Instructions

• When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!

Consider the following database schema and example instance for a race car database:

Good luck

|  |  |  |  |
| --- | --- | --- | --- |
| **Racer** | **Car** | | |
| |  |  |  |  | | --- | --- | --- | --- | | **A\_id** | **Name** | **gender** | **Birth\_year** | | 1 | Sam | M | 1984 | | 2 | Suzan | F | 1982 | | 3 | Alice | F | 1980 | | |  |  |  | | --- | --- | --- | | **c\_id** | **Manufacturer** | **Hourse\_power** | | C1 | Mercedes | 1000 | | C2 | Ferrari | 5000 | | C3 | Mercedes | 700 | | | |
| **Result** | **Race Sponsor** | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **result\_id** | **c\_id** | **r\_id** | **A\_id** | **Time** | | R1\_1 | C1 | R1 | 1 | 400 | | R1\_2 | C2 | R1 | 2 | 600 | | R1\_3 | C3 | R1 | 3 | 375 | | |  |  |  | | --- | --- | --- | | **R\_id** | **Type** | **Location** | | R1 | F1 | UK | | R2 | RallyCross | Portugal | | R3 | F1 | USA | | **R\_id** | **C\_ID** | | **Sponsor** |
| R1 | C1 | | Motorola |
| R1 | C3 | | Pepsi |
| R2 | C4 | | Michelin |

**Schema:**

Racer: (A\_id, Name, gender, birth\_year)

Race (r\_id, type,location)

Result (result\_id, c\_id, A\_id,R\_id, time) c\_id is a foreign key for car and A\_id is foreign key for Racer and R\_id is a foreign key for Race

Sponsor (R\_id, C\_iD, Sponsor)

Car (c\_id, make, horsepower)

# SQL DDL (total 7 points)

I-1 (3 points) Write an SQL statement that create a new table *raceLap* that stores the race Lap’s name, location, streetName, streetNumber and the length. The primary key being the combination of name and location. The length must be positive and not left blank.

**create** **table** raceLap(name **varchar**(255) **not** **null unique**,

location **varchar**(255) **not** **null unique**,

streetName **varchar**(255), streetNumber **int**,

length **int not** **null**,

**constraint** nameLoc **primary** **key** (name, location));

I -2 (4 points) Write an SQL statement to add an attribute difficultyLevel to the relation *raceLap*. Add an attribute name to the relation *race* and set name, location to be a foreign key in the relation *race* references the relation *raceLap*. In case of lap record is removed from the *raceLap*, this will reflect on the *race* table and remove. Note values of difficultyLevel are 1,2,3

**alter** **table** raceLap

**add** difficultyLevel **int** **check**(difficultyLevel = 1 **or** difficultyLevel = 2 **or** difficultyLevel = 3);

**alter** **table** race

**add** name **varchar**(255) **not** **null unique**, **add** **foreign** **key** (name,location) **references** raceLap(name,location);

# SQL Queries (total 30 points)

Question I.1 (3 Points) Write the SQL statement that returns the sponsor, the race and the racer name they sponsored.

**select** s.sponsor, Result.r\_id, r.name

**from** Result

**inner** **join** Sponsor **as** s

**on** s.r\_id = Result.r\_id

**inner** **join** Racer **as** r

**on** r.a\_id = Result.a\_id;

Question II-2 (4 Points) Write the SQL statement that returns the male racer ‘s name, born before 1982 with their average race time.

**select** r.name, **avg**(racetime)

**from** result

**inner** **join** racer **as** r

**on** r.a\_id = result.a\_id

**where** r.birth\_year < 1982

**group** **by** name;

Question II-3 (4 Points) Write the SQL statement that returns the racer’s name and associated race sponsor that drove a ‘Mercedes’ car during at least one of their races.

**select** **distinct** r.name, s.sponsor

**from** result

**inner** **join** racer **as** r

**on** r.a\_id = result.a\_id

**inner** **join** Sponsor **as** s

**on** s.r\_id = result.r\_id

**inner** **join** car

**on** car.c\_id = result.c\_id

**where** car.manufacturer = 'Mercedes';

Question II-4 (5 points). Write an SQL query that returns the manufacturer whose cars won the most races

**select** manufacturer **from** car

**inner** **join** (**select** result1.c\_id, **count**(result1.c\_id) **as** counting

**from** result **as** result1

**inner** **join**(**select** r\_id, **min**(time) **as** time

**from** result **group** **by** r\_id) **as** result2 **on**

result1.r\_id = result2.r\_id

**and** result1.time = result2.time

**group** **by** result1.c\_id

**order** **by** counting **desc** **limit** 1) **as** winner **on** car.c\_id = winner.c\_id;

Question II-5 (3 points) Write an SQL query that returns the racer’s name that only drives Mercedes.

**select** r.name

**from** result

**inner** **join** racer **as** r

**on** r.a\_id = result.a\_id

**inner** **join** car

**on** car.c\_id = result.c\_id

**group** **by** r.name

**having** **sum**(**case** **when** car.manufacturer != 'Mercedes' **then** 1 **else** 0 **end**) = 0;

Question II-6 (4 points) Write an SQL query that returns the racer’s name that never participated in a ‘rallycross’ race

**select** r.name

**from** result

**inner** **join** racer **as** r

**on** r.a\_id = result.a\_id

**inner** **join** race

**on** race.r\_id = result.r\_id

**group** **by** r.name

**having** **sum**(**case** **when** race.type = 'RallyCross' **then** 1 **else** 0 **end**) = 0;

Question II-7. (3 points) Write an SQL query that returns the manufacturer for cars with low horsepower, meaning all its manufactured cars are less than 800 horsepower.

**select** car.manufacturer

**from** car

**group** **by** car.manufacturer

**having** **sum**(**case** **when** car.horsepower > 800 **then** 1 **else** 0 **end**) = 0;

Question II .8. (4 points) Write an SQL query that returns the cars that finished the race in time smaller than all races average time.

**select** car.c\_id

**from** car

**inner** **join** result **as** r

**on** r.c\_id = car.c\_id

**group** **by** car.c\_id, r.time

**having** r.time < (**select** **avg**(time) **from** result);

# SQL updates (total 13 points)

Question III .1 (2 points) Write an SQL statement that increment the birth year by 2 for all the female racer

**update** racer

**set** birth\_year = birth\_year + 2

**where** gender = 'F';

Question III-2 (3 points) Write an SQL statement that reduces the horsepower by 5% of all ‘Ferrari’ car manufacturer and increases the horsepower by 10% for all the ‘Mercedes’ car manufacturer

**update** car

**set** horsepower = **case**

**when** car.manufacturer = 'Ferrari' **then** 0.95 \* horsepower

**when** car.manufacturer = 'Mercedes' **then** 1.10 \* horsepower

**else** horsepower

**end**;

Question III-3 (3 points) Write an SQL statement that deletes the all race results whose participant are over 80 years old by the end of 2019.

**delete** **from** result

**using** racer **as** r

**where** r.birth\_year < 1939

**and** r.a\_id = result.a\_id;

Question III. 4 (5 points) Insert a goodyear sponsor for all the race taken place in Italy and for ‘Ferrari’ manufactured car.

**insert** **into** sponsor(r\_id, c\_id, sponsor)

**select** **distinct** race.r\_id, car.c\_id, 'GoodYear'

**from** race,car,result

**where** car.manufacturer = 'Ferrari' **and** race.location = 'Italy'

**and** result.r\_id = race.r\_id **and** result.c\_id = car.c\_id

**group** **by** race.r\_id, car.c\_id;