

- [L3 MIAHS/Ingémath](#)
- [Université Paris Cité](#)
- Année 2023-2024
- [Course Homepage](#)
- [Moodle](#)



Écrire une fonction SQL nommée `cc_fonc_16` qui prend en argument une année `p_year`, un mois `p_month` un jour `p_day` et un aéroport de départ `p_origin` et renvoie une table de schéma

(`avg_temp float8`, `avg_visib float8`, `avg_wind_speed float8`, `avg_month_temp float8`)

qui contient pour la journée définie par `p_year`, `p_month` et `p_day`, l'aéroport défini par `p_origin`, la température moyenne de la journée `avg_temp`, la visibilité moyenne de la journée `avg_visib`, la vitesse moyenne du vent `avg_wind_speed` de la journée, et la température moyenne du mois `avg_month_temp`.

Réponse attendue pour :

```
SELECT round(avg_temp::numeric, 2) AS avg_temp,
       round(avg_visib::numeric, 2) AS avg_visib,
       round(avg_wind_speed::numeric, 2) AS avg_wind_speed,
       round(avg_month_temp::NUMERIC, 2) AS avg_month_temp
FROM cc_fonc_16(2013, 2, 16, 'EWR');
```

```
+-----+-----+-----+-----+
|avg_temp|avg_visib|avg_wind_speed|avg_month_temp|
+-----+-----+-----+-----+
| 36.07| 9.71| 12.37| 34.26|
```

```
CREATE OR REPLACE FUNCTION correction_cc3.cc_fonc_16(
    p_year int4, p_month int4, p_day int4, p_origin text)
    RETURNS TABLE(avg_temp float8, avg_visib float8, avg_wind_speed float8, avg_month_temp float8)
    LANGUAGE sql
AS $function$
WITH v AS (
    SELECT AVG(w.temp) AS avg_month_temp
    FROM nycflights13.weather w
    WHERE w."year"=p_year AND w."month"=p_month AND w.origin=p_origin
),
x AS (
    SELECT AVG(w.temp) AS avg_temp,
           AVG(w.visib) AS avg_visib,
           AVG(w.wind_speed) AS avg_wind_speed
    FROM nycflights13.weather w
    WHERE w."year"=p_year AND w."month"=p_month AND w."day"=p_day AND w.origin=p_origin
)
SELECT x.avg_temp, x.avg_visib, x.avg_wind_speed, v.avg_month_temp
FROM x, v
$function$ ;
```

Données dans le schéma `nycflights13`.

Écrire une fonction SQL nommée `cc_fonc_17` qui prend en argument un constructeur d'avion et qui renvoie une table de schéma

(`n_planes int8`, `n_carriers int8`)


chaque ligne donne pour chaque modèle vendu par le constructeur,

- le nombre d'avions de ce modèle qui ont circulé sur des vols domestiques partis de New York d'après la table `flights`
- le nombre de compagnies qui exploitent ce modèle (d'après la base `nycflights13`)
- le nombre de vols effectués par les exemplaires du modèle à partir de New York d'après la table `flights`

Réponse attendue pour :

```
SELECT *
FROM cc_fonc_17('AIRBUS')
WHERE model='A330-243' ;
```

```
+-----+-----+-----+-----+
|model  |n_planes|n_carriers|n_flights|
+-----+-----+-----+-----+
|A330-243|      14|         1|      342|
```

 **solution**

TODO

Données dans le schéma nycflights13

Écrire une vue nommée cc_vue_18 de schéma

(origin text, time_hour timestamp, flight int4, dep_delay float8, next_flight int4, next_dep_delay float8)


Pour chaque vol effectué f, la vue contient l'aéroport de départ (origin), la date et l'heure de départ programmée arrondie (time_hour), le numéro du vol flight, le retard au départ dep_delay, et aussi le numéro next_flight et le retard au départ du vol next_dep_delay, qui a décollé juste après le vol f du même aéroport de départ.

Réponse attendue pour

```
SELECT r.origin, r.time_hour::timestamp, r.flight, r.dep_time, r.dep_delay, r.next_flight, r.next_dep_delay
FROM cc_vue_18 AS r
WHERE r.origin = 'EWR' AND r.time_hour = to_timestamp('2013-01-01 06:00:00', 'YYYY-MM-DD HH:MI:SS')
LIMIT 5 ;
```

```
+-----+-----+-----+-----+-----+-----+-----+
|origin|time_hour                |flight|dep_time|dep_delay|next_flight|next_dep_delay|
+-----+-----+-----+-----+-----+-----+-----+
|EWR   |2013-01-01 06:00:00.000|  507|    555|     -5.0|    1124|         -2.0|
|EWR   |2013-01-01 06:00:00.000|  1124|    558|     -2.0|    1187|         -1.0|
|EWR   |2013-01-01 06:00:00.000|  1187|    559|     -1.0|     343|          1.0|
|EWR   |2013-01-01 06:00:00.000|   343|    601|          1.0|   1895|        -4.0|
|EWR   |2013-01-01 06:00:00.000|  1895|    606|     -4.0|   1077|          0.0|
```

NB : précision demandée pour la popularité 0.01

 **solution**

TODO

Données dans le schéma nycflights13

Écrire une vue nommée cc_vue_19 de schéma :


(origin text, dest text, year int4, month int4, carrier text)

Pour chaque liaison (origin, dest), chaque mois (year, month), carrier contient le code de la compagnie du mois, celles dont le retard moyen à l'arrivée sur cette liaison, pendant ce mois est minimal.

```
SELECT *
FROM cc_vue_19 t
WHERE t.origin='JFK' AND t.dest='ATL'
ORDER BY t."year", t"month"
```

```
LIMIT 5  
;
```

origin	dest	year	month	carrier
JFK	ATL	2013	1	9E
JFK	ATL	2013	2	DL
JFK	ATL	2013	3	DL
JFK	ATL	2013	4	DL
JFK	ATL	2013	5	DL

 **solution**


TODO

Données dans le schéma `babynames`

Créer une vue `cc_vue_20` de schéma

(`origin` TEXT, `dest` TEXT, `flight` int4, `time_hour` TIMESTAMP, `arr_delay` , `carrier` TEXT, `model` TEXT, s

qui liste pour chaque vol de la table `flights`, l'aéroport de départ, l'aéroport de destination, la date et l'heure de départ prévues trinquée à l'heure (`time_hour`), le modèle (`model`) de l'avion, sa compagnie (`carrier`), son retard à l'arrivée, ainsi que pour un des vols les plus rapides sur cette liaison pendant le mois défini par `time_hour`, le modèle `speedy_model` et la compagnie `speedy_carrier`.

 **solution**

TODO