EXERCISE 08

1.Find all minions who have trained more than one minion.

CREATE VIEW trained\_by\_many\_trainer AS

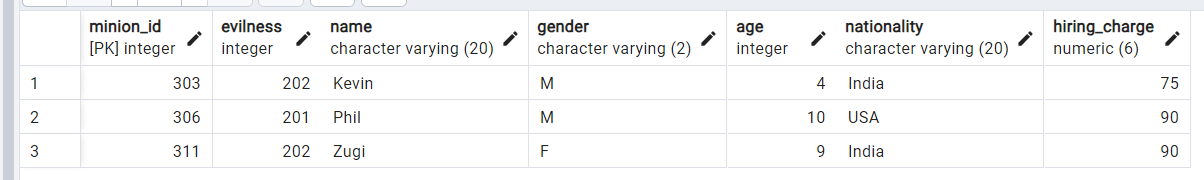
SELECT M.\* FROM minion AS M

INNER JOIN takes\_training USING(minion\_id)

INNER JOIN training USING(training\_id)

WHERE level != 4;

* SELECT \* FROM trained\_by\_many\_trainer;



2.Identify and list the first three currencies that have the highest currency conversion rate.

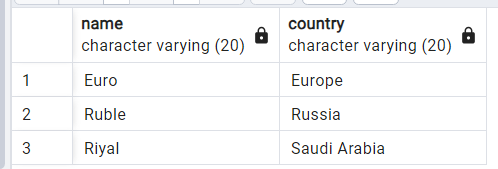
CREATE VIEW Highest\_Currency\_Rate AS

SELECT name, country FROM currency

ORDER BY b\_currency DESC

LIMIT 3;

* SELECT \* FROM Highest\_Currency\_Rate;



3.To extract the male minions of Japan or England who are committed in any mission.

CREATE VIEW Commited\_in\_mission AS

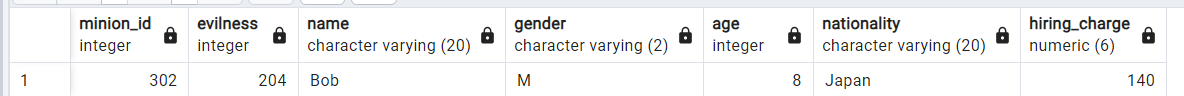
SELECT m.\* FROM MINION as m

INNER JOIN involved\_in USING(minion\_id)

WHERE gender = 'M' and

(nationality = 'Japan' or nationality = 'England');

* SELECT \* FROM Commited\_in\_mission;



4.Show the evilness description of all minions along with their hiring charge.

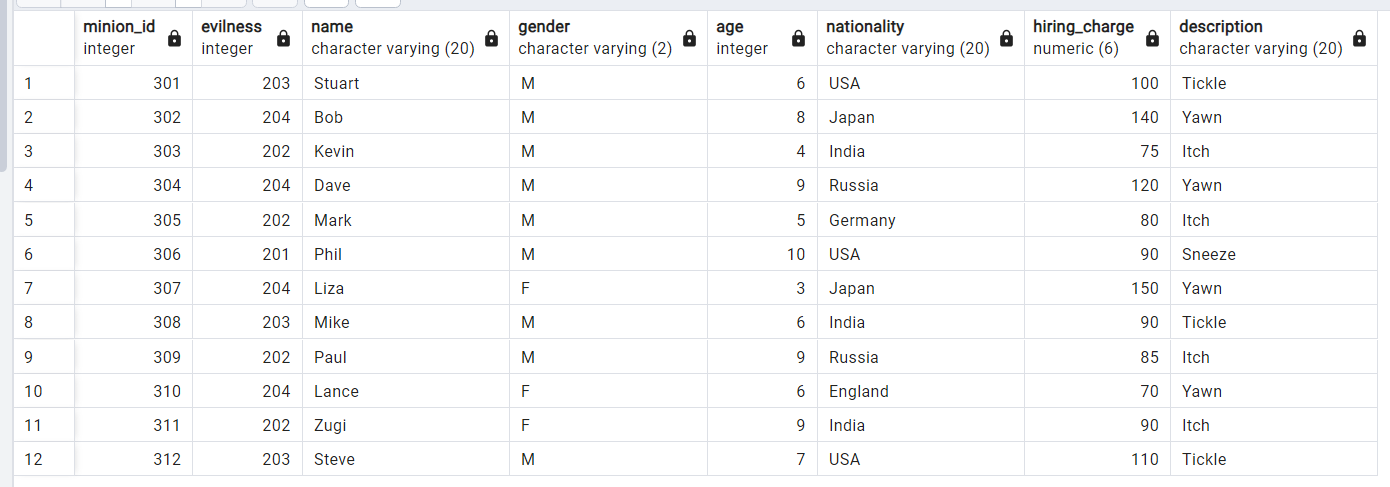
CREATE VIEW minion\_evilness\_description AS

SELECT minion\_id,evilness, name, hiring\_charge, description

FROM minion

JOIN evilness ON evilness = evil\_id;

* SELECT \* FROM minion\_evilness\_description;



5.Which minion(s) are associated with any mission currently? Who are their masters?

CREATE VIEW minions\_associated\_with\_mission AS

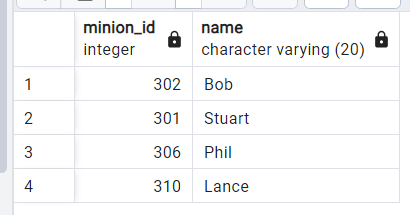
SELECT m.minion\_id, m.name FROM MINION AS m

INNER JOIN involved\_in USING(minion\_id)

INNER JOIN mission USING(mission\_id)

WHERE status = 'underway';

* SELECT \* FROM minions\_associated\_with\_mission;



6.Find all masters whose mission(s) were a success.

CREATE VIEW success\_mission\_master AS

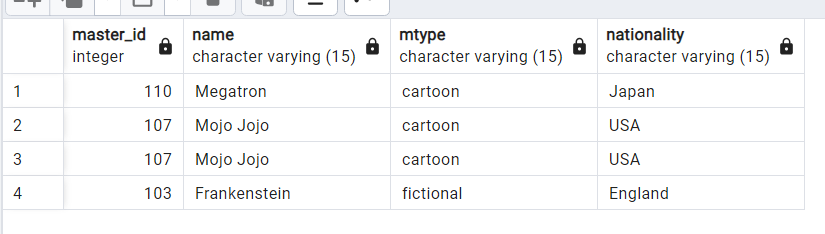
SELECT distinct m.\* FROM MASTER AS m

INNER JOIN involved\_in USING (master\_id)

INNER JOIN mission USING (mission\_id)

WHERE status = 'completed';

* SELECT \* FROM success\_mission\_master;



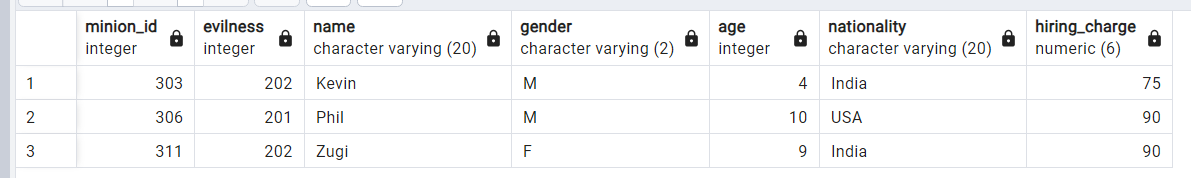
7.Create a save point.

SAVEPOINT savepoint\_A;

8.Add a trainer to the relation and check whether the view (created in Question 1) gets updated.

INSERT INTO training VALUES(405,2,310,'26-11-2023','27-11-2023')

* SELECT \* FROM trained\_by\_many\_trainer;



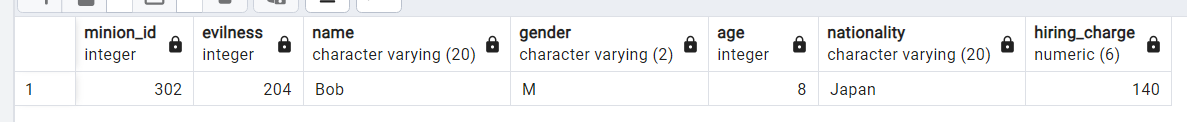
Conclusion: ***The view Does not get updated***

9.Add a male minion of Japan and check whether the view (created in Question 3) gets updated.

INSERT INTO minion VALUES

(042, 202, 'suren', 'M', 18, 'Japan', 100);

* SELECT \* FROM Commited\_in\_mission;



Conclusion: ***The view Does not get updated***

10.Create an updatable view for payment history.

CREATE VIEW updatable\_payment\_history AS

SELECT \* FROM payment;

* SELECT \* FROM updatable\_payment\_history;

