**ENTITY RELATIONSHIP DIAGRAM**

You have been asked to design a database for a horse racing enterprise. The following information has been gathered about the operation of the enterprise.

A horse has a registration number, name, type (quarter horse or thoroughbred), gender, and trainer. The heritage of every horse must also be maintained if the information is available. For example, the mother (dam) and father (sire) should be recorded. It is also necessary to identify the offspring of a given horse. Information about people involved in the horse racing business should be maintained. An identifier, name, address, and phone number should be maintained about every person. If a person is a horse trainer, the salary of the trainer should be indicated, along with the horses trained by the trainer. If a person is a jockey, a record of their weight at different times must be maintained, along with the date of recording. The numbers of wins and runs so far for each jockey need to be recorded. It is possible for a person to be both a trainer and a jockey. Name and contact information can also be maintained about people other than trainers and jockeys.

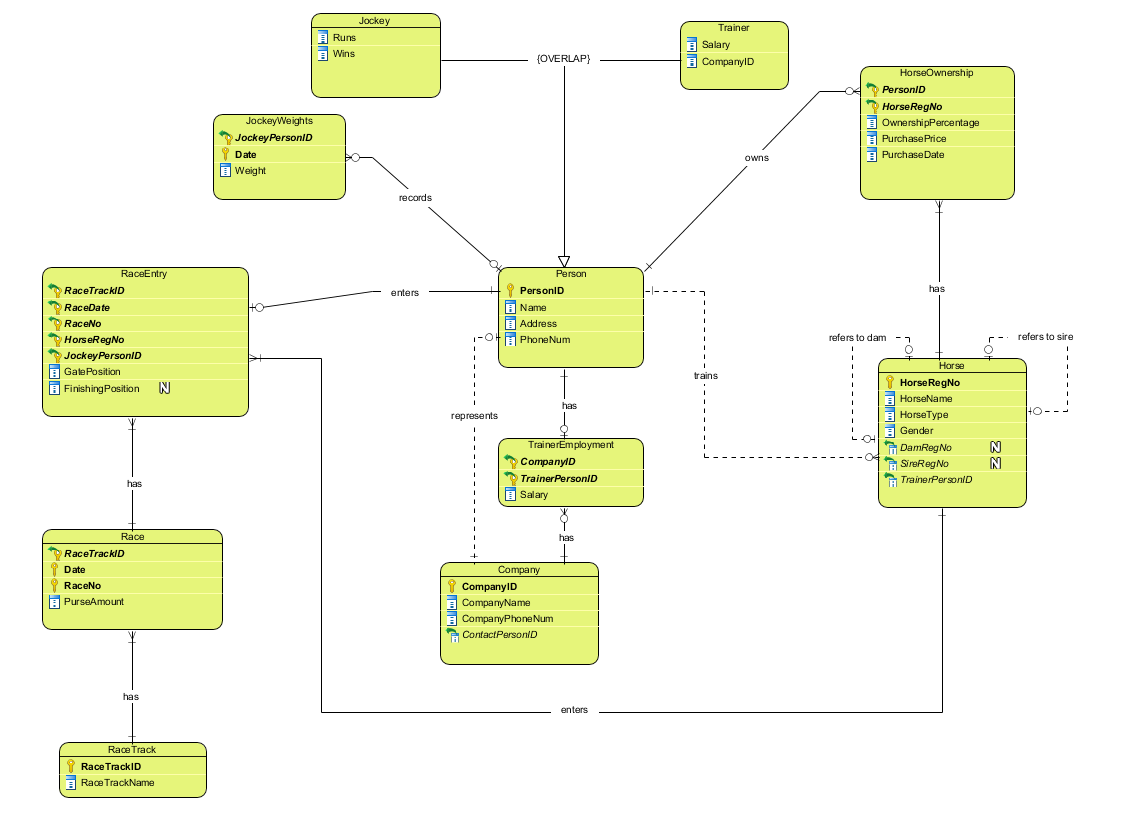
A stable is a company that breeds, trains and sells/buys horses. A stable has an identifier, a name, phone, and contact person. A horse trainer works for a specific stable.

A horse can have more than one owner. An owner can own more than one horse. An owner can either be a stable or a person. Information is always recorded about the most recent date and purchase price of a horse by its current owner(s). Each owner must also record their percentage ownership of a horse. For example, a horse could be owned by three owners, where one owner has a 50% share and the others two owners each have a 25% share.

A racetrack has a track identifier and a track name. Every racetrack has a race schedule indicating the date of each race day and the list of races for each race day.

A race day typically has ten scheduled races, where each race has a race number (from 1 to 10) and a purse. The purse is the amount of money awarded to the winner of the race.

Every race has several entries. Each entry indicates the horse, jockey, and gate position of the horse at the start of the race. After the race, the entry records the finishing position of the horse (first, second, third, etc.). Every horse and every jockey must be able to produce a history of the races in which they have participated.

**Question 1) Analyse these data requirements and produce an appropriate database design using the Enhanced Entity Relationship (EER) model. A good design will minimise duplication of attributes and show only essential relationships.**

**Question 2) Document your design using Visual Paradigm. Produce an EER diagram showing entities, attributes and named relationships in a form ready for implementation using a relational database. Print a copy of the Visual Paradigm diagram showing all attributes including foreign keys. The relationships should show their cardinality and whether they are optional or mandatory. Describe any significant assumptions you make concerning the data requirements.**

In the ERD in Question 1, the following is shown:

* Entities (including sub/super classes)
* All attributes, including foreign keys

Within the ERD, through use of crow’s foot notation, the following is shown:

* Relationships and relationship cardinality
* Participation constraints (whether relationships are optional or mandatory)

The only strong entities are Person, Company, Horse and RaceTrack. All other entities are reliant on other entities to exist (and are therefore weak entities).

Assumptions

1. It is assumed that all horses (sires, dams, offspring) are racehorses (in order to ensure the recursive relationships within the Horse entity work). In this case, it is also redundant to have a separate ‘offspring’ attribute within Horse, as offspring for a particular horse can be derived by knowing dam/sire relationships (see additional notes below).
2. Some sires and dams are not known so these attributes are nullable.
3. Within RaceEntry, ‘FinishingPosition’ is nullable in case a RaceEntry doesn’t finish a race.
4. JockeyWeights are a separate entity in order to minimise repetition. The assumption here is that jockeys are only weighed once per day so that JockeyPersonID and Date can make up a composite primary key.

Additional Notes

While certain attributes are not explicitly shown, they can be derived, as can particular views:

* The race history of a particular Jockey or Horse can be determined through filtering the RaceEntry entity type by ‘HorseRegNo’ or ‘JockeyPersonID’.
* The offspring of a Horse can be derived by filtering by ‘DamRegNo’ or ‘SireRegNo’. For example, if we want to know the offspring of a Horse (with gender=’female’ and HorseRegNo=2314), we can select only entity instances with DamRegNo=2314 and we will see a list of the Dams offspring. Again, this is under the assumption that all Horses are RaceHorses and are entered into our database.
* A race schedule for a particular day and track (if needed) can be derived by filtering ‘Date’ and ‘RaceTrackID’ within the Race entity.
* The TrainerEmployment entity is created to ensure a CompanyID attribute in Person is not necessary – without this entity, there would be numerous null values in Person.

**Question 3) Discuss what enhanced features of EER you have used for the design, if any.**

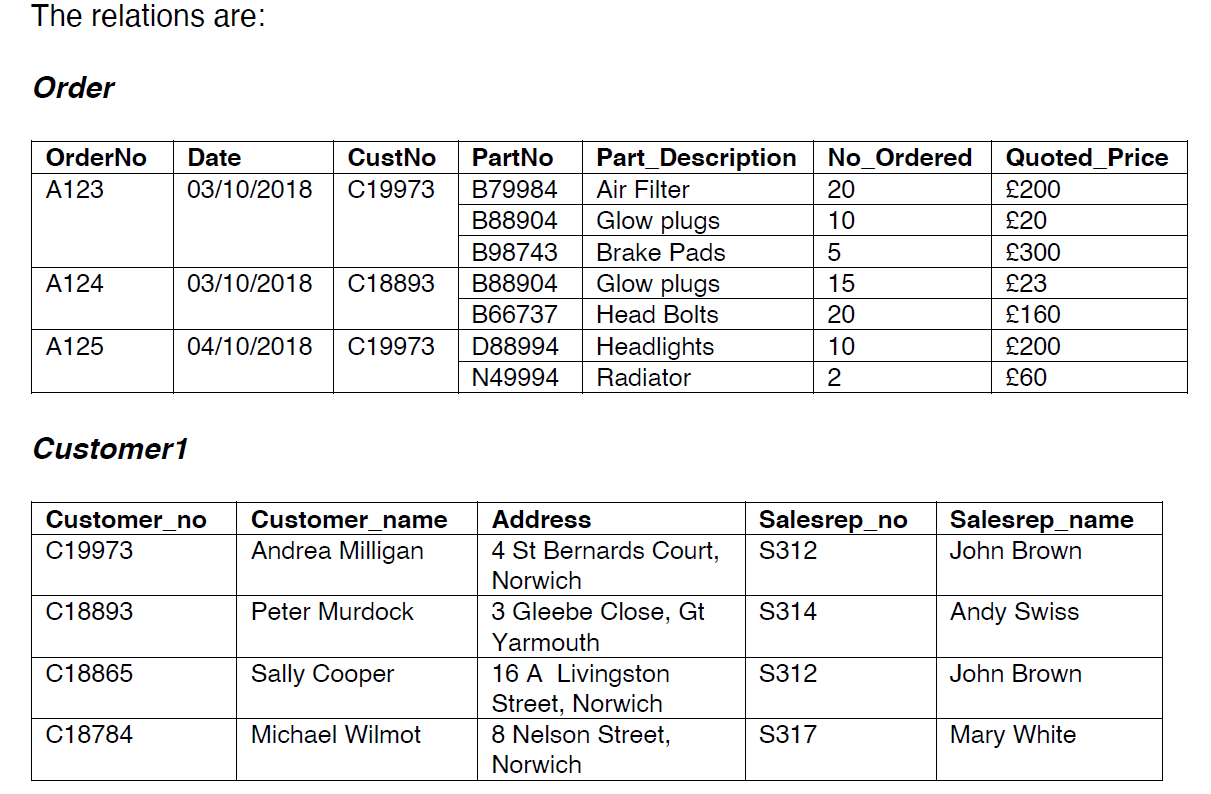
Generalisation was used when creating the Person, Jockey and Trainer categories. In this instance, Person is the superclass and Jockey and Trainer are overlapping subclasses meaning a person can be a jockey, a trainer or both.

**Question 4) Briefly discuss any constraints that may be required to maintain the model.**

* RaceNo must be between 1 and 10.
* For the HorseOwnership entity, total ownership percentages for a particular Horse should not exceed 100%.
* A single Trainer can only work for one Company.

Other constraints are denoted on the ERD.

**Database Normalisation:**

******

***Question 1) Convert the tables into relations in first normal form (1NF)***

To be in 1NF, a relation must contain only atomic values and no repeating groups. The initial Orders table contains 3 values each within PartNo, Part\_Description, No\_Ordered and Quoted\_Price cells. To convert to 1NF, we ensure each cell has only one value.

The following tables are converted into 1NF by addressing these multivalued attributes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ORDERS** | | | | | | |
| **OrderNo** | **Date** | **CustNo** | **PartNo** | **Part\_Description** | **No\_Ordered** | **Quoted\_Price** |  |
| A123 | 03/10/2018 | C19973 | B79984 | Air Filter | 20 | £200 |  |
| A123 | 03/10/2018 | C19973 | B88904 | Glow plugs | 10 | £20 |  |
| A123 | 03/10/2018 | C19973 | B98743 | Brake Pads | 5 | £300 |  |
| A124 | 03/10/2018 | C18893 | B88904 | Glow plugs | 15 | £23 |  |
| A124 | 03/10/2018 | C18893 | B66737 | Head Bolts | 20 | £160 |  |
| A125 | 04/10/2018 | C19973 | D88994 | Headlights | 10 | £200 |  |
| A125 | 04/10/2018 | C19973 | N49994 | Radiator | 2 | £60 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CUSTOMERS** | | | | |
| **Customer\_no** | **Customer\_name** | **Address** | **Salesrep\_no** | **Salesrep\_name** |
| C19973 | Andrea Milligan | 4 St Bernards Court, Norwich | S312 | John Brown |
| C18893 | Peter Murdock | 3 Gleebe Close, Gt Yarmouth | S314 | Andy Swiss |
| C18865 | Sally Cooper | 16 A Livingston Street, Norwich | S312 | John Brown |
| C18784 | Michael Wilmot | 8 Nelson Street, Norwich | S317 | Mary White |

***Question 2) Draw up a list of functional dependencies for the 1NF defined data and thus convert the relation definitions into second normal form.***

An assumption made here is that sales reps are assigned based on location. The following is a list of functional dependencies present in the 1NF defined data:

|  |  |  |
| --- | --- | --- |
| **Functional Dependencies** | | |
| **OrderNo** | **→** | **Customer\_no** |
| **OrderNo** | **→** | **Date** |
| **PartNo** | **→** | **Part\_Description** |
| **{OrderNo, PartNo}** | **→** | **No\_Ordered** |
| **{OrderNo, PartNo}** | **→** | **Quoted\_Price** |
| **Customer\_no** | **→** | **Customer\_name** |
| **Customer\_no** | **→** | **Address** |
| **Customer\_no** | **→** | **Salesrep\_no** |
| **Customer\_no** | **→** | **Salesrep\_name** |
| **Salesrep\_no** | **→** | **Salesrep\_name** |

|  |  |
| --- | --- |
| **PARTS** | |
| **PartNo** | **Part\_Description** |
| B79984 | Air Filter |
| B88904 | Glow plugs |
| B98743 | Brake Pads |
| B66737 | Head Bolts |
| D88994 | Headlights |
| N49994 | Radiator |

After defining the functional dependencies, we can convert the tables into 2NF (as below) by ensuring that no relations contain any partial dependencies.

|  |  |  |
| --- | --- | --- |
| **ORDERS** | | |
| **OrderNo** | **CustNo** | **Date** |
| A123 | C19973 | 03/10/2018 |
| A124 | C18893 | 03/10/2018 |
| A125 | C19973 | 04/10/2018 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **ORDER\_DETAILS** | | | |
| **OrderNo** | **PartNo** | | **No\_Ordered** | **Quoted\_Price** |
| A123 | B79984 | | 20 | £200 |
| A123 | B88904 | | 10 | £20 |
| A123 | B98743 | | 5 | £300 |
| A124 | B88904 | | 15 | £23 |
| A124 | B66737 | | 20 | £160 |
| A125 | D88994 | | 10 | £200 |
| A125 | N49994 | | 2 | £60 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CUSTOMERS** | | | | |
| **Customer\_no** | **Customer\_name** | **Address** | **Salesrep\_no** | **Salesrep\_name** |
| C19973 | Andrea Milligan | 4 St Bernards Court, Norwich | S312 | John Brown |
| C18893 | Peter Murdock | 3 Gleebe Close, Gt Yarmouth | S314 | Andy Swiss |
| C18865 | Sally Cooper | 16 A Livingston Street, Norwich | S312 | John Brown |
| C18784 | Michael Wilmot | 8 Nelson Street, Norwich | S317 | Mary White |

***Question 3) Consider the 2NF definitions and describe any possible problems with update, insert or delete anomalies.***

**Update anomalies:**

If we assume that sales reps are assigned based on location, if a customer moves to a new house (in a different area), a different sales rep would be assigned. We would have to alter the Salesrep\_no and the Salesrep\_name – this could be an example of an update anomaly because we are required to update 2 table columns (if we forget, this will be problematic). To solve this problem, transitive dependencies should be removed. In this example, we would keep Salesrep\_no assigned to each customer but a separate table would be created to link Salesrep\_no to Salesrep\_name (and Salesrep\_name would be removed from the Customers table). By doing this, only one update is required.

For example, if C19973 moves to Great Yarmouth, we would change Salesrep\_no to S314. We wouldn’t need to update the Salesrep\_name as this is done automatically when Salesrep\_no is reassigned.

**Insert anomalies:**

If we try to add a new sales rep into our current database, this would be impossible without first assigning our new sales rep to a customer. Functionally, this is not what we want. In the real world (which is what we are trying to model), a sales rep is usually assigned to a region before he has any customers (particularly if the sales rep is working in new territory due to business expansion). This is an example of an insertion anomaly.

For example, if the company expands and a new sales rep (S318, Emma Smith) begins work in North Norfolk (a new territory), we cannot currently add her to the database as no customers have been assigned to her.

To solve this problem, sales reps are added independently of customers in a separate table.

**Delete anomalies:**

In our current table, if we want to delete a customer record (and the customer is the only one assigned to a particular sales rep), by deleting the customer record, we also lose all data for the assigned sales rep).

For example, if we delete customer C18893 from our customer table, we would also lose all data pertaining to sales rep S314 (as this sales rep is singularly assigned to C18893). This would be an example of a delete anomaly.

To solve this problem, sales rep data needs to be maintained independently of customers – a new sales rep table is created.

**Anomalies can be reduced and eliminated by normalizing further and reducing duplication and redundancy.**

***Question 4) Convert the relation definitions into 3NF.***

For a table to be in 3NF, it must already be in 2NF and additionally, transitive dependencies must be removed. In the table above we had a transitive dependency between CustomerNo -> Salesrep\_no -> Salesrep\_name. This has been removed and Salesreps is now a new separate table.

The following tables are the result of converting the tables defined above (in Q2) into 3NF:

|  |  |
| --- | --- |
| **PARTS** | |
| **PartNo** | **Part\_Description** |
| B79984 | Air Filter |
| B88904 | Glow plugs |
| B98743 | Brake Pads |
| B66737 | Head Bolts |
| D88994 | Headlights |
| N49994 | Radiator |

|  |  |  |
| --- | --- | --- |
| **ORDERS** | | |
| **OrderNo** | **CustNo** | **Date** |
| A123 | C19973 | 03/10/2018 |
| A124 | C18893 | 03/10/2018 |
| A125 | C19973 | 04/10/2018 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **ORDER\_DETAILS** | | | |
| **OrderNo** | **PartNo** | | **No\_Ordered** | **Quoted\_Price** |
| A123 | B79984 | | 20 | £200 |
| A123 | B88904 | | 10 | 20 |
| A123 | B98743 | | 5 | 300 |
| A124 | B88904 | | 15 | 23 |
| A124 | B66737 | | 20 | 160 |
| A125 | D88994 | | 10 | 200 |
| A125 | N49994 | | 2 | 60 |

|  |  |  |  |
| --- | --- | --- | --- |
| **CUSTOMERS** | | | |
| **Customer\_no** | **Customer\_name** | **Address** | **Salesrep\_no** |
| C19973 | Andrea Milligan | 4 St Bernards Court, Norwich | S312 |
| C18893 | Peter Murdock | 3 Gleebe Close, Gt Yarmouth | S314 |
| C18865 | Sally Cooper | 16 A Livingston Street, Norwich | S312 |
| C18784 | Michael Wilmot | 8 Nelson Street, Norwich | S317 |

|  |  |
| --- | --- |
| **SALESREPS** | |
| **Salesrep\_no** | **Salesrep\_name** |
| S312 | John Brown |
| S314 | Andy Swiss |
| S317 | Mary White |

***Question 5) What is the difference between 3NF and Boyce-Codd Normal Form? Are the 3NF relations produced in 4) above in BCNF?***

Boyce-Codd Normal Form is an extension of Third Normal Form. Data should already be in 3NF and additionally, every determinant is a candidate key. In Question 4, relations are already in BCNF because every determinant is a candidate key.

***Question 6) Identify any Multi-Valued Dependencies (MVD) and functional dependencies and convert Customer2 to 4NF:***

There is one functional dependency in Customer2: CustomerNo -> CustomerName.

There are 2 multivalued dependencies in Customer2:

CustomerNo -> -> Delivery Address

CustomerNo ->-> MobileNo

To convert into 4NF, multivalued dependencies need to be removed. The below tables are a result of removing multivalued dependencies and conversion into 4NF.

|  |  |  |  |
| --- | --- | --- | --- |
| **Customers** | | | |
| **Customer\_no** | | **Customer\_name** | |
| D20205 | | Peter Milligan | |
| C20848 | | Sally Black | |
| **Customer\_Addresses** | | | |
| **Customer\_no** | **Address** | | |
| D20205 | 4 St Bernards Court, Norwich | | |
| D20205 | 15 The Avenues, Wymondham | | |
| C20848 | 3 Gleebe Close, Gt Yarmouth | | |
| **Customer\_Telephones** | | | |
| **Customer\_no** | | | **TelNum** |
| D20205 | | | 07934 848488 |
| D20205 | | | 07345 988383 |
| C20848 | | | 07329 399872 |
| C20848 | | | 07458 716283 |