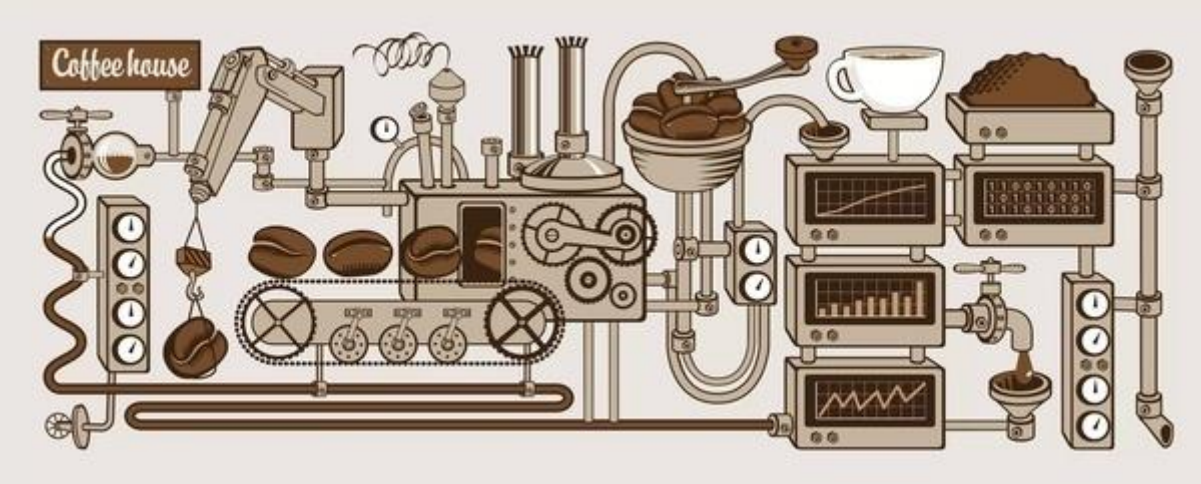
Projet :

<u>Secteur</u> des *transports* ; <u>Problématique *métier*</u> : *Trafic aérien*
<u>Données *réelles*</u>, *source : BTS*.

Mise en situation

Vous êtes *Consultant Data* au sein d'Aéroports de Paris (**ADP**), fraîchement embauché depuis une semaine ! Vous avez fait connaissance avec vos collègues, votre nouveau bureau, mais surtout, la machine à café high-tech :



Rien que ça !

Mais revenons à vos missions : il est temps de mettre les mains dans le cambouis ! Le **DSI** vous a donné l'accès à la DB : à vous de vous familiariser avec et d'analyser sa data. Votre **manager** souhaite que vous réalisiez très prochainement une présentation. L'**ADP** fait en effet face à un trafic aérien en constante augmentation annuellement MAIS en parallèle à un nombre croissant de dysfonctionnements : *retards*, *annulation de vols*, *passagers qui passent la nuit à l'aéroport*, …, et à d'éventuels autres problèmes à découvrir lors de votre fouille dans la DB.

1. Mission : Se familiariser avec les données (barème : 10 pts, chaque question : 1.25 pts)

1. Combien y-a-t-il :

   - d'aéroports en tout, de départ et de destination
   - Combien y-a-t-il d'aéroports où on ne passe pas à l'heure d'été (*indice : colonne dst : 23*) ? de fuseaux horaires (*10 voir colone tzone dont une « N »*)
   - de compagnies, d'avions, de vols annulés ?

2. Quel est l'aéroport de départ le plus emprunté ? Quelles sont les 10 destinations les plus (moins) prisées (*en indiquant le nom complet des destinations et non juste le code correspondant et le % corresp pour chaque destination*) ? Quelle sont les 10 avions qui ont le plus (moins) décollé ?

3. Combien chaque compagnie a desservi de destination ; combien chaque compagnie a desservie de destination par aéroport d'origine. Réaliser les graphiques adéquats qui synthétisent ces informations ?

4. Trouver tous les vols ayant atterri à Houston (IAH ou HOU) ? Combien de vols partent de NYC airports vers Seattle, combien de compagnies desservent cette destination et combien d'avions "uniques" ?

5. Trouver le nombre de vols par destination ? Trier les vols suivant la destination, l'aéroport d'origine, la compagnie dans un ordre alphabétique croissant (en réalisant les jointures nécessaires pour obtenir les noms explicites des aéroports) ?

6. Quelles sont les compagnies qui n'opèrent pas sur tous les aéroports d'origine ? Quelles sont les compagnies qui desservent l'ensemble de destinations ?
   Faire un tableau où l'on récupère l'ensemble des origines et des destinations pour l'ensemble des compagnies.

7. Quelles sont les destinations qui sont exclusives à certaines compagnies ?

8. Filtrer le vol pour trouver ceux exploités par United, American ou Delta ?

2. Mission : Créer la DB (barème : 10 pts)

Important : Recréer la BD **SQL**/NoSQL de l'entreprise et la mettre en production sur une plateforme de votre choix : Azure, AWS, alwaysdata, planetscale, db4free, remotemysql, …. . Je suis également ouvert à vos choix : MySQL, Oracle, SQL Server, MongoDB, … en passant ou non par Docker.

N'oubliez pas de sécuriser la connexion à la BD. Pour vous aider : voici un 1er lien, un guide pratique pour la configuration du DSN (en Fr et en En) et un dernier lien vers la documentation offcielle (si vous avez optez pour R)

Pour vous guider dans la construction de la DB, je vous ai préparé la liste des questions ci-dessous (ça va vous permettre aussi de se familiariser avec la data). Vous y répondiez dans un 1er temps avec SQL et vous vérifiez les résultats avec Python/R dans un 2è temps.

- *Indices sur les relations :*
  - Chaque vol a un avion et chaque avion effectue plusieurs vols (*one to many*).
  - Chaque compagnie a des vols vers plusieurs aéroports et chaque aéroport reçoit plusieurs compagnies (*many to many*).
  - ……
- *Indices sur les clés primaires :*
  - Une `PK` identifie chaque observation de manière unique et ne comporte pas de données manquantes.
  - `faa` est celle d'`airports` : *code de chaque aéroport*. Cette clé respecte, par ailleurs, une nomenclature type (comme c'est le cas par exemple du *num de la sécurité sociale,* le num *SIRET, ...*) => *clé naturelle*. Vous devriez alors **confirmer**, tout au long des observations, que cette nomenclature est respectée : ***`Utiliser des regex`*** ?

    Cette vérification est à reproduire sur les colonnes `origin` et `dest` (`FK` de la table `weather`), `carrier` (table `airlines`) et `tailnum` (table `planes`)?

=> Ceci permet d'éviter toute surprise ou anomalie lors de l'injection des csv dans les tables !

- `tailnum` est la `PK` de la table `planes` ;
- `carrier` est celle de `airlines` ;
- `year`, `month`, `day`, `hour`, `origin` est une `PK` composite de la table `weather` ;
- Vérifier que le numéro de vol (`flight`) n'est une PK pour la table `flights` mais la combinaison de cette dernière colonne avec `year`, `month`, `day`, `hour` et `carrier` (nom de la compagnie) l'est ?

- *Indices sur les clés étrangères :*
  - *Pb sur la FK entre flights et airports => 4 aéroports qui n'apparaissent pas dans la table airports alors qu'ils le sont dans la table flights :*
    - 'BQN', 'Rafael Hernandez Airport',
    - 'PSE', 'Mercedita Airport',
    - 'SJU', 'San Juan Airport',
    - 'STT', 'Cyril E. King Airport';
  - *Pb sur la FK entre flights et planes => des centaines d'avions qui n'apparaissent pas dans la table planes alors qu'ils le sont dans la table flights.*
  - *Eventuels autres Pb de FK …*

3. Mission : Créer une WebApp (barème : bonus pts)

- Pour le ***jour j*** (la présentation) et afin d'attirer l'attention vos collègues et de les impressionner, vous allez opter pour une **application web**. Vous avez le choix entre Shiny pour R ou Flask/Django pour Python (à défaut un Jupyter notebook peu importe le langage). Trouver un 1è lien pour démarrer avec Shiny et un 2è pour connecter Shiny à la DB cliquer ici.
- Votre rôle est de donner le maximum d'éléments (**reporting à partir des données du passé**) pour prédire les problèmes et ainsi mieux se préparer (**analyse prédictive**). L'objectif est d'aider votre manager à la prise de décision.

Ressources pour vous aider :

Aide-mémoire : dplyr / pandas / SQL

Dplyr

- Dplyr verbs:
  - select ~ SELECT
  - filter ~ WHERE
  - arrange ~ ORDER
  - summarise ~ aggregators: sum, min, sd, etc.
  - mutate ~ operators: +, *, log, etc.
- Grouping: group_by ~ GROUP BY
- Window functions: rank, dense_rank, percent_rank, ntile, row_number, cume_dist, first_value, last_value, lag, lead
- Performing joins: inner_join, semi_join, left_join, anti_join, full_join
- Sampling: sample_n, sample_frac

Combine Data Sets

a

x1	x2
A	1
B	2
C	3

+

b

x1	x3
A	T
B	F
D	T

=

Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

dplyr::**left_join(a, b, by = "x1")**
Join matching rows from b to a.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

dplyr::**right_join(a, b, by = "x1")**
Join matching rows from a to b.

x1	x2	x3
A	1	T
B	2	F

dplyr::**inner_join(a, b, by = "x1")**
Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

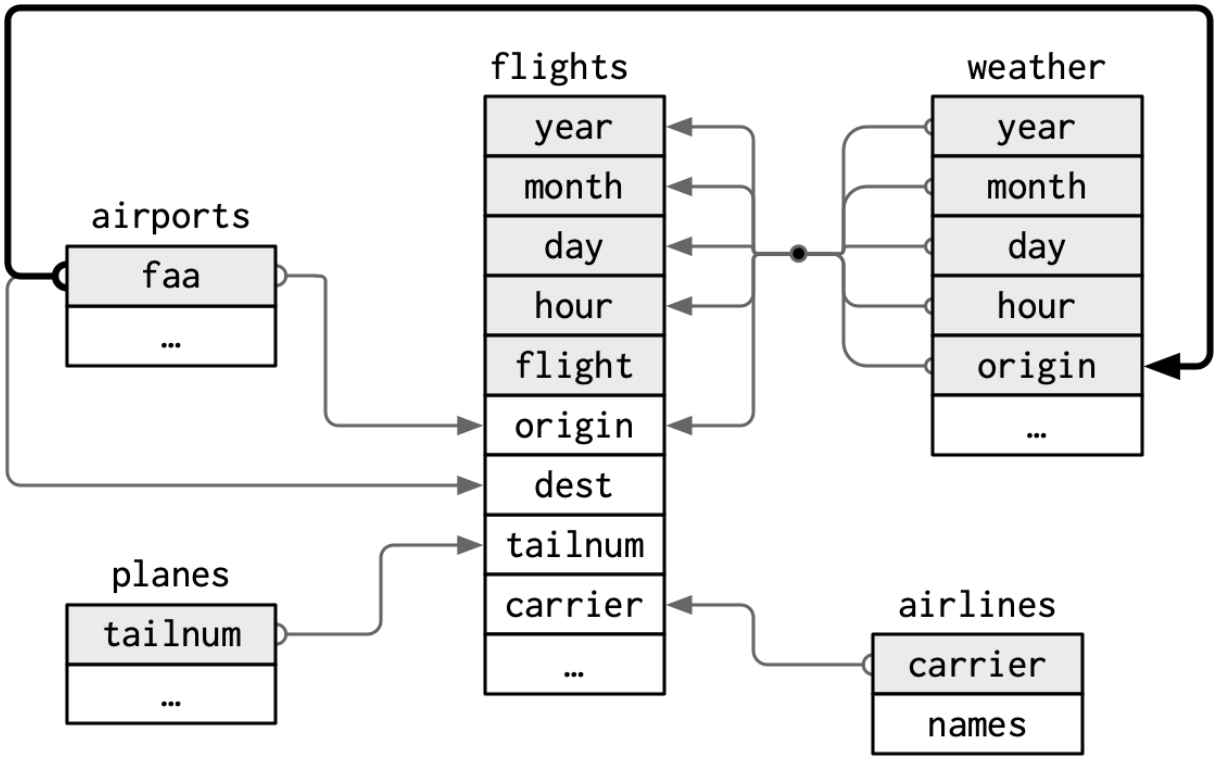
dplyr::**full_join(a, b, by = "x1")**
Join data. Retain all values, all rows.

Annexes

Données :

Source: The data is collected by the Office of Airline Information, Bureau of Transportation Statistics (BTS)

Une base de données relationnelle qui contient des informations sur les 326 776 vols ayant décollés de New York City dont le schéma est le suivant :



Conseil : ne prenez pas ce schéma pour argent comptant. Il est à améliorer

Description et Help sur les données :

*Table 1: **Flights***

Description : *data for all flights that departed NYC (i.e. JFK, LGA or EWR).*

Columns :

- year, month, day =>*Date of departure*
- dep_time, arr_time =>*Actual departure and arrival times (format HHMM or HMM), local tz.*
- sched_dep_time, sched_arr_time =>*Scheduled departure and arrival times (format HHMM or HMM), local tz.*
- dep_delay, arr_delay => Departure and arrival delays, in minutes. Negative times representearlydepartures/arrivals.
- hour, minute => Time of scheduled departure broken into hour and minutes.

- carrier => Two letter carrier abbreviation. See airlines table to get name
- tailnum => Plane tailnumber
- flight => Flight number
- origin,dest => Origin and destination. See airports table for additional metadata.
- air_time => Amount of time spent in the air, in minutes
- distance => Distance between airports, in miles
- time_hour => Scheduled date and hour of the flight as a POSIXct date along with origin, can be used to join flights data to weather data.

Source du help : RITA, Bureau of transportation statistics, https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236

*Table 2 :**Airports***

Description :Useful metadata about airports.

Columns :

- faa => FAA airport code
- name => Usual name of the airport
- lat,lon => Location of airport
- alt => Altitude, in feet
- tz =>Timezone offset from GMT
- dst => Daylight savings time zone. **A** = Standard US DST : *starts on the second Sunday of March, ends on the first Sunday of November*. **U** = *unknown*. **N** = *no dst*.
- tzone => IANA time zone, as determined by GeoNames webservice

Source du help :http://openflights.org/data.html, downloaded 2014-06-27

*Table3 :**Airlines***

Description: Look up airline names from their carrier codes.

Columns:

- Carrier =>Twoletterabbreviation
- Name => Full name

Source:https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236

*Table 4 :**Planes***

Description: Plane metadata for all plane tail numbers found in the FAA aircraft registry. American Airways (AA) and Envoy Air (MQ) report fleet numbers rather than tail numbers so can't be matched.

Columns:

- tailnum => Tail number
- year =>Year manufactured
- type => Type of plane
- manufacturer, model => Manufacturer and model
- engines, seats => Number of engines and seats
- speed => Average cruising speed in mph
- engine => Type of engine

Source du help : FAA Aircraft registry,

http://www.faa.gov/licenses_certificates/aircraft_certification/aircraft_registry/releasable_aircraft_download/

*Table 5 :**Weather***

Description : Hourly meterological data for LGA, JFK and EWR.

Columns :

- Origin => Weather station. Named origin to faciliate merging with flights() data
- Year, month, day, hour => Time of recording
- Temp, dewp => Temperature and dewpoint in F
- Humid => Relative humidity
- Wind_dir,wind_speed,wind_gust => Wind direction (in degrees), speed and gust speed (in mph)
- Precip =>Precipitation, in inches
- Pressure => Sea level pressure in millibars
- Visib =>Visibility in miles
- Time_hour => Date and hour of the recording as a POSIXct date

Source du help : ASOS download from Iowa Environmental Mesonet, https://mesonet.agron.iastate.edu/request/download.phtml