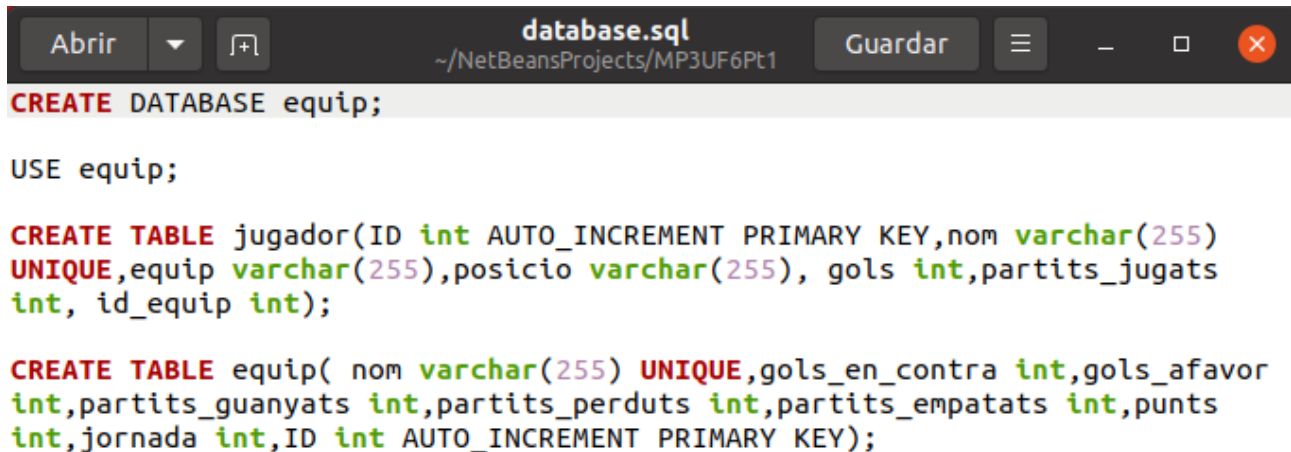


INFORMACIO ABANS DE PROVAR EL PROGRAMA

MOLT IMPORTANT: PER A QUE FUNCIONI BE TENS DE DESCOMPRIMIR L'ARXIU Programacio2nDAM-master (4).zip QUE ES ON ESTA LA LLIBRERIA I TENS D'USAR AQUESTA LLIBRERIA JA QUE HE FET UNA PETITA MODIFICACIO I SI NO USES AQUESTA LLIBRERIA NO ET FUNCIONARA BE

TAMBE TÉ FICAT L'ARXIU database.sql PER A LA CREACIO DE LA BASE DE DADES.



```
Abrir  ▾  [icon]  database.sql  Guardar  ≡  -  □  ×
~/NetBeansProjects/MP3UF6Pt1

CREATE DATABASE equip;

USE equip;

CREATE TABLE jugador(ID int AUTO_INCREMENT PRIMARY KEY,nom varchar(255)
UNIQUE,equip varchar(255),posicio varchar(255), gols int,partits_jugats
int, id_equip int);

CREATE TABLE equip( nom varchar(255) UNIQUE,gols_en_contra int,gols_afavor
int,partits_guanyats int,partits_perduts int,partits_empatats int,punts
int,jornada int,ID int AUTO_INCREMENT PRIMARY KEY);
```

No dual: fitxer Properties per connectar-se a la BD, (1 punt)(0 punts)

Aquí veus el arxiu config.properties.

```
Abrir  ▾  [icon]  config.properties  ~/NetBeansProjects/MP3UF6Pt1  C
driver=com.mysql.cj.jdbc.Driver
url=jdbc:mysql://localhost:3306/equip
usuari=alumne
password=alumne
```

El primer metode que s'executa al Controller es el metode llegirEquip per tant he agafat i al metode llegirEquip es on s'obre la connexio a la base de dades i aquesta connexio no la tanco.

```
public void llegirEquip() throws IOException {
    try {
        properties.load(new FileInputStream(new File("config.properties")));
        System.out.println(properties.get("driver"));
        String URL = (String) properties.get("url");
        String USER = (String) properties.get("usuari");
        String PASS = (String) properties.get("password");
        conn = DriverManager.getConnection(URL, USER, PASS);
    }
}
```

Tinc aquest metode tancarConn que tanca la connexio amb la base de dades.

```
public static void tancarConn() throws SQLException {
    conn.close();
}
}
```

I quan tanquen la finestra es crida al metode tancarConn i es tanca la connexio amb la base de dades.

```
view.addWindowListener(new java.awt.event.WindowAdapter() {
    @Override
    public void windowClosing(java.awt.event.WindowEvent e) {
        try {
            model.tancarConn();
        } catch (SQLException ex) {
            Logger.getLogger(Controller.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
});
```

No dual: ús de PreparedStatements,(1 punt)(0 punts)

Aquí veus com he usat PreparedStatements.

```
public static void borrarJugador(Jugador j1) {
    dadesJugador.remove(j1);
    dadesJugador2.remove(j1);
    try {
        statement = conn.createStatement();
        ResultSet resultSet = null;
        preparedStatement = conn
            .prepareStatement("DELETE FROM jugador WHERE nom=? ;");
        preparedStatement.setString(1, j1.get1_nomcognoms());

        preparedStatement.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(Model.class.getName()).log(Level.SEVERE, null, ex);
    }
    if (j1.get2_equip() != null) {
        j1.get2_equip().get9_jug().remove(j1);
    }
}
```

```
try {
    statement = conn.createStatement();
    ResultSet resultSet = null;
    preparedStatement = conn
        .prepareStatement("INSERT INTO jugador(nom,equip,posicio,goals,partits_jugats,id_equip) VALUES (?, ?, ?, ?, ?, ?) ;");
    preparedStatement.setString(1, jug1.get1_nomcognoms());
    preparedStatement.setString(2, jug1.get2_equip().toString());
    preparedStatement.setString(3, jug1.get3_posicio());
    preparedStatement.setInt(4, jug1.get4_goals());
    preparedStatement.setInt(5, jug1.get5_partits());
    preparedStatement.setInt(6, clau2);
    preparedStatement.executeUpdate();
    // conn.close();
} catch (Exception ex) {
    //Logger.getLogger(Model.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

No dual: ús de transaccions (quan insertem/borrem/modifiquem un registre a una taula heu de fer que sigue necessari usar transaccions. Això ho podeu aconseguir posant camps calculats, a la taula en cardinalitat 1, a partir dels registres de la de cardinalitat n. Per exemple si un país té un camp població i les ciutats del país també tenen el camp població, si modifiquem la de la ciutat també hauríem de modificar la del país, tot dins d'una transacció per no deixar la BD inconsistent), (1 punt)(0 punts)

Al mètode updateEquip faig un update de la taula equip i un update de la taula jugador així que aquí uso transaccions.

```
public static void updateEquip(String nomActualitzat, int gols_en_contra, int gols_afavor, int partits_guanyats, int partits_perduts,
try {
    conn.setAutoCommit(false);
    statement = conn.createStatement();
    preparedStatement = conn
        .prepareStatement("UPDATE equip SET nom = ?, gols_en_contra = ?,gols_afavor=?,partits_guanyats=?,partits_perduts=?,par
    preparedStatement.setString(1, nomActualitzat);
    preparedStatement.setInt(2, gols_en_contra);
    preparedStatement.setInt(3, gols_afavor);
    preparedStatement.setInt(4, partits_guanyats);
    preparedStatement.setInt(5, partits_perduts);
    preparedStatement.setInt(6, partits_empatats);
    preparedStatement.setInt(7, punts);
    preparedStatement.setInt(8, jornada);
    preparedStatement.setString(9, nom);
    preparedStatement.executeUpdate();
    int clau2 = 0;
    preparedStatement = conn
        .prepareStatement("select * from equip WHERE nom=?");
    preparedStatement.setString(1, nomActualitzat);
    ResultSet rs = preparedStatement.executeQuery();
    Equip obj1;
    while (rs.next()) {
        clau2 = rs.getInt("id");
    }
    ResultSet resultSet = null;
    preparedStatement = conn
        .prepareStatement("UPDATE jugador SET equip = ? WHERE id_equip=?");
    preparedStatement.setString(1, nomActualitzat);
    preparedStatement.setInt(2, clau2);
    preparedStatement.executeUpdate();
    conn.commit();
} catch (SQLException e) {
    System.out.println(e.toString());
    if (conn != null) {
        try {
            conn.rollback();
        } catch (SQLException ex) {
            System.out.println(ex.toString());
        }
    }
}
}
```

Tots: CRUD de totes les taules. (4 punts) (7 punts)

Bueno aquí no et fico captures, però em funciona be el CRUD no he vist que em falli res del CRUD. Els mètodes uso els mateixos que abans solament que ara al tenir base de dades pues uso update,select,insert.

Extres:

Tots: selecció automàtica dels registres relacionats d'ambdues taules (per exemple, si seleccionem una ciutats s'ha de seleccionar el seu país, i si seleccionem un país les seues ciutats), (2 punts) (3 punts)

Ho he fet però es té d'activar el checkbox per a que funcioni.

☐ Filtra Equip

☐ Filtra Jugadors

Aquí veus com per exemple al seleccionar el equip «algo» i tenint el checkbox activat ma seleccionat automaticament els jugadors d'aquest equip. A la foto no veus que estigui seleccionat el equip perquè automaticament me deselecciona l'equip.

LA LIGA TEMPORADA 2020/2021

EQUIPS

ID	NOM	GOLSENCON...	GOLSAFAVOR	PARTITSGUA...	PARTITSPER...	PARTITSEMP...	PUNTS	JORNADA	JUG
33	algo	4	1	1	11	11	1	1	[prova, aaa]
27	boneslaa	1111	11	111	11	11	1	1	[aaaaaa]
29	males	1	4444	1	1	1	1	0	[aa]
32	prova	1	4444	1	1	1	1	0	[]
30	zaragozaa	4	1	1	11	11	1	1	[]

DADES EQUIPS

Nom Equip

algo

4

Guanyats

Pe

1

11

Punts Equip

Jor

1

1

+

Puntuat

☒ Filtra Jugadors

JUGADORS

IDEQUIP	NOMCOGNOMS	EQUIP	POSICIO	GOLS	PARTITS
29	aa	males	delanter	1	1
27	aaaaaa	boneslaa	defensa	1	1
33	prova	algo	delanter	1	1
33	aaa	algo	defensa	111	1

DADES JUGADORS

Nom cognoms

Pos

Gols

Pa

☐ Filtra Equip

OPC

Codi

```
11 (view.getCheckBox2().isSelected() == true && filaSel2 != -1) {
    TableColumnModel tcm2 = view.getTaulaJugadors().getColumnModel();
    tcm2.addColumn(tc2);
    Jugador obj2 = (Jugador) view.getTaulaJugadors().getValueAt(filaSel2, tcm2.getColumnCount() - 1);
    tcm2.removeColumn(tc2);
    Equip eq1 = obj2.get2_equip();
    Collection<Equip> prova = new TreeSet<>();
    prova.add(eq1);
    ListSelectionModel seleccioEquip = view.getTaulaEquips().getSelectionModel();
    seleccioEquip.clearSelection();
    for (int i = 0; i < view.getTaulaEquips().getRowCount(); i++) {
        if (view.getTaulaEquips().getValueAt(i, 0).equals(obj2.get0_idequip())) {
            seleccioEquip.addSelectionInterval(i, i);
        }
    }
    // tc = Utils.<Equip>loadTable(prova, view.getTaulaEquips(), Equip.class, true, true);
}
```

```

view.getNomEquip().setText(obj.toString());
if (view.getjCheckBox1().isSelected() == true) {
    view.getNomEquip().setText(obj.toString());
    carregarTaulaEquip();
    // tc3 = Utils.<Jugador>loadTable(obj.get9_jug(), view.getTaulaJugadors(), Jugador.class, true, true);
    ListSelectionModel seleccioJugadors = view.getTaulaJugadors().getSelectionModel();
    seleccioJugadors.clearSelection();
    for (int i = 0; i < view.getTaulaJugadors().getRowCount(); i++) {
        if (view.getTaulaJugadors().getValueAt(i, 0).equals(obj.get10_id())) {
            seleccioJugadors.addSelectionInterval(i, i);
        }
    }
} else {
    carregarTaulaEquip();
    carregarTaulaJugador();
}

```

No dual: modificació de dades directament a les pròpies taules que mostren la informació (o sigui, que si canviem una dada de per exemple un país mostrat a un JTable, que automàticament s'actualitzi la info a la BD). (1 punt) (0 punts)

Això també em funciona be, el codi el tinc en aquests 2 listeners:

```

view.getTaulaJugadors().addKeyListener(new KeyListener() {
    view.getTaulaEquips().addKeyListener(new KeyListener() {

```

El camp ID de la taula equips no es pot modificar si cliques no t'el deixara editar, la resta de camps si que els pots editar però he fet que es comprovin si les dades que introdueix lo usuari son correctes així que no deixara ficar dades erronies.

A la taula JUGADORS també em funciona be es pot editar, per editar l'equip es te de modificar el camp IDEQUIP si per exemple on fica 29 fiquem 30 automaticament ens ficara al jugador a l'equip zaragozaaa.

LA LIGA TEMPORADA 2020/2021									
EQUIPS									
ID	NOM	GOLSENCON...	GOLSAFAVOR	PARTITSGUA...	PARTITSPER...	PARTITSEMP...	PUNTS	JORNADA	JUG
27	boneslaa	1111	11	111	11	11	1	0	[]
29	males	1	4444	1	1	1	1	0	[aa]
30	zaragozaaa	4	1	1	11	11	1	0	[algo]
28	a	4	4	111	111	11	11	0	[]

JUGADORS					
IDEQUIP	NOMCOGNOMS	EQUIP	POSICIO	GOLS	PARTITS
29	aa	males	delanter	1	1
30	algo	zaragozaaa	defensa	111	1