Artefakt som indeholder de queries der er blevet brugt til at lave tidsmålinger

Query 1 til Mysql

```
with cities as ( select * from Locations where name =
'Copenhagen')select distinct BookParts.id,
BookParts.title,BookParts.author
from cities
left join BookLocations on BookLocations.location_id = cities.id
left join BookParts     on BookParts.id = BookLocations.bookparts_id
order by BookParts.title;
```

Query 2 til Mysql

```
with selectedtitles as (select * from BookParts where title ='Peter
Pan' limit 1)
select selectedtitles.id, title, part, author, location_id,
index_in_book, `name`, ST_AsText(coordinate), population, timezone
from selectedtitles
left join BookLocations on BookLocations.bookparts_id =
selectedtitles.id
left join Locations on Locations.id = BookLocations.location_id;
```

Query 3 til Mysql

```
with author as (select * from BookParts where author = "Abraham
Lincoln"),
locs as (select id, JSON_OBJECT("name", name, "population",
population, "locaation", ST_AsText(coordinate)) as locObj from
Locations)
select author.id, title, part, author, json_arrayagg(locObj) as
"locations" from author
left join BookLocations on BookLocations.bookparts_id = author.id
left join locs on BookLocations.location_id = locs.id
group by author.id;
```

Query 4 til Mysql

```
with cities as (select *, ST_Distance(ST_GeomFromText("POINT(51 0)",
4326), coordinate)/1000 as km_away from Locations where
ST_Contains(ST_GeomFromText(ST_AsText(ST_Buffer(ST_GeomFromText("POI
NT(51 0)", 0), 125/111.226)), 4326), coordinate))
```

```
select distinct BookParts.id, title, part, author,
ST_AsText(coordinate) as point from cities
inner join BookLocations  on cities.id = BookLocations.location_id
left join BookParts on BookParts.id = BookLocations.bookparts_id;
```

Query 1 til MongoDb

```
db.Locations.aggregate([{$match:{name:'Copenhagen'}},{$limit:1},{$unwind:"$
booksRef"},{$project:{Ref:"$booksRef", coords: "$coordinate"}},{$lookup:{
from: "Books", localField: "Ref", foreignField: "id",
as:"Result"}},{$project:{author:{$arrayElemAt:["$Result.author",0]},
 title:{$arrayElemAt:["$Result.title",0]},
id:{$arrayElemAt:["$Result.id",0] }}},{$sort: {title: 1}}])
```

Query 2 til MongoDb

```
db.Books.aggregate([ {$match:{"title":"Peter Pan"}},
{$unwind:"$locations"}, {$project:{locRef:"$locations.locationRef"}},
{$lookup:{from: "Locations", localField: "locRef" , foreignField: "id" ,
as: "locationsInBook"}}, {$project:{
locationName:{$arrayElemAt:["$locationsInBook.name",0]},
coords:{$arrayElemAt:["$locationsInBook.coordinate",0]} }} ]).pretty()
```

Query 3 til MongoDb

```
db.Books.aggregate([ {$match:{"author":"Abraham Lincoln"}},
{$unwind:"$locations"}, {$project:{title:"$title",
locRef:"$locations.locationRef"}}, {$lookup:{from: "Locations", localField:
"locRef" , foreignField: "id" , as: "locationsInBook"}}, {$project: {
title: "$title", locationName:{$arrayElemAt:["$locationsInBook.name",0]},
coords:{$arrayElemAt: ["$locationsInBook.coordinate",0]} }} ]).pretty()
```

Query 4 til MongoDb

```
db.Locations.aggregate([ {$geoNear: { near: { type: "Point", coordinates: [
0, 51 ] }, distanceField: "distance", maxDistance: 200*1000, spherical:
true, key: "coordinate" }}, {$unwind:"$booksRef"}, {$lookup:{ from:
"Books", localField: "booksRef", foreignField: "id", as: "Book"}},
{$project: { Title: "$Book.title", Author: "$Book.author", Part:
"$Book.part", Coords: "$coordinate", Population: "$population", City:
"$name", DistanceInMeters: "$distance" }} ])
```