Sample SQL Queries for Module 7 - Multi-Tenant SaaS in Azure Cosmos DB for PostgreSQL

# 1. Prepare tables for a multitenant data architecture

-- Add tenant\_id to track ownership  
CREATE TABLE orders (  
 order\_id BIGINT,  
 tenant\_id INT,  
 customer\_id BIGINT,  
 amount DECIMAL,  
 order\_date TIMESTAMP,  
 PRIMARY KEY (order\_id, tenant\_id)  
);

# 2. Exercise – Prepare tables for distribution

-- Ensure tenant\_id is part of the primary key for distribution  
ALTER TABLE orders ADD CONSTRAINT pk\_orders PRIMARY KEY (order\_id, tenant\_id);  
-- Index for faster tenant-level access  
CREATE INDEX idx\_orders\_tenant ON orders(tenant\_id);

# 3. Distribute tables with minimal application disruption

-- Make orders a distributed table using tenant\_id  
SELECT create\_distributed\_table('orders', 'tenant\_id');

# 4. Exercise – Distribute multitenant tables

-- Validate distribution metadata  
SELECT \* FROM citus\_tables WHERE table\_name = 'orders';  
-- Confirm shards  
SELECT \* FROM pg\_dist\_shard;

# 5. Monitor tenants in a multitenant database

-- Get row counts per tenant  
SELECT tenant\_id, COUNT(\*) FROM orders GROUP BY tenant\_id;  
  
-- Monitor query stats  
SELECT \* FROM pg\_stat\_user\_tables WHERE relname = 'orders';

# 6. Isolate tenants in a multitenant SaaS database

-- Schema-per-tenant isolation  
CREATE SCHEMA tenant\_101;  
  
CREATE TABLE tenant\_101.orders (  
 order\_id BIGINT PRIMARY KEY,  
 customer\_id BIGINT,  
 amount DECIMAL,  
 order\_date TIMESTAMP  
);

# 7. Exercise – Isolate a tenant in a multitenant database

-- Move tenant data to new schema  
INSERT INTO tenant\_101.orders  
SELECT order\_id, customer\_id, amount, order\_date  
FROM orders  
WHERE tenant\_id = 101;  
  
-- Optionally remove data from shared table  
DELETE FROM orders WHERE tenant\_id = 101;