

## **Solución de Laboratorio N°1 - Ejercicio de Tablespace**

### *1. Elaborar el sizing de la base de datos a crear para un periodo de 5 años*

Paso 1: Evaluar el Volumen Actual de Datos (T0)

Jugadores: se estima que inician con ~500.

Equipos de eSports: unos 60.

Torneos: unos 40 por año.

Contratos: cada jugador puede cambiar de equipo 1-2 veces en 5 años.

Participación en Torneos: cada torneo lo juegan ~25 equipos.

Paso 2: Estimar la tasa de crecimiento

Ciclistas: +6% anual.

Equipos: +3% anual.

Pruebas: +5% anual.

Paso 3: Considerar los índices

Cada clave primaria y foránea genera índices.

Estimar un 25% adicional sobre el tamaño de las tablas.

Paso 4: Calcular el tamaño total

El cálculo teórico da un tamaño muy pequeño (< 1 MB).

Paso 5: Planificar el espacio adicional

Por los procesos de Oracle (undo, redo, estadísticas, etc.) se recomienda:

Tablespace de datos: 100 MB inicial, autoextensión hasta 1 GB.

Tablespace temporal: 50 MB inicial, autoextensión hasta 512 MB.

Paso 6: Monitorizar y ajustar

Se recomienda usar vistas como:

DBA\_DATA\_FILES, V\$DATAFILE → espacio real usado.

DBA\_TEMP\_FILES, V\$TEMPFILE → uso del tablespace temporal.

Ajustar según crecimiento real.

*2. Elaborar el tablespace para los datos y el tablespace temporal del usuario*

```

ALTER SESSION SET CONTAINER = XEPDB1;

-- Creación de tablespace de datos
CREATE TABLESPACE ts_esports_datos
DATAFILE 'C:\Apps\Oracle21c\Oracle\ts_esports_datos01.dbf'
SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE 1024M;

-- Creación de tablespace temporal
CREATE TEMPORARY TABLESPACE ts_esports_temporal
TEMPFILE 'C:\Apps\Oracle21c\Oracle\ts_esports_temporal01.dbf'
SIZE 50M AUTOEXTEND ON NEXT 10M MAXSIZE 512M;

-- Creación de usuario 'esports_db'
CREATE USER esports_db IDENTIFIED BY esports_db
DEFAULT TABLESPACE ts_esports_datos
TEMPORARY TABLESPACE ts_esports_temporal;

GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SEQUENCE,
      CREATE TRIGGER, CREATE PROCEDURE TO esports_db;

ALTER USER esports_db QUOTA UNLIMITED ON ts_esports_datos;

```

3. *Elaborar el script en SQL de creación de los objetos de la base de datos (tablas, etc) en base al enunciado anterior.*

-- Creación de tabla EQUIPO\_ESPORTS

```
CREATE TABLE equipo_esports (  
    id_equipo NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY  
    KEY,  
    nombre VARCHAR2(100) NOT NULL,  
    region VARCHAR2(50),  
    manager VARCHAR2(100)  
)  
TABLESPACE ts_esports_datos;
```

-- Creación de tabla JUGADOR

```
CREATE TABLE jugador (  
    id_jugador NUMBER GENERATED BY DEFAULT AS IDENTITY  
    PRIMARY KEY,  
    nombre VARCHAR2(100) NOT NULL,  
    nacionalidad VARCHAR2(50),  
    fecha_nacimiento DATE  
)  
TABLESPACE ts_esports_datos;
```

-- Creación de tabla CONTRATO\_JUGADOR

```
CREATE TABLE contrato_jugador (  
    id_contrato NUMBER GENERATED BY DEFAULT AS IDENTITY  
    PRIMARY KEY,  
    id_jugador NUMBER NOT NULL,  
    id_equipo NUMBER NOT NULL,  
    fecha_inicio DATE NOT NULL,  
    fecha_fin DATE,
```

```
    CONSTRAINT fk_contrato_jugador FOREIGN KEY (id_jugador)
REFERENCES jugador(id_jugador),

    CONSTRAINT fk_contrato_equipo_esports FOREIGN KEY (id_equipo)
REFERENCES equipo_esports(id_equipo)

)

TABLESPACE ts_esports_datos;
```

-- Creación de tabla TORNEO

```
CREATE TABLE torneo (

    id_torneo NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY
KEY,

    nombre VARCHAR2(120) NOT NULL,

    anio NUMBER(4) NOT NULL,

    juego VARCHAR2(50),

    premio_total NUMBER(10,2),

    id_jugador_ganador NUMBER,

    CONSTRAINT fk_torneo_ganador FOREIGN KEY
(id_jugador_ganador) REFERENCES jugador(id_jugador)

)

TABLESPACE ts_esports_datos;
```

-- Creación de tabla PARTICIPACION\_EQUIPO\_TORNEO

```
CREATE TABLE participacion_equipo_torneo (

    id_equipo NUMBER NOT NULL,

    id_torneo NUMBER NOT NULL,

    puesto_final NUMBER,

    PRIMARY KEY (id_equipo, id_torneo),
```

```
CONSTRAINT fk_part_equipo_esports FOREIGN KEY (id_equipo)
REFERENCES equipo_esports(id_equipo),

CONSTRAINT fk_part_torneo FOREIGN KEY (id_torneo)
REFERENCES torneo(id_torneo)

)

TABLESPACE ts_esports_datos;
```

-- Índices para optimización

```
CREATE INDEX idx_contrato_jugador ON
contrato_jugador(id_jugador);

CREATE INDEX idx_contrato_equipo ON contrato_jugador(id_equipo);

CREATE INDEX idx_participacion_torneo ON
participacion_equipo_torneo(id_torneo);
```

*4. Elaborar el script en SQL de inserción y consultas para la base de datos.*

-- Equipos

```
INSERT INTO equipo_esports (nombre, region, manager) VALUES
('Team Phoenix', 'América Latina', 'Daniel Rivas');
```

```
INSERT INTO equipo_esports (nombre, region, manager) VALUES
('Digital Sharks', 'Europa', 'Alice King');
```

```
INSERT INTO equipo_esports (nombre, region, manager) VALUES
('Shadow Legion', 'Norteamérica', 'Ezequiel Paz');
```

-- Jugadores

```
INSERT INTO jugador (nombre, nacionalidad, fecha_nacimiento)
VALUES ('GamerTag: Blaze', 'Chile', DATE '2000-05-12');
```

```
INSERT INTO jugador (nombre, nacionalidad, fecha_nacimiento)
VALUES ('GamerTag: Viper', 'España', DATE '2002-08-03');
```

```
INSERT INTO jugador (nombre, nacionalidad, fecha_nacimiento)
VALUES ('GamerTag: Echo', 'México', DATE '2001-02-21');
```

-- Contratos

```
INSERT INTO contrato_jugador (id_jugador, id_equipo, fecha_inicio,
fecha_fin) VALUES (1, 1, DATE '2025-01-15', NULL);
```

```
INSERT INTO contrato_jugador (id_jugador, id_equipo, fecha_inicio,
fecha_fin) VALUES (2, 1, DATE '2024-10-01', NULL);
```

```
INSERT INTO contrato_jugador (id_jugador, id_equipo, fecha_inicio,
fecha_fin) VALUES (3, 2, DATE '2025-02-05', NULL);
```

-- Contrato Histórico de Blaze

```
INSERT INTO contrato_jugador (id_jugador, id_equipo, fecha_inicio,
fecha_fin) VALUES (4, 3, DATE '2024-03-01', DATE '2024-12-31');
```

-- Torneos

```
INSERT INTO torneo (nombre, anio, juego, premio_total,
id_jugador_ganador) VALUES ('Championship Global', 2025, 'Stratégia
X', 500000.00, 1);
```

```
INSERT INTO torneo (nombre, anio, juego, premio_total,
id_jugador_ganador) VALUES ('Copa Regional', 2024, 'Fighter Z',
150000.00, 2);
```

-- Participaciones

```
INSERT INTO participacion_equipo_torneo (id_equipo, id_torneo,
puesto_final) VALUES (1, 1, 2);
```

```
INSERT INTO participacion_equipo_torneo (id_equipo, id_torneo,
puesto_final) VALUES (2, 1, 1);
```

```
INSERT INTO participacion_equipo_torneo (id_equipo, id_torneo,
puesto_final) VALUES (3, 2, 1);
```

```
COMMIT;
```

## CONSULTAS

a) Listar a todos los ciclistas con su equipo actual

SELECT

j.nombre AS jugador,

e.nombre AS equipo,

cj.fecha\_inicio,

cj.fecha\_fin

FROM

jugador j

JOIN

contrato\_jugador cj ON j.id\_jugador = cj.id\_jugador

JOIN

equipo\_esports e ON cj.id\_equipo = e.id\_equipo

The screenshot shows a software interface with two tabs: 'Hoja de Trabajo' and 'Generador de Consultas'. The 'Generador de Consultas' tab is active, displaying a SQL query. Below the query editor, there are two sub-tabs: 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, showing a table with 4 columns: 'JUGADOR', 'EQUIPO', 'FECHA\_INICIO', and 'FECHA\_FIN'. The table contains 3 rows of data. Above the table, a status bar indicates 'Todas las Filas Recuperadas: 3 en 0.002 segundos'.

```
SELECT
    j.nombre AS jugador,
    e.nombre AS equipo,
    cj.fecha_inicio,
    cj.fecha_fin
FROM
    jugador j
JOIN
    contrato_jugador cj ON j.id_jugador = cj.id_jugador
JOIN
    equipo_esports e ON cj.id_equipo = e.id_equipo
```

	JUGADOR	EQUIPO	FECHA_INICIO	FECHA_FIN
1	GamerTag: Blaze	Team Phoenix	15/01/25	(null)
2	GamerTag: Viper	Digital Sharks	01/10/24	(null)
3	GamerTag: Echo	Digital Sharks	05/02/25	(null)



b) Mostrar las pruebas con su ganador

SELECT

t.nombre AS torneo,

t.anio,

j.nombre AS jugador\_ganador

FROM

torneo t

JOIN

jugador j ON j.id\_jugador = t.id\_jugador\_ganador;

Hoja de Trabajo		Generador de Consultas	
		<pre>SELECT     t.nombre AS torneo,     t.anio,     j.nombre AS jugador_ganador FROM     torneo t JOIN     jugador j ON j.id_jugador = t.id_jugador_ganador;</pre>	
Salida de Script x		Resultado de la Consulta x	
		SQL   Todas las Filas Recuperadas: 2 en 0.002 segundos	
TORNEO		ANIO	JUGADOR_GANADOR
1	Championship Global	2025	GamerTag: Viper
2	Copa Regional	2024	GamerTag: Blaze

c) Listar equipos con las pruebas en las que participaron y su puesto final

SELECT

e.nombre AS equipo,

t.nombre AS torneo,

t.anio,

pt.puesto\_final

FROM

equipo\_esports e

JOIN

participacion\_equipo\_torneo pt ON e.id\_equipo = pt.id\_equipo

JOIN

torneo t ON pt.id\_torneo = t.id\_torneo

ORDER BY

t.anio DESC, pt.puesto\_final ASC;

Hoja de Trabajo

Generador de Consultas

```

SELECT
    e.nombre AS equipo,
    t.nombre AS torneo,
    t.anio,
    pt.puesto_final
FROM
    equipo_esports e
JOIN
    participacion_equipo_torneo pt ON e.id_equipo = pt.id_equipo
JOIN
    torneo t ON pt.id_torneo = t.id_torneo
ORDER BY
    t.anio DESC, pt.puesto_final ASC;

```

Salida de Script x

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 3 en 0.003 segundos

EQUIPO	TORNEO	ANIO	PUESTO_FINAL
1 Digital Sharks	Championship Global	2025	1
2 Team Phoenix	Championship Global	2025	2
3 Shadow Legion	Copa Regional	2024	1

d) Contar cuántos ciclistas tiene cada equipo

SELECT

e.nombre AS equipo,

COUNT(cj.id\_jugador) AS cantidad\_jugadores

FROM

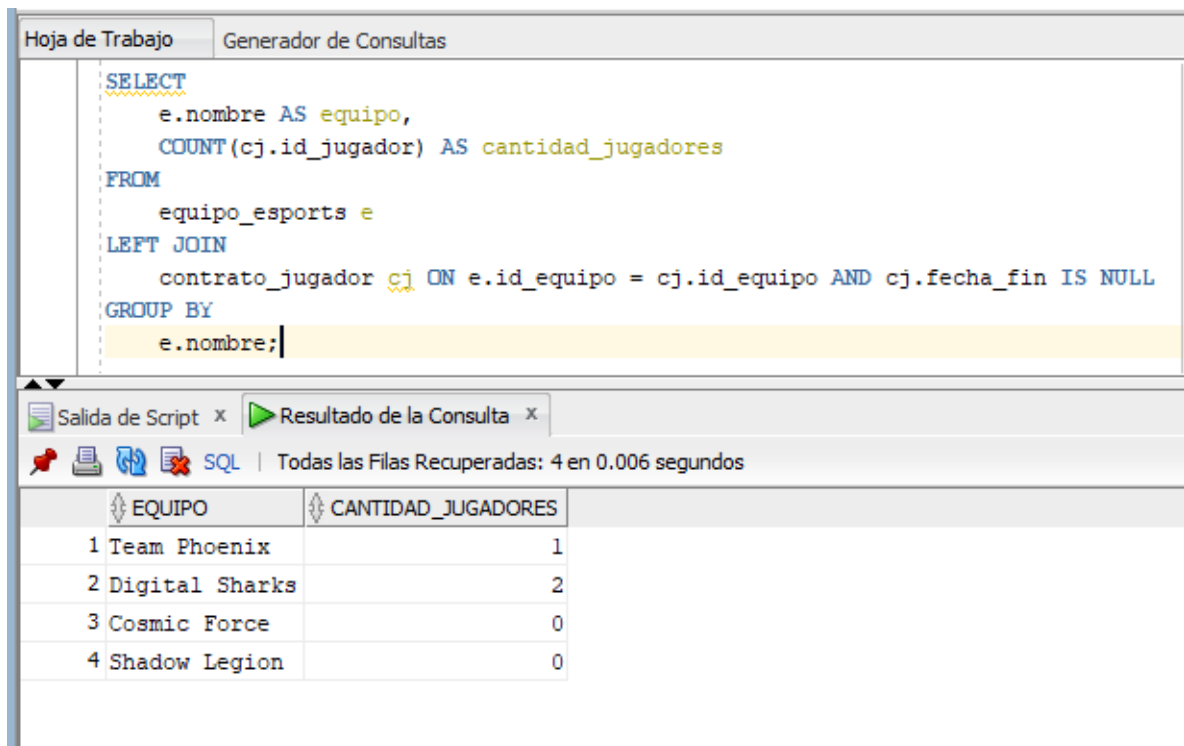
equipo\_esports e

LEFT JOIN

contrato\_jugador cj ON e.id\_equipo = cj.id\_equipo AND cj.fecha\_fin IS NULL

GROUP BY

e.nombre;



The screenshot shows a SQL query editor with two tabs: 'Hoja de Trabajo' and 'Generador de Consultas'. The query is written in the 'Generador de Consultas' tab. Below the query, there are two tabs: 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'EQUIPO' and 'CANTIDAD\_JUGADORES'. The table contains four rows of data.

```
SELECT
    e.nombre AS equipo,
    COUNT(cj.id_jugador) AS cantidad_jugadores
FROM
    equipo_esports e
LEFT JOIN
    contrato_jugador cj ON e.id_equipo = cj.id_equipo AND cj.fecha_fin IS NULL
GROUP BY
    e.nombre;
```

EQUIPO	CANTIDAD_JUGADORES
1 Team Phoenix	1
2 Digital Sharks	2
3 Cosmic Force	0
4 Shadow Legion	0