

The diagram is an Entity-Relationship (ER) model for a music and movie database. It features several entities and their relationships:

- Artist** (Entity): Attributes include ArtistID (Primary Key), Name, and Make.
- Album** (Entity): Attributes include AlbumID (Primary Key), Name, Price, Year, and Genre.
- Song** (Entity): Attributes include SongID (Primary Key) and Title.
- User** (Entity): Attributes include Username (Primary Key), Date of Birth, Date of Birth Privacy Setting, Email Privacy Setting, Email, and a relationship to **Friend Request**.
- Buy Album** (Entity): Attributes include Privacy Setting.
- Credits Account** (Entity): Attributes include Privacy Setting.
- Buy Movie** (Entity): Attributes include Privacy Setting.
- Movie** (Entity): Attributes include MovieID (Primary Key), Price, Genre, Director, Rental Fee, Title, Year, Value, and Rate.
- Friend Request** (Entity): Attributes include From, To, and Status.

Relationships:

- Make** (Relationship): Connects **Artist** and **Album**.
- Fan Of** (Relationship): Connects **Artist** and **Album**.
- On** (Relationship): Connects **Album** and **Song**.
- Own** (Relationship): Connects **User** and **Buy Album**.
- Rent** (Relationship): Connects **User** and **Buy Movie**.
- License** (Relationship): Connects **Credits Account** and **Buy Movie**.
- For** (Relationship): Connects **License** and **Movie**.
- Like/Dislike** (Relationship): Connects **User** and **Song**.
- Wait For** (Relationship): Connects **User** and **Movie**.
- Friend Request** (Relationship): Connects **User** and **User**.

Other Relationships:

- Artist** is connected to **Album** via **Make** and **Fan Of**.
- Album** is connected to **Song** via **On**.
- User** is connected to **Buy Album** via **Own**, to **Buy Movie** via **Rent**, to **Movie** via **Wait For**, and to **Friend Request**.
- Credits Account** is connected to **Buy Movie** via **License**.
- License** is connected to **Movie** via **For**.
- Movie** is connected to **Rate** via **Value**.

```
/* The Is-A group of Employee based Entities */
```

```
CREATE TABLE Technical (
    Eid          INTEGER,
    Speciality   CHAR[50],
    PRIMARY KEY (Eid),
    FOREIGN KEY (Eid) REFERENCES Employee )
```

```
/* The other Entities */
```

```
CREATE TABLE Project (  
    Id    INTEGER,  
    Name  CHAR[50],  
    Budget FLOAT,  
    PRIMARY KEY (Id),  
    FOREIGN KEY (Id) REFERENCES WorksOn )
```

```
CREATE TABLE Supplier (  
    SId    INTEGER,  
    Name   CHAR[50],  
    PRIMARY KEY (SId) )
```

```
CREATE TABLE Part (  
    PId    INTEGER,  
    Name   CHAR[50],  
    PRIMARY KEY (PId),  
    FOREIGN KEY (PId) REFERENCES Has )
```

```
CREATE TABLE PurchaseOrder (  
    POId   INTEGER,  
    PRIMARY KEY (POId),  
    FOREIGN KEY (POId) REFERENCES Approves )
```

```
CREATE TABLE Depts (  
    DId    INTEGER,  
    Name   CHAR[50],  
    Budget FLOAT,  
    PRIMARY KEY (DId),  
    FOREIGN KEY (DId) REFERENCES Manages )
```

/* The Relationships */

```
CREATE TABLE WorksOn (  
    Eid    INTEGER,  
    Id     INTEGER,  
    PRIMARY KEY (Eid, Id),  
    FOREIGN KEY (Eid) REFERENCES Technical,  
    FOREIGN KEY (Id) REFERENCES Project )
```

```
CREATE TABLE Sponsors (  
    SId    INTEGER,  
    Id     INTEGER,  
    Funding    FLOAT,  
    PRIMARY KEY (SId, Id),  
    FOREIGN KEY (SId) REFERENCES Supplier,  
    FOREIGN KEY (Id) REFERENCES Project )
```

```
CREATE TABLE SBy (  
    PId    INTEGER,  
    SId    INTEGER,  
    Price  INTEGER,
```

PRIMARY KEY (Pid, SId),
FOREIGN KEY (Pid) REFERENCES Part,
FOREIGN KEY (SId) REFERENCES Supplier)

CREATE TABLE Has (
POId INTEGER,
Pid INTEGER,
PRIMARY KEY (Pid),
FOREIGN KEY (POId) REFERENCES PurchaseOrder,
FOREIGN KEY (Pid) REFERENCES Part)

CREATE TABLE Places (
Eid INTEGER,
POId INTEGER,
PRIMARY KEY (POId),
FOREIGN KEY (Eid) REFERENCES Technical,
FOREIGN KEY (POId) REFERENCES PurchaseOrder)

CREATE TABLE Approves (
Eid INTEGER,
POId INTEGER,
PRIMARY KEY (POId),
FOREIGN KEY (Eid) REFERENCES Admin,
FOREIGN KEY (POId) REFERENCES PurchaseOrder)

CREATE TABLE IsAt (
Eid INTEGER,
DId INTEGER,
PRIMARY KEY (Eid),
FOREIGN KEY (Eid) REFERENCES Admin,
FOREIGN KEY (DId) REFERENCES Depts)

CREATE TABLE Manages (
Eid INTEGER,
DId INTEGER,
PRIMARY KEY (DId),
FOREIGN KEY (Eid) REFERENCES Admin,
FOREIGN KEY (DId) REFERENCES Depts)