**Primero tener creado el usuario principal**

mutation {

createUser(input:{

name:"Admin",

email:"admin@demo.com",

password:"Secreta12",

isAuditor:false,

permissions:[

{module: USERS, permission: MUTATE},

{module: HABITS, permission: MUTATE},

{module: ROUTINES, permission: MUTATE},

{module: GUIDES, permission: MUTATE},

{module: REMINDERS, permission: MUTATE},

{module: PROGRESS, permission: MUTATE}

]

}){

id name email isAuditor

permissions { module permission }

}

}

**Siguiente pegar en mariaDB:**

/\* =========================================================

0) Preparación

========================================================= \*/

CREATE DATABASE IF NOT EXISTS rutinasdb

DEFAULT CHARACTER SET utf8mb4

COLLATE utf8mb4\_uca1400\_ai\_ci;

USE rutinasdb;

/\* =========================================================

0.1) CONFIG: correo de tu ADMIN existente (opcional)

========================================================= \*/

SET @ADMIN\_EMAIL = 'admin@demo.com'; -- cámbialo si tu admin usa otro correo

/\* =========================================================

1) Limpieza segura (NO tocamos user ni user\_module\_permission)

========================================================= \*/

SET FOREIGN\_KEY\_CHECKS=0;

TRUNCATE TABLE completedactivity;

TRUNCATE TABLE routine\_habit;

TRUNCATE TABLE routine;

TRUNCATE TABLE favorite\_habit;

TRUNCATE TABLE reminder;

TRUNCATE TABLE guide\_habit;

TRUNCATE TABLE guide;

TRUNCATE TABLE habitactivity;

SET FOREIGN\_KEY\_CHECKS=1;

/\* =========================================================

2) Sembrar 199 usuarios demo si faltan (idempotente)

- password = "password" (bcrypt)

- Coaches: n=1..10 -> is\_coach=1

- Auditores: n=11..15 -> is\_auditor=1

- Restantes: usuarios normales

- NO usamos el campo 'role'

========================================================= \*/

INSERT INTO `user`(name,email,password,is\_auditor,is\_coach)

SELECT CONCAT('User', n),

CONCAT('user', n, '@demo.com'),

'{bcrypt}$2a$10$7EqJtq98hPqEX7fNZaFWoOe.D7Z2qxdjQ671O5j3N6zGyF/F7QKta',

CASE WHEN n BETWEEN 11 AND 15 THEN 1 ELSE 0 END,

CASE WHEN n BETWEEN 1 AND 10 THEN 1 ELSE 0 END

FROM (

SELECT a.d + 10\*b.d + 100\*c.d + 1 AS n

FROM (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1) c

) t

LEFT JOIN `user` u ON u.email = CONCAT('user', n, '@demo.com')

WHERE n BETWEEN 1 AND 199

AND u.id IS NULL;

/\* =========================================================

2.1) Asignar coachees automáticamente (round-robin) a coaches demo

========================================================= \*/

DROP TABLE IF EXISTS tmp\_coaches;

CREATE TABLE tmp\_coaches (

seq INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

coach\_id BIGINT NOT NULL

) ENGINE=InnoDB;

INSERT INTO tmp\_coaches(coach\_id)

SELECT id FROM `user` WHERE is\_coach=1 ORDER BY id;

SET @COACH\_COUNT = (SELECT COUNT(\*) FROM tmp\_coaches);

-- asignar coach a todos los NO-coach (si hay coaches)

UPDATE `user` u

JOIN (

SELECT id, ((id % @COACH\_COUNT) + 1) AS seq

FROM `user`

WHERE is\_coach = 0

) s ON u.id = s.id

JOIN tmp\_coaches tc ON tc.seq = s.seq

SET u.assigned\_coach\_id = CASE WHEN @COACH\_COUNT > 0 THEN tc.coach\_id ELSE NULL END

WHERE @COACH\_COUNT > 0 AND u.is\_coach = 0;

/\* =========================================================

3) Permisos por módulo (idempotentes)

- ADMIN (por correo): MUTATE en todos los módulos

- Coaches (is\_coach=1): MUTATE en HABITS/REMINDERS, CONSULT en otros

- Auditores (is\_auditor=1): CONSULT en todos

- Users normales: CONSULT en HABITS/REMINDERS

========================================================= \*/

SET @ADMIN\_ID = (SELECT id FROM `user` WHERE email = @ADMIN\_EMAIL LIMIT 1);

-- ADMIN: MUTATE en todos

INSERT INTO user\_module\_permission(user\_id, module, permission)

SELECT @ADMIN\_ID, m.module, 'MUTATE'

FROM (SELECT 'GUIDES' module UNION ALL SELECT 'HABITS' UNION ALL SELECT 'PROGRESS'

UNION ALL SELECT 'REMINDERS' UNION ALL SELECT 'ROUTINES' UNION ALL SELECT 'USERS') m

WHERE @ADMIN\_ID IS NOT NULL

ON DUPLICATE KEY UPDATE permission=VALUES(permission);

-- COACH: MUTATE en HABITS/REMINDERS

INSERT INTO user\_module\_permission(user\_id, module, permission)

SELECT u.id, 'HABITS', 'MUTATE' FROM `user` u WHERE u.is\_coach=1

ON DUPLICATE KEY UPDATE permission=VALUES(permission);

INSERT INTO user\_module\_permission(user\_id, module, permission)

SELECT u.id, 'REMINDERS', 'MUTATE' FROM `user` u WHERE u.is\_coach=1

ON DUPLICATE KEY UPDATE permission=VALUES(permission);

-- COACH: CONSULT en GUIDES/PROGRESS/ROUTINES/USERS

INSERT INTO user\_module\_permission(user\_id, module, permission)

SELECT u.id, m.module, 'CONSULT'

FROM `user` u

JOIN (SELECT 'GUIDES' module UNION ALL SELECT 'PROGRESS' UNION ALL SELECT 'ROUTINES' UNION ALL SELECT 'USERS') m

WHERE u.is\_coach=1

ON DUPLICATE KEY UPDATE permission=VALUES(permission);

-- AUDITOR: CONSULT en todos

INSERT INTO user\_module\_permission(user\_id, module, permission)

SELECT u.id, m.module, 'CONSULT'

FROM `user` u

JOIN (SELECT 'GUIDES' module UNION ALL SELECT 'HABITS' UNION ALL SELECT 'PROGRESS'

UNION ALL SELECT 'REMINDERS' UNION ALL SELECT 'ROUTINES' UNION ALL SELECT 'USERS') m

WHERE u.is\_auditor=1

ON DUPLICATE KEY UPDATE permission=VALUES(permission);

-- USER normal: CONSULT en HABITS/REMINDERS

INSERT INTO user\_module\_permission(user\_id, module, permission)

SELECT u.id, 'HABITS', 'CONSULT' FROM `user` u WHERE u.is\_coach=0 AND u.is\_auditor=0

ON DUPLICATE KEY UPDATE permission=VALUES(permission);

INSERT INTO user\_module\_permission(user\_id, module, permission)

SELECT u.id, 'REMINDERS', 'CONSULT' FROM `user` u WHERE u.is\_coach=0 AND u.is\_auditor=0

ON DUPLICATE KEY UPDATE permission=VALUES(permission);

/\* =========================================================

4) Preparar lista de usuarios reales para referenciar (hasta 200)

========================================================= \*/

DROP TABLE IF EXISTS tmp\_users;

CREATE TABLE tmp\_users (

seq INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

user\_id BIGINT NOT NULL

) ENGINE=InnoDB;

INSERT INTO tmp\_users(user\_id)

SELECT id FROM `user` ORDER BY id LIMIT 200;

SET @USR\_COUNT = (SELECT COUNT(\*) FROM tmp\_users);

-- seguridad: al menos 1 usuario

DO CASE WHEN @USR\_COUNT = 0 THEN (SELECT 1/0) ELSE 0 END;

/\* =========================================================

5) Catálogo de hábitos (50)

========================================================= \*/

INSERT INTO habitactivity(name,category,description,duration,targetTime,notes,is\_favorite,target\_time)

SELECT CONCAT('Habit ', n),

CASE (n % 4)

WHEN 0 THEN 'PHYSICAL'

WHEN 1 THEN 'MENTAL'

WHEN 2 THEN 'SLEEP'

ELSE 'DIET'

END,

CONCAT('Descripción del hábito #', n),

10 + (n % 50),

MAKETIME(6 + (n % 12), (n\*3) % 60, 0),

CONCAT('Nota #', n),

b'0',

MAKETIME(6 + (n % 12), (n\*3) % 60, 0)

FROM (

SELECT a.d + 10\*b.d + 1 AS n

FROM (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4) b

) nums

WHERE n <= 50;

/\* =========================================================

6) Rutinas (20) con propietarios reales (ARREGLO: genera 1..20)

========================================================= \*/

DROP TABLE IF EXISTS tmp\_owners;

CREATE TABLE tmp\_owners AS

SELECT seq, user\_id FROM tmp\_users WHERE seq <= 20;

INSERT INTO routine(title,user\_id,daysOfWeek,days\_of\_week)

SELECT CONCAT('Rutina ', n), o.user\_id, 'MON,WED,FRI', 'MON,WED,FRI'

FROM (

SELECT a.d + 10\*b.d + 1 AS n

FROM (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1) b

) nums

JOIN tmp\_owners o ON o.seq = ((n-1) % (SELECT GREATEST(1,COUNT(\*)) FROM tmp\_owners)) + 1

WHERE n <= 20;

/\* =========================================================

7) routine\_habit (5 hábitos por rutina)

========================================================= \*/

INSERT INTO routine\_habit(routine\_id, habit\_id, order\_in\_routine, target\_time\_in\_routine, notes)

SELECT r.id,

1 + ((r.id + offs) % 50) AS habit\_id,

offs + 1,

MAKETIME(6 + offs, 0, 0),

CONCAT('Item ', offs+1)

FROM routine r

JOIN (SELECT 0 AS offs UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4) o;

/\* =========================================================

8) Dataset masivo: completedactivity (~500 000) con usuarios reales

========================================================= \*/

DROP TABLE IF EXISTS tmp\_seq;

CREATE TABLE tmp\_seq (n INT NOT NULL PRIMARY KEY) ENGINE=InnoDB;

INSERT INTO tmp\_seq VALUES (1);

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 2

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 4

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 8

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 16

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 32

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 64

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 128

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 256

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 512

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 1024

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 2048

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 4096

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 8192

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 16384

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 32768

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 65536

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 131072

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 262144

INSERT INTO tmp\_seq SELECT n + (SELECT MAX(n) FROM tmp\_seq) FROM tmp\_seq; -- 524288

DELETE FROM tmp\_seq WHERE n > 500000;

INSERT INTO completedactivity(user\_id, routine\_id, habit\_id, `date`, completedAt, completed\_at, notes)

SELECT

(SELECT tu.user\_id FROM tmp\_users tu WHERE tu.seq = (n % @USR\_COUNT) + 1) AS user\_id,

1 + (n % 20) AS routine\_id,

1 + (n % 50) AS habit\_id,

DATE\_ADD('2023-01-01', INTERVAL (n % 730) DAY) AS `date`,

CONCAT(LPAD(n % 24,2,'0'), ':', LPAD(n % 60,2,'0')) AS completedAt,

CONCAT(LPAD(n % 24,2,'0'), ':', LPAD(n % 60,2,'0')) AS completed\_at,

CONCAT('auto #', n) AS notes

FROM tmp\_seq;

DROP TABLE tmp\_seq;

/\* =========================================================

9) Guides (120) y guide\_habit (2 hábitos por guía)

========================================================= \*/

INSERT INTO guide(title, content, category, user\_id)

SELECT CONCAT('Guide ', n),

CONCAT('Contenido de la guía #', n),

CASE (n % 4)

WHEN 0 THEN 'PHYSICAL'

WHEN 1 THEN 'MENTAL'

WHEN 2 THEN 'SLEEP'

ELSE 'DIET'

END,

(SELECT tu.user\_id FROM tmp\_users tu WHERE tu.seq = ((n - 1) % @USR\_COUNT) + 1)

FROM (

SELECT a.d + 10\*b.d + 1 AS n

FROM (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9

UNION ALL SELECT 10 UNION ALL SELECT 11) b

) t

WHERE n <= 120;

INSERT INTO guide\_habit(guide\_id, habit\_id)

SELECT g.id, 1 + ((g.id + o.off) % 50)

FROM guide g

JOIN (SELECT 0 AS off UNION ALL SELECT 7) o;

/\* =========================================================

10) Reminders (10 000) y Favorite Habits (10 000)

========================================================= \*/

INSERT INTO reminder(user\_id, habit\_id, `time`, frequency)

SELECT

(SELECT tu.user\_id FROM tmp\_users tu WHERE tu.seq = (n % @USR\_COUNT) + 1),

1 + (n % 50),

CONCAT(LPAD(6 + (n % 12), 2, '0'), ':', LPAD((n\*5) % 60, 2, '0')),

CASE (n % 3) WHEN 0 THEN 'DAILY' WHEN 1 THEN 'WEEKLY' ELSE 'MONTHLY' END

FROM (

SELECT (a.d + 10\*b.d + 100\*c.d + 1000\*d.d) AS n

FROM (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) c

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) d

) t

WHERE n < 10000;

INSERT INTO favorite\_habit(habit\_id, user\_id)

SELECT

1 + (n % 50) AS habit\_id,

(SELECT tu.user\_id FROM tmp\_users tu WHERE tu.seq = ((n DIV 50) % @USR\_COUNT) + 1) AS user\_id

FROM (

SELECT (a.d + 10\*b.d + 100\*c.d + 1000\*d.d) AS n

FROM (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) c

CROSS JOIN (SELECT 0 d UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4

UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) d

) t

WHERE n < 10000;

/\* =========================================================

11) Índices (no tocamos el índice del coach por la FK)

========================================================= \*/

DROP INDEX IF EXISTS idx\_completed\_date\_id ON completedactivity;

DROP INDEX IF EXISTS idx\_completed\_user\_date\_id ON completedactivity;

DROP INDEX IF EXISTS idx\_completed\_habit\_date\_id ON completedactivity;

DROP INDEX IF EXISTS idx\_completed\_routine\_date\_id ON completedactivity;

CREATE INDEX idx\_completed\_date\_id ON completedactivity(`date`, id);

CREATE INDEX idx\_completed\_user\_date\_id ON completedactivity(user\_id, `date`, id);

CREATE INDEX idx\_completed\_habit\_date\_id ON completedactivity(habit\_id, `date`, id);

CREATE INDEX idx\_completed\_routine\_date\_id ON completedactivity(routine\_id, `date`, id);

DROP INDEX IF EXISTS idx\_user\_name\_id ON `user`;

CREATE INDEX idx\_user\_name\_id ON `user`(name, id);

DROP INDEX IF EXISTS idx\_routine\_user\_id ON routine;

CREATE INDEX idx\_routine\_user\_id ON routine(user\_id, id);

DROP INDEX IF EXISTS idx\_reminder\_user\_id ON reminder;

DROP INDEX IF EXISTS idx\_reminder\_habit\_id ON reminder;

CREATE INDEX idx\_reminder\_user\_id ON reminder(user\_id, id);

CREATE INDEX idx\_reminder\_habit\_id ON reminder(habit\_id, id);

DROP INDEX IF EXISTS idx\_fav\_user\_id ON favorite\_habit;

CREATE INDEX idx\_fav\_user\_id ON favorite\_habit(user\_id, id);

DROP INDEX IF EXISTS idx\_guide\_user\_id ON guide;

CREATE INDEX idx\_guide\_user\_id ON guide(user\_id, id);

DROP INDEX IF EXISTS idx\_guide\_habit\_habit ON guide\_habit;

CREATE INDEX idx\_guide\_habit\_habit ON guide\_habit(habit\_id);

DROP INDEX IF EXISTS idx\_routine\_habit\_order ON routine\_habit;

CREATE INDEX idx\_routine\_habit\_order ON routine\_habit (routine\_id, order\_in\_routine, habit\_id);

/\* =========================================================

12) Verificación rápida

========================================================= \*/

SELECT (SELECT COUNT(\*) FROM `user`) AS users\_total,

(SELECT COUNT(\*) FROM habitactivity) AS habits\_total,

(SELECT COUNT(\*) FROM routine) AS routines\_total,

(SELECT COUNT(\*) FROM routine\_habit) AS routine\_habits\_total,

(SELECT COUNT(\*) FROM guide) AS guides\_total,

(SELECT COUNT(\*) FROM guide\_habit) AS guide\_habits\_total,

(SELECT COUNT(\*) FROM reminder) AS reminders\_total,

(SELECT COUNT(\*) FROM favorite\_habit) AS favorites\_total,

(SELECT COUNT(\*) FROM completedactivity) AS completed\_total;

**Antes de hacer consultas se ocupa el token de admin**

mutation {

login(email: "admin@demo.com", password: "Secreta12")

}

Las consultas con el admin:

HabitActivities:

query {

habitActivitiesPage(page: {page: 0, size: 10, sort: ["name,asc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

id

name

category

description

duration

targetTime

notes

# si tu DTO expone isFavorite en camelCase, agrégalo:

# isFavorite

}

}

}

CompletedActivities (500k → usa páginas pequeñas)

query {

completedActivitiesPage(page: {page: 0, size: 5, sort: ["date,desc","id,desc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

id

userId

routineId

habitId

date

completedAt

notes

}

}

}

Routines:

query {

routinesPage(page: {page: 0, size: 5, sort: ["title,asc","id,asc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

id

title

userId

daysOfWeek

}

}

}

RoutineHabits:

query {

routineHabitsPage(page: {page: 0, size: 10, sort: ["routineId,asc","orderInRoutine,asc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

routineId

habitId

orderInRoutine

targetTimeInRoutine

notes

}

}

}

GuideHabits:

query {

guideHabitsPage(page: {page: 0, size: 10, sort: ["guideId,asc","habitId,asc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

guideId

habitId

}

}

}

Reminders

query {

remindersPage(page: {page: 0, size: 10, sort: ["userId,asc","habitId,asc","id,asc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

id

userId

habitId

time

frequency

}

}

}

Favorite Habits:

query {

favoriteHabitsPage(page: {page: 0, size: 10, sort: ["userId,asc","habitId,asc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

id

userId

habitId

}

}

}

Guides:

query {

guidesPage(page: {page: 0, size: 10, sort: ["title,asc","id,asc"]}) {

pageInfo { totalElements totalPages page size numberOfElements hasNext hasPrevious }

content {

id

title

content

category

userId

}

}

}

Contadores resumen:

query {

countUsers

countHabitActivities

countGuides

countCompletedActivities

countRoutines

countReminders

countFavoriteHabits

countRoutineHabits

countGuideHabits

}

Como admin para revisar un coach específico:  
  
  
query {

coacheesPage(coachId: 2, page:{page:0, size:10, sort:["id,asc"]}) {

pageInfo { totalElements page size }

content { id name email assignedCoachId }

}

}