



IDEO·LAB

ORACLE EN ENTRETIEN

PAR IDEO-LAB

MARCH 2026

ORACLE

AGENDA ORACLE

- 2. Conception et implémentation de bases de données
 - A. Modélisation de données
 - B. Scripts de création et d'évolution
 - C. Optimisation SQL dès la conception
- 3. Optimisation des performances
 - A. Démarche générale de tuning
 - B. Indexation
 - C. Plans d'exécution
 - D. Réécriture SQL
 - E. Partitionnement
 - F. Statistiques et optimiseur



Conception et Implémentation de Bases de Données



A. Modélisation de données



Concevoir et organiser les structures de données

B. Scripts de création et d'évolution



Écrire des scripts SQL pour créer et modifier les bases de données

C. Optimisation SQL dès la conception



Optimiser les requêtes SQL dès la phase de conception



AGENDA ORACLE - 2

- 4. Participation à l'administration des environnements
 - A. Gestion des environnements
 - B. Tâches d'administration Oracle typiques
 - C. Sécurité et droits
 - D. Sauvegarde, restauration, PRA
- 5. Contribution au MCO
 - A. Support niveau 3
 - B. Gestion des incidents
 - C. Mise en production
- 6. Focus Oracle
 - A. Pourquoi Oracle est particulier
 - B. Ce qu'il faut connaître sur Oracle
 - C. PL/SQL



Participation à l'Administration des Environnements

4. Participation à l'administration des environnements

- **A. Gestion** des environnements
- **B. Tâches** d'administration Oracle typiques
- **C. Sécurité** et droits
- **D.** Sauvegarde, restauration, PRA



5. Contribution au MCO

- **A. Support** niveau 3
- **B. Gestion** des incidents
- **C.** Mise en production



5. Contribution au MCO

- **A. Support** niveau 3
- **B. Gestion** des incidents
- **C.** Mise en production



6. Focus Oracle

- **A.** Pourquoi Oracle est particulier
- **B.** Ce qu'il faut connaître sur Oracle
- **C.** PL/SQL



AGENDA ORACLE - 3

- 7. SQL Server, MySQL, Oracle
- 8. Questions techniques probables en entretien
- 9. Qualités attendues au-delà de la technique
- 10. Ce que tu dois absolument réviser
- 11. Pièges à éviter en entretien
- 12. Oracle Tools



7. SQL Server, MySQL, Oracle



Comparer et connaître les bases de données

8. Questions techniques probables en entretien



Préparer les questions usuelles d'entretien

9. Qualités attendues au-delà de la technique



Développer les qualités personnelles recherchées

10. Ce que tu dois absolument réviser



Réviser les concepts essentiels pour l'entretien

11. Pièges à éviter en entretien



Identifier les erreurs à éviter lors d'un entretien

12. Oracle Tools



Explorer les outils Oracle pour le développement

INTRO AU POSTE DE DBA

DEMANDES GENERIQUES



- développement base de données
- performance / tuning SQL
- administration Oracle
- support de production / MCO
- mise en production et fiabilisation



- concevoir une base propre et exploitable
- écrire du SQL robuste et performant
- diagnostiquer les lenteurs
- sécuriser les mises en prod
- intervenir quand ça casse en production
- dialoguer avec les développeurs, les ops, parfois les chefs de projet



A. Modélisation de données



A. Modélisation de Données



Modélisation Conceptuelle

Modélisation Logique

Modélisation Physique

2. Conception et implémentation de bases de données

A. Modélisation de données

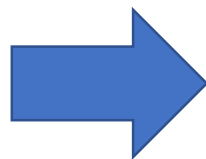
L'objectif ici est de construire un modèle :

- cohérent fonctionnellement
- maintenable
- performant
- évolutif

Ce qu'on attend

Tu dois être à l'aise avec :

- **modèle conceptuel**
- **modèle logique**
- **modèle physique**
- normalisation / dénormalisation
- clés primaires, étrangères, contraintes
- cardinalités
- types de données adaptés
- volumétrie et croissance future



Ce qu'il faut expliquer

Une bonne modélisation ne consiste pas à "faire beaucoup de tables", mais à trouver le bon compromis entre :

- **intégrité**
- **lisibilité**
- **performance**
- **simplicité d'exploitation**

Points importants à citer

Normalisation

La normalisation permet d'éviter :

- la redondance excessive
- les anomalies d'insertion / suppression / mise à jour
- les incohérences de données

Mais il faut aussi savoir dire que, dans la vraie vie :

- une base trop normalisée peut complexifier les requêtes
- certaines dénormalisations sont légitimes pour la performance ou la lecture métier

Choix des types

Sur Oracle, le choix du type a un vrai impact :

- NUMBER
- VARCHAR2
- DATE
- TIMESTAMP
- CLOB
- BLOB

Il faut montrer que tu évites les choix "paresseux" du style tout mettre en VARCHAR2 .

Clés et contraintes

Bien distinguer :

- clé primaire
- clé unique
- clé étrangère
- NOT NULL
- CHECK

En entretien, rappeler que les contraintes ne servent pas qu'à la théorie : elles protègent la qualité des données et évitent de déplacer toute la validation côté application.

B. Scripts de création et d'évolution



B. Scripts de création et d'évolution

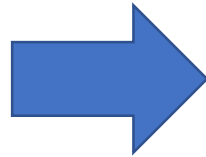


B. Scripts de création et d'évolution

B. Scripts de création et d'évolution

L'implémentation comprend :

- scripts DDL
- scripts DML
- scripts de migration
- scripts de rollback si possible
- gestion de versions



DDL

Exemples :

- CREATE TABLE
- ALTER TABLE
- CREATE INDEX
- CREATE SEQUENCE
- CREATE SYNONYM
- CREATE VIEW

DML

Exemples :

- INSERT
- UPDATE
- DELETE
- MERGE

Ce qu'il faut savoir dire

Un bon script base doit être :

- idempotent si possible
- relisible
- testé
- traçable
- compatible avec la stratégie de déploiement

C. Optimisation SQL dès la conception





Indexation efficace

Choisir les bons index

Requêtes performantes

Écrire des requêtes optimisées

Modèle de données optimisé

Optimiser la structure des tables

Indexation efficace

Requêtes performantes

Modèle de données optimisé

C. Optimisation SQL dès la conception

Très important : l'optimisation ne commence pas quand la prod est lente. Elle commence au moment du design.

Exemples :

- éviter les tables monstrueuses sans stratégie d'indexation
- anticiper les accès principaux
- penser aux volumes
- penser aux traitements batch vs temps réel
- éviter les colonnes inutilement larges
- réfléchir au partitionnement très tôt si besoin

3. Optimisation des performances



A. Démarche générale de tuning



C'est fondamental : ne jamais "tuner à l'aveugle".



A. Démarche générale de tuning

En entretien, il faut montrer une méthode.

Méthode saine

1. Comprendre le besoin fonctionnel
2. Identifier la requête ou le traitement problématique
3. Examiner le plan d'exécution
4. Vérifier volumes, statistiques, index, cardinalités
5. Déterminer le vrai goulot d'étranglement
6. Tester une amélioration
7. Mesurer avant/après

C'est fondamental : ne jamais "tuner à l'aveugle".

Indexation en Oracle : Accélérer les accès tout en maîtrisant les coûts

À quoi sert un index ?



- Colonnes de filtre (WHERE)
- Colonnes de jointure (JOIN)
- Colonnes de tri (ORDER BY)
- Colonnes fréquemment recherchées

Scans complets lents



Accélération



⚠ Erreurs classiques à éviter ⚠

- Indexer toutes les colonnes
- Indexer des colonnes très peu sélectives sans raison
- Multiplier les index composites sans analyser les accès réels
- Oublier l'ordre des colonnes dans un index composite

Les bénéfices

Réduction des lectures



Accélération des recherches sélectives

Amélioration de certaines jointures



Les coûts



Ralentissement des INSERT/UPDATE/DELETE

Consommation d'espace



Maintenance
Risque d'index inutiles ou redondants



ORACLE®

Une bonne phrase :

“Je préfère toujours raisonner à partir du plan d'exécution et des métriques réelles, plutôt que d'ajouter des index ou de réécrire du SQL sans preuve.”

B. Indexation

B. Indexation

L'indexation est souvent le premier levier évoqué en entretien, mais il faut éviter le piège du "plus d'index = plus de performance".

À quoi sert un index

Un index sert à accélérer l'accès aux lignes selon certains critères.

Typiquement utile sur :

- colonnes de filtre (WHERE)
- colonnes de jointure (JOIN)
- colonnes de tri (ORDER BY) dans certains cas
- colonnes impliquées dans des recherches fréquentes

Ce qu'il faut savoir expliquer

Les bénéfices

- réduction des lectures
- accélération des recherches sélectives
- amélioration de certaines jointures

Les coûts

- ralentissement des INSERT/UPDATE/DELETE
- consommation d'espace
- maintenance
- risque d'index inutiles ou redondants

Les erreurs classiques

- indexer toutes les colonnes
- indexer des colonnes très peu sélectives sans raison
- multiplier les index composites sans analyser les accès réels
- oublier l'ordre des colonnes dans un index composite



Index composite

Très important à maîtriser.

Un index composite doit être pensé selon :

- les filtres les plus fréquents
- la sélectivité
- l'ordre d'utilisation dans les requêtes

C. Plans d'exécution

C'est un sujet clé.

Un bon candidat Oracle doit être capable de parler de :

- plan d'exécution
- coût estimé
- accès full scan vs index scan
- nested loop / hash join / sort merge join
- cardinalité estimée
- erreurs d'estimation

Pourquoi le plan est crucial

Parce qu'une requête SQL n'est pas exécutée "comme on l'écrit", mais comme l'optimiseur décide de l'exécuter.

Ce qu'il faut regarder

- type d'accès à la table
- ordre des jointures
- opérations coûteuses
- tri
- agrégations
- volume de lignes prévu vs réel
- présence d'opérations temporaires importantes

Pièges classiques

- statistiques obsolètes
- cardinalité mal estimée
- fonctions sur colonnes indexées
- conversions implicites
- predicates non sargables
- requêtes écrites de manière peu lisible pour l'optimiseur

D. Réécriture SQL

Souvent, le meilleur tuning consiste à réécrire la requête.

Exemples de points à améliorer

- sous-requêtes corrélées coûteuses
- `SELECT *` inutile
- jointures mal pensées
- `DISTINCT` utilisé pour masquer un problème de jointure
- fonctions dans les filtres
- `OR` mal placé
- usage excessif de curseurs ou de traitements ligne à ligne en PL/SQL

Idée forte

En base de données, on cherche autant que possible à raisonner en **mode ensembliste**, pas en traitement ligne par ligne.

Très bon point à placer :

“Je privilégie toujours les traitements ensemblistes. En SQL et en PL/SQL, le row-by-row devient vite un facteur de lenteur.”

E. Partitionnement

Sujet très apprécié sur Oracle.

Le partitionnement devient important quand les volumes grossissent fortement.

Objectifs du partitionnement

- améliorer les performances
- faciliter la maintenance
- réduire la fenêtre d'exploitation
- simplifier l'archivage ou la purge

Types de partitionnement

Selon les cas :

- par plage de dates
- par liste
- par hachage
- composite

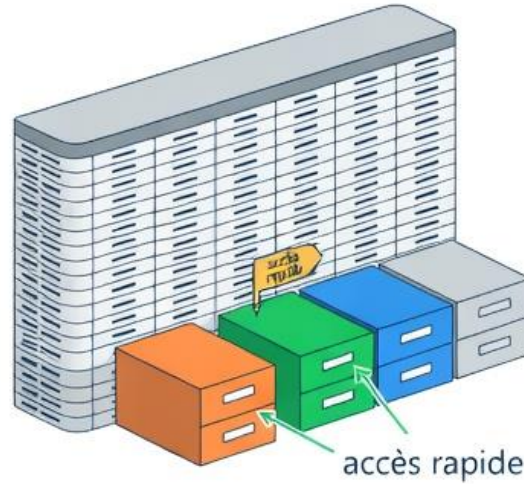
Cas typique

Une table volumineuse historisée partitionnée par mois ou par année sur une date métier ou technique.

Partitionnement en Oracle : Gérer les gros volumes avec performance et simplicité

Objectifs du partitionnement

- ✓ Améliorer les performances 
- ✓ Faciliter la maintenance 
- ✓ Réduire la fenêtre 
- ✓ Réduire d'exploitation 
- ✓ Simplifier l'archivage ou la purge 



- ▶ Sujet très apprécié en entretien Oracle – Devient essentiel quand les volumes grossissent fortement.

Types de partitionnement



Par plage de dates



Par liste



Par hachage



Composite



Cas typique

Une table volumineuse historisée partitionnée par mois ou par année sur une date métier ou technique.

Avantages

- pruning : lecture uniquement des partitions utiles
- maintenance plus simple
- purge plus rapide
- reconstruction ciblée
- meilleure gestion des gros volumes

Ce qu'il faut montrer

Tu dois montrer que tu sais que le partitionnement n'est pas une baguette magique. Il doit être :

- justifié par le volume
- cohérent avec les accès applicatifs
- pensé avec les index et la maintenance

F. Statistiques et optimiseur

Sujet souvent sous-estimé.

Oracle prend ses décisions via l'optimiseur. Si les statistiques sont mauvaises, le plan peut être mauvais.

Il faut savoir parler de

- statistiques de tables
- statistiques d'index
- histogrammes
- dérive de volumétrie
- impact sur les plans

Message important

Une requête peut devenir lente sans qu'on ait changé son code, simplement parce que :

- les volumes ont changé
- la distribution des données a changé
- les statistiques sont obsolètes

Statistiques et Optimiseur : Le cerveau silencieux des performances Oracle

Sujet souvent sous-estimé

Il faut savoir parler de

- Statistiques de tables
- Statistiques d'index
- Histogrammes
- Dérive de volumétrie
- Impact sur les plans d'exécution

Statistiques obsolètes



mauvais plan d'exécution



bon plan d'exécution



Message important

Une requête peut devenir lente sans qu'on ait changé son code, simplement parce que :

- les volumes ont changé
- la distribution des données a changé
- les statistiques sont obsolètes

G. Tuning global de la base

Le tuning ne se limite pas au SQL.

Il peut concerner :

- mémoire
- I/O
- contention
- verrous
- concurrence
- paramètres d'instance
- tablespaces
- redo / undo
- jobs batch trop lourds

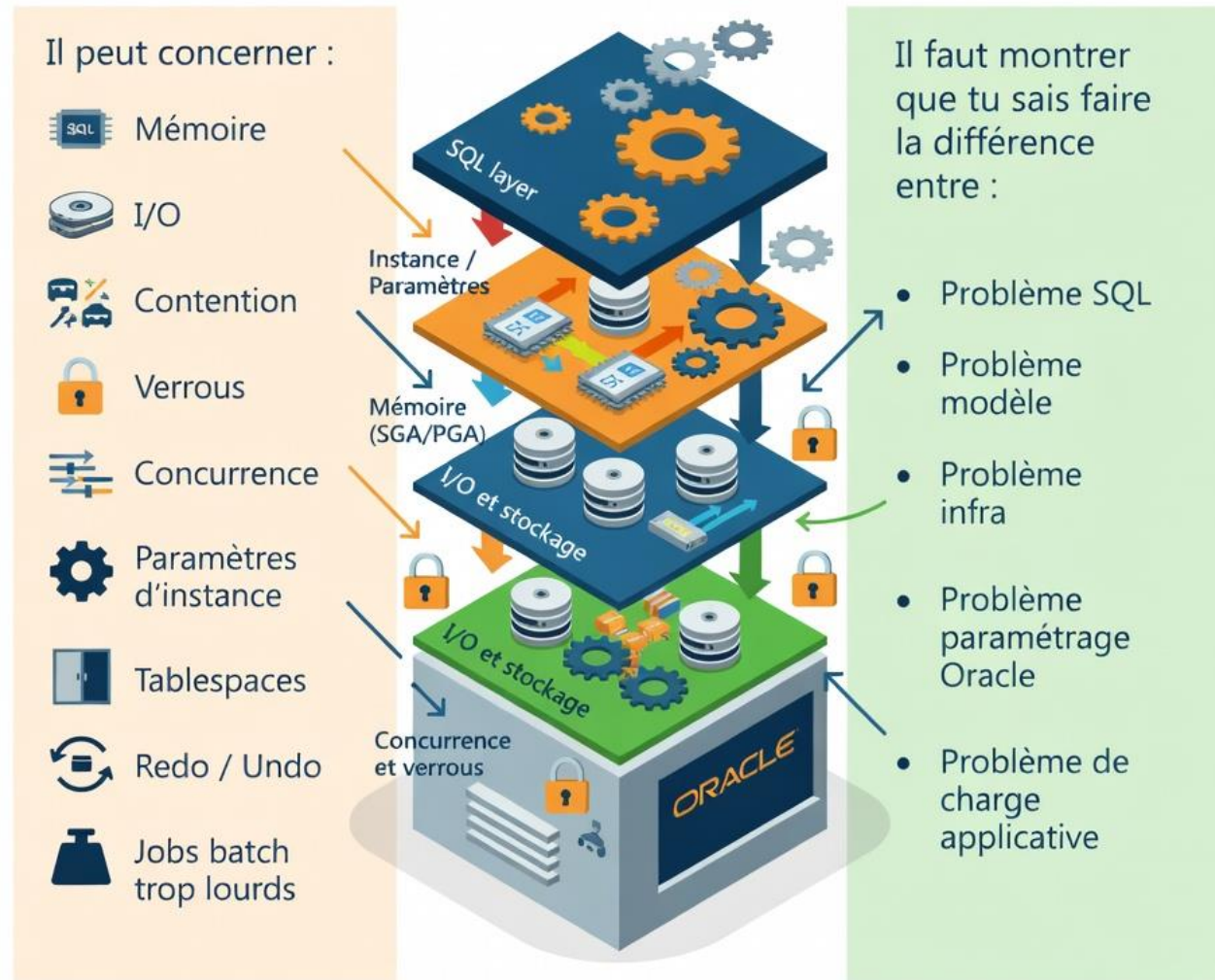
Il faut montrer que tu sais faire la différence entre :

- problème SQL
- problème modèle
- problème infra
- problème paramétrage Oracle
- problème de charge applicative

G. Tuning global de la base : Au-delà du SQL

Le tuning ne se limite pas au SQL.

Le vrai tuning global consiste à identifier la vraie cause, pas seulement le symptôme SQL



4. Participation à l'administration des environnements

A. Gestion des environnements

Tu peux avoir :

- DEV
- TEST
- RECETTE
- PREPROD
- PROD

Il faut savoir parler de :

- cohérence entre environnements
- gestion des jeux de données
- synchronisation de structures
- sécurité des accès
- industrialisation des déploiements

B. Tâches d'administration Oracle typiques

Selon les entreprises, cela peut inclure :

- création de schémas / users / rôles
- gestion des droits
- tablespaces
- surveillance de l'espace disque
- supervision
- jobs Oracle
- sauvegardes / restaurations
- suivi des alertes
- contrôle des performances
- gestion des incidents

Ce qu'il faut savoir montrer

Même si tu n'es pas DBA pur, il faut montrer que tu es à l'aise avec :

- lecture des métriques
- diagnostic technique
- intervention raisonnée
- compréhension des objets Oracle

C. Sécurité et droits

Toujours bon à évoquer.

Exemples :

- séparation des rôles
- principe du moindre privilège
- gestion des comptes applicatifs
- droits sur objets
- audit selon contexte
- sécurisation des accès de prod

Une phrase utile :

“Je fais attention à ne jamais traiter la base comme un simple espace de dev. Les droits, la traçabilité et la séparation des accès sont essentiels, surtout en production.”

D. Sauvegarde, restauration, PRA

Même si ce n'est pas explicitement écrit, c'est souvent dans le périmètre élargi.

Tu peux évoquer :

- stratégie de backup
- tests de restauration
- RPO / RTO
- cohérence applicative
- continuité d'activité






Cela donne une image plus senior.

A. Gestion des environnements Oracle

Tu peux avoir :

-  DEV
-  TEST
-  RECETTE
-  PREPROD
-  PROD

Il faut savoir parler de :

-  Cohérence entre environnements
-  Gestion des jeux de données
-  Synchronisation de structures
-  Sécurité des accès
-  Industrialisation des déploiements





B. Tâches d'administration Oracle typiques

Selon les entreprises, cela peut inclure :

-  Création de schémas / users / rôles
-  Gestion des droits
-  Tablespaces
-  Surveillance de l'espace disque
-  Supervision
-  Jobs Oracle
-  Sauvegardes / restaurations
-  Suivi des alertes
-  Contrôle des performances
-  Gestion des incidents

Ce qu'il faut savoir montrer

Même si tu n'es pas DBA pur, il faut montrer que tu es à l'aise avec :

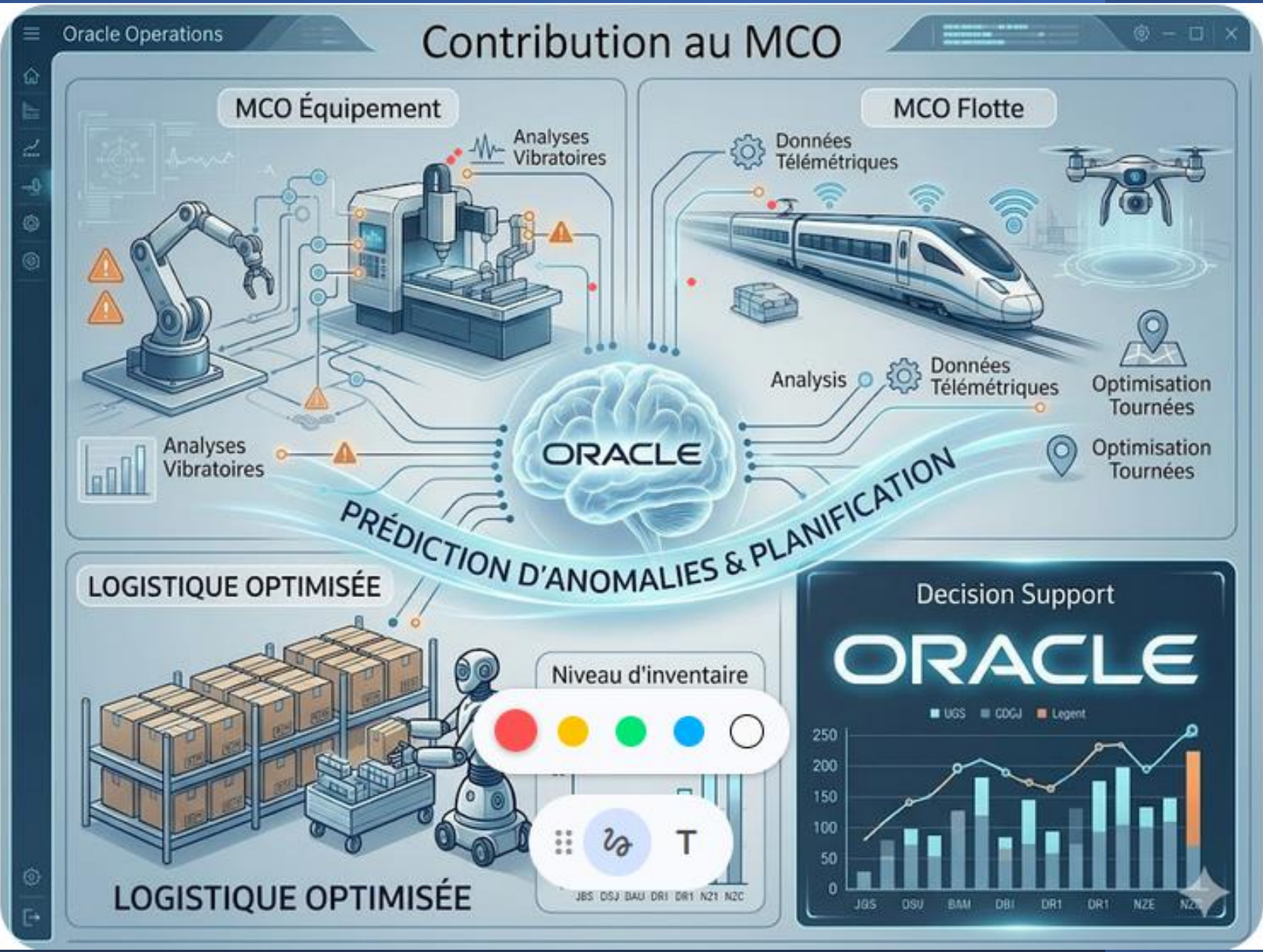
-  Lecture des métriques
-  Diagnostic technique
-  Intervention raisonnée
-  Compréhension des objets Oracle

ORACLE



IDEO-LAB

5. Contribution au MCO



Le MCO = **M**aintien en **C**onditions **O**pérationnelles.

C'est la dimension "run", souvent la plus concrète.

Ils veulent quelqu'un qui ne soit pas seulement bon en projet, mais aussi fiable dans la durée.



A. Support niveau 3

Le support N3 intervient sur les incidents complexes.

Typiquement :

- requêtes très lentes
- blocages
- incidents de mise en prod
- objets invalides
- saturation d'espace
- dérive de performances
- traitements batch en échec
- problèmes PL/SQL
- incidents liés aux données

Ce qu'on attend

- capacité d'analyse
- sang-froid
- méthode
- communication claire
- capacité à aller à la cause racine

Bonne posture à présenter

“En support N3, j’essaie de distinguer le symptôme de la cause. Une lenteur visible côté application peut venir d’une requête, d’un plan modifié, d’un verrou, d’un batch concurrent ou d’un problème d’infrastructure.”

B. Gestion des incidents

Il faut savoir expliquer une démarche type :

1. qualifier l'incident
2. mesurer l'impact
3. sécuriser l'existant
4. identifier la cause
5. proposer un contournement si nécessaire
6. corriger durablement
7. documenter le retour d'expérience

C'est très apprécié en entretien.



C. Mise en production

La mise en prod est un moment critique.

Dans le contexte base de données, cela recouvre souvent :

- exécution de scripts SQL / PL/SQL
- ajout/modification de tables, index, contraintes
- migration de données
- validation post-déploiement
- surveillance après mise en service
- rollback si besoin

Ce qu'il faut mettre en avant

Une mise en prod réussie repose sur :

- préparation
- check-list
- séquençement
- estimation des impacts
- tests en amont
- fenêtre adaptée
- supervision renforcée après déploiement



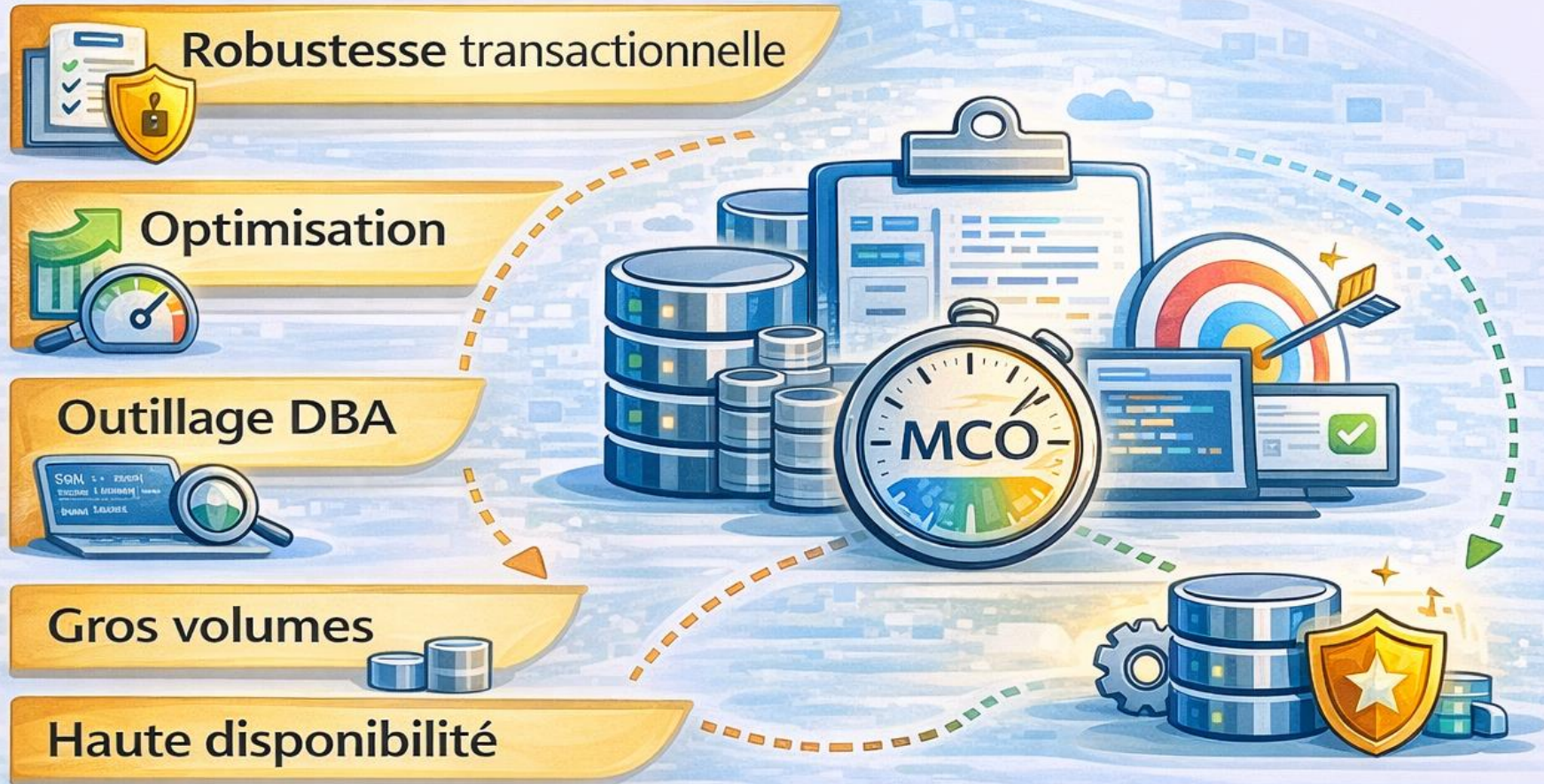
5. Contribution au MCO



6. Focus Oracle

A. Pourquoi Oracle est particulier

Oracle est historiquement très fort sur :



C'est souvent un environnement exigeant, assez structuré, avec des pratiques de prod rigoureuses.

A. Pourquoi Oracle est particulier

Oracle est historiquement très fort sur :

- robustesse transactionnelle
- optimisation
- outillage DBA
- gros volumes
- haute disponibilité
- fonctionnalités avancées

C'est souvent un environnement exigeant, assez structuré, avec des pratiques de prod rigoureuses.

B. Ce qu'il faut connaître sur Oracle

Sans forcément entrer dans le niveau expert extrême, il faut pouvoir parler de :

- schémas
- tablespaces
- index
- vues
- séquences
- packages PL/SQL
- procédures / fonctions
- triggers
- plan d'exécution
- statistiques
- verrous
- transactions
- gestion des gros volumes
- partitionnement



C. PL/SQL

Le PL/SQL est central.

Il faut pouvoir expliquer que PL/SQL sert à :

- encapsuler de la logique côté base
- industrialiser des traitements
- sécuriser certaines opérations
- optimiser certains échanges entre application et base

Objets PL/SQL classiques

- procédures
- fonctions
- packages
- curseurs
- exceptions
- collections
- triggers

Ce qu'il faut montrer

Tu sais écrire du PL/SQL propre, lisible, maintenable, avec :

- gestion d'erreurs
- journalisation
- transactions maîtrisées
- performance correcte

Piège à bien identifier

Le PL/SQL est puissant, mais il faut éviter d'y mettre n'importe quelle logique.

Une bonne réponse d'entretien consiste à dire que tu fais attention à la frontière entre :

- logique métier applicative
- logique data / traitement batch / cohérence transactionnelle

7. SQL Server, MySQL, Oracle : comment en parler intelligemment

Positionnement intelligent

Tu peux dire que les fondamentaux restent les mêmes :



ORACLE

- Très riche fonctionnellement
- Très fort sur gros SI
- PL/SQL
- Partitionnement avancé
- Culture DBA forte

SQL Server

- Écosystème Microsoft
- T-SQL
- Très utilisé en environnement entreprise
- Outillage intégré souvent apprécié

MySQL

- Très répandu
- Simple d'accès
- Fréquent sur applications web
- Nécessite malgré tout une vraie rigueur sur indexation, plans et transactions

Positionnement intelligent

Tu peux dire que les fondamentaux restent les mêmes :

- modélisation
- qualité SQL
- indexation
- plans d'exécution
- transactions
- volumétrie
- concurrence

Mais chaque moteur a ses spécificités :

Oracle

- très riche fonctionnellement
- très fort sur gros SI
- PL/SQL
- partitionnement avancé
- culture DBA forte

SQL Server

- écosystème Microsoft
- T-SQL
- très utilisé en environnement entreprise
- outillage intégré souvent apprécié

MySQL

- très répandu
- simple d'accès
- fréquent sur applications web
- nécessite malgré tout une vraie rigueur sur indexation, plans et transactions

Le bon message :

“Je distingue les concepts universels des spécificités du moteur. Les bonnes pratiques de modélisation, de tuning et d’analyse restent valables partout, mais il faut connaître les particularités d’Oracle, SQL Server ou MySQL pour être réellement efficace.”

8. Questions techniques probables en entretien

“Comment optimisez-vous une requête lente ?”

Ils veulent voir ta méthode.

Réponse attendue :

- contexte fonctionnel
- métriques
- plan d'exécution
- statistiques
- index
- volumétrie
- test avant/après

“Quand créer un index, et quand ne pas en créer ?”

Ils veulent vérifier que tu n'indexes pas au hasard.

“Comment expliquer qu'une requête performante devienne lente sans changement de code ?”

Très bonne question.

Réponse :

- statistiques
- volumétrie
- changement de distribution
- concurrence
- changement de plan
- fragmentation / maintenance / charge système

“Quelle différence entre une clé primaire et un index ?”

Basique mais classique.

La clé primaire impose une règle d'unicité et d'identification ; un index est une structure d'accès. Une PK s'appuie généralement sur un index, mais les deux notions ne sont pas équivalentes.

“Que regardez-vous lors d’une mise en production base ?”

Très bonne question senior.

Réponse :

- dépendances
- ordre d’exécution
- durée estimée
- impact sur données
- lock potentiels
- rollback
- validation post-prod
- supervision

“Comment gérez-vous un incident de production critique ?”

Ils testent ta maturité opérationnelle.



“Comment gérez-vous un incident de production critique ?”

Ils testent ta maturité opérationnelle.

“Quand utiliser du PL/SQL ?”

Réponse nuancée :

- traitements proches de la donnée
- batches
- logique transactionnelle
- mutualisation
- contrôle

Mais éviter de surcharger la base avec de la logique métier mal placée.



9. Qualités attendues au-delà de la technique

A Rigueur

- Sur une base de production, une erreur peut coûter très cher.

B Méthode

- Diagnostic, preuve, test, validation.

C Communication

- Il faut savoir expliquer à :
 - des développeurs
 - des exploitants
 - des chefs de projet
 - parfois des métiers

D Sang-froid

- Support N3 et prod = pression réelle.

ORACLE



E Sens du service

- Le but n'est pas de "faire du SQL élégant", mais de rendre le SI fiable et performant.



A. Rigueur

Sur une base de production, une erreur peut coûter très cher.

B. Méthode

Diagnostic, preuve, test, validation.

C. Communication

Il faut savoir expliquer à :

- des développeurs
- des exploitants
- des chefs de projet
- parfois des métiers

D. Sang-froid

Support N3 et prod = pression réelle.

E. Sens du service

Le but n'est pas de "faire du SQL élégant", mais de rendre le SI fiable et performant.



10. Ce que tu dois absolument réviser avant l'entretien

Oracle / SQL

- plans d'exécution
- index simples et composites
- full scan vs index scan
- nested loop / hash join
- statistiques
- partitionnement
- verrous et transactions
- différences procédure / fonction / package en PL/SQL

Administration / prod

- rôles et droits
- gestion des incidents
- mise en production
- supervision
- support N3
- stratégie de rollback



11. Pièges à éviter en entretien

Ne pas dire

- "J'ajoute un index quand c'est lent"
- "Le SQL, ça dépend surtout du serveur"
- "Le plan d'exécution, je regarde rapidement"
- "Le PL/SQL sert à tout faire"
- "Le support prod, ce n'est pas trop mon sujet"

Dire plutôt

- "Je pars des faits mesurables"
- "J'analyse les accès réels"
- "Je distingue optimisation SQL, optimisation modèle et tuning système"
- "Je sécurise les déploiements"
- "Je suis attentif à l'exploitabilité et à la production"

12. ORACLE TOOLS



Outils du DBA Oracle

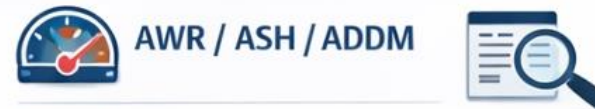
Administration Générale



Outils SQL & Développement



Performance & Tuning



Sauvegarde & Restauration



Sécurité & Audit



Supervision & Monitoring



Haute Disponibilité



Stockage & Système



AGENDA OUTILS AUTOUR DE ORACLE

- 1. Outils natifs Oracle indispensables
- 2. Outils SQL / développement / requêtage
- 3. Outils de performance et tuning
- 4. Outils de sauvegarde, restauration et PRA
- 5. Outils haute disponibilité / réplication
- 6. Outils de supervision et observabilité
- 7. Outils stockage et espace
- 8. Outils sécurité
- 9. Outils migration et comparaison
- 10. Outils système autour d'Oracle



1. Outils natifs Oracle indispensables

Administration générale

- **SQL*Plus**

L'outil historique. Très utilisé pour :

- exécuter du SQL et du PL/SQL
- administrer l'instance
- lancer des scripts DBA
- faire du diagnostic rapide

- **Oracle SQLcl**

Version plus moderne de SQL*Plus, plus confortable :

- autocomplétion
- historique
- export JSON/CSV
- meilleure ergonomie

- **Oracle Enterprise Manager (OEM / Cloud Control)**

Le gros outil central d'administration :

- supervision
- alertes
- performance
- jobs
- sauvegardes RMAN
- suivi de plusieurs bases/serveurs

2. Outils SQL / développement / requêtage

Requêtes et analyse SQL

- Oracle SQL Developer

Très utilisé par DBA et développeurs :

- éditeur SQL
- navigation schéma
- exécution de scripts
- plan d'exécution
- export/import
- interface graphique pratique

- Toad for Oracle

Très populaire dans beaucoup d'entreprises :

- développement SQL / PL/SQL
- tuning SQL
- comparaison de schémas
- administration
- génération de scripts

- PL/SQL Developer

Très apprécié pour :

- PL/SQL
- packages, procédures, fonctions
- debug
- navigation rapide dans les objets



- **Explain Plan / DBMS_XPLAN**

Base absolue du tuning SQL :

- comprendre comment Oracle exécute une requête
- voir index scan, full scan, joins, coût, cardinalité

- **SQL Trace / TKPROF**

Très utile pour l'analyse fine :

- temps CPU
- lectures logiques/physiques
- parsing
- temps d'exécution détaillés

- **SQL Tuning Advisor**

Pour obtenir des recommandations de tuning sur une requête.

- **SQL Monitor / Real-Time SQL Monitoring**

Très pratique sur les grosses requêtes :

- suivi en temps réel
- opérations coûteuses
- parallélisme
- consommation de ressources

3. Outils de performance et tuning

Diagnostic et optimisation

- **AWR (Automatic Workload Repository)**

Indispensable pour analyser :

- charge système
- top SQL
- attentes (wait events)
- CPU / I/O
- évolution des performances

- **ASH (Active Session History)**

Pour voir :

- ce que faisaient les sessions
- les blocages
- les attentes
- les pics de charge

- **ADDM (Automatic Database Diagnostic Monitor)**

Produit des recommandations automatiques à partir d'AWR.

- **Explain Plan / DBMS_XPLAN**

Base absolue du tuning SQL :

- comprendre comment Oracle exécute une requête
- voir index scan, full scan, joins, coût, cardinalité

- **SQL Trace / TKPROF**

Très utile pour l'analyse fine :



4. Outils de sauvegarde, restauration et PRA

Backup / Recovery

- **RMAN (Recovery Manager)**

L'outil majeur de sauvegarde Oracle :

- backup full / incrémental
- restore / recover
- duplication de base
- gestion archivlogs

- **Oracle Data Pump (expdp / impdp)**

Pour :

- exports/imports logiques
- migration de schémas
- copie d'objets
- refresh d'environnements

- **Flashback Technologies**

Très utile pour revenir dans le temps :

- Flashback Query
- Flashback Table
- Flashback Database

5. Outils haute disponibilité / réplication

Continuité de service

- Oracle Data Guard

Outil clé pour :

- standby databases
- reprise après sinistre
- réplication physique/logique
- bascule primaire/secondaire

- Data Guard Broker

Pour piloter plus facilement Data Guard.

- Oracle RAC (Real Application Clusters)

Pour la haute disponibilité et le scale-out :

- plusieurs nœuds
- continuité de service
- répartition de charge

- Oracle GoldenGate

Très utilisé pour :

- réplication temps réel
- migrations avec faible interruption
- alimentation de systèmes avals

6. Outils de supervision et observabilité

Monitoring

- Oracle Enterprise Manager
Encore lui, car c'est le vrai centre de supervision Oracle.
- Scripts maison + vues dynamiques Oracle

Un bon DBA utilise énormément :

- `v$session`
- `v$sql`
- `v$system_event`
- `v$active_session_history`
- `dba_segments`
- `dba_indexes`
- `dba_tablespaces`
- `dba_data_files`
- Nagios / Centreon / Zabbix / Prometheus + Grafana
Souvent utilisés en entreprise pour superviser :
 - disponibilité
 - tablespaces
 - CPU/RAM
 - sessions
 - alert log
 - temps de réponse

7. Outils stockage et espace

Capacity planning

- ASM (Automatic Storage Management)

Pour gérer le stockage Oracle de manière structurée :

- diskgroups
- fichiers de données
- redondance

- OEM + scripts DBA

Pour surveiller :

- tablespaces
- croissance des segments
- fragmentation logique
- volumétrie

8. Outils sécurité

Sécurisation et audit

- **Oracle Wallet**
Pour certificats, secrets, chiffrement, connexions sécurisées.
- **TDE (Transparent Data Encryption)**
Pour chiffrer :
 - tablespaces
 - colonnes sensibles
- **Unified Auditing / Audit Trail**
Pour tracer :
 - connexions
 - actions sensibles
 - opérations DBA
- **Database Vault / Label Security**
Selon les contextes très sensibles.

9. Outils migration et comparaison

Changements de schéma / déploiement

- Liquibase

Très utile pour industrialiser :

- versioning SQL
- déploiements
- changements de schéma

- Flyway

Plus simple, souvent adopté dans les pipelines CI/CD.

- Toad / SQL Developer / outils de compare

Pour :

- comparer schémas
- générer DDL
- synchroniser environnements

10. Outils système autour d'Oracle

Côté OS / infra

Un DBA Oracle solide travaille aussi avec :

- shell/bash
- cron
- systemctl / services
- top / vmstat / iostat / sar
- AWR + métriques OS
- logs système
- scripts Python ou shell d'automatisation

Parce qu'un problème Oracle est souvent aussi :

- CPU
- RAM
- I/O
- réseau
- filesystem
- SAN/NAS
- latence OS

11. Les outils à vraiment maîtriser en priorité

Si tu veux retenir le plus important pour un entretien DBA Oracle, je dirais :

Le socle minimal

- SQL*Plus
- SQLcl
- SQL Developer
- OEM
- RMAN
- AWR / ASH / ADDM
- Explain Plan / DBMS_XPLAN
- TKPROF
- Data Pump
- Data Guard

Le socle expert

- RAC
- ASM
- GoldenGate
- Oracle Net / Listener
- audit / sécurité / TDE
- monitoring système Linux/Unix

11. Les outils à vraiment maîtriser en priorité

Si tu veux retenir le plus important pour un entretien DBA Oracle, je dirais :



L Le socle minimal

- SQL*Plus
- SQLcl
- SQL Developer
- OEM
- RMAN
- AWR / ASH / ADDM
- Explain Plan / DBMS_XPLAN
- TKPROF
- Data Pump
- Data Guard



Le socle expert

- RAC
- ASM
- GoldenGate
- Oracle Net / Listener
- audit / sécurité / TDE
- monitoring système Linux/Unix



Bonne chance pour votre entretien de DBA Oracle !

