CS2DI17 Group Coursework

Module Code: CS2DI17

Assignment report Title: Group Coursework Report

Student Numbers: 27816792, 29012930, 29003671

Date: 25/11/2021

Actual hrs spent for the assignment: 17

Assignment evaluation (3 key points): Detailed brief, good explanation, straightforward and accomplishable.

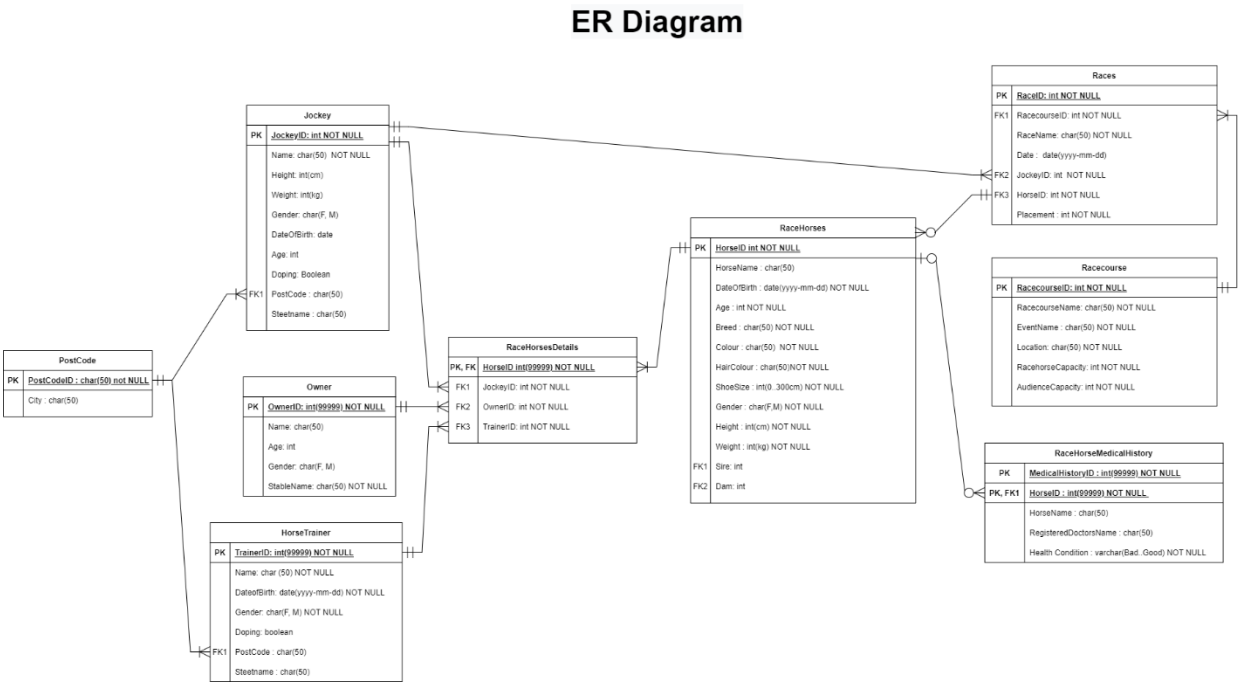
Table of Contents

Design

ER Diagram

Final Model – Third Normal Form (1NF and 2NF is in the appendix Figure 1 and 2, respectively)

Figure 1 (refer to appendix) shows the model before normalisations and in first normal form.



We assume that every horse that is existing will have a stable that they live at however we are aware that not all horses will have a particular stable and could have other living conditions however in terms of the races horses we assume they all have a stable to live in.

The second assumption is we assume every horse has a particular owner however a constraint is that some horses may be owned by organisations compared to individual people however we assume by having a owner ID they all have people as there owners.

One of our assumptions was that not all horses would have a sire and a dam alive and therefore we did not consider it as not null and therefore may not have a living parent.

We assume by linking racehorses to race that all racehorses would be participating in some sort of race however a constraint for this could be a racehorse may be injured or not be participating in races during a given time however still be classified as a racehorse.

Another assumption is assuming all racehorses have a medical history and within this have a registered doctor for them however a possibility is that not all racehorses would have a registered doctor and therefore a past medical history.

We assume every horse has a trainer as trainer is Not Null however there are possibilities that not all horses would have a trainer teaching them and instead are manged by their jockey.

Normalisation Justification

First normal form

(figure 1 in appendix) Now for our model to be in first normal form the following requirements needed to be fulfilled. Firstly, that there is no multi repeated values however a key requirement is that if the table has a unique primary key within the table, then it is automatically said to be in first normal form. Each table must be identified with a unique column called the primary key. In terms of our model in achieving first normal form this has been successful because for each of our entities there is a unique primary key which gives us the ability to identify each individual table with this primary key and therefore our model has fulfilled this expectation.

An example of why we believe our table is in first normal form is considering the racehorse entity. Some of the attributes in the racehorse entity such as colour or breed although could be repeated, based on the primary key of horse ID, no two horses can have the same horse ID and therefore we distinguished that this acted as a primary key and therefore satisfied the first normal form principles. Because of the primary key it emphasizes the fact that we wouldn't have repeated rows in our tables and therefore evidently shows our table in first normal form. In terms of the necessary steps that we needed to take in order to convert our model to first normal form, no changes were made from before normalization to satisfying first normal form as we had realized that all entities within our ER model all had a unique primary key and therefore allow us to have the assumption and belief that our ER model was already normalized to the first normal form. As you can see every entity in our model has a unique primary key before normalization therefore no changes were made.

Second normal form

(figure 2 in appendix)

In terms of satisfying second normal form in our model the requirements we followed was firstly that our model is in first normal which it is but also the requirement of all non-primary key attributes must be fully functionally dependent on the primary key. This means in terms of the non-primary key attributes, by using the primary key you can distinguish and find out the non-primary key elements. In order to satisfy second normal form, the steps we had taken was to firstly go through each entity and see which of the non-primary key attributes are fully functionally dependent on the primary key.

The entity that we felt firstly did not comply to second normal form was the racehorse entity. The racehorse entity contained attributes such as jockey id, trainer id, and owner id. Considering this we realized that these attributes were not functionally dependent on the primary key(horse id). We came to the conclusion that by knowing the racehorse ID, you are not able to find the answer to those three attributes as for example a particular race horse ID could have many trainers or many jockeys or many owners and therefore they are not fully functionally dependent on the primary key. The necessary step we took was to remove these 3 attributes from the racehorse entity and create a separate entity called racehorse details where jockey ID, trainer ID and owner ID were placed into this entity. Then in order to link this new entity (racehorse details) to the original racehorse entity we used the primary key HorseID and placed it into the new entity race horse details. This allowed the two entities to be linked by a 1 to many relationships to emphasize that a race horse can have many trainers, owners and jockeys and that you cannot distinguish these attributes from just the primary key of horse ID

The steps from first form to second form

- We removed jockey ID, trainer ID and owner ID from the race horse entity and put it in a separate entity called race horse details. We then added the horse ID as a primary key in order to link the two entities together.
- Finally, the links from the jockey, owner and horse trainer entities are now linked to the new entity racehorse details and allows the race horses entity to now be in second normal form. Refer to figure 2 (in appendix) for image of how we went from first to second normal form.

In terms of considering the rest of the entities we examined whether by using the primary key that you can distinguish the non-primary key elements. The team realized every other entity non primary key attributes were already fully functionally dependent on the primary in that entity and therefore no further changes needed to be made as we believed our model met the criteria of second normal form.

An example of how we considered this for one of the entities is considering the races entity which has race ID as a primary key, by looking at all its attributes we felt that by knowing the race ID you are able to distinguish race name as the ID would have a particular name, the racecourseID, date because by knowing the raceID you are able to search what particular date it will be help on, jockeyID as you are able to know who the particular jockeys participating in the race our by knowing the raceID, horseID and placement. We considered that all these attributes can all be obtained by the primary key and therefore complied to 2nd normal form.

This is the approach we took for all the remaining entities and no further changes were made as we believed all non-primary key attributes are dependent on the primary key. Therefore, the only change that was made was in the racehorse's entity in which the entity was split to allow it to comply to 2nd normal form.

Third normal form

In order to satisfy third normal form, the approach we took is that the model must first comply with first and second normal which we have achieved and then to check whether every non primary key attribute depends only on the primary key. This approach basically meant we tried to identify whether there were any non-primary key attributes that could be acting like a primary key. We went through each individual entity to see if any of attributes acting like a primary key but wasn't one. We discovered that two entities did not comply to third normal form. These two were the jockey entity and the horse trainer entity. These both contain two attributes city and postal code as a way of representing the jockey and horse trainer's address. Now from looking at the entity we believed that the "city" attribute acted as a primary key for postal code because if the city that the horse trainer and jockey lived in changed, this would automatically affect the postal code as no cities have the same postal code therefore it will cause it to automatically change. Therefore, the city attribute acted like a primary key which meant we needed to create a separate entity to represent these two attributes for jockey and horse trainer. This is shown below:

- Releasing city acted as a primary key we removed it from the jockey entity

- we also removed it from the horse trainer entity

- We placed into a new entity called postal code and linked the new entity to the original jockey and trainer entity with the foreign key of postal code. This allowed us to achieve third normal from for the horse trainer and jockey entity

- Refer to figure 4 (in appendix) for image on how went from second to third normal form

As we can see the street name has been left inside of the original entities and not taken. This isn't because in our opinion even though the city may change which will affect the postcode automatically, it won't always affect street name and this because sometimes the same street name can be used in different cities and therefore the same street name may exist more than once in different areas and therefore it not affected by the changing of the city and hence it doesn't affect the requirements of third normal form and hence why we decided to keep it in the original entities. Based on this approach it allowed the entity to fulfill the third normal form requirements as shown above.

In looking at the rest of the entities no further changes were made as in the remaining entities there were no other non-primary key attributes which acted like a primary key and therefore the rest of the ER model was kept the same.

Through this justification and breaking down each of the three forms we can say that our model now complies with the principles of normalization.

Implementation

Databases Implementation

Racehorses	
Create Table	**Data Insertion / Population** *appendix 1
create table RaceHorses (HorseID int NOT NULL, HorseName char(50), DateofBirth date NOT NULL, Age int NOT NULL, Breed char(50) NOT NULL, Colour char(50) NOT NULL, HairColour char(50) NOT NULL, ShoeSize int NOT NULL, Gender char NOT NULL, Height int NOT NULL, Weight int NOT NULL,	INSERT INTO RACEHORSES(HORSEID, HORSENAME, DateOfBirth, AGE, BREED, COLOUR, HAIRCOLOUR, SHOESIZE, GENDER, HEIGHT, WEIGHT, SIRE, DAM) VALUES (32048, 'Chips', '1990-09-05', 31, 'Akhal Teke', 'Chestnut', 'Brown', '144', 'M', 177, 590, NULL, NULL), (89404, 'Cotton', '1990-08-05', 31, 'Akhal Teke', 'Chestnut', 'Brown', '134', 'F', 124, 234, NULL, NULL), (11811, 'Rudolph', '1998-04-20', 23, 'Quarter', 'Black', 'Black', '116', 'M', 162, 564, NULL, NULL), (89802, 'Lucy', '1990-09-05', 31, 'Quarter', 'Black', 'Black', '144', 'F', 177, 590, NULL, NULL);

Sire int, Dam int, constraint pk_racehorses primary key (HorseID), constraint fk1_racehorses foreign key (Sire) references RaceHorses(HorseID), constraint fk2_racehorses foreign key (Dam) references RaceHorses(HorseID));	

Evidence

select * from racehorses;

ata Output

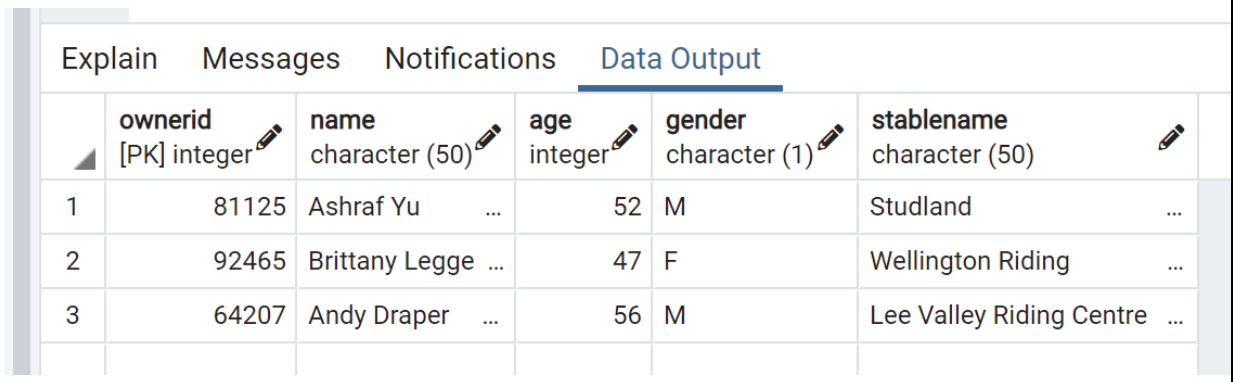
horseid [PK] integer	horsename character (50)	dateofbirth date	age integer	breed character (50)	colour character (50)	haircolour character (50)	shoesize integer	gender character (1)	height integer	weight integer	sire integer	dam integer
32048	Chips	1990-09-05	31	Akhal Teke	Chestnut	Brown	144	M	177	590	[null]	[null]
89404	Cotton	1990-08-05	31	Akhal Teke	Chestnut	Brown	134	F	124	234	[null]	[null]
11811	Rudolph	1998-04-20	23	Quarter	Black	Black	116	M	162	564	[null]	[null]
89802	Lucy	1990-09-05	31	Quarter	Black	Black	144	F	177	590	[null]	[null]
26877	Caviar	1998-11-23	23	Thoroughbred	Chestnut	Brown	116	M	162	564	[null]	[null]
49357	Luna	1990-05-03	31	Thoroughbred	Chestnut	Brown	134	F	124	234	[null]	[null]
63430	Cobra	1990-09-05	31	Thoroughbred	Chestnut	Brown	144	M	177	590	26877	49357
88859	Midnight	1990-11-06	31	Quarter	Black	Black	134	M	124	234	11811	89802
41681	Powder	1998-03-20	23	Quarter	Black	Black	116	F	162	564	11811	89802
21252	Othello	1998-12-03	22	Quarter	Black	Black	155	M	129	240	11811	89802
4648	Bubble	2000-09-21	21	Akhal Teke	Chestnut	Brown	143	M	179	632	32048	89404
39194	Money	2000-11-28	20	Akhal Teke	Chestnut	Brown	127	F	165	568	32048	89404
65394	Uno	2001-02-20	20	Akhal Teke	Chestnut	Brown	130	M	167	568	32048	89404
51724	Buck	2003-06-06	18	Akhal Teke	Chestnut	Brown	146	F	123	234	32048	89404
30548	Asti	2003-12-24	17	Tennessee Walker	Dun	Brown	118	M	123	234	[null]	[null]
81152	Alfie	2004-02-12	17	Tennessee Walker	Dun	Brown	122	M	134	30	[null]	[null]
40520	Linguini	2007-07-17	14	Standardbred	Bay	Gray	152	M	170	572	[null]	[null]
1334	Iron	2009-12-05	11	Standardbred	Bay	Gray	138	F	166	563	[null]	[null]
15393	Stormy	2010-12-05	10	Thoroughbred	Chestnut	Brown	126	M	163	563	26877	49357
85789	Money	2011-09-06	10	Thoroughbred	Chestnut	Brown	139	M	127	238	26877	49357
64558	Bubble	2013-12-17	7	Standardbred	Bay	Gray	141	F	150	402	[null]	[null]

Owner

Create Table	**Population**
create table Owner (OwnerID int NOT NULL, Name char(50), Age int, Gender char, StableName char(50) NOT NULL, constraint pk_owner primary key (OwnerID));	INSERT INTO OWNER (OWNERID, NAME, AGE, GENDER, STABLENAME) VALUES (81125, 'Ashraf Yu', 52, 'M', 'Studland'), (92465, 'Brittany Legge', 47, 'F', 'Wellington Riding'), (64207, 'Andy Draper', 56, 'M', 'Lee Valley Riding Centre');

Evidence

select * from owner;



Explain | Messages | Notifications | Data Output

	ownerid [PK] integer	name character (50)	age integer	gender character (1)	stablename character (50)
1	81125	Ashraf Yu	52	M	Studland
2	92465	Brittany Legge	47	F	Wellington Riding
3	64207	Andy Draper	56	M	Lee Valley Riding Centre

Jockey

Create Table

```
create table Jockey (
JockeyID int NOT NULL,
Name char(50),
Height int,
Weight int,
Gender char,
DateofBirth date,
Age int,
Doping boolean,
Postcode char(50),
Streetname char(50),
constraint pk_jockey primary key (JockeyID),
constraint fk1_jockey foreign key (Postcode)
references Postcode);
```

Population *appendix 3

```
INSERT INTO JOCKEY (JOCKEYID, NAME, HEIGHT,
WEIGHT, GENDER, DateOfBirth, AGE, DOPING,
POSTCODE, STREETNAME)
VALUES
    (30606, 'Aleyna Maxwell', 173, 55, 'F', '1994-02-
08', 27, True, 'RG26 4HT', '4 Searing Way, Tadley'),
    (59348, 'Borys Snow', 176, 77, 'M', '1994-06-04',
27, False, 'RG30 4SN', '6 Ash Road, Tilehurst'),
    (76066, 'Mohammod Whitworth', 183, 84, 'M',
'1995-08-24', 26, True, 'RG4 9PQ', '7 Thanksgiving
Lane, Henley-on-Thames');
```

Evidence

select * from jockey;

jockeyid [PK] integer	name character (50)	height integer	weight integer	gender character (1)	dateofbirth date	age integer	doping boolean	postcode character (50)	streetname character (50)
30606	Aleyna Maxwell	173	55	F	1994-02-08	27	false	RG26 4HT	4 Searing Way, Tadley
59348	Borys Snow	176	77	M	1994-06-04	27	false	RG30 4SN	6 Ash Road, Tilehurst
76066	Mohammod Whitworth	183	84	M	1995-08-24	26	false	RG4 9PQ	7 Thanksgiving Lane, Henley-on-Thames
73595	Asiyah Huffman	155	46	F	1997-08-26	24	false	RG7 2PH	4 Lovegrove Gardens, Silchester
32447	Winifred Ibarra	180	82	M	1997-09-14	24	false	RG5 3RP	5 Burns Close, Woodley
68923	Nichola Stacey	186	86	M	1998-04-28	23	false	RG22 4DZ	45 Grainger Close, Basingstoke
15247	Levi Wheatley	154	45	F	1998-02-22	23	false	RG27 9DF	2A Bell Meadow Road, Hook
77899	Thomas Salinas	174	74	M	1999-01-09	22	false	RG27 0DL	63 Longbridge Close, Hook
26153	Hania Miranda	168	53	F	1999-11-13	22	false	RG30 4ST	103 Corwen Road, Tilehurst
12403	Rick Morales	179	79	M	2000-10-06	21	false	RG2 6UW	60 Longwater Avenue, Reading

HorseTrainer

Create Table	**Population**
create table HorseTrainer (TrainerID int NOT NULL, Name char(50) NOT NULL, DateofBirth date NOT NULL, Gender char NOT NULL, Doping boolean, PostCode char(50), Streetname char(50), constraint pk_horsetrainer primary key (TrainerID), constraint fk1_horsetrainer foreign key (Postcode) references Postcode);	INSERT INTO horsetrainer (TRAINERID, NAME, DateOfBirth, GENDER, DOPING, POSTCODE, STREETNAME) VALUES (28819, 'Manal Mueller', '1984-11-08', 'M', True, 'RG8 9RD', 'Cintra, Streatley Hill, Streatley'), (32831, 'Alayna Norman', '1985-02-10', 'F', False, 'RG30 6UJ', '3 Wimborne Gardens, Tilehurst'), (56321, 'Asma Gray', '1987-01-09', 'F', False, 'RG40 4RZ', '18 Springdale, Finchampstead');

Evidence

select * from trainer;

Explain Messages Notifications Data Output

	trainerid [PK] integer	name character (50)	dateofbirth date	gender character (1)	doping boolean	postcode character (50)	streetname character (50)
1	28819	Manal Mueller	1984-11-08	M	true	RG8 9RD	Cintra, Streatley Hill, Streatley
2	32831	Alayna Norman	1985-02-10	F	false	RG30 6UJ	3 Wimborne Gardens, Tilehurst
3	56321	Asma Gray	1987-01-09	F	false	RG40 4RZ	18 Springdale, Finchampstead

RaceHorseDetails

Create Table	**Population** *appendix 4
create table RaceHorseDetails (HorseID int NOT NULL, JockeyID int NOT NULL, OwnerID int, TrainerID int NOT NULL, constraint pk_racehorsedetails primary key (HorseID), constraint fk_racehorsedetails foreign key (HorseID) references RaceHorses, constraint fk1_racehorsedetails foreign key (JockeyID) references Jockey, constraint fk2_racehorsedetails foreign key (OwnerID) references Owner, constraint fk3_racehorsedetails foreign key (TrainerID) references HorseTrainer);	INSERT INTO racehorsedetails (HORSEID, JOCKEYID, OWNERID, TRAINERID) VALUES (63430, 30606, 81125, 28819), (88859, 59348, 92465, 32831), (41681, 76066, 64207, 56321);

Evidence

select * from racehorsedetails;

Data Output | Explain | Messages | Notifications

	horseid [PK] integer	jockeyid integer	ownerid integer	trainerid integer
1	63430	30606	81125	28819
2	88859	59348	92465	32831
3	41681	76066	64207	56321
4	21252	73595	81125	28819
5	4648	32447	92465	32831
6	39194	68923	64207	56321
7	65394	15247	81125	28819
8	51724	77899	92465	32831
9	30548	26153	64207	56321
10	81152	12403	81125	28819
11	40520	73595	92465	32831
12	1334	59348	64207	56321
13	15393	30606	81125	28819
14	85789	59348	92465	32831
15	64558	26153	64207	56321

RaceHorseMedicalHistory

Create Table	**Population** *appendix 5
create table RaceHorseMedicalHistory(MedicalHistoryID int NOT NULL, HorseID int NOT NULL, HorseName char(50), RegisteredDoctorName char(50), HealthCondition varchar NOT NULL, constraint pk_racehorsemedicalhistory primary key (HorseID), constraint fk1_racehorsemedicalhistory foreign key (HorseID) references RaceHorses);	INSERT INTO racehorsemedicalhistory (MEDICALHISTORYID, HORSEID, REGISTEREDDOCTORNAME, HEALTHCONDITION) VALUES (50569, 63430, 'Evie-May Bone', 'Okay'), (85958, 88859, 'Annabella Bishop', 'Good'), (33793, 41681, 'Hadiya Golden', 'Okay');

Evidence

select * from racehorsemedicalhistory;

medicalhistoryid integer	horseid [PK] integer	registereddoctorname character (50)	healthcondition character varying
50569	63430	Evie-May Bone	Okay
85958	88859	Annabella Bishop	Good
33793	41681	Hadiya Golden	Okay
57044	21252	Zayden Samuels	Okay
50705	4648	Harlow Carson	Good
94676	39194	Evie-May Bone	Okay
30441	65394	Annabella Bishop	Good
10133	51724	Hadiya Golden	Okay
13665	30548	Zayden Samuels	Bad
43235	81152	Harlow Carson	Okay
81668	40520	Evie-May Bone	Good
31761	1334	Annabella Bishop	Good
77732	15393	Hadiya Golden	Bad
79654	85789	Zayden Samuels	Okay
41894	64558	Harlow Carson	Good

Races

Create Table	**Population** *appendix 6
create table Races (RaceID int NOT NULL,	INSERT INTO RACES (RACEID, RACECOURSEID, RACENAME, Date, HORSEID, JOCKEYID, PLACEMENT)

RacecourseID int NOT NULL, RaceName char(50) NOT NULL, Date date, HorseID int NOT NULL, JockeyID int NOT NULL, Placement int NOT NULL, constraint pk_races primary key (RaceID), constraint fk1_races foreign key (RacecourseID) references Racecourses, constraint fk2_races foreign key (HorseID) references RaceHorses, constraint fk3_races foreign key (JockeyID) references Jockey);	VALUES (63622, 47335, 'Coventry Stakes', '2021-06-14', 63430, 30606, 1), (52432, 90540, 'Lonsdale Cup', '2019-06-14', 88859, 59348, 3), (20852, 24871, 'King George Stakes', '2019-05-02', 41681, 76066, 5);

Evidence

select * from races;

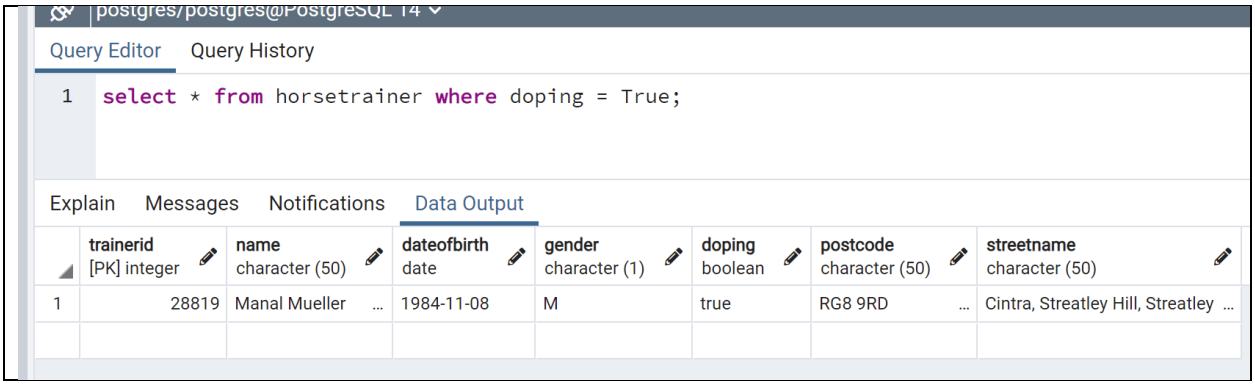
	raceid [PK] integer	racecourseid integer	racename character (50)	date date	horseid integer	jockeyid integer	placement integer
1	63622	47335	Coventry Stakes	2021-06-14	63430	30606	1
2	52432	90540	Lonsdale Cup	2019-06-14	88859	59348	3
3	20852	24871	King George Stakes	2019-05-02	41681	76066	5
4	79376	47335	Coventry Stakes	2021-06-14	21252	73595	2
5	92335	90540	Lonsdale Cup	2019-06-14	4648	32447	5
6	92548	24871	King George Stakes	2019-05-02	39194	68923	4
7	17306	47335	Coventry Stakes	2021-06-14	65394	15247	3
8	43445	90540	Lonsdale Cup	2019-06-14	51724	77899	1
9	37960	24871	King George Stakes	2019-05-02	30548	26153	3
10	28973	47335	Coventry Stakes	2021-06-14	81152	12403	4
11	32197	90540	Lonsdale Cup	2019-06-14	40520	73595	2
12	20017	24871	King George Stakes	2019-05-02	1334	59348	2
13	55788	47335	Coventry Stakes	2021-06-14	15393	30606	5
14	48534	90540	Lonsdale Cup	2019-06-14	85789	59348	4
15	1756	24871	King George Stakes	2019-05-02	64558	26153	1
16	63621	47335	Hungry Stakes	2019-06-14	63430	30606	1
17	52431	90540	Homeland Races	2020-06-14	88859	59348	3
18	20851	24871	Queens Gambit	2019-06-14	41681	76066	5
19	79371	47335	Hungry Stakes	2019-06-14	21252	73595	2
20	92331	90540	Homeland Races	2020-06-14	4648	32447	5
21	92541	24871	Queens Gambit	2019-06-14	39194	68923	4
22	17301	47335	Hungry Stakes	2019-06-14	65394	15247	3
23	43441	90540	Homeland Races	2020-06-14	51724	77899	1
24	37961	24871	Queens Gambit	2019-06-14	30548	26153	3
25	28971	47335	Hungry Stakes	2019-06-14	81152	12403	4
26	32191	90540	Homeland Races	2020-06-14	40520	73595	2
27	20011	24871	Queens Gambit	2019-06-14	1334	59348	2
28	55781	47335	Hungry Stakes	2019-06-14	15393	30606	5
29	48531	90540	Homeland Races	2020-06-14	85789	59348	4
30	1751	24871	Queens Gambit	2019-06-14	64558	26153	1

Racecourses	
Create Table	**Population**
create table Racecourses (RacecourseID int NOT NULL, RacecourseName char(50) NOT NULL, EventName char(50) NOT NULL, Location char(50) NOT NULL, RaceHorseCapacity int NOT NULL, AudienceCapacity int NOT NULL, constraint pk_racecourses primary key (RacecourseID));	INSERT INTO RACECOURSES (RACECOURSEID, RACECOURSENAME, EVENTNAME, LOCATION, RACEHORSECAPACITY, AUDIENCECAPACITY) VALUES (47335, 'Ascot', 'Marathons', 'Ascot, Berkshire, England', 5, 20000), (90540, 'York Racecourse', 'Wokingham', 'York, England', 5, 20000), (24871, 'Goodwood Racecourse', 'Reading', 'W. Sussex, England', 5, 20000);
Evidence	
select * from racecourses;	

	racecourseid [PK] integer	racecoursename character (50)	eventname character (50)	location character (50)	racehorsecapacity integer	audiencecapacity integer
1	47335	Ascot	Marathons	Ascot, Berkshire, England	5	20000
2	90540	York Racecourse	Wokingham	York, England	5	20000
3	24871	Goodwood Racecourse	Reading	W. Sussex, England	5	20000

Testing

Test 1: Which trainers are suspected of doping?
Test Description
The objective of this test is to retrieve all horse trainers suspected of doping. The query to execute is the following: select * from horsetrainer where doping = True;
Expected Outcome
Only 1 trainer has doped
Test Evaluation
The test is considered successful because the query retrieved the expected data
Test Evidence



Test 2: Produce a league table of winning trainers.

Test Description

The objective of this test is to produce a league table of all winning trainers.
The query to produce that is the following:

```
select races.placement,  races.racename, racehorsedetails.trainerID, horsetrainer.name,
racehorsedetails.horseID
from racehorsedetails join races
on racehorsedetails.horseID = races.horseID
join horsetrainer on racehorsedetails.trainerID = horsetrainer.trainerID
where races.placement = 1;
```

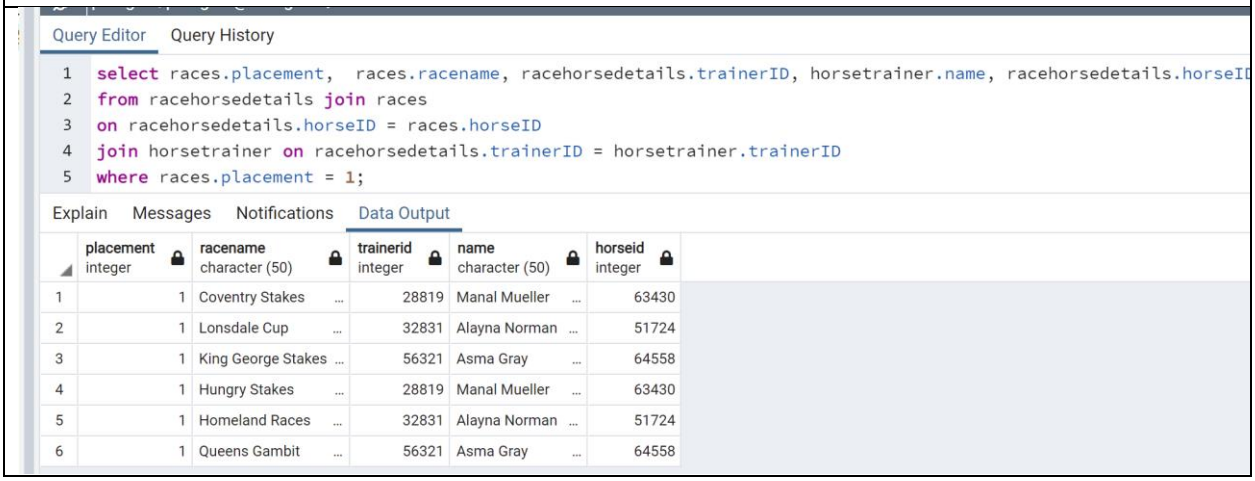
Expected Outcome

Because we have 6 races, we will get 6 winners

Test Evaluation

The test is considered successful because the query retrieved the expected data

Test Evidence



Test 3: List all winners of Reading and Wokingham events in the same year.

Test Description

The objective of this test is to list the winners in the Reading and Wokingham events in 2019.
The query for this test is the following:

```
select races.date,
```

```
racecourses.eventname,
races.racename,
racehorsedetails.jockeyID,
racehorsedetails.trainerID,
racehorsedetails.horseID,
races.placement
from racehorsedetails join races
on racehorsedetails.horseID = races.horseID
join racecourses
on racecourses.racecourseID = races.racecourseID
where placement = 1 and 2019 = (select extract(year from date))
and racecourses.eventnamein  ('Reading', 'Wokingham');
```

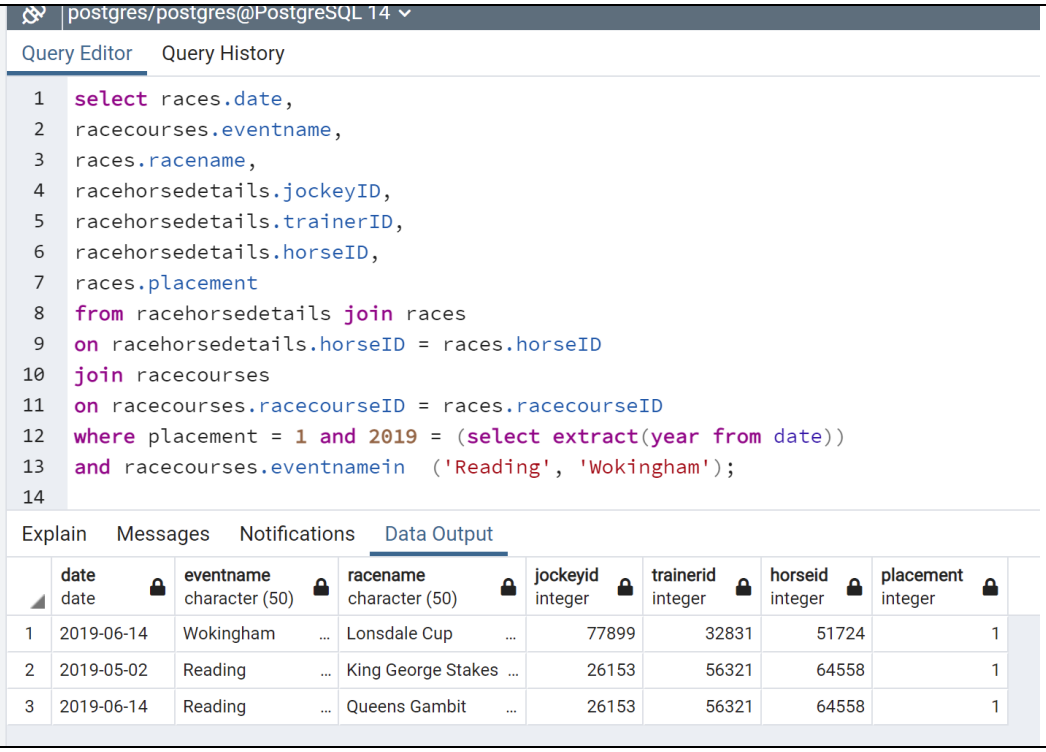
Expected Outcome

Multiple winners for each event (one winner per each race)

Test Evaluation

The test is considered successful because the query retrieved the expected data

Test Evidence



postgres/postgres@PostgreSQL 14

Query Editor | Query History

```
1  select races.date,
2  racecourses.eventname,
3  races.racename,
4  racehorsedetails.jockeyID,
5  racehorsedetails.trainerID,
6  racehorsedetails.horseID,
7  races.placement
8  from racehorsedetails join races
9  on racehorsedetails.horseID = races.horseID
10 join racecourses
11 on racecourses.racecourseID = races.racecourseID
12 where placement = 1 and 2019 = (select extract(year from date))
13 and racecourses.eventnamein  ('Reading', 'Wokingham');
14
```

Explain | Messages | Notifications | Data Output

	date date	eventname character (50)	racename character (50)	jockeyid integer	trainerid integer	horseid integer	placement integer
1	2019-06-14	Wokingham	Lonsdale Cup	77899	32831	51724	1
2	2019-05-02	Reading	King George Stakes	26153	56321	64558	1
3	2019-06-14	Reading	Queens Gambit	26153	56321	64558	1

Appendix

Appendix 1

```
INSERT INTO RACEHORSES(HORSEID, HORSENAME, DateOfBirth, AGE, BREED, COLOUR, HAIRCOLOUR, SHOESIZE, GENDER, HEIGHT, WEIGHT, SIRE, DAM)

VALUES

        (32048, 'Chips', '1990-09-05', 31, 'Akhal Teke', 'Chestnut', 'Brown', '144', 'M', 177, 590, NULL, NULL),

   (89404, 'Cotton', '1990-08-05', 31, 'Akhal Teke', 'Chestnut', 'Brown', '134', 'F', 124, 234, NULL, NULL),

   (11811, 'Rudolph', '1998-04-20', 23, 'Quarter', 'Black', 'Black', '116', 'M', 162, 564, NULL, NULL),

   (89802, 'Lucy', '1990-09-05', 31, 'Quarter', 'Black', 'Black', '144', 'F', 177, 590, NULL, NULL),

(26877, 'Caviar', '1998-11-23', 23, 'Thoroughbred', 'Chestnut', 'Brown', '116', 'M', 162, 564, NULL, NULL),

   (49357, 'Luna', '1990-05-03', 31, 'Thoroughbred', 'Chestnut', 'Brown', '134', 'F', 124, 234, NULL, NULL);

   (63430, 'Cobra', '1990-09-05', 31, 'Thoroughbred', 'Chestnut', 'Brown', 144, 'M', 177, 590, 26877, 49357),

   (88859, 'Midnight','1990-11-06', 31, 'Quarter', 'Black', 'Black', 134, 'M', 124, 234, 11811, 89802),

   (41681, 'Powder', '1998-03-20', 23, 'Quarter', 'Black', 'Black', 116, 'F', 162, 564, 11811, 89802),

   (21252, 'Othello', '1998-12-03', 22, 'Quarter', 'Black', 'Black', 155, 'M', 129, 240, 11811, 89802),

   (04648, 'Bubble', '2000-09-21', 21, 'Akhal Teke', 'Chestnut', 'Brown', 143, 'M', 179, 632, 32048, 89404),

   (39194, 'Money', '2000-11-28', 20, 'Akhal Teke', 'Chestnut', 'Brown', 127, 'F', 165, 568, 32048, 89404),

   (65394, 'Uno', '2001-02-20', 20, 'Akhal Teke', 'Chestnut', 'Brown', 130, 'M', 167, 568, 32048, 89404),

   (51724, 'Buck', '2003-06-06', 18, 'Akhal Teke', 'Chestnut', 'Brown', 146, 'F', 123, 234, 32048, 89404),

   (30548, 'Asti', '2003-12-24', 17, 'Tennessee Walker', 'Dun', 'Brown', 118, 'M', 123, 234, NULL, NULL),

   (81152, 'Alfie', '2004-02-12', 17, 'Tennessee Walker', 'Dun', 'Brown', 122, 'M', 134, 30, NULL, NULL),

   (40520, 'Linguini', '2007-07-17', 14, 'Standardbred', 'Bay', 'Gray', 152, 'M', 170, 572, NULL, NULL),

   (01334, 'Iron', '2009-12-05', 11, 'Standardbred', 'Bay', 'Gray', 138, 'F', 166, 563, NULL, NULL),

   (15393, 'Stormy', '2010-12-05', 10, 'Thoroughbred', 'Chestnut', 'Brown', 126, 'M', 163, 563, 26877, 49357),
```

```
    (85789, 'Money', '2011-09-06', 10, 'Thoroughbred', 'Chestnut', 'Brown', 139, 'M', 127, 238, 26877, 49357),
    (64558, 'Bubble', '2013-12-17', 7, 'Standardbred', 'Bay', 'Gray', 141, 'F', 150, 402, NULL, NULL);
```

Appendix 2

```
insert into postcode (postcode, city)
values ('RG26 4HT', 'Reading'),
('RG30 4SN', 'Reading'),
('RG4 9PQ', 'Reading'),
('RG7 2PH', 'Reading'),
('RG22 4DZ', 'Reading'),
('RG5 3RP', 'Reading'),
('RG27 9DF', 'Reading'),
('RG30 4ST', 'Reading'),
('RG2 6UW', 'Reading'),
('RG8 9RD', 'Reading'),
('RG30 6UJ', 'Reading'),
('RG27 0DL', Reading),
 ('RG40 4RZ', 'Reading');
```

Appendix 3

```
INSERT INTO JOCKEY (JOCKEYID, NAME, HEIGHT, WEIGHT, GENDER, DateOfBirth, AGE, DOPING, POSTCODE, STREETNAME)
VALUES
    (30606, 'Aleyna Maxwell', 173, 55, 'F', '1994-02-08', 27, True, 'RG26 4HT', '4 Searing Way, Tadley'),
    (59348, 'Borys Snow', 176, 77, 'M', '1994-06-04', 27, False, 'RG30 4SN', '6 Ash Road, Tilehurst'),
    (76066, 'Mohammod Whitworth', 183, 84, 'M', '1995-08-24', 26, True, 'RG4 9PQ', '7 Thanksgiving Lane, Henley-on-Thames'),
    (73595, 'Asiyah Huffman', 155, 46, 'F', '1997-08-26', 24, False, 'RG7 2PH', '4 Lovegrove Gardens, Silchester'),
```

```
    (32447, 'Winifred Ibarra', 180, 82, 'M', '1997-09-14', 24, False, 'RG5 3RP', '5 Burns Close, Woodley'),

    (68923, 'Nichola Stacey', 186, 86, 'M', '1998-04-28', 23, False, 'RG22 4DZ', '45 Grainger Close, Basingstoke'),

    (15247, 'Levi Wheatley', 154, 45, 'F', '1998-02-22', 23, True, 'RG27 9DF', '2A Bell Meadow Road, Hook'),

    (77899, 'Thomas Salinas', 174, 74, 'M', '1999-01-09', 22, False, 'RG27 0DL', '63 Longbridge Close, Hook'),

    (26153, 'Hania Miranda', 168, 53, 'F', '1999-11-13', 22, False, 'RG30 4ST', '103 Corwen Road, Tilehurst'),

    (12403, 'Rick Morales', 179, 79, 'M', '2000-10-06', 21, False, 'RG2 6UW', '60 Longwater Avenue, Reading');
```

Appendix 4

```
INSERT INTO racehorsedetails (HORSEID, JOCKEYID, OWNERID, TRAINERID)

VALUES

    (63430, 30606, 81125, 28819),

    (88859, 59348, 92465, 32831),

    (41681, 76066, 64207, 56321),

    (21252, 73595, 81125, 28819),

    (04648, 32447, 92465, 32831),

    (39194, 68923, 64207, 56321),

    (65394, 15247, 81125, 28819),

    (51724, 77899, 92465, 32831),

    (30548, 26153, 64207, 56321),

    (81152, 12403, 81125, 28819),

    (40520, 73595, 92465, 32831),

    (01334, 59348, 64207, 56321),

    (15393, 30606, 81125, 28819),

    (85789, 59348, 92465, 32831),

    (64558, 26153, 64207, 56321);
```

Appendix 5

```
INSERT INTO racehorsemedicalhistory (MEDICALHISTORYID, HORSEID, REGISTEREDDOCTORNAME, HEALTHCONDITION)
VALUES
    (50569, 63430, 'Evie-May Bone', 'Okay'),
    (85958, 88859, 'Annabella Bishop', 'Good'),
    (33793, 41681, 'Hadiya Golden', 'Okay'),
    (57044, 21252, 'Zayden Samuels', 'Okay'),
    (50705, 04648, 'Harlow Carson', 'Good'),
    (94676, 39194, 'Evie-May Bone', 'Okay'),
    (30441, 65394, 'Annabella Bishop', 'Good'),
    (10133, 51724, 'Hadiya Golden', 'Okay'),
    (13665, 30548, 'Zayden Samuels', 'Bad'),
    (43235, 81152, 'Harlow Carson', 'Okay'),
    (81668, 40520, 'Evie-May Bone', 'Good'),
    (31761, 01334, 'Annabella Bishop', 'Good'),
    (77732, 15393, 'Hadiya Golden', 'Bad'),
    (79654, 85789, 'Zayden Samuels', 'Okay'),
    (41894, 64558, 'Harlow Carson', 'Good');
```

Appendix 6

```
INSERT INTO RACES (RACEID, RACECOURSEID, RACENAME, Date, HORSEID, JOCKEYID, PLACEMENT)
VALUES
    (63622, 47335, 'Coventry Stakes', '2021-06-14', 63430, 30606, 1),
    (52432, 90540, 'Lonsdale Cup', '2019-06-14', 88859, 59348, 3),
    (20852, 24871, 'King George Stakes', '2019-05-02', 41681, 76066, 5),
    (79376, 47335, 'Coventry Stakes', '2021-06-14', 21252, 73595, 2),
    (92335, 90540, 'Lonsdale Cup', '2019-06-14', 04648, 32447, 5),
    (92548, 24871, 'King George Stakes', '2019-05-02', 39194, 68923, 4),
```

```
(17306, 47335, 'Coventry Stakes', '2021-06-14', 65394, 15247, 3),
(43445, 90540, 'Lonsdale Cup', '2019-06-14', 51724, 77899, 1),
(37960, 24871, 'King George Stakes', '2019-05-02', 30548, 26153, 3),
(28973, 47335, 'Coventry Stakes', '2021-06-14', 81152, 12403, 4),
(32197, 90540, 'Lonsdale Cup', '2019-06-14', 40520, 73595, 2),
(20017, 24871, 'King George Stakes', '2019-05-02', 01334, 59348, 2),
(55788, 47335, 'Coventry Stakes', '2021-06-14', 15393, 30606, 5),
(48534, 90540, 'Lonsdale Cup', '2019-06-14', 85789, 59348, 4),
(01756, 24871, 'King George Stakes', '2019-05-02', 64558, 26153, 1),
    (63621, 47335, 'Hungry Stakes', '2019-06-14', 63430, 30606, 1),
    (52431, 90540, 'Homeland Races', '2020-06-14', 88859, 59348, 3),
    (20851, 24871, 'Queens Gambit', '2019-06-14', 41681, 76066, 5),
    (79371, 47335, 'Hungry Stakes', '2019-06-14', 21252, 73595, 2),
    (92331, 90540, 'Homeland Races', '2020-06-14', 04648, 32447, 5),
    (92541, 24871, 'Queens Gambit', '2019-06-14', 39194, 68923, 4),
    (17301, 47335, 'Hungry Stakes', '2019-06-14', 65394, 15247, 3),
    (43441, 90540, 'Homeland Races', '2020-06-14', 51724, 77899, 1),
    (37961, 24871, 'Queens Gambit', '2019-06-14', 30548, 26153, 3),
    (28971, 47335, 'Hungry Stakes', '2019-06-14', 81152, 12403, 4),
    (32191, 90540, 'Homeland Races', '2020-06-14', 40520, 73595, 2),
    (20011, 24871, 'Queens Gambit', '2019-06-14', 01334, 59348, 2),
    (55781, 47335, 'Hungry Stakes', '2019-06-14', 15393, 30606, 5),
    (48531, 90540, 'Homeland Races', '2020-06-14', 85789, 59348, 4),
    (01751, 24871, 'Queens Gambit', '2019-06-14', 64558, 26153, 1);
```

Figure 1

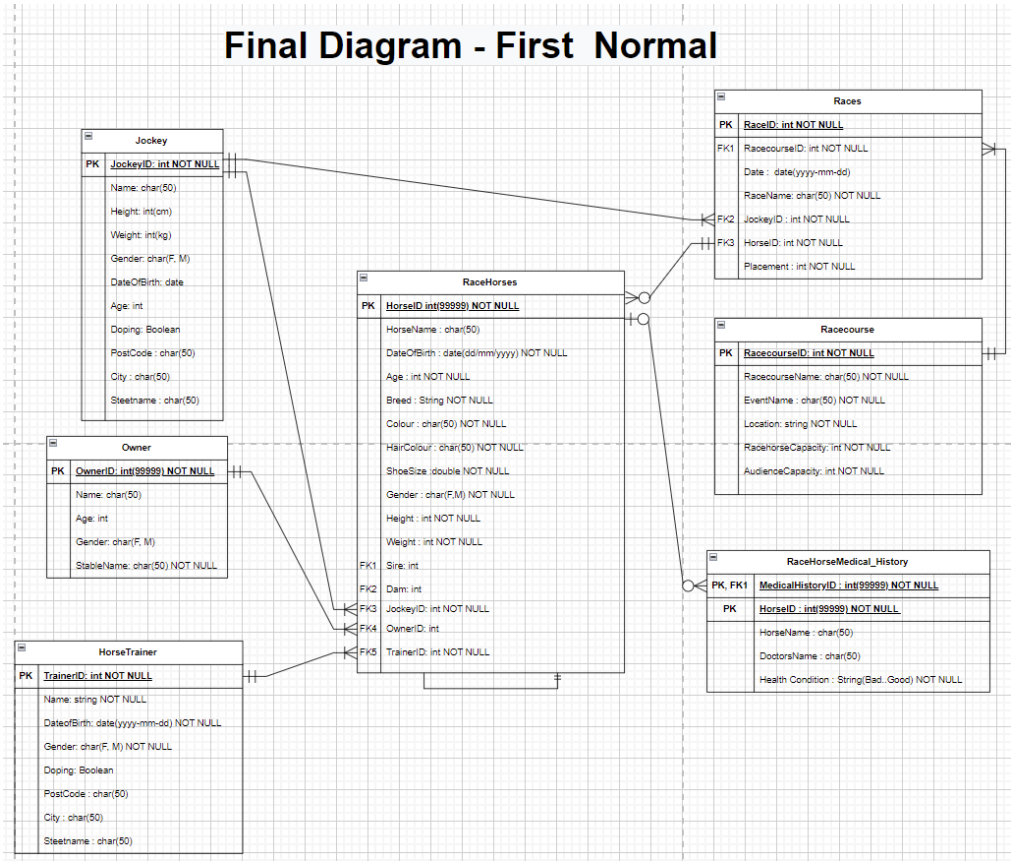

Final Diagram - First Normal
Jockey
PK JockeyID: int NOT NULL
Name: char(50)
Height: int(cm)
Weight: int(kg)
Gender: char(F, M)
DateOfBirth: date
Age: int
Doping: Boolean
PostCode : char(50)
City : char(50)
Steetname : char(50)
Races
PK RaceID: int NOT NULL
FK1 RacecourseID: int NOT NULL
Date : date(yyyy-mm-dd)
RaceName: char(50) NOT NULL
FK2 JockeyID : int NOT NULL
FK3 HorseID: int NOT NULL
Placement : int NOT NULL
RaceHorses
PK HorseID int(99999) NOT NULL
HorseName : char(50)
DateOfBirth : date(dd/mm/yyyy) NOT NULL
Age : int NOT NULL
Breed : String NOT NULL
Colour : char(50) NOT NULL
HairColour : char(50) NOT NULL
ShoeSize :double NOT NULL
Gender : char(F,M) NOT NULL
Height : int NOT NULL
Weight : int NOT NULL
FK1 Sire: int
FK2 Dam: int
FK3 JockeyID: int NOT NULL
FK4 OwnerID: int
FK5 TrainerID: int NOT NULL
Racecourse
PK RacecourseID: int NOT NULL
RacecourseName: char(50) NOT NULL
EventName : char(50) NOT NULL
Location: string NOT NULL
RacehorseCapacity: int NOT NULL
AudienceCapacity: int NOT NULL
Owner
PK OwnerID: int(99999) NOT NULL
Name: char(50)
Age: int
Gender: char(F, M)
StableName: char(50) NOT NULL
RaceHorseMedical_History
PK, FK1 MedicalHistoryID : int(99999) NOT NULL
PK HorseID : int(99999) NOT NULL
HorseName : char(50)
DoctorsName : char(50)
Health Condition : String(Bad..Good) NOT NULL
HorseTrainer
PK TrainerID: int NOT NULL
Name: string NOT NULL
DateofBirth: date(yyyy-mm-dd) NOT NULL
Gender: char(F, M) NOT NULL
Doping: Boolean
PostCode : char(50)
City : char(50)
Steetname : char(50)

Figure 2

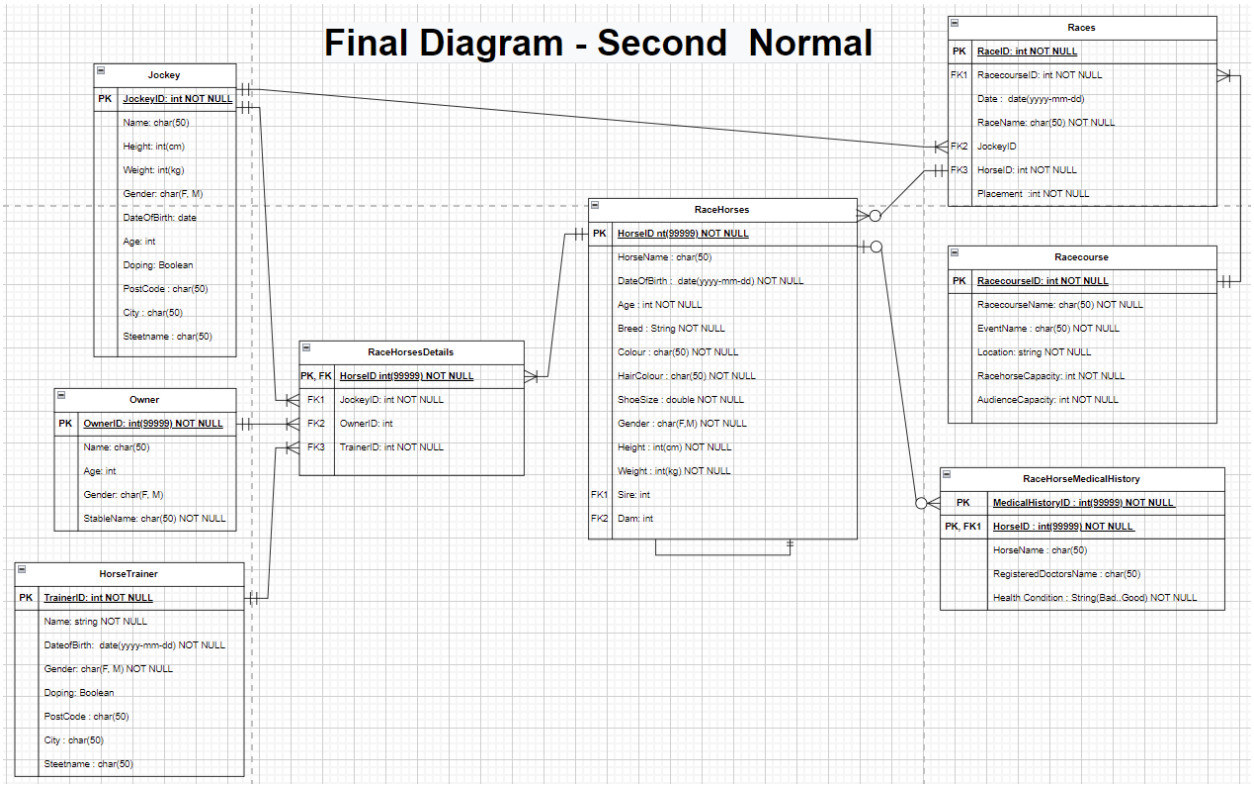


Figure 3

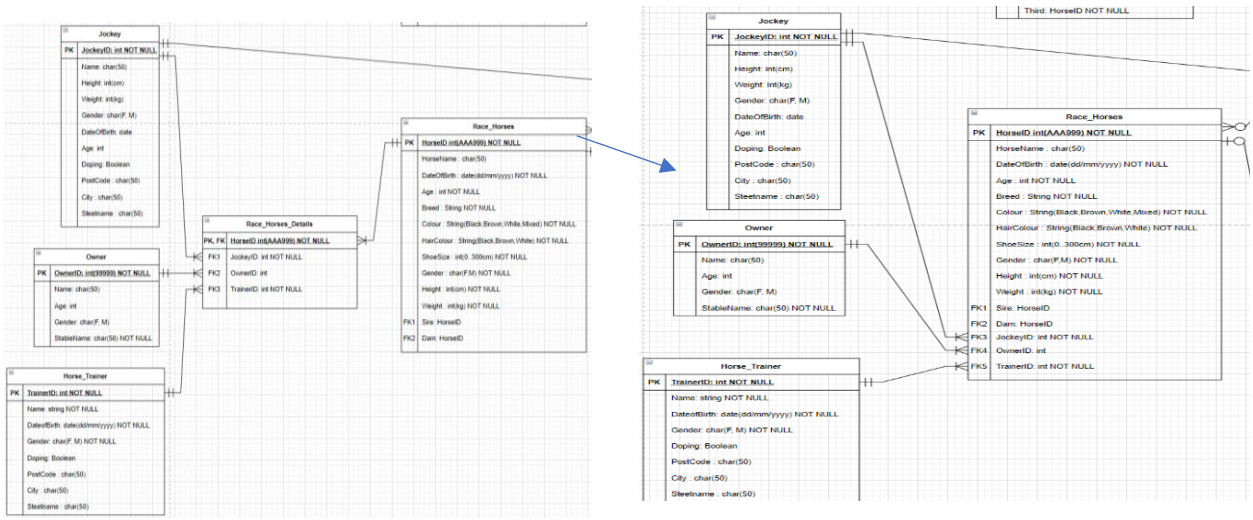
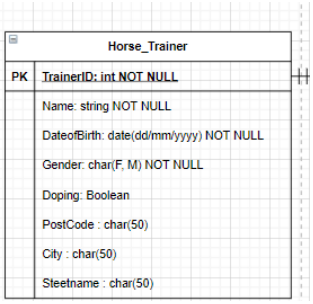
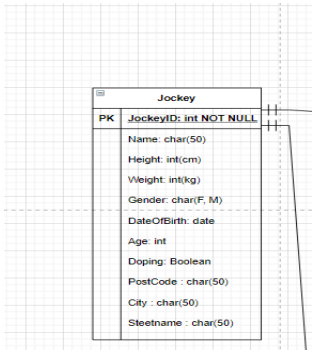
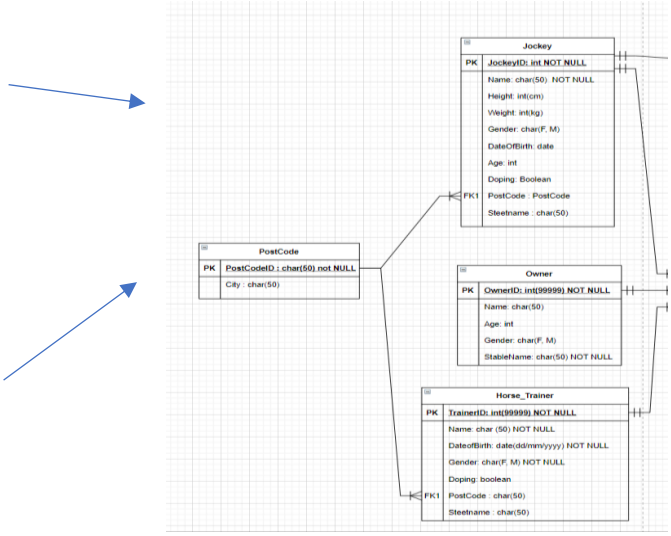


Figure 4


Horse_Trainer
PK
TrainerID: int NOT NULL
Name: string NOT NULL
DateofBirth: date(dd/mm/yyyy) NOT NULL
Gender: char(F, M) NOT NULL
Doping: Boolean
PostCode : char(50)
City : char(50)
Steetname : char(50)


Jockey
PK
JockeyID: int NOT NULL
Name: char(50)
Height: int(cm)
Weight: int(kg)
Gender: char(F, M)
DateOfBirth: date
Age: int
Doping: Boolean
PostCode : char(50)
City : char(50)
Steetname : char(50)


Jockey
PK
JockeyID: int NOT NULL
Name: char(50) NOT NULL
Height: int(cm)
Weight: int(kg)
Gender: char(F, M)
DateOfBirth: date
Age: int
Doping: Boolean
FK1
PostCode : PostCode
Steetname : char(50)
PostCode
PK
PostCodeID : char(50) not NULL
City : char(50)
Owner
PK
OwnerID: int(99999) NOT NULL
Name: char(50)
Age: int
Gender: char(F, M)
StableName: char(50) NOT NULL
Horse_Trainer
PK
TrainerID: int(99999) NOT NULL
Name: char (50) NOT NULL
DateofBirth: date(dd/mm/yyyy) NOT NULL
Gender: char(F, M) NOT NULL
Doping: boolean
FK1
PostCode : char(50)
Steetname : char(50)

References

https://www.racingexplained.co.uk/racehorses/all-about-the-thoroughbred/#:~:text=Jump%20racehorses%20generally%20don't,age%20of%20two%20years%20old. Racing age

https://www.fao.org/dad-is/browse-by-country-and-species/en/ horse breeds

https://www.richardashhorseshoes.co.uk/horseshoe-size-guides/ shoe sizes

https://www.britishhorseracing.com/ examples of horses

Horse Racing Cards, Results & Betting | Racing Post horse racing

https://www.racingandsports.com/