

TRABALHO

Projeto mySQL

Artur José Gomes Pereira

Nº 2040415

João José da Costa Cabral

Nº 2020919

**Tecnologias e Programação de Sistemas de
Informação**

UNIDADE CURRICULAR:
Sistemas Gestores de Bases de Dados I

DOCENTE:
Magno Andrade

ESCOLA SUPERIOR DE TECNOLOGIAS E GESTÃO

Cofinanciado por:



Índice

Índice	1
Introdução	3
Modelo Entidade - Associação.....	4
Modelo Relacional.....	5
Entidades e Relações.....	6
Restrições de Integridade	7
Normalizações	8
Queries das Tabelas em SQL.....	9
CRIAÇÃO DA BASE DE DADOS	9
CRIAÇÃO DA TABELA CAR.....	9
CRIAÇÃO DA TABELA DRIVER.....	9
CRIAÇÃO DA TABELA DRIVER_GRAND_PRIX.....	9
CRIAÇÃO DA TABELA DRIVER_TEAM	9
CRIAÇÃO DA TABELA ENGINE.....	9
CRIAÇÃO DA TABELA GRAND_PRIX	10
CRIAÇÃO DA TABELA MANUFACTURER	10
CRIAÇÃO DA TABELA TEAM.....	10
Queries da Criação de FK PK e Indexes em SQL.....	10
Queries da Inserção de dados em SQL.....	12
INSERÇÃO DA INFORMAÇÃO NA TABELA CAR	12
INSERÇÃO DA INFORMAÇÃO NA TABELA DRIVER	12
INSERÇÃO DA INFORMAÇÃO NA TABELA	13
DRIVER_TEAM	13
INSERÇÃO DA INFORMAÇÃO NA TABELA	13
ENGINE	13
INSERÇÃO DA INFORMAÇÃO NA TABELA	14
GRAND_PRIX	14
INSERÇÃO DA INFORMAÇÃO NA TABELA	14
MANUFACTURER	14
INSERÇÃO DA INFORMAÇÃO NA TABELA	14
TEAM	14
UPDATES DA BASE DE DADOS	15
QUERIES DE PESQUISA NA BASE DE DADOS	16
Qual é o valor total de pontos que o condutor Lando Norris fez em todos os Grand Prix?	16
Total de corridas feitas no Grand Prix por britânicos	16
Todos os pontos conseguidos pelo motor da MERCEDES NO GP da AUSTRALIA.....	17
quais são os 3 carros que conseguiram mais podios no grand prix?	17
Quais são os condutores que já acabaram em primeiro lugar em um Grand Prix?	18
.....	18
Todos os Pontos por Grand Prix que a equipa da Ferrari fez.....	18
Qual é o driver mais novo que participou no gp e qual a sua idade atual	19
Qual foi o driver que recuperou mais lugares num Grand Prix	19

QUAL era a idade dos pilotos na altura que ocorreu o gp da australia?	20
Listar todos os vencedores dos Grand Prix	20
Média da posição final do Valtteri bottas	21
Drivers mais velhos a pontuar Fora Do Podium No gp da china	21
Lista dos drivers e respetivas equipas	22
CONCLUSÃO	23

Introdução

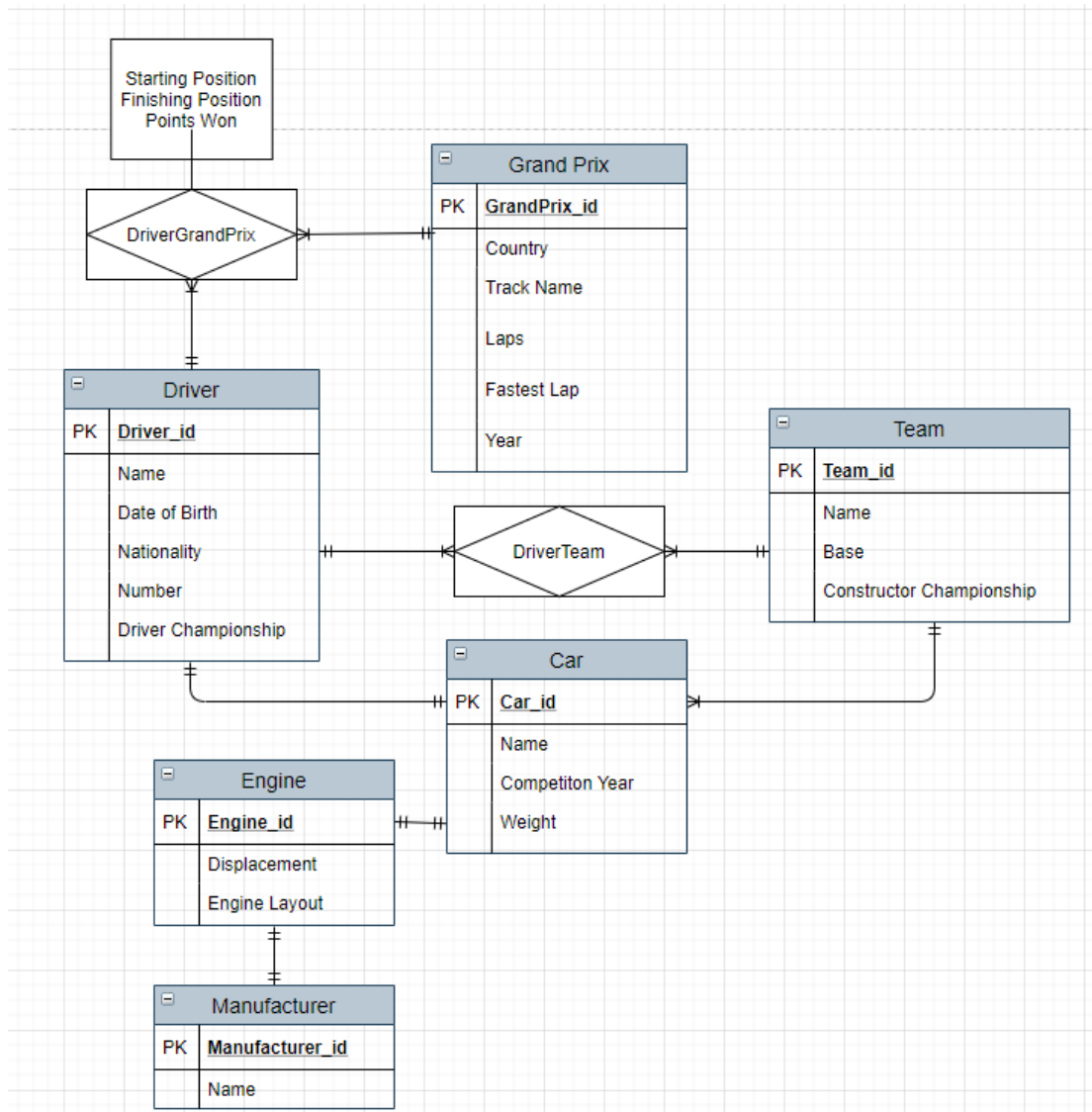
Este relatório é referente ao trabalho solicitado pelo professor Magno Andrade. O trabalho tem como objetivo aplicar as principais técnicas de modelação de dados, bem como implementar numa base de dados criada pelo nosso grupo os conhecimentos adquiridos na unidade de Sistemas Gestores de Base de Dados 1.

Os seus principais objetivos consistem na análise de um problema, e aplicar vários métodos para a resolução desse mesmo problema.

Tomamos a decisão de criar uma base de dados acerca de Formula 1, uma vez que é um tema que ambos consideramos interessante e devido a vasta informação que podemos ter acesso.

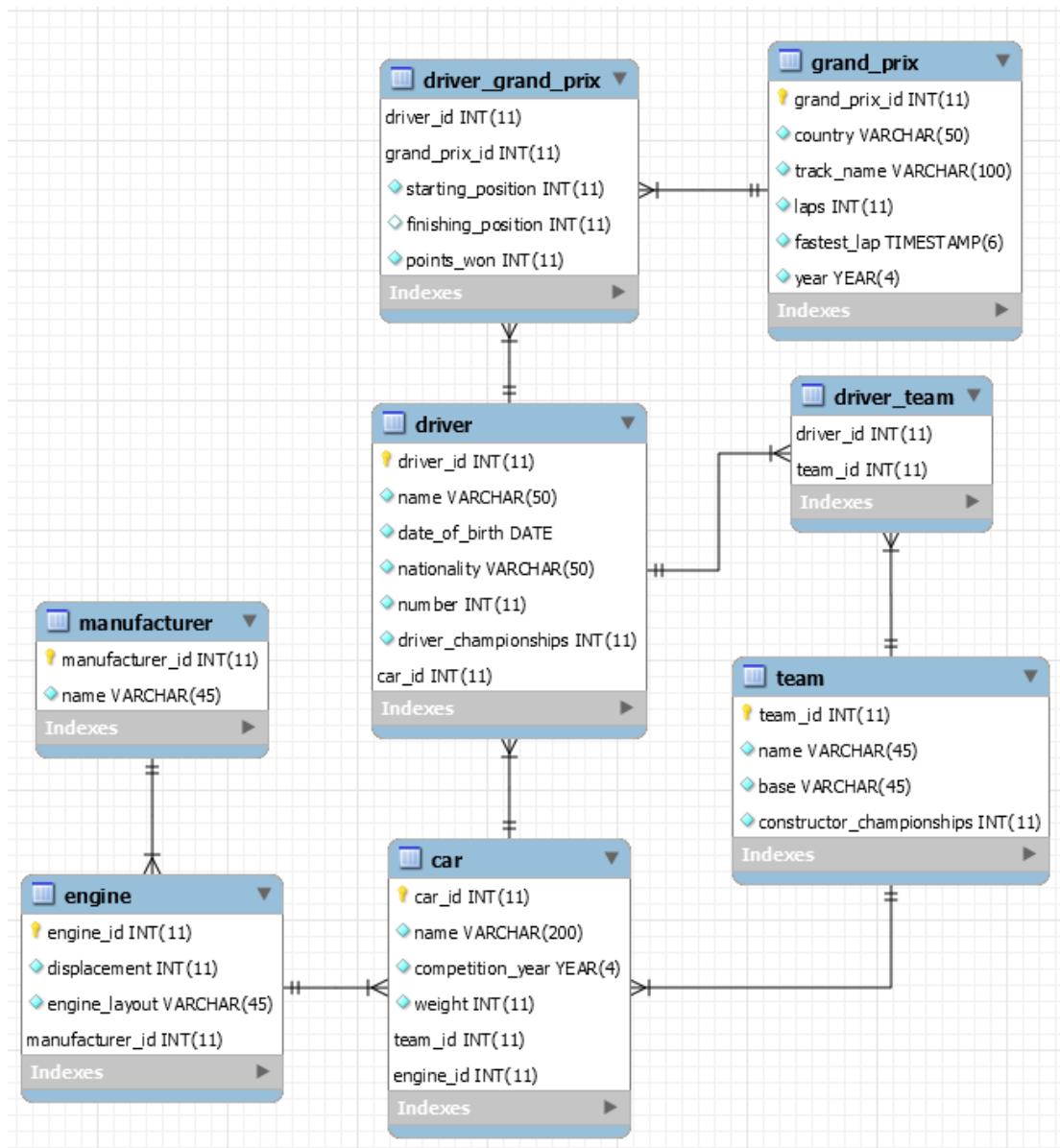
A ideia por trás da base de dados consiste em poder pesquisar todo o tipo de informação acerca das equipas, carros e pilotos assim como poder verificar quais os melhores pilotos ou equipas tendo em conta os pontos realizados nos Grand Prix, Constructor Championships e Driver Championships.

Modelo Entidade - Associação



Notação de Crow Foot 1

Modelo Relacional



Modelo Relacional 1

Entidades e Relações

Na entidade Grand Prix, apresenta informação relevante para conseguir verificar onde as corridas ocorreram, assim como a track, qual a fastest lap e o numero total de laps que a pista tem.

Criamos uma relação de Muitos para Muitos com a entidade Driver uma vez que em um Grand Prix existem vários drivers e os drivers conduzem em múltiplos Grand Prix.

Da ligação criada anteriormente foi criado uma nova entidade que demonstra a relação entre as duas entidades, e onde se pode verificar informação acerca dos resultados pessoais de cada Driver.

Na entidade Team é onde pode consultar a informação sobre onde a equipa está localizada, o seu nome e também os Constructor Championships ganhos pela equipa.

A relação que existe de Muitos para Muitos com o driver é devido ao facto que varias equipas têm vários drivers e os drivers estão inseridos em várias equipas o que também originou a entidade DriverTeam em que faz a ligação dos condutores com as respetivas equipas.

Depois criamos uma entidade com o nome Car em que têm a informação relacionada com os carros, relativamente às relações, criámos uma ligação da Team para o Car de 1 para Muitos com obrigatoriedade uma vez que uma equipa tem vários carros, mas o modelo do carro só pode fazer parte de 1 equipa.

A relação criada da entidade Car para o Driver inserimos uma relação de 1 para 1 com obrigatoriedade uma vez que o Driver tem apenas um carro e um carro têm só um Driver.

Posteriormente criamos a entidade Engine onde mostra informação acerca do motor e criamos uma ligação de 1 para 1 com a entidade Car devido a um carro ter apenas um motor.

E para finalizar foi criada a entidade Manufacturer de forma a podermos ter acesso a empresa que terá feito o motor que vai estar no carro, a relação criada foi novamente uma relação de 1 para 1 uma vez que o Manufacturer cria apenas 1 tipo de motor.

Restrições de Integridade

Respetivamente as restrições de integridade na entidade Grand Prix todas as informações colocadas na tabela não podem ser NULL uma vez que a base de dados que estamos a utilizar é sobre acontecimentos que já ocorreram e por isso a informação já existe, nesta mesma entidade decidimos colocar o campo Fastest Lap como TIMESTAMP uma vez que era a forma mais simples de introduzir o dia/mês/ano dos acontecimentos e os respetivos segundos e milésimos de segundos.

Só seria viável ter campos em NULL se a nossa base de dados fosse o caso de também constar eventos futuros, por exemplo um GP a se realizar em 2040 já podia ter o Country e o Year onde o GP iria ocorrer, mas podia ainda não ter sido decidido qual a pista e devido isso não ia existir as Laps, Track Name e Fastest lap uma vez que essa informação seria desconhecida.

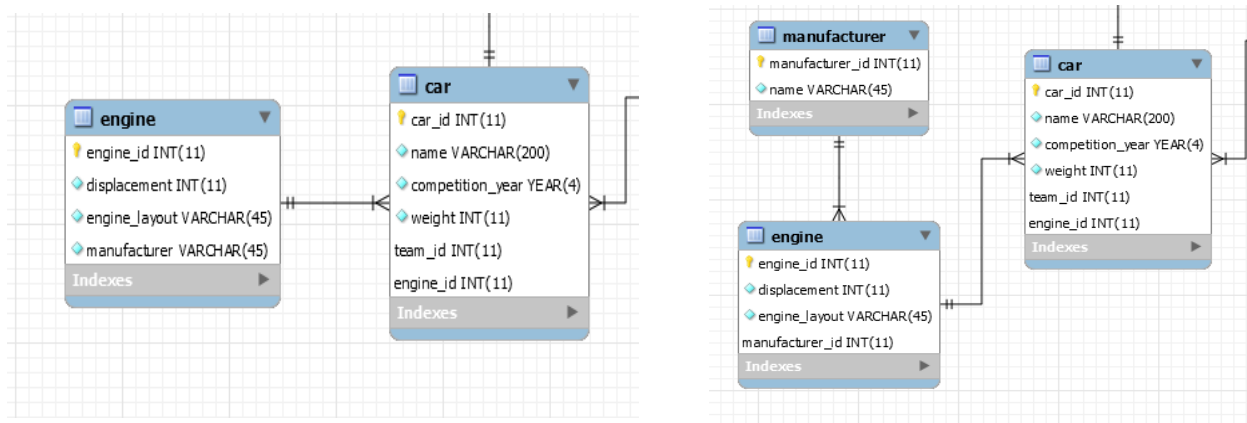
Novamente as restantes tabelas e os campos encontram-se NOT NULL devido ao facto que foi exposto a cima a nossa tabela apenas trata de acontecimentos que já ocorreram a única exceção que temos na nossa BD é na entidade DriverGrandPrix na Finishing Position uma vez que o Driver até pode ter começado a corrida, mas pode não ter terminado a mesma.

E também na entidade DriverGrandPrix foi pré-definido no Points Won o valor de 0 uma vez que caso o Driver não conclua a corrida por algum motivo automaticamente é alocado 0 pontos.

Na entidade Team foi criada uma restrição no Name uma vez que o nome das equipas no formula 1 é único e não existem equipas com nome igual, devido a situação introduzimos o Name como um índice único de forma a evitar a existência de repetições na mesma entidade, também pré-definimos os Constructor_Championship a 0 uma vez que nem todas as equipas ganharam um championship.

Normalizações

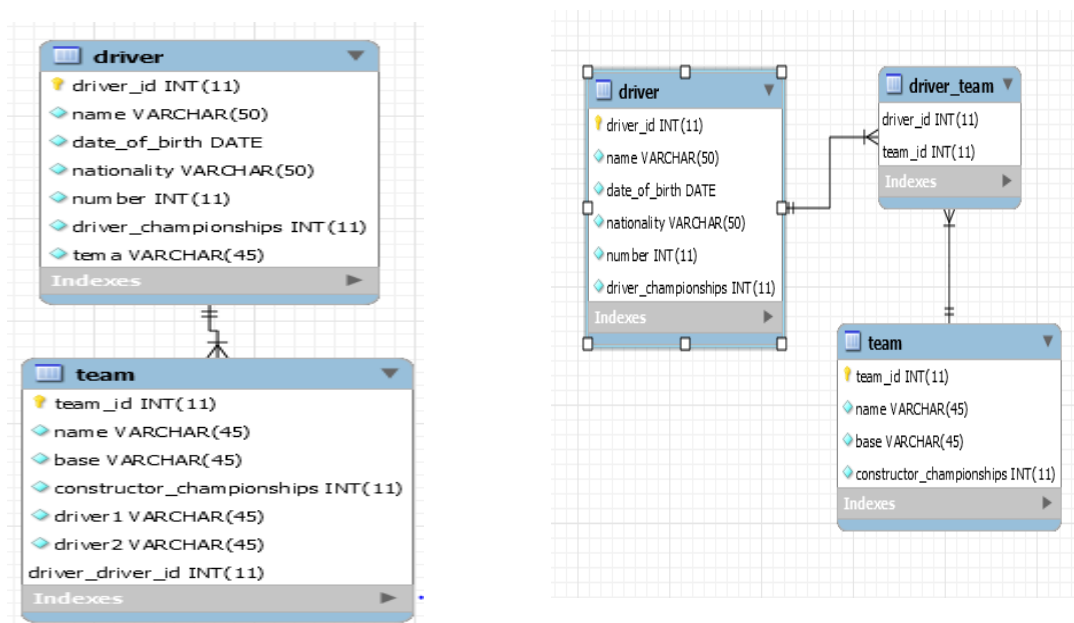
Normalização na Primeira Forma Normal 1



Normalização na Primeira Forma Normal 2



Normalização na Primeira Forma Normal 3



Queries das Tabelas em SQL

CRIAÇÃO DA BASE DE DADOS

```
CREATE DATABASE formula_01;
```

CRIAÇÃO DA TABELA CAR

```
CREATE TABLE `car` (  
  `car_id` int(11) NOT NULL,  
  `name` varchar(200) NOT NULL,  
  `competition_year` year(4) NOT NULL,  
  `weight` int(11) NOT NULL,  
  `team_id` int(11) NOT NULL,  
  `engine_id` int(11) NOT NULL  
);
```

CRIAÇÃO DA TABELA DRIVER

```
CREATE TABLE `driver` (  
  `driver_id` int(11) NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `date_of_birth` date NOT NULL,  
  `nationality` varchar(50) NOT NULL,  
  `number` int(11) NOT NULL,  
  `driver_championships` int(11) NOT NULL DEFAULT 0,  
  `car_id` int(11) NOT NULL  
);
```

CRIAÇÃO DA TABELA DRIVER_GRAND_PRIX

```
CREATE TABLE `driver_grand_prix` (  
  `driver_id` int(11) NOT NULL,  
  `grand_prix_id` int(11) NOT NULL,  
  `starting_position` int(11) NOT NULL,  
  `finishing_position` int(11),  
  `points_won` int(11) NOT NULL DEFAULT 0  
);
```

CRIAÇÃO DA TABELA DRIVER_TEAM

```
CREATE TABLE `driver_team` (  
  `driver_id` int(11) NOT NULL,  
  `team_id` int(11) NOT NULL  
);
```

CRIAÇÃO DA TABELA ENGINE

```
CREATE TABLE `engine` (  
  `engine_id` int(11) NOT NULL,  
  `displacement` int(11) NOT NULL,  
  `engine_layout` varchar(45) NOT NULL,  
  `manufacturer_id` int(11) NOT NULL  
);
```

CRIAÇÃO DA TABELA GRAND_PRIX

```
CREATE TABLE `grand_prix` (  
  `grand_prix_id` int(11) NOT NULL,  
  `country` varchar(50) NOT NULL,  
  `track_name` varchar(100) NOT NULL,  
  `laps` int(11) NOT NULL,  
  `fastest_lap` timestamp(6) DEFAULT current_timestamp(6) ON UPDATE  
  current_timestamp(6),  
  `year` year(4) NOT NULL  
);
```

CRIAÇÃO DA TABELA MANUFACTURER

```
CREATE TABLE `manufacturer` (  
  `manufacturer_id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL  
);
```

CRIAÇÃO DA TABELA TEAM

```
CREATE TABLE `team` (  
  `team_id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `base` varchar(45) NOT NULL,  
  `constructor_championships` int(11) NOT NULL DEFAULT 0,  
  UNIQUE INDEX `name_UNIQUE` (`name` ASC)  
);
```

Queries da Criação de FK PK e Indexes em SQL

```
ALTER TABLE `car`  
  ADD PRIMARY KEY (`car_id`, `team_id`, `engine_id`),  
  ADD UNIQUE KEY `car_id_UNIQUE` (`car_id`),  
  ADD KEY `fk_car_team_idx` (`team_id`),  
  ADD KEY `fk_car_engine1_idx` (`engine_id`);
```

```
ALTER TABLE `driver`  
  ADD PRIMARY KEY (`driver_id`, `car_id`),  
  ADD UNIQUE KEY `driver_id_UNIQUE` (`driver_id`),  
  ADD KEY `fk_driver_car1_idx` (`car_id`);
```

```
ALTER TABLE `driver_grand_prix`  
  ADD PRIMARY KEY (`driver_id`, `grand_prix_id`),  
  ADD KEY `fk_driver_has_grand_prix_grand_prix1_idx` (`grand_prix_id`),  
  ADD KEY `fk_driver_has_grand_prix_driver_idx` (`driver_id`);
```

```
ALTER TABLE `driver_team`  
  ADD PRIMARY KEY (`driver_id`, `team_id`),  
  ADD KEY `fk_driver_has_team_team1_idx` (`team_id`),  
  ADD KEY `fk_driver_has_team_driver1_idx` (`driver_id`);
```

```

ALTER TABLE `engine`
  ADD PRIMARY KEY (`engine_id`,`manufacturer_id`),
  ADD UNIQUE KEY `idengine_UNIQUE` (`engine_id`),
  ADD KEY `fk_engine_manufacturer1_idx` (`manufacturer_id`);

ALTER TABLE `grand_prix`
  ADD PRIMARY KEY (`grand_prix_id`),
  ADD UNIQUE KEY `grand_prix_id_UNIQUE` (`grand_prix_id`);

ALTER TABLE `manufacturer`
  ADD PRIMARY KEY (`manufacturer_id`),
  ADD UNIQUE KEY `idmanufacturer_UNIQUE` (`manufacturer_id`);

ALTER TABLE `team`
  ADD PRIMARY KEY (`team_id`),
  ADD UNIQUE KEY `team_id_UNIQUE` (`team_id`);

ALTER TABLE `car`
  MODIFY `car_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

ALTER TABLE `driver`
  MODIFY `driver_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;

ALTER TABLE `engine`
  MODIFY `engine_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

ALTER TABLE `grand_prix`
  MODIFY `grand_prix_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=23;

ALTER TABLE `manufacturer`
  MODIFY `manufacturer_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

ALTER TABLE `team`
  MODIFY `team_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

ALTER TABLE `car`
  ADD CONSTRAINT `fk_car_engine1` FOREIGN KEY (`engine_id`) REFERENCES `engine`
    (`engine_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_car_team` FOREIGN KEY (`team_id`) REFERENCES `team`
    (`team_id`) ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE `driver`
  ADD CONSTRAINT `fk_driver_car1` FOREIGN KEY (`car_id`) REFERENCES `car` (`car_id`)
    ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE `driver_grand_prix`
  ADD CONSTRAINT `fk_driver_has_grand_prix_driver` FOREIGN KEY (`driver_id`)
    REFERENCES `driver` (`driver_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_driver_has_grand_prix_grand_prix1` FOREIGN KEY
    (`grand_prix_id`) REFERENCES `grand_prix` (`grand_prix_id`) ON DELETE NO ACTION ON
    UPDATE NO ACTION;

ALTER TABLE `driver_team`
  ADD CONSTRAINT `fk_driver_has_team_driver1` FOREIGN KEY (`driver_id`) REFERENCES
    `driver` (`driver_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_driver_has_team_team1` FOREIGN KEY (`team_id`) REFERENCES
    `team` (`team_id`) ON DELETE NO ACTION ON UPDATE NO ACTION;

```

```
ALTER TABLE `engine`  
  ADD CONSTRAINT `fk_engine_manufacturer1` FOREIGN KEY (`manufacturer_id`)  
  REFERENCES `manufacturer` (`manufacturer_id`) ON DELETE NO ACTION ON UPDATE  
  NO ACTION;
```

Queries da Inserção de dados em SQL

INSERÇÃO DA INFORMAÇÃO NA TABELA CAR

```
INSERT INTO `car` (`car_id`, `name`, `competition_year`, `weight`, `team_id`,  
`engine_id`) VALUES  
(1, 'Mercedes-AMG F1 W10 EQ POWER+', 2019, 743, 1, 1),  
(2, 'Ferrari SF90', 2019, 743, 7, 3),  
(3, 'McLaren MCL34', 2019, 743, 3, 4),  
(4, 'Red Bull RB15', 2019, 743, 5, 2),  
(5, 'Haas VF-19', 2019, 743, 2, 3),  
(6, 'Scuderia Toro Rosso STR14', 2019, 743, 8, 2),  
(7, 'Williams FW42', 2019, 745, 9, 1),  
(8, 'Racing Point RP19', 2019, 743, 10, 1),  
(9, 'Alfa Romeo Racing C38', 2019, 743, 4, 3),  
(10, 'Renault R.S.19', 2019, 743, 6, 4);
```

INSERÇÃO DA INFORMAÇÃO NA TABELA DRIVER

```
INSERT INTO `driver` (`driver_id`, `name`, `date_of_birth`, `nationality`, `number`,  
`driver_championships`, `car_id`) VALUES  
(1, 'Lewis Hamilton', '1985-01-07', 'British', 44, 6, 1),  
(2, 'Valtteri Bottas', '1989-08-28', 'Finnish', 77, 0, 1),  
(3, 'Sebastian Vettel', '1987-07-03', 'German', 5, 4, 2),  
(4, 'Charles Leclerc', '1997-10-16', 'Monegasque', 16, 0, 2),  
(5, 'Lando Norris', '1999-11-13', 'British', 4, 0, 3),  
(6, 'Carlos Sainz', '1994-09-01', 'Spanish', 55, 0, 3),  
(7, 'Alexander Albon', '1996-03-23', 'British', 23, 0, 4),  
(8, 'Max Verstappen', '1997-09-30', 'Dutch', 33, 0, 4),  
(9, 'Daniel Ricciardo', '1989-07-01', 'Australian', 3, 0, 10),  
(10, 'Nico Hulkenberg', '1987-08-17', 'German', 27, 0, 10),  
(11, 'Pierre Gasly', '1996-02-07', 'French', 10, 0, 6),  
(12, 'Daniil Kvyat', '1994-04-26', 'Russian', 26, 0, 6),  
(13, 'Sergio Perez', '1990-01-26', 'Mexican', 11, 0, 8),  
(14, 'Lance Stroll', '1998-10-29', 'Canadian', 18, 0, 8),  
(15, 'Kimi Raikkonen', '1979-10-17', 'Finnish', 7, 1, 9),  
(16, 'Antonio Giovinazzi', '1993-12-14', 'Italian', 99, 0, 9),  
(17, 'Romain Grosjean', '1986-04-17', 'French', 8, 0, 5),  
(18, 'Kevin Magnussen', '1992-10-05', 'Danish', 20, 0, 5),  
(19, 'George Russell', '1998-02-15', 'British', 63, 0, 7),  
(20, 'Robert Kubica', '1984-12-07', 'Polish', 88, 0, 7);
```

INSERÇÃO DA INFORMAÇÃO NA TABELA DRIVER_GRAND_PRIX

*São cerca de 120 registos inseridos manualmente só nesta tabela, de forma a poder apresentar os valores introduzidos retiramos os espaços, pode depois verificar a informação de forma mais detalhada no anexo formula_01.

```
INSERT INTO `driver_grand_prix` (`driver_id`, `grand_prix_id`, `starting_position`,  
`finishing_position`, `points_won`) VALUES  
(1, 2, 1, 2, 18), (1, 3, 3, 1, 25), (1, 4, 2, 1, 25), (1, 5, 2, 2, 18), (1, 6, 2, 1, 26), (1, 7, 1, 1, 25), (1,  
8, 2, 1, 25), (1, 9, 1, 1, 25), (1, 10, 4, 5, 10), (1, 11, 2, 1, 26), (1, 12, 1, 9, 2), (1, 13, 3, 1,  
25), (2, 2, 2, 1, 26), (2, 3, 4, 2, 18), (2, 4, 1, 2, 18), (2, 5, 1, 1, 25), (2, 6, 1, 2, 18), (2, 7, 2, 3,  
15), (2, 8, 6, 4, 13), (2, 9, 2, 2, 18), (2, 10, 3, 3, 15), (2, 11, 1, 2, 18), (2, 13, 2, 8, 4), (3, 2, 3,  
4, 12), (3, 3, 2, 5, 10), (3, 4, 3, 3, 15), (3, 5, 3, 3, 15), (3, 6, 3, 4, 12), (3, 7, 4, 2, 18), (3, 8, 1,  
2, 18), (3, 9, 7, 5, 11), (3, 10, 9, 4, 12), (3, 12, 20, 2, 18), (3, 13, 5, 3, 15), (4, 2, 5, 5, 10), (4,  
3, 1, 3, 16), (4, 4, 4, 5, 10), (4, 5, 8, 5, 11), (4, 6, 5, 5, 10), (4, 8, 3, 3, 15), (4, 9, 3, 3, 15), (4,  
10, 1, 2, 18), (4, 11, 3, 3, 15), (4, 13, 4, 4, 12), (5, 3, 9, 6, 8), (5, 5, 7, 8, 4), (5, 9, 5, 9, 2), (5,  
10, 5, 6, 8), (5, 13, 7, 9, 2), (6, 5, 9, 7, 6), (6, 6, 12, 8, 4), (6, 7, 9, 6, 8), (6, 9, 6, 6, 8), (6, 10,  
19, 8, 4), (6, 11, 13, 6, 8), (6, 12, 7, 5, 10), (6, 13, 8, 5, 10), (7, 3, 12, 9, 2), (7, 4, 20, 10, 1), (7,  
7, 10, 8, 4), (7, 12, 16, 6, 8), (7, 13, 12, 10, 1), (8, 2, 4, 3, 15), (8, 3, 5, 4, 12), (8, 4, 5, 4,  
12), (8, 5, 4, 4, 12), (8, 6, 4, 3, 15), (8, 7, 3, 4, 12), (8, 8, 9, 5, 10), (8, 9, 4, 4, 12), (8, 10, 2, 1,  
26), (8, 11, 4, 5, 10), (8, 12, 2, 1, 26), (8, 13, 1, 2, 19), (9, 4, 7, 7, 6), (9, 7, 6, 9, 2), (9, 8, 4, 6,  
8), (9, 11, 7, 7, 6), (10, 2, 11, 7, 6), (10, 8, 7, 7, 6), (10, 9, 13, 8, 4), (10, 11, 10, 10, 1), (11, 3,  
13, 8, 4), (11, 4, 6, 6, 9), (11, 6, 6, 6, 8), (11, 7, 8, 5, 11), (11, 8, 5, 8, 4), (11, 9, 9, 10, 1), (11,  
10, 8, 7, 6), (11, 11, 5, 4, 12), (11, 13, 6, 6, 8), (12, 2, 15, 10, 1),  
(12, 6, 9, 9, 2), (12, 7, 7, 7, 6), (12, 8, 10, 10, 1), (12, 11, 17, 9, 2), (12, 12, 14, 3, 15),  
(13, 3, 14, 10, 1), (13, 4, 12, 8, 4), (13, 5, 5, 6, 8), (14, 2, 16, 9, 2), (14, 5, 13, 9, 2), (14, 8, 17,  
9, 2), (14, 12, 15, 4, 12), (15, 2, 9, 8, 4), (15, 3, 8, 7, 6), (15, 4, 13, 9, 2), (15, 5, 19, 10, 1), (15,  
9, 12, 7, 6), (15, 10, 6, 9, 2), (15, 11, 12, 8, 4), (15, 13, 10, 7, 6), (16, 10, 7, 10, 1), (17, 6, 7,  
10, 1), (17, 7, 13, 10, 1), (17, 12, 6, 7, 6), (18, 2, 7, 6, 8), (18, 6, 8, 7, 6), (18, 12, 12, 8, 4), (20,  
12, 18, 10, 1);
```

INSERÇÃO DA INFORMAÇÃO NA TABELA DRIVER_TEAM

```
INSERT INTO `driver_team` (`driver_id`, `team_id`) VALUES  
(1, 1), (2, 1), (3, 7), (4, 7), (5, 3), (6, 3), (7, 5), (8, 5), (9, 6), (10, 6), (11, 8),  
(12, 8), (13, 10), (14, 10), (15, 4), (16, 4), (17, 2), (18, 2), (19, 9), (20, 9);
```

INSERÇÃO DA INFORMAÇÃO NA TABELA ENGINE

```
INSERT INTO `engine` (`engine_id`, `displacement`, `engine_layout`,  
`manufacturer_id`) VALUES  
(1, 1600, 'V6 TURBO HYBRID', 1),  
(2, 1600, 'V6 TURBO HYBRID', 2),  
(3, 1600, 'V6 TURBO HYBRID', 3),  
(4, 1600, 'V6 TURBO HYBRID', 4);
```

INSERÇÃO DA INFORMAÇÃO NA TABELA GRAND_PRIX

```
INSERT INTO `grand_prix` (`grand_prix_id`, `country`, `track_name`, `laps`, `fastest_lap`,  
`year`) VALUES  
(2, 'Australia', 'Melbourne Grand Prix Circuit', 58, '2019-03-17 00:01:25.580000', 2019),  
(3, 'Bahrain', 'Sakhir', 57, '2019-03-31 00:01:33.411000', 2019),  
(4, 'China', 'Shanghai International Circuit', 56, '2019-04-13 23:01:34.742000', 2019),  
(5, 'Azerbaijan', 'Baku City Circuit', 51, '2019-04-27 23:01:43.009000', 2019),  
(6, 'Spain', 'Circuit de Barcelona', 66, '2019-05-11 23:01:18.492000', 2019),  
(7, 'Monaco', 'Circuit de Monaco', 78, '2019-06-11 23:01:14.279000', 2019),  
(8, 'Canada', 'Circuit Gilles', 70, '2019-06-08 23:01:13.078000', 2019),  
(9, 'France', 'Circuit Paul Ricard', 53, '2019-06-22 23:01:32.740000', 2019),  
(10, 'Austria', 'Red Bull Ring', 71, '2019-06-29 23:01:07.475000', 2019),  
(11, 'Great Britain', 'Silverstone Circuit', 52, '2019-07-13 23:01:27.369000', 2019),  
(12, 'Germany', 'Hockenheimring', 67, '2019-07-27 23:01:16.645000', 2019),  
(13, 'Hungary', 'Hungaroring', 70, '2019-08-03 23:01:17.103000', 2019),  
(14, 'Belgium', 'Circuit de Spa', 44, '2019-08-31 23:01:46.409000', 2019),  
(15, 'Italy', 'Autodromo Nazionale Monza', 53, '2019-09-07 23:01:21.779000', 2019),  
(16, 'Singapore', 'Marina Bay Street Circuit', 61, '2019-09-21 23:01:42.301000', 2019),  
(17, 'Russia', 'Sochi Autodrom', 53, '2019-09-28 23:01:35.761000', 2019),  
(18, 'Japan', 'Suzuka International Racing Course', 53, '2019-10-12 23:01:30.983000',  
2019),  
(19, 'Mexico', 'Autódromo Hermanos Rodríguez', 71, '2019-10-26 23:01:19.232000', 2019),  
(20, 'United States', 'Circuit of The Americas', 56, '2019-11-03 00:01:36.169000', 2019),  
(21, 'Brasil', 'Autódromo José Carlos Pace', 71, '2019-11-17 00:01:10.698000', 2019),  
(22, 'Abu Dhabi', 'Yas Marina Circuit', 55, '2019-12-01 00:01:34.715000', 2019);
```

INSERÇÃO DA INFORMAÇÃO NA TABELA MANUFACTURER

```
INSERT INTO `manufacturer` (`manufacturer_id`, `name`) VALUES  
(1, 'Mercedes'),  
(2, 'Honda'),  
(3, 'Ferrari'),  
(4, 'Renault');
```

INSERÇÃO DA INFORMAÇÃO NA TABELA TEAM

```
INSERT INTO `team` (`team_id`, `name`, `base`, `constructor_championships`) VALUES  
(1, 'Mercedes-AMG Petronas F1 Team', 'United Kingdom', 6),  
(2, 'Haas F1 Team', 'United States', 0),  
(3, 'McLaren F1 Team', 'United Kingdom', 8),  
(4, 'Alfa Romeo Racing ORLEN', 'Switzerland', 0),  
(5, 'Aston Martin Red Bull Racing', 'United Kingdom', 4),  
(6, 'Renault DP World F1 Team', 'United Kingdom', 2),  
(7, 'Scuderia Ferrari Mission Winnow', 'Italy', 16),  
(8, 'Scuderia AlphaTauri Honda', 'Italy', 0),  
(9, 'Williams Racing', 'United Kingdom', 9),  
(10, 'BWT Racing Point F1 Team', 'United Kingdom', 0);
```

UPDATES DA BASE DE DADOS

UPDATE driver

SET driver.name = "Valteri Botas"

WHERE driver.name = "Valtteri Bottas";

UPDATE engine

SET engine.displacement = 6000, engine_layout = "V12 BITURBO"

WHERE engine.manufacturer_id = 1;

UPDATE team

SET team.name = "Pay to Win F1 Team"

WHERE team.team_id = 10;

UPDATE car

SET car.name = "Renault Unreliable.19"

WHERE car.name like "Renault %";

UPDATE manufacturer

SET manufacturer.name = "Toyota"

WHERE manufacturer.name = "Honda";

UPDATE grand_prix

SET grand_prix.country = "Portugal" , grand_prix.track_name = "Autodromo do Estoril"

WHERE grand_prix.country = "Singapore";

UPDATE car

SET car.weight = 1500

WHERE car.name like "Will%";

QUERIES DE PESQUISA NA BASE DE DADOS

QUAL É O VALOR TOTAL DE PONTOS QUE O CONDUTOR LANDO NORRIS FEZ EM TODOS OS GRAND PRIX?

```
SELECT SUM(points_won) AS 'Total de pontos', name FROM driver_grand_prix  
INNER JOIN driver ON driver_grand_prix.driver_id = driver.driver_id  
WHERE driver.name = "Lando Norris";
```

- ➔ O SUM é uma função de SQL que soma os valores que estão na tabela indicada, é necessário fazer um INNER JOIN uma vez que temos de ir buscar a tabela driver a condição de procura neste caso o nome do driver, isto serve para apenas mostrar os pontos que foram realizados pelo Lando Norris.

OUTPUT

	Total de pontos	name
▶	24	Lando Norris

TOTAL DE CORRIDAS FEITAS NO GRAND PRIX POR BRITANICOS

```
SELECT COUNT(starting_position) AS 'TOTAL DE CORRIDAS NO GRAND PRIX',name  
FROM driver_grand_prix  
INNER JOIN driver ON driver_grand_prix.driver_id = driver.driver_id  
WHERE driver.nationality = "British"  
GROUP BY driver.number;
```

- ➔ O COUNT é uma função de SQL que conta o numero de vezes que um campo aparece na tabela, foi necessário criar um INNER JOIN de forma aceder a nacionalidade dos condutores e depois agrupamos o resultado pelo numero dos condutores.

OUTPUT

	TOTAL DE CORRIDAS NO GRAND PRIX	name
▶	5	Lando Norris
	5	Alexander Albon
	12	Lewis Hamilton

TODOS OS PONTOS CONSEGUIDOS PELO MOTOR DA MERCEDES NO GP DA AUSTRALIA

```
SELECT SUM(points_won) AS 'Pontos Ganhos Motor Mercedes' FROM driver_grand_prix
INNER JOIN grand_prix ON driver_grand_prix.grand_prix_id = grand_prix.grand_prix_id
INNER JOIN driver ON driver_grand_prix.driver_id = driver.driver_id
INNER JOIN car ON driver.car_id = car.car_id
INNER JOIN engine ON car.engine_id = engine.engine_id
INNER JOIN manufacturer ON engine.manufacturer_id =
manufacturer.manufacturer_id
WHERE (manufacturer.name, grand_prix.country ) = ("MERCEDES","Australia");
```

- Utilizamos novamente o função SUM() do SQL no entanto uma vez que é necessário ter acesso aos dados da tabela manufacturer e também do grandprix foi necessário criar INNER JOINS de forma a juntar as respetivas tabelas e depois utilizar a condição para apresentar o pretendido.

OUTPUT

	Pontos Ganhos Motor Mercedes
▶	46

QUAIS SÃO OS 3 CARROS QUE CONSEGUIRAM MAIS PODIOS NO GRAND PRIX?

```
SELECT COUNT(driver_grand_prix.finishing_position) AS "Podios_Grand_Prix", car.name
FROM driver_grand_prix
INNER JOIN driver ON driver_grand_prix.driver_id = driver.driver_id
INNER JOIN car ON driver.car_id = car.car_id
GROUP BY car.name
ORDER BY Podios_Grand_Prix DESC LIMIT 3;
```

- Utilizamos o COUNT() para fazer a contagem de pódios, depois temos de juntar as tabelas com um INNER JOIN de forma a conseguir aceder a informação necessária após simplesmente agrupamos pelo nome do carro e uma vez que é apenas os 3 primeiros que conseguiram mais pódios utilizamos o LIMIT 3 de forma a limitar a informação que aparece, também utilizamos o DESC de forma a que a informação apareça de forma descendente.

OUTPUT

	Podios_Grand_Prix	name
▶	23	Mercedes-AMG F1 W10 EQ POWER+
	21	Ferrari SF90
	17	Red Bull RB15

QUAIS SÃO OS CONDUTORES QUE JÁ ACABARAM EM PRIMEIRO LUGAR EM UM GRAND PRIX?

```
SELECT driver.name from driver
INNER JOIN driver_grand_prix ON driver.driver_id = driver_grand_prix.driver_id
WHERE finishing_position = 1
GROUP BY driver.name;
```

- ➔ Seleccionamos o nome do condutor e depois a condição de procura o INNER JOIN é novamente para conectar as tabelas, depois apresentamos todos os condutores que já ficaram em primeiro lugar.

OUTPUT

	name
▶	Lewis Hamilton
	Max Verstappen
	Valtteri Bottas

TODOS OS PONTOS POR GRAND PRIX QUE A EQUIPA DA FERRARI FEZ

```
SELECT SUM(driver_grand_prix.points_won) AS 'Total Pontos' , grand_prix.country FROM
driver_grand_prix
INNER JOIN grand_prix ON driver_grand_prix.grand_prix_id = grand_prix.grand_prix_id
INNER JOIN driver ON driver_grand_prix.driver_id = driver.driver_id
INNER JOIN driver_team ON driver.driver_id = driver_team.driver_id
INNER JOIN team ON driver_team.team_id = team.team_id
WHERE team.name LIKE '%Ferrari%'
GROUP BY grand_prix.country;
```

- ➔ Utilizamos a função SUM() para somar os pontos que as equipas fizeram depois criamos INNER JOINS para ter acesso à informação das tabelas, como condição utilizamos o LIKE para procurar o nome da equipa que têm "Ferrari" e depois agrupamos os países dos GP para mostrar todos os GP e os pontos que fizeram em cada GP.

OUTPUT

	Total Pontos	country
▶	22	Australia
	30	Austria
	26	Azerbaijan
	26	Bahrain
	33	Canada
	25	China
	26	France
	18	Germany
	15	Great Britain
	27	Hungary
	18	Monaco
	22	Spain

QUAL É O DRIVER MAIS NOVO QUE PARTICIOU NO GP E QUAL A SUA IDADE ATUAL

```
SELECT MIN(YEAR(CURDATE())-YEAR(driver.date_of_birth)) AS 'Idade_Atual',  
driver.name FROM driver  
INNER JOIN driver_grand_prix ON driver.driver_id = driver_grand_prix.driver_id  
WHERE driver_grand_prix.starting_position >= 1  
GROUP BY driver.name  
ORDER BY Idade_Atual ASC LIMIT 1;
```

- ➔ Com a função de SQL YEAR() é possível aceder apenas a propriedade do ano por isso em cima utilizamos está mesma propriedade, inserimos dentro a função MIN() de forma a que selecione o valor mínimo possível, utilizamos o YEAR(CURDATE()) para que vá buscar data de hoje mas só o ano atual após solicitamos o nome desse condutor, criamos um INNER JOIN de forma a unir as tabelas e utilizamos a condição WHERE de forma a ter a certeza que o piloto que estamos a procurar já entrou num GP depois ordenamos os resultados pela Idade_Atual tabela que foi criada inicialmente colocamos por ascendente e com limit de 1 uma vez que só queremos ver o mais novo.

OUTPUT

	Idade_Atual	name
▶	21	Lando Norris

QUAL FOI O DRIVER QUE RECUPEROU MAIS LUGARES NUM GRAND PRIX

```
SELECT MAX(starting_position - finishing_position) AS "Ganhos", driver.name FROM  
driver_grand_prix  
INNER JOIN driver ON driver_grand_prix.driver_id = driver.driver_id  
GROUP BY driver.name  
ORDER BY Ganhos DESC LIMIT 1
```

- ➔ Nesta querie inicialmente fazemos um calculo de subtração de forma a verificar qual o driver que conseguiu subir mais lugares num GP, e depois usamos a função MAX() para selecionar o condutor com mais lugares ganhos, a seguir fizemos um INNER JOIN para unir as tabelas e poder ter acesso a toda a informação, indicamos que é para agrupar o resultado pelo nome do driver e depois para ordenar pela tabela Ganhos que foi criada anteriormente, utilizamos o DESC para que a informação estava de forma descendente com LIMIT de 1, ou seja apenas para aparecer 1.

OUTPUT

	Ganhos	name
▶	18	Sebastian Vettel

QUAL ERA A IDADE DOS PILOTOS NA ALTURA QUE OCORREU O GP DA AUSTRALIA?

```
SELECT YEAR(grand_prix.year) - YEAR(driver.date_of_birth) AS 'Idade_do_Piloto',  
driver.name FROM driver  
INNER JOIN driver_grand_prix ON driver.driver_id = driver_grand_prix.driver_id  
INNER JOIN grand_prix ON driver_grand_prix.grand_prix_id = grand_prix.grand_prix_id  
WHERE grand_prix.country = "Australia"  
GROUP BY driver.name  
ORDER BY Idade_do_Piloto ASC;
```

- ➔ A querie no inicio utiliza novamente a função YEAR() e faz uma subtração das idades, foi criado INNER JOINS de forma a ter acesso a informação, como condição selecionamos o GP da Austrália agrupando pelo nome dos pilotos e para ordenar de forma ascendente pela idade.

OUTPUT

	Idade_do_Piloto	name
▶	21	Lance Stroll
	22	Max Verstappen
	22	Charles Leclerc
	25	Daniil Kvyat
	27	Kevin Magnussen
	30	Valtteri Bottas
	32	Sebastian Vettel
	32	Nico Hulkenberg
	34	Lewis Hamilton
	40	Kimi Raikkonen

LISTAR TODOS OS VENCEDORES DOS GRAND PRIX

```
SELECT driver.name AS "WINNER", grand_prix.country AS "GP" FROM driver  
INNER JOIN driver_grand_prix ON driver.driver_id = driver_grand_prix.driver_id  
INNER JOIN grand_prix ON driver_grand_prix.grand_prix_id = grand_prix.grand_prix_id  
WHERE driver_grand_prix.finishing_position = 1;
```

- ➔ A querie usa da ligação entre as tabelas grand_prix e driver_grand_prix para verificar qual o piloto que ganhou cada GP. Foi criado INNER JOINS de forma a ter acesso a essa informação, como condição selecionamos a finishing_position 1 de todos os GPs.

OUTPUT

	WINNER	GP
▶	Lewis Hamilton	Bahrain
	Lewis Hamilton	China
	Lewis Hamilton	Spain
	Lewis Hamilton	Monaco
	Lewis Hamilton	Canada
	Lewis Hamilton	France
	Lewis Hamilton	Great Britain
	Lewis Hamilton	Hungary
	Valtteri Bottas	Australia
	Valtteri Bottas	Azerbaijan
	Max Verstappen	Austria
	Max Verstappen	Germany

MÉDIA DA POSIÇÃO FINAL DO VALTTERI BOTTAS

```
SELECT ROUND(AVG(driver_grand_prix.finishing_position),1) AS "Média da posição final" FROM driver_grand_prix  
INNER JOIN driver ON driver_grand_prix.driver_id = driver.driver_id  
WHERE driver.name = "Valtteri Bottas";
```

- ➔ A função ROUND(,<casasDecimais>) é utilizada pra arredondar um valor, neste caso, o valor medio, que foi calculado pela função AVG(), da posição final das corridas do Valtteri Bottas. A querir usa da finishing_position do Valtteri Bottas em todas as corridas para calcular a sua média. Foi criado um INNER JOIN para associar o nome do driver com a sua posição.

OUTPUT

	Média da posição final
▶	2.7

5 DRIVERS MAIS VELHOS A PONTUAR FORA DO PODIUM NO GP DA CHINA

```
SELECT MAX(YEAR(CURDATE())-YEAR(driver.date_of_birth)) AS "idade", driver.name  
FROM driver  
INNER JOIN driver_grand_prix ON driver.driver_id = driver_grand_prix.driver_id  
INNER JOIN grand_prix ON driver_grand_prix.grand_prix_id = grand_prix.grand_prix_id  
WHERE driver_grand_prix.finishing_position >= 4 AND  
driver_grand_prix.finishing_position <= 10 AND grand_prix.country="China"  
GROUP BY driver.name  
ORDER BY idade DESC  
LIMIT 5;
```

- ➔ O uso das funções MAX, YEAR e CURDATE servem para, neste caso, calcular a idade dos drivers. A querie usa da finishing_position de todos os drivers para ver quais os 7 que acabaram entre a 4ª e 10ª posição e seleciona os 5 mais velhos no GP da China. Foram feitos INNER JOINS para buscar a finishing_position dos drivers e o pais do GP.

OUTPUT

	idade	name
▶	41	Kimi Raikkonen
	31	Daniel Ricciardo
	30	Sergio Perez
	24	Alexander Albon
	24	Pierre Gasly

LISTA DOS DRIVERS E RESPETIVAS EQUIPAS

```
SELECT driver.name AS "Driver", driver.number AS "Number", team.name AS "Team"
FROM driver
INNER JOIN driver_team ON driver.driver_id = driver_team.driver_id
INNER JOIN team ON driver_team.team_id = team.team_id
ORDER BY driver.driver_id;
```

- ➔ A querie junta o nome e numero de cada driver e associa a equipa correspondente. Foram feitos INNER JOINS para poder buscar tanto os dados da equipa como os do piloto

OUTPUT

	Driver	Number	Team
►	Lewis Hamilton	44	Mercedes-AMG Petronas F1 Team
	Valtteri Bottas	77	Mercedes-AMG Petronas F1 Team
	Sebastian Vettel	5	Scuderia Ferrari Mission Winnow
	Charles Leclerc	16	Scuderia Ferrari Mission Winnow
	Lando Norris	4	McLaren F1 Team
	Carlos Sainz	55	McLaren F1 Team
	Alexander Albon	23	Aston Martin Red Bull Racing
	Max Verstappen	33	Aston Martin Red Bull Racing
	Daniel Ricciardo	3	Renault DP World F1 Team
	Nico Hulkenberg	27	Renault DP World F1 Team
	Pierre Gasly	10	Scuderia AlphaTauri Honda
	Daniil Kvyat	26	Scuderia AlphaTauri Honda
	Sergio Perez	11	BWT Racing Point F1 Team
	Lance Stroll	18	BWT Racing Point F1 Team
	Kimi Raikkonen	7	Alfa Romeo Racing ORLEN
	Antonio Giovinazzi	99	Alfa Romeo Racing ORLEN
	Romain Grosjean	8	Haas F1 Team
	Kevin Magnussen	20	Haas F1 Team
	George Russell	63	Williams Racing
	Robert Kubica	88	Williams Racing

CONCLUSÃO

Após concluirmos a nossa base de dados, chegamos a conclusão que de forma a ser eficiente é necessária organização e uma definição de objetivos a atingir para poder trabalhar em equipa é necessária comunicação entre os dois membros da equipa e partilha de informação entre ambos, apesar de algumas dificuldades iniciais em definir os nossos objetivos conseguimos realizar a base de dados assim como toda a informação que terá sido solicitado pelo professor.

Finalizamos este relatório com a certeza de ter aumentado os nossos conhecimentos sobre SQL, que com certeza será útil durante toda a nossa futura jornada como programadores, temos a agradecer ao professor Magno Andrade por todos os conhecimentos que nos foram transmitidos durante a realização de este trabalho e durante todo o semestre.