

# Joined Table Search

Case *Gesprek* has an N:1 relation with *Student*. *Gesprek* has a FK *studentid* The table *student* has a *naam*

## Gesprek Model

Create relation

In het Model *Gesprek* van het object

```
public function getStudent()
{
    return $this->hasOne(Student::className(), ['id' => 'studentid']);
}
```

## Gesprek View

In de GridView Widget

```
[
    'attribute' => 'student',
    'value' => 'student.naam',
],
```

## GesprekSearch model

1. line 3 add the public variable
2. line 9 add student searchbox
3. line 19 add join
4. line 22-24 add sorting
5. line 27 add filter in query

```
class GesprekSearch extends Gesprek
{
    public $student; // ADD student!
```

```

public function rules()
{
    return [
        [['formid', 'rolspelerid', 'status'], 'integer'],
        [['opmerking', 'student'], 'safe'], // ADD student!
    ];
}

...

public function search($params)
{

$query = Gesprek::find()
    ->joinwith(['examen', 'form', 'student']) // ADD student!
    ...

$dataProvider->sort->attributes['student'] = [ // ADD this block to support sorting
    'asc' => ['student.naam' => SORT_ASC],
    'desc' => ['student.naam' => SORT_DESC],
];
...
$query->andWhere(['like', 'student.naam', $this->student]); // ADD filter!

```

See: <https://www.yiiframework.com/wiki/653/displaying-sorting-and-filtering-model-relations-on-a-gridview>

---

Revision #1

Created 21 November 2020 18:05:26 by Admin

Updated 21 November 2020 18:18:33 by Admin