DROP TABLE IF EXISTS student;

DROP TABLE IF EXISTS bootcamp;

DROP TABLE IF EXISTS matricula;

DROP TABLE IF EXISTS teacher;

DROP TABLE IF EXISTS modulo;

CREATE TABLE Student(

student\_id SERIAL PRIMARY KEY

,name VARCHAR (50)

,surname VARCHAR (255)

,email VARCHAR (255)

,DNI VARCHAR (50)

,Phone VARCHAR (50)

,Adress VARCHAR (255)

CREATE TABLE Bootcamp (

bootcamp\_id SERIAL PRIMARY KEY

,service\_id INT NOT NULL

,name VARCHAR (255)

,start\_date DATE

,end\_date DATE

,stands\_number INT,

FOREIGN KEY(service\_id)REFERENCES services(service\_id)

);

CREATE TABLE matricula (

matricula\_id SERIAL PRIMARY KEY,

bootcamp\_id INT NOT NULL,

alumno\_id INT NOT NULL,

fecha\_matriculacion DATE NOT NULL,

FOREIGN KEY (bootcamp\_id) REFERENCES bootcamp(bootcamp\_id),

FOREIGN KEY (alumno\_id) REFERENCES alumno(alumno\_id),

UNIQUE (bootcamp\_id, alumno\_id, fecha\_matriculacion)

);

CREATE TABLE teacher (

teacher\_id SERIAL PRIMARY KEY,

nombre VARCHAR(255) NOT NULL,

apellido VARCHAR(255) NOT NULL,

identificador\_fiscal VARCHAR(20) NOT NULL UNIQUE,

email VARCHAR(255) NOT NULL UNIQUE

);

CREATE TABLE modulo (

modulo\_id SERIAL PRIMARY KEY,

teacher\_id INT,

bootcamp\_id INT,

nombre VARCHAR(255) NOT NULL,

horas INT NOT NULL,

FOREIGN KEY (bootcamp\_id) REFERENCES bootcamp(bootcamp\_id),

FOREIGN KEY (teacher\_id) REFERENCES teacher (teacher\_id)

);

INSERT INTO student(nombre, apellido, identificador\_fiscal, email) VALUES

('Pedro', 'Pascal', '123456789', 'pascal.pedro@email.com'),

('Maria', 'Gomez', '458796121', 'maria.gomez@email.com'),

('Cristiano', 'Ronaldo', '624893256', 'elbicho.CR7@email.com'),

('Lucia', 'Diaz', '315689255', 'lucia.diaz@email.com'),

('Paco', 'Rabbane', '234567890', 'paquito@email.com');

INSERT INTO bootcamp (nombre) VALUES

('Developer),

('Web'),

('IA'),

(‘Big Data'),

('Ciberseguridad');

INSERT INTO matricula (bootcamp\_id, student\_id, fecha\_matriculacion) VALUES

(1, 1, '2023-01-01'),

(2, 2, '2023-08-06'),

(3, 3, '2023-03-05'),

(4, 4, '2023-09-19'),

(5, 5, '2023-03-20');

INSERT INTO teacher (nombre, apellido, identificador\_fiscal, email) VALUES

('María', 'Perez', '012345678', 'mrp@email.com'),

('Laura', 'González', '687436102', 'lglez@email.com'),

('Alex', 'Bento', '6328997451', ‘abent@email.com'),

('Angel', 'García', '795446321', 'ag@email.com'),

('Nerea', 'Trapero', '6897965589', 'nerea.tb@email.com');

INSERT INTO modulo (teacher\_id, bootcamp\_id, nombre, horas) VALUES

(1, 1, 'Cloud', 10),

(2, 2, 'Python', 6),

(3, 3, 'SQL', 25),

(4, 4, 'NLP', 8),

(5, 5, 'ML, 23);

INSERT INTO modulo\_bootcamp (modulo\_id, bootcamp\_id) VALUES

(1, 1),

(2, 4),

(3, 5),

(4, 2),

(5, 3);

SELECT alumno.studentAS email\_student

, matricula.matricula\_id

, matricula.fecha\_matriculacion

, bootcamp.nombre AS bootcamp

, modulo.nombre AS modulo

, teacher.email AS teacher\_profesor

FROM student

JOIN matricula

ON student.student\_id = matricula.student\_id

JOIN bootcamp

ON matricula.bootcamp\_id = bootcamp.bootcamp\_id

JOIN modulo

ON bootcamp.bootcamp\_id = modulo.bootcamp\_id

JOIN profesor

ON modulo.teacher\_id = teacher.teacher\_id

JOIN modulo\_bootcamp

ON modulo.modulo\_id = modulo\_bootcamp\_id

CREATE OR REPLACE TABLE keepcoding.ivr\_detail AS

SELECT calls.ivr\_id

, calls.phone\_number

, calls.ivr\_result

, calls.vdn\_label

, calls.start\_date

, FORMAT\_TIMESTAMP('%Y%m%d', calls.start\_date) AS start\_date\_id

, calls.end\_date

, FORMAT\_TIMESTAMP('%Y%m%d', calls.end\_date) AS end\_date\_id

, calls.total\_duration

, calls.customer\_segment

, calls.ivr\_language

, calls.steps\_module

, calls.module\_aggregation

, modules.module\_sequece

, modules.module\_name

, modules.module\_duration

, modules.module\_result

, steps.step\_sequence

, steps.step\_name

, steps.step\_result

, steps.step\_description\_error

, steps.document\_type

, steps.document\_identification

, steps.customer\_phone

, steps.billing\_account\_id

FROM keepcoding.ivr\_calls calls

LEFT

JOIN keepcoding.ivr\_modules modules

ON calls.ivr\_id = modules.ivr\_id

LEFT

JOIN keepcoding.ivr\_steps steps

ON calls.ivr\_id = steps.ivr\_id

AND modules.module\_sequece = steps.module\_sequece;

CREATE OR REPLACE TABLE keepcoding.ivr\_summary AS

WITH documentation

AS (SELECT CAST(ivr\_id AS STRING) AS ivr\_id

, document\_identification

, document\_type

, module\_sequece

, step\_sequence

FROM keepcoding.ivr\_detail

WHERE document\_identification NOT IN ('UNKNOWN', 'DESCONOCIDO')

QUALIFY ROW\_NUMBER() OVER(PARTITION BY ivr\_id ORDER BY module\_sequece DESC, step\_sequence DESC) = 1)

SELECT calls.ivr\_id

, calls.phone\_number

, calls.ivr\_result

, CASE WHEN LEFT(calls.vdn\_label, 3) = 'ATC' THEN 'FRONT'

WHEN LEFT(calls.vdn\_label, 4) = 'TECH' THEN 'TECH'

WHEN calls.vdn\_label = 'ABSORPTION' THEN 'ABSORPTION'

ELSE 'RESTO'

END AS vdn\_aggregation

, calls.start\_date

, calls.end\_date

, calls.total\_duration

, calls.customer\_segment

, calls.ivr\_language

, calls.steps\_module

, calls.module\_aggregation

, IFNULL(documentation.document\_type, 'DESCONOCIDO') AS document\_type

, IFNULL(documentation.document\_identification, 'DESCONOCIDO') AS document\_identification

, IFNULL(MAX(NULLIF(calls.customer\_phone, 'UNKNOWN')), 'DESCONOCIDO') AS customer\_phone

, IFNULL(MAX(NULLIF(calls.billing\_account\_id, 'UNKNOWN')), 'DESCONOCIDO') AS billing\_account\_id

, MAX(IF(calls.module\_name = "AVERIA\_MASIVA", 1, 0)) AS masiva\_lg

, MAX(IF(calls.step\_name = 'CUSTOMERINFOBYPHONE.TX' AND calls.step\_description\_error = 'UNKNOWN', 1, 0)) AS info\_by\_phone\_lg

, MAX(IF(calls.step\_name = 'CUSTOMERINFOBYDNI.TX' AND calls.step\_description\_error = 'UNKNOWN', 1, 0)) AS info\_by\_dni\_lg

, MAX(IF(DATE\_DIFF(calls.start\_date, recalls.start\_date, SECOND) BETWEEN 1 AND 24\*60\*60, 1, 0)) AS repeated\_phone\_24H

, MAX(IF(DATE\_DIFF(calls.start\_date, recalls.start\_date, SECOND) BETWEEN -24\*60\*60 AND -1, 1, 0)) AS cause\_recall\_phone\_24H

FROM keepcoding.ivr\_detail calls

LEFT

JOIN documentation

ON CAST(calls.ivr\_id AS STRING) = documentation.ivr\_id

LEFT

JOIN keepcoding.ivr\_detail recalls

ON calls.phone\_number <> 'UNKNOWN'

AND calls.phone\_number = recalls.phone\_number

AND calls.ivr\_id <> recalls.ivr\_id

GROUP BY ivr\_id

, phone\_number

, ivr\_result

, vdn\_aggregation

, start\_date

, end\_date

, total\_duration

, customer\_segment

, ivr\_language

, steps\_module

, module\_aggregation

, document\_type

, document\_identification;

CREATE OR REPLACE FUNCTION keepcoding.clean\_integer (numero INT64) RETURNS INT64

AS ((SELECT IFNULL(numero, -999999)));