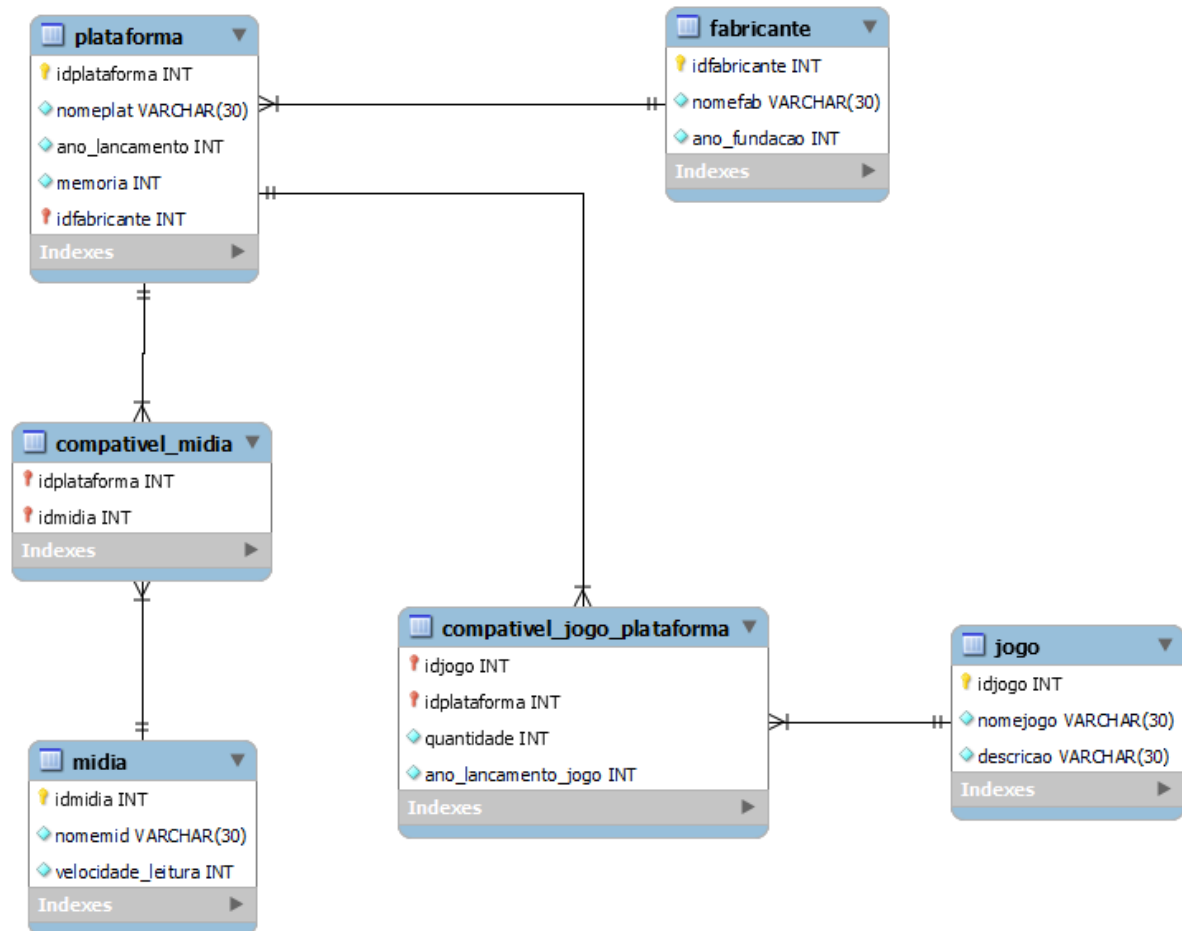


Trabalho Prático 02 - Banco de Dados

Nome: Miguel Antônio Ribeiro e Silva

Matrícula: 4680

Diagrama de Classes



Forward Engineer

já com atualização das questões 11 a 13.

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_
DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBS
TITUTION';
```

```
-- -----
-- Schema tp02
-- -----
```

```
-- -----
-- Schema tp02
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `tp02` DEFAULT CHARACTER SET utf8 ;
USE `tp02` ;
```

```
-- -----
-- Table `tp02`.`fabricante`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `tp02`.`fabricante` (
  `idfabricante` INT NOT NULL AUTO_INCREMENT,
  `nomefab` VARCHAR(30) NOT NULL,
  `ano_fundacao` INT NOT NULL,
  PRIMARY KEY (`idfabricante`))
ENGINE = InnoDB;
```

```
-- -----  
-- Table `tp02`.`plataforma`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `tp02`.`plataforma` (  
  `idplataforma` INT NOT NULL AUTO_INCREMENT,  
  `nomeplat` VARCHAR(30) NOT NULL,  
  `ano_lancamento` INT NOT NULL,  
  `memoria` INT NOT NULL,  
  `idfabricante` INT NOT NULL,  
  PRIMARY KEY (`idplataforma`, `idfabricante`),  
  INDEX `fk_plataforma_fabricante1_idx` (`idfabricante` ASC) VISIBLE,  
  CONSTRAINT `fk_plataforma_fabricante1`  
    FOREIGN KEY (`idfabricante`)  
    REFERENCES `tp02`.`fabricante` (`idfabricante`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `tp02`.`midia`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `tp02`.`midia` (  
  `idmidia` INT NOT NULL AUTO_INCREMENT,  
  `nomemid` VARCHAR(30) NOT NULL,  
  `velocidade_leitura` INT NOT NULL,  
  PRIMARY KEY (`idmidia`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `tp02`.`compativel_midia`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `tp02`.`compativel_midia` (  
  `idplataforma` INT NOT NULL,  
  `idmidia` INT NOT NULL,  
  PRIMARY KEY (`idplataforma`, `idmidia`),  
  INDEX `fk_plataforma_has_midia_midia1_idx` (`idmidia` ASC) VISIBLE,  
  INDEX `fk_plataforma_has_midia_plataforma_idx` (`idplataforma` ASC) VISIBLE,  
  CONSTRAINT `fk_plataforma_has_midia_plataforma`  
    FOREIGN KEY (`idplataforma`)  
      REFERENCES `tp02`.`plataforma` (`idplataforma`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_plataforma_has_midia_midia1`  
    FOREIGN KEY (`idmidia`)  
      REFERENCES `tp02`.`midia` (`idmidia`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `tp02`.`desenvolvedora`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `tp02`.`desenvolvedora` (  
  `iddesenvolvedora` INT NOT NULL,  
  `nomedev` VARCHAR(30) NOT NULL,  
  `ano_craicao` VARCHAR(30) NOT NULL,  
  PRIMARY KEY (`iddesenvolvedora`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `tp02`.`jogo`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `tp02`.`jogo` (  
  `idjogo` INT NOT NULL AUTO_INCREMENT,  
  `nomejogo` VARCHAR(30) NOT NULL,  
  `descricao` VARCHAR(30) NOT NULL,  
  `iddesenvolvedora` INT NOT NULL,  
  PRIMARY KEY (`idjogo`, `iddesenvolvedora`),  
  INDEX `fk_jogo_desenvolvedora1_idx` (`iddesenvolvedora` ASC) VISIBLE,  
  CONSTRAINT `fk_jogo_desenvolvedora1`  
    FOREIGN KEY (`iddesenvolvedora`)  
    REFERENCES `tp02`.`desenvolvedora` (`iddesenvolvedora`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `tp02`.`compativel_jogo_plataforma`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `tp02`.`compativel_jogo_plataforma` (  
  `idjogo` INT NOT NULL,  
  `idplataforma` INT NOT NULL,  
  `quantidade` INT NOT NULL,  
  `ano_lancamento_jogo` INT NOT NULL,  
  PRIMARY KEY (`idjogo`, `idplataforma`),  
  INDEX `fk_jogo_has_plataforma_plataforma1_idx` (`idplataforma` ASC) VISIBLE,  
  INDEX `fk_jogo_has_plataforma_jogo1_idx` (`idjogo` ASC) VISIBLE,  
  CONSTRAINT `fk_jogo_has_plataforma_jogo1`  
    FOREIGN KEY (`idjogo`)  
    REFERENCES `tp02`.`jogo` (`idjogo`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_jogo_has_plataforma_plataforma1`  
    FOREIGN KEY (`idplataforma`)  
    REFERENCES `tp02`.`plataforma` (`idplataforma`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Exercícios

- 1) Faça uma consulta SQL para recuperar os nomes das plataformas que possuam pelo menos uma mídia com velocidade de leitura superior a 100. A tabela resultante da execução desta consulta não deverá ter repetição de nomes das plataformas.

```
SELECT nomeplat  
FROM plataforma  
NATURAL JOIN compative_l_midia  
NATURAL JOIN midia  
WHERE velocidade_leitura > 100  
GROUP BY nomeplat;
```

	nomeplat
▶	Atari 2600
	NES
	SNES

- 2) Escreva uma consulta SQL que retorne os nomes das plataformas, cada uma juntamente com o seu ano de lançamento, desde que seu fabricante tenha sido fundado após o ano de 1970.

```
SELECT nomeplat, ano_lancamento  
FROM plataforma  
NATURAL JOIN fabricante  
WHERE ano_fundacao > 1970;
```

	nomeplat	ano_lancamento
▶	Atari 2600	1977
	Xbox360	2005

- 3) Faça uma consulta SQL que recupere os nomes das mídias e suas respectivas velocidades de leitura, desde que suas velocidades de leitura estejam entre 10 e 30. Os resultados devem aparecer em ordem decrescente

de velocidades de leitura. No caso de duas mídias empatarem neste critério, elas deverão ser ordenadas pelos seus nomes, em ordem alfabética.

```
SELECT nomemid, velocidade_leitura
FROM midia
WHERE velocidade_leitura BETWEEN 10 AND 30
ORDER BY velocidade_leitura DESC, nomemid;
```

	nomemid	velocidade_leitura
▶	DVD	20
	CD	10

4) Escreva o comando para inserir a plataforma "WiiU", cujo valor de chave primária será 7. O ano de lançamento desta plataforma é 2012 e ela possui 2147483648 bytes de memória. O fabricante desta Plataforma é a Nintendo. Você deverá fazer a inserção sem pesquisar qual é o valor da chave primária da Nintendo na tabela fabricante, ou seja, usando para isso uma sub-consulta.

```
INSERT INTO `tp02`.`plataforma` (`idplataforma`, `nomeplat`, `ano_lancamento`,
`memoria`, `idfabricante`)
VALUES ('7', 'WiiU', '2012', '2147483648', (
SELECT idfabricante
FROM fabricante
WHERE nomefab = 'Nintendo'
));
```

	idplataforma	nomeplat	ano_lancamento	memoria	idfabricante
▶	1	Atari 2600	1977	128	4
	2	NES	1985	2048	1
	3	SNES	1990	131072	1
	4	Wii	2006	92274688	1
	5	Xbox360	2005	536870912	3
	6	PlayStation3	2006	536870912	2
	7	WiiU	2012	2147483648	1

5) Faça uma consulta para recuperar em cada linha do resultado o nome de uma plataforma, o nome de seu fabricante, e a quantidade total de jogos desta plataforma que o colecionador possui (se tiver dois jogos iguais de acordo com o atributo quantidade, deverão ser contados os dois). A coluna que irá

apresentar a quantidade de jogos deverá ser exibida com o nome “numero_jogos”. O resultado deve ser ordenado em ordem alfabética pelo nome da plataforma.

```
SELECT nomeplat, nomefab, SUM(quantidade) AS "numero_jogos"
FROM plataforma
NATURAL JOIN compativel_jogo_plataforma
NATURAL JOIN fabricante
GROUP BY idplataforma
ORDER BY nomeplat;
```

	nomeplat	nomefab	numero_jogos
▶	Atari 2600	Atari	4
	NES	Nintendo	2
	PlayStation3	Sony	5
	SNES	Nintendo	1
	Wii	Nintendo	4
	Xbox360	Microsoft	4

6) Faça uma consulta para recuperar os nomes dos jogos que foram lançados para mais de uma plataforma.

```
SELECT nomejogo
FROM jogo
WHERE idjogo IN (SELECT idjogo
                 FROM compativel_jogo_plataforma
                 GROUP BY idjogo
                 HAVING COUNT(idjogo) > 1);
```

	nomejogo
▶	Mario Bros
	F-Zero
	Super Mario Bros
	FIFA12
	Street Fighter IV

7) Recuperar quantos jogos foram lançados em cada ano, ordenando do ano mais antigo para o mais novo. A coluna com este número de jogos deverá se chamar “jogos_lancados”. Neste caso não são unidades do jogo, então mesmo se o colecionador tiver 2 ou mais unidades de determinado jogo (no atributo “quantidade”), deverá ser considerado apenas 1 na contagem. Mas se

um jogo tiver sido lançado para duas plataformas, aí sim contará como 2 jogos lançados.

```
SELECT      ano_lancamento_jogo,      COUNT(ano_lancamento_jogo)      AS  
'jogos_lancados'  
FROM compativel_jogo_plataforma  
NATURAL JOIN plataforma  
GROUP BY ano_lancamento_jogo  
ORDER BY ano_lancamento_jogo;
```

	ano_lancamento_jogo	jogos_lancados
▶	1982	1
	1983	2
	1985	1
	1990	1
	2006	1
	2007	2
	2008	2
	2010	1
	2011	4
	2012	1

8)Recupere o nome do jogo com o ano de lançamento mais antigo, juntamente com o próprio ano e o nome da plataforma para o qual foi lançado.

```
SELECT nomejogo, ano_lancamento_jogo, nomeplat  
FROM jogo  
NATURAL JOIN compativel_jogo_plataforma  
NATURAL JOIN plataforma  
WHERE ano_lancamento_jogo = (SELECT MIN(ano_lancamento_jogo)  
FROM compativel_jogo_plataforma);
```

	nomejogo	ano_lancamento_jogo	nomeplat
▶	River Raid	1982	Atari 2600

9) Considerando que se um jogo é lançado para uma determinada plataforma ele utiliza toda a memória daquela plataforma, qual é a média de memória utilizada pelos jogos do colecionador? Obs.: novamente considerando apenas jogos diferentes, ou seja, aqueles que o colecionador possuir mais de uma unidade deverão ser considerados no cálculo apenas uma vez. No entanto, se

ele possuir este jogo para mais de uma plataforma, deverá ser considerada uma unidade para cada plataforma que ele possuir o jogo.

```
SELECT AVG(memoria)
FROM jogo
NATURAL JOIN plataforma
NATURAL JOIN compativel_jogo_plataforma;
```

	AVG(memoria)
▶	257958160.0000

10) Recupere nesta questão apenas os nomes dos fabricantes que já utilizaram mais de 1 tipo de mídia em suas plataformas, colocando-os em ordem alfabética.

```
SELECT nomefab
FROM compativel_midia
NATURAL JOIN plataforma
NATURAL JOIN fabricante
GROUP BY nomefab
HAVING COUNT(idmidia) > 1
ORDER BY nomefab;
```

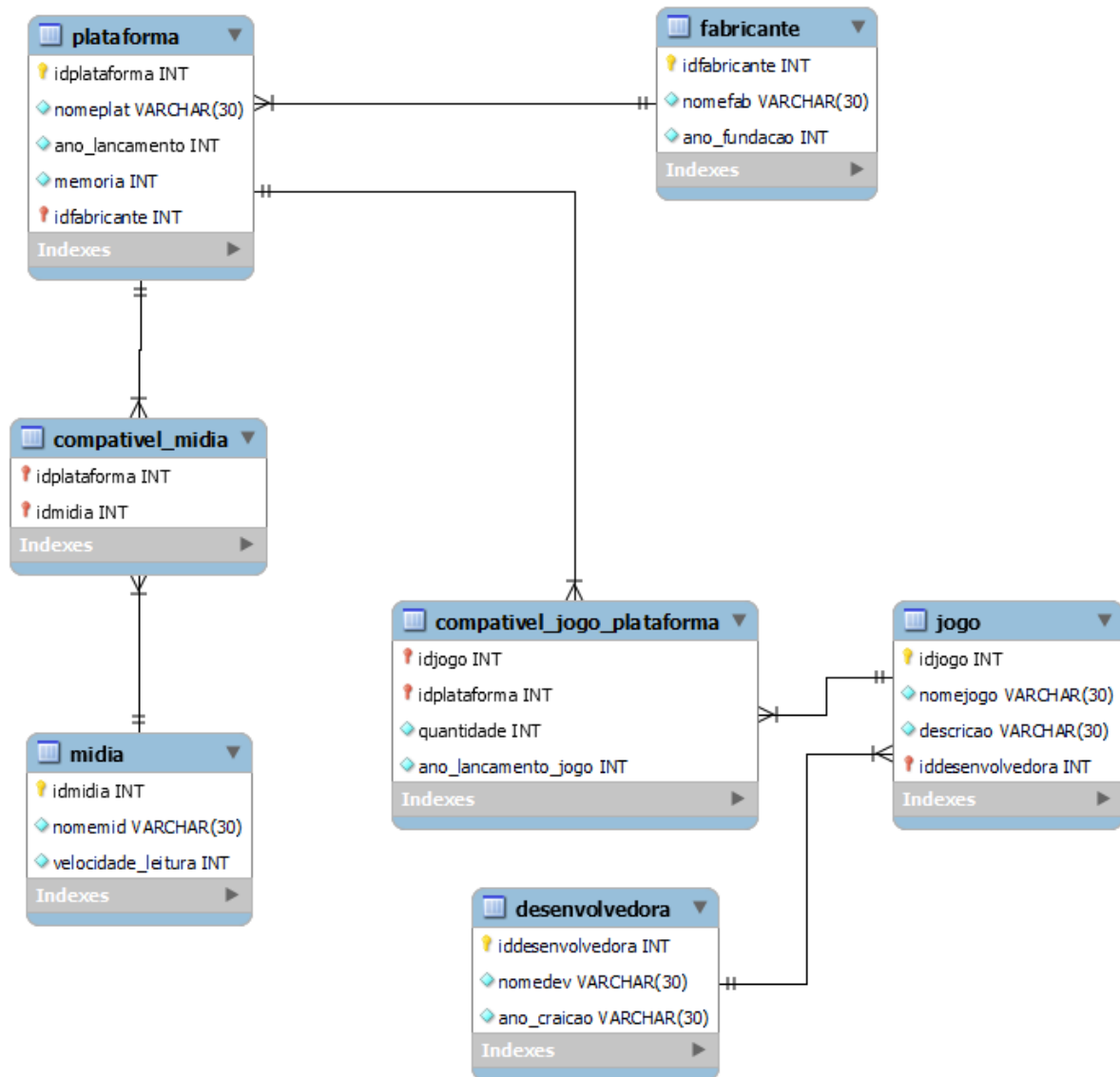
	nomefab
▶	Microsoft
	Nintendo
	Sony

11) Proponha uma expansão do esquema deste banco de dados, criando mais tabelas e/ou atributos, explicando o significado destas estruturas. Entregue os comandos criados, e também o diagrama atualizado.

Criação da tabela **DESENVOLVEDORA**, cada jogo possui 1 desenvolvedora e uma desenvolvedora possui N jogos.

A tabela, possui os atributos iddesenvolvedora (primária), nomedev, ano_criacao.

A tabela **jogo** possui a chave estrangeira iddesenvolvedora.



-- Table `tp02`.`desenvolvedora`

```

CREATE TABLE IF NOT EXISTS `tp02`.`desenvolvedora` (
  `iddesenvolvedora` INT NOT NULL,
  `nomedev` VARCHAR(30) NOT NULL,
  `ano_craicao` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`iddesenvolvedora`))
ENGINE = InnoDB;
  
```

12) Entregue comandos INSERT e/ou UPDATE que utilizem as novas estruturas criadas na questão anterior, ao inserir novos dados no banco de dados.

Foram adicionados as seguintes desenvolvedoras:

```
INSERT INTO `tp02`.`desenvolvedora` (`iddesenvolvedora`, `nomedev`,  
`ano_craicao`)  
VALUES ('1', 'Activision', '1979');
```

```
INSERT INTO `tp02`.`desenvolvedora` (`iddesenvolvedora`, `nomedev`,  
`ano_craicao`)  
VALUES ('2', 'Nintendo', '1889');
```

```
INSERT INTO `tp02`.`desenvolvedora` (`iddesenvolvedora`, `nomedev`,  
`ano_craicao`)  
VALUES ('3', 'EA', '1982');
```

```
INSERT INTO `tp02`.`desenvolvedora` (`iddesenvolvedora`, `nomedev`,  
`ano_craicao`)  
VALUES ('4', 'Santa Mônica', '1999');
```

```
INSERT INTO `tp02`.`desenvolvedora` (`iddesenvolvedora`, `nomedev`,  
`ano_craicao`)  
VALUES ('5', 'Playground Games', '2009');
```

```
INSERT INTO `tp02`.`desenvolvedora` (`iddesenvolvedora`, `nomedev`,  
`ano_craicao`)  
VALUES ('6', 'Polyphony Digital', '1998');
```

```
INSERT INTO `tp02`.`desenvolvedora` (`iddesenvolvedora`, `nomedev`,  
`ano_craicao`)  
VALUES ('7', 'Capcom', '1979');
```

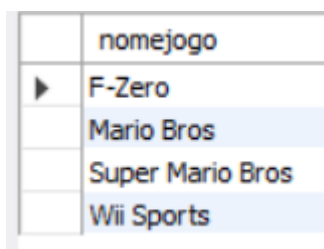
E os seguintes UPDATES aos jogos:

```
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '1' WHERE (`idjogo` = '1');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '2' WHERE (`idjogo` = '2');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '2' WHERE (`idjogo` = '3');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '2' WHERE (`idjogo` = '4');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '2' WHERE (`idjogo` = '5');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '3' WHERE (`idjogo` = '6');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '4' WHERE (`idjogo` = '7');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '5' WHERE (`idjogo` = '8');  
UPDATE `tp02`.`jogo` SET `iddesenvolvedora` = '7' WHERE (`idjogo` = '10');
```

13) Proponha pelo menos duas perguntas que levem em consideração o que foi criado nas questões 12 e 13. Dê como resposta para cada uma delas: a pergunta criada, a consulta SQL que a responde, e o screenshot com o resultado de sua execução.

Retorne em ordem alfabética, o nome dos jogos que foram desenvolvidos pela Nintendo.

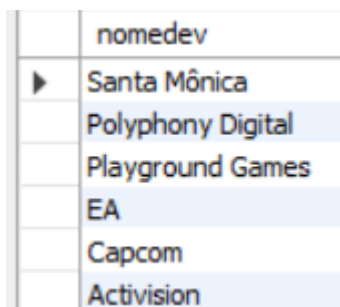
```
SELECT nomejogo
FROM jogo
NATURAL JOIN desenvolvedora
WHERE nomedev = "Nintendo"
GROUP BY nomejogo
ORDER BY nomejogo;
```



	nomejogo
▶	F-Zero
	Mario Bros
	Super Mario Bros
	Wii Sports

Retorne o nome das desenvolvedoras, em ordem decrescente, que possuem somente um jogo cadastrado no banco de dados.

```
SELECT nomedev
FROM jogo
NATURAL JOIN desenvolvedora
GROUP BY nomedev
HAVING COUNT(nomedev) = 1
ORDER BY nomedev DESC;
```



	nomedev
▶	Santa Mônica
	Polyphony Digital
	Playground Games
	EA
	Capcom
	Activision