

1. No feedback was given, thus no changes have been made.

2.

A virtual view is just a stored query that runs when a user wants to see the view.
A materialized view runs the query when it is created and stores the resulting data.

3.

```
CREATE VIEW getAllUser1sFriends AS
  SELECT u.name
  FROM Users u JOIN (
    SELECT userID1, userID2 FROM friendships
    UNION SELECT userID2 as userID1, userID1 as userID2 FROM friendships
  ) AS f
  ON u.userID = f.UserID2
  WHERE f.userID1 = 1;
```

4.

- a) True.
- b) False.
- c) False.
- d) True.

5. We weren't sure whether to create a function or procedure, since the task says to create a procedure that returns the result, we ended up creating both a function and a procedure that is pretty close to 'returning' the result.

```
CREATE OR REPLACE FUNCTION
investment_return
(initial_investment INTEGER,
yearly_return FLOAT,
number_of_years INTEGER)
RETURNS Integer
LANGUAGE plpgsql
as
$$
declare
    investment_result INTEGER;
begin
    SELECT CAST(initial_investment * POWER(1 + yearly_return, number_of_years) AS INT)
    INTO investment_result;

    return investment_result;
end;
$$;
```

```

CREATE OR REPLACE PROCEDURE
investment_return
(initial_investment INTEGER,
yearly_return FLOAT,
number_of_years INTEGER,
INOUT investment_result INTEGER)
LANGUAGE plpgsql
AS $$
BEGIN
SELECT CAST(initial_investment * POWER(1 + yearly_return, number_of_years) AS INT)
INTO investment_result;
return;
END
$$;

```

6.

- a) True.
- b) False.

7.

An index in a database is basically a reference to one or more columns in a table. It can, for example, be used to efficiently query large tables for a small amount of data. Instead of looking through the entire table and selecting all the needed rows, you can search by index and only look for the data in the rows pertaining to a certain index, resulting in a more efficient query.

8.

- a) False.
- b) True.
- c) True.