3. Build and release pipelines

└ **3.5 Implement deployment solutions**

└ **3.5.3 Database Task Deployments**

---

1. What are the main methods for deploying database changes in Azure DevOps pipelines?
2. How do you configure and use the Azure SQL Database Deployment task?
3. What are the best practices for database schema migrations in CI/CD?
4. How do you manage rollback and recovery for failed database deployments?
5. What security considerations must be addressed for database deployments in pipelines?
6. How do you automate database deployments using DACPACs?
7. How can you validate and test database changes before deployment?
8. What tools are commonly integrated for database deployments in Azure Pipelines?
9. How do you handle environment-specific database settings during deployment?
10. How do you use release gates or approvals for database deployment steps?

---

1. What are the main methods for deploying database changes in Azure DevOps pipelines?

- Use built-in *Azure SQL Database Deployment* tasks,
- SQL scripts,
- DACPAC deployment,
- third-party tools (e.g., Redgate),
- or custom scripts.

The most common approaches are DACPAC and SQL script execution.

---

2. **How do you configure and use the Azure SQL Database Deployment task?**

Add the "*Azure SQL Database Deployment*" task to your pipeline, specify the Azure service connection, target server/database, authentication, and select deployment type (DACPAC, SQL script). Set necessary parameters like SQL user/password or managed identity.

---

3. **What are the best practices for database schema migrations in CI/CD?**

- Version control all database changes,
- use incremental scripts or migration frameworks,
- validate in lower environments,
- run automated tests post-deployment,
- use idempotent scripts,
- and ensure deployment steps are repeatable.

---

4. **How do you manage rollback and recovery for failed database deployments?**

- Implement automated database backups before deployment,
- use transaction blocks in scripts,
- maintain rollback scripts,
- and configure point-in-time restore for Azure SQL.

For DACPAC, keep previous versions for re-deployment.

5. **What security considerations must be addressed for database deployments in pipelines?**
- Store credentials securely (*Azure Key Vault*, pipeline secrets),
- use managed identities or service principals,
- restrict permissions to least privilege,
- and audit all deployment activities.

---

6. **How do you automate database deployments using DACPACs?**
Build the DACPAC artifact from source, publish it as a pipeline artifact, and deploy using the "*Azure SQL Database Deployment*" task with the DACPAC file, specifying connection and deployment options.

---

7. **How can you validate and test database changes before deployment?**
- Run schema validation tasks,
- execute unit and integration tests against a staging database,
- and use automated smoke tests post-deployment.

Use tools like tSQLt or *SQL Server Data Tools* (SSDT) for validation.

---

8. **What tools are commonly integrated for database deployments in Azure Pipelines?**
- *Azure SQL Database Deployment* task,
- *SQL Server Data Tools* (SSDT),
- Redgate SQL Change Automation,
- custom PowerShell or Bash scripts,
- and *Azure Key Vault* for secrets management.

---

9. **How do you handle environment-specific database settings during deployment?**
- Use pipeline variables/groups,
- parameterize scripts or DACPAC publish profiles,
- and maintain separate configuration files or ARM templates per environment.

---

10. **How do you use release gates or approvals for database deployment steps?**
- Configure pipeline environments with manual approval checks,
- set up release gates based on monitoring/alerts or pre/post-deployment conditions,
- and require authorized personnel to approve before production deployments.