



# Monitoratge i Optimització del SGBD





Identificar les eines de monitoratge disponibles per al sistema gestor.

Descriure els avantatges i inconvenients de la creació d'índexs.

Crear índexs en taules i vistes.

Optimitzar l'estructura de la base de dades.

Optimitzar els recursos del sistema gestor.

Obtindre informació sobre el rendiment de les consultes per a la seua optimització.

Programar alertes de rendiment.

Realitzar modificacions en la configuració del sistema operatiu per a millorar el rendiment del gestor.



# Monitoratge

Rendiment

Errors, logs

DD – Diccionari de Dades

# Optimització

Entorn (SO, xarxa)

Sistema Gestor

BD ( índex, consultes )

# Monitoratge

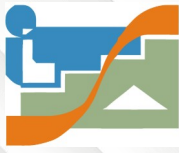
---



S'ha de tractar que les tasques de monitoratge i diagnòstic del sistema siguen el **menys intrusives possible** perquè no penalitzen el rendiment del sistema gestor, i relegar les tasques que requerisquen un consum de recursos mitjà o elevat per a realitzar-les en moments de baixa càrrega.

### Eines principals de monitoratge

- Monitor de rendiment
  - El log d'execució
  - El diccionari de dades



## Monitor de rendiment

- Seguiment de mètriques
- Detecció de bloquejos
- Detecció de processos acaparadors
  - Consum de recursos
- Definició d'alertes o llindars

Transaccions / ut  
Temps de resposta  
Escalabilitat  
Concurrencia





## Registre d'errors -> fitxers LOGs

- Inicis i parades del sistema
  - Operacions rellevants
  - Errors o warnings
- Registre de consultes que tarden molt en acabar

## Nivell de detall dels logs

- Debug
- Warning
- Error



## Diccionari de dades DD

Es pot comprovar:

- Consultes que se estan executant, i consum de recursos
- Qui les executa
- Bloquejos actius i a que sentències afecten
- etc..

Les tasques de monitorització i diagnòstic deuen ser el menys intrusives possible !!

Les ferramentes gràfiques consumeixen més recursos que les consultes al DD

Les ferramentes gràfiques requereixen més permisos que les consultes al DD



# Monitoratge en ORACLE

---



## Monitor de rendiment

- EM Enterprise Manager (deprecated)
  - ✓ Monitor de rendiment
  - ✓ Gestió d'incidències
  - ✓ Definició d'accions correctives
  - ✓ Definició de notificacions
  - ✓ Consulta d'informes predefinit
- Diccionari de Dades
  - Vistes dinàmiques => v\$...... p.exemple v\$sqlarea
- **SQL Developer** ( o altres com TOAD for Oracle, de pagament)
  - ✓ Monitor de rendiment
  - ✓ Historial de SQLs
  - ✓ Informes predefinit

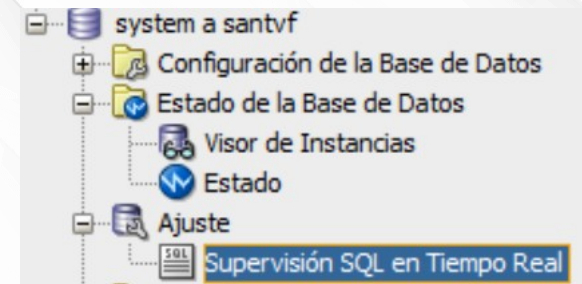
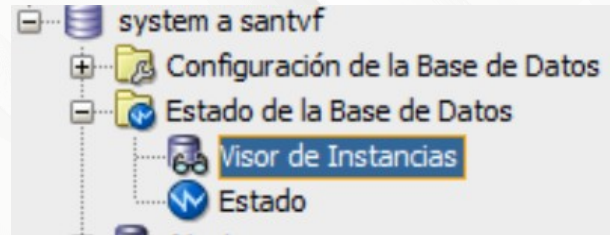
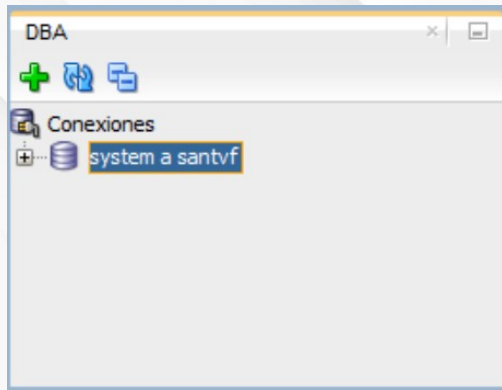


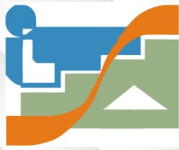
## Monitor de rendiment

- SQL Developer ( o altres com TOAD for Oracle, de pagament)  
Historial de SQLs  
Informes predefinitos

Prova-ho en SQL Developer

Obrir finestra de DBA amb ( Ver- DBA )



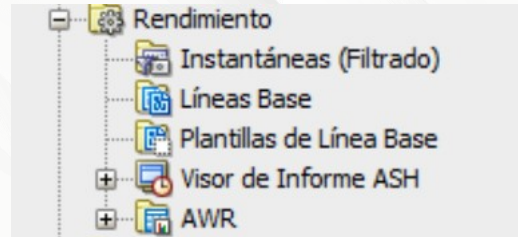
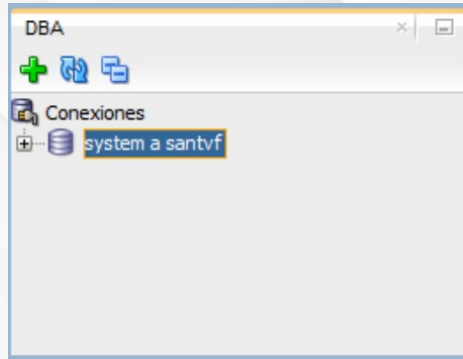


## Monitor de rendiment

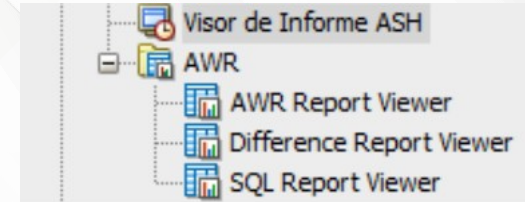
- SQL Developer ( o altres com TOAD for Oracle, de pagament)  
Historial de SQLs  
Informes predefinit

Prova-ho en SQL Developer

Obrir finestra de DBA amb ( Ver- DBA )



### ASH: Active Session History



**AWR: Automatic Workload Repository**



## Monitor de rendiment

- SQL Developer ( o altres com TOAD for Oracle, de pagament)  
Historial de SQLs  
Informes predefinitos

Prova-ho en SQL Developer

Obrir finestra de sessions ( Herramientas – Controlar sesiones )

	INST_ID	SID	SERIAL	SQL_ID	SQL_CHILD_NUMBER	Username	Seconds in Wait	Command
1	1	55	51524	dhc40pc5aj6p9	0	SYSTEM	0	SELECT
2	1	267	38963	(null)	(null)	USUARI2	(null)	(null)
3	1	276	61029	4k27svf3davg0	(null)	SYS	(null)	GRANT OBJECT
4	1	288	9131	cvvys7ublhzy3	0	USUARI1	(null)	SELECT





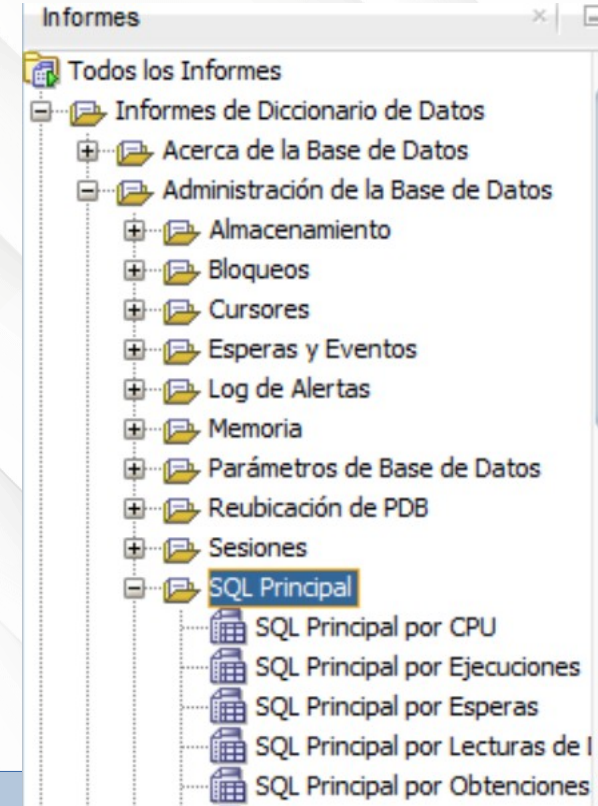
## Monitor de rendiment

- SQL Developer

Monitoritzar Top SQL

Obrir finestra de informes ( Ver – Informes )

Prova-ho en SQL Developer







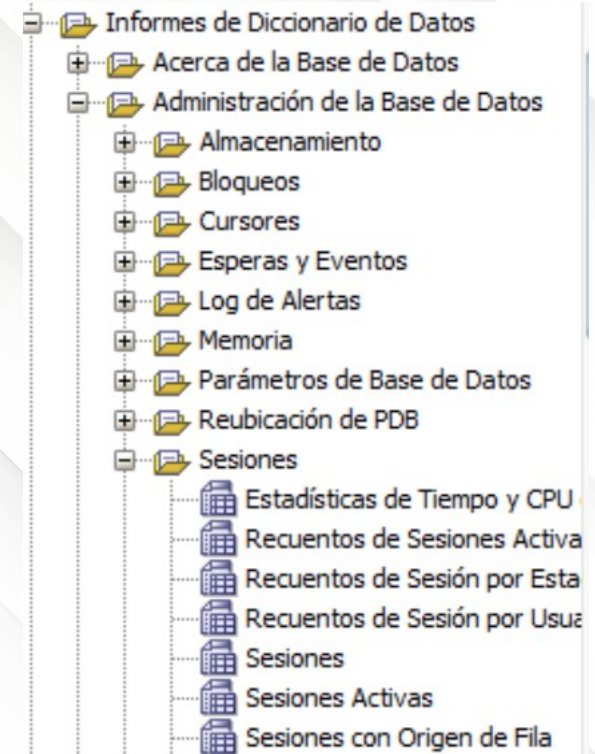
## Monitor de rendiment

- SQL Developer

Monitoritzar Tasques de llarga duració

Obrir finestra de informes ( Ver – Informes - Sesiones)

Prova-ho en SQL Developer





## Registre d'errors

Des de la versió 11 d'Oracle -> ADR Automatic Diagnostic Repository

- ORACLE\_BASE/diag
- Diferents subdirectoris
- vista V\$DIAG\_INFO

alert.log  
cdump  
incident  
trace  
others



Automatic Diagnostic Repository (ADR)



Background  
Trace Files



Foreground  
Trace Files



Dump  
Files

DD:  
V\$DIAG\_INFO



## Registre d'errors

```
select * from v$diag_info;
```

Prova-ho en SQL Developer

NAME	VALUE
Diag Enabled	TRUE
ADR Base	C:\APP\USUARI73
ADR Home	C:\APP\USUARI73\diag\rdbms\safor\safor
Diag Trace	C:\APP\USUARI73\diag\rdbms\safor\safor\trace
Diag Alert	C:\APP\USUARI73\diag\rdbms\safor\safor\alert
Diag Incident	C:\APP\USUARI73\diag\rdbms\safor\safor\incident
Diag Cdump	C:\app\usuari73\diag\rdbms\safor\safor\cdump
Health Monitor	C:\APP\USUARI73\diag\rdbms\safor\safor\hm
Default Trace File	C:\APP\USUARI73\diag\rdbms\safor\safor\trace\safor_ora_188.trc
Active Problem Count	0
Active Incident Count	0
ORACLE_HOME	C:\app\usuari73\product\213000\db_home
Attention Log	C:\APP\USUARI73\diag\rdbms\safor\safor\trace\attention safor.log

# Optimització

---



# Optimització

de l'Entorn

a nivell de SO

a nivell de Xarxa

del SG (Sistema Gestor)

- Grandària de blocs de dades
- Grandària i ubicació de fitxers de dades
- Deshabilitar processos ocults (auto-grow, auto-shrink)  
i executar-los fora d'horari de producció
- Grandària del emmagatzemament temporal





# Optimització

de la BD ( índex, consultes )

- Disseny de taules i tipus de dades (Ajustar al necessari)
- Camps calculats (intentar mantindre els menys possibles)
- Desnormalització (reduir JOIN a costa de augmentar redundància)
- Particionament
- Desfragmentació
- Balanceig d'índex
- Crear, modificar o eliminar índex





# Optimització

## Particionament

- S'evita processar tota una taula (sols es processa la partició)
- Permet guardar en una sola taula més dades que en un disc
- Les dades poden ser accedides en paral·lel
- Facilita operacions com p.e. el purgat de dades



# Optimització

## Creació d'índex

Sobre que columnes crear ( i sobre que columnes NO crear)

Que tipus d'índex crear

- Per organització

  - Agrupats / No agrupats

- Per estructura

  - Índex B-tree

  - Índex bitmap

  - Índex hash



# Optimització

## Creació d'índex

### Sobre que columnes crear

- Claus primaries i alienes ( normalment ja ho fa el SGBD)
- Columnes que habitualment apareixen en SELECT i en WHERE
- Columnes amb bona selectivitat ( si poques files tenen el mateix valor)

### Sobre que columnes NO crear

- Taules amb poques dades
- Columnes amb molts valors NULL
- Columnes amb valors que es modifiquen molt sovint
- Índex sobre moltes columnes
- Molts índex per taula



# Optimització

## Optimització de consultes

- Consulta estadístiques de taules **en el DD**
- Reescriptura de consultes

Actualitzar estadístiques  
periòdicament

- 1- Substituir els OR per UNION
2. IN vs EXISTS
3. IN vs BETWEEN
4. Comparacions. Evitar IS NULL, i <>
5. Usar taules derivades, subconsultes i joins
6. Evitar el GROUP BY
7. Cursors i funcions

### Objectius:

- Evitar recórrer tota la taula si es possible.
- Simplificar i reduir la grandària de les taules abans de fer JOINS



# Optimització

## Ferramentes d'optimització

Basant-se en:

1. Monitor de rendiment
2. Registre d'errors
3. Diccionari de dades
4. Pla d'execució

# Optimització en ORACLE

---





# Optimització del sistema gestor

## Optimització dels objectes de la bbdd

- Fragmentació / Desfragmentació
- Indexació
- Estadístiques
- Particionament

## Optimització de consultes



Paràmetre comú a  
la instància (CDB)

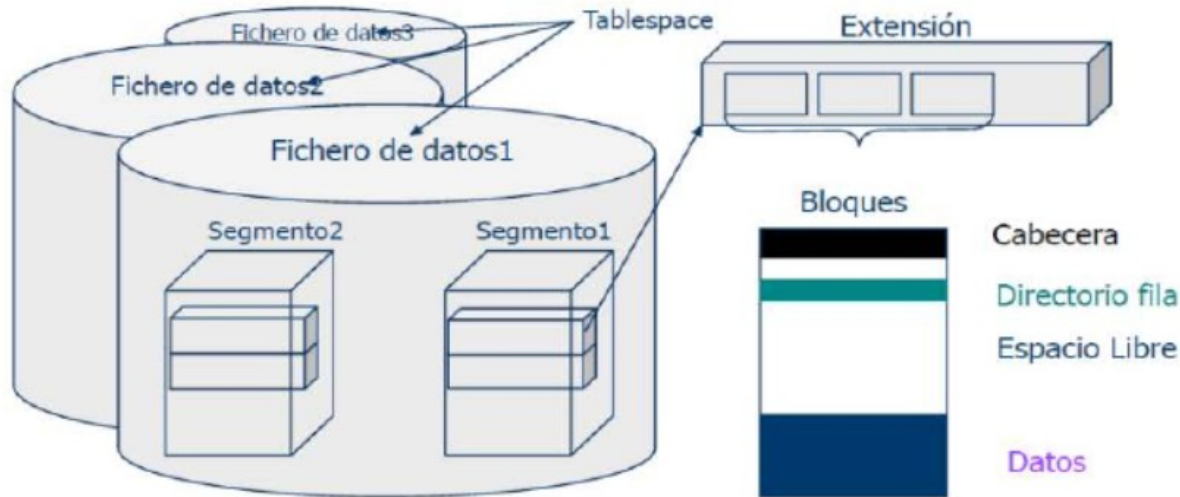
## Optimització del sistema gestor

1. Grandària dels blocs de dades -> paràmetre 'db\_block\_size' del spfile , 8k, 16k en datawarehouse
2. Grandària i ubicació dels fitxers de dades -> tablespaces
3. Deshabilitar processos 'ocults' -> autoextensible dels datafiles
4. Grandària de l'emmagatzemament temporal -> afegir datafile al tablespace TEMP



# Optimització

Blocs – Extensions – Segments – **fixers** – Tablespaces





## Optimització dels objectes de la bbdd

Prova-ho en SQL Developer

### 1. Fragmentació de les taules

```
alter table nom_taula move [compress];
```

### 2. Índex

```
create index nom_ind on nom_taula (camp1, camp2,..);  
alter index nom_ind rebuild;
```

### 3. Actualització d'estadístiques

```
execute dbms_stats.gather_table_stats('esquema','taula');  
execute dbms_stats.gather_schema_stats('esquema');
```

### 4. Particionament

```
create table ....( ) partition by range(nomcamp) (....);
```



## Optimització dels objectes de la bbdd - índex

1. Quan es crea una taula amb una clau primaria o unique, ORACLE crea un índex automàticament

### 2. Tipus Índex

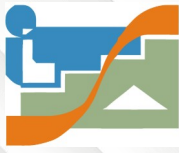
```
create [bitmap | unique] index nom_ind on nom_taula (camp1, camp2,..);  
alter index nom_ind rebuild;
```

- **Com explorar-los** En DD `user_indexes`

```
select index_name , index_type , table_name , tablespace_name , secondary  
from all_indexes where table_name = 'TAULA_A_CONSULTAR';
```

Prova-ho en SQL Developer

DD:  
all\_indexes



## Optimització dels objectes de la bbdd - Particionament

Per exemple imaginem una taula de factures, on tenim el detall de la nostra facturació al llarg de 6 anys, 2017, 2018... 2022, si volguérem fer:

```
SELECT SUM(total_fac) FROM facturacio WHERE any = 2019;
```

En aquest exemple s'hauria de recórrer tota la taula (imaginem que parlem de 30 milions de registres en total, és molt no?), per aquest motiu un criteri possible per a particionar la taula seria per l'any de la data de la factura





# Optimització dels objectes de la bbdd

## Particionament – exemple - ( by range )

```
create table nom_taula (idCOD NUMBER(6), idFECHA DATE)
partition by range( idFECHA )
( partition p_1 values less than (TO_DATE('2006-01-01', 'YYYY-MM-DD')),
  partition p_2 values less than (TO_DATE('2010-01-01', 'YYYY-MM-DD')),
  partition p_3 values less than (TO_DATE('2016-01-01', 'YYYY-MM-DD')),
  partition p_4 values less than (TO_DATE('2020-01-01', 'YYYY-MM-DD')),
  partition p_5 values less than (MAXVALUE) );
```

Prova-ho en SQL Developer

## - Com explorar-les En DD user\_tab\_partitions

```
SELECT table_name, partition_name, high_value
FROM user_tab_partitions WHERE table_name = 'NOM_TAUULA';
```



# Optimització dels objectes de la bbdd

## Particionament – exemple - ( by range ) - especificant tablespace

```
create table nom_taula (idCOD NUMBER(6), idFECHA DATE)
partition by range( idFECHA )
( partition p_1 values less than (TO_DATE('2006-01-01', 'YYYY-MM-DD')) tablespace tab1,
  partition p_2 values less than (TO_DATE('2010-01-01', 'YYYY-MM-DD')) tablespace tab2,
  partition p_3 values less than (TO_DATE('2016-01-01', 'YYYY-MM-DD')) tablespace tab3,
  partition p_4 values less than (TO_DATE('2020-01-01', 'YYYY-MM-DD')) tablespace tab4,
  partition p_5 values less than (MAXVALUE) );
```

Prova-ho en SQL Developer



## Optimització dels objectes de la bbdd

### Particionament – exemples

```
ALTER TABLE facturacion ADD PARTITION (PARTITION p4 VALUES LESS THAN (2010));
```

```
ALTER TABLE facturacion DROP PARTITION p2;
```

```
ALTER TABLE facturacion DROP PARTITION p2 UPDATE GLOBAL INDEX;
```

```
ALTER TABLE facturacion MERGE PARTITION p2 AND p3 INTO PARTITION pnueva;
```

```
ALTER TABLE facturacion SPLIT PARTITION p1 INTO  
(PARTITION p11 VALUES LESS THAN (2006)  
(PARTITION p12 VALUES LESS THAN (MAXVALUE)));
```

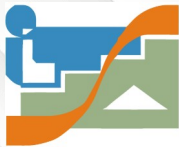
```
ALTER TABLE facturacion TRUNCATE PARTITION p11; (ESBORRA DADES !!)
```



# Optimització dels objectes de la bbdd

## Particionament – exemple ( by list)

```
CREATE TABLE q1_sales_by_region
(deptno number,
deptname varchar2(20),
quarterly_sales number(10, 2),
state varchar2(2))
PARTITION BY LIST (state)
(PARTITION q1_CV VALUES ('VA', 'AL', 'CS'),
PARTITION q1_CA VALUES ('BA', 'TA', 'LL', 'GI'),
PARTITION q1_MU VALUES ('MU'),
PARTITION q1_EU VALUES ('BI', 'SS', 'VI'),
PARTITION q1_GA VALUES ('SC', 'LU', 'VG'),
PARTITION q1_IB VALUES ('MA', 'ME', 'FO' ),
PARTITION q1_nulos VALUES (NULL ),
PARTITION q1_desconegut VALUES (DEFAULT ) );
```



# Optimització dels objectes de la bbdd

Particionament – exemple - passar una taula no particionada a particionada

```
create table nom_taula (idCOD NUMBER(6), idFECHA DATE) ;

alter table nom_taula modify partition by range( idFECHA )
( partition p_1 values less than (TO_DATE('2006-01-01', 'YYYY-MM-DD')),
  partition p_2 values less than (TO_DATE('2010-01-01', 'YYYY-MM-DD')),
  partition p_3 values less than (TO_DATE('2016-01-01', 'YYYY-MM-DD')),
  partition p_4 values less than (TO_DATE('2020-01-01', 'YYYY-MM-DD')),
  partition p_5 values less than (MAXVALUE) );
```

Prova-ho en SQL Developer





## Optimització de consultes

Oracle activa un optimitzador de consultes automàticament i reescriu les consultes si ho estima necessari. Realitza les següents operacions.

1. Avalua expressions i condicions
2. Transforma sentències complexes
3. Transforma vistes en consultes
4. Avalua els JOIN i ordena el accés i la forma d'accés.



Usa estadístiques de taules !!

## Ferramentes d'Optimització Pla d'execució

Oracle guarda en el DD , les estadístiques de les taules

En la Vista ==> `user_tables`

Usant les estadístiques (DD) , el monitor de rendiment i el registre d'errors, proposa un **pla d'execució**, que determina com es pot resoldre una consulta de la forma més eficient

Una vegada executades les consultes, els plans d'execució s'emmagatzemen en la **cau de consultes**



Usa estadístiques de taules !!

## Ferramentes d'Optimització Pla d'execució

Cada vegada que executem una sentència una de les coses que fa Oracle és crear un pla d'execució de la sentència. (SELECT, UPDATE, INSERT o DELETE)

Un pla d'execució defineix la forma en què Oracle cerca o grava les dades. Decideix, per exemple, si usará o no els índexs en una sentència SELECT

```
DELETE PLAN_TABLE;  
EXPLAIN PLAN FOR SELECT * FROM T_PEDIDOS WHERE CODPEDIDO = 5;  
select * from plan_table;
```



Usa estadístiques de taules !!

## Ferramentes d'Optimització

### Pla d'execució

```
grant select_catalog_role to nom_usu;  
grant select any dictionary to nom_usu;
```

🔊 Un usuari necessita tindre permís per revisar els plans d'execució de les consultes

El **SQL\*Plus** permet consultar el pla d'execució de les consultes, executant la següent instrucció

```
set autotrace traceonly explain
```

El **SQL Developer** permet consultar el pla d'execució de forma gràfica, polsant F10 sobre la consulta abans de llançar-la

F10



## Pla d'execució

El **SQL Developer** permet consultar el pla d'execució de forma gràfica, polsant F10 sobre la consulta abans de llançar-la

Prova-ho en SQL Developer

select \* from llibres3;

F10

Salida de Script x Explicación del Plan x

SQL | 6,113 segundos

OPERATION	OBJECT_NAME	OPTIONS
SELECT STATEMENT		
TABLE ACCESS	LIBRES3	FULL
Other XML		
{info}		
info type="has_user_tab"		
yes		
info type="db_version"		
19.0.0.0		
info type="parse_schema"		
"USUARI1"		





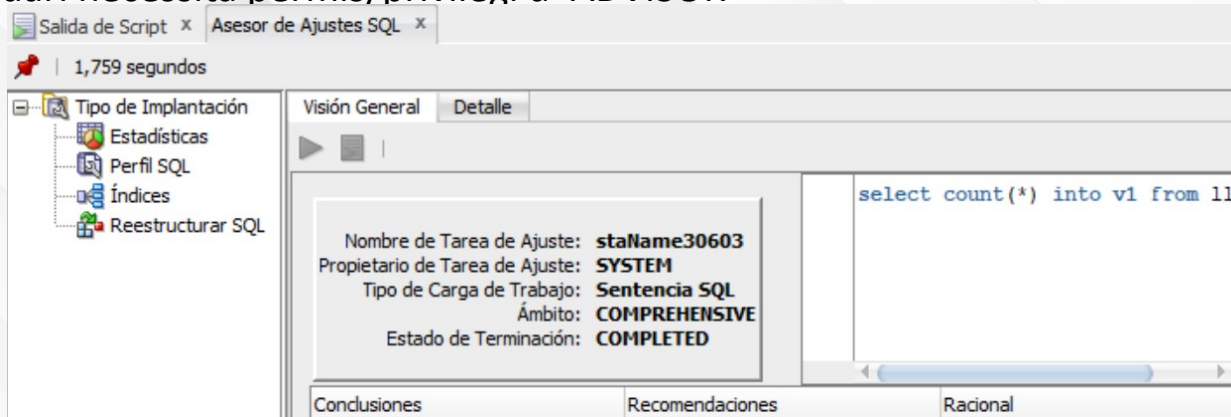
# Ferramentes d'Optimització SQL Tuning Advisor

- SQL Developer - SQL Tuning Advisor



En una SQL, abans d'executar, pulsa **ctrl + F12**  
L'usuari necessita permís/privilegi d' ADVISOR

Un usuari necessita tindre permís per executar el SQL tuning advisor



Run sql:  
alt+F11



## Ferramentes d'Optimització

### Monitor d'operacions

Es consulta des de EM , secció SQL Monitor  
o des del paquet DBMS\_SQL\_MONITOR,

- procediment report\_sql\_monitor
- vista V\$SQL\_MONITOR



# Ferramentes d'Optimització

## Operacions particulars d'Oracle

Insercions massives : SQL Loader

Insert Append: `INSERT /*+ APPEND */ INTO NOM_taula VALUES (...);`

Nologging `alter table t1 nologging;`

Truncate table `truncate table t1 ;`

Intercanvi de particions

Vistes materialitzades. Guarden consulta i dades. Solen guardar càlculs massius

Merge Combina la inserció i la modificació en una sola instrucció

Hints Els hints s'incorporen a una sentència DML en forma de comentari i han d'anar just darrere del comando principal. Per exemple, si es tractara d'una sentència SELECT el format seria el següent:  
`SELECT /*+ COMANDO-HINT */ ...`

S'ha d'anar amb compte amb estes operacions, donat que redueixen la seguretat i la possibilitat de recuperació davant d'operacions no desitjades !!