Database Report – Motorsport racing series

id = 15316452

The database I decided to model was for a motorsport racing series, consisting of tables that hold data relating to the drivers, the teams, the mechanics, the events, the results of drivers at those events and the cars used by the drivers. While such a database would contain much more data and many more tables in the real world, it would have been unrealistic to model such a system for the given assignment. The resulting database is really more of a sub-section of what such a database would look like if it was to be put to use in the real world.

The database consists of a car, driver, team, mechanic, event, car_race and driver_result table and these are discussed below.

The car table has a car_id, an engine number and a driver number value. The car_id acts as the primary key (the engine number also could have been the primary key). The driver number is a foreign key, refering to the driver table. (I have added in a constraint that the driver number value be unique, as a driver has a specific car they use).

The driver table has a driver number (the primary key), a name, age, blood type, street, town, country and team_id (a foreign key refering to the team table). (Note: multiple drivers can be on the same team).

The mechanic table has an id (the primary key), a name, team id (a foreign key), age, driver number (a foreign key – a mechanic supports one driver and a driver can be supported by many mechanics), a car id (a foreign key – the car that they service for their driver) and a blood type.

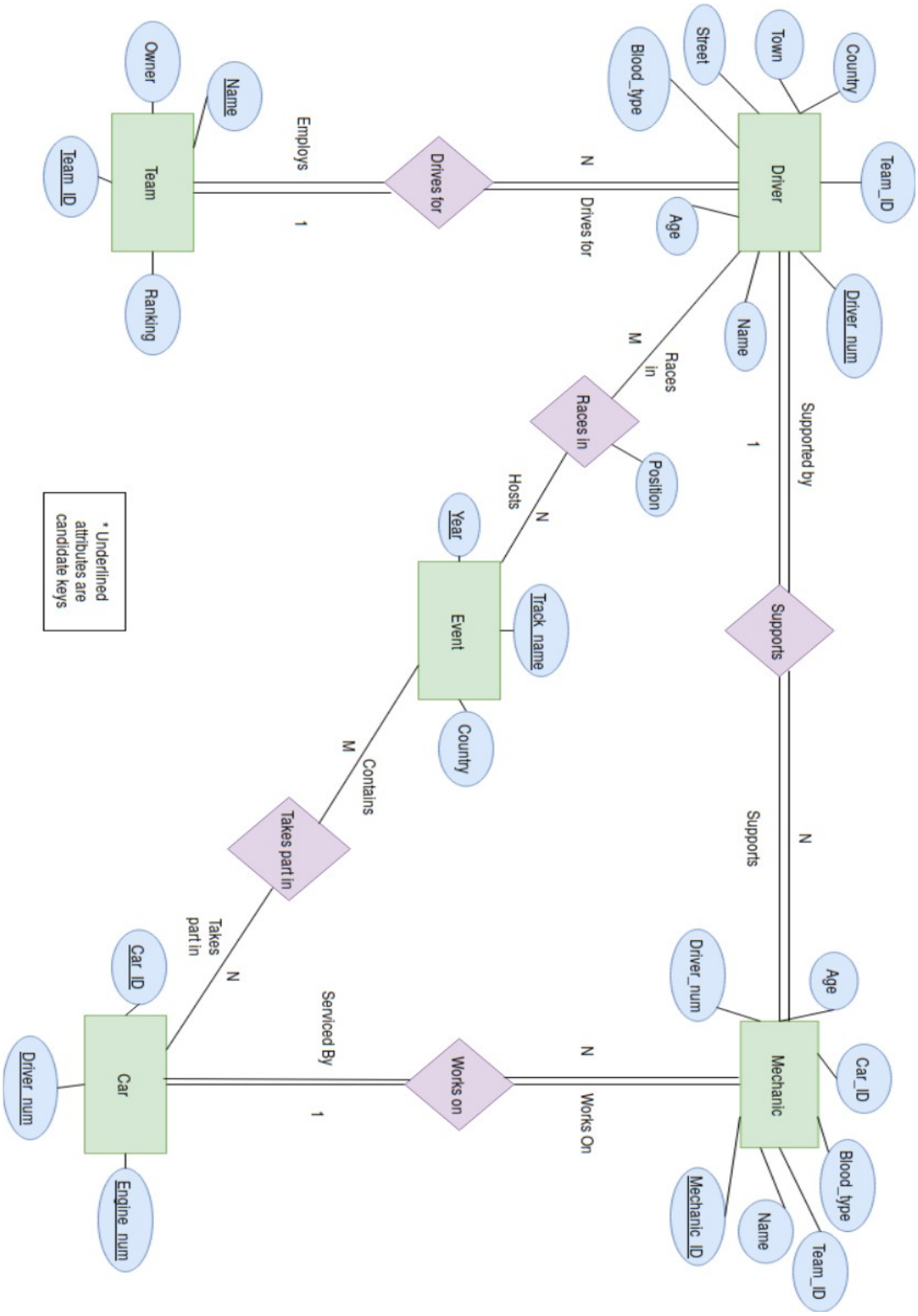
The team table has a team id (primary key), a name, ranking, and an owner value.

The event table has a track name, year and country value. The primary key is a composite of the track name and year value (events can return to the same track, but wont do so in the same year).

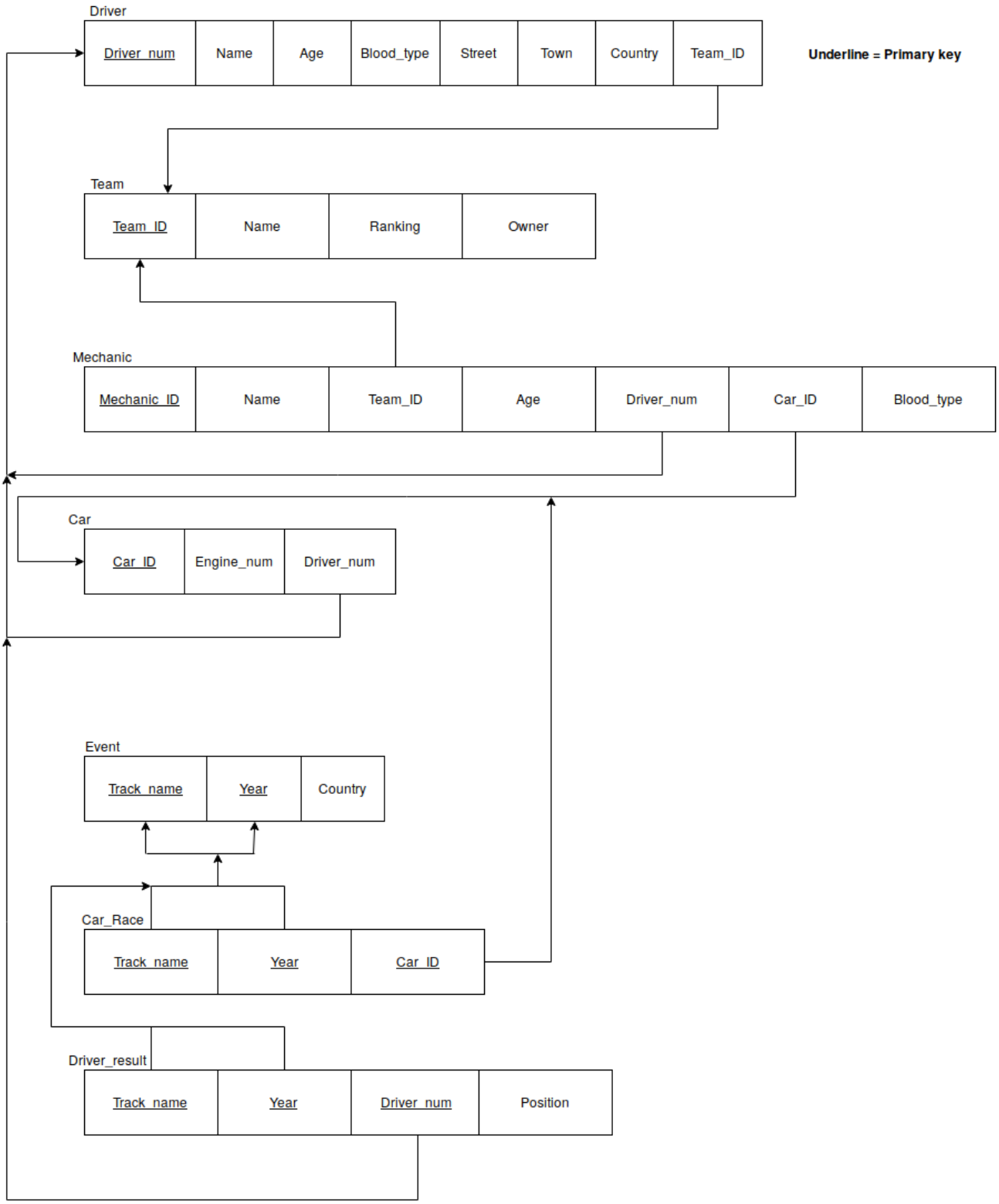
The driver_result table has a track name, year, driver number and position value. (The track name and year values are a foreign key refering to the event table and the driver number is a foreign key refering to the driver table).

The car_race table is used to determine if a car was used at a given event. It has a track name, year and car_id value. (The track name and year values are a foreign key refering to the event table and the car_id is a foreign key refering to the car table).

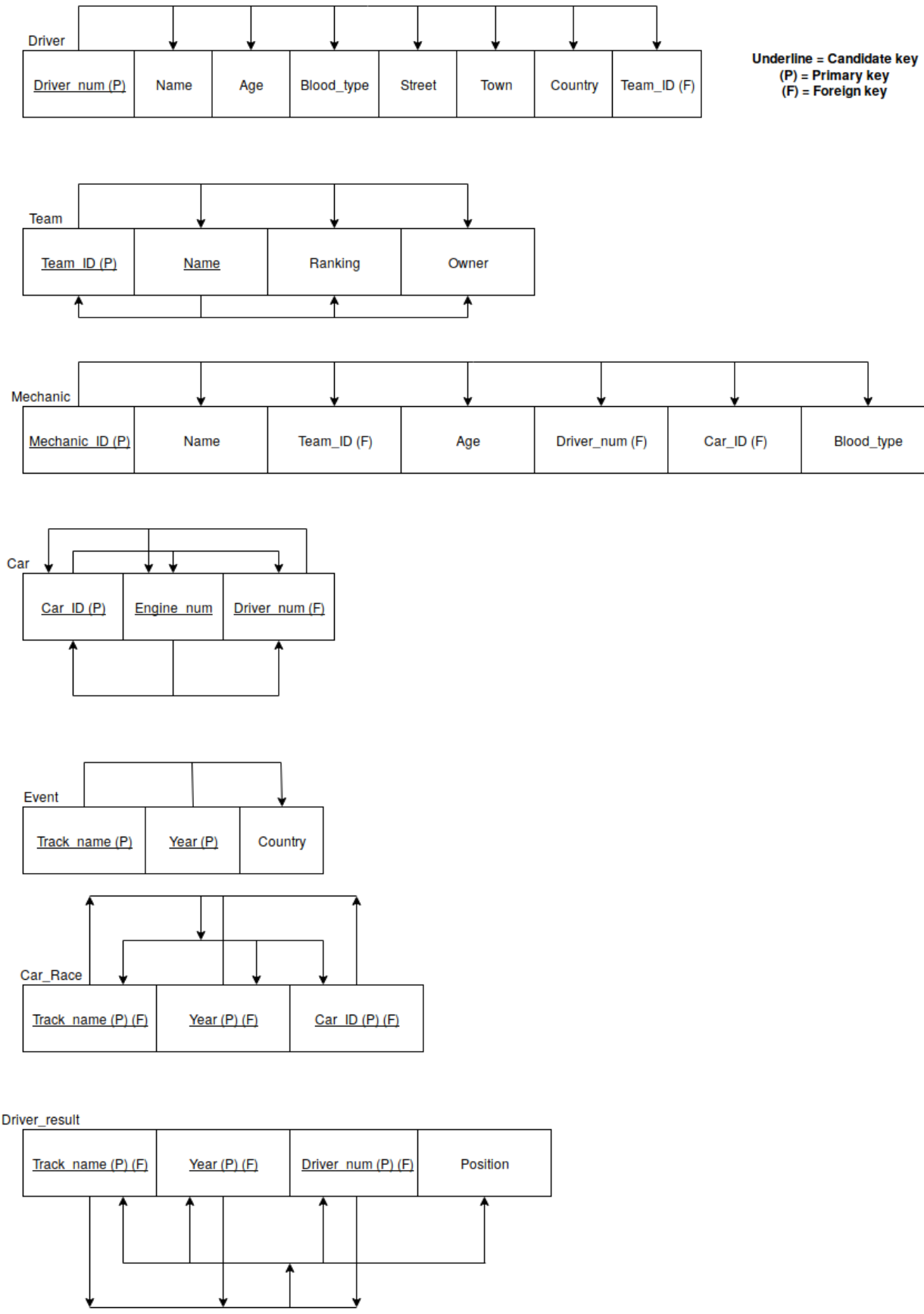
Entity Relationship Diagram



Relational Schema



Functional Dependency Diagram



Normalization:
The majority of tables were already normalized as a result of the attributes within those tables and the constraints within the database and those determined by the data it would be storing. Two tables were added in to represent to two M-N relationships (one between the driver and event table and another between the event and car table).

Semantic Constraints:
The first semantic constraint implemented across almost every value in the database is that it can't be null. This is as a result of the nature of a motorsport series, i.e teams need drivers, driver need mechanics, drivers will always have a result if they enter a race, etc.

In the team table I have included constraints that the name and ranking be unique and that the ranking must be between 1 and the number of rows in the team table (a team can't be 10^{th} if there's only 5 teams).

In the driver and mechanic I have constraints that check they are over of a possible age (older than 16 and no older than 100). I also check that they are in one of the possible 8 blood type groups.

In the car table, I check that the engine number and driver number be unique (as the engine number uniquely identifies the car and a driver can only have one car so their driver number should only appear once in the table).

In the event, driver_result and car_race tables, I check that the year is reasonable (between 1970 and 2050). In the driver_result table, I have a constraint that ensures the position is a positive number.

I also have a trigger on the driver table and mechanic table that ensures the blood type can't be changed (as in the real world). If there is an attempt made to update the blood type an error message is returned saying 'Warning: cannot change blood type'. The code for this is shown later (examples).

Security:
To add some degree of security to the described database, the intended security policy was to give members of the public and race officials two different roles.
The public should not be allowed to see sensitive data (mainly blood types) and should only be given 'select' privileges (they shouldn't be allowed to alter the data). The race officials on the other hand should be able to conduct most operations to the database and should be allowed to see sensitive information, as they may need to know it.
The intent was to create to roles that meet this description and assign them to users as needed (see below):

```
CREATE ROLE Race_Official IDENTIFIED BY pirelli;
GRANT CREATE TO Race_Official;
GRANT UPDATE TO Race_Official;
GRANT DELETE TO Race_Official;

CREATE ROLE Public_member;
GRANT SELECT TO Public_member;
REVOKE SELECT(Blood_type), SELECT (Owner), SELECT (Street), SELECT (Town)
FROM Public_member;

GRANT Race_Official to official_1;
GRANT Public_member to public_1;
```

View:
A view was also created in order to show what would be the most commonly requested data (a driver's number, the track they raced on, the year it was and what position they got).
See the code in the examples section.

Examples

The view as described above:

```
CREATE VIEW Driver_name_and_results AS SELECT Name, Driver.Driver_num,
Track_name, Year, Position FROM Driver, Driver_result WHERE Driver.Driver_num =
Driver_result.Driver_num;
```

Use of relational select, table joins (across more than one table) operations:

```
SELECT Driver.Driver_num, Driver.name, Car.Car_ID, Car.Engine_num FROM Driver
LEFT JOIN Car ON Driver.Driver_num = Car.Driver_num ORDER BY Driver.Driver_num;

SELECT Team.name, Driver.name FROM Team LEFT JOIN Driver on Driver.Team_ID =
Team.Team_ID ORDER BY Team.name;
```

Update operations:

```
UPDATE Driver SET Age = 29 WHERE Name = 'Tim Parker';
UPDATE Driver SET Name = 'Joe Maden' WHERE Name = 'Tim Parker';
UPDATE Driver SET Age = 28 WHERE Name = 'Joe Maden';
UPDATE Driver SET Name = 'Tim Parker' WHERE Name = 'Joe Maden';
```

Note: These operations are simply examples of updates and don't really serve a purpose, seen as they just update Tim Parker's age, change his name, and then revert the changes.

Trigger command:

```
DELIMITER $$

CREATE TRIGGER blood_update_check_driver
        BEFORE UPDATE ON Driver
        FOR EACH ROW BEGIN
        IF (NEW.Blood_type != OLD.Blood_type) THEN
                SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Warning: cannot
                change blood type';
        END IF;
END$$

DELIMITER ;


DELIMITER $$

CREATE TRIGGER blood_update_check_mechanic
```

```
        BEFORE UPDATE ON Mechanic
        FOR EACH ROW BEGIN
        IF (NEW.Blood_type != OLD.Blood_type) THEN
                SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Warning: cannot
                change blood type';
        END IF;
END$$

DELIMITER ;
```

Note: The following commands will cause the triggers to execute:

```
        UPDATE Driver SET Blood_type = 'O-' WHERE Name = 'Tim Parker';
        UPDATE Mechanic SET Blood_type = 'O-' WHERE Name = 'Dave Knight';
```

APPENDIX (Listing of Database Content):

```
CREATE TABLE Team (Team_ID INT NOT NULL, Name VARCHAR(255) NOT NULL, Ranking INT NOT NULL, Owner VARCHAR(255) NOT NULL, PRIMARY KEY (Team_ID), UNIQUE (Name), UNIQUE (Ranking), CONSTRAINT check_ranking CHECK (Ranking >=1 AND Ranking <= (COUNT(*))));

CREATE TABLE Driver (Driver_num INT NOT NULL, Name VARCHAR(255) NOT NULL, Age INT NOT NULL, Blood_type VARCHAR(3) NOT NULL, Street VARCHAR(255) NOT NULL, Town VARCHAR(255) NOT NULL, Country VARCHAR(100) NOT NULL, Team_ID INT NOT NULL, PRIMARY KEY (Driver_num), FOREIGN KEY (Team_ID) REFERENCES Team (Team_ID), CONSTRAINT check_age CHECK (Age >= 16 AND Age <= 100), CONSTRAINT check_blood_type CHECK (Blood_type IN ('O+','O-','A+','A-','B+','B-','AB+','AB-')));

ALTER TABLE Team DROP INDEX Ranking;

CREATE TABLE Car (Car_ID INT NOT NULL, Engine_num INT NOT NULL, Driver_num INT NOT NULL, PRIMARY KEY (Car_ID), UNIQUE (Engine_num), UNIQUE (Driver_num), FOREIGN KEY (Driver_num) REFERENCES Driver (Driver_num));

CREATE TABLE Mechanic (Mechanic_ID INT NOT NULL, Name VARCHAR(255) NOT NULL, Team_ID INT NOT NULL, Age INT NOT NULL, Driver_num INT NOT NULL, Car_ID INT NOT NULL, Blood_type VARCHAR(3) NOT NULL, PRIMARY KEY (Mechanic_ID), FOREIGN KEY (Team_ID) REFERENCES Team (Team_ID), FOREIGN KEY (Driver_num) REFERENCES Driver (Driver_num), FOREIGN KEY (Car_ID) REFERENCES Car (Car_ID), CONSTRAINT check_blood_type CHECK (Blood_type IN ('O+','O-','A+','A-','B+','B-','AB+','AB-')));

CREATE TABLE Event (Track_name VARCHAR(255) NOT NULL, Year INT NOT NULL, Country VARCHAR(100) NOT NULL, PRIMARY KEY (Track_name, Year), CONSTRAINT check_year CHECK (Year >= 1970 AND Year <= 2050));

ALTER TABLE Mechanic ADD CONSTRAINT check_age CHECK (Age >= 16 AND Age <= 100);

CREATE TABLE Race (Track_name VARCHAR(255) NOT NULL, Year INT NOT NULL, Driver_num INT NOT NULL, Position INT NOT NULL, PRIMARY KEY (Track_name, Year, Driver_num), FOREIGN KEY (Track_name, Year) REFERENCES Event (Track_name, Year), FOREIGN KEY (Driver_num) REFERENCES Driver (Driver_num), CONSTRAINT check_position CHECK (Position > 0));

CREATE TABLE Car_Race (Track_name VARCHAR(255) NOT NULL, Year INT NOT NULL, Car_ID INT NOT NULL, PRIMARY KEY (Track_name, Year, Car_ID), FOREIGN KEY (Track_name, Year) REFERENCES Event (Track_name, Year), FOREIGN KEY (Car_ID) REFERENCES Car (Car_ID), CONSTRAINT check_year CHECK (Year >= 1970 AND Year <= 2050));

ALTER TABLE Race ADD CONSTRAINT check_year CHECK (Year >= 1970 AND Year <= 2050);

RENAME TABLE Race TO Driver_result;
```

```
INSERT INTO Team VALUES (124, 'Max Racing', 2, 'Exor');
INSERT INTO Team VALUES (298, 'Red Bull', 1, 'Dietrich Mateschitz');
INSERT INTO Team VALUES (310, 'Richard Mille Racing', 3, 'Richard Mille');
INSERT INTO Team VALUES (201, 'Aston Martin Racing', 4, 'David Richards');
INSERT INTO Team VALUES (301, 'McLaren', 5, 'Amanda McLaren');
INSERT INTO Team VALUES (410, 'Ferrari', 6, 'Exor');

INSERT INTO Driver VALUES (1246, 'Tom Masey', 22, 'A+', 'Harcourt Street', 'London',
'England', 124);
INSERT INTO Driver VALUES (1283, 'Alan Wright', 24, 'O-', 'Seaview Road', 'Glasgow',
'Scotland', 298);
INSERT INTO Driver VALUES (4821, 'Jean le Pen', 21, 'AB-', 'Rue de Seine', 'Paris', 'France',
310);
INSERT INTO Driver VALUES (382, 'Oscar Daly', 26, 'A-', 'Main Street', 'Hull', 'England', 201);
INSERT INTO Driver VALUES (294, 'Tim Parker', 28, 'AB+', 'Park Lane', 'New York', 'USA', 310);
INSERT INTO Driver VALUES (521, 'Frank Hill', 26, 'O-', 'Coast Road', 'Galway', 'Ireland', 298);
INSERT INTO Driver VALUES (820, 'Matt West', 29, 'B-', 'Green Lane', 'Brisbane', 'Australia',
301);
INSERT INTO Driver VALUES (904, 'Rick Walker', 26, 'B-', 'Park Lane', 'New York', 'USA', 410);

INSERT INTO Car VALUES (12, 124683, 1246);
INSERT INTO Car VALUES (4, 248932, 1283);
INSERT INTO Car VALUES (8, 149682, 4821);
INSERT INTO Car VALUES (2, 24962, 382);
INSERT INTO Car VALUES (6, 102346, 294);
INSERT INTO Car VALUES (1, 82985, 521);
INSERT INTO Car VALUES (9, 111125, 820);
INSERT INTO Car VALUES (3, 696325, 904);

INSERT INTO Mechanic VALUES (129, 'Tom Arnold', 124, 32, 1246, 12, 'AB+');
INSERT INTO Mechanic VALUES (260, 'Rick Martin', 298, 36, 1283, 4, 'AB-');
INSERT INTO Mechanic VALUES (410, 'Tom Wright', 310, 29, 4821, 8, 'O-');
INSERT INTO Mechanic VALUES (560, 'Alan Spencer', 201, 31, 382, 2, 'A+');
INSERT INTO Mechanic VALUES (111, 'Dave Knight', 310, 32, 294, 6, 'A+');
INSERT INTO Mechanic VALUES (130, 'Sam Hill', 298, 30, 521, 1, 'O+');
INSERT INTO Mechanic VALUES (266, 'Greg Day', 301, 29, 820, 9, 'A-');
INSERT INTO Mechanic VALUES (310, 'Rick Hill', 410, 30, 904, 3, 'AB+');
INSERT INTO Mechanic VALUES (400, 'Jared Graves', 298, 31, 1283, 4, 'O-');
INSERT INTO Mechanic VALUES (811, 'Curtis Keene', 124, 26, 1246, 12, 'O-');
INSERT INTO Mechanic VALUES (929, 'Richie King', 310, 31, 4821, 8, 'AB+');

INSERT INTO Event VALUES ('Brands Hatch', 2018, 'England');
INSERT INTO Event VALUES ('Donington Park', 2018, 'England');
INSERT INTO Event VALUES ('Silverstone', 2017, 'England');
INSERT INTO Event VALUES ('Monaco Grand Prix', 2018, 'Monaco');
INSERT INTO Event VALUES ('Mondello', 2017, 'Ireland');
INSERT INTO Event VALUES ('Anglesey', 2017, 'England');

INSERT INTO Driver_result VALUES ('Brands Hatch', 2018, 1246, 1);
INSERT INTO Driver_result VALUES ('Brands Hatch', 2018, 1283, 2);
INSERT INTO Driver_result VALUES ('Silverstone', 2017, 1246, 3);
INSERT INTO Driver_result VALUES ('Anglesey', 2017, 521, 1);
```

```
INSERT INTO Driver_result VALUES ('Mondello', 2017, 904, 2);
INSERT INTO Driver_result VALUES ('Mondello', 2017, 382, 1);
INSERT INTO Driver_result VALUES ('Donington Park', 2018, 294, 3);

INSERT INTO Car_Race VALUES ('Brands Hatch', 2018, 12);
INSERT INTO Car_Race VALUES ('Brands Hatch', 2018, 4);
INSERT INTO Car_Race VALUES ('Silverstone', 2017, 12);
INSERT INTO Car_Race VALUES ('Anglesey', 2017, 1);
INSERT INTO Car_Race VALUES ('Mondello', 2017, 3);
INSERT INTO Car_Race VALUES ('Mondello', 2017, 2);
INSERT INTO Car_Race VALUES ('Donington Park', 2018, 6);
```