**۱. رابطه بین Velociraptor و Interaction**

* یک **Velociraptor** می‌تواند چندین بار به در ضربه بزند و چند تعامل (Interaction) داشته باشد.
* اما هر **Interaction** فقط به یک Velociraptor مربوط می‌شود.  
  ✅ **رابطه: 1-M**  
  🔹 **(یک Velociraptor می‌تواند چند Interaction داشته باشد، ولی هر Interaction فقط به یک Velociraptor مربوط است.)**  
  **👨‍🎨 نحوه رسم:**
* یک خط از **Velociraptor** به **Interaction** بکش.
* در سمت Velociraptor علامت **1** بگذار و در سمت Interaction علامت **M**.

**۲. رابطه بین Children و Interaction**

* یک **کودک (Child)** می‌تواند در چندین تعامل (Interaction) شرکت کند (چندین بار بترسد یا واکنش نشان دهد).
* یک **Interaction** هم ممکن است شامل چند کودک باشد (همه بچه‌ها باهم درگیر ماجرا شوند).  
  ✅ **رابطه: M-M (چند به چند)**  
  🔹 **(یک کودک می‌تواند در چند Interaction حضور داشته باشد و یک Interaction هم می‌تواند شامل چند کودک باشد.)**  
  **👨‍🎨 نحوه رسم:**
* برای ارتباط **M-M** باید یک **جدول واسط (Interaction)** داشته باشیم.
* دو خط بکش: **یک خط از Children به Interaction (با M) و یک خط از Interaction به Children (با M).**

**۳. رابطه بین FridgeDoor و Interaction**

* یک **درب یخچال (FridgeDoor)** می‌تواند در چندین تعامل (Interaction) حضور داشته باشد (مثلاً ولوسی‌رپتور چند بار به در ضربه بزند).
* اما هر Interaction فقط مربوط به یک در مشخص است.  
  ✅ **رابطه: 1-M**  
  🔹 **(یک درب یخچال می‌تواند در چند Interaction شرکت کند، ولی هر Interaction فقط به یک در خاص مربوط است.)**  
  **👨‍🎨 نحوه رسم:**
* یک خط از **FridgeDoor** به **Interaction** بکش.
* در سمت **FridgeDoor** علامت **1** بگذار و در سمت **Interaction** علامت **M**.

**۴. رابطه بین Velociraptor و Impact**

* یک **Velociraptor** می‌تواند تأثیرات (Impact) مختلفی روی کودکان و درب یخچال بگذارد (مثلاً هم در را بشکند، هم باعث ترس بچه‌ها شود).
* اما هر **Impact** فقط مربوط به یک Velociraptor خاص است.  
  ✅ **رابطه: 1-M**  
  🔹 **(یک Velociraptor می‌تواند چندین Impact داشته باشد، ولی هر Impact فقط از یک Velociraptor می‌آید.)**  
  **👨‍🎨 نحوه رسم:**
* یک خط از **Velociraptor** به **Impact** بکش.
* در سمت **Velociraptor** علامت **1** بگذار و در سمت **Impact** علامت **M**.

**۵. رابطه بین Children و Impact**

* یک **کودک (Child)** ممکن است تأثیرات مختلفی دریافت کند (مثلاً هم بترسد، هم فرار کند).
* یک **Impact** هم ممکن است چند کودک را تحت تأثیر قرار دهد.  
  ✅ **رابطه: M-M (چند به چند)**  
  🔹 **(یک کودک می‌تواند تحت تأثیر چند Impact قرار گیرد و یک Impact هم می‌تواند روی چند کودک تأثیر بگذارد.)**  
  **👨‍🎨 نحوه رسم:**
* برای **M-M**، از **جدول واسط (Impact)** استفاده کن.
* یک خط از **Children به Impact** بکش (با M) و یک خط از **Impact به Children** بکش (با M).

**۶. رابطه بین FridgeDoor و Impact**

* یک **درب یخچال (FridgeDoor)** ممکن است تأثیرات مختلفی دریافت کند (مثلاً مقاومتش کم شود، خراب شود).
* یک **Impact** هم فقط به یک در مشخص اعمال می‌شود.  
  ✅ **رابطه: 1-M**  
  🔹 **(یک FridgeDoor می‌تواند چندین Impact داشته باشد، ولی هر Impact فقط به یک در خاص مربوط است.)**  
  **👨‍🎨 نحوه رسم:**
* یک خط از **FridgeDoor** به **Impact** بکش.
* در سمت **FridgeDoor** علامت **1** بگذار و در سمت **Impact** علامت **M**.

BEGIN;

-- تعریف انواع داده‌های سفارشی (Custom Types)

CREATE TYPE impact\_type AS ENUM (

'physical',

'psychological'

);

-- ایجاد جداول

CREATE TABLE IF NOT EXISTS Velociraptor (

id SERIAL PRIMARY KEY,

power INT NOT NULL,

behavior TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS Children (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

reaction TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS FridgeDoor (

id SERIAL PRIMARY KEY,

resistance INT NOT NULL,

status TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS Interaction (

id SERIAL PRIMARY KEY,

velociraptor\_id INT NOT NULL REFERENCES Velociraptor(id),

door\_id INT NOT NULL REFERENCES FridgeDoor(id),

result TEXT NOT NULL

);

-- جدول واسط برای رابطه چند به چند بین Children و Interaction

CREATE TABLE IF NOT EXISTS Children\_Interaction (

child\_id INT NOT NULL REFERENCES Children(id),

interaction\_id INT NOT NULL REFERENCES Interaction(id),

PRIMARY KEY (child\_id, interaction\_id)

);

CREATE TABLE IF NOT EXISTS Impact (

id SERIAL PRIMARY KEY,

velociraptor\_id INT NOT NULL REFERENCES Velociraptor(id),

door\_id INT NOT NULL REFERENCES FridgeDoor(id),

impact\_type impact\_type NOT NULL,

description TEXT NOT NULL

);

-- جدول واسط برای رابطه چند به چند بین Children و Impact

CREATE TABLE IF NOT EXISTS Children\_Impact (

child\_id INT NOT NULL REFERENCES Children(id),

impact\_id INT NOT NULL REFERENCES Impact(id),

PRIMARY KEY (child\_id, impact\_id)

);

-- درج داده‌های تستی

INSERT INTO Velociraptor (power, behavior) VALUES (10, 'Aggressive');

INSERT INTO Velociraptor (power, behavior) VALUES (8, 'Calm');

INSERT INTO Children (name, reaction) VALUES ('Tim', 'Scared');

INSERT INTO Children (name, reaction) VALUES ('Alice', 'Curious');

INSERT INTO FridgeDoor (resistance, status) VALUES (15, 'Closed');

INSERT INTO FridgeDoor (resistance, status) VALUES (10, 'Open');

INSERT INTO Interaction (velociraptor\_id, door\_id, result) VALUES (1, 1, 'Door held');

INSERT INTO Interaction (velociraptor\_id, door\_id, result) VALUES (2, 2, 'Door opened');

INSERT INTO Children\_Interaction (child\_id, interaction\_id) VALUES (1, 1); -- Tim در Interaction 1 شرکت کرد

INSERT INTO Children\_Interaction (child\_id, interaction\_id) VALUES (2, 1); -- Alice در Interaction 1 شرکت کرد

INSERT INTO Children\_Interaction (child\_id, interaction\_id) VALUES (1, 2); -- Tim در Interaction 2 شرکت کرد

INSERT INTO Impact (velociraptor\_id, door\_id, impact\_type, description) VALUES (1, 1, 'physical', 'Door resisted');

INSERT INTO Impact (velociraptor\_id, door\_id, impact\_type, description) VALUES (2, 2, 'psychological', 'Children scared');

INSERT INTO Children\_Impact (child\_id, impact\_id) VALUES (1, 1); -- Tim تحت تأثیر Impact 1 قرار گرفت

INSERT INTO Children\_Impact (child\_id, impact\_id) VALUES (2, 2); -- Alice تحت تأثیر Impact 2 قرار گرفت

SELECT \* FROM Children;

SELECT \* FROM Velociraptor;

SELECT \* FROM Interaction;

SELECT \* FROM Children\_Interaction;

SELECT \* FROM Impact;

SELECT \* FROM Children\_Impact;;

END;

کد الان من با توضیحاتش:  
-- Drop existing tables if they exist

DROP TABLE IF EXISTS Children\_Impact CASCADE;

DROP TABLE IF EXISTS Children\_Interaction CASCADE;

DROP TABLE IF EXISTS Impact CASCADE;

DROP TABLE IF EXISTS Interaction CASCADE;

DROP TABLE IF EXISTS FridgeDoor CASCADE;

DROP TABLE IF EXISTS Children CASCADE;

DROP TABLE IF EXISTS Velociraptor CASCADE;

-- Create tables

CREATE TABLE Velociraptor (

Id SERIAL PRIMARY KEY,

Power INT NOT NULL,

Behavior TEXT NOT NULL

);

CREATE TABLE Children (

Id SERIAL PRIMARY KEY,

Name TEXT NOT NULL,

Reaction TEXT NOT NULL

);

CREATE TABLE FridgeDoor (

Id SERIAL PRIMARY KEY,

Resistance INT NOT NULL,

Status TEXT NOT NULL

);

CREATE TABLE Interaction (

Id SERIAL PRIMARY KEY,

VelociraptorId INT REFERENCES Velociraptor(Id),

DoorId INT REFERENCES FridgeDoor(Id),

Result TEXT NOT NULL

);

CREATE TABLE Children\_Interaction (

ChildId INT REFERENCES Children(Id),

InteractionId INT REFERENCES Interaction(Id),

PRIMARY KEY (ChildId, InteractionId)

);

CREATE TABLE Impact (

Id SERIAL PRIMARY KEY,

VelociraptorId INT REFERENCES Velociraptor(Id),

DoorId INT REFERENCES FridgeDoor(Id),

ImpactType TEXT NOT NULL,

Description TEXT NOT NULL

);

CREATE TABLE Children\_Impact (

ChildId INT REFERENCES Children(Id),

ImpactId INT REFERENCES Impact(Id),

PRIMARY KEY (ChildId, ImpactId)

);

-- Insert data into Velociraptor table

INSERT INTO Velociraptor (Power, Behavior) VALUES (10, 'Aggressive');

INSERT INTO Velociraptor (Power, Behavior) VALUES (8, 'Calm');

-- Insert data into Children table

INSERT INTO Children (Name, Reaction) VALUES ('Tim', 'Scared');

INSERT INTO Children (Name, Reaction) VALUES ('Alice', 'Curious');

-- Insert data into FridgeDoor table

INSERT INTO FridgeDoor (Resistance, Status) VALUES (15, 'Closed');

INSERT INTO FridgeDoor (Resistance, Status) VALUES (10, 'Open');

-- Insert data into Interaction table

INSERT INTO Interaction (VelociraptorId, DoorId, Result) VALUES (1, 1, 'Door held');

INSERT INTO Interaction (VelociraptorId, DoorId, Result) VALUES (2, 2, 'Door opened');

-- Insert data into Children\_Interaction table

INSERT INTO Children\_Interaction (ChildId, InteractionId) VALUES (1, 1); -- Tim participated in Interaction 1

INSERT INTO Children\_Interaction (ChildId, InteractionId) VALUES (2, 1); -- Alice participated in Interaction 1

INSERT INTO Children\_Interaction (ChildId, InteractionId) VALUES (1, 2); -- Tim participated in Interaction 2

-- Insert data into Impact table

INSERT INTO Impact (VelociraptorId, DoorId, ImpactType, Description) VALUES (1, 1, 'Physical', 'Door resisted');

INSERT INTO Impact (VelociraptorId, DoorId, ImpactType, Description) VALUES (2, 2, 'Psychological', 'Children scared');

-- Insert data into Children\_Impact table

INSERT INTO Children\_Impact (ChildId, ImpactId) VALUES (1, 1); -- Tim was affected by Impact 1

INSERT INTO Children\_Impact (ChildId, ImpactId) VALUES (2, 2); -- Alice was affected by Impact 2

کد تصحیح شده  
  
DROP TABLE IF EXISTS Children\_Impact CASCADE;

DROP TABLE IF EXISTS Children\_Interaction CASCADE;

DROP TABLE IF EXISTS Impact CASCADE;

DROP TABLE IF EXISTS Interaction CASCADE;

DROP TABLE IF EXISTS FridgeDoor CASCADE;

DROP TABLE IF EXISTS Children CASCADE;

DROP TABLE IF EXISTS Velociraptor CASCADE;

-- Create tables

CREATE TABLE Velociraptor (

velociraptor\_id SERIAL PRIMARY KEY,

Power INT NOT NULL,

Behavior TEXT NOT NULL

);

CREATE TABLE Children (

child\_id SERIAL PRIMARY KEY,

Name TEXT NOT NULL,

Reaction TEXT NOT NULL

);

CREATE TABLE FridgeDoor (

door\_id SERIAL PRIMARY KEY,

Resistance INT NOT NULL,

Status TEXT NOT NULL

);

CREATE TABLE Interaction (

Interaction\_id SERIAL PRIMARY KEY,

VelociraptorId INT REFERENCES Velociraptor(velociraptor\_id),

DoorId INT REFERENCES FridgeDoor(door\_id),

Result TEXT NOT NULL

);

CREATE TABLE Children\_Interaction (

ChildId INT REFERENCES Children(child\_id),

InteractionId INT REFERENCES Interaction(interaction\_id),

PRIMARY KEY (ChildId, InteractionId)

);

CREATE TABLE Impact (

impact\_id SERIAL PRIMARY KEY,

VelociraptorId INT REFERENCES Velociraptor(velociraptor\_id),

DoorId INT REFERENCES FridgeDoor(door\_id),

ImpactType TEXT NOT NULL,

Description TEXT NOT NULL

);

CREATE TABLE Children\_Impact (

ChildId INT REFERENCES Children(child\_id),

ImpactId INT REFERENCES Impact(impact\_id),

PRIMARY KEY (ChildId, ImpactId)

);

--INSERTING TEST DATA IN TO THE TABLES

INSERT INTO Velociraptor (Power, Behavior) VALUES

(10, 'Aggressive'),

(8, 'Calm');

INSERT INTO Children (Name, Reaction) VALUES

('Tim', 'Scared'),

('Alice', 'Curious');

INSERT INTO FridgeDoor (Resistance, Status) VALUES

(15, 'Closed'),

(10, 'Open');

INSERT INTO Interaction (VelociraptorId, DoorId, Result) VALUES

(1, 1, 'Door held'),

(2, 2, 'Door opened');

INSERT INTO Children\_Interaction (ChildId, InteractionId) VALUES

(1, 1), -- Tim participated in Interaction 1

(2, 1), -- Alice participated in Interaction 1

(1, 2); -- Tim participated in Interaction 2

INSERT INTO Impact (VelociraptorId, DoorId, ImpactType, Description) VALUES

(1, 1, 'Physical', 'Door resisted'),

(2, 2, 'Psychological', 'Children scared');

INSERT INTO Children\_Impact (ChildId, ImpactId) VALUES

(1, 1), -- Tim was affected by Impact 1

(2, 2); -- Alice was affected by Impact 2