

# Test exam

## Exercise 1 – Databases

Given the following table containing data on projects and tasks:

**project:**

ProjectID	Name	Leader	Task	StartDate
1	Omega	Omar Nilsen	Planning	2022.09.01
1	Omega	Omar Nilsen	Construction	2023.03.10
2	Tunnel2	Turid Nelly	Startup	2022.11.10
1	Omega	Omar Nilsen	Finalizing	2023.11.01
2	Tunnel2	Turid Nelly	Make contract	2022.11.10
3	Seawind	Sean Winderly	Hire personell	2023.01.01
3	Seawind	Sean Winderly	Start planning	2023.01.01

where **ProjectID** is a unique ID for each project, **Name** is the name of the project, **Leader** is the person that is leading the project, **Task** is the name of a task associated with the project, and **StartDate** is the date that task is started.

1. What is wrong with the structure of the table?
2. How can these problems be fixed?

### Solution

**1** The table combines projects with their tasks in a single table. Each project may have multiple tasks, which leads to repeated data: Each project's data is repeated once for every task. This duplication makes the table much harder to maintain.

**2** We can fix the problem by splitting the table up, so that we have one table for projects, and one table for the tasks associated with each project, i.e.:

**project:**

ProjectID	Name	Leader
1	Omega	Omar Nilsen
2	Tunnel2	Turid Nelly
3	Seawind	Sean Winderly

**tasks:**

ProjectID	Task	StartDate
1	Planning	2022.09.01
1	Construction	2023.03.10
2	Startup	2022.11.10
1	Finalizing	2023.11.01
2	Make contract	2022.11.10
3	Hire personell	2023.01.01
3	Start planning	2023.01.01

## Exercise 2 – SQL

Given the following table containing data on music albums:

**album:**

album_id	name	artist	length	tracks
1	White Pony	Deftones	53	12
2	Abbey Road	The Beatles	47	17
3	Meta	Thy Catafalque	67	9
4	My Hero (Single)	Foo Fighters	4	1
5	Greatest Hits	Queen	19	5
6	Yellow Submarine	The Beatles	39	13
7	A Night At The Opera	Queen	43	12
8	Fear Inoculum	Tool	87	10
9	Folkesange	Myrkur	47	12
10	Rituals (Single)	Rolo Tomassi	4	1

Here `album_id` is a unique ID for each album, `name` is the name of the album, `artist` is the name of the artist that made the album, `length` is the play length in minutes for the album, and `tracks` is the number of tracks on the album.

1. What would the following query return:

```
SELECT name, length, tracks
FROM album
WHERE album_id = 2;
```

2. What would the following query return:

```
SELECT count(*) AS num_albums
FROM album
WHERE artist = 'The Beatles' OR length >= 60;
```

3. Write an SQL-query that finds the `name` of all albums with between 3 and 11 tracks.

4. Write an SQL-query that finds the average length and average number of tracks for all albums made by **Queen**.
5. Write an SQL-query that finds the name and length, in seconds, of all albums a length longer than 1000 seconds.

**Solution**

1

name	length	tracks
Abbey Road	47	17

2

num_albums
4

3

```
SELECT name
FROM album
WHERE tracks >= 3 AND tracks <= 11;
```

4

```
SELECT avg(length) AS avg_len, avg(tracks) AS avg_tracks
FROM album
WHERE artist = 'Queen';
```

5

```
SELECT name, length * 60 AS length_in_seconds
FROM album
WHERE length * 60 > 1000;
```