Design Patters

Database Patterns

| Standalone App | Database per Tenant | Sharded Multi-tenant |

Standalone application pattern
Single tenant application with a single tenant database for each tenant
Each tenant's app is deployed into a separate Azure resource group
Provides the greatest tenant isolation

Database per tenant pattern
Database is created for each venue, or tenant, and all the databases are centrally managed
Databases can be hosted in elastic pools
A catalog database holds the mapping between tenants and their databases. This mapping is managed using the shard map management features of the Elastic Database Client Library

Sharded multi-tenant database pattern
This pattern allows packing large numbers of tenants into a single database
Near infinite scale is possible by sharding the tenants across multiple database
A catalog database maps tenants to databases
The choice can be made on a tenant-by-tenant basis

https://docs.microsoft.com/en-us/azure/sql-database/saas-tenancy-welcome-wingtip-tickets-app

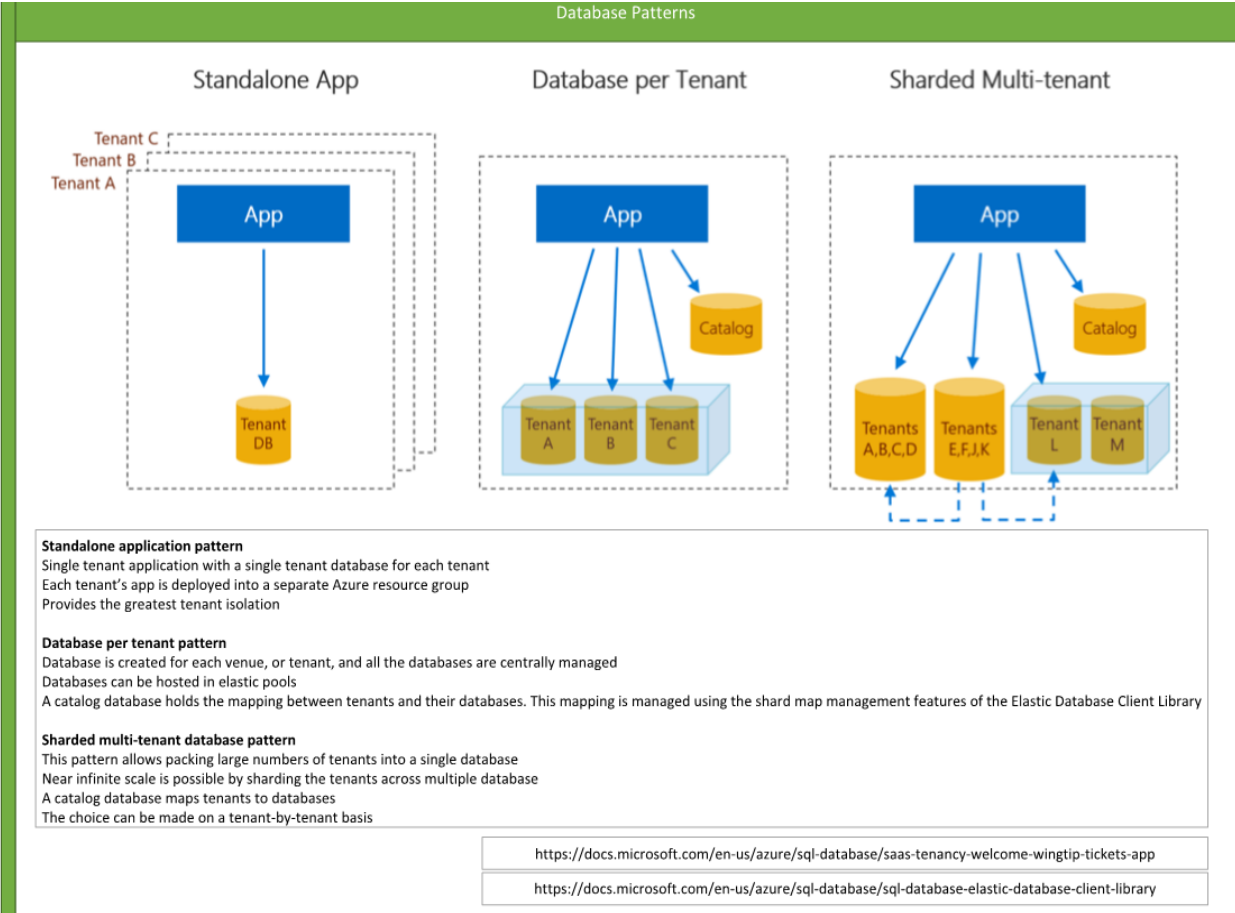https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-database-client-library
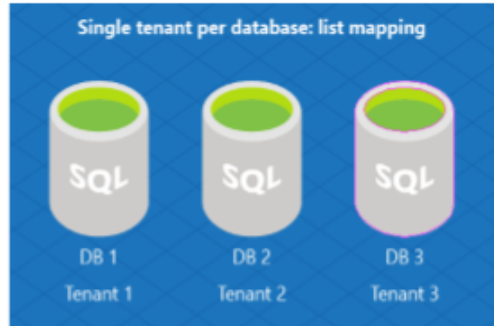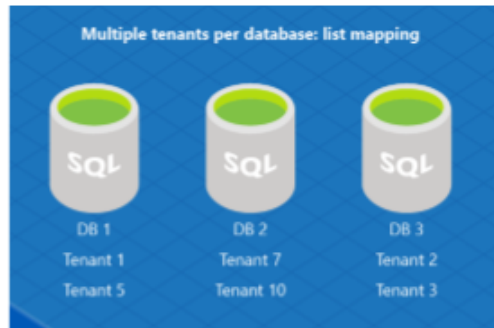
List Shard mappings

List Mapping Shard Maps

List Mapping Shard Patterns

- The single-tenant model assigns one database per tenant.
- This is the simplest model supported and simplifies management.

Single tenant per database: list mapping

- This multi-tenant model uses list mapping to assign multiple tenants to a single database.
- Any number of tenants can be assigned to any database and tenants can easily be moved between databases.
- This model increases management complexity but can provide cost benefits and additional flexibility.

Multiple tenants per database: list mapping

Supported types for sharding keys

Elastic Scale support the following types as sharding keys:

.NET	Java
integer	integer
long	long
guid	uuid
byte[]	byte[]
datetime	timestamp
timespan	duration
datetimeoffset	offsetdatetime

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-scale-shard-map-management

Processes flow

Provisioning Process Flow

Start

If Not Exist - Create Shard Map Manager

If Not Exist - Create List Shard Map

Need New Shard

Add New Database — Yes

No

Add New Shard Mapping

Need New List Mapping — Yes — Add New List Mapping

No

End