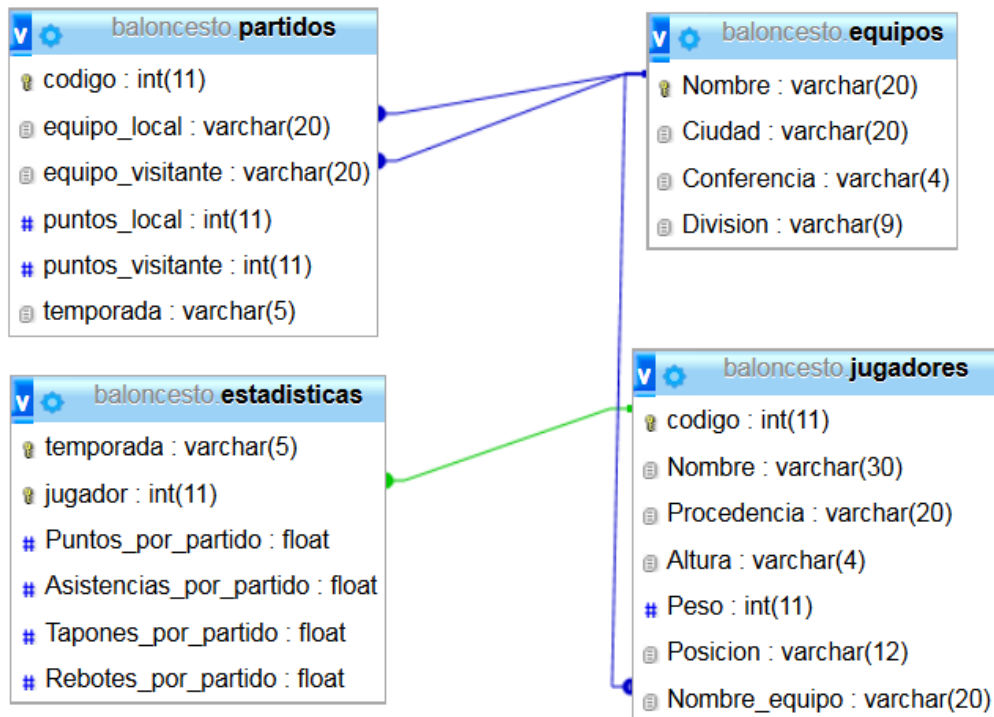


## EJERCICIOS RA2

Partimos de la base de datos MySQL **baloncesto**, usuario y clave **baloncesto** (debeis crearla y el usuario para acceder a la BD también). Para conectarnos se usará la siguiente conexión:

```
Connection conexion = DriverManager.getConnection  
("jdbc:mysql://localhost/baloncesto","baloncesto", "baloncesto");
```

A partir de las siguientes tablas de la liga de baloncesto de la NBA:



La descripción de las tablas es la siguiente:

**EQUIPOS** - Contiene información de los equipos que participan en la liga.

```
CREATE TABLE equipos (  
  Nombre varchar(20) NOT NULL,  
  Ciudad varchar(20) DEFAULT NULL,  
  Conferencia varchar(4) DEFAULT NULL,  
  Division varchar(9) DEFAULT NULL,  
  PRIMARY KEY (Nombre) )engine=innodb;
```

**JUGADORES** - Contiene información de los datos de los jugadores de los equipos. La altura viene en pies y el peso en libras. Al lado se indica la conversión a metros y gramos respectivamente.

```
CREATE TABLE jugadores (  
  codigo int NOT NULL,  
  Nombre varchar(30) DEFAULT NULL,  
  Procedencia varchar(20) DEFAULT NULL,  
  Altura varchar(4) DEFAULT NULL, -- en pies, 1 pie 0,3048 metros  
  Peso int DEFAULT NULL, -- en libras 1 libra 453.59 gramos  
  Posicion varchar(12) DEFAULT NULL,  
  Nombre_equipo varchar(20) DEFAULT NULL,  
  PRIMARY KEY (codigo),  
  FOREIGN KEY (Nombre_equipo) References equipos(Nombre)  
 )engine=innodb;
```

## EJERCICIOS RA2

**ESTADISTICAS** – Esta tabla contiene la información de las estadísticas de los jugadores. La media de puntos por temporada, las asistencias realizadas, los tapones, los rebotes, etc.

```
CREATE TABLE estadisticas (  
    temporada varchar(5) NOT NULL ,  
    jugador int NOT NULL ,  
    Puntos_por_partido number(5,2) DEFAULT NULL,  
    Asistencias_por_partido number(5,2) DEFAULT NULL,  
    Tapones_por_partido number(5,2) DEFAULT NULL,  
    Rebotes_por_partido number(5,2) DEFAULT NULL,  
    PRIMARY KEY (temporada,jugador),  
    FOREIGN KEY (jugador) REFERENCES Jugadores(Codigo)  
)engine=innodb;
```

**PARTIDOS** – Esta tabla contiene la información de los partidos disputados, los equipos y los puntos. Las columnas son:

```
CREATE TABLE partidos (  
    codigo int NOT NULL,  
    equipo_local varchar(20) DEFAULT NULL,  
    equipo_visitante varchar(20) DEFAULT NULL,  
    puntos_local number(5) DEFAULT NULL,  
    puntos_visitante number(5) DEFAULT NULL,  
    temporada varchar(5) DEFAULT NULL,  
    PRIMARY KEY (codigo),  
    FOREIGN KEY (equipo_local) REFERENCES equipos(nombre),  
    FOREIGN KEY (equipo_visitante) REFERENCES equipos(nombre)  
)engine=innodb;
```

Realiza un **programa Java** de nombre **EjercicioConsultas.java** que visualice los siguientes listados (cada listado en un método).

**1º)** En el primero se debe pedir por teclado un número de jugador y se mostrará el listado indicado a continuación ordenado por temporada. Si el número de jugador no existe se debe mostrar un mensaje indicándolo. Se realizará un proceso repetitivo hasta que el número de jugador sea 0. La ejecución se hará desde la línea de comandos del DOS. Ejemplo de ejecución:

-----  
Introduce código de jugador: 227

DATOS DEL JUGADOR: 227

Nombre : Kirk Hinrich

Equipo : Bulls

Temporada	Ptos	Asis	Tap	Reb
03/04	12,00	6,80	0,30	3,40
04/05	15,70	6,40	0,30	3,90
05/06	15,90	6,30	0,30	3,60
06/07	16,60	6,30	0,30	3,40
07/08	12,00	6,00	0,30	3,40

-----  
Num de registros: 5

-----  
Introduce código de jugador: 700

=====

EL JUGADOR NO EXISTE

-----  
Introduce código de jugador: 1

DATOS DEL JUGADOR: 1

Nombre : Corey Brever

## EJERCICIOS RA2

```
Equipo : Timberwolves
Temporada   Ptos   Asis   Tap   Reb
=====
04/05       8,00   0,00   4,00   0,00
05/06       5,60   1,40   0,30   3,60
=====
```

```
Num de registros: 2
```

```
-----
Introduce código de jugador: 0
-----
```

```
Fin listado 1 ....
-----
```

**2º)** Este segundo listado debe pedir por teclado un nombre de equipo y debe mostrar por cada equipo la lista de sus jugadores con la media de los puntos por partido. Se hará en un proceso repetitivo hasta que el nombre de equipo sea \*. La ejecución se hará desde la línea de comandos del DOS. El listado debe aparecer ordenado por equipo. Ejemplo de ejecución (empezando con el listado 1):

```
-----
Introduce código de jugador: 0
-----
```

```
Fin listado 1 ....
-----
```

```
Introduce nombre de equipo: 76ers
```

```
=====
Equipo: 76ers
120,   Louis Amundson           :   1,45
121,   Calvin Booth             :   3,51
122,   Rodney Carney            :   6,20
123,   Samuel Dalembert         :  11,67
124,   Reggie Evans             :   4,50
125,   Willie Green             :   9,04
126,   Herbert Hill             :   0,00
127,   Andre Iguodala           :  14,82
128,   Andre Miller             :  14,34
129,   Kevin Ollie              :   3,10
130,   Shavlik Randolph         :  65,67
131,   Jason Smith              :   4,50
132,   Louis Williams           :   5,90
133,   Thaddeus Young           :  7,80
=====
```

```
Número de JUGADORES: 14
```

```
-----
Introduce nombre de equipo: yure
=====
```

```
EL EQUIPO NO EXISTE
-----
```

```
Introduce nombre de equipo: *
-----
```

```
Fin listado 2 ....
-----
```

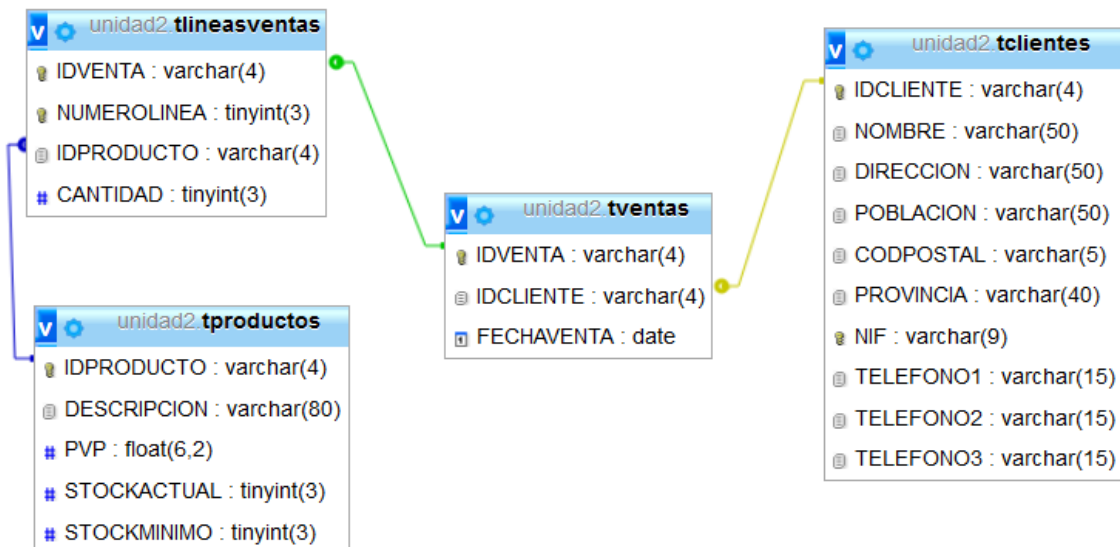
CREA EL FICHERO JAR PARA EJECUTAR EL EJERCICIO DESDE LA LINEA DE  
COMANDOS

## EJERCICIOS RA2

Partimos de la base de datos MySQL de nombre **unidad2**, usuario y clave **unidad2**. Para conectarnos usamos esta conexión:

```
Connection conexion = DriverManager.getConnection  
("jdbc:mysql://localhost/unidad2","unidad2", "unidad2");
```

Disponemos de las siguientes tablas:



**TCLIENTES:** Datos de clientes, IDCLIENTE Clave primaria.

**TPRODUCTOS:** Datos de productos, IDPRODUCTO Clave primaria.

**TVENTAS:** Datos de las ventas de los clientes, IDVENTA Clave primaria. Un cliente puede tener muchas ventas. Cada venta tiene 1 o más líneas de venta.

**TLINEASVENTAS:** Detalle de las ventas, Clave Primaria: IDVENTA + NUMEROLINEA. Cada venta está formada por un número de líneas de ventas. Si una venta tiene 3 líneas, en esta tabla habrá 3 filas para esa venta, y el NUMERODELINEA tendrá los valores 1, 2 y 3. Si tiene 4 líneas 1, 2, 3 y 4, etc.

Realiza un **programa Java** (y luego el JAR para ejecutarlo desde la línea de comandos) que permita introducir por teclado los datos de una venta (y sus líneas de venta) e insertarlos en la base de datos. La fecha de la venta es la fecha del sistema. El programa debe actualizar el **STOCKACTUAL** del producto a partir de la cantidad introducida por teclado, se restará esta cantidad al stock. Se pedirán los siguientes datos por teclado:

En primer lugar:

**IDVENTA:** se debe comprobar que no exista.

**IDCLIENTE:** se debe comprobar que exista.

(si alguna de las condiciones anteriores no se cumple **finaliza el programa**)

Superados esos controles se piden los siguientes datos:

**Número de líneas de la venta** (debe ser un número entre 1 y 5, si no es correcto **finaliza el programa**)

Introducir a continuación tantas veces como números de línea se ha escrito:

**IDPRODUCTO:** se debe comprobar que exista, si no existe se deben pedir los datos de nuevo.

**CANTIDAD:** debe ser numérico > que 0. Si no se cumple se debe pedir de nuevo la cantidad.

No se insertarán líneas de venta si al calcular el stock actual es menor que el stock mínimo

No se insertará una venta si no tiene líneas de venta.

## EJERCICIOS RA2

**Se incluirá todo lo implicado en la venta en una transacción.**

(Convertir a mayúscula el IDVENTA, IDCLIENTE e IDPRODUCTO una vez leídos por teclado)

### CREA EL FICHERO JAR PARA EJECUTAR EL EJERCICIO DESDE LA LINEA DE COMANDOS

Ejemplos de ejecución:

**D:\ADAT\uni2>JAVA -jar Ejercicio.jar**

Introduce IDVENTA: 3  
IDVENTA EXISTENTE...  
FIN DE PROCESO....

**D:\ADAT\uni2>JAVA -jar Ejercicio.jar**

Introduce IDVENTA: 4  
Introduce IDCLIENTE: C002  
IDCLIENTE NO EXISTE...  
FIN DE PROCESO....

**D:\ADAT\uni2>JAVA -jar Ejercicio.jar**

Introduce IDVENTA: 5  
Introduce IDCLIENTE: c001  
Introduce Número de líneas: 0  
NÚMERO DE LÍNEAS INCORRECTO (valor de 1 a 5)...  
FIN DE PROCESO....

**D:\ADAT\uni2>JAVA -jar Ejercicio.jar**

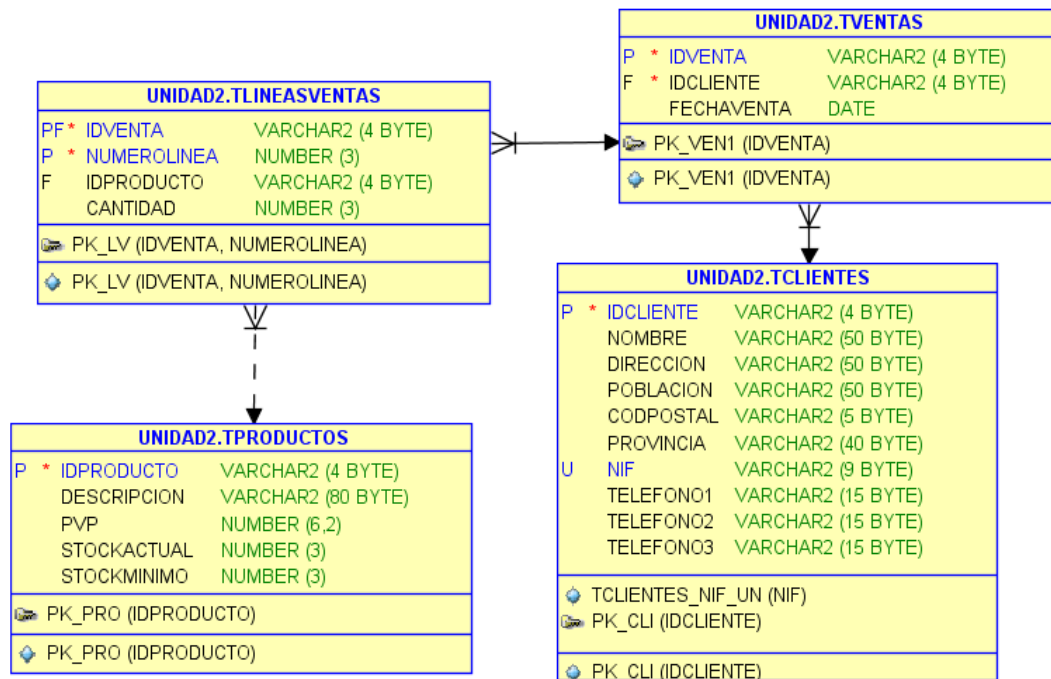
Introduce IDVENTA: 4  
Introduce IDCLIENTE: C001  
Introduce Número de líneas: 2  
Linea 1  
    Introduce IDPRODUCTO: FF  
    IDPRODUCTO NO EXISTE...  
Linea 1  
    Introduce IDPRODUCTO: P001  
    Introduce CANTIDAD: 2  
    Linea de Venta Insertada: 1  
    Producto: P001 actualizado...  
Linea 2  
    Introduce IDPRODUCTO: AB02  
    Introduce CANTIDAD: 300  
    No hay stock para el producto AB02  
    Introduce de nuevo los datos de la línea 2  
Linea 2  
    Introduce IDPRODUCTO: AB02  
    Introduce CANTIDAD: 2  
    Linea de Venta Insertada: 2  
    Producto: AB02 actualizado...  
FIN DE PROCESO....

**D:\ADAT\uni2>JAVA -jar Ejercicio.jar**

Introduce IDVENTA: 5  
Introduce IDCLIENTE: c001  
Introduce Número de líneas: 2  
Linea 1  
    Introduce IDPRODUCTO: p001  
    Introduce CANTIDAD: 0  
CANTIDAD INCORRECTA...  
    Introduce CANTIDAD: 1  
    Linea de Venta Insertada: 1  
    Producto: p001 actualizado...  
    . . . . .

## EJERCICIOS RA2

Partimos de la base de datos **ORACLE** con nombre de usuario **UNIDAD2**, y clave **UNIDAD2**. Disponemos de las siguientes tablas:



**TCLIENTES:** Datos de clientes, IDCLIENTE Clave primaria.

**TPRODUCTOS:** Datos de productos, IDPRODUCTO Clave primaria.

**TVENTAS:** Datos de las ventas de los clientes, IDVENTA Clave primaria. Un cliente puede tener muchas ventas. Cada venta tiene 1 o más líneas de venta.

**TLINEASVENTAS:** Detalle de las ventas, Clave Primaria: IDVENTA + NUMEROLINEA. Cada venta está formada por un número de líneas de ventas. Si una venta tiene 3 líneas, en esta tabla habrá 3 filas para esa venta, y el NUMERODELINEA tendrá los valores 1, 2 y 3. Si tiene 4 líneas 1, 2, 3 y 4, etc.

Crea desde ORACLE una función que reciba un **IDCLIENTE** y devuelva el importe total de las ventas que ha realizado y el número de ventas realizadas

(dado que las funciones devuelven un valor define parámetros de salida). Si el cliente no existe debe devolver -1 en cada uno de los campos. El nombre de la función es **FUNCION1**.

Crea desde ORACLE otra función que reciba los datos de un producto y los inserte en la tabla TPRODUCTOS. La función debe devolver una cadena de caracteres que indica si se ha insertado o no el producto, y si no se ha insertado debe indicar el motivo. El nombre de la función es **FUNCION2**.

Realiza un **programa Java** de nombre que permita introducir por teclado el IDCLIENTE que se desea consultar y desde la línea de argumentos de **main()** los datos de un producto (IDPRODUCTO, DESCRIPCIÓN, PVP, STOCKACTUAL y STOCKMINIMO) que se desea insertar en la tabla TPRODUCTOS. Controlar que se introduzcan todos los argumentos de **main()** y que el PVP, el STOCKACTUAL y el STOCKMINIMO sean numéricos y > 0. Además el STOCKMINIMO debe ser < que el STOCKACTUAL y > 0. Si alguna de esas condiciones falla visualizar mensaje y **finalizar programa**.

Usa la **FUNCION1** sobre el IDCLIENTE introducido por teclado para mostrar el importe total de las ventas que ha realizado y el número de ventas realizadas.

## EJERCICIOS RA2

Usa la **FUNCION2** para insertar el producto (cuyos datos has introducido desde los argumentos de *main()*) en la tabla TPRODUCTOS. Debes mostrar el mensaje que devuelve la función.

**LAS FUNCIONES LAS TIENES QUE CREAR DESDE EL PROPIO PROGRAMA JAVA.  
CREA EL FICHERO JAR PARA EJECUTAR EL EJERCICIO DESDE LA LINEA DE COMANDOS**

### EJEMPLOS DE EJECUCIÓN:

```
D:\ADAT\uni2>java -jar Ejercicio.jar  
ARGUMENTOS INCORRECTOS ....
```

```
D:\ADAT\uni2>java -jar Ejercicio.jar P999 "LAPIZ DE MADERA" 1.5 90 5  
Introduce IDCLIENTE: C001  
Cliente: C001  
Número de ventas: 3, Total ventas: 2.547,50
```

Resultado de la inserción: PRODUCTO INSERTADO

```
D:\ADAT\uni2>java -jar Ejercicio.jar P999 "LAPIZ DE MADERA" 1.5 90 5  
Introduce IDCLIENTE: DDD  
Cliente: DDD  
Número de ventas: -1, Total ventas: -1,00
```

Resultado de la inserción: CLAVE DUPLICADA

```
D:\ADAT\uni2>java -jar Ejercicio.jar P919 "GAFAS DE SOL" 1.5 AA 1  
Introduce IDCLIENTE: W  
Cliente: W  
Número de ventas: -1, Total ventas: -1,00
```

STOCK INCORRECTO

```
D:\ADAT\uni2>Java -jar Ejercicio.jar PDp9 "JARRA DE AGUA" 0 3 1  
Introduce IDCLIENTE: C009  
Cliente: C009  
Número de ventas: 2, Total ventas: 1.645,00
```

EL PVP DEBE SER > 0

```
D:\ADAT\uni2>java -jar Ejercicio.jar 9900 "JARRA DE AGUA" 333330 3 1  
Introduce IDCLIENTE: X001  
Cliente: X001  
Número de ventas: -1, Total ventas: -1,00
```

Resultado de la inserción: ERROR INESPERADO -1438\*ORA-01438: valor mayor que el que permite la precisión especificada para esta columna

```
D:\ADAT\uni2> java -jar Ejercicio.jar P998 "LAPIZ DE METAL" 1.5 90 100  
Introduce IDCLIENTE: c001  
Cliente: C001  
Número de ventas: 3, Total ventas: 2.547,50
```

EL STOCK MIN NO DEBE SUPERAR AL ACTUAL