DDL

-- 최종 PostgreSQL 스크립트: 테이블 생성 및 연동된 더미 데이터 삽입

-- =====================================================================

-- 주의사항:

-- 1. \*\*기존 테이블\*\*: 이 스크립트는 'users' 테이블과 'user\_tokens' 테이블이 \*\*이미 존재한다\*\*고 가정합니다.

-- 2. \*\*users.id 타입\*\*: 기존 'users' 테이블의 기본 키(PK)인 'id' 컬럼 타입이 \*\*BIGINT\*\* 라고 가정합니다. (UUID가 아님)

-- 3. \*\*실제 사용자 ID 필요\*\*: 아래 `DO $$... END$$` 블록 내에 \*\*자리 표시자(placeholder)\*\*로 되어 있는 사용자 ID (`v\_admin\_user\_id` 등)를

-- \*\*실제로 존재하는 `users` 테이블의 `user\_id` 값으로 반드시 수정\*\*해야 합니다. 수정하지 않으면 사용자 조회 시 오류가 발생합니다.

-- 4. \*\*UUID 생성\*\*: 새로 생성하는 테이블들의 ID(PK)는 UUID 타입을 사용하며 `gen\_random\_uuid()` 함수로 기본값을 설정합니다.

-- - PostgreSQL 13 버전 이상에서는 이 함수가 내장되어 있어 별도 설정이 필요 없습니다.

-- - PostgreSQL 12 버전 이하에서는 `gen\_random\_uuid()` 함수가 작동하지 않을 수 있습니다. 이 경우,

-- a) Azure Portal 등에서 `uuid-ossp` 확장을 허용 목록에 추가하고, 스크립트 상단에 `CREATE EXTENSION IF NOT EXISTS "uuid-ossp";` 를 추가한 뒤, 모든 `gen\_random\_uuid()` 를 `uuid\_generate\_v4()` 로 변경하거나,

-- b) 테이블 ID 타입을 `BIGSERIAL` 등으로 변경하거나, 모든 UUID 값을 직접 하드코딩해야 합니다.

-- 5. \*\*실행 환경\*\*: DBeaver 또는 유사한 SQL 클라이언트에서 '스크립트 실행' 기능을 사용하여 전체 스크립트를 실행하는 것이 좋습니다.

-- =====================================================================

-- #######################################

-- ## TABLE DEFINITIONS ##

-- ## (users, user\_tokens 제외) ##

-- #######################################

-- Courses Table

**CREATE** **TABLE** courses (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

title **VARCHAR**(255) **NOT** **NULL**,

description **TEXT**,

category **VARCHAR**(100),

start\_date **DATE**,

end\_date **DATE**,

capacity **INTEGER**,

**location** **VARCHAR**(255),

status **VARCHAR**(50) **DEFAULT** **'draft'**

);

-- Course Instructors Junction Table

**CREATE** **TABLE** course\_instructors (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

course\_id **UUID** **NOT** **NULL** **REFERENCES** courses(id) **ON** **DELETE** **CASCADE**,

instructor\_id **BIGINT** **NOT** **NULL** **REFERENCES** users(id) **ON** **DELETE** **RESTRICT**, -- users.id가 BIGINT이므로 BIGINT

**UNIQUE** (course\_id, instructor\_id)

);

-- Enrollments Table

**CREATE** **TABLE** enrollments (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

course\_id **UUID** **NOT** **NULL** **REFERENCES** courses(id) **ON** **DELETE** **CASCADE**,

user\_id **BIGINT** **NOT** **NULL** **REFERENCES** users(id) **ON** **DELETE** **CASCADE**, -- users.id가 BIGINT이므로 BIGINT

enroll\_date **DATE** **DEFAULT** **CURRENT\_DATE**,

status **VARCHAR**(50) **DEFAULT** **'enrolled'**,

cancel\_date **DATE**,

**UNIQUE** (course\_id, user\_id)

);

-- Learning Progress Table

**CREATE** **TABLE** learning\_progress (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

enrollment\_id **UUID** **NOT** **NULL** **REFERENCES** enrollments(id) **ON** **DELETE** **CASCADE**,

content\_type **VARCHAR**(50),

content\_id **UUID**,

progress\_rate **REAL** **CHECK** (progress\_rate >= 0 **AND** progress\_rate <= 1),

last\_accessed\_at **TIMESTAMP** **WITH** **TIME** **ZONE**

);

-- Course Materials Table

**CREATE** **TABLE** course\_materials (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

course\_id **UUID** **NOT** **NULL** **REFERENCES** courses(id) **ON** **DELETE** **CASCADE**,

title **VARCHAR**(255) **NOT** **NULL**,

file\_path **VARCHAR**(512) **NOT** **NULL**,

uploaded\_by **BIGINT** **REFERENCES** users(id) **ON** **DELETE** **SET** **NULL** -- users.id가 BIGINT이므로 BIGINT

);

-- Assessments Table

**CREATE** **TABLE** assessments (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

course\_id **UUID** **NOT** **NULL** **REFERENCES** courses(id) **ON** **DELETE** **CASCADE**,

**type** **VARCHAR**(50),

title **VARCHAR**(255) **NOT** **NULL**,

description **TEXT**,

open\_date **DATE**,

close\_date **DATE**

);

-- Assessment Questions Table

**CREATE** **TABLE** assessment\_questions (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

assessment\_id **UUID** **NOT** **NULL** **REFERENCES** assessments(id) **ON** **DELETE** **CASCADE**,

question\_type **VARCHAR**(50),

question\_text **TEXT** **NOT** **NULL**,

**options** **TEXT**,

correct\_answer **TEXT**

);

-- Assessment Answers Table

**CREATE** **TABLE** assessment\_answers (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

assessment\_id **UUID** **NOT** **NULL** **REFERENCES** assessments(id) **ON** **DELETE** **CASCADE**,

user\_id **BIGINT** **NOT** **NULL** **REFERENCES** users(id) **ON** **DELETE** **CASCADE**, -- users.id가 BIGINT이므로 BIGINT

question\_id **UUID** **NOT** **NULL** **REFERENCES** assessment\_questions(id) **ON** **DELETE** **CASCADE**,

answer **TEXT**,

score **REAL**,

**UNIQUE** (assessment\_id, user\_id, question\_id)

);

-- Certificates Table

**CREATE** **TABLE** certificates (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

enrollment\_id **UUID** **NOT** **NULL** **REFERENCES** enrollments(id) **ON** **DELETE** **CASCADE**,

issued\_at **TIMESTAMP** **WITH** **TIME** **ZONE** **DEFAULT** **CURRENT\_TIMESTAMP**,

certificate\_url **VARCHAR**(512)

);

-- Notices Table

**CREATE** **TABLE** notices (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

title **VARCHAR**(255) **NOT** **NULL**,

**content** **TEXT** **NOT** **NULL**,

pinned **BOOLEAN** **DEFAULT** **FALSE**,

writer\_id **BIGINT** **REFERENCES** users(id) **ON** **DELETE** **SET** **NULL**, -- users.id가 BIGINT이므로 BIGINT

created\_at **TIMESTAMP** **WITH** **TIME** **ZONE** **DEFAULT** **CURRENT\_TIMESTAMP**

);

-- Comments Table

**CREATE** **TABLE** **comments** (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

notice\_id **UUID** **NOT** **NULL** **REFERENCES** notices(id) **ON** **DELETE** **CASCADE**,

user\_id **BIGINT** **NOT** **NULL** **REFERENCES** users(id) **ON** **DELETE** **CASCADE**, -- users.id가 BIGINT이므로 BIGINT

**content** **TEXT** **NOT** **NULL**,

created\_at **TIMESTAMP** **WITH** **TIME** **ZONE** **DEFAULT** **CURRENT\_TIMESTAMP**

);

-- Stats Logs Table

**CREATE** **TABLE** stats\_logs (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

**type** **VARCHAR**(100) **NOT** **NULL**,

target\_id **UUID**,

executed\_by **BIGINT** **REFERENCES** users(id) **ON** **DELETE** **SET** **NULL**, -- users.id가 BIGINT이므로 BIGINT

executed\_at **TIMESTAMP** **WITH** **TIME** **ZONE** **DEFAULT** **CURRENT\_TIMESTAMP**,

details **JSONB**

);

-- System Codes Table

**CREATE** **TABLE** system\_codes (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

code\_type **VARCHAR**(50) **NOT** **NULL**,

code\_key **VARCHAR**(100) **NOT** **NULL**,

code\_value **VARCHAR**(255) **NOT** **NULL**,

description **TEXT**,

is\_active **BOOLEAN** **DEFAULT** **TRUE**,

sort\_order **INTEGER** **DEFAULT** 0,

**UNIQUE** (code\_type, code\_key)

);

-- Payments Table

**CREATE** **TABLE** payments (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

user\_id **BIGINT** **NOT** **NULL** **REFERENCES** users(id) **ON** **DELETE** **RESTRICT**, -- users.id가 BIGINT이므로 BIGINT

course\_id **UUID** **NOT** **NULL** **REFERENCES** courses(id) **ON** **DELETE** **RESTRICT**,

enrollment\_id **UUID** **REFERENCES** enrollments(id) **ON** **DELETE** **SET** **NULL**,

amount **INTEGER** **NOT** **NULL** **CHECK** (amount >= 0),

paid\_at **TIMESTAMP** **WITH** **TIME** **ZONE** **DEFAULT** **CURRENT\_TIMESTAMP**,

status **VARCHAR**(50) **NOT** **NULL**,

payment\_method **VARCHAR**(50),

transaction\_id **VARCHAR**(255) **UNIQUE**

);

-- Refunds Table

**CREATE** **TABLE** refunds (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

payment\_id **UUID** **NOT** **NULL** **REFERENCES** payments(id) **ON** **DELETE** **RESTRICT**,

refund\_amount **INTEGER** **NOT** **NULL** **CHECK** (refund\_amount > 0),

refund\_date **TIMESTAMP** **WITH** **TIME** **ZONE** **DEFAULT** **CURRENT\_TIMESTAMP**,

status **VARCHAR**(50) **NOT** **NULL**,

reason **TEXT**

);

-- Files Table

**CREATE** **TABLE** files (

id **UUID** **PRIMARY** **KEY** **DEFAULT** gen\_random\_uuid(),

file\_name **VARCHAR**(255) **NOT** **NULL**,

file\_path **VARCHAR**(512) **NOT** **NULL** **UNIQUE**,

mime\_type **VARCHAR**(100),

file\_size\_bytes **BIGINT**,

uploaded\_by **BIGINT** **REFERENCES** users(id) **ON** **DELETE** **SET** **NULL**, -- users.id가 BIGINT이므로 BIGINT

created\_at **TIMESTAMP** **WITH** **TIME** **ZONE** **DEFAULT** ***CURRENT\_TIMESTAMP***

);

-- Optional: Add indexes

**CREATE** **INDEX** idx\_enrollments\_user\_course **ON** enrollments(user\_id, course\_id);

**CREATE** **INDEX** idx\_course\_instructors\_course **ON** course\_instructors(course\_id);

**CREATE** **INDEX** idx\_payments\_user **ON** payments(user\_id);

**CREATE** **INDEX** idx\_payments\_status **ON** payments(status);

**CREATE** **INDEX** idx\_system\_codes\_type\_key **ON** system\_codes(code\_type, code\_key);

DML

**DO** **$$**

**DECLARE**

-- 사용할 사용자 ID 변수 (BIGINT 타입)

v\_admin\_id **BIGINT**;

v\_user\_id **BIGINT**; -- 일반 사용자는 한 명만 생성

-- 사용할 사용자 정보 (ID/Email 중복 주의)

v\_admin\_user\_id **TEXT** := **'dummy\_admin'**;

v\_admin\_email **TEXT** := **'dummy\_admin@example.com'**;

v\_user\_user\_id **TEXT** := **'dummy\_user'**;

v\_user\_email **TEXT** := **'dummy\_user@example.com'**;

-- 임시 비밀번호 (실제 사용 시 주의)

v\_dummy\_password **TEXT** := **'password123'**; -- <<< 임시 비밀번호, 필요시 변경

-- 다른 테이블 ID 변수 (UUID 타입)

course1\_id **UUID**; course2\_id **UUID**; course3\_id **UUID**; course4\_id **UUID**; course5\_id **UUID**;

enrollment1\_s1\_c1 **UUID**; enrollment2\_s2\_c1 **UUID**; enrollment3\_s1\_c2 **UUID**; enrollment4\_s3\_c2 **UUID**; enrollment5\_s2\_c3 **UUID**;

material1\_c1 **UUID**; material2\_c1 **UUID**; material3\_c2 **UUID**; material4\_c3 **UUID**; material5\_c4 **UUID**;

assessment1\_c1 **UUID**; assessment2\_c2 **UUID**; assessment3\_c3 **UUID**;

question1\_a1 **UUID**; question2\_a1 **UUID**; question3\_a2 **UUID**; question4\_a2 **UUID**; question5\_a3 **UUID**;

notice1\_id **UUID**; notice2\_id **UUID**; notice3\_id **UUID**; notice4\_id **UUID**; notice5\_id **UUID**;

payment1\_s1\_c1 **UUID**; payment2\_s2\_c1 **UUID**; payment3\_s1\_c2 **UUID**; payment4\_s3\_c2 **UUID**; payment5\_s2\_c3 **UUID**;

file1\_id **UUID**; file2\_id **UUID**; file3\_id **UUID**; file4\_id **UUID**; file5\_id **UUID**;

**BEGIN**

-- ============================================================================

-- ====== 단계 0: 더미 사용자 생성 (password 컬럼 추가) =======================

-- ============================================================================

-- users 테이블에 password 컬럼이 NOT NULL 이므로 값을 추가합니다.

-- !!주의!!: 여기서 넣는 비밀번호는 일반 텍스트일 수 있습니다.

-- 실제 애플리케이션이 해시된 비밀번호를 사용한다면, 이 사용자는 앱 로그인이 안될 수 있습니다.

-- ============================================================================

**INSERT** **INTO** users (user\_id, **name**, **role**, email, **password**) **VALUES** -- <<< password 컬럼 추가

(v\_admin\_user\_id, **'Dummy Admin'**, **'ADMIN'**, v\_admin\_email, v\_dummy\_password) -- <<< 임시 비밀번호 값 추가

**ON** **CONFLICT** (user\_id) **DO** **NOTHING**

**RETURNING** id **INTO** v\_admin\_id;

**INSERT** **INTO** users (user\_id, **name**, **role**, email, **password**) **VALUES** -- <<< password 컬럼 추가

(v\_user\_user\_id, **'Dummy User'**, **'STUDENT'**, v\_user\_email, v\_dummy\_password) -- <<< 임시 비밀번호 값 추가

**ON** **CONFLICT** (user\_id) **DO** **NOTHING**

**RETURNING** id **INTO** v\_user\_id;

-- 만약 ON CONFLICT로 인해 ID가 안들어왔다면, 기존 ID를 조회 시도

**IF** v\_admin\_id **IS** **NULL** **THEN**

**SELECT** id **INTO** v\_admin\_id **FROM** users **WHERE** user\_id = v\_admin\_user\_id;

**END** **IF**;

**IF** v\_user\_id **IS** **NULL** **THEN**

**SELECT** id **INTO** v\_user\_id **FROM** users **WHERE** user\_id = v\_user\_user\_id;

**END** **IF**;

-- 필수 사용자 생성/조회 확인

**IF** v\_admin\_id **IS** **NULL** **OR** v\_user\_id **IS** **NULL** **THEN**

**RAISE** **EXCEPTION** **'Failed to create or find required dummy users (admin: %, user: %). Check INSERT permissions or existing user conflicts.'**,

v\_admin\_user\_id, v\_user\_user\_id;

**END** **IF**;

-- 1. Insert System Codes (기존과 동일)

**INSERT** **INTO** system\_codes (id, code\_type, code\_key, code\_value, description, is\_active) **VALUES**

(gen\_random\_uuid(), **'USER\_ROLE'**, **'ADMIN'**, **'Administrator'**, **'System Administrator'**, **true**),

(gen\_random\_uuid(), **'USER\_ROLE'**, **'INSTRUCTOR'**, **'Instructor'**, **'Course Instructor'**, **true**),

(gen\_random\_uuid(), **'USER\_ROLE'**, **'STUDENT'**, **'Student'**, **'Course Participant'**, **true**),

(gen\_random\_uuid(), **'COURSE\_CATEGORY'**, **'TECHNOLOGY'**, **'Technology'**, **'Tech related courses'**, **true**),

(gen\_random\_uuid(), **'COURSE\_CATEGORY'**, **'BUSINESS'**, **'Business'**, **'Business related courses'**, **true**),

(gen\_random\_uuid(), **'COURSE\_CATEGORY'**, **'ARTS'**, **'Arts'**, **'Art related courses'**, **true**),

(gen\_random\_uuid(), **'ENROLLMENT\_STATUS'**, **'ENROLLED'**, **'Enrolled'**, **'Actively enrolled'**, **true**),

(gen\_random\_uuid(), **'ENROLLMENT\_STATUS'**, **'COMPLETED'**, **'Completed'**, **'Successfully completed course'**, **true**),

(gen\_random\_uuid(), **'ENROLLMENT\_STATUS'**, **'CANCELLED'**, **'Cancelled'**, **'Enrollment cancelled'**, **true**),

(gen\_random\_uuid(), **'PAYMENT\_STATUS'**, **'COMPLETED'**, **'Completed'**, **'Payment successful'**, **true**),

(gen\_random\_uuid(), **'PAYMENT\_STATUS'**, **'PENDING'**, **'Pending'**, **'Payment awaiting confirmation'**, **true**),

(gen\_random\_uuid(), **'PAYMENT\_STATUS'**, **'FAILED'**, **'Failed'**, **'Payment failed'**, **true**),

(gen\_random\_uuid(), **'PAYMENT\_STATUS'**, **'REFUNDED'**, **'Refunded'**, **'Payment refunded'**, **true**),

(gen\_random\_uuid(), **'REFUND\_STATUS'**, **'COMPLETED'**, **'Completed'**, **'Refund successful'**, **true**),

(gen\_random\_uuid(), **'REFUND\_STATUS'**, **'PENDING'**, **'Pending'**, **'Refund processing'**, **true**)

**ON** **CONFLICT** (code\_type, code\_key) **DO** **NOTHING**;

-- 3. Insert Courses (기존과 동일)

**INSERT** **INTO** courses (id, title, description, category, start\_date, end\_date, capacity, **location**, status) **VALUES**

(gen\_random\_uuid(), **'React 기초'**, **'React 프레임워크 기본 개념과 실습'**, **'TECHNOLOGY'**, **CURRENT\_DATE** + 10, **CURRENT\_DATE** + 40, 50, **'Online'**, **'published'**) **RETURNING** id **INTO** course1\_id;

**INSERT** **INTO** courses (id, title, description, category, start\_date, end\_date, capacity, **location**, status) **VALUES**

(gen\_random\_uuid(), **'데이터베이스 설계'**, **'관계형 데이터베이스 모델링 및 설계'**, **'TECHNOLOGY'**, **CURRENT\_DATE** + 15, **CURRENT\_DATE** + 45, 40, **'Online'**, **'published'**) **RETURNING** id **INTO** course2\_id;

**INSERT** **INTO** courses (id, title, description, category, start\_date, end\_date, capacity, **location**, status) **VALUES**

(gen\_random\_uuid(), **'마케팅 전략 입문'**, **'기초 마케팅 이론과 전략 수립 방법'**, **'BUSINESS'**, **CURRENT\_DATE** + 20, **CURRENT\_DATE** + 50, 60, **'본사 대강의실'**, **'published'**) **RETURNING** id **INTO** course3\_id;

**INSERT** **INTO** courses (id, title, description, category, start\_date, end\_date, capacity, **location**, status) **VALUES**

(gen\_random\_uuid(), **'디지털 드로잉 기초'**, **'타블렛을 이용한 디지털 드로잉 기법'**, **'ARTS'**, **CURRENT\_DATE** + 25, **CURRENT\_DATE** + 55, 30, **'미술 스튜디오 A'**, **'draft'**) **RETURNING** id **INTO** course4\_id;

**INSERT** **INTO** courses (id, title, description, category, start\_date, end\_date, capacity, **location**, status) **VALUES**

(gen\_random\_uuid(), **'Python 중급 프로그래밍'**, **'Python 심화 문법 및 라이브러리 활용'**, **'TECHNOLOGY'**, **CURRENT\_DATE** + 30, **CURRENT\_DATE** + 60, 50, **'Online'**, **'archived'**) **RETURNING** id **INTO** course5\_id;

-- 4. Insert Files (uploaded\_by는 v\_admin\_id 사용 - 기존과 동일)

**INSERT** **INTO** files (id, file\_name, file\_path, mime\_type, file\_size\_bytes, created\_at, uploaded\_by) **VALUES**

(gen\_random\_uuid(), **'react\_intro.pdf'**, **'/files/c1/react\_intro.pdf'**, **'application/pdf'**, 1024000, **NOW**() - **interval** **'5 day'**, v\_admin\_id) **RETURNING** id **INTO** file1\_id;

**INSERT** **INTO** files (id, file\_name, file\_path, mime\_type, file\_size\_bytes, created\_at, uploaded\_by) **VALUES**

(gen\_random\_uuid(), **'db\_design\_notes.docx'**, **'/files/c2/db\_design\_notes.docx'**, **'application/vnd.openxmlformats-officedocument.wordprocessingml.document'**, 51200, **NOW**() - **interval** **'4 day'**, v\_admin\_id) **RETURNING** id **INTO** file2\_id;

**INSERT** **INTO** files (id, file\_name, file\_path, mime\_type, file\_size\_bytes, created\_at, uploaded\_by) **VALUES**

(gen\_random\_uuid(), **'marketing\_case\_study.zip'**, **'/files/c3/marketing\_cases.zip'**, **'application/zip'**, 5120000, **NOW**() - **interval** **'3 day'**, v\_admin\_id) **RETURNING** id **INTO** file3\_id;

**INSERT** **INTO** files (id, file\_name, file\_path, mime\_type, file\_size\_bytes, created\_at, uploaded\_by) **VALUES**

(gen\_random\_uuid(), **'drawing\_brushes.abr'**, **'/files/c4/drawing\_brushes.abr'**, **'application/octet-stream'**, 204800, **NOW**() - **interval** **'2 day'**, v\_admin\_id) **RETURNING** id **INTO** file4\_id;

**INSERT** **INTO** files (id, file\_name, file\_path, mime\_type, file\_size\_bytes, created\_at, uploaded\_by) **VALUES**

(gen\_random\_uuid(), **'admin\_avatar.png'**, **'/files/users/admin\_avatar.png'**, **'image/png'**, 51200, **NOW**() - **interval** **'6 day'**, v\_admin\_id) **RETURNING** id **INTO** file5\_id;

-- 5. Insert Notices (writer\_id는 v\_admin\_id 사용 - 기존과 동일)

**INSERT** **INTO** notices (id, title, **content**, pinned, created\_at, writer\_id) **VALUES**

(gen\_random\_uuid(), **'플랫폼 점검 안내 (5/1 02:00 ~ 04:00)'**, **'서비스 개선을 위한 시스템 점검 예정입니다.'**, **true**, **NOW**() - **interval** **'3 day'**, v\_admin\_id) **RETURNING** id **INTO** notice1\_id;

**INSERT** **INTO** notices (id, title, **content**, pinned, created\_at, writer\_id) **VALUES**

(gen\_random\_uuid(), **'React 과정 Q&A 세션 안내'**, **'React 과정 관련 질문 답변 시간을 갖습니다.'**, **false**, **NOW**() - **interval** **'2 day'**, v\_admin\_id) **RETURNING** id **INTO** notice2\_id;

**INSERT** **INTO** notices (id, title, **content**, pinned, created\_at, writer\_id) **VALUES**

(gen\_random\_uuid(), **'신규 마케팅 강좌 오픈!'**, **'마케팅 전략 입문 강좌가 개설되었습니다.'**, **false**, **NOW**() - **interval** **'1 day'**, v\_admin\_id) **RETURNING** id **INTO** notice3\_id;

**INSERT** **INTO** notices (id, title, **content**, pinned, created\_at, writer\_id) **VALUES**

(gen\_random\_uuid(), **'학습 자료 업데이트 알림 (DB 설계)'**, **'데이터베이스 설계 강의 자료가 업데이트 되었습니다.'**, **false**, **NOW**(), v\_admin\_id) **RETURNING** id **INTO** notice4\_id;

**INSERT** **INTO** notices (id, title, **content**, pinned, created\_at, writer\_id) **VALUES**

(gen\_random\_uuid(), **'여름 특강 수요 조사'**, **'여름방학 특강 개설을 위한 설문을 진행합니다.'**, **false**, **NOW**() - **interval** **'5 day'**, v\_admin\_id) **RETURNING** id **INTO** notice5\_id;

-- 7. Insert Course Instructors (instructor\_id는 v\_admin\_id 사용 - 기존과 동일)

**INSERT** **INTO** course\_instructors (id, course\_id, instructor\_id) **VALUES**

(gen\_random\_uuid(), course1\_id, v\_admin\_id),

(gen\_random\_uuid(), course2\_id, v\_admin\_id),

(gen\_random\_uuid(), course3\_id, v\_admin\_id),

(gen\_random\_uuid(), course4\_id, v\_admin\_id),

(gen\_random\_uuid(), course5\_id, v\_admin\_id);

-- 8. Insert Enrollments (user\_id는 v\_user\_id 사용 - 기존과 동일)

**INSERT** **INTO** enrollments (id, course\_id, user\_id, enroll\_date, status, cancel\_date) **VALUES**

(gen\_random\_uuid(), course1\_id, v\_user\_id, **CURRENT\_DATE** - 5, **'ENROLLED'**, **NULL**) **RETURNING** id **INTO** enrollment1\_s1\_c1;

**INSERT** **INTO** enrollments (id, course\_id, user\_id, enroll\_date, status, cancel\_date) **VALUES**

(gen\_random\_uuid(), course2\_id, v\_user\_id, **CURRENT\_DATE** - 4, **'ENROLLED'**, **NULL**) **RETURNING** id **INTO** enrollment2\_s2\_c1;

**INSERT** **INTO** enrollments (id, course\_id, user\_id, enroll\_date, status, cancel\_date) **VALUES**

(gen\_random\_uuid(), course3\_id, v\_user\_id, **CURRENT\_DATE** - 3, **'CANCELLED'**, **CURRENT\_DATE** -1) **RETURNING** id **INTO** enrollment3\_s1\_c2;

**INSERT** **INTO** enrollments (id, course\_id, user\_id, enroll\_date, status, cancel\_date) **VALUES**

(gen\_random\_uuid(), course4\_id, v\_user\_id, **CURRENT\_DATE** - 2, **'ENROLLED'**, **NULL**) **RETURNING** id **INTO** enrollment4\_s3\_c2;

**INSERT** **INTO** enrollments (id, course\_id, user\_id, enroll\_date, status, cancel\_date) **VALUES**

(gen\_random\_uuid(), course5\_id, v\_user\_id, **CURRENT\_DATE** - 1, **'COMPLETED'**, **NULL**) **RETURNING** id **INTO** enrollment5\_s2\_c3;

-- 9. Insert Payments (user\_id는 v\_user\_id 사용 - 기존과 동일)

**INSERT** **INTO** payments (id, user\_id, course\_id, enrollment\_id, amount, paid\_at, status, payment\_method, transaction\_id) **VALUES**

(gen\_random\_uuid(), v\_user\_id, course1\_id, enrollment1\_s1\_c1, 100000, **NOW**() - **interval** **'5 day'**, **'COMPLETED'**, **'credit\_card'**, **'txn\_'** || **substr**(**md5**(**random**()::**text**), 1, 10)) **RETURNING** id **INTO** payment1\_s1\_c1;

**INSERT** **INTO** payments (id, user\_id, course\_id, enrollment\_id, amount, paid\_at, status, payment\_method, transaction\_id) **VALUES**

(gen\_random\_uuid(), v\_user\_id, course2\_id, enrollment2\_s2\_c1, 150000, **NOW**() - **interval** **'4 day'**, **'COMPLETED'**, **'bank\_transfer'**, **'txn\_'** || **substr**(**md5**(**random**()::**text**), 1, 10)) **RETURNING** id **INTO** payment2\_s2\_c1;

**INSERT** **INTO** payments (id, user\_id, course\_id, enrollment\_id, amount, paid\_at, status, payment\_method, transaction\_id) **VALUES**

(gen\_random\_uuid(), v\_user\_id, course3\_id, enrollment3\_s1\_c2, 120000, **NOW**() - **interval** **'3 day'**, **'REFUNDED'**, **'credit\_card'**, **'txn\_'** || **substr**(**md5**(**random**()::**text**), 1, 10)) **RETURNING** id **INTO** payment3\_s1\_c2;

**INSERT** **INTO** payments (id, user\_id, course\_id, enrollment\_id, amount, paid\_at, status, payment\_method, transaction\_id) **VALUES**

(gen\_random\_uuid(), v\_user\_id, course4\_id, enrollment4\_s3\_c2, 80000, **NOW**() - **interval** **'2 day'**, **'COMPLETED'**, **'credit\_card'**, **'txn\_'** || **substr**(**md5**(**random**()::**text**), 1, 10)) **RETURNING** id **INTO** payment4\_s3\_c2;

**INSERT** **INTO** payments (id, user\_id, course\_id, enrollment\_id, amount, paid\_at, status, payment\_method, transaction\_id) **VALUES**

(gen\_random\_uuid(), v\_user\_id, course5\_id, enrollment5\_s2\_c3, 200000, **NOW**() - **interval** **'1 day'**, **'COMPLETED'**, **'bank\_transfer'**, **'txn\_'** || **substr**(**md5**(**random**()::**text**), 1, 10)) **RETURNING** id **INTO** payment5\_s2\_c3;

-- 10. Insert Refunds (기존과 동일)

**INSERT** **INTO** refunds (id, payment\_id, refund\_amount, refund\_date, status, reason) **VALUES**

(gen\_random\_uuid(), payment3\_s1\_c2, 120000, **NOW**() - **interval** **'1 day'**, **'COMPLETED'**, **'User cancelled enrollment'**);

-- 11. Insert Course Materials (uploaded\_by는 v\_admin\_id 사용 - 기존과 동일)

**INSERT** **INTO** course\_materials (id, course\_id, title, file\_path, uploaded\_by) **VALUES**

(gen\_random\_uuid(), course1\_id, **'React 소개 (PDF)'**, (**SELECT** file\_path **FROM** files **WHERE** id = file1\_id), v\_admin\_id) **RETURNING** id **INTO** material1\_c1;

**INSERT** **INTO** course\_materials (id, course\_id, title, file\_path, uploaded\_by) **VALUES**

(gen\_random\_uuid(), course1\_id, **'컴포넌트 실습 코드'**, **'/materials/c1/components.zip'**, v\_admin\_id) **RETURNING** id **INTO** material2\_c1;

**INSERT** **INTO** course\_materials (id, course\_id, title, file\_path, uploaded\_by) **VALUES**

(gen\_random\_uuid(), course2\_id, **'ERD 설계 가이드'**, (**SELECT** file\_path **FROM** files **WHERE** id = file2\_id), v\_admin\_id) **RETURNING** id **INTO** material3\_c2;

**INSERT** **INTO** course\_materials (id, course\_id, title, file\_path, uploaded\_by) **VALUES**

(gen\_random\_uuid(), course3\_id, **'마케팅 사례 분석'**, (**SELECT** file\_path **FROM** files **WHERE** id = file3\_id), v\_admin\_id) **RETURNING** id **INTO** material4\_c3;

**INSERT** **INTO** course\_materials (id, course\_id, title, file\_path, uploaded\_by) **VALUES**

(gen\_random\_uuid(), course4\_id, **'추천 브러쉬 세트'**, (**SELECT** file\_path **FROM** files **WHERE** id = file4\_id), v\_admin\_id) **RETURNING** id **INTO** material5\_c4;

-- 12. Insert Assessments (기존과 동일)

**INSERT** **INTO** assessments (id, course\_id, **type**, title, description, open\_date, close\_date) **VALUES**

(gen\_random\_uuid(), course1\_id, **'quiz'**, **'React 기본 퀴즈'**, **'1-3주차 내용 확인 퀴즈'**, **CURRENT\_DATE** + 20, **CURRENT\_DATE** + 27) **RETURNING** id **INTO** assessment1\_c1;

**INSERT** **INTO** assessments (id, course\_id, **type**, title, description, open\_date, close\_date) **VALUES**

(gen\_random\_uuid(), course2\_id, **'assignment'**, **'DB 모델링 과제'**, **'요구사항 기반 ERD 설계 과제 제출'**, **CURRENT\_DATE** + 25, **CURRENT\_DATE** + 35) **RETURNING** id **INTO** assessment2\_c2;

**INSERT** **INTO** assessments (id, course\_id, **type**, title, description, open\_date, close\_date) **VALUES**

(gen\_random\_uuid(), course3\_id, **'quiz'**, **'마케팅 용어 퀴즈'**, **'주요 마케팅 용어 확인'**, **CURRENT\_DATE** + 30, **CURRENT\_DATE** + 37) **RETURNING** id **INTO** assessment3\_c3;

-- 13. Insert Assessment Questions (기존과 동일)

**INSERT** **INTO** assessment\_questions (id, assessment\_id, question\_type, question\_text, **options**, correct\_answer) **VALUES**

(gen\_random\_uuid(), assessment1\_c1, **'multiple\_choice'**, **'React에서 state를 변경하는 기본 함수는?'**, **'["setState", "changeState", "updateState", "modifyState"]'**, **'setState'**) **RETURNING** id **INTO** question1\_a1;

**INSERT** **INTO** assessment\_questions (id, assessment\_id, question\_type, question\_text, **options**, correct\_answer) **VALUES**

(gen\_random\_uuid(), assessment1\_c1, **'short\_answer'**, **'JSX는 무엇의 약자인가?'**, **NULL**, **'JavaScript XML'**) **RETURNING** id **INTO** question2\_a1;

**INSERT** **INTO** assessment\_questions (id, assessment\_id, question\_type, question\_text, **options**, correct\_answer) **VALUES**

(gen\_random\_uuid(), assessment2\_c2, **'essay'**, **'정규화(Normalization)의 필요성에 대해 설명하시오.'**, **NULL**, **NULL**) **RETURNING** id **INTO** question3\_a2;

**INSERT** **INTO** assessment\_questions (id, assessment\_id, question\_type, question\_text, **options**, correct\_answer) **VALUES**

(gen\_random\_uuid(), assessment2\_c2, **'multiple\_choice'**, **'기본 키(Primary Key)의 특징이 아닌 것은?'**, **'["Unique", "Not Null", "Can be multiple per table", "Identifies rows"]'**, **'Can be multiple per table'**) **RETURNING** id **INTO** question4\_a2;

**INSERT** **INTO** assessment\_questions (id, assessment\_id, question\_type, question\_text, **options**, correct\_answer) **VALUES**

(gen\_random\_uuid(), assessment3\_c3, **'short\_answer'**, **'마케팅 믹스(4P)의 4가지 요소는 무엇인가?'**, **NULL**, **'Product, Price, Place, Promotion'**) **RETURNING** id **INTO** question5\_a3;

-- 14. Insert Assessment Answers (user\_id는 v\_user\_id 사용 - 기존과 동일)

**INSERT** **INTO** assessment\_answers (id, assessment\_id, user\_id, question\_id, answer, score) **VALUES**

(gen\_random\_uuid(), assessment1\_c1, v\_user\_id, question1\_a1, **'setState'**, 10.0);

**INSERT** **INTO** assessment\_answers (id, assessment\_id, user\_id, question\_id, answer, score) **VALUES**

(gen\_random\_uuid(), assessment1\_c1, v\_user\_id, question2\_a1, **'JavaScript XML'**, 10.0);

**INSERT** **INTO** assessment\_answers (id, assessment\_id, user\_id, question\_id, answer, score) **VALUES**

(gen\_random\_uuid(), assessment2\_c2, v\_user\_id, question3\_a2, **'데이터 중복을 줄이고 무결성을 높이기 위함...'**, 8.0);

**INSERT** **INTO** assessment\_answers (id, assessment\_id, user\_id, question\_id, answer, score) **VALUES**

(gen\_random\_uuid(), assessment2\_c2, v\_user\_id, question4\_a2, **'Can be multiple per table'**, 10.0);

**INSERT** **INTO** assessment\_answers (id, assessment\_id, user\_id, question\_id, answer, score) **VALUES**

(gen\_random\_uuid(), assessment3\_c3, v\_user\_id, question5\_a3, **'Product, Price, Place, Promotion'**, 10.0);

-- 15. Insert Learning Progress (v\_user\_id의 enrollment 참조 - 기존과 동일)

**INSERT** **INTO** learning\_progress (id, enrollment\_id, content\_type, content\_id, progress\_rate, last\_accessed\_at) **VALUES**

(gen\_random\_uuid(), enrollment1\_s1\_c1, **'material'**, material1\_c1, 1.0, **NOW**() - **interval** **'2 day'**);

**INSERT** **INTO** learning\_progress (id, enrollment\_id, content\_type, content\_id, progress\_rate, last\_accessed\_at) **VALUES**

(gen\_random\_uuid(), enrollment1\_s1\_c1, **'material'**, material2\_c1, 0.6, **NOW**() - **interval** **'1 day'**);

**INSERT** **INTO** learning\_progress (id, enrollment\_id, content\_type, content\_id, progress\_rate, last\_accessed\_at) **VALUES**

(gen\_random\_uuid(), enrollment2\_s2\_c1, **'material'**, material3\_c2, 0.9, **NOW**() - **interval** **'1 day'**);

**INSERT** **INTO** learning\_progress (id, enrollment\_id, content\_type, content\_id, progress\_rate, last\_accessed\_at) **VALUES**

(gen\_random\_uuid(), enrollment2\_s2\_c1, **'assessment'**, assessment2\_c2, 0.5, **NOW**() - **interval** **'1 hour'**);

**INSERT** **INTO** learning\_progress (id, enrollment\_id, content\_type, content\_id, progress\_rate, last\_accessed\_at) **VALUES**

(gen\_random\_uuid(), enrollment5\_s2\_c3, **'material'**, **NULL**, 1.0, **NOW**() - **interval** **'10 day'**);

-- 16. Insert Certificates (v\_user\_id의 enrollment 참조 - 기존과 동일)

**INSERT** **INTO** certificates (id, enrollment\_id, issued\_at, certificate\_url) **VALUES**

(gen\_random\_uuid(), enrollment5\_s2\_c3, **NOW**() - **interval** **'9 day'**, **'/certificates/cert\_c5\_dummy\_user.pdf'**);

**INSERT** **INTO** certificates (id, enrollment\_id, issued\_at, certificate\_url) **VALUES**

(gen\_random\_uuid(), enrollment1\_s1\_c1, **NOW**() - **interval** **'1 day'**, **'/certificates/cert\_c1\_dummy\_user.pdf'**);

-- 17. Insert Comments (user\_id는 v\_user\_id 또는 v\_admin\_id 사용 - 기존과 동일)

**INSERT** **INTO** **comments** (id, notice\_id, user\_id, **content**, created\_at) **VALUES**

(gen\_random\_uuid(), notice1\_id, v\_user\_id, **'점검 시간 확인했습니다. 감사합니다.'**, **NOW**() - **interval** **'2 day'**);

**INSERT** **INTO** **comments** (id, notice\_id, user\_id, **content**, created\_at) **VALUES**

(gen\_random\_uuid(), notice2\_id, v\_user\_id, **'Q&A 세션 참여하겠습니다!'**, **NOW**() - **interval** **'1 day'**);

**INSERT** **INTO** **comments** (id, notice\_id, user\_id, **content**, created\_at) **VALUES**

(gen\_random\_uuid(), notice2\_id, v\_admin\_id, **'네, 온라인 Zoom으로 진행될 예정입니다.'**, **NOW**() - **interval** **'1 day'** + **interval** **'1 hour'**);

**INSERT** **INTO** **comments** (id, notice\_id, user\_id, **content**, created\_at) **VALUES**

(gen\_random\_uuid(), notice3\_id, v\_user\_id, **'마케팅 강좌 기대됩니다!'**, **NOW**());

**INSERT** **INTO** **comments** (id, notice\_id, user\_id, **content**, created\_at) **VALUES**

(gen\_random\_uuid(), notice5\_id, v\_user\_id, **'설문 참여했습니다.'**, **NOW**() - **interval** **'4 day'**);

-- 18. Insert Stats Logs (executed\_by는 v\_user\_id 또는 v\_admin\_id 사용 - 기존과 동일)

**INSERT** **INTO** stats\_logs (id, **type**, target\_id, executed\_by, executed\_at, details) **VALUES**

(gen\_random\_uuid(), **'USER\_LOGIN'**, **NULL**, v\_user\_id, **NOW**() - **interval** **'6 hour'**, **jsonb\_build\_object**(**'user\_id'**, v\_user\_id, **'ip'**, **'192.168.0.10'**, **'result'**, **'success'**));

**INSERT** **INTO** stats\_logs (id, **type**, target\_id, executed\_by, executed\_at, details) **VALUES**

(gen\_random\_uuid(), **'ENROLLMENT\_CREATED'**, enrollment1\_s1\_c1, v\_user\_id, **NOW**() - **interval** **'5 day'**, **jsonb\_build\_object**(**'course\_id'**, course1\_id));

**INSERT** **INTO** stats\_logs (id, **type**, target\_id, executed\_by, executed\_at, details) **VALUES**

(gen\_random\_uuid(), **'PAYMENT\_STATUS\_CHANGED'**, payment3\_s1\_c2, v\_admin\_id, **NOW**() - **interval** **'1 day'**, **jsonb\_build\_object**(**'old\_status'**, **'COMPLETED'**, **'new\_status'**, **'REFUNDED'**));

**INSERT** **INTO** stats\_logs (id, **type**, target\_id, executed\_by, executed\_at, details) **VALUES**

(gen\_random\_uuid(), **'COURSE\_STATUS\_CHANGED'**, course4\_id, v\_admin\_id, **NOW**() - **interval** **'1 hour'**, **jsonb\_build\_object**(**'old\_status'**, **'draft'**, **'new\_status'**, **'published'**));

**INSERT** **INTO** stats\_logs (id, **type**, target\_id, executed\_by, executed\_at, details) **VALUES**

(gen\_random\_uuid(), **'NOTICE\_CREATED'**, notice5\_id, v\_admin\_id, **NOW**() - **interval** **'5 day'**, **jsonb\_build\_object**(**'title'**, **'여름 특강 수요 조사'**));

**UPDATE** courses **SET** status = **'published'** **WHERE** id = course4\_id;

**END** **$$**;

-- Verification queries (Optional)

-- (이전 답변의 검증 쿼리 사용 가능)