Task 7:

Query 1: Show all the races a specific swimmer participates in.

```
SELECT race.Event_ID, race_swimmer.Swimmer_ID, race.Race_ID, race.Pool_ID,
race.Distance, race.Stroke_Type, race.date

FROM race_swimmer, race

WHERE race_swimmer.Race_ID = race.Race_ID AND Swimmer_ID = 4;
```

A where clause join is called a cartesian join, with this join all-possible combinations of tuples are created when the query is run. In the example above all possible combinations of a race and the swimmer will be created. This is extremely inefficient and wastes computational resources.

The same result but more efficient query, which will only output the needed data.

```
SELECT race.Event_ID, race_swimmer.Swimmer_ID, race.Race_ID, race.Pool_ID,
race.Distance, race.Stroke_Type, race.date

FROM race_swimmer INNER JOIN race ON race_swimmer.Race_ID = race.Race_ID

WHERE race_swimmer.Swimmer_ID = 4;
```

Query 2: Selects all the swimmers participate in an event. Inefficient query:

```
SELECT *

FROM event_swimmer INNER JOIN swimmer ON event_swimmer.Swimmer_ID =
swimmer.Swimmer_ID

WHERE     event_swimmer.Event_ID = 1;
```

Selecting all the columns (SELECT *) in the result set will waste system resources.

Select the specific columns in the result set instead of selecting all of them.

Efficient query:

```
SELECT event_swimmer.Event_ID, swimmer.Swimmer_ID, swimmer.Fname,
swimmer.Lname, swimmer.sex

FROM event_swimmer INNER JOIN swimmer ON event_swimmer.Swimmer_ID =
swimmer.Swimmer_ID

WHERE     event_swimmer.Event_ID = 1;
```

Thus, selecting specific attributes in the result set will use less system resources for the same result, making it more efficient.

Query 3: Select all the swimmers that are in teams.

Inefficient query:

```
SELECT Fname, Lname, Sex
```

```
FROM swimmer
WHERE exists (
SELECT *
FROM team_swimmer
WHERE team_swimmer.Swimmer_ID = swimmer.Swimmer_ID
);
```

In this nested query it takes the team which the swimmer is in as a parameter and returns true if the swimmer is a team. This is inefficient because it will evaluate every tuple of the outside query.

Nesting queries is inefficient.

Efficient query:

```
SELECT Fname, Lname, Sex
FROM swimmer, team_swimmer
WHERE team_swimmer.Swimmer_ID = swimmer.Swimmer_ID;
```

This new query will make sure that there is a tuple from the tale swimmer will match with at most 1 tuple from the relation department.