# **CREATE**

Создание таблицы drivers.

CREATE TABLE drivers (

driverId INT,

driverRef TEXT,

number INT,

code TEXT,

forename TEXT,

surname TEXT,

dob DATE,

nationality TEXT,

url TEXT

);

# **INSERT**

Добавление записи в таблицу drivers.

INSERT INTO drivers (driverId, driverRef, number, code, forename, surname, dob, nationality, url)

VALUES (844, 'speed', 17, 'BAD', 'Ksenia', 'Pavlova', '2000-10-26', 'Russian', NULL);

# **SELECT**

Получение всех данных из таблицы drivers.

SELECT \* FROM drivers;

# **WHERE**

Выборка гонщиков с национальностью British из таблицы drivers.

SELECT forename, surname

FROM drivers

WHERE nationality = 'British';

# **UPDATE**

Обновление добавленной ранее записи из таблицы drivers.

UPDATE drivers

SET url = 'https://ru.wikipedia.org/wiki/Елизавета\_II'

WHERE surname = 'Pavlova';

# **DELETE**

Удаление из таблицы добавленной ранее записи из таблицы drivers.

DELETE FROM drivers

WHERE surname = 'Pavlova';

# **ALTER**

Удаление столбца url из таблицы drivers.

ALTER TABLE drivers DROP url;

# **DISTINCT**

Выборка уникальных национальностей из таблицы drivers.

SELECT DISTINCT nationality

FROM drivers;

# **MAX**

Самая северная трасса из таблицы circuits.

SELECT name, location, country

FROM circuits

WHERE lat = (

SELECT MAX(lat)

FROM circuits

);

# **ORDER BY, LIMIT**

Топ-10 молодых гонщиков из таблицы drivers.

SELECT forename, surname, dob, date\_part('year', AGE(now(), dob)) AS age

FROM drivers

ORDER BY dob DESC

LIMIT 10;

# **COUNT, AND, OR, BETWEEN**

Количество гонщиков в возрасте от 35 до 50 лет из Франции или России из таблицы drivers.

SELECT COUNT(\*) FROM (

SELECT forename, surname, dob,

date\_part('year', AGE(now(), dob)) AS age, nationality

FROM drivers

WHERE date\_part('year', AGE(now(), dob)) BETWEEN 35 AND 50

AND (nationality = 'French' OR nationality = 'Russian')

) AS amount

# **UNION**

Вывести гонщиков в возрасте от 35 до 50 лет из Франции или России из таблицы drivers.

SELECT forename, surname, dob,

date\_part('year', AGE(now(), dob)) AS age, nationality

FROM drivers

WHERE date\_part('year', AGE(now(), dob)) BETWEEN 35 AND 50

AND nationality = 'French'

UNION

SELECT forename, surname, dob,

date\_part('year', AGE(now(), dob)) AS age, nationality

FROM drivers

WHERE date\_part('year', AGE(now(), dob)) BETWEEN 35 AND 50

AND nationality = 'Russian';

# **GROUP BY**

Вывести количество гонщиков (в порядке убывания) каждой национальности из таблицы drivers.

SELECT nationality, COUNT(forename) AS amount

FROM drivers

GROUP BY nationality

ORDER BY amount DESC;

# **HAVING**

Вывести из таблицы drivers национальности, количество гонщиков которых больше 50.

SELECT nationality, COUNT(forename) AS amount

FROM drivers

GROUP BY nationality

HAVING COUNT(forename) >= 50;

# **INNER JOIN**

Вывести информацию о каждом гонщике: имя, фамилия, очки, позиция.

SELECT forename, surname, points, position

FROM drivers

INNER JOIN driver\_standings USING(driverid);

# **LEFT JOIN**

Вывести имена и фамилии гонщиков, для которых нет информации об очках.

SELECT forename, surname, points

FROM drivers

LEFT JOIN driver\_standings USING(driverid)

WHERE points IS NULL;

# **CROSS**

Вывести позиции, на которых никогда не был гонщик с именем Kimi (но на которые попадали другие гонщики).

SELECT position

FROM drivers

CROSS JOIN driver\_standings

WHERE drivers.forename = 'Kimi'

EXCEPT

SELECT position

FROM drivers

INNER JOIN driver\_standings USING(driverid)

WHERE forename = 'Kimi'

ORDER BY position;