Glöm inte att skriv in dnr

Spökboll

Databaser 1

Fastställd 2021-12-16 av

Axel Starck, Baozhen Wen, Lovisa Gustavsson, Philip Wahlberg och Sophia Quirk

OBS! Avsnittsbrytningen får ej skjutas över på nästa sida

Innehållsförteckning

1. Uppgiften 3

2. ERD och schema 3

2.1 ERD 3

2.2 DB-schema 4

2.3 SQLite 4

3. Queries 5

3.1 Queries 1 5

3.2 Queries 2 5

3.3 Queries 3 6

3.4 Queries 4 7

3.5 Queries 5 8

3.6 Queries 6 9

3.7 Queries 7 9

3.8 Queries 8 10

3.9 Queries 9 11

3.10 Queries 10 12

3.11 Views 14

players\_wins 14

winner 14

opponent\_2 14

total\_hits 14

hit\_avg 15

team\_result 15

player\_linkoping 16

4. Arbetsgång 16

5. Etik, säkerhet och integritet 17

5.1 GDPR och integritet 17

5.2 Etik 18

5.3 Säkerhet 18

6. Reflektioner 19

7. Källhänvisning 19

# 1. Uppgiften

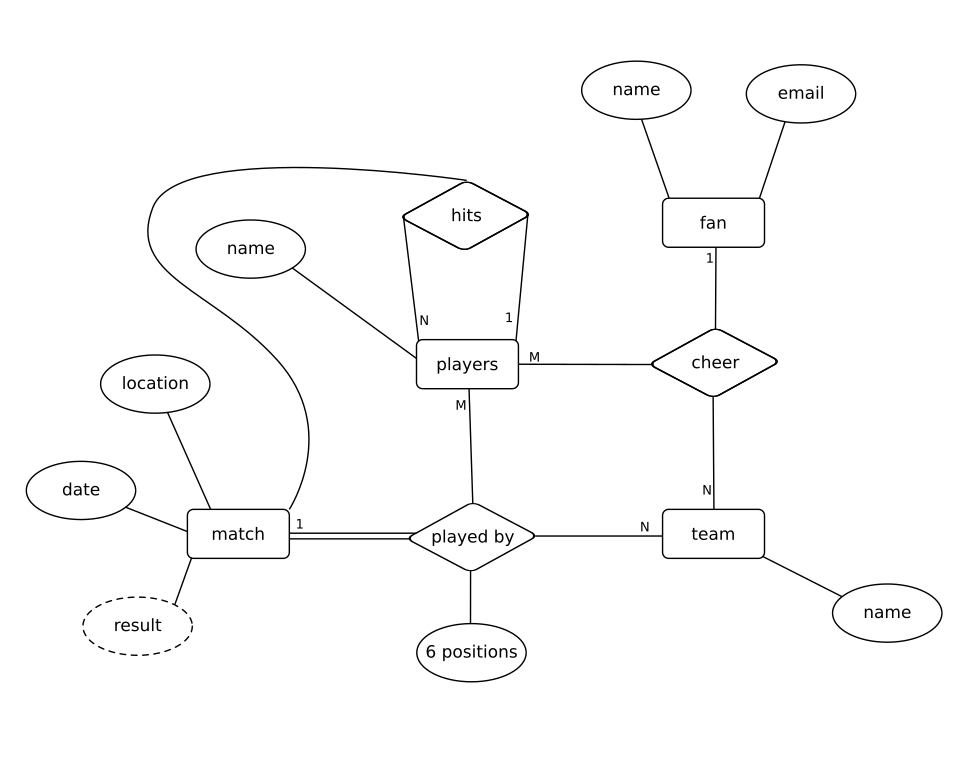
Syftet med uppgiften är att designa en databas utifrån en beställning från en kund. Bakgrunden till uppgiften är att arbetsgruppen läser kursen Databaser 1 där de ska driva ett projektarbete som går ut på att bygga en fungerande databas. Gruppen har fått tio queries som ska utföras. Det ingår även i uppgiften att redovisa ett ERD samt tabellschema för databasen.

Beställning som gruppen fått är att designa en databas till spökboll där data lagras om spelare och matcher. Vart matchen spelas och vilket datum ska lagras. Det ska gå att följa vilka spelare som spelar i vilka lag samt vilka tidigare lag de tillhört. Det är två lag per match och sex spelare från varje lag i en match och det ska lagras i databasen. Det ska även dokumenteras vilka spelare som träffat en annan spelare och hur många gånger samt om en spelare blivit träffad. De laget som först träffat alla i motståndarlaget har vunnit och det vinande laget dokumenteras. Databasen ska hantera fans som hejar på lag och spelare. Fansens har namn och epost som lagras i databasen.

# 2. ERD och schema

## 2.1 ERD

ER-modellering eller ERD är en relationsmodell vilken beskriver samband mellan saker inom domänen. En grundläggande ER-modell består av enhetstyper(entitet) och anger samband och egenskaper som kan finnas mellan enheter (Padron-McCarthy och Risch 2019, 31–32).

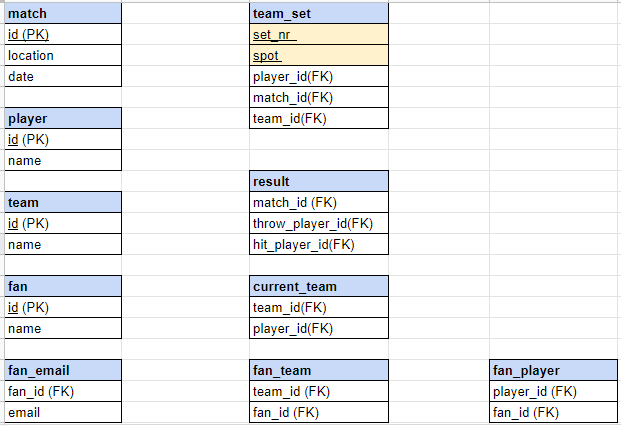


*Figur 1: ERD till databasen.*

ER-diagrammet som gruppen arbetat fram under arbetets gång, se figur 1, är utformat för att passa ihop med tabellerna, se figur 2, och stämma överens med beställningen som de fått. ER-diagrammet är utformat för att hjälpa gruppen med arbetet och att ha något att falla tillbaka på.

## 2.2 DB-schema

En tabell är en samling av data som lagras i ett tabellformat i en databas. Den består av kolumner och rader. I relationsmodellen är uppsättning av värde i tabeller. Det använder en modell av vertikala kolumner och horisontella rader. Ruta är enheten, där skär en rad och kolumn med varandra. En tabell har ett specificerat antal kolumner, men kan ha valfritt antal rader. Varje rad identifieras av ett eller flera värden. Kolumnen som markerar PK (primärnyckeln) där har olika värde på varje rad, dessutom kan det identifiera rader (Padron-McCarthy och Risch 2019, 83–93).



*Figur 2: Tabellerna som finns i databasen.*

Tabellerna över databasen, se figur 2, är det i projektarbetet som tagit längst tid. Det är en viktigt grund i arbetet och behövde få ta tid för att få det bra. De tabeller som gjorde det svårt var ”result” och ”team\_set” eftersom de behövdes för att lösa vissa queries men det var samtidigt svårt att få ihop dem på ett bra sätt. Det tabeller som visas i figur 2 är de som gruppen slutligen kom fram till och använt under arbetet med databasen.

## 2.3 SQLite

Gruppen valde SQLite för att utföra arbetet. SQLite är en relationsdatabas som är kompatibel med SQL. Det finns flera SQL-baserade system såsom MySQL och PostgreSQL. Till skillnad från andra SQL-baserade system använder SQLite inte en klient-serverarkitektur. Det undviker resurskrävande fristående processer vilken blir mer användarvänlig.

SQLite är ett smidigt verktyg med databashantering. Bärbara filer, lätta att kopiera, enklare att konfigurera. SQLite kan även utföra sig i en fullfjädrad utvecklingsmiljö.

# 3. Queries

## 3.1 Queries 1

Lista datum och ort för alla matcher som en viss spelare varit med i.

/\*Här tar vi ett exempel på spelare med player.id=6 eller med namn= Nobe Buckenhill\*/

|  |  |
| --- | --- |
| select player.name as player, match.date as date, match.location as location  from match  join team\_set  on match.id = team\_set.match\_id  join player  on team\_set.player\_id = player.id  where player.id = 6 |  |

## 3.2 Queries 2

Lista namn och antal träffar (utdelade och mottagna var för sig), antal vinster och antal matcher spelade för alla spelare som en given användare följer.

/\*Här listar vi en fan med id=3.

I denna query används view “player\_wins”.\*/

select fan.name as fan,

player.name as player,

throw\_count,

hit\_count,

player\_wins.match\_wins,

count(team\_set.set\_nr) as total\_matches

from player

join player\_wins

on player\_wins.player\_id = fan\_player.player\_id

join team\_set

on team\_set.player\_id = player.id

join fan

on fan.id = fan\_player.fan\_id

join fan\_player

on fan\_player.player\_id = player.id

left outer join (select throw\_player\_id,

count(throw\_player\_id) as throw\_count

from result

group by throw\_player\_id) as throws

on throws.throw\_player\_id = player.id

left outer join (select hit\_player\_id,

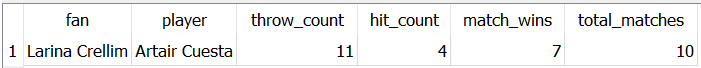
count(hit\_player\_id) as hit\_count

from result

group by hit\_player\_id) as hits

on hits.hit\_player\_id = player.id

where fan\_player.fan\_id = 3



## 3.3 Queries 3

Lista namn på de tre lagen som vunnit flest matcher.

/\*I denna query används view "winner”.\*/

select team.name as team, count(match\_id) as wins

from winner

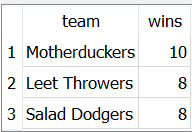
join team

on team.id = winner.winner

group by winner

order by wins desc

limit 3;



## 3.4 Queries 4

Lista alla matcher som ett lag spelat samt namn på motståndarlaget och status med ”Vinst”, ”Förlust” för varje match.

/\*Här listar vi matcher spelade av laget med id = 1,

dvs “The Artful Dodgers”.

I denna query används view “opponent\_2”. \*/

select opponent\_2.match,

team.name,

team1.name,

case when (winner.winner=opponent\_2.team)

then "vinst"

else "förlust"

end as resultat

from opponent\_2

join team

on team.id = opponent\_2.team

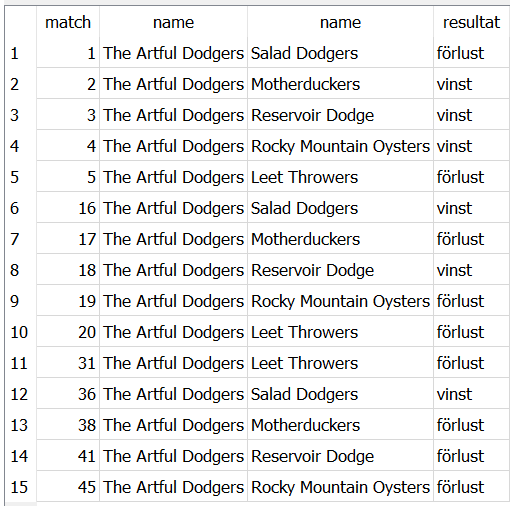
join team team1

on team1.id = opponent\_2.opponent

join winner

on opponent\_2.match=winner.match\_id

where opponent\_2.team="1"



## 3.5 Queries 5

Lista namn på alla lag som en viss spelare varit med i och antal träffar spelaren gjort och antal matcher spelaren spelat för varje lag.

/\*Här har vi valt spelaren med id = 2.\*/

select matches\_select.player\_id,

team,

throws,

matches\_played

from (select player\_id,

team.name as team,

count(result.throw\_player\_id) as throws

from team\_set

join result

on team\_set.player\_id = result.throw\_player\_id

and team\_set.team\_id = team.id

join team

on team.id = team\_set.team\_id

group by team.id, player\_id) as throws\_select

join (select player\_id,

team.name,

count(team\_set.team\_id) as matches\_played

from team\_set

join team

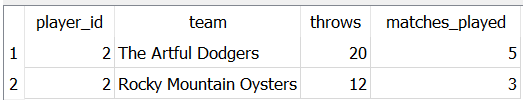
on team.id = team\_set.team\_id

group by team.id, player\_id) as matches\_select

on throws\_select.team = matches\_select.name

and throws\_select.player\_id = matches\_select.player\_id

where matches\_select.player\_id = 2



## 3.6 Queries 6

Lista namn på de tre spelare som varit med och vunnit flest matcher.

/\*Här används view “player\_wins”. \*/

select player.name,

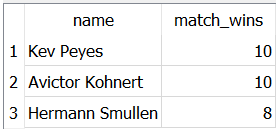
match\_wins from player\_wins

join player

on player.id = player\_wins.player\_id

order by match\_wins desc

limit 3



## 3.7 Queries 7

Lista namn på alla spelare som spelat match ett visst datum på en viss ort för att kunna lämna det vidare till polisen som utreder misstanke om ett brott.

/\*För att kunna få fram svar behöver användare ange datum och plats. Här tar vi ett exempel på matchen som skedde på “15/08/2019” och platsen på “Uppsala”\*/

SELECT player.name, match.date,match.location

FROM player

JOIN team\_set

ON player.id = team\_set.player\_id

JOIN match

ON team\_set.match\_id = match.id

WHERE match.date = "15/08/2019" and match.location = "Uppsala"



## 3.8 Queries 8

Lista namna och högst antal träffar (utdelade), lägst antal träffar (mottagna) och medelpoäng per match för alla spelare som en angiven användare följer.

/\*Vi avänder view “Total\_Hits” och “Hit\_avg.”

Här tar vi ett exempel på fan med id=10.

Under “Hits”, 0 betyder spelare hade inte blivit träffat under matchen och 1 menar tvärtom.\*/

SELECT Total\_Hits.id,player.name AS player,

hit\_avg.AVG\_hits AS AVG\_hits,

Total\_Hits.Hits AS highest\_hits,

CASE WHEN Total\_Hits.throw\_player NOT IN (SELECT result.hit\_player\_id FROM result

WHERE result.match\_id = Total\_Hits.id)

THEN 0

ELSE 1

END AS Hts

FROM player

JOIN Total\_Hits

ON player.id= Total\_Hits.throw\_player

JOIN hit\_avg

ON player.id=hit\_avg.player

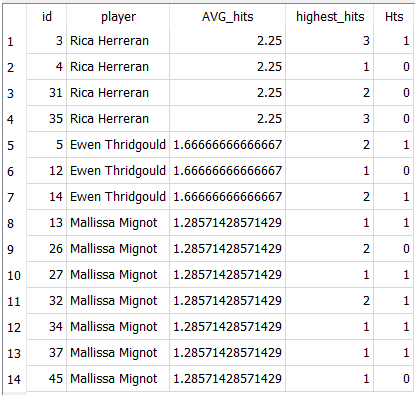
JOIN fan\_player

ON player.id=fan\_player.player\_id

WHERE fan\_player.fan\_id=10

GROUP BY Total\_Hits.throw\_player, Total\_Hits.id

ORDER BY hit\_avg.AVG\_hits DESC;



## 3.9 Queries 9

Lista namn på de spelare som vunnit en match i Visby eller Uppsala men aldrig spelat någon match i Linköping.

/\*Här använder vi en view “player\_linkoping”\*/

select player.name as player

from team\_set

join player

on player.id = team\_set.player\_id

join winner

on winner.winner = team\_set.team\_id

join match

on team\_set.match\_id = match.id

join (select player\_id, case when (player\_id = player\_in\_linkoping) then "L" else "ok" end as status

from team\_set

left join player\_linkoping

on player\_in\_linkoping = player\_id

group by player\_id) as not\_linkoping

on not\_linkoping.player\_id = team\_set.player\_id

where (location = "Visby"

or location = "Uppsala")

and status = "ok"

group by player

order by status desc



## 3.10 Queries 10

För en viss spelare visa namnen på de spelare den någon gång har träffat (med boll) i match och medeltalet för hur många träffar det är per match. Sortera så att högst medelvärde kommer först och sedan fallande.

/\* Vi tar ett exempel på spelare med id=10.

Vi använder view “team\_result” för att få fram medelvärde av team under matchen.\*/

SELECT match.id as MatchID, p1.name AS Throw\_Player,

p2.name as Hit\_Player, (CAST(team\_result.throws AS float)/6) as Average\_Throws

FROM match

JOIN team\_result

ON result.match\_id = team\_result.match\_id

JOIN result

ON match.id= result.match\_id

JOIN player p1

ON p1.id=result.throw\_player\_id

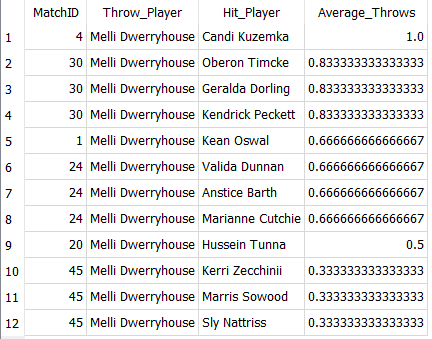
JOIN player p2

ON p2.id= result.hit\_player\_id

WHERE result.throw\_player\_id = 43

GROUP BY result.match\_id, result.throw\_player\_id, result.hit\_player\_id

ORDER BY Average\_Throws DESC



## 3.11 Views

### players\_wins

CREATE VIEW player\_wins as

select team\_set.player\_id,

count(team\_set.player\_id) as match\_wins

from team\_set

join winning\_sets

on team\_set.set\_nr = winning\_sets.set\_nr

group by team\_set.player\_id

### winner

CREATE VIEW winner as

select match\_id,

team\_id as winner

from team\_result

group by match\_id

### opponent\_2

CREATE VIEW opponent\_2 as

select distinct team\_set.match\_id as "match",

a.id as "team",

b.id as "opponent"

from team

join team a

on team.id=a.id

join team b

on b.id=playing\_teams.team\_id

join team\_set

on team\_set.team\_id=team.id

join playing\_teams

on b.id=playing\_teams.team\_id and team\_set.match\_id=playing\_teams.match\_id

where team != opponent

### total\_hits

CREATE VIEW Total\_Hits AS

SELECT match.id, p1.id AS throw\_player,

count(p2.id) AS "Hits"

FROM match

JOIN result

ON match.id= result.match\_id

JOIN player p1

ON p1.id=result.throw\_player\_id

JOIN player p2

ON p2.id= result.hit\_player\_id

GROUP BY result.match\_id, result.throw\_player\_id;

### hit\_avg

CREATE VIEW hit\_avg AS

SELECT player.id AS player,

MAX(Total\_Hits.Hits) AS highest\_hits,

MIN(Total\_Hits.Hits) AS lowest\_hits,

AVG(Total\_Hits.Hits) AS AVG\_hits FROM player

JOIN Total\_Hits

ON player.id= Total\_Hits.throw\_player

JOIN fan\_player

ON player.id=fan\_player.player\_id

GROUP BY Total\_Hits.throw\_player

ORDER BY highest\_hits DESC;

### team\_result

CREATE VIEW team\_result as

select result.match\_id, team\_set.team\_id, count(result.throw\_player\_id) as throws from result

join team\_set

on team\_set.player\_id = result.throw\_player\_id and result.match\_id = team\_set.match\_id

group by result.match\_id, set\_nr

order by result.match\_id, throws desc;

### player\_linkoping

CREATE VIEW player\_linkoping as

select team\_set.player\_id as player\_in\_linkoping

from team\_set

join match

on team\_set.match\_id = match.id

where location = "Linkoping"

group by player\_id;

# 4. Arbetsgång

När arbetsgruppen fick uppgiften började de med att lära igenom den och försöka bena ut vad som skulle göras. De började med att skissa upp ERD och tabeller på en whiteboardtavla för att få överblick och enkelt kunna ändra och göra om. Gruppen började också med att sätta sig in i det valda arbetsverktyget, SQLite, för att sedan lättare kunna arbeta i programmet. Arbetet med tabellerna tog väldigt mycket tid och gjordes om flera gånger innan de slutgiltiga tabellerna blev satta. Anledningen till att det tog lång tid var att gruppen ville få till perfekta tabeller för att kunna underlätta arbetet med queries längre fram. Tabellerna gjordes i Google kalkylark men även en del diktioners och skisser skedde vid whiteboardtavlan. Alla i gruppen var delaktiga i tabelluppbyggnade och var delaktiga i diktionerna. Det momentet fugerade bra för gruppen och kändes viktigt att göra tillsammans eftersom alla i gruppen fick förståelse för tabellerna och hur de hänger ihop vilket underlätta vid arbete med queries.

Samtidigt som arbetet med tabellerna och ERD pågick påbörjade arbetet med rapporten. Det på grund av att gruppen inte skulle hamna i tidsnöd på slutet utan istället jobba kontinuerligt med rapporten så att det inte skulle vara för mycket på slutet. Parallellarbete tog mycket tid eftersom tabellerna gjordes om mycket under arbetets gång. Det gruppen upplevde som tog tid med tabellerna var att det var mycket detaljer från uppgiften som skulle gå ihop på ett fungerande sätt vilket krävde ti och tankeverksamhet från gruppen. Gruppen har lagt ner mest tid på tabellerna under arbetet för att de är en väldigt stor grund i arbetet. Även ERD krävde en del korgringar fram och tillbaka eftersom det ska stämma överens med tabellerna och vara tydlig och enkel att förstå.

När gruppen kände sig nöjda med tabeller och ER-diagrammet började de men att bygga databasen i det valda programmet. Alla tabeller började läggas in och där efter fylla på dem med data. En del av datan gick att slumpa och där efter läggas in enkelt som exempelvis namna på fans, spelare och lag. All data gick dock inte att slumpa utan behövdes göras förhand för att det skulle stämma som exempelvis vem som träffat vem i vilken match. Det gjorde att det tog tid och att gruppen behövde göra det noggrant för att få allt att stämma överens. Detta arbete delade gruppen upp på så sätt att en började med queries med bara få data, en fortsatte med rapporten och tre jobbade med datan till databasen. Det gjorde att arbetet hela tiden gick framåt och att gruppen inte fastna eller tog för lång tid på sig på ett moment.

När all data var inlagd började gruppen lösa queries. Till en början gick det lite trögt och gruppmedlemmarna körde lätt fast men efter det löst sig lossnade det och det kändes lättare. Efter det lossnat blev arbetet med queries enklare och det gick ganska fort. Arbetet med rapporten fortsatte och det har under hela arbetsgången flytet på.

Gruppen har använt den schemalagda tiden i schemat till arbetet vilket har fungerat bra eftersom de då vet att de får ihop de beräknade timmarna som krävs för arbetet. I gruppen har de inte varit stenhårda med att komma i tid utan det har varit okej att komma lite sent. Det har ändå fungerat bra och gruppmedlemmarna har då inte behövt känna någon stress och press kring tider vilket de tro har lett till att arbetet har flytet på bra. Gruppen har använt lite olika medel för att kommunicera med varandra. De har haft en Messenger grupp för att enkel och smidigt kunna nå varandra om det varit något man velat informera de resterande i gruppen om. Det har även varit så att medlemmar i gruppen haft förkylningssymtom och varit bortrest vilket då gjort att gruppen använt sig att zoom för att kommunicera och lätt kunna prata och diskutera med varandra. Det har varit en väldigt bra lösning på problemet eftersom den som varit sjuk då ändå kunnat delta och vara med i diskussion. Det har också gjort det möjligt att hjälpa varandra i gruppen och de har då kunnat ta upp problem som uppstått. Gruppen har även delat en Google drivemapp för att kunna dela dokument och dylikt för att alla i gruppen ska ha tillgång till allt och komma åt det som behövs för arbetet.

# 5. Etik, säkerhet och integritet

## 5.1 GDPR och integritet

GDPR, även kalladdataskyddsförordningen, är en lag för att skydda och stärka den personliga integriteten. Lagen finns bland annat för att företag och dylikt inte ska kunna behandla personuppgifter hur som helst eller på ett felaktigt sätt. För att det ska skötas på rätt sätt behövs en registerförteckning och dataskydds ombud. Registerförteckningen ska innehålla hur informationen förvaras och vilken information som lagras. Det är även viktigt att kunderna vet vilken data som lagras om dem och godkänt det. Dataskydds ombudets uppgift är att se till att detta följs. All information som sparas måste ha ett syfte och det syftet måste finnas från början. Syftet är inget man kan ändra på i efterhand utan måste var förutbestämt. Det är även viktigt att endas de som behöver informationen har tillgång till den. Det gör att informationen behöver vara i säkert förvar så inte obehöriga kommer åt den (Sveriges kommuner och regioner. 2018).

Utifrån de uppgifter arbetsgruppen fått från beställaren kan gruppen utgå ifrån att beställaren har läst på om GDPR eftersom de inte vill lagra något olagligt eller oetiskt. Den enda personliga informationen som lagras är namn på spelare och fans samt epost till fansen. Beställaren har inget intresse av att lagra oetisk eller olaglig information som tex religion, etnicitet eller medicins bakgrund. Det gör gruppens arbete lättare eftersom de inte behöver ta ställning till det. Det som eventuellt skulle kunna ifrågasättas är varför beställaren vill ha tillgång till fansens epost eftersom det inte framgår i beställningen. Det skulle dock kunna förklaras med exempelvis att de vill kunna skicka ut information till fansen när deras favoritlag eller spelare, när de har match samt hur resultatet i matchen blev. De har då ett syfte med informationen vilket gör det okej att lagra om användaren godkänner det. Det gör att databasen följer lagarna för personlig integritet enligt GDPR.

## 5.2 Etik

Databasen lagra ingen oetisk information. Det som kan uppfattas som oetisk information är ofta samma som olaglig information som exempelvis religion och sexuell läggning. Denna information kan uppfattas som oetiskt eftersom det är personlig information som inte rör någon annan än individen. Eftersom beställaren inte frågar om något oetiskt minskar risken att användare tar illa upp eller väljer att inte använda tjänsten. Som tidigare nämnt hänger ofta det som är olagligt att lagra ihop med det som anses som oetiskt vilket gör att om lagen följs är det troligt att ingen oetisk information lagras. Ett problem med etik är att den skiljer sig från person till person vilket gör att det som en person upplever som oetiskt kan en annan tycka är etiskt rätt. Genom att undvika känslig information minska risken att uppröra människor eftersom det som riskeras att uppfattas på olika sätt då tas bort.

## 5.3 Säkerhet

Om databasen skulle sättas i användning på riktigt är det viktigt att tänka på säkerheten. Det är något som skulle behövas ses över om det skulle bli aktuellt att sätta databasen i drift. Det skulle behövas göra är att kontrollera vilka som har tillgång till databasen och bestämma vilka som ska få ha tillgång. Det behövs också sättas begräsningar på dem som har tillgång så att inte alla kan gå in och ändra i databasen hur som hällts. Det kan därför behövas olika access nivåer på användare som kan styra tillgången hos användare till databasen. Det måste gå att lita på datan vilket gör att begräsningarna behövs. Enligt GDPR behöver informationen vara så pass sker att ingen obehörig kommer åt den utan endas de som ska ha tillgång till informationen har det. Det blir viktigt att följa GDPR eftersom det kan bli höga böter om det inte följs. Då behövs de göras en registerförteckning samt tillsättas ett dataskydds ombud. Det behöver ses över att lagring och ändringar sker på ett bra sätt så att inget försvinner och inte går att få tillbaka. Därför kan det vara viktigt att införa backups som skulle kunna hjälpa om det skulle uppstå att något försvinner eftersom det kan vara väldigt dyrt att ersätta data eller helt omöjligt.

# 6. Reflektioner

Vi i gruppen upplever att uppgiften varit klurig. Vi har behövt tänka efter flera gånger och gjort om när vi kommit på en bättre lösning på ett problem. Det har gjort att vi vänt och vidrigt på problemen fram och tillbaka för att nå den optimala lösningen. Vi har tyckt att det varit mycket att ha i beaktning när vi bland annat arbetat med tabellerna eftersom det gäller att de blir rätt från början och det varit mycket fakta som ska gå ihop med varandra. Det har även upplevts att det varit svår att läsa sig till hur man skriver specifika “queries”.

Under arbetets gång har vi upplevt att det varit mycket ”ett steg fram och två steg bak” vilket ibland känts frustrerande. Detta uppstod mest i början av arbetet då vi bollat fram och tillbaka med olika lösningar. Vi hade ibland en lösning när vi gick hem men när vi sågs näst gång kändes det inte bra och vi gjorde om. Det känns som starten av uppgiften tog mycket tid samtidigt som vi lät de ta tid då det kändes som det var viktig att lägga tiden på det så vi blev nöjda. Mycket av arbetet har skett gemensamt vilket fugerat bra för oss eftersom det gett oss olika infallsvinklar på problemen. Vi har även delat upp arbetet men alltid visat och tagit hjälp av varandra för att dra fördel av varandras styrkor.

Det vi tar med oss till framtida liknade arbeten är att det underlättar arbetet om man lägger tid på grundarbetet så men har en bra bas att jobba ifrån. Vi skulle även våga testa mer än vad vi gjorde under detta arbete. Vi har varit kritiska mot oss själva och vårt arbete och istället för att testa om det fungerar har vi suttit och grubblat. Vi upplever också att vi fått en större och djupare förståelse för databaser och hur de fungerar av att få göra ett större projekt vilket också är något vi ta med oss till framtiden.

Vi har haft ett fint samarbete i gruppen. Alla har ställt upp och haft tålamod med varandra i en väldigt tålamodsprövande uppgift. Det har varit nyttigt att vi alla har haft olika ingångspunkter för att förstå att det finns flera olika sätt att lösa uppgiften på. Vi har haft bra dialog under hela arbetet vilket har underlätta och vi har uppmuntrat och stöttat varandra så gott vi kunnat.

# 7. Källhänvisning

Sveriges kommuner och regioner. 2018. *Den nya dataskyddsförordningen*. [Video]. <https://www.youtube.com/watch?v=Oba-E3B2ebo> (Hämtad 2/12 2021)

Padron-McCarthy T och Risch T. 2019. *Databasteknik.* 2.2. uppl. Studentlitteratur AB, Lund. Si. 31–32, Si. 83–93.