1. Consider the following scenario:

A policy-holder may have a number of policies with the insurance company. Each policy is given a policy number and relates to a single policy-holder. The company has a range of insurance products and may put together a range of products to form a policy. Examples of motor products are: third party, fire, theft, accident damage, windscreen cover, etc. Brokers sell policies for commission and any one policy may have commission payable to more than one broker. Claims are made against policies. A claim relates to only one policy and each claim is classified according to one of six claim types. The company's products are grouped by business area, i.e. life, motor, marine, etc. Any particular product belongs to only one business area. The company holds information on clubs and associations, for promotions and selective mailings. A policy-holder may belong to a number of different associations.

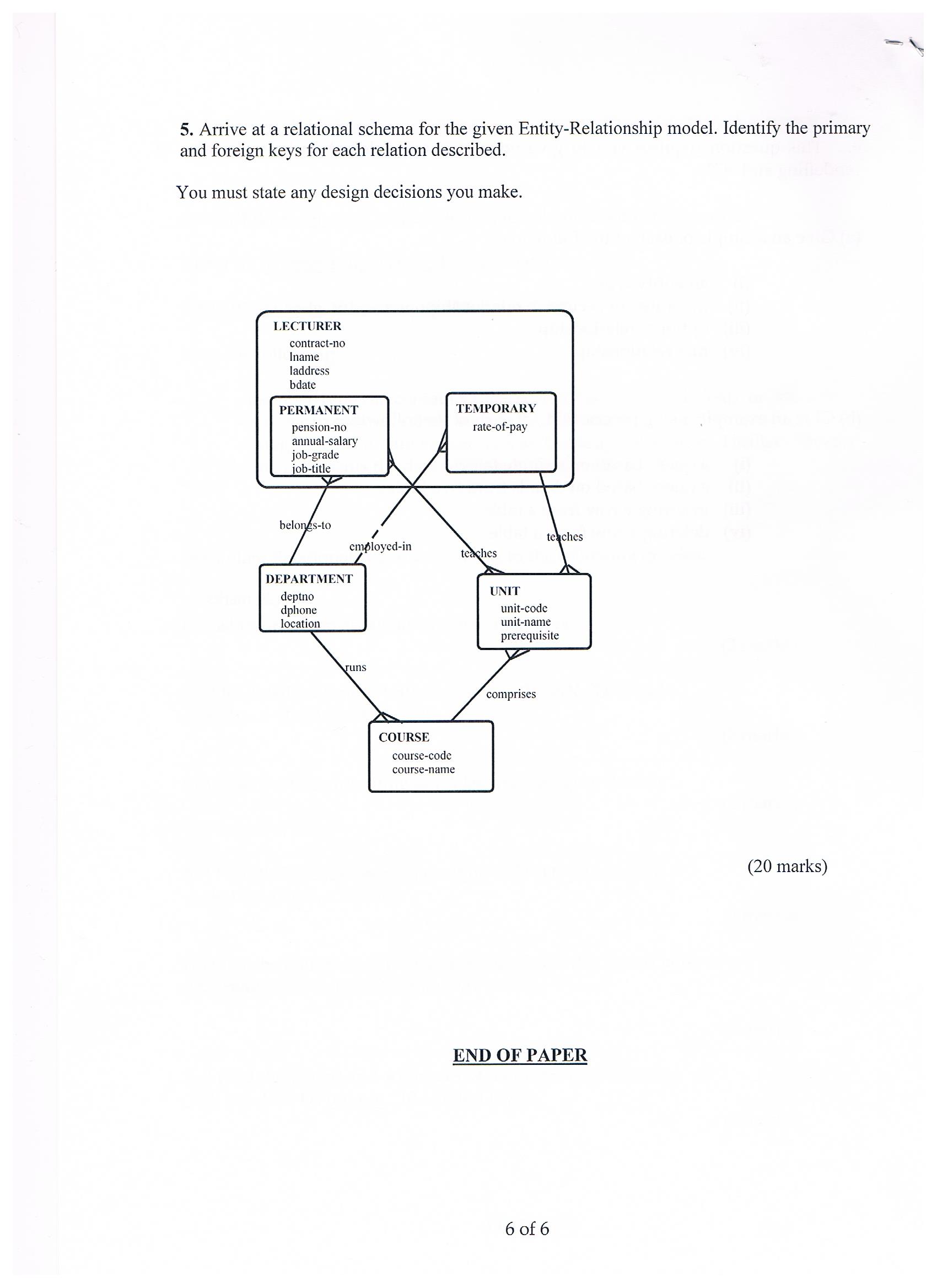
(a) Produce an Entity-Relationship diagram which represents the above activities of the insurance company. Indicate the names and degrees of all the relationships. Give appropriate attributes for each entity shown, and identify the primary key for each entity.

(b) Give three queries which would be most likely to be asked in this scenario and give a clear explanation of how each may be answered.

2.

Arrive at a relational schema for the given Entity-Relationship model. Identify the primary and foreign keys for each relation described.

You must state any design decisions you make.



3. This question requires you to give your own examples of various elements of modelling and SQL.

**(a)** Give an example of each of the following:

(i) an entity type

(ii) involuted or recursive relationship

(iii) optional relationship

(iv) m:n relationship.

**(b)** Give an example, using precise SQL, of each of the following:

(i) a query based on a single table

(ii) a query based on details from two tables

(iii) inserting a row from a table

(iv) deleting a row from a table.

**(C)** A relational database management system for DVD rentals includes the following tables:

MEMBER (member-id, name, address, telephone, member-type, dob) RENTAL (member-id, DVD-code, date-out, date-due)

DVD (DVD-code, title, category, format, rating)

Note the following:

- member-type can contain values such as 'adult', 'child', 'oap' or 'student'. - rating can contain values such as 'U', 'PG', '12A', '15' or '18'.

- category can contain values such as 'Comedy', 'Romance', Thriller', 'Horror'. - dob represents the date of birth of the member.

Formulate Relational Algebra solutions to the following queries:

(i) List information about all 'student' members.

(ii) List details of rentals with date-out = '01-JAN-2009' and date-due = '03-JAN-2009'.

(iii) Find the title and rating of DVDs with category 'Horror'.

(iv) Find the title, category and rating of DVDs which were rented by member 'Smith'.

(v) Find the names and addresses of members who rented both `PG' rated DVDs and '12A' rated DVDs.

(vi) Find the names and addresses of members who rented 'PG' rated DVDs but not '12A' rated DVDs.

**4.**

**2.** Consider the Sales Order Form below:Normalise the Sales Order form to produce an equivalent set of entities in third normal form.

3 of 6

TeleCom Ltd

**sale number: 0057435**

**sale date: 07-July-2012**

**customer number: 238790 customer name:** Tommy Delores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **model number** | **product type** | **unit price** | **sale quantity** | **line total** |
| T5060 | mobile phone | 220 | 2 | £440 |
| PT42 | earphones | 10 | 1 | £10 |
| QZE248 | earphones | 12 | 1 | E12 |

|  |  |
| --- | --- |
| **sale total:** | £462 |

1. Draw an Entity-Relational model diagram showing the entities and relationships that correspond to your third normal form structure.