```sql
BEGIN;

-- ============================================================
-- 1) USERS
--    (admin, organizers, visitors)
-- ============================================================
INSERT INTO app_user (role, full_name, email)
VALUES
  ('admin',     'Basel Kurian',          'basel@asu.edu'),
  ('organizer', 'ASU Engineering Ops', 'eng-ops@asu.edu'),
  ('organizer', 'ASU Arts & Events',   'arts@asu.edu'),
  ('organizer', 'Venture Devils',      'venturedevils@asu.edu'),
  ('visitor',   'Dens Sumesh',         'dens@asu.edu'),
  ('visitor',   'Maadhava Subramanian','maadhava@asu.edu')
ON CONFLICT (email) DO NOTHING;


-- ============================================================
-- 2) ORGANIZERS
--    (map to specific organizer users)
-- ============================================================
INSERT INTO organizer (user_id, org_name, website_url)
SELECT u.user_id, 'ASU Ira A. Fulton Schools of Engineering', 'https://engineering.asu.edu'
FROM app_user u WHERE u.email = 'eng-ops@asu.edu'
ON CONFLICT (user_id) DO NOTHING;

INSERT INTO organizer (user_id, org_name, website_url)
SELECT u.user_id, 'ASU Arts & Events', 'https://arts.asu.edu'
FROM app_user u WHERE u.email = 'arts@asu.edu'
ON CONFLICT (user_id) DO NOTHING;

INSERT INTO organizer (user_id, org_name, website_url)
SELECT u.user_id, 'ASU Venture Devils', 'https://entrepreneurship.asu.edu/venture-devils'
FROM app_user u WHERE u.email = 'venturedevils@asu.edu'
ON CONFLICT (user_id) DO NOTHING;


-- ============================================================
-- 3) VENUES (with PostGIS geography points)
--    lat/lon are stored for convenience; "location" is GEOGRAPHY(Point,4326)
-- ============================================================
-- Memorial Union (MU)
INSERT INTO venue (name, street_address, city, state, postal_code, lat, lon, location)
VALUES ('Memorial Union', '301 E Orange St', 'Tempe', 'AZ', '85287',
       33.421204, -111.933158,
       ST_SetSRID(ST_MakePoint(-111.933158, 33.421204), 4326)::geography);
```

```sql
-- Sun Devil Fitness Complex (SDFC)
INSERT INTO venue (name, street_address, city, state, postal_code, lat, lon, location)
VALUES ('Sun Devil Fitness Complex', '400 E Apache Blvd', 'Tempe', 'AZ', '85281',
    33.416948, -111.932857,
    ST_SetSRID(ST_MakePoint(-111.932857, 33.416948), 4326)::geography);


-- ASU Gammage
INSERT INTO venue (name, street_address, city, state, postal_code, lat, lon, location)
VALUES ('ASU Gammage', '1200 S Forest Ave', 'Tempe', 'AZ', '85281',
    33.414939, -111.938034,
    ST_SetSRID(ST_MakePoint(-111.938034, 33.414939), 4326)::geography);


-- Hayden Library
INSERT INTO venue (name, street_address, city, state, postal_code, lat, lon, location)
VALUES ('Hayden Library', '300 E Orange Mall', 'Tempe', 'AZ', '85281',
    33.419807, -111.935257,
    ST_SetSRID(ST_MakePoint(-111.935257, 33.419807), 4326)::geography);


-- ISTB7 (Interdisciplinary Science & Tech Bldg 7)
INSERT INTO venue (name, street_address, city, state, postal_code, lat, lon, location)
VALUES ('ISTB7', '777 E University Dr', 'Tempe', 'AZ', '85281',
    33.420980, -111.929940,
    ST_SetSRID(ST_MakePoint(-111.929940, 33.420980), 4326)::geography);


-- Tempe Marketplace (off-campus)
INSERT INTO venue (name, street_address, city, state, postal_code, lat, lon, location)
VALUES ('Tempe Marketplace', '2000 E Rio Salado Pkwy', 'Tempe', 'AZ', '85281',
    33.431736, -111.901596,
    ST_SetSRID(ST_MakePoint(-111.901596, 33.431736), 4326)::geography);


-- ============================================================
-- 4) EVENTS
--   Use INSERT…SELECT to attach organizer/venue by name/email
--   Times use Arizona local offset (-07) for clarity.
-- ============================================================

-- A. Tech lecture (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
    'AI for Everyone',
    'Intro talk on practical AI tools for students and non-CS majors.',
    '2025-11-15 17:00-07'::timestamptz, '2025-11-15 18:30-07'::timestamptz,
    'scheduled'
```

```sql
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'eng-ops@asu.edu'
JOIN venue v ON v.name = 'Memorial Union'
RETURNING event_id;

-- B. Entrepreneurship showcase (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
    'Venture Devils Demo Night',
    'Student founders pitch their startups; investor + mentor networking.',
    '2025-11-20 18:00-07'::timestamptz, '2025-11-20 20:30-07'::timestamptz,
    'scheduled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'venturedevils@asu.edu'
JOIN venue v ON v.name = 'ISTB7'
RETURNING event_id;

-- C. Theater performance (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
    'Fall Theatre Gala',
    'An evening of student performances curated by ASU Arts.',
    '2025-11-08 19:00-07'::timestamptz, '2025-11-08 21:30-07'::timestamptz,
    'scheduled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'arts@asu.edu'
JOIN venue v ON v.name = 'ASU Gammage'
RETURNING event_id;

-- D. Library workshop (postponed)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
    'Zotero + Research Strategies',
    'Hands-on workshop on citation managers and literature search.',
    '2025-11-05 15:00-07'::timestamptz, '2025-11-05 16:15-07'::timestamptz,
    'postponed'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'eng-ops@asu.edu'
JOIN venue v ON v.name = 'Hayden Library'
RETURNING event_id;

-- E. Fitness class (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
```

```sql
        'HIIT at SDFC',
        'High-intensity interval training—open to all experience levels.',
        '2025-11-07 07:00-07'::timestamptz, '2025-11-07 07:50-07'::timestamptz,
        'scheduled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'eng-ops@asu.edu'
JOIN venue v ON v.name = 'Sun Devil Fitness Complex'
RETURNING event_id;


-- F. Study jam (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
        'CSE 412 Study Jam',
        'Peer-led exam prep—relational algebra, SQL, and indexing.',
        '2025-11-10 18:00-07'::timestamptz, '2025-11-10 20:00-07'::timestamptz,
        'scheduled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'eng-ops@asu.edu'
JOIN venue v ON v.name = 'Memorial Union'
RETURNING event_id;


-- G. Outdoor concert (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
        'Market Beats Live',
        'Local bands, food trucks, and community booths.',
        '2025-11-22 17:30-07'::timestamptz, '2025-11-22 21:30-07'::timestamptz,
        'scheduled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'arts@asu.edu'
JOIN venue v ON v.name = 'Tempe Marketplace'
RETURNING event_id;


-- H. Hack night (cancelled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
        'Hack Night: Data Pipelines',
        'Build an ETL to ingest ICS/RSS feeds into Postgres.',
        '2025-11-12 19:00-07'::timestamptz, '2025-11-12 22:00-07'::timestamptz,
        'cancelled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'eng-ops@asu.edu'
JOIN venue v ON v.name = 'ISTB7'
RETURNING event_id;
```

```sql
-- I. Founder fireside (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
    'Founder Fireside: Scaling from 0→1',
    'Q&A with alumni founders on MVPs, PMF, and fundraising.',
    '2025-11-29 18:30-07'::timestamptz, '2025-11-29 20:00-07'::timestamptz,
    'scheduled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'venturedevils@asu.edu'
JOIN venue v ON v.name = 'Memorial Union'
RETURNING event_id;

-- J. Gallery opening (scheduled)
INSERT INTO event (organizer_id, venue_id, title, description, start_at, end_at, status)
SELECT o.organizer_id, v.venue_id,
    'Student Art Gallery Opening',
    'Reception + artist talks; refreshments served.',
    '2025-11-03 17:00-07'::timestamptz, '2025-11-03 19:00-07'::timestamptz,
    'scheduled'
FROM organizer o
JOIN app_user u ON u.user_id = o.user_id AND u.email = 'arts@asu.edu'
JOIN venue v ON v.name = 'Hayden Library'
RETURNING event_id;


-- ============================================================
-- 5) EVENT CATEGORIES
--    (attach tags to events by title; add 2–3 per event)
-- ============================================================
-- Helper NOTE: titles are unique in this seed; in real life you'd use IDs.

-- AI for Everyone
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('lecture'),('tech'),('workshop')) AS t(c)
WHERE e.title = 'AI for Everyone'
ON CONFLICT DO NOTHING;

-- Venture Devils Demo Night
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('entrepreneurship'),('pitch'),('networking')) AS
t(c)
WHERE e.title = 'Venture Devils Demo Night'
ON CONFLICT DO NOTHING;
```

```sql
-- Fall Theatre Gala
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('arts'),('theatre'),('performance')) AS t(c)
WHERE e.title = 'Fall Theatre Gala'
ON CONFLICT DO NOTHING;

-- Zotero + Research Strategies
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('academic'),('library'),('workshop')) AS t(c)
WHERE e.title = 'Zotero + Research Strategies'
ON CONFLICT DO NOTHING;

-- HIIT at SDFC
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('fitness'),('health')) AS t(c)
WHERE e.title = 'HIIT at SDFC'
ON CONFLICT DO NOTHING;

-- CSE 412 Study Jam
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('study'),('academic'),('cs')) AS t(c)
WHERE e.title = 'CSE 412 Study Jam'
ON CONFLICT DO NOTHING;

-- Market Beats Live
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('music'),('festival'),('community')) AS t(c)
WHERE e.title = 'Market Beats Live'
ON CONFLICT DO NOTHING;

-- Hack Night: Data Pipelines
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('hackathon'),('tech'),('data')) AS t(c)
WHERE e.title = 'Hack Night: Data Pipelines'
ON CONFLICT DO NOTHING;

-- Founder Fireside: Scaling from 0→1
INSERT INTO event_category (event_id, category)
SELECT e.event_id, c FROM event e, (VALUES ('entrepreneurship'),('talk'),('startup')) AS t(c)
WHERE e.title = 'Founder Fireside: Scaling from 0→1'
ON CONFLICT DO NOTHING;

-- Student Art Gallery Opening
INSERT INTO event_category (event_id, category)
```

```
SELECT e.event_id, c FROM event e, (VALUES ('arts'),('exhibit'),('reception')) AS t(c)
WHERE e.title = 'Student Art Gallery Opening'
ON CONFLICT DO NOTHING;

COMMIT;


-- ========================================================
-- 6) QUICK DEMO QUERIES (optional, nice to run right after)
-- ========================================================
-- Show everything you just added
SELECT user_id, role, full_name, email FROM app_user ORDER BY user_id;
SELECT organizer_id, org_name FROM organizer ORDER BY organizer_id;
SELECT venue_id, name, city, state FROM venue ORDER BY venue_id;
SELECT event_id, title, status, start_at, end_at FROM event ORDER BY start_at;
SELECT e.title, ec.category
FROM event_category ec JOIN event e ON e.event_id = ec.event_id
ORDER BY e.title, ec.category;

-- Example: events within 5km of MU
WITH origin AS (
  SELECT ST_SetSRID(ST_MakePoint(-111.933158, 33.421204),4326)::geography AS g
)
SELECT e.title, v.name AS venue, ROUND(ST_Distance(v.location, o.g)) AS meters_away
FROM event e
JOIN venue v ON v.venue_id = e.venue_id
CROSS JOIN origin o
WHERE v.location IS NOT NULL
  AND ST_DWithin(v.location, o.g, 5000)
ORDER BY meters_away, e.start_at;
```