1. How many schemas will be there in the database:
   * + 1. STND: Standard Schema track every standard aspect with Error logging and API trail with API error logging.
          1. All the procedures and functions related to the calculation on all other schemas.
          2. All the standard triggers and cursors.
       2. USR: All the user related tables will remain in this schema also the procedures with perform DDL, DML operations on the same.
       3. ADMN: All the tables and procedures related to admin.
       4. EMP: All the trainers and the employees with the support staff with all the specified EMPLOYEE\_TYPE (i.e., Trainers and Helpers will be contract type employee and all the employees will be permanent).
       5. EQUP: All the machineries and the best time for revival of the machineries with the price and other details.
       6. TRANSACTIONS: ALL Things related to payments and transaction and payments.
2. Tables to be created in STND Schema: Examples

[STND]. [LOG\_API\_AUDIT\_TRAIL]

[STND]. [LOG\_STORE]

[STND]. [MENU\_MASTER]

[STND]. [STND\_EMAIL\_TEMPLATE]

[STND]. [STND\_MAIL\_ERRORS]

[STND]. [STND\_MAIL\_PROCESS]

[STND]. [STND\_TRANSLATION]

[STND]. [SYSTEM\_CONFIG]

[STND]. [USER\_USAGE\_TRACKER]

[STND]. [USER\_USAGE\_TRACKER\_DETAIL\_UUTD]

1. Tables in reference:

CREATE TABLE "users" (

"id" SERIAL PRIMARY KEY,

"username" varchar,

"password" varchar,

"full\_name" varchar,

"dob" date,

"created\_at" timestamp,

"city" varchar,

"state" varchar,

"contact" int,

"email" varchar,

"Gender" char,

"country\_code" int,

"user\_id" int

);

CREATE TABLE "member" (

"id" SERIAL PRIMARY KEY,

"joining\_date" date,

"end\_of\_membership" date,

"membership\_id" int,

"user\_id" int

);

CREATE TABLE "instructor" (

"id" int PRIMARY KEY,

"level" int,

"user\_id" int

);

CREATE TABLE "admin" (

"id" int PRIMARY KEY,

"user\_id" int

);

CREATE TABLE "countries" (

"code" int PRIMARY KEY,

"name" varchar,

"continent\_name" varchar

);

CREATE TABLE "membership\_type" (

"id" int PRIMARY KEY,

"name" varchar,

"membership\_amount" int,

"membership\_period" int

);

CREATE TABLE "workout\_plan" (

"id" int PRIMARY KEY,

"instructor\_id" int,

"member\_id" int,

"workout\_id" int,

"workout\_date" date,

"workout\_time" time

);

CREATE TABLE "workout" (

"id" SERIAL PRIMARY KEY,

"name" varchar,

"equipment\_id" int,

"description" varchar

);

CREATE TABLE "payment" (

"id" int PRIMARY KEY,

"amount" float4,

"status" varchar,

"member\_id" int,

"user\_id" int,

"timestamp" datetime

);

CREATE TABLE "equipment" (

"id" int PRIMARY KEY,

"amount" float4,

"stateus" varchar,

"revival\_date" date

);

CREATE TABLE "credit\_card" (

"credicard\_id" int PRIMARY KEY,

"cc\_number" varchar,

"cc\_name" varchar,

"expiry\_date" varchar

);

CREATE TABLE "invoice" (

"id" int PRIMARY KEY,

"state" varchar,

"description" varchar,

"notes" varchar,

"timestamp" datetime,

"payment\_id" int

);

ALTER TABLE "users" ADD FOREIGN KEY ("country\_code") REFERENCES "countries" ("code");

ALTER TABLE "member" ADD FOREIGN KEY ("membership\_id") REFERENCES "membership\_type" ("id");

ALTER TABLE "member" ADD FOREIGN KEY ("user\_id") REFERENCES "users" ("id");

ALTER TABLE "instructor" ADD FOREIGN KEY ("user\_id") REFERENCES "users" ("id");

ALTER TABLE "admin" ADD FOREIGN KEY ("user\_id") REFERENCES "users" ("id");

ALTER TABLE "payment" ADD FOREIGN KEY ("member\_id") REFERENCES "member" ("id");

ALTER TABLE "payment" ADD FOREIGN KEY ("user\_id") REFERENCES "users" ("id");

ALTER TABLE "invoice" ADD FOREIGN KEY ("payment\_id") REFERENCES "payment" ("id");

ALTER TABLE "workout\_plan" ADD FOREIGN KEY ("workout\_id") REFERENCES "workout" ("id");

ALTER TABLE "workout\_plan" ADD FOREIGN KEY ("member\_id") REFERENCES "member" ("id");

ALTER TABLE "workout\_plan" ADD FOREIGN KEY ("instructor\_id") REFERENCES "instructor" ("id");

ALTER TABLE "workout" ADD FOREIGN KEY ("equipment\_id") REFERENCES "equipment" ("id");