

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



“LABORATORIO 01”

ASIGNATURA

Administración de Base de Datos

DOCENTE

Chávez Soto, Jorge Luis

ESTUDIANTE

Carhuaricra Anco, Heidy Nicole - 23200150

Lima, Perú

2025

ÍNDICE

LABORATORIO 01.....	4
I. OBJETIVOS	4
II. RESUMEN	4
III. INFORMACIÓN DE LA BASE DE DATOS	4
IV. CONSTRUCCIÓN DE LA APLICACIÓN.....	5
V. ACTIVIDADES QUE REALIZAR.....	5

LABORATORIO 01

I. OBJETIVOS

El presente laboratorio tiene por objetivos:

- Determine el sizing de la base de datos.
- Elabore los tablespaces de la base de datos.
- Elabore los scripts SQL de creación de los objetos de la base de datos.
- Elabore los scripts SQL de inserción y consulta.

II. RESUMEN

En este ejercicio planteado, se revisará un enunciado de un negocio para determinar el tamaño ideal de la base de datos para un ciclo de vida de 5 años. Crear los tablespaces de acuerdo con el tamaño planteado para su ciclo de vida. Elaborar los scripts de creación de los objetos de la base de datos en base al enunciado planteado. Finalmente, los scripts de inserción de los datos en la base de datos, junto con las consultas más frecuentes de la base de datos planteada.

III. INFORMACIÓN DE LA BASE DE DATOS

El "sizing" de una base de datos es un proceso para estimar sus necesidades de almacenamiento y planificar la capacidad de hardware necesaria, considerando el volumen de datos actual, la tasa de crecimiento esperada y el tamaño de las tablas e índices. Implica calcular el tamaño de cada tabla, sumar sus volúmenes y proyectar el crecimiento futuro para asegurar el rendimiento y prevenir problemas de espacio. Pasos para el dimensionamiento de una base de datos:

1. **Evaluar el volumen actual de datos:** Determine la cantidad de datos existentes en la base de datos y el tamaño de cada tabla y columna.
2. **Estimar la tasa de crecimiento:** Calcule la velocidad a la que se añadirán nuevos datos, lo que es clave para predecir el crecimiento futuro.
3. **Considerar los índices:** Los índices aumentan el tamaño de la base de datos, por lo que su tipo y cantidad también deben ser considerados en el cálculo.
4. **Calcular el tamaño total:** Suma los tamaños de todas las tablas, teniendo en cuenta los índices y el crecimiento futuro, para obtener una estimación del tamaño necesario.
5. **Planificar el espacio adicional:** Añade un espacio de contingencia para permitir flexibilidad y adaptarte a imprevistos.
6. **Monitorizar y ajustar:** Después de la implementación, monitoriza el uso real del almacenamiento y los datos para hacer ajustes y previsiones para compras futuras.
7. **Importancia del dimensionamiento:**

- ✓ *Planificación de la capacidad:* Permite determinar los requisitos de hardware (CPU, memoria, disco) de manera anticipada.
- ✓ *Optimización del rendimiento:* Un dimensionamiento adecuado evita cuellos de botella y asegura un buen tiempo de respuesta para las consultas.
- ✓ *Prevención de problemas:* Ayuda a anticipar problemas de espacio y evitar la pérdida de datos por desbordamiento de capacidad.
- ✓ *Toma de decisiones:* Proporciona información valiosa para planificar el almacenamiento y el diseño de la base de datos de manera eficiente.

IV. CONSTRUCCIÓN DE LA APLICACIÓN

- **El enunciado es:** Se desea construir una base de datos para mantener la información sobre las pruebas ciclistas por etapas.
 - ✓ La información sobre los ciclistas, los equipos a los que pertenecen y las pruebas en las que cada equipo ha participado (se asume que participa todo el equipo).
 - ✓ De cada ciclista, se desea conocer su nombre, nacionalidad y fecha de nacimiento, así como el equipo al que pertenece, manteniendo la fecha de inicio y fin de contrato con el equipo.
 - ✓ De cada equipo también se desea conocer su nombre, su nacionalidad, el nombre del director y las pruebas en las que ha participado, con su nombre, año de edición, número de etapas, kilómetros totales y puesto que ocupó el equipo en la clasificación final. Un dato adicional para las pruebas es saber el nombre del ciclista que quedó ganador.
- **Tarea para realizar:** Estas son las entregables de la tarea a realizar:
 - ✓ Elaborar el sizing de la base de datos a crear para un periodo de 5 años.
 - ✓ Elaborar el tablespace para los datos y el tablespace temporal del usuario.
 - ✓ Elaborar el script en SQL de creación de los objetos de la base de datos (tablas, etc) en base al enunciado anterior.
 - ✓ Elaborar el script en SQL de inserción y consultas para la base de datos.

V. ACTIVIDADES QUE REALIZAR

- Realizamos el modelo físico para poder organizar, más adelante, mejor nuestros tablespaces y sizing.

Ciclista

- ✓ IdCiclista : VARCHAR (20) - (PK)
- ✓ Nombre: VARCHAR (50)
- ✓ Apellido: VARCHAR (100)
- ✓ Nacionalidad: VARCHAR (50)
- ✓ FechaNacimiento: DATE

Equipo

- ✓ IdEquipo: NUMBER (20) - (PK)
- ✓ Nombre: VARCHAR (100)
- ✓ Nacionalidad: VARCHAR (100)
- ✓ Director: VARCHAR (100)

Contrato

- ✓ IdContrato: NUMBER (20) - (PK)
- ✓ IdCiclista: VARCHAR (20) - (FK)
- ✓ IdEquipo: NUMBER (20) - (FK)
- ✓ FechaInicio: DATE
- ✓ FechaFin: DATE

Campeonato

- ✓ IdCampeonato: NUMBER (20) - (PK)
- ✓ Nombre: VARCHAR (150)
- ✓ Año: INTEGER
- ✓ NumEdicion: INTEGER

Prueba

- ✓ IdPrueba: NUMBER (20) - (PK)
- ✓ IdCampeonato: NUMBER (20) - (FK)
- ✓ IdCiclistaGanador: VARCHAR (20) - (FK)
- ✓ Nombre: VARCHAR (100)
- ✓ KmTotales: NUMBER (8, 2)
- ✓ Etapas: INTEGER

Campeonato_Equipo

- ✓ IdCampeonato: NUMBER (20) - (FK) (PK)
- ✓ IdEquipo: NUMBER (20) - (FK) (PK)
- ✓ PuestoFinal: INTEGER

- Elaboramos el sizing, teniendo en cuenta la cantidad de registros esperados por tabla, tamaño de cada registro (según tipo de dato), índices y crecimiento proyecto, y un espacio extra de contingencia.

Estimar cantidad de registros

- ✓ Ciclista: 500 en el año inicial con un incremento de 100 ciclistas nuevos por año. → 3500 ciclistas en 5 años.
- ✓ Equipo: En el caso que un equipo esté conformado por 10 ciclistas, serían 50 equipos en el año inicial con un incremento de 10 equipos nuevos por año. → 350 equipos en 5 años.
- ✓ Contrato: Cada ciclista tiene al menos 1 contrato en 5 años. → 7000 contratos en 5 años (Suponiendo que la mayoría de ciclistas decidan cambiar de equipo).
- ✓ Campeonato: En el caso que haya 2 campeonatos por año. → 10 campeonatos en 5 años.
- ✓ Prueba: Cada campeonato tiene 20 pruebas y son 10 campeonatos en total. → 200 pruebas en 5 años.
- ✓ Campeonato_Equipo: 50 equipos en el año inicial con un incremento de 10 equipos nuevos por año X 2 campeonatos por año → 700 registros en 5 años.

Tamaño de registros

Ciclista ≈ 447 bytes/tupla

- ✓ IdCiclista : VARCHAR (20) - 40 bytes
- ✓ Nombre: VARCHAR (50) - 100 bytes
- ✓ Apellido: VARCHAR (100) - 200 bytes
- ✓ Nacionalidad: VARCHAR (50) - 100 bytes
- ✓ FechaNacimiento: DATE – 7 bytes

Equipo ≈ 520 bytes/tupla

- ✓ IdEquipo: NUMBER (20) - 20 bytes
- ✓ Nombre: VARCHAR (100) - 200 bytes
- ✓ Nacionalidad: VARCHAR (50) - 100 bytes
- ✓ Director: VARCHAR (100) - 200 bytes

Contrato ≈ 94 bytes/tupla

- ✓ IdContrato: NUMBER (20) - 20 bytes
- ✓ IdCiclista: VARCHAR (20) - 40 bytes
- ✓ IdEquipo: NUMBER (20) - 20 bytes
- ✓ FechaInicio: DATE – 7 bytes
- ✓ FechaFin: DATE – 7 bytes

Campeonato ≈ 340 bytes/tupla

- ✓ IdCampeonato: NUMBER (20) - 20 bytes
- ✓ Nombre: VARCHAR (150) - 300 bytes

- ✓ Año: INTEGER – 10 bytes
- ✓ NumEdicion: INTEGER – 10 bytes

Prueba \approx 302 bytes/tupla

- ✓ IdPrueba: NUMBER (20) - 20 bytes
- ✓ IdCampeonato: NUMBER (20) - 20 bytes
- ✓ IdCiclistaGanador: VARCHAR (20) - 40 bytes
- ✓ Nombre: VARCHAR (100) - 200 bytes
- ✓ KmTotales: NUMBER (8, 2) - 12 bytes
- ✓ Etapas: INTEGER – 10 bytes

Campeonato_Equipo \approx 50 bytes/tupla

- ✓ IdCampeonato: NUMBER (20) - 20 bytes
- ✓ IdEquipo: NUMBER (20) - 20 bytes
- ✓ PuestoFinal: INTEGER – 10 bytes

Total

- ✓ Ciclista: $447 \frac{\text{bytes}}{\text{tupla}} \times 3500 \text{ tuplas} = 1,564,500 \text{ bytes} \approx 1.6 \text{ Mb}$
- ✓ Equipo: $520 \frac{\text{bytes}}{\text{tupla}} \times 350 \text{ tuplas} = 182,000 \text{ bytes} \approx 182 \text{ Kb}$
- ✓ Contrato: $94 \frac{\text{bytes}}{\text{tupla}} \times 7000 \text{ tuplas} = 658,000 \text{ bytes} \approx 658 \text{ Kb}$
- ✓ Campeonato: $340 \frac{\text{bytes}}{\text{tupla}} \times 10 \text{ tuplas} = 3400 \text{ bytes} \approx 3.4 \text{ Kb}$
- ✓ Prueba: $302 \frac{\text{bytes}}{\text{tupla}} \times 200 \text{ tuplas} = 60,400 \text{ bytes} \approx 60.4 \text{ Kb}$
- ✓ Campeonato_Equipo: $50 \frac{\text{bytes}}{\text{tupla}} \times 700 \text{ tuplas} = 35,000 \text{ bytes} \approx 35 \text{ Kb}$

Total Bruto = 1.6 Mb + 182 Kb + 658 Kb + 3.4 Kb + 60.4 Kb + 35 Kb \approx 2538.8 Kb

Total Bruto \approx 2.5Mb

- ✓ Índices: 30% del Total Bruto = 2.5 Mb x 0.3 \approx 0.75 Mb

Total bruto + Índices = 2.5 Mb + 0.75 Mb = 3.25 Mb

- ✓ Contingencia: 50% (Total Bruto + Índices) = 3.25 Mb x 0.5 \approx 1.625 Mb

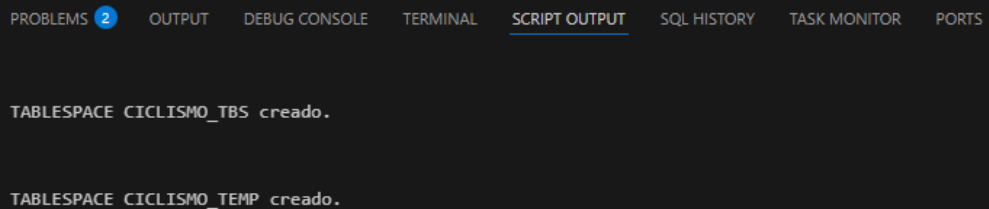
Total estimado = Total bruto + Índices + Contingencia = 3.25 Mb + 1.625 Mb = 4.875 Mb

Total estimado \approx 5 Mb

- Ahora, ejecutamos los scripts de creación de tablespaces permanentes para los datos y tablespaces temporales para el usuario. Asignamos un tamaño mayor a lo estimado por precaución.

```
---Creación de tablespace
CREATE TABLESPACE ciclismo_tbs
  DATAFILE 'ciclismo_tbs01.dbf'
  SIZE 10M AUTOEXTEND ON NEXT 5M
  MAXSIZE 100M EXTENT MANAGEMENT LOCAL
  SEGMENT SPACE MANAGEMENT AUTO;

---Creación de tablespace temporal
CREATE TEMPORARY TABLESPACE ciclismo_temp
  TEMPFILE 'ciclismo_temp01.dbf'
  SIZE 10M AUTOEXTEND ON NEXT 5M
  MAXSIZE 100M EXTENT MANAGEMENT LOCAL;
```



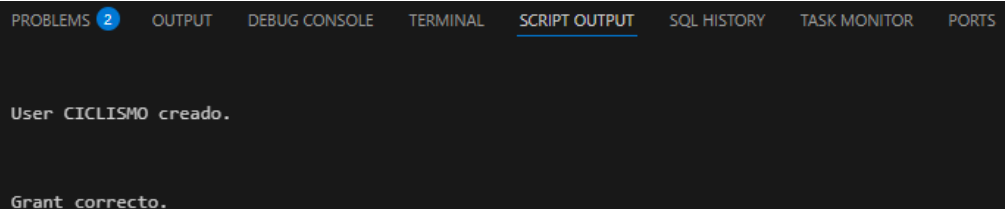
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

TABSPACE CICLISMO_TBS creado.

TABSPACE CICLISMO_TEMP creado.

```
---Creación de usuario CICLISMO
CREATE USER ciclismo IDENTIFIED BY ciclismo
  DEFAULT TABLESPACE ciclismo_tbs
  TEMPORARY TABLESPACE ciclismo_temp
  QUOTA UNLIMITED ON ciclismo_tbs;

---Otorgar permisos al usuario
GRANT CONNECT, RESOURCE TO ciclismo;
```



PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

User CICLISMO creado.

Grant correcto.

---Creación de tablespace Untitled-1 • New connection: CICLISMO_BD X

Create Connection CICLISMO_BD

Connection Name
CICLISMO_BD

User Info Proxy User

Authentication Type Default Role Default

Username ciclismo Password *****

☒ Save Password

Connection Type
Basic

Details Advanced

Hostname localhost Port 1521

Type Service Name XEPDB1

Cancel Test Connect Save

---Creación de tablas

```
CREATE TABLE CICLISTA (
    ID_CICLISTA      VARCHAR(20) PRIMARY KEY,
    NOMBRE           VARCHAR(50) NOT NULL,
    APELLIDO         VARCHAR(100) NOT NULL,
    NACIONALIDAD     VARCHAR(50),
    FECHA_NACIMIENTO DATE
);

CREATE TABLE EQUIPO (
    ID_EQUIPO        NUMBER(20) PRIMARY KEY,
    NOMBRE           VARCHAR(100) NOT NULL,
    DIRECTOR         VARCHAR(100) NOT NULL,
    NACIONALIDAD     VARCHAR(50)
);

CREATE TABLE CONTRATO (
    ID_CONTRATO      NUMBER(20) PRIMARY KEY,
    FECHA_INICIO     DATE,
    FECHA_FIN        DATE,
    ID_CICLISTA      VARCHAR(20) NOT NULL,
    ID_EQUIPO        NUMBER(20) NOT NULL,
    CONSTRAINT FK_CONTRATO_CICLISTA FOREIGN KEY (ID_CICLISTA)
REFERENCES CICLISTA(ID_CICLISTA),
    CONSTRAINT FK_CONTRATO_EQUIPO FOREIGN KEY (ID_EQUIPO)
REFERENCES EQUIPO(ID_EQUIPO)
);

CREATE TABLE CAMPEONATO (
    ID_CAMPEONATO    NUMBER(20) PRIMARY KEY,
    NOMBRE           VARCHAR(150) NOT NULL,
```

```

        ANIO                INTEGER NOT NULL,
        NUM_EDICION         INTEGER NOT NULL
    );

CREATE TABLE PRUEBA (
    ID_PRUEBA                NUMBER(20) PRIMARY KEY,
    NOMBRE                   VARCHAR(100) NOT NULL,
    KM_TOTALES               NUMBER(8, 2) NOT NULL,
    ETAPAS                   INTEGER NOT NULL,
    ID_CAMPEONATO            NUMBER(20) NOT NULL,
    ID_CICLISTA_GANADOR     VARCHAR(20) NOT NULL,
    CONSTRAINT FK_PRUEBA_CAMPEONATO FOREIGN KEY (ID_CAMPEONATO)
REFERENCES CAMPEONATO(ID_CAMPEONATO),
    CONSTRAINT FK_PRUEBA_GANADOR FOREIGN KEY (ID_CICLISTA_GANADOR)
REFERENCES CICLISTA(ID_CICLISTA)
);

CREATE TABLE CAMPEONATO_EQUIPO (
    ID_CAMPEONATO            NUMBER(20) NOT NULL,
    ID_EQUIPO                NUMBER(20) NOT NULL,
    PUESTO_FINAL             INTEGER,
    CONSTRAINT PK_CAMPEONATO_EQUIPO PRIMARY KEY (ID_CAMPEONATO,
ID_EQUIPO),
    CONSTRAINT FK_CE_CAMPEONATO FOREIGN KEY (ID_CAMPEONATO) REFERENCES
CAMPEONATO(ID_CAMPEONATO),
    CONSTRAINT FK_CE_EQUIPO FOREIGN KEY (ID_EQUIPO) REFERENCES
EQUIPO(ID_EQUIPO)
);

```

The image displays three sequential screenshots of a database IDE's 'SCRIPT OUTPUT' tab. Each screenshot shows the successful execution of SQL commands to create tables. The first screenshot shows 'Table CICLISTA creado.' and 'Table EQUIPO creado.'. The second screenshot shows 'Table CONTRATO creado.' and 'Table CAMPEONATO creado.'. The third screenshot shows 'Table PRUEBA creado.' and 'Table CAMPEONATO_EQUIPO creado.'. Each screenshot has a tab bar at the top with 'PROBLEMS 2', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'SCRIPT OUTPUT' (which is selected and underlined), followed by a three-dot menu icon.

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL SCRIPT OUTPUT ...

Table CICLISTA creado.

Table EQUIPO creado.

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL SCRIPT OUTPUT ...

Table CONTRATO creado.

Table CAMPEONATO creado.

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL SCRIPT OUTPUT ...

Table PRUEBA creado.

Table CAMPEONATO_EQUIPO creado.

```

```

---Scripts de inserción de datos
INSERT INTO CICLISTA (ID_CICLISTA, NOMBRE, APELLIDO, NACIONALIDAD,
FECHA_NACIMIENTO)
VALUES ('C0000001', 'Pedro', 'Ramírez', 'Colombia',
TO_DATE('1995-03-21', 'YYYY-MM-DD'));
INSERT INTO CICLISTA (ID_CICLISTA, NOMBRE, APELLIDO, NACIONALIDAD,
FECHA_NACIMIENTO)
VALUES ('C0000002', 'Luis', 'Fernández', 'España',
TO_DATE('1998-07-10', 'YYYY-MM-DD'));
INSERT INTO CICLISTA (ID_CICLISTA, NOMBRE, APELLIDO, NACIONALIDAD,
FECHA_NACIMIENTO)
VALUES ('C0000003', 'Marco', 'Gómez', 'Perú', TO_DATE('1993-01-
15', 'YYYY-MM-DD'));
INSERT INTO CICLISTA (ID_CICLISTA, NOMBRE, APELLIDO, NACIONALIDAD,
FECHA_NACIMIENTO)
VALUES ('C0000004', 'Keysi', 'Anco', 'Perú', TO_DATE('2000-06-
25', 'YYYY-MM-DD'));
INSERT INTO CICLISTA (ID_CICLISTA, NOMBRE, APELLIDO, NACIONALIDAD,
FECHA_NACIMIENTO)
VALUES ('C0000005', 'Thais', 'Chavez', 'Brasil', TO_DATE
('1992-07-20', 'YYYY-MM-DD'));

INSERT INTO EQUIPO (ID_EQUIPO, NOMBRE, NACIONALIDAD, DIRECTOR)
VALUES ('1234', 'Team Andes', 'Ecuador', 'Carlos Reynoso');
INSERT INTO EQUIPO (ID_EQUIPO, NOMBRE, NACIONALIDAD, DIRECTOR)
VALUES ('1235', 'Iberia Cycling', 'Inglaterra', 'Javier
Martin');
INSERT INTO EQUIPO (ID_EQUIPO, NOMBRE, NACIONALIDAD, DIRECTOR)
VALUES ('1236', 'Sporting Ciclismo', 'Canadá', 'Ricardo
Gareca');

INSERT INTO CONTRATO (ID_CONTRATO, FECHA_INICIO, FECHA_FIN,
ID_CICLISTA, ID_EQUIPO)
VALUES ('123456', TO_DATE('2022-01-01', 'YYYY-MM-DD'),
TO_DATE('2025-12-31', 'YYYY-MM-DD'), 'C0000004', '1235');
INSERT INTO CONTRATO (ID_CONTRATO, FECHA_INICIO, FECHA_FIN,
ID_CICLISTA, ID_EQUIPO)
VALUES ('123457', TO_DATE('2023-06-01', 'YYYY-MM-DD'),
TO_DATE('2024-06-01', 'YYYY-MM-DD'), 'C0000002', '1236');
INSERT INTO CONTRATO (ID_CONTRATO, FECHA_INICIO, FECHA_FIN,
ID_CICLISTA, ID_EQUIPO)
VALUES ('123458', TO_DATE('2024-01-01', 'YYYY-MM-DD'),
TO_DATE('2024-06-01', 'YYYY-MM-DD'), 'C0000001', '1234');

INSERT INTO CAMPEONATO (ID_CAMPEONATO, NOMBRE, ANIO, NUM_EDICION)
VALUES('11111111', 'Vuelta Andina', 2024, 4);

INSERT INTO PRUEBA (ID_PRUEBA, NOMBRE, KM_TOTALES, ETAPAS,
ID_CAMPEONATO, ID_CICLISTA_GANADOR)
VALUES('4567', 'Prueba Montaña', 120.5, 1, '11111111',
'C0000004');

```

```

INSERT INTO PRUEBA (ID_PRUEBA, NOMBRE, KM_TOTALES, ETAPAS,
ID_CAMPEONATO, ID_CICLISTA_GANADOR)
VALUES('4568', 'Prueba Ciudad', 50.0, 1, '11111111',
'C0000001');
INSERT INTO PRUEBA (ID_PRUEBA, NOMBRE, KM_TOTALES, ETAPAS,
ID_CAMPEONATO, ID_CICLISTA_GANADOR)
VALUES('4569', 'Prueba Contrarreloj', 20.0, 1, '11111111',
'C0000004');

INSERT INTO CAMPEONATO_EQUIPO (ID_CAMPEONATO, ID_EQUIPO,
PUESTO_FINAL)
VALUES ('11111111', '1235', 1);
INSERT INTO CAMPEONATO_EQUIPO (ID_CAMPEONATO, ID_EQUIPO,
PUESTO_FINAL)
VALUES ('11111111', '1236', 2);
INSERT INTO CAMPEONATO_EQUIPO (ID_CAMPEONATO, ID_EQUIPO,
PUESTO_FINAL)
VALUES ('11111111', '1234', 3);

```

---Scripts de prueba

---PRUEBA: Ver la tabla Ciclista

```
SELECT * FROM CICLISTA;
```

---PRUEBA: Ver la tabla Equipo

```
SELECT * FROM EQUIPO;
```

---PRUEBA: Ver la tabla Contrato

```
SELECT * FROM CONTRATO;
```

---PRUEBA: Ver la tabla Campeonato

```
SELECT * FROM CAMPEONATO;
```

---PRUEBA: Ver la tabla Prueba

```
SELECT * FROM PRUEBA;
```

---PRUEBA: Ver la tabla Campeonato_Equipo

```
SELECT * FROM CAMPEONATO_EQUIPO;
```

ID_CICLISTA	NOMBRE	APELLIDO	NACIONALIDAD	FECHA_NACIMIENTO
1 C0000001	Pedro	Ramírez	Colombia	21/03/95
2 C0000002	Luis	Fernández	España	10/07/98
3 C0000003	Marco	Gómez	Perú	15/01/93
4 C0000004	Keysi	Anco	Perú	25/06/00
5 C0000005	Thais	Chavez	Brasil	20/07/92

All rows fetched: 5 in 0.194 seconds

ID_EQUIPO	NOMBRE	DIRECTOR	NACIONALIDAD
1 1234	Team Andes	Carlos Reynoso	Ecuador
2 1235	Iberia Cycling	Javier Martin	Inglaterra
3 1236	Sporting Ciclismo	Ricardo Gareca	Canadá

All rows fetched: 3 in 0.126 seconds

ID_CONTRATO	FECHA_INICIO	FECHA_FIN	ID_CICLISTA	ID_EQUIPO
1 123456	01/01/22	31/12/25	C0000004	1235
2 123457	01/06/23	01/06/24	C0000002	1236
3 123458	01/01/24	01/06/24	C0000001	1234

ID_CAMPEONATO	NOMBRE	ANIO	NUM_EDICION
1 11111111	Vuelta Andina	2024	4

	ID_PRUEBA	NOMBRE	KM_TOTALES	ETAPAS	ID_CAMPEONATO	ID_CICLISTA_GANADOR
1	4567	Prueba Montaña	120.5	1	11111111	C0000004
2	4568	Prueba Ciudad	50	1	11111111	C0000001
3	4569	Prueba Contrarreloj	20	1	11111111	C0000004

	ID_CAMPEONATO	ID EQUIPO	PUESTO_FINAL
1	11111111	1235	1
2	11111111	1236	2
3	11111111	1234	3

```

---PRUEBA: Prueba(s) de un campeonato con su ciclista ganador
SELECT p.NOMBRE AS PRUEBA, p.KM_TOTALES, c.NOMBRE || ' ' ||
c.APELLIDO AS GANADOR
FROM PRUEBA p
JOIN CICLISTA c ON p.ID_CICLISTA_GANADOR = c.ID_CICLISTA
WHERE p.ID_CAMPEONATO = 11111111;

```

	PRUEBA	KM_TOTALES	GANADOR
1	Prueba Montaña	120.5	Keysi Anco
2	Prueba Ciudad	50	Pedro Ramírez
3	Prueba Contrarreloj	20	Keysi Anco

```

---PRUEBA: Ciclista(s) con contrato vigente
SELECT c.ID_CICLISTA, c.NOMBRE || ' ' || c.APELLIDO AS CICLISTA,
e.NOMBRE AS EQUIPO, ct.FECHA_INICIO, ct.FECHA_FIN
FROM CICLISTA c
JOIN CONTRATO ct ON c.ID_CICLISTA = ct.ID_CICLISTA
JOIN EQUIPO e ON ct.ID_EQUIPO = e.ID_EQUIPO
WHERE SYSDATE BETWEEN ct.FECHA_INICIO AND ct.FECHA_FIN;

```

	ID_CICLISTA	CICLISTA	EQUIPO	FECHA_INICIO	FECHA_FIN
1	C0000004	Keysi Anco	Iberia Cycling	01/01/22	31/12/25

```

---PRUEBA: Ranking final de equipos en el campeonato en orden
descendente
SELECT ce.PUESTO_FINAL, e.NOMBRE AS EQUIPO, c.NOMBRE AS CAMPEONATO
FROM CAMPEONATO_EQUIPO ce
JOIN EQUIPO e ON ce.ID_EQUIPO = e.ID_EQUIPO
JOIN CAMPEONATO c ON ce.ID_CAMPEONATO = c.ID_CAMPEONATO
WHERE ce.ID_CAMPEONATO = 11111111
ORDER BY ce.PUESTO_FINAL DESC;

```

	PUESTO_FINAL	EQUIPO	CAMPEONATO
1	3	Team Andes	Vuelta Andina
2	2	Sporting Ciclismo	Vuelta Andina
3	1	Iberia Cycling	Vuelta Andina

```

---PRUEBA: Campeonato(s) con la(s) prueba(s) que lo componen
SELECT ca.ID_CAMPEONATO, ca.NOMBRE AS CAMPEONATO, ca.ANIO,
ca.NUM_EDICION, COUNT(p.ID_PRUEBA) AS TOTAL_PRUEBAS
FROM CAMPEONATO ca
LEFT JOIN PRUEBA p ON ca.ID_CAMPEONATO = p.ID_CAMPEONATO
GROUP BY ca.ID_CAMPEONATO, ca.NOMBRE, ca.ANIO, ca.NUM_EDICION
ORDER BY ca.ANIO;

```

	ID_CAMPEONATO	CAMPEONATO	ANTO	NUM_EDICION	TOTAL_PRUEBAS
1	11111111	Vuelta Andina	2024	4	3