TRUE FALSE Queries

Explanation: Append operating hours to every bill. Then, select all entries where the bill time is outside the operating hours, aka outside the range [Opening time, closing time]. If any entry is returned with this pattern, then return false, otherwise return true.

Assumes no bars are open overnight, as that would require either a comparison using full datetime objects with day/month/year included, or a much more complicated where condition. This pattern is currently true in the provided database, I checked before writing this query.

```
CREATE DEFINER=`av591`@`%` PROCEDURE `checkBillTimes`()
BEGIN

SELECT
CASE
        WHEN COUNT(*) > 0 THEN FALSE
   ELSE TRUE
END
AS AllBillsValid
FROM
        (select DISTINCT  Bills.time as billTime, Operates.start as barOpen, Operates.end as
barClose, Bills.bill_id
        from Bills
        INNER JOIN Operates
        ON Operates.bar = Bills.bar AND Operates.day = Bills.day) as uniqueBills
WHERE barOpen > billTime > barClose;

END
```

Explanation: Join frequents, drinker, and bar. Return entries where the state of the bar is not equal to the state of the drinker. Return true if no results are returned, otherwise return false.

```
CREATE DEFINER=`av591`@`%` PROCEDURE `checkFrequentsStates`()
BEGIN
SELECT
CASE
        WHEN COUNT(*) = 0 THEN true
   ELSE FALSE
END
AS sameState
FROM Frequents
INNER JOIN Drinker ON Drinker.name = Frequents.drinker
INNER JOIN Bar ON Frequents.bar = Bar.name
```

```
WHERE Drinker.state != Bar.state;


END
```

Explanation: Run subquery with every possible combo of beers sold by a bar where the left column of beers is always cheaper than the right column. Ignore =, as equal prices are permitted either way. Then, return any pair of beer where there is also an entry in the subquery where that combo is flipped. In other words, return all the possible combinations of beers with != prices, and if there exists two entries, one where b1 > b2 and one where b2 > b1, return that entry. If 0 entries like that are returned, return true, otherwise return false.

Also maybe I'm missing something really obvious, but this query took me like 2 hours to come up with. If it just really took me that long, fine, but please tell me if there was a better query to write.

```
CREATE DEFINER=`av591`@`%` PROCEDURE `validPrices`()
BEGIN
        (SELECT
   CASE
                WHEN COUNT(*) > 0 THEN FALSE
      ELSE TRUE
   END
   AS validPrices
        FROM
                (SELECT DISTINCT b1.barname as bar, b1.price as price1, b1.beername as
beer1,b2.price as price2,b2.beername as beer2
                FROM SellsBeer b1
                INNER JOIN SellsBeer b2
                ON b1.barname = b2.barname
                WHERE b1.price < b2.price)
                AS beerCombos,

                (SELECT DISTINCT b1.barname as bar, b1.price as price1, b1.beername as
beer1,b2.price as price2,b2.beername as beer2
                FROM SellsBeer b1
                INNER JOIN SellsBeer b2
                ON b1.barname = b2.barname
                WHERE b1.price < b2.price)
                AS beerCombos2
        WHERE
```

```
                beerCombos.beer1 = beerCombos2.beer2 and beerCombos.beer2 =
beerCombos2.beer1);
END
```