JUNIA Grande école d'ingénieurs
HEI·ISEN·ISA

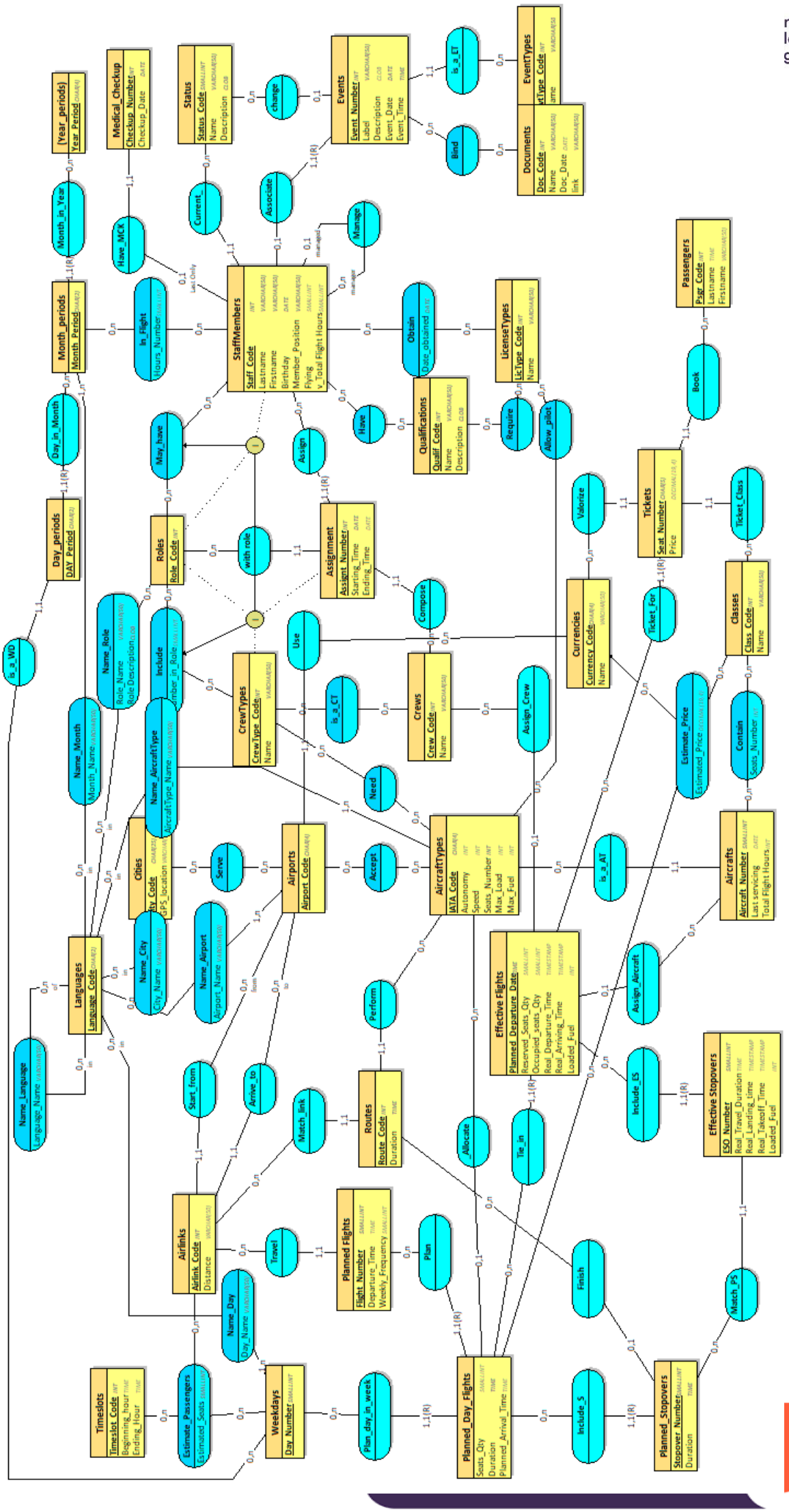
Compte rendu de Base de données

Groupe 6

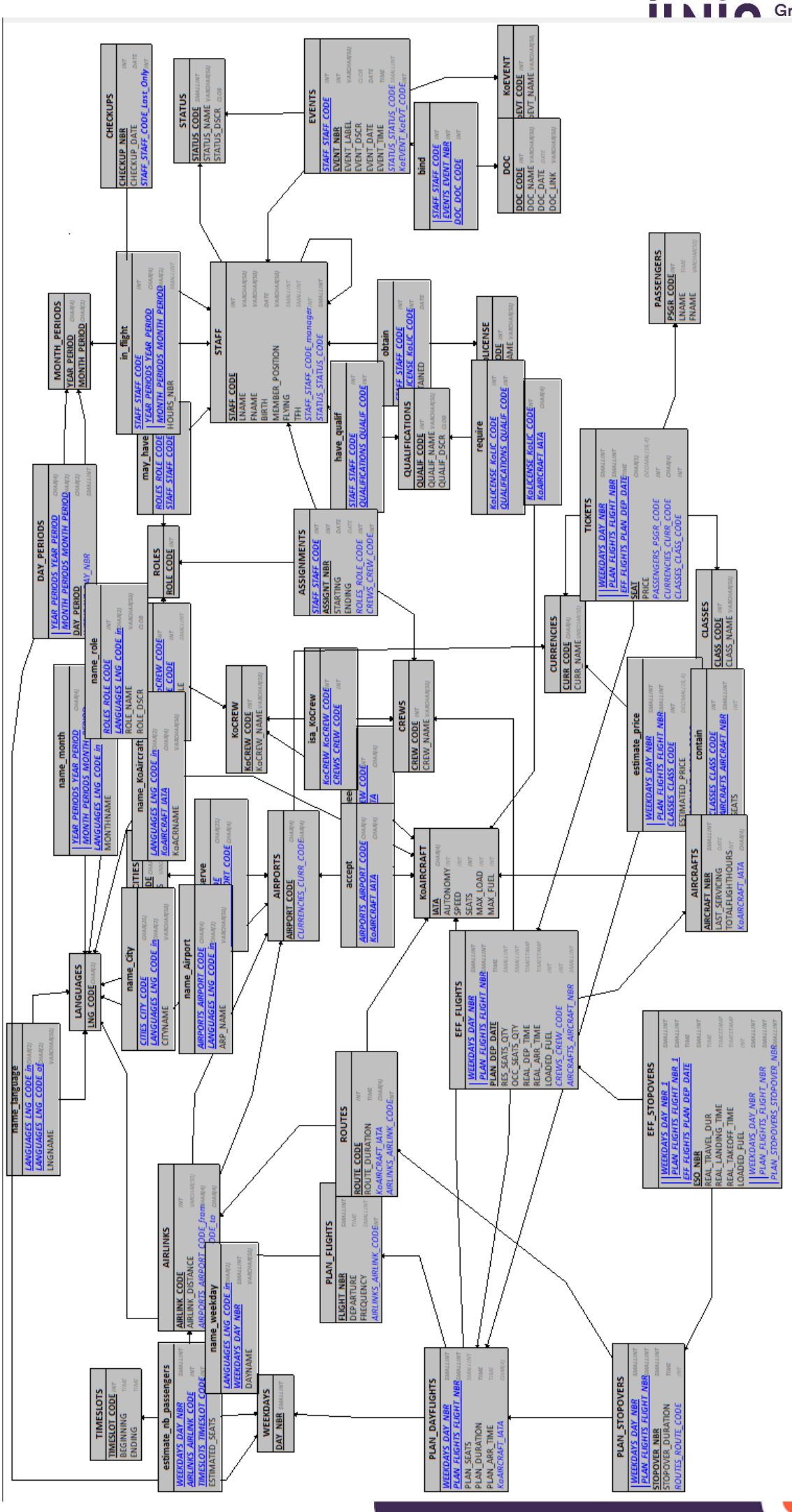
Louis Lecouturier - Sam Leclercq - Jules Maisonnave - Franck Valmont - Antoine Maes

JUNIA Grande école d'ingénieurs

MCD


(Year_periods)
Year_Period
Medical_Checkup
Checkup_Number
Checkup_Date
Status
Status_Code
Name
Description
Events
Event_Number
Label
Description
Event_Date
Event_Time
EventTypes
Documents
Doc_Code
Name
Doc_Date
link
Month_in_Year
Have_MCR
Current_
Associate
change
Is_a_ET
Bind
Month_periods
Month_Period
In_Flight
Hours_Number
StaffMembers
Staff_Code
Lastname
Firstname
Birthday
Member_Position
Flying
y_Total Flight Hours
Manage
Obtain
Date_obtained
LicenseTypes
LicType_Code
Name
Passengers
Pasgr_Code
Lastname
Firstname
Book
Day_in_Month
Day_periods
DAY_Period
May_have
Roles
Role_Code
with role
Assign
Assignment
Assignt_Number
Starting_Time
Ending_Time
Have
Qualifications
Qualif_Code
Name
Description
Require
Allow_pilot
Valorize
Tickets
Seat_Number
Price
Ticket_Class
Is_a_WD
Name_Role
Role_Name
Role_Description
Include
Member_in_Role
Compose
Use
Currencies
Currency_Code
Name
Ticket_For
Classes
Class_Code
Name
Name_Month
Month_Name
Name_AircraftType
AircraftType_Name
CrewTypes
CrewType_Code
Name
Is_a_CT
Crews
Crew_Code
Name
Assign_Crew
Estimate_Price
Estimated_Price
Contain
Seats_Number
Cities
City_Code
GPS_location
Need
Serve
Airports
Airport_Code
Accept
AircraftTypes
IATA_Code
Autonomy
Speed
Seats_Number
Max_Load
Max_Fuel
Is_a_AT
Aircrafts
Aircraft_Number
Last servicing
Total Flight Hours
Name_City
City_Name
Name_Airport
Airport_Name
Languages
Language_Code
Perform
Effective Flights
Planned_Departure_Date
Reserved_Seats_Qty
Occupied_seats_Qty
Real_Departure_Time
Real_Arriving_Time
Loaded_Fuel
Assign_Aircraft
Name_Language
Language_Name
Start_from
Arrive_to
Match_link
Routes
Route_Code
Duration
Include_ES
Effective Stopovers
ESO_Number
Real_Travel_Duration
Real_Landing_Time
Real_Takeoff_Time
Loaded_Fuel
Allocate
Tie_in
Airlinks
Airlink_Code
Distance
Travel
Planned Flights
Flight_Number
Departure_Time
Weekly_Frequency
Plan
Finish
Name_Day
Day_Name
Match_PS
Timeslots
Timeslot_Code
Beginning_Hour
Ending_Hour
Estimate_Passengers
Estimated_Seats
Weekdays
Day_Number
Plan_day_in_week
Planned_Day_Flights
Seats_Qty
Duration
Planned_Arrival_Time
Include_S
Planned_Stopovers
Stopover_Number
Duration

MLD


CHECKUPS
CHECKUP_NBR
CHECKUP_DATE
STAFF_STAFF_CODE_Last_Only
STATUS
STATUS_CODE
STATUS_NAME
STATUS_DSCR
EVENTS
STAFF_STAFF_CODE
EVENT_NBR
EVENT_LABEL
EVENT_DSCR
EVENT_DATE
EVENT_TIME
STATUS_STATUS_CODE
KoEVENT_KoEVT_CODE
KoEVENT
EVT_CODE
EVT_NAME
bind
STAFF_STAFF_CODE
EVENTS_EVENT_NBR
DOC_DOC_CODE
DOC
DOC_CODE
DOC_NAME
DOC_DATE
DOC_LINK
MONTH_PERIODS
YEAR_PERIOD
MONTH_PERIOD
in_flight
STAFF_STAFF_CODE
YEAR_PERIODS_YEAR_PERIOD
MONTH_PERIODS_MONTH_PERIOD
HOURS_NBR
STAFF
STAFF_CODE
LNAME
FNAME
BIRTH
MEMBER_POSITION
FLYING
TFH
STAFF_STAFF_CODE_manager
STATUS_STATUS_CODE
obtain
STAFF_STAFF_CODE
LICENSE_KoLIC_CODE
TRAINED
LICENSE
CODE
NAME
may_have
ROLES_ROLE_CODE
STAFF_STAFF_CODE
have_qualif
STAFF_STAFF_CODE
QUALIFICATIONS_QUALIF_CODE
QUALIFICATIONS
QUALIF_CODE
QUALIF_NAME
QUALIF_DSCR
require
KoLICENSE_KoLIC_CODE
QUALIFICATIONS_QUALIF_CODE
KoLICENSE_KoLIC_CODE
KoAIRCRAFT_IATA
PASSENGERS
PSGR_CODE
LNAME
FNAME
DAY_PERIODS
YEAR_PERIODS_YEAR_PERIOD
MONTH_PERIODS_MONTH_PERIOD
DAY_PERIOD
DAY_NBR
ROLES
ROLE_CODE
ASSIGNMENTS
STAFF_STAFF_CODE
ASSIGNT_NBR
STARTING
ENDING
ROLES_ROLE_CODE
CREWS_CREW_CODE
TICKETS
WEEKDAYS_DAY_NBR
PLAN_FLIGHTS_FLIGHT_NBR
EFF_FLIGHTS_PLAN_DEP_DATE
SEAT
PRICE
PASSENGERS_PSGR_CODE
CURRENCIES_CURR_CODE
CLASSES_CLASS_CODE
CURRENCIES
CURR_CODE
CURR_NAME
CLASSES
CLASS_CODE
CLASS_NAME
name_role
ROLES_ROLE_CODE
LANGUAGES_LNG_CODE_in
ROLE_NAME
ROLE_DSCR
name_month
YEAR_PERIODS_YEAR_PERIOD
MONTH_PERIODS_MONTH
LANGUAGES_LNG_CODE_in
MONTHNAME
name_KoAircraft
LANGUAGES_LNG_CODE_in
KoAIRCRAFT_IATA
KoACRNAME
KoCREW
KoCREW_CODE
KoCREW_NAME
isa_KoCrew
KoCREW_KoCREW_CODE
CREWS_CREW_CODE
CREWS
CREW_CODE
CREW_NAME
estimate_price
WEEKDAYS_DAY_NBR
PLAN_FLIGHTS_FLIGHT_NBR
CLASSES_CLASS_CODE
ESTIMATED_PRICE
contain
CLASSES_CLASS_CODE
AIRCRAFTS_AIRCRAFT_NBR
SEATS
AIRPORTS
AIRPORT_CODE
CURRENCIES_CURR_CODE
accept
AIRPORTS_AIRPORT_CODE
KoAIRCRAFT_IATA
KoAIRCRAFT
IATA
AUTONOMY
SPEED
SEATS
MAX_LOAD
MAX_FUEL
AIRCRAFTS
AIRCRAFT_NBR
LAST_SERVICING
TOTALFLIGHTHOURS
KoAIRCRAFT_IATA
name_City
CITIES_CITY_CODE
LANGUAGES_LNG_CODE_in
CITYNAME
name_Airport
AIRPORTS_AIRPORT_CODE
LANGUAGES_LNG_CODE_in
ARP_NAME
LANGUAGES
LNG_CODE
name_language
LANGUAGES_LNG_CODE_in
LANGUAGES_LNG_CODE_of
LNGNAME
EFF_FLIGHTS
WEEKDAYS_DAY_NBR
PLAN_FLIGHTS_FLIGHT_NBR
PLAN_DEP_DATE
RES_SEATS_QTY
OCC_SEATS_QTY
REAL_DEP_TIME
REAL_ARR_TIME
LOADED_FUEL
CREWS_CREW_CODE
AIRCRAFTS_AIRCRAFT_NBR
ROUTES
ROUTE_CODE
ROUTE_DURATION
KoAIRCRAFT_IATA
AIRLINKS_AIRLINK_CODE
EFF_STOPOVERS
WEEKDAYS_DAY_NBR_1
PLAN_FLIGHTS_FLIGHT_NBR_1
EFF_FLIGHTS_PLAN_DEP_DATE
STO_NBR
REAL_TRAVEL_DUR
REAL_LANDING_TIME
REAL_TAKEOFF_TIME
LOADED_FUEL
WEEKDAYS_DAY_NBR
PLAN_FLIGHTS_FLIGHT_NBR
PLAN_STOPOVERS_STOPOVER_NBR
AIRLINKS
AIRLINK_CODE
AIRLINK_DISTANCE
AIRPORTS_AIRPORT_CODE_from
AIRPORTS_AIRPORT_CODE_to
PLAN_FLIGHTS
FLIGHT_NBR
DEPARTURE
FREQUENCY
AIRLINKS_AIRLINK_CODE
name_weekday
LANGUAGES_LNG_CODE_in
WEEKDAYS_DAY_NBR
DAYNAME
estimate_nb_passengers
WEEKDAYS_DAY_NBR
AIRLINKS_AIRLINK_CODE
TIMESLOTS_TIMESLOT_CODE
ESTIMATED_SEATS
TIMESLOTS
TIMESLOT_CODE
BEGINNING
ENDING
WEEKDAYS
DAY_NBR
PLAN_DAYFLIGHTS
WEEKDAYS_DAY_NBR
PLAN_FLIGHTS_FLIGHT_NBR
PLAN_SEATS
PLAN_DURATION
PLAN_ARR_TIME
KoAIRCRAFT_IATA
PLAN_STOPOVERS
WEEKDAYS_DAY_NBR
PLAN_FLIGHTS_FLIGHT_NBR
STOPOVER_NBR
STOPOVER_DURATION
ROUTES_ROUTE_CODE

Interprétation de l’environnement

Cet environnement est conçu pour gérer les opérations d'une compagnie aérienne, avec une attention particulière aux détails des vols, du personnel, des avions, et plus encore. Voici les points principaux :

Gestion du Personnel et des Rôles : Tables comme STAFF, ROLES, QUALIFICATIONS, et STATUS pour suivre les informations sur les membres du personnel, leurs rôles, qualifications, et leur statut actuel.

Opérations de Vol : Tables comme AIRCRAFTS, EFF_FLIGHTS, PLAN_FLIGHTS, et ROUTES fournissent des informations détaillées sur les avions, les vols planifiés et effectués, ainsi que les routes et durées de vol.

Gestion des Passagers et des Tickets : PASSENGERS et TICKETS pour gérer les informations sur les passagers et les détails de leurs billets, incluant la classe de voyage, le prix, et la monnaie.

Géographie et Langues : Tables comme CITIES, AIRPORTS, LANGUAGES, et diverses tables de noms (name_...) pour traiter les informations géographiques, les aéroports desservis, et pour soutenir le multilinguisme dans la base de données.

Planification et Statistiques : Tables comme PLAN_DAYFLIGHTS, ESTIMATE_PRICE, et ESTIMATE_NB_PASSENGERS pour la planification des vols quotidiens, l'estimation des prix des billets, et le nombre prévu de passagers, respectivement.

Maintenance et Qualifications : CHECKUPS pour les examens médicaux du personnel, obtain pour suivre quel personnel a obtenu quelles licences, et allow_pilot pour déterminer quelles licences permettent de piloter quels types d'avions.

Relations et Restrictions : Plusieurs tables de relations (include, may_have, need, etc.) définissent les contraintes et les exigences pour le personnel, les équipages, et les types d'avions pour certaines opérations ou rôles

Ceci vise à couvrir tous les aspects nécessaires à la gestion d’une compagnie aérienne.

Environnement d'exécution et gestion du projet

Pour répondre aux questions, nous avons installé un environnement PostgreSQL en local sur nos ordinateurs.

Nous avons collaboré via GitHub pour partager nos requêtes entre les membres du groupe.

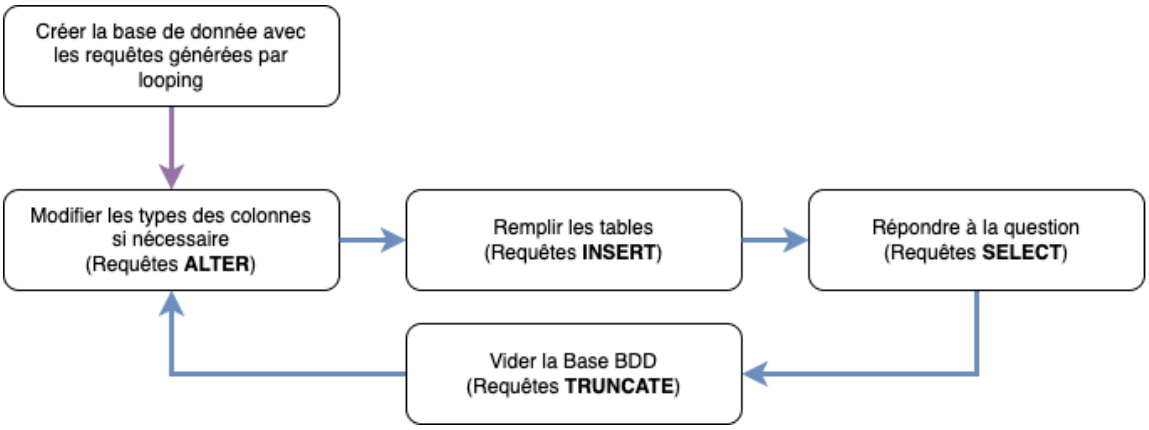
Certains membres du projet ont utilisé le logiciel DataGrip et d'autres pgAdmin pour réaliser et exécuter leurs requêtes

Pour chaque exercice, nous avions un fichier SQL avec les requêtes pour remplir les tables requises pour l'exercice.

Nous avons également réalisé un script pour vider entièrement la BDD pour que les données des exercices ne rentrent pas en conflits entre elles.

```
DO $$
DECLARE
   table_name text;
BEGIN
   FOR table_name IN (SELECT tablename FROM pg_tables WHERE schemaname = 'public')
   LOOP
      EXECUTE 'TRUNCATE TABLE ' || table_name || ' CASCADE';
   END LOOP;
END $$;
```

Ainsi nous avions le workflow suivant :



De cette manière, personne ne se gênait. Lorsque quelqu'un voulait tester un exercice, il avait juste à vider la BDD avec le script et exécuter l'exercice. Pas de conflits !

Réponses aux questions

1. Give a list of all staff members sorted by position descending and last name ascending.

```
INSERT INTO status (status_code, status_name, status_dscr)
VALUES (0, 'Hiring', 'In hiring status'),
       (1, 'Available', 'Available status'),
       (2, 'Not Available', 'Not Available status'),
       (3, 'Sick', 'Sick status'),
       (4, 'On boarding', 'On boarding status'),
       (5, 'On leave', 'On leave status'),
       (6, 'Retired', 'Retired status'),
       (7, 'Resigned', 'Resigned status');

INSERT INTO staff (staff_code, lname, fname, birth, member_position, flying,
tfh, status_status_code)
VALUES
    (0, 'Michael', 'Brown', TO_DATE('1990-01-01', 'YYYY-MM-DD'), 'CEO', false,
0, 1),
    (1, 'Emily', 'Clark', TO_DATE('1991-02-02', 'YYYY-MM-DD'), 'CFO', false,
0, 1),
    (2, 'David', 'Moore', TO_DATE('1992-03-03', 'YYYY-MM-DD'), 'CIO', false,
0, 1),
    (3, 'Olivia', 'Davis', TO_DATE('1993-04-04', 'YYYY-MM-DD'), 'COO', false,
0, 1),
    (4, 'William', 'Wilson', TO_DATE('1993-05-04', 'YYYY-MM-DD'), 'emp',
false, 0, 1),
    (5, 'Sophia', 'Taylor', TO_DATE('1993-06-04', 'YYYY-MM-DD'), 'emp', false,
0, 1),
    (6, 'James', 'Anderson', TO_DATE('1993-07-04', 'YYYY-MM-DD'), 'emp',
false, 0, 1),
    (7, 'Isabella', 'Martinez', TO_DATE('1999-08-04', 'YYYY-MM-DD'), 'emp',
false, 0, 2),
    (8, 'Benjamin', 'Hernandez', TO_DATE('1998-09-04', 'YYYY-MM-DD'), 'emp',
false, 0, 2),
    (9, 'Emma', 'Gonzalez', TO_DATE('1993-10-04', 'YYYY-MM-DD'), 'emp', false,
0, 2),
    (10, 'Alexander', 'Lopez', TO_DATE('1993-11-04', 'YYYY-MM-DD'), 'emp',
true, 0, 2),
    (11, 'Mia', 'Perez', TO_DATE('1993-12-04', 'YYYY-MM-DD'), 'emp', true, 0,
2),
    (12, 'Daniel', 'Torres', TO_DATE('1994-01-04', 'YYYY-MM-DD'), 'emp', true,
0, 2),
    (13, 'Charlotte', 'Ramirez', TO_DATE('1993-02-04', 'YYYY-MM-DD'), 'emp',
true, 0, 2),
    (14, 'Henry', 'Flores', TO_DATE('1995-03-04', 'YYYY-MM-DD'), 'emp', true,
0, 1),
    (15, 'Ava', 'Rivera', TO_DATE('1993-04-04', 'YYYY-MM-DD'), 'emp', true, 0,
1),
```

```
    (16, 'Matthew', 'Gomez', TO_DATE('1996-05-04', 'YYYY-MM-DD'), 'emp', true,
0, 6),
    (17, 'Ella', 'Sanchez', TO_DATE('1997-06-04', 'YYYY-MM-DD'), 'emp', true,
0, 1),
    (18, 'Luke', 'Mitchell', TO_DATE('1993-07-04', 'YYYY-MM-DD'), 'emp', true,
0, 1),
    (19, 'Lily', 'King', TO_DATE('1993-08-04', 'YYYY-MM-DD'), 'emp', true, 0,
1);

SELECT * FROM staff ORDER BY staff.member_position DESC, staff.lname;
```

Sortie:

	staff_code [PK] integer	lname character varying (50)	fname character varying (50)	birth date	member_position character varying (50)	flying boolean	tfh smallint	staff_staff_code_manager integer	status_status_code smallint
1	10	Alexander	Lopez	1993-11-04	emp	true	0	[null]	2
2	15	Ava	Rivera	1993-04-04	emp	true	0	[null]	1
3	8	Benjamin	Hernandez	1998-09-04	emp	false	0	[null]	2
4	13	Charlotte	Ramirez	1993-02-04	emp	true	0	[null]	2
5	12	Daniel	Torres	1994-01-04	emp	true	0	[null]	2
6	17	Ella	Sanchez	1997-06-04	emp	true	0	[null]	1
7	9	Emma	Gonzalez	1993-10-04	emp	false	0	[null]	2
8	14	Henry	Flores	1995-03-04	emp	true	0	[null]	1
9	7	Isabella	Martinez	1999-08-04	emp	false	0	[null]	2
10	6	James	Anderson	1993-07-04	emp	false	0	[null]	1
11	19	Lily	King	1993-08-04	emp	true	0	[null]	1
12	18	Luke	Mitchell	1993-07-04	emp	true	0	[null]	1
13	16	Matthew	Gomez	1996-05-04	emp	true	0	[null]	6
14	11	Mia	Perez	1993-12-04	emp	true	0	[null]	2
15	5	Sophia	Taylor	1993-06-04	emp	false	0	[null]	1
16	4	William	Wilson	1993-05-04	emp	false	0	[null]	1
17	3	Olivia	Davis	1993-04-04	COO	false	0	[null]	1
18	2	David	Moore	1992-03-03	CIO	false	0	[null]	1
19	1	Emily	Clark	1991-02-02	CFO	false	0	[null]	1
20	0	Michael	Brown	1990-01-01	CEO	false	0	[null]	1

2. For each staff member give total flight hours per month of last year (2023)

```
INSERT INTO public.month_periods (year_period, month_period)
VALUES
(2022, 01),
(2022, 02),
(2022, 03),
(2022, 04),
(2022, 05),
(2022, 06),
(2022, 07),
(2022, 08),
(2022, 09),
(2022, 10),
(2022, 11),
(2022, 12),
(2023, 01),
```

```
(2023, 02),
(2023, 03),
(2023, 04),
(2023, 05),
(2023, 06),
(2023, 07),
(2023, 08),
(2023, 09),
(2023, 10),
(2023, 11),
(2023, 12),
(2024, 01),
(2024, 02),
(2024, 03);

SELECT
    STAFF_STAFF_CODE AS STAFF_CODE,
    YEAR_PERIODS_YEAR_PERIOD AS YEAR,
    MONTH_PERIODS_MONTH_PERIOD AS MONTH,
    SUM(HOURS_NBR) AS HOURS
FROM in_flight
WHERE YEAR_PERIODS_YEAR_PERIOD = '2023'
GROUP BY
    STAFF_STAFF_CODE,
    YEAR_PERIODS_YEAR_PERIOD,
    MONTH_PERIODS_MONTH_PERIOD
ORDER BY
    STAFF_STAFF_CODE,
    MONTH_PERIODS_MONTH_PERIOD;
```

Sortie:

	staff_code integer	year character	month character	hours bigint
1	1	2023	1	120
2	1	2023	3	130
3	1	2023	6	160
4	2	2023	10	200
5	2	2023	5	150
6	2	2023	8	180
7	3	2023	2	110
8	3	2023	9	190
9	4	2023	7	170
10	5	2023	4	140

3. For each staff member give total flight hours per year

```
INSERT INTO public.in_flight (staff_staff_code, year_periods_year_period,
month_periods_month_period, hours_nbr)
VALUES
    (1, 2023, 01, 120),
    (3, 2023, 02, 110),
    (1, 2023, 03, 130),
    (5, 2023, 04, 140),
    (2, 2023, 05, 150),
    (1, 2023, 06, 160),
    (4, 2023, 07, 170),
    (2, 2023, 08, 180),
    (3, 2023, 09, 190),
    (2, 2023, 10, 200);
    (1, 2022, 01, 125),
    (11, 2022, 02, 115),
    (10, 2022, 03, 135),
    (14, 2022, 04, 145),
    (19, 2022, 05, 155),
    (4, 2022, 06, 165),
    (1, 2024, 01, 125),
    (11, 2024, 02, 115),
    (10, 2024, 01, 135),
    (14, 2024, 02, 145),
    (19, 2024, 03, 155),
    (4, 2024, 01, 165),
    (8, 2022, 07, 175);

SELECT
    STAFF_STAFF_CODE AS STAFF_CODE,
    YEAR_PERIODS_YEAR_PERIOD AS YEAR,
    SUM(HOURS_NBR) AS HOURS
FROM in_flight
GROUP BY
    STAFF_STAFF_CODE,
    YEAR_PERIODS_YEAR_PERIOD
ORDER BY
    STAFF_STAFF_CODE;
```

Sortie:

	staff_code integer	year character	hours bigint
1	1	2022	125
2	1	2024	125
3	1	2023	410
4	2	2023	530
5	3	2023	300
6	4	2024	165
7	4	2023	170
8	4	2022	165
9	5	2023	140
10	8	2022	175
11	10	2024	135
12	10	2022	135
13	11	2022	115
14	11	2024	115
15	14	2022	145
16	14	2024	145
17	19	2022	155
18	19	2024	155

4. List all staff members that manage persons and their managed persons.

```
INSERT INTO status (status_code, status_name, status_dscr)
VALUES (0, 'Hiring', 'In hiring status'),
       (1, 'Available', 'Available status'),
       (2, 'Not Available', 'Not Available status'),
       (3, 'Sick', 'Sick status'),
       (4, 'On boarding', 'On boarding status'),
       (5, 'On leave', 'On leave status'),
       (6, 'Retired', 'Retired status'),
       (7, 'Resigned', 'Resigned status');

INSERT INTO staff (staff_code, fname, lname, birth, member_position, flying,
tfh, status_status_code, staff_staff_code_manager)
VALUES
    (0, 'Michael', 'Brown', TO_DATE('1990-01-01', 'YYYY-MM-DD'), 'CEO', false,
0, 1, null),
    (1, 'Emily', 'Clark', TO_DATE('1991-02-02', 'YYYY-MM-DD'), 'CFO', false,
0, 1, null),
    (2, 'David', 'Moore', TO_DATE('1992-03-03', 'YYYY-MM-DD'), 'CIO', false,
0, 1, null),
    (3, 'Olivia', 'Davis', TO_DATE('1993-04-04', 'YYYY-MM-DD'), 'COO', false,
0, 1, null),
    (4, 'William', 'Wilson', TO_DATE('1993-05-04', 'YYYY-MM-DD'), 'emp',
false, 0, 1, 3),
    (5, 'Sophia', 'Taylor', TO_DATE('1993-06-04', 'YYYY-MM-DD'), 'emp', false,
0, 1, 3),
    (6, 'James', 'Anderson', TO_DATE('1993-07-04', 'YYYY-MM-DD'), 'emp',
false, 0, 1, 3),
    (7, 'Isabella', 'Martinez', TO_DATE('1999-08-04', 'YYYY-MM-DD'), 'emp',
false, 0, 2, 3),
    (8, 'Benjamin', 'Hernandez', TO_DATE('1998-09-04', 'YYYY-MM-DD'), 'emp',
false, 0, 2, 3),
    (9, 'Emma', 'Gonzalez', TO_DATE('1993-10-04', 'YYYY-MM-DD'), 'emp', false,
0, 2, 2),
    (10, 'Alexander', 'Lopez', TO_DATE('1993-11-04', 'YYYY-MM-DD'), 'emp',
true, 0, 2, 2),
    (11, 'Mia', 'Perez', TO_DATE('1993-12-04', 'YYYY-MM-DD'), 'emp', true, 0,
2, 2),
    (12, 'Daniel', 'Torres', TO_DATE('1994-01-04', 'YYYY-MM-DD'), 'emp', true,
0, 2, 2),
    (13, 'Charlotte', 'Ramirez', TO_DATE('1993-02-04', 'YYYY-MM-DD'), 'emp',
true, 0, 2, 2),
    (14, 'Henry', 'Flores', TO_DATE('1995-03-04', 'YYYY-MM-DD'), 'emp', true,
0, 1, 1),
    (15, 'Ava', 'Rivera', TO_DATE('1993-04-04', 'YYYY-MM-DD'), 'emp', true, 0,
1, 1),
    (16, 'Matthew', 'Gomez', TO_DATE('1996-05-04', 'YYYY-MM-DD'), 'emp', true,
0, 6, 1),
```

```
    (17, 'Ella', 'Sanchez', TO_DATE('1997-06-04', 'YYYY-MM-DD'), 'emp', true,
0, 1, 1),
    (18, 'Luke', 'Mitchell', TO_DATE('1993-07-04', 'YYYY-MM-DD'), 'emp', true,
0, 1, 1),
    (19, 'Lily', 'King', TO_DATE('1993-08-04', 'YYYY-MM-DD'), 'emp', true, 0,
1, 1);



SELECT staff.fname, staff.lname, manager.fname AS "manager firstname",
manager.lname AS "manager_lastname"
FROM staff
         INNER JOIN staff as manager ON staff.staff_staff_code_manager =
manager.staff_code;
```

Sortie:

	fname	lname	"manager firstname"	manager_lastname
1	William	Wilson	Olivia	Davis
2	Sophia	Taylor	Olivia	Davis
3	James	Anderson	Olivia	Davis
4	Isabella	Martinez	Olivia	Davis
5	Benjamin	Hernandez	Olivia	Davis
6	Emma	Gonzalez	David	Moore
7	Alexander	Lopez	David	Moore
8	Mia	Perez	David	Moore
9	Daniel	Torres	David	Moore
10	Charlotte	Ramirez	David	Moore
11	Henry	Flores	Emily	Clark
12	Ava	Rivera	Emily	Clark
13	Matthew	Gomez	Emily	Clark
14	Ella	Sanchez	Emily	Clark
15	Luke	Mitchell	Emily	Clark
16	Lily	King	Emily	Clark

5. List all staff members who actually have been a pilot last year (2023)

```
INSERT INTO QUALIFICATIONS (qualif_code, qualif_name, qualif_dscr)
VALUES ('1', 'Pilot', 'Pilote'),
       ('2', 'Flight Attendant', 'Hotesse de lair'),
       ('3', 'Mechanic', 'mecano');

INSERT INTO have_qualif (staff_staff_code, qualifications_qualif_code)
VALUES ('0', '1'),  -- pilote
       ('1', '1'),
```

```
      ('2', '1'),
      ('3', '1'),
      ('4', '1'),
      ('5', '2'), -- hotesse de l'air
      ('6', '2'),
      ('7', '2'),
      ('8', '2'),
      ('9', '2'),
      ('10', '3'), -- mecano
      ('11', '3'),
      ('12', '3'),
      ('13', '3'),
      ('14', '3'),
      ('15', '3'),
      ('16', '3'),
      ('17', '3'),
      ('18', '3'),
      ('19', '3');

INSERT INTO KoCREW (kocrew_code, kocrew_name)
VALUES ('1', 'Pilot Crew'),
       ('2', 'Cabin Crew'),
       ('3', 'Maintenance Crew');


INSERT INTO roles (role_code)
VALUES ('1'),('2'),('3');

INSERT INTO CREWS (CREW_CODE, CREW_NAME)
VALUES
    ('1', 'Alpha'),
    ('2', 'Bravo'),
    ('3', 'Charlie');



INSERT INTO ASSIGNMENTS (staff_staff_code, crews_crew_code, starting,
roles_role_code, assignt_nbr)
VALUES  ('1', '1', '2023-01-01', '1', 1),
        ('2', '2', '2022-07-15', '1', 2),
        ('5', '1', '2023-01-01', '2', 3),
        ('6', '2', '2022-07-15', '2', 4),
        ('10', '1', '2023-01-01', '3', 5),
        ('11', '2', '2022-07-15', '3', 6);

truncate table assignments cascade;

SELECT staff.fname, staff.lname
FROM staff
```

```
INNER JOIN have_qualif ON staff.staff_code = have_qualif.staff_staff_code
INNER JOIN qualifications ON have_qualif.qualifications_qualif_code =
qualifications.qualif_code
INNER JOIN ASSIGNMENTS ON staff.staff_code = ASSIGNMENTS.staff_staff_code
WHERE qualifications.qualif_name = 'Pilot'  -- Filter for Pilot qualification
  AND EXTRACT(YEAR FROM ASSIGNMENTS.starting) = 2023; -- Filter for
assignments in 2023
```

Sortie:

	fname	lname
1	Clark	Emily

6. List all staff members without assignments this year (2024)

```
SELECT staff.staff_code, staff.lname, staff.fname
FROM staff
LEFT JOIN ASSIGNMENTS ON staff.staff_code = ASSIGNMENTS.staff_staff_code
WHERE ASSIGNMENTS.starting IS NULL OR EXTRACT(YEAR FROM ASSIGNMENTS.starting)
<> 2023;
```

Sortie:

	staff_code	lname	fname
1	2	David	Moore
2	6	James	Anderson
3	11	Mia	Perez
4	17	Ella	Sanchez
5	0	Michael	Brown
6	12	Daniel	Torres
7	18	Luke	Mitchell
8	15	Ava	Rivera
9	13	Charlotte	Ramirez
10	19	Lily	King
11	8	Benjamin	Hernandez
12	16	Matthew	Gomez
13	4	William	Wilson
14	3	Olivia	Davis
15	14	Henry	Flores
16	9	Emma	Gonzalez
17	7	Isabella	Martinez

7. What are the flights performed without stopovers

```
-- Exercice 7 --

INSERT INTO CURRENCIES(CURR_CODE, CURR_NAME)
VALUES ('EUR', 'Euro'),
       ('USD', 'US Dollar'),
       ('AUD', 'Australian Dollar'),
       ('JPY', 'Japanese Yen');

INSERT INTO AIRPORTS(AIRPORT_CODE, CURRENCIES_CURR_CODE)
VALUES ('CDG', 'EUR'),
       ('JFK', 'USD'),
       ('LAX', 'USD'),
       ('SYD', 'AUD'),
       ('HND', 'JPY');

INSERT INTO KoAIRCRAFT(IATA, AUTONOMY, SPEED, SEATS, MAX_LOAD, MAX_FUEL)
VALUES ('A320', 6000, 840, 180, 78000, 24210),
       ('B737', 5600, 853, 178, 70533, 20820),
       ('A380', 15200, 1020, 853, 577000, 323546),
       ('B747', 14815, 988, 660, 442000, 216840);

INSERT INTO AIRLINKS(AIRLINK_CODE, AIRLINK_DISTANCE,
AIRPORTS_AIRPORT_CODE_from, AIRPORTS_AIRPORT_CODE_to)
VALUES (1, '5000', 'CDG', 'JFK'),
       (2, '7000', 'CDG', 'LAX'),
       (3, '15000', 'CDG', 'SYD'),
       (4, '8000', 'CDG', 'HND');

INSERT INTO ROUTES(ROUTE_CODE, ROUTE_DURATION, KoAIRCRAFT_IATA,
AIRLINKS_AIRLINK_CODE)
VALUES (1, '06:00:00', 'A320', 1),
       (2, '08:20:00', 'B737', 2),
       (3, '17:00:00', 'A380', 3),
       (4, '09:30:00', 'B747', 4);

INSERT INTO public.plan_flights (flight_nbr, departure, frequency,
airlinks_airlink_code)
VALUES (1, '08:00:00', 7, 1),
       (2, '12:00:00', 5, 2),
       (3, '16:00:00', 3, 3),
       (4, '20:00:00', 4, 4),
       (5, '10:00:00', 6, 1),
       (6, '14:00:00', 7, 2),
       (7, '18:00:00', 5, 3),
       (8, '22:00:00', 3, 4),
       (9, '09:00:00', 4, 1),
```

```
       (10, '13:00:00', 7, 2);
INSERT INTO public.weekdays (day_nbr)
VALUES (1),
       (2),
       (3),
       (4),
       (5),
       (6),
       (7);



INSERT INTO public.plan_dayflights (weekdays_day_nbr, plan_flights_flight_nbr,
plan_seats, plan_duration, plan_arr_time,
                                    koaircraft_iata)
VALUES (7, 1, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (5, 2, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (1, 3, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (4, 4, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (5, 1, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (3, 1, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (6, 1, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (1, 1, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320'),
       (2, 2, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '02:00:00', '00:00:00',
'B747'),
       (6, 2, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '02:00:00', '00:00:00',
'B747'),
       (7, 2, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '02:00:00', '00:00:00',
'B747'),
       (4, 3, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '04:00:00', '06:00:00',
'A380'),
       (5, 3, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '05:00:00', '07:00:00',
'B737'),
       (6, 4, FLOOR(RANDOM() * (200 - 1 + 1) + 1), '06:00:00', '08:00:00',
'A320');



INSERT INTO public.plan_stopovers (weekdays_day_nbr, plan_flights_flight_nbr,
stopover_nbr, stopover_duration,
                                   routes_route_code)
```

```
VALUES (7, 1, FLOOR(RANDOM() * (10 - 1 + 1) + 1), '01:00:00', 1),
       (5, 2, FLOOR(RANDOM() * (10 - 1 + 1) + 1), '02:00:00', 2),
       (1, 3, FLOOR(RANDOM() * (10 - 1 + 1) + 1), '03:00:00', 3),
       (4, 4, FLOOR(RANDOM() * (10 - 1 + 1) + 1), '04:00:00', 4);




SELECT pd.*
FROM public.plan_dayflights pd
         LEFT JOIN public.plan_stopovers ps
                   ON pd.weekdays_day_nbr = ps.weekdays_day_nbr AND
                      pd.plan_flights_flight_nbr = ps.plan_flights_flight_nbr
WHERE ps.plan_flights_flight_nbr IS NULL;
```

Sortie:

	weekdays_day_nbr [PK] smallint	plan_flights_flight_nbr [PK] smallint	plan_seats smallint	plan_duration time without time zone	plan_arr_time time without time zone	koaircraft_iata character
1	5	1	175	06:00:00	08:00:00	A320
2	3	1	104	06:00:00	08:00:00	A320
3	6	1	153	06:00:00	08:00:00	A320
4	1	1	165	06:00:00	08:00:00	A320
5	2	2	86	02:00:00	00:00:00	B747
6	6	2	121	02:00:00	00:00:00	B747
7	7	2	196	02:00:00	00:00:00	B747

8. What are the aircrafts not able to perform the longest air link without stopovers

```
INSERT INTO CURRENCIES(CURR_CODE, CURR_NAME) VALUES
('EUR', 'Euro'),
('USD', 'US Dollar'),
('AUD', 'Australian Dollar'),
('JPY', 'Japanese Yen');

INSERT INTO AIRPORTS(AIRPORT_CODE, CURRENCIES_CURR_CODE) VALUES
('CDG', 'EUR'),
('JFK', 'USD'),
('LAX', 'USD'),
('SYD', 'AUD'),
('HND', 'JPY');


INSERT INTO KoAIRCRAFT(IATA, AUTONOMY, SPEED, SEATS, MAX_LOAD, MAX_FUEL)
VALUES
('A320', 6000, 840, 180, 78000, 24210),
('B737', 5600, 853, 178, 70533, 20820),
('A380', 15200, 1020, 853, 577000, 323546),
```

```
('B747', 14815, 988, 660, 442000, 216840);


ALTER TABLE AIRLINKS
ALTER COLUMN airlink_distance TYPE INT USING airlink_distance::INT;

INSERT INTO AIRLINKS(AIRLINK_CODE, AIRLINK_DISTANCE,
AIRPORTS_AIRPORT_CODE_from, AIRPORTS_AIRPORT_CODE_to) VALUES
(1, 5000, 'CDG', 'JFK'),
(2, 7000, 'CDG', 'LAX'),
(3, 15000, 'CDG', 'SYD'),
(4, 8000, 'CDG', 'HND');

INSERT INTO ROUTES(ROUTE_CODE, ROUTE_DURATION, KoAIRCRAFT_IATA,
AIRLINKS_AIRLINK_CODE) VALUES
(1, '06:00:00', 'A320', 1),
(2, '08:20:00', 'B737', 2),
(3, '17:00:00', 'A380', 3),
(4, '09:30:00', 'B747', 4);


SELECT koaircraft.iata AS "Aircraft", koaircraft.autonomy AS "Autonomy",
MAX(airlinks.airlink_distance) AS "Max air connection distance"
FROM koaircraft, airlinks
WHERE koaircraft.autonomy < airlinks.airlink_distance
GROUP BY koaircraft.iata, koaircraft.autonomy
```

Sortie:

	Aircraft character	Autonomy integer	Max air connection distance integer
1	B737	5600	15000
2	B747	14815	15000
3	A320	6000	15000

9. List of all staff members who worked during a weekend (Saturday or Sunday)

```
INSERT INTO public.languages(lng_code)
VALUES
    ('es'), -- Espagnol
    ('de'), -- Allemand
    ('fr'), -- Français
    ('it'), -- Italien
    ('ru'), -- Russe
    ('zh'), -- Chinois
```

```
    ('ja'), -- Japonais
    ('ar'); -- Arabe
    ('en'); -- Englais


ALTER TABLE public.languages RENAME COLUMN lng_code TO lng_code_in;

INSERT INTO public.name_weekday(languages_lng_code_in, weekdays_day_nbr,
dayname)
VALUES
    ('en', 1, 'Monday'), ('fr', 1, 'Lundi'), ('es', 1, 'Lunes'), ('de', 1,
'Montag'),
    ('en', 2, 'Tuesday'), ('fr', 2, 'Mardi'), ('es', 2, 'Martes'), ('de', 2,
'Dienstag'),
    ('en', 3, 'Wednesday'), ('fr', 3, 'Mercredi'), ('es', 3, 'Miércoles'),
('de', 3, 'Mittwoch'),
    ('en', 4, 'Thursday'), ('fr', 4, 'Jeudi'), ('es', 4, 'Jueves'), ('de', 4,
'Donnerstag'),
    ('en', 5, 'Friday'), ('fr', 5, 'Vendredi'), ('es', 5, 'Viernes'), ('de',
5, 'Freitag'),
    ('en', 6, 'Saturday'), ('fr', 6, 'Samedi'), ('es', 6, 'Sábado'), ('de', 6,
'Samstag'),
    ('en', 7, 'Sunday'), ('fr', 7, 'Dimanche'), ('es', 7, 'Domingo'), ('de',
7, 'Sonntag');


INSERT INTO ASSIGNMENTS (staff_staff_code, crews_crew_code, starting, ending,
roles_role_code, assignt_nbr)
VALUES
(0, 3, '01/02/2024', '01/02/2024', 1, 7),
(3, 2, '01/03/2024', '01/03/2024', 2, 8),
(7, 3, '01/04/2024', '01/04/2024', 3, 9),
(8, 1, '01/05/2024', '01/05/2024', 2, 10),
(12, 3, '01/06/2023', '01/06/2023', 3, 11),
(13, 2, '01/07/2023', '01/07/2023', 2, 12),
(14, 2, '01/08/2022', '01/08/2022', 3, 13),
(15, 1, '01/09/2022', '01/09/2022', 3, 14),
(16, 3, '01/10/2022', '01/10/2022', 3, 15),
(17, 2, '01/11/2022', '01/11/2022', 2, 16);

SELECT s.lname, s.fname
FROM staff s
JOIN assignments a ON s.staff_code = a.staff_staff_code
WHERE (EXTRACT(DOW FROM a.starting) IN (6, 0) OR EXTRACT(DOW FROM a.ending) IN
(6, 0));
```

Sortie:

	lname character varying (50)	fname character varying (50)
1	Emily	Clark
2	Sophia	Taylor
3	Alexander	Lopez
4	Charlotte	Ramirez
5	Matthew	Gomez

10. Give a list of all flights having an actual number of seats occupied different from expected number of seat occupied

```
-- Exercice 10 --

INSERT INTO CURRENCIES(CURR_CODE, CURR_NAME)
VALUES ('EUR', 'Euro'),
       ('USD', 'US Dollar'),
       ('AUD', 'Australian Dollar'),
       ('JPY', 'Japanese Yen');

INSERT INTO AIRPORTS(AIRPORT_CODE, CURRENCIES_CURR_CODE)
VALUES ('CDG', 'EUR'),
       ('JFK', 'USD'),
       ('LAX', 'USD'),
       ('SYD', 'AUD'),
       ('HND', 'JPY');

INSERT INTO airlinks(AIRLINK_CODE, AIRLINK_DISTANCE,
AIRPORTS_AIRPORT_CODE_from, AIRPORTS_AIRPORT_CODE_to)
VALUES (1, '5000', 'CDG', 'JFK'),
       (2, '7000', 'CDG', 'LAX'),
       (3, '15000', 'CDG', 'SYD'),
       (4, '8000', 'CDG', 'HND');

INSERT INTO plan_flights(flight_nbr, departure, frequency,
airlinks_airlink_code)
VALUES (1, '08:00:00', 7, 1),
       (2, '10:00:00', 7, 2),
       (3, '12:00:00', 7, 3);
```

```
INSERT INTO weekdays (day_nbr)
VALUES (1),
       (2),
       (3),
       (4),
       (5),
       (6),
       (7);


INSERT INTO koaircraft(iata, autonomy, speed, seats, max_load, max_fuel)
VALUES ('A320', 3500, 840, 150, 15000, 24000),
       ('A330', 3500, 840, 150, 15000, 24000),
       ('A340', 3500, 840, 150, 15000, 24000);

INSERT INTO aircrafts(aircraft_nbr, last_servicing, totalflighthours,
koaircraft_iata)
VALUES (1, '2022-01-01', 1000, 'A320'),
       (2, '2022-01-01', 1000, 'A330'),
       (3, '2022-01-01', 1000, 'A340');



INSERT INTO plan_dayflights(weekdays_day_nbr, plan_flights_flight_nbr,
plan_seats, plan_duration, plan_arr_time,
                            koaircraft_iata)
VALUES (1, 1, 100, '02:00:00', '10:00:00', 'A320'),
       (2, 2, 200, '02:00:00', '12:00:00', 'A330'),
       (3, 3, 300, '02:00:00', '14:00:00', 'A340');


INSERT INTO crews(crew_code, crew_name)
VALUES (1, 'Crew 1'),
       (2, 'Crew 2'),
       (3, 'Crew 3');


INSERT INTO eff_flights(weekdays_day_nbr, plan_flights_flight_nbr,
plan_dep_date, res_seats_qty, occ_seats_qty,
                        real_dep_time, real_arr_time, loaded_fuel,
crews_crew_code, aircrafts_aircraft_nbr)
VALUES (1, 1, '08:00:00', 80, 70, '2022-01-01 08:00:00', '2022-01-01
10:00:00', 5000, 1, 1),
       (2, 2, '10:00:00', 210, 200, '2022-01-02 10:00:00', '2022-01-02
12:00:00', 6000, 2, 2),
       (3, 3, '12:00:00', 280, 250, '2022-01-03 12:00:00', '2022-01-03
14:00:00', 7000, 3, 3);
```

```
SELECT DISTINCT eff_flights.plan_flights_flight_nbr AS "Flight number",
                plan_dayflights.plan_seats          AS "Planned seats",
                eff_flights.occ_seats_qty           AS "Occupied seats"
FROM eff_flights
         INNER JOIN plan_flights ON plan_flights.flight_nbr =
eff_flights.plan_flights_flight_nbr
         INNER JOIN plan_dayflights ON plan_flights.flight_nbr =
eff_flights.plan_flights_flight_nbr AND
                                       plan_dayflights.weekdays_day_nbr =
eff_flights.weekdays_day_nbr
WHERE plan_dayflights.plan_seats <> eff_flights.occ_seats_qty
```

Sortie:

	Flight number smallint	Planned seats smallint	Occupied seats smallint
1	1	104	280
2	2	86	145
3	2	175	170
4	2	181	170
5	1	165	170
6	4	85	340
7	1	64	170
8	4	79	340
9	1	121	130
10	2	153	110
11	4	17	340
12	2	121	110
13	1	153	130
14	4	121	340
15	1	2	280

11. List of all flight that have been done a weekend (Saturday or Sunday)

```
INSERT INTO aircrafts(aircraft_nbr, last_servicing, totalflighthours,
koaircraft_iata)
VALUES
(1, '01/01/2024', 5000, 'A320'),
(2, '01/02/2024', 5100, 'B737'),
(3, '01/03/2024', 5200, 'A380'),
(4, '01/04/2024', 5300, 'B747'),
(5, '01/05/2024', 5400, 'A320'),
(6, '01/06/2024', 5500, 'B737'),
(7, '01/07/2024', 5600, 'A380'),
(8, '01/08/2024', 5700, 'B747'),
(9, '01/09/2024', 5800, 'A320'),
(10, '01/10/2024', 5900, 'B737'),
(11, '01/11/2024', 6000, 'A380'),
(12, '01/12/2024', 6100, 'B747');

INSERT INTO EFF_FLIGHTS (WEEKDAYS_DAY_NBR, PLAN_FLIGHTS_FLIGHT_NBR,
PLAN_DEP_DATE, RES_SEATS_QTY, OCC_SEATS_QTY, REAL_DEP_TIME, REAL_ARR_TIME,
LOADED_FUEL, CREWS_CREW_CODE, AIRCRAFTS_AIRCRAFT_NBR)
VALUES
  (1, 1, '2024-03-25 10:00:00', 180, 170, '2024-03-25 10:15:00', '2024-03-25
11:30:00', 25000, 1, 1),
  (2, 2, '2024-03-26 15:00:00', 160, 145, '2024-03-26 15:10:00', '2024-03-26
16:45:00', 20000, 1, 2),
  (3, 1, '2024-03-27 08:00:00', 290, 280, '2024-03-27 08:15:00', '2024-03-27
12:15:00', 45000, 2, 3),
  (4, 4, '2024-03-28 12:00:00', 360, 340, '2024-03-28 12:15:00', '2024-03-28
15:30:00', 50000, 3, 4),
  (5, 2, '2024-03-29 10:00:00', 180, 170, '2024-03-29 10:15:00', '2024-03-29
11:30:00', 25000, 1, 1),
  (6, 1, '2024-03-30 10:00:00', 140, 130, '2024-03-30 10:15:00', '2024-03-30
11:00:00', 20000, 1, 1),
  (6, 2, '2024-03-30 13:00:00', 120, 110, '2024-03-30 13:15:00', '2024-03-30
15:00:00', 18000, 2, 2);

SELECT ef.*
FROM EFF_FLIGHTS ef
INNER JOIN PLAN_DAYFLIGHTS pd ON ef.WEEKDAYS_DAY_NBR = pd.WEEKDAYS_DAY_NBR
                             AND ef.PLAN_FLIGHTS_FLIGHT_NBR =
pd.PLAN_FLIGHTS_FLIGHT_NBR
WHERE pd.WEEKDAYS_DAY_NBR IN (6, 7);
```

Sortie:

	weekdays_day_nbr [PK] smallint	plan_flights_flight_nbr [PK] smallint	plan_dep_date [PK] time without time zone	res_seats_qty smallint	occ_seats_qty smallint	real_dep_time timestamp without time zone	real_arr_time timestamp without time zone	loaded_fuel integer
1	6	1	10:00:00	140	130	2024-03-30 10:15:00	2024-03-30 11:00:00	20000
2	6	2	13:00:00	120	110	2024-03-30 13:15:00	2024-03-30 15:00:00	18000

12. For each flight, what is the total amount perceived for tickets compared to the expected one.

```
INSERT INTO CURRENCIES(CURR_CODE, CURR_NAME)
VALUES ('EUR', 'Euro'),
       ('USD', 'US Dollar'),
       ('AUD', 'Australian Dollar'),
       ('JPY', 'Japanese Yen');

INSERT INTO AIRPORTS(AIRPORT_CODE, CURRENCIES_CURR_CODE)
VALUES ('CDG', 'EUR'),
       ('JFK', 'USD'),
       ('LAX', 'USD'),
       ('SYD', 'AUD'),
       ('HND', 'JPY');


-- Insertion dans la table `weekdays`
INSERT INTO weekdays (day_nbr)
VALUES (1),
       (2),
       (3),
       (4),
       (5),
       (6),
       (7);

ALTER TABLE AIRLINKS
    ALTER COLUMN airlink_distance TYPE INT USING airlink_distance::INT;

INSERT INTO AIRLINKS(AIRLINK_CODE, AIRLINK_DISTANCE,
AIRPORTS_AIRPORT_CODE_from, AIRPORTS_AIRPORT_CODE_to)
VALUES (1, 5000, 'CDG', 'JFK'),
       (2, 7000, 'CDG', 'LAX'),
       (3, 15000, 'CDG', 'SYD'),
       (4, 8000, 'CDG', 'HND');

INSERT INTO KoAIRCRAFT(IATA, AUTONOMY, SPEED, SEATS, MAX_LOAD, MAX_FUEL)
VALUES ('A320', 6000, 840, 180, 78000, 24210),
       ('B737', 5600, 853, 178, 70533, 20820),
       ('A380', 15200, 1020, 853, 577000, 323546),
```

```
       ('B747', 14815, 988, 660, 442000, 216840);



-- Insertion dans la table `classes`
INSERT INTO classes (class_code, class_name)
VALUES (1, 'Economy'),
       (2, 'Business'),
       (3, 'First');


ALTER TABLE passengers
    ALTER COLUMN lname TYPE varchar USING lname::varchar;

-- Insertion dans la table `passengers`
INSERT INTO passengers (psgr_code, fname, lname)
VALUES (1, 'John', 'Doe'),
       (2, 'Jane', 'Doe'),
       (3, 'Louis', 'Lecout'),
       (4, 'Guy', 'Liguili');

-- Insertion dans la table `plan_flights`
INSERT INTO plan_flights (flight_nbr, departure, frequency,
airlinks_airlink_code)
VALUES (1, '06:00:00', 7, 1),
       (2, '06:00:00', 7, 1);


-- Insertion dans la table `plan_dayflights`
INSERT INTO plan_dayflights (weekdays_day_nbr, plan_flights_flight_nbr,
plan_seats, plan_duration, plan_arr_time,
                            koaircraft_iata)
VALUES (1, 1, 100, '02:00:00', '08:00:00', 'A320'),
       (1, 2, 100, '02:00:00', '08:00:00', 'A320'),
       (2, 2, 100, '02:00:00', '08:00:00', 'A320');


-- Insertion into the `crews` table
INSERT INTO crews (crew_code, crew_name)
VALUES (1, 'Crew 1');

-- Insertion into the `aircrafts` table
INSERT INTO aircrafts (aircraft_nbr, last_servicing, totalflighthours,
koaircraft_iata)
VALUES (1, '2023-01-01', 1000, 'A320');

-- Insertion into the `eff_flights` table
INSERT INTO eff_flights (weekdays_day_nbr, plan_flights_flight_nbr,
plan_dep_date, res_seats_qty, occ_seats_qty,
```

```
                        real_dep_time, real_arr_time, loaded_fuel,
crews_crew_code, aircrafts_aircraft_nbr)
VALUES (1, 1, '2023-03-01 06:00:00', 100, 100, '2023-03-01 06:00:00', '2023-
03-01 08:00:00', 10000, 1, 1),
       (1, 2, '2023-03-01 06:00:00', 100, 100, '2023-03-01 06:00:00', '2023-
03-01 08:00:00', 10000, 1, 1);

-- Now you can insert into the `tickets` table
INSERT INTO tickets (weekdays_day_nbr, plan_flights_flight_nbr,
eff_flights_plan_dep_date, seat, price,
                     passengers_psgr_code, currencies_curr_code,
classes_class_code)
VALUES (1, 1, '2023-03-01 06:00:00', 'A1', 100.00, 1, 'USD', 1),
       (1, 1, '2023-03-01 06:00:00', 'A2', 200.00, 2, 'USD', 2),
       (1, 2, '2023-03-01 06:00:00', 'A1', 200.00, 3, 'USD', 1),
       (1, 2, '2023-03-01 06:00:00', 'A2', 200.00, 2, 'USD', 1),
       (1, 1, '2023-03-01 06:00:00', 'A3', 400.00, 3, 'USD', 3);


SELECT
    estimated_revenue."Flight Number",
    estimated_revenue."Estimated Total Revenue",
    actual_revenue."Actual Total Revenue"
FROM
    (
        SELECT
            plan_dayflights.plan_flights_flight_nbr AS "Flight Number",
            SUM(plan_dayflights.plan_seats * estimate_price.estimated_price)
AS "Estimated Total Revenue"
        FROM
            plan_dayflights
        JOIN
            estimate_price ON plan_dayflights.weekdays_day_nbr =
estimate_price.weekdays_day_nbr
                           AND plan_dayflights.plan_flights_flight_nbr =
estimate_price.plan_flights_flight_nbr
        GROUP BY
            plan_dayflights.plan_flights_flight_nbr
    ) AS estimated_revenue
LEFT JOIN
    (
        SELECT tickets.plan_flights_flight_nbr, SUM(tickets.price) AS "Actual
Total Revenue"
        FROM tickets
        GROUP BY tickets.plan_flights_flight_nbr
    ) AS actual_revenue ON estimated_revenue."Flight Number" =
actual_revenue.plan_flights_flight_nbr
ORDER BY
    estimated_revenue."Flight Number";
```

Sortie:

	Flight Number smallint	Estimated Total Revenue money	Actual Total Revenue money
1	1	16 500,00 €	700,00 €
2	2	12 900,00 €	400,00 €

13. What is the five most frequently used aircrafts

```
INSERT INTO KoAIRCRAFT (IATA, AUTONOMY, SPEED, SEATS, MAX_LOAD, MAX_FUEL)
VALUES
  ('A320', 4400, 840, 180, 70000, 30000),
  ('B737', 5000, 800, 160, 60000, 27000),
  ('A330', 11000, 870, 290, 230000, 110000),
  ('B777', 14600, 905, 360, 230000, 170000);

INSERT INTO AIRCRAFTS (AIRCRAFT_NBR, LAST_SERVICING, TOTALFLIGHTHOURS,
KoAIRCRAFT_IATA)
VALUES
  (1, '2023-12-25', 2500, 'A320'),
  (2, '2024-01-10', 3100, 'B737'),
  (3, '2023-11-18', 7800, 'A330'),
  (4, '2024-02-15', 5200, 'B777');


INSERT INTO EFF_FLIGHTS (WEEKDAYS_DAY_NBR, PLAN_FLIGHTS_FLIGHT_NBR,
PLAN_DEP_DATE, RES_SEATS_QTY, OCC_SEATS_QTY, REAL_DEP_TIME, REAL_ARR_TIME,
LOADED_FUEL, CREWS_CREW_CODE, AIRCRAFTS_AIRCRAFT_NBR)
VALUES
  (1, 101, '2024-03-25 10:00:00', 180, 170, '2024-03-25 10:15:00', '2024-03-25
11:30:00', 25000, 1, 1),
  (2, 102, '2024-03-26 15:00:00', 160, 145, '2024-03-26 15:10:00', '2024-03-26
16:45:00', 20000, 1, 2),
  (3, 103, '2024-03-27 08:00:00', 290, 280, '2024-03-27 08:15:00', '2024-03-27
12:15:00', 45000, 2, 3),
  (4, 104, '2024-03-28 12:00:00', 360, 340, '2024-03-28 12:15:00', '2024-03-28
15:30:00', 50000, 3, 4),
  (5, 105, '2024-03-29 10:00:00', 180, 170, '2024-03-29 10:15:00', '2024-03-29
11:30:00', 25000, 1, 1),
  (6, 101, '2024-03-30 10:00:00', 140, 130, '2024-03-30 10:15:00', '2024-03-30
11:00:00', 20000, 1, 1),
  (6, 102, '2024-03-30 13:00:00', 120, 110, '2024-03-30 13:15:00', '2024-03-30
15:00:00', 18000, 2, 2);
```

```sql
INSERT INTO CURRENCIES (CURR_CODE, CURR_NAME)
VALUES
  ('USD', 'US Dollar'),
  ('EUR', 'Euro'),
  ('GBP', 'British Pound'),
  ('AUD', 'Australian Dollar'),
  ('JPY', 'Japanese Yen'),
  ('CNY', 'Chinese Yuan Renminbi'),
  ('CHF', 'Swiss Franc'),
  ('SGD', 'Singapore Dollar'),
  ('HKD', 'Hong Kong Dollar'),
  ('AED', 'United Arab Emirates Dirham'),
  ('QAR', 'Qatari Rial'),
  ('BRL', 'Brazilian Real'),
  ('CAD', 'Canadian Dollar'),
  ('MXN', 'Mexican Peso'),
  ('INR', 'Indian Rupee');


INSERT INTO AIRPORTS (AIRPORT_CODE, CURRENCIES_CURR_CODE)
VALUES
  ('CDG', 'EUR'), -- Paris, France (using Euro)
  ('LHR', 'GBP'), -- London, United Kingdom (using British Pound)
  ('JFK', 'USD'), -- New York, United States (using US Dollar)
  ('LAX', 'USD'), -- Los Angeles, United States (using US Dollar)
  ('FRA', 'EUR'), -- Frankfurt, Germany (using Euro)
  ('SIN', 'SGD'), -- Singapore (using Singapore Dollar)
  ('DXB', 'AED'), -- Dubai, United Arab Emirates (assuming AED currency
exists)
  ('HKG', 'HKD'), -- Hong Kong (assuming HKD currency exists)
  ('SYD', 'AUD'), -- Sydney, Australia (using Australian Dollar)
  ('PEK', 'CNY'), -- Beijing, China (assuming CNY currency exists)
  ('DOH', 'QAR'), -- Doha, Qatar (assuming QAR currency exists)
  ('GRU', 'BRL'); -- São Paulo, Brazil (assuming BRL currency exists)



INSERT INTO AIRLINKS (AIRLINK_CODE, AIRLINK_DISTANCE,
AIRPORTS_AIRPORT_CODE_from, AIRPORTS_AIRPORT_CODE_to)
VALUES
  (1234, '1500', 'CDG', 'LHR'), -- Airlink 1234 from Paris (CDG) to London
(LHR), 1500 km
  (2345, '2800', 'JFK', 'LAX'), -- Airlink 2345 from New York (JFK) to Los
Angeles (LAX), 2800 km
```

```
  (3456, '4200', 'FRA', 'SIN'), -- Airlink 3456 from Frankfurt (FRA) to
Singapore (SIN), 4200 km
  (4567, '3500', 'DXB', 'HKG'); -- Airlink 4567 from Dubai (DXB) to Hong Kong
(HKG), 3500 km


INSERT INTO PLAN_FLIGHTS (FLIGHT_NBR, DEPARTURE, FREQUENCY,
AIRLINKS_AIRLINK_CODE)
VALUES
  (101, '10:00:00', 7, 1234), -- Daily flight (7 times a week) with Airlink
code 1234
  (102, '13:00:00', 5, 2345), -- Flight on weekdays (5 times a week) with
Airlink code 2345
  (103, '09:00:00', 2, 3456), -- Flight twice a week (e.g., Tuesdays &
Thursdays) with Airlink code 3456
  (104, '15:00:00', 3, 4567), -- Flight three times a week (specify weekdays
if needed) with Airlink code 4567
  (105, '10:00:00', 7, 1234); -- Daily flight (7 times a week) with Airlink
code 1234 (Assuming repeated schedule



INSERT INTO PLAN_DAYFLIGHTS (WEEKDAYS_DAY_NBR, PLAN_FLIGHTS_FLIGHT_NBR,
PLAN_SEATS, PLAN_DURATION, PLAN_ARR_TIME, KoAIRCRAFT_IATA)
VALUES
  (1, 101, 180, '01:15:00', '11:45:00', 'A320'), -- Monday Flight 101
  (2, 102, 160, '01:35:00', '16:45:00', 'B737'), -- Tuesday Flight 102
  (3, 103, 290, '04:15:00', '12:15:00', 'A330'), -- Wednesday Flight 103
  (4, 104, 360, '03:30:00', '15:30:00', 'B777'), -- Thursday Flight 104
  (5, 105, 180, '01:15:00', '11:45:00', 'A320'), -- Friday Flight 105
  (6, 101, 140, '01:00:00', '11:00:00', 'A320'), -- Saturday Flight 101
  (6, 102, 120, '01:20:00', '15:00:00', 'B737'); -- Saturday Flight 102

INSERT INTO WEEKDAYS (DAY_NBR)
VALUES (1), (2), (3), (4), (5), (6), (7);



SELECT KoAIRCRAFT.IATA, COUNT(*) AS nb_flights
FROM EFF_FLIGHTS
INNER JOIN AIRCRAFTS ON EFF_FLIGHTS.AIRCRAFTS_AIRCRAFT_NBR =
AIRCRAFTS.AIRCRAFT_NBR
INNER JOIN KoAIRCRAFT ON AIRCRAFTS.KoAIRCRAFT_IATA = KoAIRCRAFT.IATA
GROUP BY KoAIRCRAFT.IATA
ORDER BY nb_flights DESC
LIMIT 5;
```

Sortie:

	iata	nb_flights
1	A320	3
2	B737	2
3	B777	1
4	A330	1