**I N T R O D U C T I O N**

1. There is no way to organize full team software development automatization with Oracle, because, at the least:
   1. There is no possibility to deploy separate development Oracle instance at each developer’s PC.
   2. There is no possibility to connect naturally Oracle source base and version control system.
   3. Code is tightly coupled with data.
   4. Git cannot merge PL/SQL code correctly, which is frequent subject for merge conflict.
2. Oracle has Edition Based Redefinition (EBR) feature, but:
   1. It is too complicated and fragile. There is a risk to crush whole database.
   2. It has nothing to do with build pipeline and DevOps at all. It is strictly internal Oracle feature.
3. So, the only way to eliminate chaos is to follow strict protocol and restrictions.
4. Oracle instance **MUST NOT** serve as natural code repository. We **MUST** use files and version control system to manage code.
5. Repository **MUST** have at least 2 branches: **master** and **development**. No need in staging branch, because then there will be needed to sync it with staging environment, which is unnecessary for our case.
6. All SQL files must be collected to the repository in order that is convenient for the team.
7. There **MUST** be a staging environment (pre-production) – exact copy of production. There is a huge amount of work to create rollback strategy both on Oracle and DevOps sides and there is no guarantee that rollback will be successful without human intervention. So, pre-production **MUST BE** exact copy of production to check everything.
8. Liquibase or similar tools are not suitable for Oracle development, because:
   1. There is no team development. Finally, packages are being compiled on the similar development instance for the whole team.
   2. Integrating Liquibase or similar tools create additional amount of work and unstable releases probability.
9. Latest version of Allround Automations’s PL/SQL Developer IDE has something of integrated source and project control, but it has nothing to do with build pipeline. And it is paid.
10. I have researched a tool named Gitora. But:
    1. It is paid - $169 per developer machine.
    2. Some like “handicraft”.
    3. Unstable.

**This is 100% tried and reliable technical and collaborative guide to release software within Oracle environment:**

**- - - P R E P A R A T I O N - - -**

1. Plan release date with the team, do a 15-minute meeting and see which tasks are going to be released.
2. Define:
   1. Responsible for release.
   2. Release participants.
3. Inform DevOps and DBA colleagues about release beforehand. Always be sure that you have DevOps and DBA colleagues phone numbers.
4. Create branch from development branch (as a rule, named like JIRA task number) and checkout.
5. Inform development team about forthcoming changes, indicating what you are going to change. Make sure, that there is no one working with needed objects right now.
6. Make necessary changes on development instance.
7. Send all additional DDLs to the person responsible for release.
8. Commit and merge branch to development branch.
9. Create release branch from master and name it like “release08122020” (if today is 08 December 2020 for example).
10. Team implements tasks to release branch. One task – one commit.
11. Be sure that all release participants have been committed and pushed to release branch everything they need.
12. Each release participant must be online to solve possible problems quickly.
13. In case if table data is being changed:
    1. Create CTAS as backup. Mark it accordingly with prefixes and timestamp in order to delete them later.
    2. Commit only after data has been verified.
    3. Use flashback query if needed. Determine UNDO\_RETENTION from DBA.

**- - - S T A G I N G - - -**

1. All Oracle jobs must be stopped.
2. If (very unlikely) there are some blocking sessions, then DBA assistance will be needed.
3. Run all additional DDLs (if any) sent by release participants.
4. Open each commit in bitbucket on the separate tab.
5. Create the Oracle object list participated in build.
6. Compare these objects’ code in master branch with current Oracle instance code. If there is no object definition in the root of repository, then create its DDL.
7. Pull release branch.
8. Compile code of these objects from release branch on staging environment. You must understand order of compilation.
9. If there are some minor omissions, then fix them hotly and commit directly to the release branch.
10. Check for invalid objects.
11. Test on staging environment.
12. If success, then you have two options (choose the most convenient for you):
    1. Create release patch set (script, containing all previous scripts in the correct order).
    2. Repeat all steps on production.

**- - - P R O D U C T I O N - - -**

1. All external consumers (front-end, web-services, buses etc.) could be turned off through the following 3 options:
   1. Must be shut down, if there is no stand-by.
   2. Standby sync must be shut down (in case of failover).
   3. Must be redirected to standby.
   4. Database could be put into restricted mode, but all consumers will produce errors.
2. All Oracle jobs must be stopped.
3. If (very unlikely) there are some blocking sessions, then DBA assistance will be needed.
4. Create Oracle restore point.
5. Run all additional DDLs (if any) sent by release participants.
6. Open each commit in bitbucket on the separate tab.
7. Create the Oracle object list participated in build.
8. Compare these objects’ code in master branch with current Oracle instance code. If there is no object definition in the root of repository, then create its DDL.
9. Pull release branch.
10. Check for invalid objects.
11. Compile code of these objects from release branch on staging environment. You must understand order of compilation.
12. If there are some minor omissions, then fix them hotly and commit directly to the release branch.
13. All external consumers (front-end, web-services, buses etc.) could be turned on through the following 3 options:
    1. Must be turned on, if there is no stand-by.
    2. Standby sync must be enabled.
    3. Must be redirected to production. Then all steps must be reproduced on standby.
    4. Database could be put into normal mode.
14. Smoke test on production environment.
15. Delete Oracle restore point.
16. Inform team about release results.
17. Merge release branch to master.
18. If release has been declined, do not forget to inform all colleagues about it.
19. Write “Thanks” e-mail to colleagues.