

```
DROP TABLE IF EXISTS membre CASCADE;
DROP TABLE IF EXISTS jeu CASCADE;
DROP TABLE IF EXISTS categorie CASCADE;
DROP TABLE IF EXISTS splits CASCADE;
DROP TABLE IF EXISTS run CASCADE;
DROP TABLE IF EXISTS jeuSpeedrun CASCADE;
```

```
CREATE TABLE membre(
    idMembre SERIAL PRIMARY KEY,
    pseudoMembre VARCHAR(40) UNIQUE NOT NULL,
    passwordMembre VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE jeu(
    idJeu SERIAL PRIMARY KEY,
    nomJeu varchar(100) UNIQUE NOT NULL,
    typeJeu varchar(30),
    descriptionJeu varchar(500)
);
```

```
CREATE TABLE categorie(
    idCategorie SERIAL UNIQUE,
    idJeu INTEGER,
    nomCategorie varchar(30),
    descriptionCategorie varchar(500),
    PRIMARY KEY (idCategorie, idJeu),
    UNIQUE (idJeu,nomcategorie),
    FOREIGN KEY (idJeu) REFERENCES jeu (idJeu)
);
```

```
CREATE TABLE splits(  
    idSplits SERIAL UNIQUE,  
    idCategorie SERIAL,  
    dataSplits varchar(300),  
    PRIMARY KEY(idSplits,idCategorie),  
    FOREIGN KEY (idCategorie) REFERENCES categorie (idCategorie)  
);
```

```
CREATE TABLE run(  
    idRun SERIAL,  
    idSplits SERIAL,  
    idCategorie INTEGER NOT NULL,  
    idJeu INTEGER NOT NULL,  
    idMembre INTEGER NOT NULL,  
    dateRun varchar(50) NOT NULL,  
    finalTime INTEGER NOT NULL,  
    dataFinalTimes varchar(200),  
    PRIMARY KEY(idRun,idCategorie,idMembre,dateRun),  
    FOREIGN KEY (idCategorie) REFERENCES categorie (idCategorie),  
    FOREIGN KEY (idJeu) REFERENCES Jeu(idJeu),  
    FOREIGN KEY (idSplits) REFERENCES splits(idSplits),  
    FOREIGN KEY (idMembre) REFERENCES membre(idMembre)  
);
```

```
CREATE TABLE jeuSpeedrun(  
    idMembre SERIAL,  
    idCategorie SERIAL,  
    PRIMARY KEY(idMembre,idCategorie),  
    FOREIGN KEY (idMembre) REFERENCES membre(idMembre),
```

```
FOREIGN KEY (idCategorie) REFERENCES Categorie(idCategorie)
);
```

```
INSERT INTO jeu (nomjeu,typejeu,descriptionjeu) VALUES('Super Meat
Boy','Platform','Indy Game by E. McMillen');
```

```
INSERT INTO jeu (nomjeu,typejeu,descriptionjeu)
VALUES('Skyrim','RPG','The Elders Scrolls V');
```

```
INSERT INTO jeu (nomjeu,typejeu,descriptionjeu)
VALUES('Portal','Puzzle','Fisrt Person Puzzle Game by Valve');
```

```
CREATE FUNCTION split_newCategorie() RETURNS trigger AS
$split_newCategorie$
```

```
BEGIN
```

```
INSERT INTO splits (idcategorie,datasplits) VALUES
(NEW.idcategorie,'["End"]');
```

```
RETURN NULL;
```

```
END;
```

```
$split_newCategorie$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER split_newCategorie AFTER INSERT ON categorie
```

```
FOR EACH ROW EXECUTE PROCEDURE split_newCategorie();
```

```
INSERT INTO categorie(idjeu,nomcategorie,descriptioncategorie) VALUES
(1,'106%','Easy category');
```