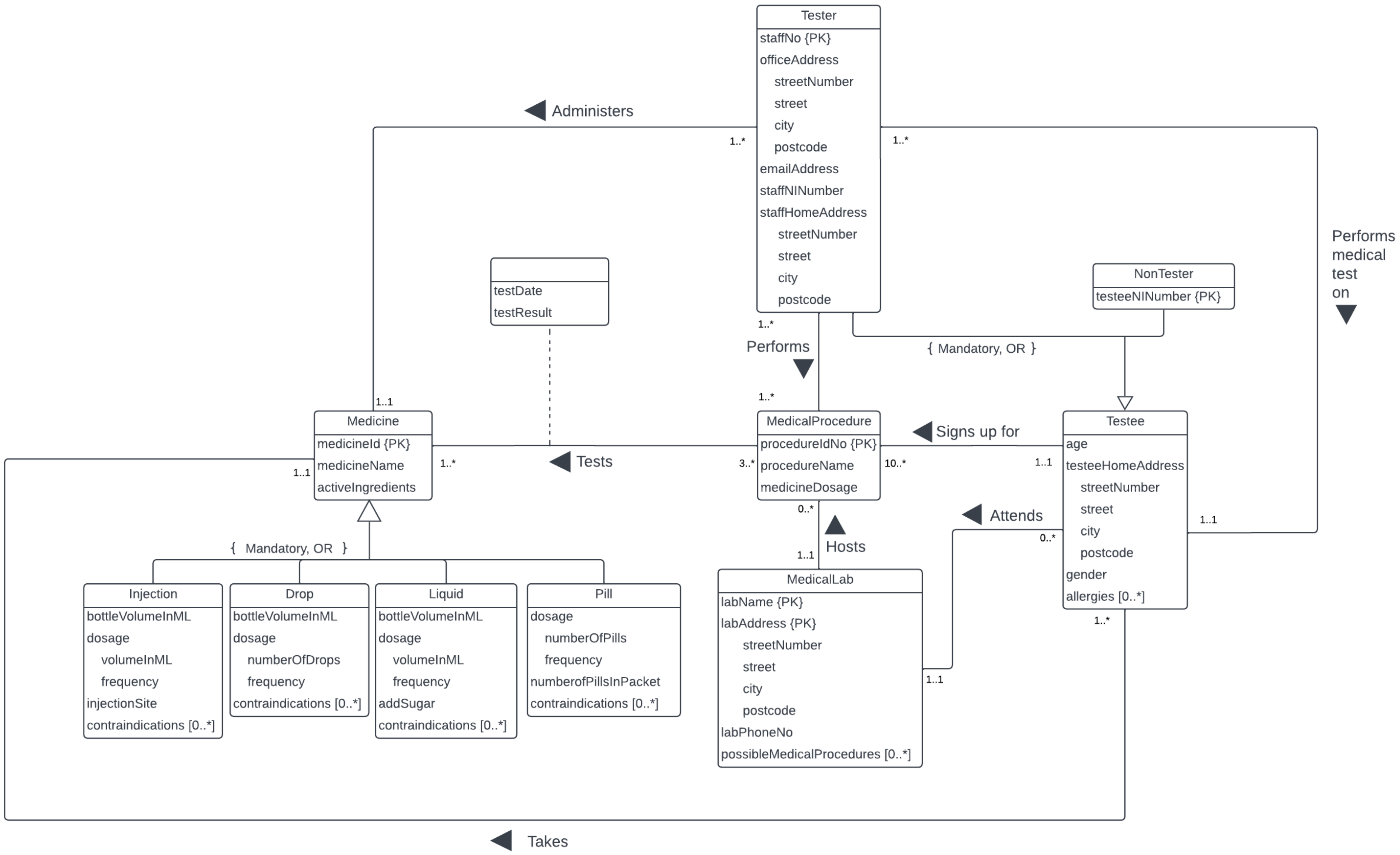
$\Pi_{Driver.name} (\rho(D,Driver) \bowtie_{(D.driverId \neq Driver.driverId) \cap (D.name = Driver.name)} Driver)$

4. Find the ids of the cars that participated in all accidents located in ‘Liverpool’

$\Pi_{carId,\ driverId}(Report) \div (\Pi_{driverId}(\sigma_{location = \text{“Liverpool”}} (Report \bowtie Accident)))$

(if we assume that the accidentNumber isn’t unique to individuals, and instead is an Id which can have multiple drivers involved)

QUESTION TWO



<u>Comments:</u> about general assumptions and comments about the data, explanations about some of the key choices, multiplicity constraints, and relationships.

- There is some ambiguity about whether the database (a.k.a. test record) acts as a record of specific test results (after the fact), or is used to plan tests (before the fact). For example, in the relationship Administers, it could be argued that there are several medicines that a Tester can administer, but in once test, they only administer one, as this is the one being measured and tested. This model assumes that the database is used for a mixture of both.
- If a Tester acts as a Testee during some test, then they cannot be their own Tester (hence no recursive relationship).
- There is only one email address per Tester, and only one phone number per Medical Lab.
- NonTester is assumed to be a member of the public who isn't a member of staff.
- No types of medical procedures have been provided in the brief; however it would be reasonable to assume that there are several different common procedures. If they had been named, they would have been subclasses of the MedicalProcedure entity.
- Each lab must be able to perform at least one medical procedure, otherwise it wouldn't make much sense to have it on the database. Hence, the multivalued attribute 'possibleMedicalProcedures' with cardinality range [1..*]
- 'Testee' is a weak entity. If 'Tester' weren't also allowed to be 'Testee'
- In each medical test, a 'Testee' takes exactly one 'Medicine'. Therefore, the multiplicity on the 'Medicine' side of the 'Takes' relationship is 1..1. Of course, if the model recorded all the medicines that a 'Testee' has ever taken for the school, then the cardinality would be many.
- Each 'Medicine' is taken by more than one 'Testee' in different tests and with several medical procedures, and thus is involved in more than one test.
- There is ambiguity about whether the different medical procedures (at least three) needed to test the new medicine should be different instances or of completely different types of procedure.
- Each medical procedure can be used to test one or several medicines.

Relationships and multiplicities

- 'Performs' relationship
  - Tester is 1..* because a MedicalProcedure can't be carried out without a Tester, however it is feasible that more than one tester is used to perform a procedure.

    - The brief states that a tester is a well-trained member of staff, so it is reasonable to assume that they know how to perform the medical procedures which are used by the School to test the new medicines, so they could perform any number of procedures. The number of procedures used by the School to test medicines has not been provided.
    - The brief states that a medical lab may not be able to conduct a given medical procedure, meaning that the participation on MedicalProcedure is optional. Different labs will be able to perform a different number of medical procedures, hence the cardinality of many (*).
- ‘Hosts’ relationship
    - 1..1 on conducts, depends upon whether we define the medicalprocedure as the instance of it happening or a description of how it happens (which I think is more accurate)
    - Given that there is always a lab capable of conducting each medical procedure, it stands that the participation in ‘Conducts’ has to be mandatory. Furthermore, a test occurrence will only be performed in one lab (it can’t simultaneously occur in more than one lab with the same Testee and Tester), hence the cardinality is 1.
- ‘Signs up for’ relationship
    - Brief doesn’t specify how many Testees are needed for each medical procedure testing a medicine, so I will assume that each procedure requires exactly 1 subject for testing
- ‘Administers’ relationship
    - Medicine is 1..1 because we can assume that the School would only test one medicine per test in order to make the observations and results meaningful and accurate.
    - Tester is 1..* because there may be more than one Tester in a Test, e.g. one Tester may give the first dose of drops and another may give the second.
- ‘Performs medical test on’ relationship
    - Tester is 1..* because there could be multiple Testers involved
    - Testee is 1..1 because the purpose of the model is to record information about individual participants in a test. It is possible that there could be many Testees involved in a test, however this is outside of the scope of this model.
- ‘Attends’ relationship
    - Testee has optional participation because there may not be a test going on in a given lab at any time. It has a cardinality of many because there

may be several tests going on concurrently in a given lab, for example in different rooms.
    - MedicalLab is 1..1, because a Testee must have a Test done in a location, and a Testee can't have a Test in more than one place at once (although obviously they are likely to go to different labs for different tests. This is however not what the relationship represents).

Primary keys

- MedicalLab: The primary key is a composite key comprising of labName and labAddress. It couldn't be phone number, because there may be several labs within the same building or complex which use the same phone number. Labs might have the same name therefore the address is also required to make the lab uniquely identifiable.
- Testee: where it is a NonTester, primary key is testeeNINumber; where it is a Tester, primary key is staffNo.
- Tester: the primary key could be either staffNo or staffNINumber as both can uniquely identify tuples. In this model, staffNo is the primary key with staffNINumber as an alternate key.

Attributes

- Address (officeAddress and StaffHomeAddress) are composite attributes
- In 'Testee' there is no attribute as NI number. This is because this information is covered in the subclasses 'Tester' and 'NonTester'. It has been renamed staffNINumber in the entity 'Tester' because the attribute names need to be distinct. There will always be two NI numbers when the entities are joined for the test record, therefore they need to be differentiated.

Miscellaneous

- testDate and testResult have been designated as attributes of the relationship 'Tests' between Medicine and MedicalProcedure, because they are attributes of the actual occurrences of tests rather than part of an entity involved in the medical tests.
- schoolOfMedicine is another potential entity, however it would be redundant as a) it doesn't add a significant amount of value to the model, and b) the commissioning body is the school so it doesn't need to be included – e.g. they know that Testers are members of staff from the school