

Pràctica 2. Optimització de consultes en Oracle

Sistemes de Gestió de Bases de Dades
Grau en Enginyeria Informàtica



DEPARTAMENT D'INFORMÀTICA
UNIVERSITAT DE VALÈNCIA

Curs 2025 / 2026
Autors: Ramón Cirilo / Vicente Cerverón

Pràctica 2. Optimització de consultes en Oracle

Sistemes de Gestió de Bases de Dades

INDICE DEL DOCUMENTO

Contingut

PART I. INTRODUCCIÓ A LA OPTIMITZACIÓ DE CONSULTES EN ORACLE	2
GESTIÓ D'ESTADÍSTIQUES.....	2
DESCRIPCIÓ BÀSICA DE LES ESTADÍSTIQUES	2
MÈTODES D'ADQUISICIÓ D'ESTADÍSTIQUES.....	3
PLANS D'EXECUCIÓ	8
DESCRIPCIÓ BÀSICA DELS PLANS D'EXECUCIÓ.....	8
CÀLCUL I CONSULTA DEL PLA D'EXECUCIÓ	9
BIBLIOGRAFIA.....	12
PART II. DESENVOLUPAMENT DE LA PRÀCTICA	13
CREACIÓ DE TAULES BASE	13
GESTIÓ D'ESTADÍSTIQUES.....	13
EXERCICI 1	13
PLANS D'EXECUCIÓ DE CONSULTES	14
EXERCICI 2	14
EXERCICI 3	15
EXERCICI 4	15
EXERCICI 5	16
EXERCICI 6	17
PRESENTACIÓ I AVALUACIÓ DEL TREBALL REALITZAT.....	18

Part I. Introducció a la optimització de consultes en Oracle

Una sentència SQL pot ser executada pel SGBD de moltes formes diferents. Cadascuna d'aquestes formes segueix un pla d'execució particular, que no és més que una seqüència ordenada d'accions que porten a l'obtenció dels resultats especificats en la consulta. D'entre tots els plans d'execució, hi ha un que és l'òptim i que es caracteritza per fer un ús mínim dels recursos disponibles. La tasca de descobrir el millor pla d'execució la realitza l'**Optimitzador de Consultes**, que és un component habitual en els SGBD.

L'optimitzador de consultes determina la manera més eficient d'executar una sentència SQL després de la consideració de molts factors relacionats amb els objectes involucrats i les condicions específiques de la consulta. L'optimitzador genera un **pla d'execució** per a cada consulta que descriu el mètode d'execució a través d'un conjunt de passos que, o bé recuperen registres de la base de dades o bé els preparen per al seu ús en el següent pas.

En el SGBD de Oracle, l'optimitzador ha canviat molt des de les seues primeres versions. Inicialment, l'optimitzador utilitzava un conjunt de regles per a trobar el millor pla (RBO – Rule Based Optimizer). El principal problema d'un optimitzador d'aquestes característiques és que aplica les mateixes regles a tots els objectes, independentment de les seues condicions particulars, la qual cosa pot conduir a errors (no trobar el pla òptim). A partir de la versió 7, es va millorar el funcionament de l'optimitzador introduint un mecanisme de càlcul basat en costos i estadístiques (CBO – Cost Based Optimizer). El principal avantatge que suposava pel que fa a la versió anterior és que tenia en compte el contingut real de dades i la seua distribució (estadístiques sobre els objectes). En aquest cas, l'optimitzador calcula els costos d'execució de diversos plans i selecciona el de menor cost. Els plans que es consideren es fan sobre la base d'uns criteris de sedàs que limiten les possibilitats (que creixen exponencialment segons el nombre d'objectes involucrats en la consulta).

Per tant, per a entendre com funciona l'optimitzador de consultes és necessari saber la forma en la qual Oracle emmagatzema i gestiona les estadístiques sobre els objectes, i entendre les diferents alternatives que es poden utilitzar per a executar un pla. En aquesta pràctica, de 2 sessions de durada, es treballarà sobre tots dos conceptes.

GESTIÓ D'ESTADÍSTIQUES

Descripció bàsica de les estadístiques

Les estadístiques que utilitza l'optimitzador són una col·lecció de dades que descriuen la base de dades i els seus objectes, i permeten el càlcul del millor pla d'execució per a cada sentència SQL. Les estadístiques s'emmagatzemen en el diccionari de dades de la base de dades, i són accessibles a través d'algunes vistes, com USER_TABLES, USER_TAB_COLUMNS, USER_INDEXES, USER_TAB_STATISTICS, USER_IND_STATISTICS o USER_TAB_COL_STATISTICS.

Les estadístiques corresponents a les taules contenen informació sobre el nombre de files, el nombre de blocs de dades (emmagatzematge lògic) que usa la taula, i longitud mitjana dels registres, entre uns altres. Pel que fa a les columnes, les estadístiques inclouen informació sobre el nombre de valors

diferents que conté, així com el valor màxim i mínim de cadascuna. L'optimitzador usa les estadístiques de les taules per a estimar el cost d'accés a les dades, i usa les estadístiques de columnes per a estimar el nombre de registres que seran recuperats en cada pas del pla d'execució.

Aquestes estadístiques bàsiques són un punt de partida important per a l'optimitzador, però no aporten tota la informació necessària per a un càlcul correcte. Per a açò, és necessari conèixer la naturalesa de la informació, com podrien ser els biaixos, distorsions i distribucions de valors en una columna, o l'existència d'algun tipus de correlació entre columnes. Oracle proporciona, a aquest efecte, un conjunt d'estadístiques addicionals:

- **Histogrames:** contenen informació sobre la distribució de les dades d'una columna. En cas de no existir aquesta estadística, Oracle assumeix una distribució de dades uniforme.
 - Histogrames de freqüència: s'usen quan el nombre de valors diferents en una columna és inferior a 254.
 - Histogrames d'altura equilibrada: s'usen quan el nombre de valors diferents en una columna és superior a 254.
- **Grups de columnes:** contenen informació sobre columnes que tenen algun tipus de correlació (p. ex. província – país). Té sentit quan les columnes d'una mateixa taula s'usen de forma conjunta en la clàusula WHERE d'una consulta, ja que permet estimar de forma més exacta la cardinalitat del conjunt resultant.
- **Expressions:** Oracle permet també la creació d'estadístiques sobre expressions, incloent funcions, per a millorar l'estimació de costos. Per exemple, si s'usa amb freqüència la funció UPPER sobre una columna, convindria estendre les estadístiques per a incloure aquesta expressió.

Les estadístiques ampliades es capturen de forma manual, usant el procediment DBMS_STATS.CREATE_EXTENDED_STATS i poden ser consultades en les vistes del diccionari de dades USER_TAB_COL_STATISTICS i USER_STAT_EXTENSIONS.

Oracle també manté informació estadística sobre l'ús dels índexs. Aquesta informació és essencial per a poder determinar si és més o menys convenient en un pla d'execució usar l'accés a dades a través d'índexs o directament sobre la taula. La informació estadística dels índexs inclou el nombre de valors diferents (diferents claus), la profunditat de l'índex (altura), el nombre de blocs en els nodes fulla, i el factor de clustering.

Mètodes d'adquisició d'estadístiques

La forma d'obtenir estadístiques d'objectes en la base de dades Oracle és a través del paquet PL/SQL DBMS_STATS. Conté més de 50 procediments diferents per a recopilar estadístiques, però els més importants són els que tenen la forma GATHER_*_STATS, ja que permeten arrebregar informació sobre taules, columnes i índexs. Per a poder executar els procediments d'adquisició d'estadístiques sobre els objectes és necessari ser el propietari dels mateixos (o disposar del privilegi de sistema ANALYZE ANY).

Aquest mateix paquet disposa d'un procediment que automatitza el procés d'obtenció d'estadístiques durant una finestra de temps per a aquells objectes que ho necessiten. Perquè funcione correctament, és convenient configurar certs paràmetres relatius a la forma de recopilar la informació. Però a part del procediment automàtic, les estadístiques es poden arrebregar de forma manual quan es necessita. A continuació veurem amb una mica més de detall els principals procediments manuals i la forma d'utilitzar-los.

GATHER_TABLE_STATS

El procediment té la següent sintaxi:

```
DBMS_STATS.GATHER_TABLE_STATS (
  ownname          VARCHAR2,
  tablename         VARCHAR2,
  partname          VARCHAR2 DEFAULT NULL,
  estimate_percent  NUMBER      DEFAULT to_estimate_percent_type
                                     (get_param('ESTIMATE_PERCENT')),
  block_sample      BOOLEAN     DEFAULT FALSE,
  method_opt        VARCHAR2   DEFAULT get_param('METHOD_OPT'),
  degree            NUMBER      DEFAULT to_degree_type(get_param('DEGREE')),
  granularity       VARCHAR2   DEFAULT GET_PARAM('GRANULARITY'),
  cascade           BOOLEAN     DEFAULT to_cascade_type(get_param('CASCADE')),
  stattab           VARCHAR2   DEFAULT NULL,
  statid            VARCHAR2   DEFAULT NULL,
  statown           VARCHAR2   DEFAULT NULL,
  no_invalidate     BOOLEAN     DEFAULT to_no_invalidate_type (
                                     get_param('NO_INVALIDATE')),
  force             BOOLEAN     DEFAULT FALSE);
```

Els paràmetres obligatoris són el propietari de la taula i el nom de la taula, i la resta són opcionals i es prenen en alguns casos de la configuració de la base de dades. També són importants el paràmetre CASCADE, que permet la recopilació d'estadístiques en els índexs associats a la taula, i el paràmetre FORCE, que arrebrega estadístiques a pesar que la taula estiga bloquejada.

Exemple d'ús¹:

```
exec dbms_stats.gather_table_stats(
  ownname=>'GIISGBD',
  tablename=>'ACTORES',
  cascade=>true,
  force=>true);

begin
  dbms_stats.gather_table_stats(
    ownname=>'GIISGBD',
    tablename=>'ACTORES',
    cascade=>true,
    force=>true);
end;
/
```

¹ Totes les sentències exec s'han de posar en una única línia.

DELETE_TABLE_STATS

El procediment té la següent sintaxi:

```
DBMS_STATS.DELETE_TABLE_STATS (
  ownname          VARCHAR2,
  tabname          VARCHAR2,
  partname         VARCHAR2 DEFAULT NULL,
  stattab         VARCHAR2 DEFAULT NULL,
  statid          VARCHAR2 DEFAULT NULL,
  cascade_parts    BOOLEAN  DEFAULT TRUE,
  cascade_columns  BOOLEAN  DEFAULT TRUE,
  cascade_indexes  BOOLEAN  DEFAULT TRUE,
  statown         VARCHAR2 DEFAULT NULL,
  no_invalidate    BOOLEAN  DEFAULT to_no_invalidate_type (
                                get_param('NO_INVALIDATE')),
  force           BOOLEAN  DEFAULT FALSE);
```

Els paràmetres obligatoris són el propietari de la taula i el nom de la taula, i la resta són opcionals. També són importants el paràmetre CASCADE_COLUMNS i CASCADE_INDEXES, que permet l'eliminació d'estadístiques de les columnes i dels índexs associats a la taula, i el paràmetre FORCE, que elimina estadístiques a pesar que la taula estiga bloquejada.

Exemple d'ús:

```
exec dbms_stats.delete_table_stats(
  ownname=>'GIISGBD',
  tabname=>'ACTORES');
```

DELETE_COLUMN_STATS

El procediment té la següent sintaxi:

```
DBMS_STATS.DELETE_COLUMN_STATS (
  ownname          VARCHAR2,
  tabname          VARCHAR2,
  colname          VARCHAR2,
  partname         VARCHAR2 DEFAULT NULL,
  stattab         VARCHAR2 DEFAULT NULL,
  statid          VARCHAR2 DEFAULT NULL,
  cascade_parts    BOOLEAN  DEFAULT TRUE,
  statown         VARCHAR2 DEFAULT NULL,
  no_invalidate    BOOLEAN  DEFAULT to_no_invalidate_type (
                                get_param('NO_INVALIDATE')),
  force           BOOLEAN  DEFAULT FALSE);
```

Els paràmetres obligatoris són el propietari de la taula, el nom de la taula i el nom de la columna, i la resta són opcionals. El paràmetre FORCE permet l'eliminació estadístiques corresponents a la columna fins i tot si està bloquejada.

Exemple d'ús:

```
exec dbms_stats.delete_column_stats(
  ownname=>'GIISGBD',
  tabname=>'ACTORES',
  colname=>'SEXO');
```

GATHER_INDEX_STATS

El procediment té la següent sintaxi:

```
DBMS_STATS.GATHER_INDEX_STATS (
  ownname          VARCHAR2,
  indname          VARCHAR2,
  partname         VARCHAR2 DEFAULT NULL,
  estimate_percent NUMBER   DEFAULT to_estimate_percent_type
                                (GET_PARAM('ESTIMATE_PERCENT')),
  stattab          VARCHAR2 DEFAULT NULL,
  statid           VARCHAR2 DEFAULT NULL,
  statown          VARCHAR2 DEFAULT NULL,
  degree           NUMBER   DEFAULT to_degree_type(get_param('DEGREE')),
  granularity      VARCHAR2 DEFAULT GET_PARAM('GRANULARITY'),
  no_invalidate    BOOLEAN  DEFAULT to_no_invalidate_type
                                (GET_PARAM('NO_INVALIDATE')),
  force            BOOLEAN  DEFAULT FALSE);
```

Els paràmetres obligatoris són el propietari de l'índex i el nom de l'índex, i la resta són opcionals i de nou es prenen en alguns casos de la configuració de la base de dades. El paràmetre FORCE permet que s'arreglessen estadístiques de l'índex fins i tot si està bloquejat.

Exemple d'ús:

```
exec dbms_stats.gather_index_stats(
  ownname=>'GIISGBD',
  indname=>'IDX_SEXO',
  force =>true);
```

DELETE_INDEX_STATS

El procediment té la següent sintaxi:

```
DBMS_STATS.DELETE_INDEX_STATS (
  ownname          VARCHAR2,
  indname          VARCHAR2,
  partname         VARCHAR2 DEFAULT NULL,
  stattab          VARCHAR2 DEFAULT NULL,
  statid           VARCHAR2 DEFAULT NULL,
  cascade_parts    BOOLEAN  DEFAULT TRUE,
  statown          VARCHAR2 DEFAULT NULL,
  no_invalidate    BOOLEAN  DEFAULT to_no_invalidate_type (
                                get_param('NO_INVALIDATE')),
  force            BOOLEAN  DEFAULT FALSE);
```

Els paràmetres obligatoris són el propietari de l'índex i el nom de l'índex. El paràmetre FORCE permet que s'eliminen estadístiques de l'índex fins i tot si està bloquejat.

Exemple d'ús:

```
exec dbms_stats.delete_index_stats(
  ownname=>'GIISGBD',
  indname=>'IDX_SEXO',
  force=>true);
```

GATHER_SCHEMA_STATS

El procediment té la següent sintaxi:

```
DBMS_STATS.GATHER_SCHEMA_STATS (
  ownname          VARCHAR2,
  estimate_percent  NUMBER    DEFAULT to_estimate_percent_type
                                (get_param('ESTIMATE_PERCENT')),
  block_sample      BOOLEAN    DEFAULT FALSE,
  method_opt        VARCHAR2  DEFAULT get_param('METHOD_OPT'),
  degree            NUMBER    DEFAULT to_degree_type(get_param('DEGREE')),
  granularity       VARCHAR2  DEFAULT GET_PARAM('GRANULARITY'),
  cascade           BOOLEAN    DEFAULT to_cascade_type(get_param('CASCADE')),
  stattab           VARCHAR2  DEFAULT NULL,
  statid            VARCHAR2  DEFAULT NULL,
  options           VARCHAR2  DEFAULT 'GATHER',
  objlist           OUT       ObjectTab,
  statown           VARCHAR2  DEFAULT NULL,
  no_invalidate     BOOLEAN    DEFAULT to_no_invalidate_type (
                                get_param('NO_INVALIDATE')),
  force             BOOLEAN    DEFAULT FALSE);
```

L'únic paràmetre obligatori és el nom de l'esquema del que es volen prendre les estadístiques. L'opció CASCADE permet arregar de forma addicional estadístiques dels índexs (la qual cosa no es realitza per defecte) i l'opció FORCE esquivia els bloquejos que puguen existir en els objectes.

Exemple d'ús:

```
exec dbms_stats.gather_schema_stats(
  ownname=>'GIISGBD',
  cascade=>true,
  force=>true);
```

DELETE_SCHEMA_STATS

El procediment té la següent sintaxi:

```
DBMS_STATS.DELETE_SCHEMA_STATS (
  ownname          VARCHAR2,
  stattab           VARCHAR2  DEFAULT NULL,
  statid            VARCHAR2  DEFAULT NULL,
  statown           VARCHAR2  DEFAULT NULL,
  no_invalidate     BOOLEAN    DEFAULT to_no_invalidate_type (
                                get_param('NO_INVALIDATE')),
  force             BOOLEAN    DEFAULT FALSE);
```

L'únic paràmetre obligatori és el nom de l'esquema del que es volen esborrar les estadístiques. L'opció FORCE esquivia els bloquejos que puguen existir en els objectes.

Exemple d'ús:

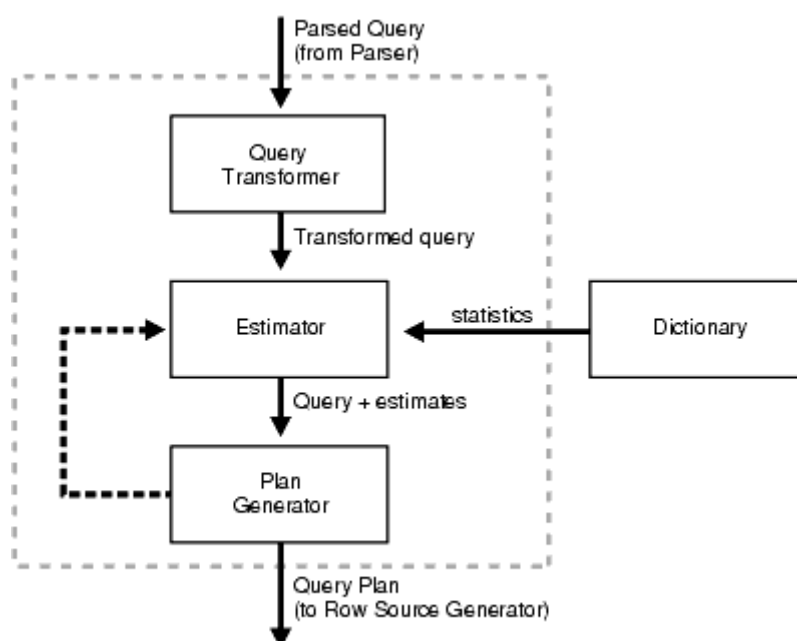
```
exec dbms_stats.delete_schema_stats(
  ownname=>'GIISGBD',
  force=>true);
```


PLANS D'EXECUCIÓ

Descripció bàsica dels plans d'execució

Un pla d'execució mostra de manera detallada els passos necessaris per a executar una sentència SQL. Aquests passos s'expressen en forma d'operacions sobre la base de dades que consumeixen i produeixen registres. L'ordre de les operacions i la forma d'implementar-les les decideix l'optimitzador de consultes, i per a açò utilitza una combinació de transformacions de consultes i tècniques d'optimització físiques (d'accés a dades).

Els components de l'optimitzador de consultes de Oracle són:



El generador de plans és l'encarregat provar diferents alternatives per a l'execució d'una consulta concreta i seleccionar aquella que ofereix un menor cost. La raó que existisquen moltes alternatives és que existeixen diverses combinacions de rutes d'accés a les dades, mètodes de reunió i ordre de reunions que generarien el mateix resultat final, però amb diferents costos.

L'optimitzador de Oracle seleccionarà, sobre la base de les estadístiques que posseïska dels objectes involucrats en una consulta, el millor pla d'execució possible per a ella. Així doncs, quan es dissenyen consultes que involucren objectes molt pesats (amb molta quantitat de dades), és obligatori mantenir les estadístiques actualitzades perquè l'optimitzador trobe el millor pla (i consumeixi els mínims recursos possibles), i es fa necessari revisar el pla d'execució per si existira la possibilitat de millorar l'execució a través d'una re-organització física de les dades (índexs, particions, etc.). A aquest procés se li denomina Ajust de Consultes SQL (SQL Tuning).

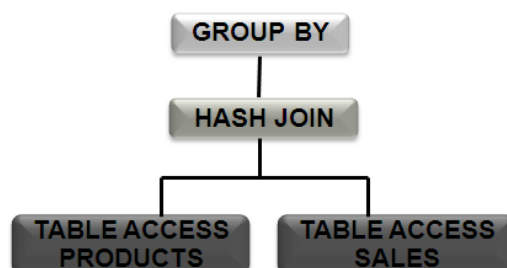
A continuació es descriuen els elements bàsics per a l'obtenció i l'anàlisi de plans d'execució de consultes en Oracle.

Càlcul i consulta del pla d'execució

En Oracle, el pla d'execució d'una consulta es mostra de forma tabular, per a la seua millor comprensió per part de l'usuari, en lloc d'una representació en forma d'arbre (com és habitual en la representació dels plans, però que no es pot implementar en un interfície de tipus text com el qual ofereixen les eines de consulta de base de dades Oracle, tipus SQL*Plus). Té un aspecte similar a est:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				1140 (100)			
1	HASH GROUP BY		4	80	1140 (45)	00:00:14		
* 2	HASH JOIN		489K	9555K	792 (21)	00:00:10		
3	TABLE ACCESS FULL	PRODUCTS	767	8437	10 (0)	00:00:01		
4	PARTITION RANGE ALL		489K	4300K	741 (17)	00:00:09	1	16
5	TABLE ACCESS FULL	SALES	489K	4300K	741 (17)	00:00:09	1	16

Aquest pla d'execució correspondria a un arbre d'execució així:



El pla d'execució mostra informació sobre les operacions realitzades, els objectes involucrats, l'estimació dels registres que maneja cada operació, la memòria necessària per a cada operació, el cost associat a cada operació i el temps estimat d'execució, entre uns altres. Corresponen, respectivament, a les columnes Operation, Name, Rows, Bytes, Cost i Time. La mesura de cost és una mesura interna a la base de dades i només té sentit usar-la quan es comparen plans d'execució.

La part més complexa d'entendre, potser, resideix en les operacions. Existeixen diferents tipus d'operacions que es poden realitzar en els registres, i que es poden classificar en:

- Mètodes d'accés
 - Full Table Scan. Lliga tots els registres d'una taula i filtra els que satisfan el criteri de la clàusula WHERE.
 - Table Access by ROWID. El ROWID especifica la posició exacta on resideix un registre. Oracle obté primer el ROWID a través d'un índex o d'un filtrat de la clàusula WHERE, i després accedeix a les dades registre a registre.
 - Index Unique Scan. Es recupera només un registre després de recórrer un índex únic.
 - Index Range Scan. Oracle accedeix a entrades d'índex adjacents i després localitza els registres corresponents als ROWID recuperats.
 - Index Range Scan Descending. Igual que el cas anterior, però aplicat quan es recorre l'índex de forma inversa a causa d'un ORDER BY .. DESC.

- Index Skip Scan. S'usa quan apareixen totes les columnes presents en l'índex excepte la primera.
- Full Index Scan. Malgrat el que suggereix, no realitza una lectura de tots els blocs de l'índex. El que fa Oracle és processar tots els blocs fulles de l'índex fins a trobar el primer que necessita.
- Fast Full Index Scan. S'usa com a alternativa al full table scan, quan l'índex conté totes les columnes que es necessiten per a la consulta i almenys una columna de l'índex té la restricció NOT NULL.
- Index Join. Es tracta d'una reunió d'índexs de la mateixa taula que de forma col·lectiva contenen totes les columnes necessàries per a la consulta.
- Bitmap Index. Utilitza un mapa de bits per als valors de les claus, i cada posició es enllaça amb un ROWID.
- Mètodes de reunió
 - Hash Join. S'usa per a conjunts molt grans de dades. L'optimitzador usa el més xicotet per a construir una taula hash, basada en la clau de reunió, i escaneja posteriorment la taula més gran aplicant la mateixa funció de hash, per a acabar cercant els valors coincidents.
 - Nested Loop Join. S'aplica per a conjunts de dades xicotetes. Per a cada registre de la primera taula, s'accedeix a tots els registres de la segona taula.
 - Sort Merge Join. Són útils quan la condició de reunió de les dues taules és una desigualtat. En aquests casos, funcionen millor que els Nested Loop. S'aplica una ordenació a tots dos conjunts sobre la clau de la reunió i a continuació es procedeix a la reunió.
 - Cartesian Join. Es combinen totes les tuples d'una taula amb les de l'altra taula. Només s'aplica per a conjunts de dades xicotetes o quan ho exigeix la consulta per no haver-hi criteri de reunió.

Adicionalment, els mètodes de reunió impliquen un ordre en la realització de l'operació, que en la taula es mostra en forma d'indentació. Açò significa que els objectes amb més indentació es reuneixen en primer lloc; a partir d'ací, el resultat es reuneix amb el següent element indentat, i així successivament fins a arribar a l'última operació. L'ordre de reunió és important, en funció del mètode emprat, ja que en funció de la cardinalitat de les reunions pot comportar costos diferents, i és per açò que han d'interpretar-se correctament en el pla.

Els plans d'execució s'emmagatzemen en una taula del diccionari de dades de la base de dades anomenada PLAN_TABLE. Aquesta taula sol estar creada en la base de dades, i està preparada perquè cada usuari la utilitze amb independència. En cas que la taula no estiga creada, pot crear-se executant el script utlxplan.sql que es troba situat en el directori \$ORACLE_HOME/rdbms/admin on està instal·lada la base de dades. El script també pot utilitzar-se per a personalitzar la taula de plans i evitar problemes d'accés concurrent entre diferents sessions i/o usuaris.

Per a poder obtenir el pla d'execució d'una consulta s'utilitza el comando EXPLAIN PLAN, que emplena la taula PLAN_TABLE. La visualització del pla com una consulta sobre la taula és una mica tediosa i complicada, per la qual cosa habitualment s'utilitza el paquet DBMS_XPLAN, que s'encarrega de

proporcionar la vista tabular formatada adequadament per a la seua millor comprensió, tal com s'ha mostrat al principi d'aquesta secció.

El comando EXPLAIN PLAN té aquesta sintaxi:

```
EXPLAIN PLAN FOR
  <query>;
```

Per exemple:

```
EXPLAIN PLAN FOR
SELECT last_name FROM employees;
```

Aquest comando emmagatzema el pla d'execució en la taula PLAN_TABLE. No obstant açò, és possible emmagatzemar el pla en una taula personalitzada (una còpia de la taula PLAN_TABLE), utilitzant aquesta sintaxi:

```
EXPLAIN PLAN
  INTO <table>
  FOR <query>;
```

Per exemple:

```
EXPLAIN PLAN
INTO my_plan_table
FOR
SELECT last_name FROM employees;
```

Una vegada executat el procés de creació i emmagatzematge del pla, podrà ser consultat. Per a açò, s'utilitza el procediment DISPLAY del paquet DBMS_XPLAN, que consulta l'última sentència SQL analitzada i emmagatzemada en la taula de plans. La sintaxi del procediment és:

```
DBMS_XPLAN.DISPLAY (
  table_name      IN  VARCHAR2  DEFAULT 'PLAN_TABLE',
  statement_id    IN  VARCHAR2  DEFAULT NULL,
  format          IN  VARCHAR2  DEFAULT 'TYPICAL',
  filter_preds    IN  VARCHAR2  DEFAULT NULL);
```

Com pot apreciar-se, tots els paràmetres són opcionals. No obstant açò, si s'està utilitzant una taula de plans personalitzada, serà necessari indicar-ho en el paràmetre TABLE_NAME. Igualment, el paràmetre FORMAT permet indicar la quantitat d'informació a visualitzar, podent-se especificar algun dels següents valors:

- BASIC. Només mostra els detalls mínims del pla, que són l'operació i l'objecte sobre el qual es realitza l'acció.
- TYPICAL. Conté la informació més rellevant per a entendre completament el pla, la qual cosa inclou estimacions de registres i memòria, costos, estimacions temporals, i graus de paral·lelització.
- SERIAL. Conté la mateixa informació que TYPICAL a excepció de la paral·lelització.
- ALL. Mostra tota la informació disponible.

També és possible especificar en aquest camp combinacions de paràmetres a visualitzar, com per exemple 'ALL -PROJECTION -NOTE', 'TYPICAL PROJECTION', '-BYTES -COST -PREDICATE'.

La forma d'usar el procediment és aquesta:

```
SELECT plan_table_output  
FROM  
      table (DBMS_XPLAN.DISPLAY ('PLAN_TABLE',  
      NULL, 'TYPICAL')) ;
```

o bé

```
SELECT * FROM table (DBMS_XPLAN.DISPLAY) ;
```

Amb aquests conceptes bàsics sobre plans d'execució, i assegurant que les estadístiques sobre els objectes de treball estan degudament actualitzades, es pot emprendre l'anàlisi de consultes SQL per a la seua execució òptima.

BIBLIOGRAFIA

Understanding Optimizer Statistics.

<https://www.oracle.com/technetwork/database/bi-datawarehousing/twp-optimizer-stats-concepts-110711-1354477.pdf>

<https://www.oracle.com/technetwork/database/bi-datawarehousing/twp-statistics-concepts-12c-1963871.pdf>

Best practices for gathering optimizer statistics.

<https://www.oracle.com/technetwork/database/bi-datawarehousing/twp-bp-optimizer-stats-04042012-1577139.pdf>

DBMS_STATS package.

https://docs.oracle.com/en/database/oracle/oracle-database/21/arpls/DBMS_STATS.html

Query Optimizer Concepts

<https://docs.oracle.com/en/database/oracle/oracle-database/21/tgsql/query-optimizer-concepts.html>

Explaining and Displaying Execution Plans

<https://docs.oracle.com/en/database/oracle/oracle-database/21/tgsql/generating-and-displaying-execution-plans.html>

EXPLAIN PLAN

<https://docs.oracle.com/en/database/oracle/oracle-database/21/sqlrf/EXPLAIN-PLAN.html>

DBMS_XPLAN package.

https://docs.oracle.com/en/database/oracle/oracle-database/21/arpls/DBMS_XPLAN.html

Part II. Desenvolupament de la pràctica

El desenvolupament d'aquesta pràctica es realitzarà sobre la base de dades Oracle 11g instal·lada en el servidor `pokemon.uv.es`, per a la qual els alumnes disposen de les credencials d'accés.

CREACIÓ DE TAULES BASE

El desenvolupament d'aquesta pràctica requereix la creació de 3 taules en l'esquema de l'usuari tal com es descriuen a continuació.

Taula: ACTORES

COLUMNA	TIPUS	PK	NN	UNI	CHK	FK	COMENTARI
OID	NUMBER(10,0)	✓					
NOM	CHAR(100)		✓				
SEXO	CHAR(1)		✓				

Taula: PELÍCULAS

COLUMNA	TIPUS	PK	NN	UNI	CHK	FK	COMENTARI
OID	NUMBER(10,0)	✓					
TITULO	CHAR(200)		✓				
ANYO	NUMBER(4,0)		✓				

Taula: ACTUACIÓN

COLUMNA	TIPUS	PK	NN	UNI	CHK	FK	COMENTARI
ACTOR	NUMBER(10,0)	✓					
PELI	NUMBER(10,0)	✓					
PAPEL	CHAR(100)	✓					

El script per a generar aquestes taules es proporciona juntament amb l'enunciat en el fitxer `crea_tablas.sql`.

Per a facilitar la càrrega de les dades en els esquemes dels alumnes, el resultat de la inserció del script `datos1.sql` es pot trobar a les taules `ACTORES_DATOS1`, `PELICULAS_DATOS1` i `ACTUACION_DATOS1` del esquema `GIISGBD`, i el resultat de la execució del script `datos2.sql` es pot trobar a les taules `ACTORES_DATOS2`, `PELICULAS_DATOS2` i `ACTUACION_DATOS2`, també del esquema `GIISGBD`. De esta manera, és més ràpid fer una càrrega amb la execució de una sentència de tipus `INSERT ... SELECT`.

GESTIÓ D'ESTADÍSTIQUES

Exercici 1

Realitzar les següents tasques:

1. Omplir les taules amb les dades del script `datos2.sql`.
2. Crear els següents índexs:
 - a. `IDX_SEXO` ON `ACTORES (SEXO)`
 - b. `IDX_ANYO` ON `PELICULAS (ANYO)`

3. Generar estadístiques completes de les taules ACTORES, PELÍCULAS i ACTUACIÓN, incloent els índexs associats.
4. Consultar la informació estadística generada i proporciona en forma tabular la següent informació:
 - a. Taules: nom, nombre de files, nombre de blocs, longitud mitjana dels registres.
 - b. Columnes: nom de taula i de columna, quantitat de valors diferents, valors mínim i màxim, nombre de nuls, longitud mitjana de la columna, histograma.
 - c. Índexs: nom, taula, profunditat, blocs en les fulles, quantitat de claus diferents, mitjana de blocs fulla per clau, mitjana de blocs de dades per clau, factor de clustering, nombre de registres.
5. Identifiqueu alguna cosa estrany en les estadístiques (valors estranys o inconsistents)?
6. Eliminar els índexs i les estadístiques generades.
7. Consultar de nou la informació estadística emmagatzemada (no és necessari proporcionar els resultats).

PLANS D'EXECUCIÓ DE CONSULTES

Exercici 2

Per a la realització d'aquest exercici les taules han d'estar emplenades amb les dades del script `dades2.sql` i sense estadístiques. Considerar les següents consultes:

1. **SELECT ***
FROM PELICULAS
WHERE ANYO > 1992;
2. **SELECT ***
FROM PELICULAS P, ACTUACION ACT
WHERE P.OID = ACT.PELI
AND ANYO > 1992;
3. **SELECT P.***
FROM ACTORES A, ACTUACION X, PELICULAS P
WHERE A.NOM = 'Loy, Myrna'
AND A.OID = X.ACTOR
AND P.OID = X.PELI;

Realitzar les següents tasques:

1. Executar les consultes i anotar els registres recuperats i el temps d'execució.
2. Obtenir per a cadascuna d'elles el seu pla d'execució.
 - a. Existeix alguna contradicció entre els plans d'execució i els continguts de les taules?
3. Investiga el significat del missatge "Note - dynamic sampling used for this statement".
4. Crear estadístiques per a tots els objectes
5. Tornar a obtenir el pla d'execució de cada consulta i comparar-los amb els anteriors.
 - a. Quins canvis s'aprecien i en quins casos? Justifica-ho.

Exercici 3

Les taules han d'estar emplenades amb les dades del script `dades2.sql` i sense estadístiques. Considerar les següents consultes:

- 1. SELECT * FROM PELICULAS WHERE ANYO=1920;**
- 2. SELECT * FROM PELICULAS WHERE ANYO<1920;**
- 3. SELECT * FROM PELICULAS WHERE ANYO>1920;**
- 4. SELECT * FROM PELICULAS WHERE ANYO<>1920;**

Realitzar les següents tasques:

1. Obtenir per a cadascuna d'elles el seu pla d'execució.
2. Generar les estadístiques corresponents i tornar a obtenir el pla d'execució per a cadascuna d'elles.
 - a. Han canviat els plans d'execució? Per què motiu?
3. Crear l'índex `IDX_ANYO ON PELICULAS (ANYO)`, generar les estadístiques adequades i obtenir de nou el pla d'execució per a totes les consultes.
 - a. Analitzar les diferències trobades i explicar-les.
4. Ja hauràs interpretat que l'índex `IDX_ANYO` s'utilitza sempre que el nombre de registres recuperats siga inferior a una determinada quantitat. Fes les proves necessàries per a determinar un llindar inferior i superior d'aquesta quantitat. No cal que poses a la memòria totes les proves que has fet, però indica què proves son i quins valors llindars has trobat.
5. Eliminar l'índex creat.

Exercici 4

Les taules han d'estar emplenades amb les dades del script `dades1.sql` i sense estadístiques. Donades les següents consultes:

- 1. SELECT * FROM ACTORES WHERE SEXO = 'H';**
- 2. SELECT COUNT(*) FROM ACTORES WHERE SEXO = 'H';**

Realitzar les següents tasques:

1. Generar les estadístiques adequades (per a les taules que intervenen en les consultes).
2. Obtenir el pla d'execució d'ambdues consultes.
3. Crear l'índex de tipus B-Tree `IDX_SEXO ON ACTORES (SEXO)` i generar les estadístiques adequades.

4. Tornar a obtenir el pla d'execució de les consultes.
5. Eliminar l'índex creat en el pas 3, i crear un nou índex sobre el mateix camp de la mateixa taula, però aquesta vegada de tipus BITMAP, generant de nou les estadístiques adequades.
6. Tornar a obtenir el pla d'execució de les consultes.
7. Analitzar les diferències trobades i justificar-les.
8. Eliminar l'índex creat.

Exercici 5

Donades les següents consultes:

1.

```
SELECT A.NOM, A.SEXO
FROM GIISGBD.ACTORES_BASE A
WHERE EXISTS (
  SELECT *
  FROM GIISGBD.PELICULAS_BASE P,
       GIISGBD.ACTUACION_BASE ACT
  WHERE P.OID = ACT.PELI
        AND P.ANYO = 1980
        AND A.OID = ACT.ACTOR) ;
```
2.

```
SELECT A.NOM, A.SEXO
FROM GIISGBD.ACTORES_BASE A
WHERE A.OID IN (
  SELECT ACT.ACTOR
  FROM GIISGBD.PELICULAS_BASE P,
       GIISGBD.ACTUACION_BASE ACT
  WHERE P.OID = ACT.PELI
        AND P.ANYO = 1980) ;
```
3.

```
SELECT A.NOM, A.SEXO
FROM GIISGBD.ACTORES_BASE A
WHERE A.OID = ANY (
  SELECT ACT.ACTOR
  FROM GIISGBD.PELICULAS_BASE P,
       GIISGBD.ACTUACION_BASE ACT
  WHERE P.OID = ACT.PELI
        AND P.ANYO = 1980) ;
```
4.

```
SELECT DISTINCT A.NOM, A.SEXO
FROM GIISGBD.PELICULAS_BASE P,
     GIISGBD.ACTUACION_BASE ACT, GIISGBD.ACTORES_BASE A
WHERE P.OID = ACT.PELI
      AND A.OID = ACT.ACTOR
      AND P.ANYO = 1980 ;
```

Les 4 consultes són conceptualment idèntiques, ja que obtenen exactament els mateixos resultats, encara que estan expressades de formes diferents: subconsulta correlacionada (EXISTS), subconsulta no correlacionada (amb IN i =ANY) i reunió. Realitzar les següents tasques:

1. Generar les estadístiques adequades (per a les taules que intervenen en les consultes).
2. Obtenir el pla d'execució de les 4 consultes.
 - a. S'observa alguna diferència o similitud entre els plans? A què es deuen?
3. Canviar el filtre del camp P.ANYO en totes les consultes al valor 2000. Obtenir de nou els plans d'execució de les 4 consultes.
 - a. S'observa alguna diferència pel que fa als plans anteriors? Per què motiu?

Exercici 6

Les taules de l'usuari local han d'estar emplenades amb les dades del script *datos1.sql* i sense estadístiques. Donades les següents consultes:

1. **SELECT A.NOM, P.TITULO, ACT.PAPEL, P.ANYO
FROM ACTORES A, PELICULAS P, ACTUACION ACT
WHERE A.OID = ACT.ACTOR
AND P.OID = ACT.PELI
AND SEXO = 'H'
AND P.TITULO LIKE 'Lost%';**
2. **SELECT A.NOM, P.TITULO, ACT.PAPEL, P.ANYO
FROM ACTORES_BASE A, PELICULAS_BASE P,
ACTUACION_BASE ACT
WHERE A.OID = ACT.ACTOR
AND P.OID = ACT.PELI
AND SEXO = 'H'
AND P.TITULO LIKE 'Lost%';**
3. **SELECT A.NOM, P.TITULO, ACT.PAPEL, P.ANYO
FROM ACTORES_BASE A, PELICULAS_BASE P,
ACTUACION_BASE ACT
WHERE A.OID = ACT.ACTOR
AND P.OID = ACT.PELI
AND SEXO = 'H'
AND P.TITULO LIKE '%Empire%';**

Realitzar les següents tasques:

1. Generar les estadístiques adequades (per a les taules que intervenen en les consultes).
2. Obtenir el pla d'execució d'ambdues consultes.

3. Proposar les millores necessàries perquè les consultes s'executen de manera més eficient.
(Ajuda: crear indexes sobre totes les columnes rellevants i analitzar la seua efectivitat dins del pla d'execució)

Nota: Per a la realitzar proves en la consulta 2 i 3 s'han de copiar les taules en els esquemes locals (`create table XXX as select * from giisgbd.XXX`) i esborrar-les en acabar el treball (`drop table XXX`).

PRESENTACIÓ I AVALUACIÓ DEL TREBALL REALITZAT

El resultat de la pràctica es lliurarà en forma d'un script sql que realitzi totes les accions necessàries per a dur a terme la mateixa i obtenir la informació necessària, exceptuant la creació inicial de les taules i les càrregues de dades. El script haurà de contenir en forma de comentari les respostes a les preguntes plantejades en cada apartat al llarg de la pràctica.

Els alumnes hauran de pujar a l'aula virtual, al final de la segona sessió de la pràctica, els resultats esperats.

De forma addicional, el professor valorarà el treball realitzat pels estudiants en finalitzar la primera i segona sessió de pràctiques, de manera que els alumnes hauran d'indicar el grau de desenvolupament de la mateixa previ al seu lliurament a través de l'aula virtual.