

## Machine test - Django/ Laravel(Database)

### Instructions

- Six questions in total.
- Maximum 10 points for each question.
- Points will be awarded for correctness and also for the simplicity and elegance of your code.

1. The following script creates and populates a table with some sample data.

```
CREATE TABLE rawdata (txt VARCHAR(100) PRIMARY KEY);

INSERT INTO rawdata (txt)
VALUES
  ('P237JONES|02AUG1979')
,('A71831ROGERSON|12MAY1990')
,('P192HALL|27JUN2001')
,('A11102DOUGLAS|07AUG1985')
,('A91811LLOYD|19OCT2000')
,('A31222WALSH|15FEB1997')
,('P915MERRYMAN|02JUL1989');
```

Create a query that queries this rawdata table and returns the **prefix**, **idnum**, **name** and **date** as separate columns. The date column has to be converted to a DATE type. The result of your query should look like the following result.

You can assume that where the prefix is **P** the idnum will consist of exactly 3 digits and where the prefix is **A** the idnum will consist of exactly 5 digits. You can also assume that there is a single characters ( | ) dividing the name from the date.

```
prefix idnum name date
-----
P 237 JONES 1979-08-02
A 71831 ROGERSON 1990-05-12
P 192 HALL 2001-06-27
A 11102 DOUGLAS 1985-08-07
A 91811 LLOYD 2000-10-19
A 31222 WALSH 1997-02-15
P 915 MERRYMAN 1989-07-02
```

2. **(10 points)** Create the following table and data which includes employee numbers and names and their managers where applicable.

```
CREATE TABLE Org
(emp_no INTEGER NOT NULL PRIMARY KEY,
 emp_name VARCHAR(20) NOT NULL,
 manager_no INT NULL REFERENCES Org (emp_no) );
```

```
INSERT INTO Org
VALUES
(102, 'Sean', NULL)
,(103, 'Richard', 102)
,(104, 'Joanna', 102)
,(105, 'Frank', 103)
,(106, 'Danny', 103)
,(107, 'Alice', 105)
,(108, 'Eve', 106);
```

Write a query to show the employee number and name and the number of immediate subordinates each employee has. The result should look like this:

```
emp_no emp_name num_direct_reports
-----
102 Sean 2
103 Richard 2
104 Joanna 0
105 Frank 1
106 Danny 1
107 Alice 0
108 Eve 0
```

3. **(10 points)** The **airplane** and **flight** tables and sample data are defined below. These tables record journeys for different aircraft. Write a query to show one row per aircraft with the following data for each of the four aircraft in the flight table: 1) the tailnumber, 2) the model, where known, 3) the total hours flown, 4) the total number of flights, 5) the total number of just those flights which lasted 4 hours or more. The result of your query should look like this:

```
tailnumber model hours total_flights four_hr_flights -----
-----
D-IEVX 1 1 0
N155UP Airbus A300 11 3 1
A6-ERG Airbus A340 3 2 0
SP-LPC Boeing 767 18 2 2
```

```
CREATE TABLE airplane
(model VARCHAR(25), tailnumber VARCHAR(12) PRIMARY KEY, code CHAR(3) NOT NULL);
```

```
CREATE TABLE flight
(departuredt DATETIME2(0), arrivaldt DATETIME2(0), origin
```

```
CHAR(3),destination CHAR(3),tailnumber VARCHAR(12), PRIMARY KEY
(departuredt,arrivaldt,tailnumber));
```

```
INSERT INTO airplane
values
```

```
('Airbus A300','N155UP','AB3')
,('Airbus A340','A6-ERG','340')
,('Airbus A300','N155UQ','AB3')
,('Airbus A340','A6-ERH','340')
,('Boeing 727','D-ABCE','727')
,('Boeing 767','SP-LPC','767')
,('Lockheed L-188 Electra','CH-06','LOE');
```

```
INSERT INTO flight ( departuredt, arrivaldt, origin, destination,
tailnumber ) VALUES
```

```
('2018-01-10 09:17:00', '2018-01-10 11:20:00', 'LHR', 'BFS', 'A6-ERG')
,('2018-03-21 09:45:00', '2018-03-21 15:25:00', 'BGI', 'LHR', 'N155UP')
,('2018-04-02 20:15:00', '2018-04-02 22:30:00', 'AMS', 'BFS', 'N155UP')
,('2018-04-05 05:35:00', '2018-04-05 14:45:00', 'LHR', 'BGI', 'SP-LPC')
,('2018-06-13 11:45:00', '2018-06-13 14:55:00', 'LHR', 'AMS', 'N155UP')
,('2018-07-12 06:25:00', '2018-07-12 07:50:00', 'AMS', 'LGW', 'A6-ERG')
,('2018-11-14 12:20:00', '2018-11-14 13:25:00', 'LGW', 'BFS', 'D-IEVX')
,('2019-04-05 05:35:00', '2019-04-05 14:36:00', 'LHR', 'BGI', 'SP-LPC');
```

4. **(10 points)** Using the same tables and data as in the question above, write a query that lists the same 4 aircraft and the total number of different airports visited by each. The airports are given by the IATA airport code in the origin and destination columns. If a given aircraft visits an airport multiple times, either as origin or destination, then that airport should only be counted once. The correct result should be as follows:

```
tailnumber num_airports
-----
A6-ERG 4
D-IEVX 2
N155UP 4
SP-LPC 2
```

5. Create the following tables and data:

```
CREATE TABLE customer(
customer_num int NOT NULL PRIMARY KEY,
title varchar(5) NULL,
first_name varchar(20) NOT NULL,
last_name varchar(20) NOT NULL);
```

```
CREATE TABLE purchase(
customer_num int NOT NULL PRIMARY KEY,
purchase_date DATETIME2(0) NOT NULL,
amount DECIMAL(8,2) NOT NULL);
```

```
INSERT INTO customer ( customer_num, title, first_name, last_name )
```

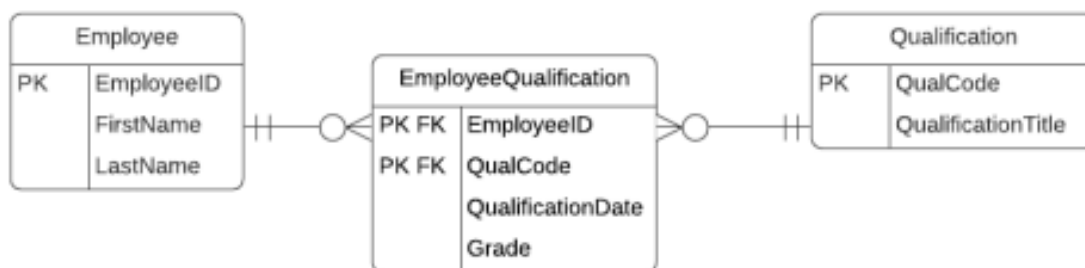
```
VALUES
  (103, 'Mr', 'Blake', 'Cooper')
, (106, 'Ms', 'Gabriel', 'Howard')
, (116, 'Mr', 'Alex', 'Sanders')
, (132, 'Ms', 'Jamie', 'Butler')
, (135, NULL, 'Carter', 'Gonzales')
, (142, 'Mrs', 'Amelia', 'Myers')
, (199, NULL, 'Bella', 'Stone');

INSERT INTO purchase (customer_num, purchase_date, amount)
VALUES
  (103, '2018-03-04', 122.50)
, (106, '2018-03-04', 149.50)
, (116, '2018-03-05', 89.99)
, (132, '2018-03-05', 71.24)
, (135, '2018-03-06', 29.99)
, (142, '2018-03-06', 24.50)
, (199, '2018-03-06', 161.20);
```

Write a query to show the customer name, purchase date and amount for each purchase where the purchase\_amount is greater than the average for the entire table. Calculate the average in your query rather than specify it as a fixed amount.

```
first_name last_name purchase_date amount
-----
Blake Cooper 2018-03-04 122.50
Gabriel Howard 2018-03-04 149.50
Bella Stone 2018-03-06 161.20
```

6. **(10 points)** Create three tables as shown in the following ER diagram. Your tables should include primary and foreign keys and specify suitable data types for each column. EmployeeID in the Employee table should be an IDENTITY column, Grade should be a nullable VARCHAR column and all other columns should be non-nullable.



**THE END**