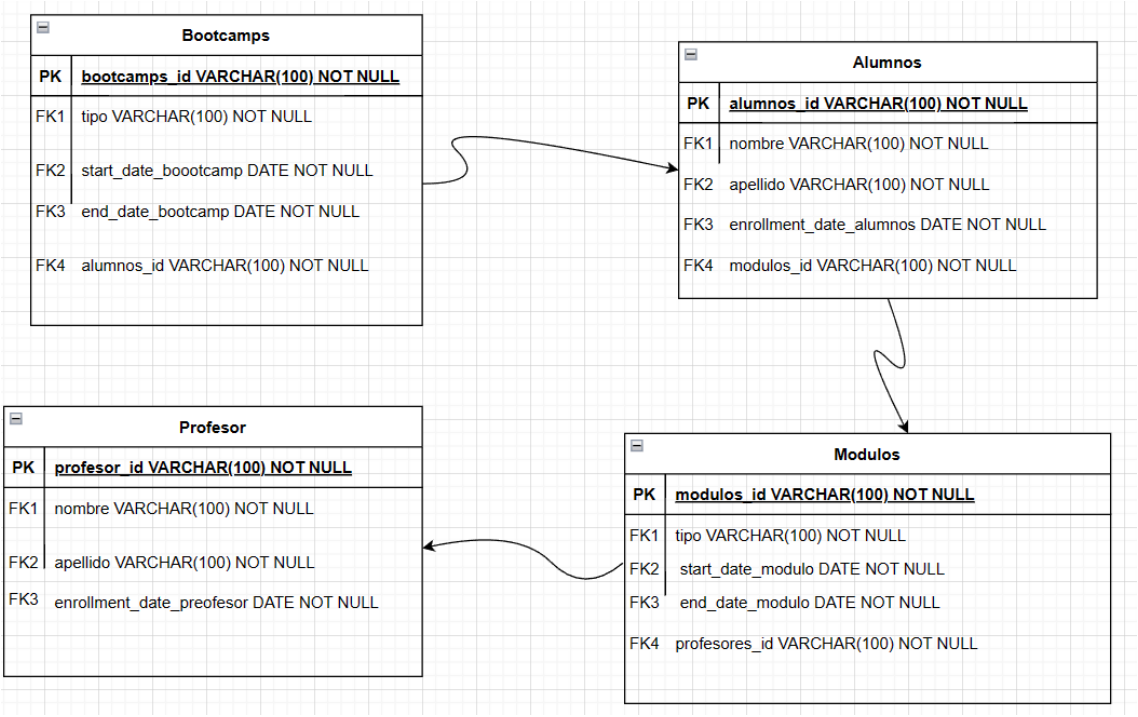
Practica SQL Avanzado, ETL y SQL Avanzado, ETL y DataWarehouse

Ejercicio 1



Diseñe mi modelo con una cardinalidad 1:1, ya que creo que es la que da una visión sencilla y clara, en la cual los bootcamps tienen alumnos los cuales asisten a los modulos y cada modulo tiene su profesor, lo realice asi ya que los alumnos que llevan un bootcamp llevaran los mismos modulos luego con esto evito redundancia y mejoro la integridad y bueno al final los relaciones de acuerdo al orden de las entidades es decir las claves primarias

Ejercicio 2

```
CREATE TABLE Bootcamps (
    bootcamps_id SERIAL PRIMARY KEY,
    tipo VARCHAR(100),
    start_date_bootcamp DATE NOT NULL,
    end_date_bootcamp DATE NOT NULL,
    alumnos_id INT NOT NULL
);
CREATE TABLE Alumnos (
    alumnos_id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    enrollment_date_alumnos DATE NOT NULL,
    modulos_id INT NOT NULL,
    bootcamps_id INT,
    FOREIGN KEY (bootcamps_id) REFERENCES Bootcamps(bootcamps_id)
);
CREATE TABLE Modulos (
    modulos_id SERIAL PRIMARY KEY,
    tipo VARCHAR(100) NOT NULL,
    start_date_modulo DATE NOT NULL,
    end_date_modulo DATE NOT NULL,
    profesores_id INT NOT NULL,
    alumnos_id INT,
    FOREIGN KEY (alumnos_id) REFERENCES Alumnos(alumnos_id)
);
CREATE TABLE Profesor (
    profesor_id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    enrollment_date_profesor DATE NOT NULL,
    modulos_id INT,
    FOREIGN KEY (modulos_id) REFERENCES Modulos(modulos_id)
);
```

Ejercicio 3

```
CREATE OR REPLACE TABLE keepcoding.ivr_detail AS
SELECT calls.ivr_id
     , calls.phone_number
     , calls.ivr_result
     , calls.vdn_label
     , calls.start_date
     , calls.end_date
     , calls.total_duration
     , calls.customer_segment
     , calls.ivr_language
     , calls.steps_module
     , calls.module_aggregation
     , modules.module_name
     , modules.module_duration
     , modules.module_result
     , steps.module_sequece
     , 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
     , FORMAT_DATE('%Y%m%d', start_date) AS calls_start_date_id,
     FORMAT_DATE('%Y%m%d', end_date) AS calls_end_date_id,
 FROM practica-keep-coding.keepcoding.ivr_calls as calls
 LEFT
 JOIN practica-keep-coding.keepcoding.ivr_modules as modules
 ON calls.ivr_id = modules.ivr_id
 LEFT
 JOIN practica-keep-coding.keepcoding.ivr_steps as steps
 ON calls.ivr_id = steps.ivr_id;
```

Ejercicio 4

```
create or replace table keepcoding.ivr_summary AS

SELECT  detail.ivr_id
      , detail.phone_number
      , detail.ivr_result
      , detail.vdn_label
      , detail.start_date
      , detail.end_date
      , detail.total_duration
      , detail.customer_segment
      , detail.ivr_language
      , detail.steps_module
      , detail.module_aggregation
      , detail.module_name
      , detail.module_duration
      , detail.module_result
      , detail.module_sequece
      , detail.step_sequence
      , detail.step_name
      , detail.step_result
      , detail.step_description_error
      , detail.document_type
      , detail.document_identification
      , detail.customer_phone
      , detail.billing_account_id
      , CASE WHEN detail.module_aggregation = 'AVERIA_MASIVA' THEN 1 ELSE 0 END AS
masiva_lg,
        CASE WHEN detail.step_name = 'CUSTOMERINFOBYPHONE.TX'
              AND detail.step_description_error = 'UNKNOWN'
          THEN 1
          ELSE NULL
          END AS info_by_phone_lg,
        CASE WHEN detail.step_name = 'CUSTOMERINFOBYDNI.TX'
              AND detail.step_description_error = 'UNKNOWN'
          THEN 1
          ELSE 0
          END AS info_by_dni_lg,
       CASE WHEN (
        SELECT COUNT(*)
        FROM keepcoding.ivr_detail as previous_calls
        WHERE previous_calls.phone_number = detail.phone_number
        AND TIMESTAMP_DIFF(detail.start_date, previous_calls.end_date, HOUR) <= 24
    ) > 0 THEN 1 ELSE 0 END AS repeated_phone_24H,
      CASE WHEN (
        SELECT COUNT(*)
        FROM keepcoding.ivr_detail as future_calls
        WHERE future_calls.phone_number = detail.phone_number
        AND TIMESTAMP_DIFF(future_calls.start_date, detail.end_date, HOUR) <= 24
    ) > 0 THEN 1 ELSE 0 END AS cause_recall_phone_24H,
  FROM practica-keep-coding.keepcoding.ivr_detail as detail
```

Ejercicio 5

```
CREATE FUNCTION keepcoding.clean_integer(dato INT64) AS (
  IFNULL(dato, -999999));
```