

## 3ª PRÁCTICA – Diseño Físico en Oracle® BD

---

**Implementar** una carga de trabajo estándar (workload) basada en las consultas de la 2ª práctica y las siguientes actualizaciones:

- **INSERT: 4803 propuestas (una de cada miembro de los 362 clubes activos) y 69611 comentarios (uno para cada una de esas propuestas y cada miembro de su club).**
- **Ejecutar cada una de las (tres) consultas y consultar ambas vistas.**

Nota: dado que la carga de prueba altera significativamente el estado de la base, aumentando además su volumen de datos, es necesario restablecer el estado inicial de la base de datos después de realizar cada ciclo de pruebas borrando los nuevos registros.

**Medir** el rendimiento de la carga de trabajo estándar (tanto en tiempo como en número de accesos) a través de varias repeticiones.

**Analiza** el plan de ejecución para cada operación, extrayendo los procesos (la tipología) de los que se compone. Clasifica y ordena todos los procesos en la estructura física, su frecuencia y una aproximación del coste.

A partir de ese estudio, desarrolla una propuesta para reducir el número de accesos a memoria secundaria.

Describe el **diseño físico completo**, comenzando por las organizaciones básicas involucradas en el mismo así como las estructuras auxiliares utilizadas para mejorar el rendimiento.

**Implementa** el diseño físico completo propuesto, y mide su rendimiento sobre la carga de trabajo estándar propuesta.

**Compara** el rendimiento del diseño físico original frente al propuesto e interpreta los resultados. Opcionalmente, puedes refinar la propuesta, e implementar mejoras al diseño físico (cuyo rendimiento después analizarás y compararás con los anteriores).

**Documenta** todo el trabajo realizado en esta práctica, poniendo especial atención al diseño físico, a las medidas de rendimiento, a los planes de ejecución, a las propuestas de mejora, y a la comparación de resultados.

---

### Material de Apoyo:

- Transparencias de laboratorio.
- SQL Scripts (sintaxis Oracle PL/SQL): carga de trabajo estándar y procedimientos relacionados con las medidas de rendimiento basadas en consultar las estadísticas que recoge y almacena Oracle.
- Cuenta de Oracle DBMS 11g.

## ANEXO: Código de apoyo (para generar la *carga de trabajo*)

---

### Creación de elementos:

- Paquete estadístico (ver script)
- Inserción de fila auxiliar en 'movies'

```
INSERT INTO movies(title,director)
  SELECT DISTINCT 'workload_'||nick,'Common' FROM membership;
-- 5297 rows inserted.

commit;
```

- Creación de vistas

```
CREATE VIEW CP_aragna AS (
SELECT nick, 100*sum(nuls)/count('c') pct_aragna
  FROM (SELECT nick,club,title,director,
              NVL2(msg_date,0,1) nuls
        FROM (SELECT member nick, club, title, director
              FROM proposals)
        LEFT OUTER JOIN comments
        USING(nick,club,title,director)
      )
  GROUP BY nick
  HAVING sum(nuls)>0
) order by pct_aragna desc WITH READ ONLY;

CREATE VIEW leader AS(
SELECT * FROM (
  SELECT * FROM (
    SELECT member nick,club, sum(nvl(coms,0))/count('c') total
      FROM proposals LEFT OUTER JOIN
        (SELECT club,title,director,count('c') coms
          FROM comments GROUP BY club,title,director) A
      USING(club,title,director)
    GROUP BY club,member
  ) ORDER BY total DESC
) WHERE rownum<=10
) WITH READ ONLY;
```

---

### Cursores (implícitos en bucle) para la ejecución de las tres consultas:

```
FOR fila in (
  SELECT title,director
    FROM GENRES_MOVIES WHERE UPPER(genre)='COMEDY'
  INTERSECT
  SELECT title,director
    FROM GENRES_MOVIES WHERE UPPER(genre)='DRAMA'
) LOOP null; END LOOP;
```

```

FOR fila in (
    SELECT nick FROM users WHERE (sysdate-reg_date)>182
    MINUS
    SELECT nick FROM membership
    MINUS
    SELECT nick FROM contracts JOIN profiles USING(citizenID)
) LOOP null; END LOOP;

```

```

FOR fila in (
    SELECT * FROM (
        SELECT * FROM (
            SELECT director, avg(coms) meancomm
            FROM (SELECT title,director,count('c') coms
                  FROM comments
                  GROUP BY title,director)
            GROUP BY director
        ) ORDER BY meancomm DESC
    ) WHERE rownum=1
) LOOP null; END LOOP;

```

```

FOR fila in (SELECT * FROM CP_aragna
) LOOP null; END LOOP;

```

```

FOR fila in (SELECT * FROM leader
) LOOP null; END LOOP;

```

---

### **Insertión de filas:**

```

INSERT INTO proposals(title, director, club, member, prop_date,
slogan, message)
    SELECT 'workload_'||nick,'Common',club,nick,sysdate,
           DBMS_RANDOM.string('P',TRUNC(DBMS_RANDOM.value(20,96))),
           DBMS_RANDOM.string('P',TRUNC(DBMS_RANDOM.value(200,1496)))
    FROM (SELECT name club FROM clubs WHERE end_date is null)
        JOIN membership USING(club);

INSERT INTO comments(club, nick, msg_date, title, director, subject,
message, valoration)
    SELECT club,nick,sysdate+sec/43200,title,'Common',
           DBMS_RANDOM.string('P',TRUNC(DBMS_RANDOM.value(20,96))),
           DBMS_RANDOM.string('P',TRUNC(DBMS_RANDOM.value(200,1496))),
           TRUNC(DBMS_RANDOM.value(0,11))
    FROM (SELECT title,club, rownum sec FROM proposals WHERE title LIKE
        'workload_') JOIN membership USING(club);

```

---

### **Borrado de filas (y purga de espacio):**

```

delete from proposals where title LIKE 'workload_%';
delete from comments where title LIKE 'workload_%';

alter table proposals deallocate unused;
alter table comments deallocate unused;

```

```
alter index PK_PROPOSALS deallocate unused;
alter index UK_PROPOSALS deallocate unused;
alter index PK_COMMENTS deallocate unused;

-- purge any other index, materialized view or cluster involving
   tables proposals and/or comments
-- alter index name_of_the_index deallocate unused;
-- alter cluster name_of_the_cluster deallocate unused;
-- alter materialized view name_of_the_view deallocate unused;
```