



Almqvist Software Solutions

Modelleringsförslag

Skultorps Golfklubb

Sammanfattning

Ett uppdrag i att ta fram en fungerande databasprototyp erhöles av organisationen "Skultorps Golfklubb". I arbetet har en modell för databassystemet tagits fram samt grundläggande SQL-operationer för uthämtning av data skrivits. Avslutningsvis presenteras ett vidareutvecklingsförslag för hur en välfungerande helhetslösning med integrerad databas och webb lösning kan se ut.

Olof Almqvist

ITBIG – 890123-4214

Innehållsförteckning

1 Inledning	1
1 Relationsmodell	2
2 DML för konstruktion av tabellerna	4
2.1 Översättning från relationsmodellen till tabeller	4
2.1.1 Entiteten Tävling	4
2.1.2 Entiteten Domare	4
2.1.3 Entiteten Ansvara	5
2.1.4 Entiteten Spelare	5
2.1.6 Entiteten Boll	6
2.1.7 Entiteten Golfbag	6
2.1.8 Entiteten Caddy	7
2.1.9 Entiteten Klubba	7
3 Transaktionsinmatning	8
3.1 Transaktionskommandon	8
3.2 Transaktionsuppgifter	8
3.3 Komplimenterande insättningar	10
4 Frågeoperationer	12
4.1 Lösta uppgifter frågeoperationer	13
4 Svar på frågor från Skultorps Golfklubb	20
5 Svar på frågor rörande XML	22
6 Egna antaganden och krav	25
7 Eventuellt förlorad semantik	26
7.1 Relationen Domare och Tävling	26
7.2 Namnförkortning på Klubba	26
7.3 Data som kan saknas i databasen	26
8 Förbättringsförslag	27
7 Bilagor	29
7.1 Källkod-DML för konstruktion av tabellerna	29
7.2 Källkod för transaktionsinmatning	31
7.3 Källkod för frågeoperationer	33
Referenser	38

1 Inledning

Syftet med detta arbete har varit att ta fram en fungerande prototypdatabas samt en dokumentation och användarmanual för verksamheten "Skultorps Golfklubb" (SG). Vid arbetets påbörjan hade en initial insats gjorts av ett flertal konsultfirmor som slutfört nödvändig kravspecifikation och tagit fram ett ER-diagram som modellerat och sammanfattat kraven.

Ett specifikt behov som verksamheten lyft fram inför arbetet är möjligheten att kunna vidareutveckla systemet så att data kan presenteras i ett användargränssnitt på en webbsida.

Systemutvecklingsfirman *Almqvist Software Solutions* har i och med denna rapport fullföljt erhållet uppdrag genom att ta fram en fungerande databasprototyp samt tillhandahållit dokumentation för hur information kan sättas in och tas ut ur databasen. Avslutningsvis har rekommendationer för hur informationen kan presenteras i ett GUI beskrivits.

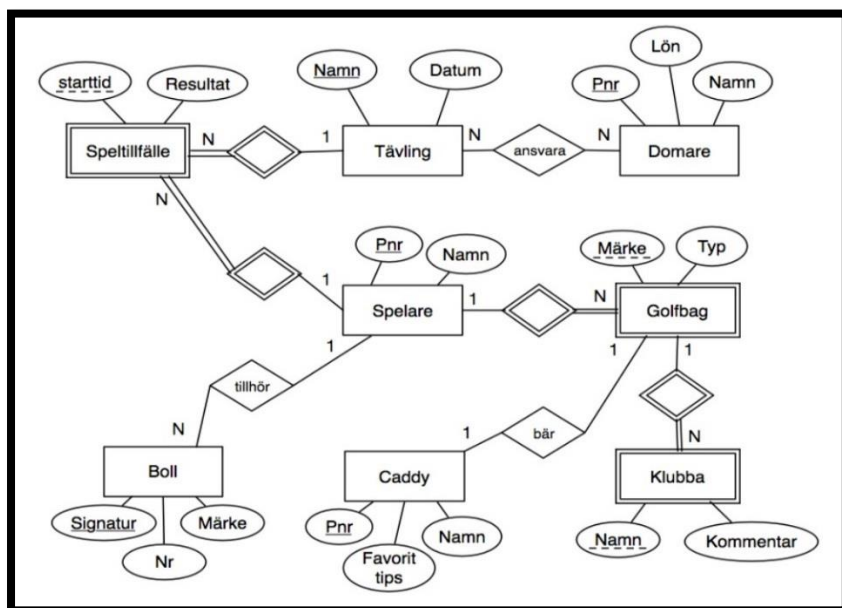
För att konstruera databasen har frågespråket SQL valts ut eftersom det är det tveklöst mest välkända, effektiva och uppdaterade frågespråket på marknaden enligt Kim (2015). I arbetet att ta fram en rekommendation för hur databassystemet kan expanderas till att presenteras i ett användargränssnitt har märkningsspråket XML använts eftersom det är lättanvänt och kompatibelt med i stort sett alla andra plattformar – vilket erbjuder SG framtida valfrihet och flexibilitet (IBM, u.å.).

I och med denna korta presentation hoppas vi att SG finner prototypen tillfredsställande och önskar förlänga kontraktet och arbeta vidare i samklang med vårt företag för att ta fram en vinnande helhetslösning.

1 Relationsmodell

Översättning från organisationens ER-diagram till en fungerande databas med Boyce-Codds Normalform (Jajodia, Ng & Springsteel 1983).

Det ER – eller Entity Relationship – diagram som på förhand upprättats av en konsultfirma översattes till en relationsdatamodell för att beskriva de tabeller, attribut, primär- och sekundärnycklar som kommer att utgöra de tabeller som följaktligen skapas i databashanteraren MySQL.



Figur 1. Det ER-diagram som en eller flera konsultfirmor förberett inför databaskonstruktionen.

Resultatet av analysen blev följande tabeller.

Tävling(<u>Namn</u> , Datum)	Domare(<u>Personnummer</u> , Namn, Lön)	Ansvara(<u>Tävling</u> , <u>Domare</u>)
Spelare(<u>Personnummer</u> , Namn)	Speltillfälle(<u>Starttid</u> , <u>Spelare</u> , <u>Tävling</u> , Resultat)	Boll(<u>Signatur</u> , <u>Spelare</u> , Nummer, Märke)
Golfbag(<u>Märke</u> , <u>Spelare</u> , Typ)	Caddy(<u>Personnummer</u> , <u>Golfbag</u> , <u>Märke</u> , Favorit_tips, Namn)	Klubba(<u>Namn</u> , <u>Golfbag.spelare</u> , <u>Golfbag.märke</u> , Kommentar)

Figur 2. Sammanfattning av de tabeller som upprättades vid översättning av ER-diagram till relationsmodell.

Tabell 1. Sammanfattning av de attribut som tagits med i relationsmodellen samt de referens- och primärnycklar som utkristalliserats.

<i>Tabell</i>	<i>Attribut</i>	<i>Referensnycklar</i>	<i>Primärnycklar</i>
Tävling	Namn, Datum		Namn
Domare	Personnummer, Namn, Lön		Personnummer
Ansvara	Tävling, Domare	Tävling, Domare	Tävling, Domare
Spelare	Personnummer, Namn		Personnummer
Speltillfälle	Spelare, Starttid, Namn, Resultat	Spelare, Tävling	Spelare, Tävling, Starttid
Boll	Spelare, Signatur, Nummer, Märke	Spelare	Signatur
Golfbag	Spelare, Märke, Typ	Spelare	Spelare, Märke
Caddy	Personnummer, Golfbag.Märke, Favorit_tips, Namn	Golfbag	Personnummer
Klubba	Spelare, Namn, Golfbag, Kommentar	Golfbag.märke, Golfbag.personnummer	Namn, Golfbag.märke, Golfbag.personnummer

2 DML för konstruktion av tabellerna

Tabellerna skapas i databasmiljön MySQL Workbench enligt följande syntax och semantik.

```
Create Table x (  
  y datatype  
  z datatype  
  Primary Key(y)  
  Foreign Key(z) references B(z)  
)engine=innodb;
```

Figur 3. Mall för hur skapelsen av en tabell ser ut i MYSQL Workbench.

“Create Table x” skapar tabellen med namnet ”x”. Kodraden ”y datatype” skapar en kolumn med namnet y med lämplig datatype exempelvis varchar om man vill att variabeln ska anpassa sig dynamiskt efter namnets storlek eller ”char” om man på förhand vill bestämma en behållare som består av ett fast antal tecken, d.v.s. om man på förhand vet hur stort ett insättningsvärde kommer att vara.

Primary Key(y) utnämner attributet y till primärnyckel för tabellen. Den är alltså unikt determinerande för de övriga attributen. Kommandot ”Foreign Key(z)” beskriver attributet z som referensnyckel till en annan tabell, ”B”.

Den sista kodraden ”engine=innodb” är en lagringsmotor som beskriver hur tabellen lagras på disken. En fördel med lagringsmotorn innodb är att den har stöd för primärnyckelsintegritet. Satsen stängs avslutningsvis med ett semikolon.

2.1 Översättning från relationsmodellen till tabeller

2.1.1 Entiteten Tävling

I relationsdatamodellen beskrevs tävlings-entiteten som ”Tävling(**Namn**, Datum)”. I tabellen finns inga referensnycklar och endast en unik primärnyckel. Eftersom namnen kommer vara av okänd storlek används datatypen ”varchar”.

Datum har flera olika förbestämda datatyper i MySQL Workbench (date, datetime, timestamp, time och year) men den mest lämpade i detta fall är typen ”date” som består av år, månader och dagar.

```
5 • create table tavling(  
6     namn varchar(20),  
7     datum date,  
8     primary key(namn)  
9 )engine=innodb;
```

Figur 4. Tabellen tävling i SQL.

2.1.2 Entiteten Domare

Nästa tabell som skapades var entiteten domare beskrivet i relationsmodellen som ”Domare(**Personnummer**, Namn, Lön)”. I detta fall användes två nya datatyper.

Dels användes datatypen "integer" för att beskriva attributet "lön". Detta eftersom integer lämpar sig väl för beräkningar och manipulation av värdet som det är anknutet till. Vidare användes datatypen "char" vid attributet "personnummer" eftersom personnumret är ett konstant värde som alltid är lika långt i alla lägen. Att då använda char optimerar prestanda och minskar risken för felinsättningar.

```
11 • create table domare(  
12     namn varchar(20),  
13     lon integer(5),  
14     personnummer varchar(11),  
15     primary key(personnummer)  
16 )engine=innodb;
```

Figur 5. Tabellen domare i SQL.

2.1.3 Entiteten Ansvara

Eftersom det råder ett N:M-samband mellan tävling och domare uppstår en särskild relation mellan entiteterna. I enlighet med de regler som beskriver relationsmodellen skall en särskild tabell – en sambandstabell – som består av primärnycklarna från domare och tävling läggas in i en mellantabell som kopplar samman de båda ursprungsentiteterna. Detta beskrivs i relationsmodellen som "Ansvara(Tävling, Domare)".

Padron-McCarthy & Risch (2015) beskriver sambandet på s.90 med orden: "Varje N:M-sambandstyp blir en egen tabell".

```
18 • create table ansvara(  
19     namn varchar(20),  
20     personnummer varchar(11),  
21     primary key(personnummer, namn),  
22     foreign key(personnummer) references domare(personnummer),  
23     foreign key(namn) references tavling(namn)  
24 )engine=innodb;
```

Figur 6. Tabellen Ansvara i SQL.

2.1.4 Entiteten Spelare

Entiteten spelare – "Spelare(Personnummer, Namn)" - översätts till en tämligen okomplicerad tabell med en unik primärnyckel i form av spelarens personnummer.

```
26 • create table spelare(  
27     personnummer varchar(11),  
28     namn varchar(20),  
29     primary key(personnummer)  
30 )engine=innodb;
```

Figur 7. Tabellen Spelare i SQL.

2.1.5 Entiteten Speltillfälle

Speltillfälle har en speciell relation gentemot två andra entiteter. Det råder en svag relation mot både "Spelare" och "Tävling". Detta leder till att primärnycklarna i dessa entiteter skall inkluderas i Speltillfälle-tabellen. Speltillfälle har alltså "Starttid" som en partiell nyckel som endast blir en unik nyckel tillsammans med

"Namn" från tabellen "Tävling" samt "Personnummer" från tabellen "Spelare". Dessa två nycklar är självfallet också referensattribut.

I relationsmodellen skrivs detta som "Speltillfälle(Starttid, Spelare, Tävling, Resultat)".

Denna arbetsgång beskrivs i Padron-McCarthy & Risch (2015) s.94 som *"Varje svag entitetstyp blir en tabell. Primärnyckeln består av den svaga entitetstypens partiella nyckel, kombinerad med den identifierade entitetstypens primärnyckel"*.

```
32 • create table speltillfalle(  
33     starttid varchar(10),  
34     resultat varchar(10),  
35     namn varchar(20),  
36     personnummer varchar(11),  
37     primary key(starttid, namn, personnummer),  
38     foreign key (namn) references tavling(namn),  
39     foreign key(personnummer) references spelare(personnummer)  
40 )engine=innodb;
```

Figur 8. Tabellen Speltillfälle i SQL.

2.1.6 Entiteten Boll

Mellan entiteterna "Boll" och "Spelare" råder ett kardinalitetssamband på 1:N vilket innebär att primärnyckeln från "1"-entiteten skall placeras i "N"-entiteten för att ge upphov till ett referensattribut som kopplar samman tabellerna.

Denna praxis beskrivs i Padron-McCarthy & Risch (2015) s.87 som *"Varje 1:N-sambandstyp blir ett referensattribut i "många"-entitetstypens tabell."*

I relationsmodellen skrivs detta som "Boll(Signatur, Spelare, Nummer, Märke)".

```
42 • create table boll(  
43     signatur varchar(20),  
44     nummer integer,  
45     marke varchar(15),  
46     personnummer varchar(11),  
47     primary key(signatur),  
48     foreign key(personnummer) references spelare(personnummer)  
49 )engine=innodb;
```

Figur 9. Tabellen Boll i SQL.

2.1.7 Entiteten Golfbag

Golfbag – "Golfbag(Märke, Spelare, Typ)" - har en egen partialnyckel i form av "Märke". Vidare är entiteten svag mot "Spelare" och stark mot "Klubba". Detta leder precis som i exempel 2.1.5 att "Personnummer" från "Spelare" placeras i tabellen golfbag som både referens- och del av primärnyckeln.

```
51 • create table golfbag(  
52     marke varchar(15),  
53     personnummer varchar(11),  
54     typ varchar(12),  
55     primary key(marke, personnummer),  
56     foreign key(personnummer) references spelare(personnummer)  
57 )engine=innodb;
```

Figur 10. Tabellen Golfbag i SQL.

2.1.8 Entiteten Caddy

Mellan Caddy och Golfbag råder ett 1:1-samband. Sambandstypen beskrivs av Padron-McCarthy & Risch (2015) s.88 som *"Varje 1:1-sambandstyp blir ett referensattribut i den ena entitetstypens tabell, men ibland kan man i stället slå ihop tabellerna"*. Man får alltså välja vilken tabell som skall inkorporera den andra tabellens primärnyckel som referensattribut. Ett sista alternativ är att göra en ny tabell.

I detta fall har vi valt att modellera så att attributet "märke" från tabellen "Golfbag" hamnar i tabellen "Caddy" som referensnyckel. Detta beskrivs i relationsmodellen som "Caddy(**Personnummer**, **Golfbag**, Favorit_tips, Namn)".

```
59 • create table caddy(  
60     personnummer varchar(11),  
61     favorit_tips varchar(30),  
62     namn varchar(20),  
63     marke varchar(15),  
64     primary key(personnummer),  
65     foreign key(marke) references golfbag(marke)  
66 )engine=innodb;
```

Figur 11. Tabellen Caddy

2.1.9 Entiteten Klubba

Entiteten Klubba är svag mot Golfbag som i sin tur är svag mot Spelare. Detta leder till att primärnycklarna från både Golfbag och Spelare placeras i tabellen Klubba och blir såväl referensnycklar som delar av tabellens primärnyckel.

Detta sammanfattas i relationsmodellen som "Klubba(**Namn**,

Golfbag.märke, **Golfbag.personnummer**, Kommentar)".

```
68 • create table klubba(  
69     namn varchar(20),  
70     kommentar varchar(40),  
71     marke varchar(15),  
72     personnummer char(11),  
73     primary key(namn, marke, personnummer),  
74     foreign key(marke) references golfbag(marke),  
75     foreign key(personnummer) references golfbag(personnummer)  
76 )engine=innodb;
```

Figur 12. Tabellen Klubba

3 Transaktionsinmatning

Transaktionsinsättningar gjordes initialt för att testa den modellerade databasen, de olika nycklarna, samt hur deras kopplingar till varandra genom referensnycklar.

Insättningar gjordes med frågespråket SQL enligt syntax och semantik i exemplet nedan:

```
Insert into x(A, B, C) values ('a','b','c');
```

Figur 13. Syntax för insättning av data i SQL-tabell.

Alltså, "insert into x" betyder ungefär "sätt in värden i tabell x. Kolumner "A", "B" och "C". Kolumnernas värden "a", "b" och "c".

3.1 Transaktionskommandon

För att minimera risken för misstag under processen av insättningar av data (inserts) och deletes samt uppdateringar av tabeller har kommandona **commit** och **rollback** använts.

Transaktioner används genom att initiera en specifik transaktion med kommandot "start transaction". Efter detta kan flera satser skrivas in i databashanteraren och avslutligen sparas i programmet med kommandot "commit". I det fallet att oönskade ändringar sparas med "**start transaction**" och "commit" kan man använda kommandot "rollback". Detta kommando tar bort de senaste sparade transaktionsändringarna.

3.2 Transaktionsuppgifter

Uppgift 1: Tävligen Sigges Sommargolf som hålls 10/7

```
78 /* Uppgift 1 */
79 • insert into tavling (namn, datum) values ('Sigges_Sommargolf','2016-07-10');
```

Figur 14. Transaktioner: uppgift 1.

Uppgift 2: Domarna med personnummer 790123-4444 och 810912-5555 som ansvarar på Sigges sommargolf och heter Simon och Sven. 790612-1212 ansvarar också för Sigges sommargolf och heter Eva. Alla har 12000kr i lön.

```
81 /* Uppgift 2 */
82 • insert into domare (namn, lon, personnummer) values ('Simon', '12000', '790129-4444');
83 • insert into domare (namn, lon, personnummer) values ('Sven', '12000', '810912-5555');
84 • insert into domare (namn, lon, personnummer) values ('Eva', '12000', '790612-1212');
85
86 • insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '790129-4444');
87 • insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '810912-5555');
88 • insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '790612-1212');
```

Figur 15. Transaktioner: uppgift 2.



Uppgift 3: Spelaren Bosse med personnummer 560123-6666 som startar 10:25 i tävlingen Sigges sommargolf.

```
90  /* Uppgift 3 */
91  • insert into spelare (personnummer, namn) values ('560123-6666', 'Bosse');
92  • insert into speltillfalle (starttid, namn, personnummer, resultat) values ('10:25', 'Sigges_Sommargolf', '560123-6666', '44');
93  • insert into golfbag (marke, personnummer, typ) values ('titleist', '560123-6666', 'Helgbag');
94  • insert into klubba (namn, personnummer, marke, kommentar) values ('Swinger', '560123-6666', 'titleist', 'Klubba att lita på');
95  • insert into caddy (marke, personnummer, namn, favorit_tips) values ('titleist', '560123-6666', 'Johannes_Brost', 'Tarecool');
```

Figur 16. Transaktioner: uppgift 3.

Uppgift 4: Golfbagen med märket "Nike" som är av typen "Tour" och tillhör spelaren med personnummer 730909-1111.

```
97  /* Uppgift 4 */
98  • insert into spelare (namn, personnummer) values ('Gudinnan Nike', '730909-1111');
99  • insert into golfbag (marke, personnummer, typ) values ('Nike', '730909-1111', 'Tour');
```

Figur 17. Transaktioner: uppgift 4.

Uppgift 5: Klubban med namn "Driver" som har en kommentar "Spikrak och kort" som tillhör spelaren med personnummer 730909-1111.

```
101 /* Uppgift 5 */
102 • insert into klubba (namn, kommentar, personnummer, marke) values ('Driver', 'Spikrak och kort', '730909-1111', 'Nike');
```

Figur 18. Transaktioner: uppgift 5.

Uppgift 6: Caddyn som heter Anna och ofta ger favorittipset "Knyt skorna hårdare". Hon bär golfbagen till spelaren som heter Sune som har personnummer "670808-2222".

```
104 /* Uppgift 6 */
105 • insert into spelare (namn, personnummer) values ('Sune', '670808-2222');
106 • insert into golfbag (marke, typ, personnummer) values ('Adidas', 'Svart', '670808-2222');
107 • insert into caddy (namn, favorit_tips, personnummer, marke) values ('Anna', 'Knyt skorna hårdare', '890123-4214', 'Adidas');
```

Figur 19. Transaktioner: uppgift 6.

Uppgift 7: Speltillfället som startade 13:10 med resultatet 72 slag som utfördes av Sune. Bollen som användes hade signaturen "Hjärta" och var av märket "Titleist".

```
109 /* Uppgift 7 */
110 • insert into speltillfalle (starttid, resultat, namn, personnummer) values ('13:10', '72', 'Sigges_Sommargolf', '670808-2222');
111 • insert into boll (signatur, marke, personnummer, nummer) values ('Hjärta', 'Titleist', '670808-2222', '583');
```

Figur 20. Transaktioner: uppgift 7.

Uppgift 8: Klubban med namn "Driver" som har kommentaren "Längst och snedast på touren" och som caddyn Petra bär. Hon ger alltid tipset "Svinga lugnt".

```
113 /* Uppgift 8 */
114 • insert into spelare (namn, personnummer) values ('Olle', '900101-8888');
115 • insert into golfbag (marke, personnummer, typ) values ('Lacoste', '900101-8888', 'Getskinn');
116 • insert into caddy (namn, personnummer, favorit_tips, marke) values ('Petra', '910202-7777', 'Svinga lugnt', 'Lacoste');
117 • insert into klubba (namn, kommentar, personnummer, marke) values ('Driver', 'Längst och snedast på touren', '900101-8888', 'Lacoste');
```

Figur 21. Transaktioner: uppgift 8.

Uppgift 9: Bollen med signaturen ”Tre prickar” dömdes ut som ospelbar av domaren med personnummer 770202-3333. Bollen spelades av Benny med personnummer 790101-4343. Eftersom bollen blev ospelbar blev Benny diskad från tävlingen.

```
119 /* Uppgift 9 */
120 • insert into spelare (namn, personnummer) values ('Benny', '790101-4343');
121 • insert into domare (namn, personnummer, lon) values ('Petrus', '770202-3333', '18990');
122 • insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '770202-3333');
123 • insert into boll (signatur, marke, personnummer, nummer) values ('Treprickar', 'Zeitgeist', '790101-4343', '985');
124 • insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Sigges_Sommargolf', '790101-4343', 'Diskad', '08:00');
```

Figur 22. Transaktioner: uppgift 9.

Uppgift 10: Klubban med kommentaren ”Säkra Bettan” som tillhör Stina. Hon spelade 12:05 i Sigges sommargolf men fick resultatet ”Diskad” eftersom domaren Jens dömde ut hennes boll av märke ”Nike” de den var modifierad att gå extra långt.

```
126 /* Uppgift 10 */
127 • insert into spelare (namn, personnummer) values ('Stina', '950101-4141');
128 • insert into klubba (kommentar, marke, personnummer, namn) values ('Säkra Bettan', 'Nike', '950101-4141', 'Putt');
129 • insert into golfbag (marke, personnummer, typ) values ('Nike', '950101-4141', 'Ormskinn');
130 • insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Sigges_Sommargolf', '950101-4141', 'Diskad', '12:05');
131 • insert into domare (namn, personnummer, lon) values ('Jens', '850102-7878', '18000');
132 • insert into boll (signatur, marke, personnummer, nummer) values ('Extralångt', 'Nike', '950101-4141', '156');
```

Figur 23. Transaktioner: uppgift 10.

3.3 Komplimenterande insättningar

För att få in mer data så att frågeoperationernas resultat illustreras bättre har extrainsättningar gjorts utöver de grundläggande komplement som efterfrågades.

```
134 /* Extrainserts */
135
136 • insert into spelare (namn, personnummer) values ('Siegfried', '880101-8888');
137 • insert into spelare (namn, personnummer) values ('Beowulf', '770101-7777');
138 • insert into spelare (namn, personnummer) values ('Lancelot', '660101-6666');
139
140 • insert into tavling (namn, datum) values ('Vikingatävlingen', '2016-07-15');
141 • insert into tavling (namn, datum) values ('Riddarspelen', '2014-10-06');
142 • insert into tavling (namn, datum) values ('Excaliburtävlingen', '2012-09-10');
143
144 • insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Vikingatävlingen', '880101-8888', '150', '13:00');
145 • insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Riddarspelen', '770101-7777', '125', '15:00');
146 • insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Excaliburtävlingen', '660101-6666', '55', '16:00');
147
148 • insert into domare (namn, personnummer, lon) values ('Augustus', '550102-5589', '10500');
149 • insert into domare (namn, personnummer, lon) values ('Julius', '400230-7810', '73500');
150 • insert into domare (namn, personnummer, lon) values ('Nero', '731224-6532', '12453');
151
152 • insert into ansvara (namn, personnummer) values ('Vikingatävlingen', '550102-5589');
153 • insert into ansvara (namn, personnummer) values ('Riddarspelen', '400230-7810');
154 • insert into ansvara (namn, personnummer) values ('Excaliburtävlingen', '731224-6532');
```

Figur 24. Extrainsättningar.

```

156 • insert into boll (signatur, marke, personnummer, nummer) values ('Vinnarbollen', 'CarlLagerfeldt', '880101-8888', '123456789');
157 • insert into boll (signatur, marke, personnummer, nummer) values ('Apollobollen', 'AthenaSports', '770101-7777', '987654321');
158 • insert into boll (signatur, marke, personnummer, nummer) values ('Zeusbollen', 'AthenaSports', '660101-6666', '13579');
159
160 • insert into golfbag (marke, personnummer, typ) values ('CarlLagerfeldt', '880101-8888', 'Brun');
161 • insert into golfbag (marke, personnummer, typ) values ('AthenaSports', '770101-7777', 'Guld');
162 • insert into golfbag (marke, personnummer, typ) values ('AthenaSports', '660101-6666', 'Silver');
163
164 • insert into klubba (namn, kommentar, marke, personnummer) values ('Putt', 'Vann tävlingen', 'CarlLagerfeldt', '880101-8888');
165 • insert into klubba (namn, kommentar, marke, personnummer) values ('Slag', 'Hole in one', 'AthenaSports', '770101-7777');
166 • insert into klubba (namn, kommentar, marke, personnummer) values ('Slag', 'Ok', 'AthenaSports', '660101-6666');
167
168 • insert into caddy (personnummer, favorit_tips, marke, namn) values ('550123-4214', 'NEVER Give UP', 'CarlLagerfeldt', 'Olaus');
169 • insert into caddy (personnummer, favorit_tips, marke, namn) values ('400901-5598', 'Se saken positivt', 'AthenaSports', 'Linn');
170 • insert into caddy (personnummer, favorit_tips, marke, namn) values ('800420-5665', 'Kör bara!', 'AthenaSports', 'Tony');

```

Figur 25. Extrainsättningar.



4 Frågeoperationer

I materialet som gavs erhöles en lista av frågeoperationer som databasprototypen skall kunna besvara. Ett arbete har gjorts för att ta fram förslag på hur informationen kan uthämtas samt de alternativ och utmaningar som föreligger.

Nedan följer en kort sammanfattning av de kommandon som använts i arbetet.

Tabell 2. Sammanfattning av grundläggande SQL-kommandon.

Grundläggande kommandon	Förklaring
SELECT	Väljer ut en viss kolumn i en tabell.
FROM	Väljer ut den tabell värden skall tas "IFRÅN".
WHERE	Specifiserar ett villkor för SELECT & FROM t.ex. "SELECT namn from spelare WHERE personnummer = X"
INSERT INTO	Sätter in data i en på förhand skapad tabell.
CREATE TABLE	Skapar en tabell av data. I själva verket två kommandon "CREATE" och "TABLE".

Tabell 3. De "jämförelsekommandon" som har använts eller övervägts i arbetet.

Jämförelsekommandon	Förklaring
LIKE	Används för sökningar på t.ex. en bokstav. Tar fram ord där t.ex. bokstaven "J" finns.
RLIKE	Liknande LIKE.
AND	Kopplar samman två villkor.
OR	Två alternativa villkor.
NOT	Ett villkor som inte är lika med.
IN/NOT IN	Villkor där en kolumn finns (IN) eller inte finns (NOT IN).
EXISTS/NOT EXISTS	Undersöker om några rader ges som svar på frågan.
ANY	Om det finns några som uppfyller villkoret.
SOME	Om det finns några som uppfyller villkoret.
ALL	Om alla uppfyller villkoret.
GROUP BY	En s.k. "mängdoperation". Grupperar värden enligt en mängd t.ex. alla som har namnet "Sven".
HAVING	Kan användas tillsammans med GROUP BY t.ex. enligt "select namn from person group by hemstad having lön > 15 000".

Tabell 4. Beräknings- och svarskommandon.

Beräkningskommandon och svarskommandon	Förklaring
AVG()	Tar fram medelvärdet av en serie värden. En s.k. aggregatfunktion eftersom den arbetar på en hel kolumn.
MAX()	Tar fram det högsta värdet av en serie värden. En s.k. aggregatfunktion eftersom den arbetar på en hel kolumn.
SUM()	Tar fram summan av en serie värden. En s.k. aggregatfunktion eftersom den arbetar på en hel kolumn.
DISTINCT	Tar bort dubletter.
ORDER BY	Beskriver vilka värden svaret ska sorteras efter t.ex. namn eller nummer.
DESC	Sortering från början till slut.
ASC	Sortering från slut till början.
COUNT	Räknar förekomsten av ett värde.

Tabell 5. Kommandon för hantering av tabelldata.

Hantering av tabelldata	Förklaring
UPDATE	Implementerar en förändring i data i en tabell.
DELETE	Raderar en cell, tupel, kolumn eller tabell.

CREATE VIEW	Skapar en temporär tabell som inte lagras i databasen.
STAR TRANSACTION	Påbörjar en transaktion som kan sträcka sig över flera satser kod. Inom denna "ram" kan ändringar göras på databasen som sedan kan ångras.
COMMIT	Bekräftar de ändringar som gjorts efter "START TRANSACTION"-kommandot.
ROLLBACK	Backar tillbaka till innan ett "COMMIT"-kommando kördes.

4.1 Lösta uppgifter frågeoperationer

Uppgift 1: Hämta ut namnet på domaren med personnummer 790123-4444.

```
174  /* Uppgift 1 */
175  • select namn from domare
176  where personnummer = '790123-4444';
```

Figur 26. Frågeoperationer: uppgift 1.

Svar: Namnet är "Simon".

Uppgift 2: Hämta ut signaturen på bollen som spelades av 560123-6666.

```
178  /* Uppgift 2 */
179  • select signatur from boll
180  where personnummer = '560123-6666';
```

Figur 27. Frågeoperationer: uppgift 2.

Svar: Frågan ger NULL som svar eftersom ingen insert gjorts med en golfboll kopplad till personnumret "560123-6666".

Uppgift 3: Hämta typen på golfbagen som ägs av spelaren 560123-6666.

```
183  /* Uppgift 3 */
184  • select typ from golfbag
185  where personnummer = '560123-6666';
```

Figur 28. Frågeoperationer: uppgift 3.

Svar: Frågan ger "Helgbag" som svar. Det är den golfbag som ägs av spelaren Bosse – som saknar boll.

Uppgift 4: Hämta ut namnet på spelarna som har bollar av märket "Titleist".

```
187  /* Uppgift 4 */
188  • select namn from spelare
189  where spelare.personnummer = (
190      select boll.personnummer
191      from boll
192      where marke = 'Titleist');
```

Figur 29. Frågeoperationer: uppgift 4.

Svar: Svaret blir spelarnamnet "Sune".

Uppgift 5: Hämta resultatet för speltillfällena som spelades med bollar av märket "Nike".

```
195  /* Uppgift 5 */
196  • select resultat from speltillfalle
197  where speltillfalle.personnummer =
198  (select spelare.personnummer from spelare
199  where spelare.personnummer =
200  (select boll.personnummer from boll
201  where marke='Nike'));
```

Figur 30. Frågeoperationer: uppgift 5.

Svar: Svaret blir resultatet "Diskad".

Uppgift 6: Hämta personnumret för de spelare som har samma namn. (Tips: skapa instanser av samma tabell och jämför dess namn).

```
208  /*Uppgift 6 */
209  • insert into spelare (namn, personnummer) values ('Sune', '888888-7777');
210
211  • CREATE VIEW dubblettnamn AS
212  SELECT namn, count(*)
213  FROM spelare
214  group by namn
215  having COUNT(*) > 1;
216
217  • create view alla_namn AS
218  select namn, personnummer from spelare;
219
220  • select personnummer as 'Personnummer dubblettnamn' from alla_namn
221  where alla_namn.namn = (select dubblettnamn.namn from dubblettnamn);
```

Figur 31. Frågeoperationer: uppgift 6.

Svar: Frågan ger "670808-2222" samt "888888-7777" som svar. Detta är personnummer till de namn som finns i minst dubbel uppsättning i tabellen "spelare". I detta fall är det två olika personnummer till de spelare som har namnet "Sune" vilket är det enda namn som det finns fler än en av.

Uppgift 6 kräver användningen av "Views". Det vill säga en instans av en tabell där man valt ut särskilda kolumner eller tupler. I detta fall skapas två instanser av tabellen "spelare". Dels en tabell som innehåller namnet på de namn som förekommer mer än en gång; samt en tabell som innehåller alla namn. Avslutningsvis hämtas personnummer från "alla.namn"-instansen för de namn som förekommer i "dubblettnamn"-instansen.

Uppgift 7: Hämta personnumret för de domare som inte ansvarat för någon tävling.

```
224  /* Uppgift 7 */
225  • select domare.personnummer from domare
226  where domare.personnummer not in (
227  select ansvara.personnummer
228  from ansvara);
```

Figur 32. Frågeoperationer: uppgift 7.

Svar: Svaret blir personnummer "850102-7878" vilket är personnummer till den enda domare som inte har haft ansvar för någon tävling och därmed inte finns med i "ansvara"-tabellen. För att hitta detta värde används kommandot "not in" dvs "finns inte i".

Personnummer dublettnamn
670808-2222
888888-7777

Figur 33. Svar på uppgift 7.

Uppgift 8: Hämta tipset som gavs åt spelaren med personnummer 660808-5555 som hade resultat 72 slag i tävlingen "Ryder Cup".

```

242 • select favorit_tips from caddy
243   where caddy.marke = (
244     select golfbag.marke from golfbag
245     where golfbag.personnummer = (
246       select spelare.personnummer from spelare
247       where spelare.personnummer = (
248         select speltillfalle.personnummer from speltillfalle
249         where speltillfalle.resultat = 72 and speltillfalle.namn = 'Ryder Cup' and speltillfalle.personnummer = '660808-5555'))));

```

Figur 34. Frågeoperationer: uppgift 8.

Svar: I denna frågeoperation eftersöks ett tips som gav till en specifik spelare med ett känt personnummer som fick 72 slag i en tävling vid namn "Ryder Cup". **Eftersom inget resultat med värdet 72 kopplat till en tävling med namnet "Ryder Cup" samt personnummer "660808-5555" lagts in i databasen ges inget svar.** Det finns en spelare med 72 slag i databasen men detta resultat är kopplat till en annan tävling. Eftersom villkoret var att både resultat skulle vara 72 och tävlingen heta Ryder Cup finns inget favorittips med de eftersökta kriterierna.

Figur 35. Svar på uppgift 8.

Självklart kan ett svar erhållas genom att göra en serie inserts i tabellerna "tavling, spelare, golfbag, klubba, caddy och speltillfalle" i respektive ordning.

Uppgift 9: Visa spelarna som deltagit i ett speltillfälle.

```

242 /* Uppgift 9 */
243 • select spelare.namn from spelare join speltillfalle
244   on spelare.personnummer = speltillfalle.personnummer;

```

Figur 36. Frågeoperationer: uppgift 9.

Svar: I denna fråga "joinas" de två tabellerna "spelare" och "speltillfalle" under villkoret att personnummer i de båda tabellerna skall vara likadana. Alla namn hämtas ut på de personer som både finns i spelartabellen och i speltillfalle-tabellen. Det är alltså de namn på de spelare som deltagit vid något av de inlagrade speltillfällena. **Svaret på frågan blir "Bosse, Lancelot, Sune, Beowulf, Benny, Siegfried och Stina".**

namn
Bosse
Lancelot
Sune
Beowulf
Benny
Siegfried
Stina

Uppgift 10: Visa de domare som ansvarat för exakt två tävlingar. (Tips: använd count).

```

246 /* Uppgift 10 */
247 • insert into tavling (namn) value ('Supertävlingen');
248 • insert into ansvara (namn, personnummer) values ('Supertävlingen', '550102-5589');
249
250 • select domare.namn from domare
251   where domare.personnummer = (
252     select ansvara.personnummer
253     from ansvara
254     group by personnummer
255     having count(*) = 2);

```

Figur 37. Svar på uppgift 10.

Figur 38. Frågeoperationer: uppgift 10.

Svar: I denna fråga uthämtas namnet på de domare som ansvarat för mer än en tävling. Initialt görs en insert för att säkerställa att det faktiskt finns ett fall med en domare som dömt två olika tävlingar. **Svaret på frågan blir "Augustus".**

Uppgift 11: Lista alla golfbagar och sortera märket i omvänd ordning (Z först, A sist).

```
257  /* Uppgift 11 */
258  •  select marke from golfbag
259      order by marke DESC;
```

Figur 39. Frågeoperationer: uppgift 11.

Svar: I hämtas märken på alla golfbagar och de sorteras i bokstavsordning med A i slutet av tabellen. Svaret blir "titleist, Nike, Nike, Lacoste, CarlLagerfeldt, AthenaSports, AthenaSports och Adidas". I svaret finns dubletter på de namn som förekommer flera gånger. Om man önskar undvika detta kan kommandot "**distinct**" användas.

Uppgift 12: hämta medelresultatet för alla speltillfällen.

```
261  /* Uppgift 12 */
262  •  select * from speltillfalle;
263  •  select AVG(resultat) from speltillfalle;
```

Figur 40. Frågeoperationer: uppgift 12.

marke
titleist
Nike
Nike
Lacoste
CarlLagerfeldt
AthenaSports
AthenaSports
Adidas

Svar: Här efterfrågas det genomsnittliga värdet för alla resultat som erhållits på alla speltillfällen. Notera att frågan summerar alla värden och delar med alla resultatinstanser. Värden som "diskad" osv drar alltså ned medelresultatet. **Svaret i denna frågeoperation blir 63,7".**

Uppgift 13: Hämta medelresultatet för alla speltillfällen på respektive tävling (Tips: använd Group by).

Figur 41. Svar på uppgift 12.

```
265  /* Uppgift 13 */
266  •  select AVG(resultat) from speltillfalle
267      where resultat group by namn;
```

Figur 42. Frågeoperationer: uppgift 13.

Svar: Frågeoperationen hämtar ut grupperade medelresultat för var och en av de olika tävlingarna som lagrats i tabellen speltillfalle. Funktionen "AVG" tar alltså fram medelvärden genom att summera alla värden och dividera med antalet celler men ignorera NULL-värden. **Svaret på frågan blir "55, 125, 58, 150".**

Uppgift 14: hämta all information om klubborna som har ett namn som börjar på bokstaven "J". (Tips: använd funktionen rlike eller liknande).

```
269  /* Uppgift 14 */
270  •  select * from klubba
271      where namn LIKE 'J%';
```

Figur 43. Frågeoperationer: uppgift 14.

Svar: I fråga 14 görs en sökning efter en specifik bokstav – J. Villkoret är att J skall befinna sig i början på namnet och skrivs därför med "LIKE"-funktionen enligt "J%". Om man i stället velat hitta ett namn där J var den sista bokstaven i ordet hade man flyttat procenttecknet till slutet enligt "%J".

I just detta fall erhöles inget svar eftersom inga namn på klubbor med bokstaven J i början av namnet fanns.

Uppgiften hade kunnat lösas med "RLIKE"-funktionen också varvid annorlunda "wildcards" eller syntaktiska tecken hade använts.

Uppgift 15: Hämta namnet på den spelare som hade bäst resultat i tävlingen "Masters".

```
273  /* Uppgift 15 */
274  • insert into tavling (namn, datum) values ('Masters', '2016-01-23');
275  • insert into spelare (namn, personnummer) values ('HermannBrant', '888888-5555');
276  • insert into speltillfalle (starttid, resultat, namn, personnummer) values ('10:15', '14', 'Masters', '888888-5555');
277
278  • insert into spelare (namn, personnummer) values ('GöstaVarg', '123456-4444');
279  • insert into speltillfalle (starttid, resultat, namn, personnummer) values ('11:15', '11', 'Masters', '123456-4444');
280
281  • select namn from spelare
282  where spelare.personnummer =
283  (select speltillfalle.personnummer from speltillfalle
284   where resultat =
285   (select MAX(resultat) from speltillfalle
286    where namn = 'Masters'));
```

Figur 44. Frågeoperationer: uppgift 15.

Svar: Först görs en serie inserts för att säkerställa att det faktiskt finns en spelare som haft bäst resultat i den efterfrågade tävlingen. Sedan konstrueras en nästlad fråga med funktionen "MAX()" som tar fram det högsta värdet i en kolumn av data. **Svaret på frågan blir spelarnamnet "HermannBrant".**

Uppgift 16: Hämta namnet på de spelare som inte slutförde sitt speltillfälle.

```
288  /* Uppgift 16 */
289  • select namn from spelare
290  where spelare.personnummer = ANY
291  (select speltillfalle.personnummer from speltillfalle
292   where resultat = 'Diskad');
```

Figur 45. Frågeoperationer: uppgift 16.

Svar: Här eftersöks namnet på de spelare som har resultatet "Diskad" i tabellen "speltillfälle". För att uppnå lösningen används kommandot "ANY" som i detta fall ungefär "selektera namnet i spelartabellen där spelare.personnummer är lika med NÅGOT personnummer i speltillfälle som har resultatet DISKAD.

I denna uppgift är det lätt att få felkod 1242 eftersom speltillfalle-tabellen returnerar mer än ett värde då två olika spelare fick resultatet diskad. Genom att lägga till ANY-operatören hämtas namn på alla de spelare som fick resultatet diskad (Oracle Corporation, 2017).

Uppgift 17: Lista de speltillfällen som pågick under golfveckan 10/7 – 17/7. (Tips: använd funktioner som hanterar datum).

```
295  /* Uppgift 17 */
296  • select speltillfalle.namn from speltillfalle
297  where speltillfalle.namn = any (
298  select tavling.namn from tavling
299  where (datum between '2016-07-10' and '2016-07-17'));
```

Figur 46. Frågeoperationer: uppgift 17.

Svar: Här används operatören "between" som används enligt "kolumnnamn BETWEEN värde1 AND värde2" (W3 Schools, 2017). I uppgiften hämtas namnet på de speltillfällen som befinner sig mellan den 10 juli och den 17 juli 2016. En förutsättning för att utföra

frågeoperationen framgångsrikt är att initialt lagra datum med datatypen "date" och inte "varchar" eller dylikt. **Svaret på frågan blir Sigges_Sommargolf; Sigges_Sommargolf; Sigges_Sommargolf; Sigges_Sommargolf; Vikingatävlingen.**

Här visas återigen dubletter på de tävlingar som hade fler än ett speltillfälle. Om man önskar erhålla unika värden kan kommandot "distinct" användas. Vidare kan svaret grupperas och funktionen "count()" kan användas för att få svaret listat som grupper med antalet förekomster i en deltabell.

Uppgift 18: Öka lönen för alla domare som tjänar mellan 10000 och 12000 med 3 %.

```
302  /* Uppgift 18 */
303  • SET SQL_SAFE_UPDATES = 0;
304
305  • update domare
306    set lon = lon * 1.03
307    where lon >= 10000 and lon <= 12000;
```

Figur 47. Frågeoperationer: uppgift 18.

Svar: Här uppstår en ny utmaning som är inbyggd i utvecklingsmiljön. MySQL Workbench är inställt på "säkert läge" i default och det leder till att radering av hela tabeller eller uppdateringar av hela kolumner leder till att kommandot inte genomförs. För att lösa situationen används kommandot "SET SQL_SAFE_UPDATES = 0" som stänger av säkert läge (Zala, 2011).

Resterande kod uppdaterar kolumnen "lon" eller lön i tabellen domare genom att öka alla löner som befinner sig i intervallet 10 tkr till 12 tkr med 3 %.

Om man önskar gå särdeles försiktigt framåt kan här vara ett bra läge att använda transaktioner och rollback/commit för att minimera risken att felaktigheter uppstår.

Avslutningsvis kan man återgå till säkert läge genom att skriva "SET SQL_SAFE_UPDATES = 1;".

Uppgift 19: Ta bort caddyn som heter Jeppe och som alltid ger tipset "Sluta stirra".

```
309  /* Uppgift 19 */
310  • insert into spelare (namn, personnummer) values ('Jesper', '440101-8989');
311  • insert into golfbag (marke, typ, personnummer) values ('Decker', 'Snygg', '440101-8989');
312  • insert into caddy (personnummer, favorit_tips, namn, marke) values ('990123-4214', 'Sluta stirra', 'OlausP', 'Decker');
313  • insert into klubba (namn, kommentar, personnummer, marke) values ('Driver', 'Svag', '440101-8989', 'Decker');
314
315  • delete from caddy
316    where favorit_tips = 'Sluta stirra';
```

Figur 48. Frågeoperationer: uppgift 19.

Svar: Här görs först en serie extra inserts för att få in en caddy som använder sig av favorittipset "sluta stirra". Sedan raderas hela tupler från tabellen caddy där favorittipset är lika med "Sluta stirra". Säkert läge behöver inte avaktiveras eftersom det är en specifik rad som tas bort och inte en hel kolumn eller hel tabell.

Uppgift 20: Ta bort golfbagen av märket "Titleist" som tillhör spelaren med personnummer 560123-6666. (Om det inte fungerar, beskriv varför).

```
318  /* Uppgift 20 */
319  • delete from golfbag
320    where marke = 'titleist' and personnummer = '560123-6666';
```

Figur 49. Frågeoperationer: uppgift 20.

Svar: Vid försök att radera tupler med market "titleist" uppstår ett fel eftersom jag modellerat 1:1-relationen mellan golfbag och caddy genom att placera golfbagens märke i tabellen caddy som en referensnyckel samt marke som en del av den kandidatnyckel som används i tabellen klubba.

Kolumnen marke i tabellen caddy har flera barn länkade till sig som är beroende av värdet. För att kunna radera marke = "titleist" i tabellen golfbag måste först "marke = titleist" raderas från tabellerna caddy och klubba (Payam, 2014). Alltså måste man först ta bort de tupler som använder "titleist" som nyckel i andrahand innan man kan ta bort den primära källan.

4 Svar på frågor från Skultorps Golfklubb

Förklara skillnaden mellan datatyperna "CHAR" och "VARCHAR"

En CHAR är en datatyp som sparar stringvärden av förutbestämd längd, data lagras alltså enligt statisk minnesallokering d.v.s. minnet för inlagringar allokeras när programmet startar och storleken bestäms då programmet eller tabellen skapas. Det maximala antal datatecken som en CHAR kan innehålla är 255 karaktärer.

En VARCHAR däremot använder dynamisk minnesallokering vilket innebär att programmeraren bestämmer potentiell maxstorlek och livstid för minnesallokeringen. Minneslagringen i VARCHAR är långsammare än i CHAR men kan ta fler tecken efter MYSQL 5.0.3+ där hela 65, 535 karaktärer utgör maxinsättningen.

CHAR är alltså bäst att använda sig av då man på förhand vet storleken på insättningarna i databasen – exempelvis ett personnummer eller ett registreringsnummer som alltid har lika många tecken. VARCHAR lämpar sig bättre för tabeller där inlagringen kan variera – exempelvis i ett kommentatorsfält eller en kolumn för användarnamn.

Konsekvens av att sätta en längre "CHAR" än vad som behövs.

Konsekvensen av att sätta en längre CHAR än vad som senare används är att databashanteraren kommer ta i anspråk det utrymme vi bestämt CHAR variabeln till att bli oavsett hur lite vi försöker spara i den. Så t.ex. kanske vi initierar en CHAR variabel till att vara 10 tecken, men vi kanske bara registrerar 7 tecken i den. Då kommer utrymmet för tre tecken tas i anspråk i onödan och det är det som är fördelen med en VARCHAR som i ett sådant fall dynamiskt hade använt ett mindre lagringsutrymme.

Förklara varför NULL-värden är något som i största möjliga mån bör undvikas med avseende på relationsdatabaser? Beskriv hur en relationsdatamodell kan designas för att enkelt undvika uppenbara potentiella NULL-värden.

Ett NULL-värde i en cell innebär att cellen saknar ett värde – den är tom. Det är inte samma sak som noll eftersom noll är ett värde medan NULL är avsaknaden av något värde över huvud taget.

NULL-värden kan påverka frågeoperationer eftersom exempelvis en AVG()-funktion summerar värdena i alla celler i en kolumn och sedan dividerar med antalet celler. NULL-värden kommer att sänka medelvärdet eftersom det inte finns något värde att summera med men väl en extra cell att dividera med. På detta vis försämrar de databasadministratörens möjligheter att hämta ut information ur det data som finns lagrad. Liknande problem kan uppstå vid användning av COUNT.

Förutom de otydligheter som kan uppstå vid användning av aggregatfunktioner bör användningen av utrymme tas i beaktande. Ett NULL-värde kan inte lagras utan att ta upp plats i databasen. Därför är det bättre att på förhand planera en databas så att man slipper stora mängder NULL-värden. Det kan göras genom att exempelvis

välja bort attribut i ER-modellen som kan komma att innehålla mycket NULL-värden och som kanske inte är särskilt viktiga att lagra. Genom att skapa en bra modelleringsgrund kan man bidra till att minska förekomsten av celler som innehåller mycket NULL.

5 Svar på frågor rörande XML

Beskriv varför HTML inte är lämpligt för att strukturera data från en databas. Beskriv också skillnaderna mellan HTML och XML.

HTML är ett så kallat "markup"-språk som använder en uppsättning standardiserade "taggar" för att märka upp text. Taggarna kommunicerar till webbläsaren hur informationen skall visas på skärmen. HTML är anpassat för ostrukturerade data d.v.s. data som existerar i formen av löpande text. Vidare finns en begränsad uppsättning generella alternativ som kan användas som taggar.

XML däremot är ett annat "markup"-språk specialiserat på att organisera data med taggar som beskriver betydelsen av informationen i stället för – som i HTML – strukturen på webbsidan. De taggar som används i XML kan definieras på egen hand av programmeraren och därmed vara specialanpassad till den specifika datauppsättning som de skall beskriva.

Beskriv varför trasiga XML-filer existerar (där start och sluttaggar inte matchar varandra) och hur detta kan undvikas.

Trasiga XML-filer kan uppstå genom att felaktig "markup" gjorts. Starttaggar komplimenteras inte med sluttaggar. För att minimera risken för att fel uppstår bör dokumentet alltid valideras av en extern validerare. Den stora utmaningen med XML är att "markup"-koden inte omfattas av likadana integritetskontroller som beräknande programmeringsspråk som C++ eller PHP vilka kräver korrekt syntax och semantik för att över huvud taget kunna kompileras eller inte ge felmeddelanden.

Skapa en XML-fil som följer nedstående DTD:

```
<!ELEMENT tävling (domare*, speltillfällen*)>
<!ELEMENT domare (namn, lön)>
<!ELEMENT namn (#PCDATA)>
<!ELEMENT lön (#PCDATA)>
<!ELEMENT speltillfälle (resultat)>
<!ELEMENT resultat (#PCDATA)>
<!ATTLIST tävling namn ID #REQUIRED>
<!ATTLIST domare pnr ID #REQUIRED>
<!ATTLIST speltillfälle starttid ID #REQUIRED>
```


I den här uppgiften skall en XML-fil skapas i korrekt syntax enligt följande regler. Genom att utelämna användningen av en kardinalitetsymbol av typen (?, *, +) åsyftas en tvingande användning av elementet där det alltid är exakt ett.

Tabell 6. Sammanfattning av DTD-syntax.

Syntax	Förklaring
!ELEMENT	Beskriver ett element som skrivs enligt t.ex. <Car> </Car>. Element kan komplimenteras med attribut och vara nästlade.
!ATTLIST	Beskriver förekomsten av ett attribut, vanligen en nyckel, som ger extra information till ett element enligt <Car regn=123> </Car>
?	Betyder att ett element förekommer 0 eller 1 gånger.
*	Ett element som förekommer 0 eller flera gånger.
+	Ett element som kan existera 1 eller flera gånger.
PCDATA	Betyder "Parsed Character Data" och är den information som parse-programmet läser in.

```
<tävling namn=Sigges Sommargolf>
    <domare pnr=19890123-4214>
        <namn> Olof Almqvist</namn>
        <lön> 55 000 </lön>
    </domare>
    <domare pnr=19850410-5565>
        <namn> Sven Svensson</namn>
        <lön> 30 000 </lön>
    </domare>
    <speltillfällen starttid=13:00>
        <resultat> Första Vinst </resultat>
    </speltillfällen>
    <speltillfällen starttid=14:00>
        <resultat> Tionde plats </resultat>
    </speltillfällen>
    <speltillfällen starttid=15:00>
        <resultat> Sist i tävlingen </resultat>
    </speltillfällen>
</tävling>
```

Skapa en DTD som gör det möjligt att inkludera information om spelare, golfbag, klubbor och caddy. Tanken är att föreningen vill använda denna information för att visa exempelvis spelarinformation på hemsidan.

```
<!ELEMENT spelare (namn?, golfbag*)  
<!ELEMENT namn (#PCDATA)>  
<!ATTLIST spelare pnr ID #REQUIRED>  
  
<!ELEMENT golfbag (typ?, klubbor*)  
<!ELEMENT typ (#PCDATA)>  
<!ATTLIST golfbag märke ID #REQUIRED>  
  
<!ELEMENT klubbor (kommentar?)>  
<!ELEMENT kommentar (#PCDATA)>  
<!ATTLIST klubbor namn ID #REQUIRED>  
  
<!ELEMENT caddy (namn?, favorit_tips?)>  
<!ELEMENT namn (#PCDATA)>  
<!ELEMENT favorit_tips (#PCDATA)>  
<!ATTLIST caddy pnr ID #REQUIRED>  
<!ATTLIST caddy märke IDREF #IMPLIED>
```

6 Egna antaganden och krav

I arbetet har vi försökt hålla oss så nära som möjligt de modeller och beskrivningar som ni har tillhandahållit oss. Vi ser dock stora möjligheter att vidareutveckla systemet mot en fullt fungerande mjukvarulösning där en stor del av er verksamhets marknadsföring och strategiska samt taktiska planering kan ske via ett webbinterface. Allteftersom tiden fortlöper och mer data samlas in angående den operativa driften kan mer förfinade analysverktyg ta fram och på så sätt minimera utgifter och maximera intäkterna.

Vi föreslår en systemarkitektur där databasen färdigställs i MySQL Workbench och kompletteras med en hemsida i HTML/CSS/Javascript för front-end och kopplas ihop med databasen i något av "serversida"-språken PHP eller Django-Python. För dataanalys föreslår vi språket R eller enklare diagram och statistik i Microsoft Excel.

7 Eventuellt förlorad semantik

I den ER-modell som tagits fram av tidigare konsultfirmor finns det beskrivningar som kan uppfattas tvetydigt vid översättning till SQL.

7.1 Relationen Domare och Tävling

I organisationsbeskrivningen står det att *"varje tävling har en domare som ansvarar för alla regler"*. Detta kan tolkas som att varje tävling har exakt en domare – och inte noll eller flera vilket ER-diagrammet påvisar.

7.2 Namnförkortning på Klubba

Vidare står det i beskrivningen av arbetssystemet att *"varje klubba har även en namnförkortning som identifierar den, ex. J3 som betyder järn 3 osv."*. I de initiala datainsättningar som föreslagits beskrivs dock klubborna med namn som exempelvis "Driver" – d.v.s. inte en namnförkortning. Det är otydligt huruvida uppdragsgivaren önskar ha två separata namn – ett fullständigt och ett förkortat – eller om de skall ses som samma.

7.3 Data som kan saknas i databasen

I verksamhetsbeskrivningen står att tävlingen "Sigges Sommargolf" som anordnas årligen har en hög ranking på PGA-Touren. Ett övervägande vi vill skicka med till er är huruvida SG önskar lagra information om historisk ranking i denna tävling samt lagra information om andra tävlingars resultat i touren.

Vidare beskrivs behovet av funktionärer som hjälper till att arrangera tävlingen. Kanske kan det finnas ett värde i att lagra data om hur många funktionärer som deltagit i de olika tävlingarna varje år samt data som reflekterar deltagarnas nöjdhet med arrangemanget. På detta vis kan vi ha möjlighet att ta fram analyser av det optimala antalet funktionärer relativt till olika stora mängder deltagare och besökare. På detta sätt kan verksamheten effektiviseras.

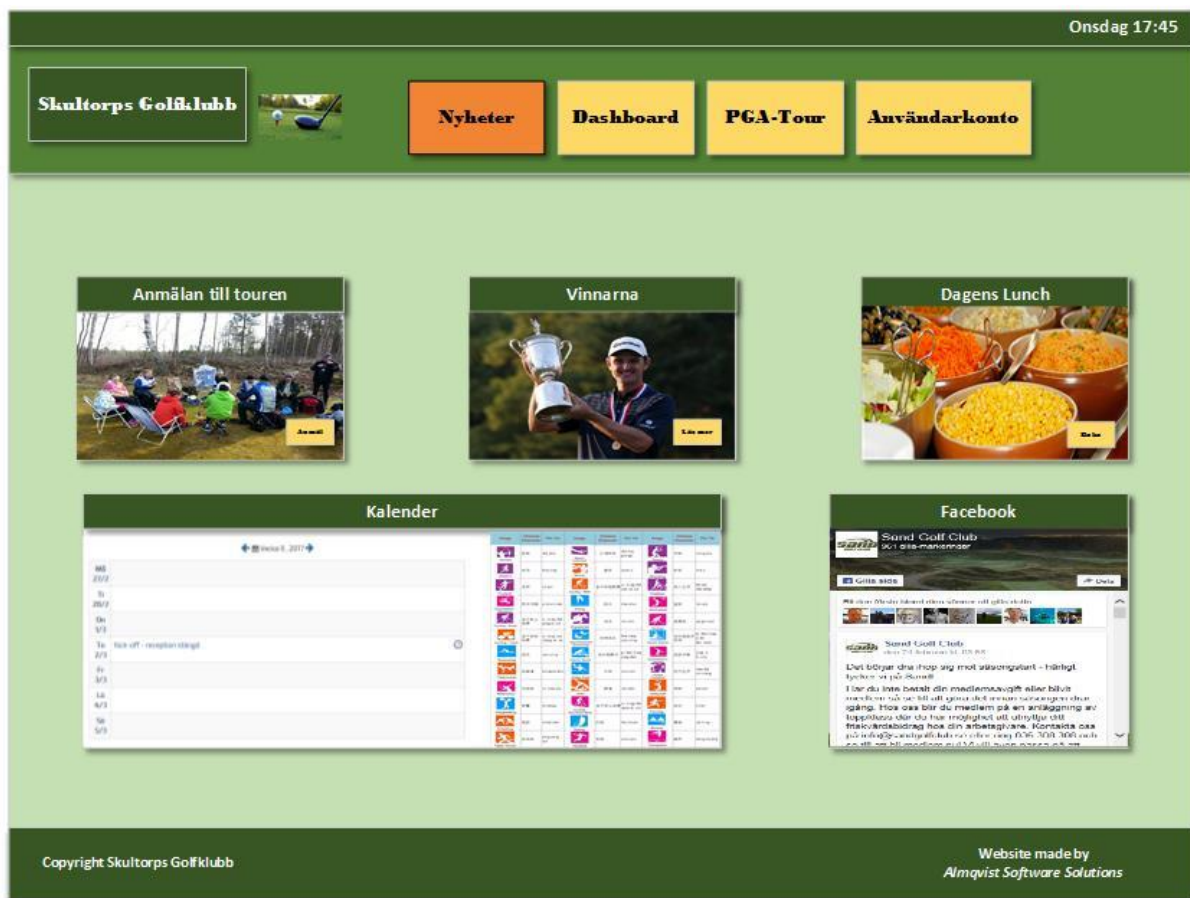
8 Förbättringsförslag

Vi hoppas att ni är nöjda med den databasprototyp som tagits fram och ser värdet i att vidareutveckla systemet till en helhetslösning med ett webbinterface. Vi tänker oss en relativt enkel internetsida där medlemmar i SG samt funktionärer och annan personal med anknytning till klubben kan logga in och ta del av utökad funktionalitet.

En naturlig del av webbsidan kan vara en nyhetssida där ledningen kan presentera information om vad som pågår. Vidare föreslår vi att en intern nyhetsström finns för att snabbt kunna dela information mellan dem som deltar i en tävling som arrangeras.

Avslutningsvis erbjuder vi att ta fram en dashboard som tydligt och koncist sammanfattar nyckeltal och annan värdefull information som bidrar till att arbetssystemet fungerar så effektivt som möjligt.

Exempel på hur startsidan kan se ut. På förstasidan finns direktkopplingar till andra sidor. Exempelvis finns ett API som kopplar samman klubbens facebooksida med hemsidan. Vidare finns direktuppdateringar i kalendern baseras på de tävlingar och nyheter som läggs in i nyhetsflödet.



Figur 50. Förslag på startsida för hemsidan.

Avslutningsvis kan vi nämna en prototyp för hur en dashboard skulle kunna se ut på hemsida. Självfallet är det här en design som vi vill ta fram mer i detalj tillsammans med er i en kravinsamlingsprocess, men det här är en initial ritning för hur sektionen skulle fungera.



Figur 51. Prototyp för dashboard-delen på webbsidan.

7 Bilagor

7.1 Källkod för konstruktion av tabellerna

```
drop database a15o1oal;
create database a15o1oal;
use a15o1oal;

create table tavling(
    namn varchar(20),
    datum date,
    primary key(namn)
)engine=innodb;

create table domare(
    namn varchar(20),
    lon integer(5),
    personnummer char(11),
    primary key(personnummer)
)engine=innodb;

create table ansvara(
    namn varchar(20),
    personnummer char(11),
    primary key(personnummer, namn),
    foreign key(personnummer) references domare(personnummer),
    foreign key(namn) references tavling(namn)
)engine=innodb;

create table spelare(
    personnummer char(11),
    namn varchar(20),
    primary key(personnummer)
)engine=innodb;

create table speltillfalle(
    starttid varchar(10),
    resultat varchar(10),
    namn varchar(20),
    personnummer char(11),
    primary key(starttid, namn, personnummer),
    foreign key (namn) references tavling(namn),
```

```
foreign key(personnummer) references spelare(personnummer)
)engine=innodb;
```

```
create table boll(
    signatur varchar(20),
    nummer integer,
    marke varchar(15),
    personnummer char(11),
    primary key(signatur),
    foreign key(personnummer) references spelare(personnummer)
)engine=innodb;
```

```
create table golfbag(
    marke varchar(15),
    personnummer char(11),
    typ varchar(12),
    primary key(marke, personnummer),
    foreign key(personnummer) references spelare(personnummer)
)engine=innodb;
```

```
create table caddy(
    personnummer char(11),
    favorit_tips varchar(30),
    namn varchar(20),
    marke varchar(15),
    primary key(personnummer),
    foreign key(marke) references golfbag(marke)
)engine=innodb;
```

```
create table klubba(
    namn varchar(20),
    kommentar varchar(40),
    marke varchar(15),
    personnummer char(11),
    primary key(namn, marke, personnummer),
    foreign key(marke) references golfbag(marke),
    foreign key(personnummer) references golfbag(personnummer)
)engine=innodb;
```


7.2 Källkod för transaktionsinmatning

```
/* Uppgift 1 */
insert into tavling (namn, datum) values ('Sigges_Sommargolf','2016-07-10');

/* Uppgift 2 */
insert into domare (namn, lon, personnummer) values ('Simon', '12000', '790129-4444');
insert into domare (namn, lon, personnummer) values ('Sven', '12000', '810912-5555');
insert into domare (namn, lon, personnummer) values ('Eva', '12000', '790612-1212');

insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '790129-4444');
insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '810912-5555');
insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '790612-1212');

/* Uppgift 3 */
insert into spelare (personnummer, namn) values ('560123-6666', 'Bosse');
insert into speltillfalle (starttid, namn, personnummer, resultat) values ('10:25',
'Sigges_Sommargolf', '560123-6666', '44');
insert into golfbag (marke, personnummer, typ) values ('titleist', '560123-6666', 'Helgbag');
insert into klubba (namn, personnummer, marke, kommentar) values ('Swinger', '560123-6666',
'titleist', 'Klubba att lita på');
insert into caddy (marke, personnummer, namn, favorit_tips) values ('titleist', '560123-6666',
'Johannes_Brost', 'Tarecool');

/* Uppgift 4 */
insert into spelare (namn, personnummer) values ('Gudinnan Nike', '730909-1111');
insert into golfbag (marke, personnummer, typ) values ('Nike', '730909-1111', 'Tour');

/* Uppgift 5 */
insert into klubba (namn, kommentar, personnummer, marke) values ('Driver', 'Spikrak och
kort', '730909-1111', 'Nike');

/* Uppgift 6 */
insert into spelare (namn, personnummer) values ('Sune', '670808-2222');
insert into golfbag (marke, typ, personnummer) values ('Adidas', 'Svart', '670808-2222');
insert into caddy (namn, favorit_tips, personnummer, marke) values ('Anna', 'Knyt skorna
hårdare', '890123-4214', 'Adidas');

/* Uppgift 7 */
insert into speltillfalle (starttid, resultat, namn, personnummer) values ('13:10', '72',
'Sigges_Sommargolf', '670808-2222');
insert into boll (signatur, marke, personnummer, nummer) values ('Hjärta', 'Titleist',
'670808-2222', '583');

/* Uppgift 8 */
```

```

insert into spelare (namn, personnummer) values ('Olle', '900101-8888');
insert into golfbag (marke, personnummer, typ) values ('Lacoste', '900101-8888', 'Getskinn');
insert into caddy (namn, personnummer, favorit_tips, marke) values ('Petra', '910202-7777', 'Svinga lungt', 'Lacoste');
insert into klubba (namn, kommentar, personnummer, marke) values ('Driver', 'Längst och snedast på touren', '900101-8888', 'Lacoste');

```

/* Uppgift 9 */

```

insert into spelare (namn, personnummer) values ('Benny', '790101-4343');
insert into domare (namn, personnummer, lon) values ('Petrus', '770202-3333', '18990');
insert into ansvara (namn, personnummer) values ('Sigges_Sommargolf', '770202-3333');
insert into boll (signatur, marke, personnummer, nummer) values ('Treprickar', 'Zeitgeist', '790101-4343', '985');
insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Sigges_Sommargolf', '790101-4343', 'Diskad', '08:00');

```

/* Uppgift 10 */

```

insert into spelare (namn, personnummer) values ('Stina', '950101-4141');
insert into golfbag (marke, personnummer, typ) values ('Nike', '950101-4141', 'Ormskinn');
insert into klubba (kommentar, marke, personnummer, namn) values ('Säkra Bettan', 'Nike', '950101-4141', 'Putt');
insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Sigges_Sommargolf', '950101-4141', 'Diskad', '12:05');
insert into domare (namn, personnummer, lon) values ('Jens', '850102-7878', '18000');
insert into boll (signatur, marke, personnummer, nummer) values ('Extralångt', 'Nike', '950101-4141', '156');

```

/* Extrainserts */

```

insert into spelare (namn, personnummer) values ('Siegfried', '880101-8888');
insert into spelare (namn, personnummer) values ('Beowulf', '770101-7777');
insert into spelare (namn, personnummer) values ('Lancelot', '660101-6666');

insert into tavling (namn, datum) values ('Vikingatävlingen', '2016-07-15');
insert into tavling (namn, datum) values ('Riddarspelen', '2014-10-06');
insert into tavling (namn, datum) values ('Excaliburtävlingen', '2012-09-10');

insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Vikingatävlingen', '880101-8888', '150', '13:00');
insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Riddarspelen', '770101-7777', '125', '15:00');
insert into speltillfalle (namn, personnummer, resultat, starttid) values ('Excaliburtävlingen', '660101-6666', '55', '16:00');

insert into domare (namn, personnummer, lon) values ('Augustus', '550102-5589', '10500');
insert into domare (namn, personnummer, lon) values ('Julius', '400230-7810', '73500');
insert into domare (namn, personnummer, lon) values ('Nero', '731224-6532', '12453');

```

```

insert into ansvara (namn, personnummer) values ('Vikingatävlingen', '550102-5589');
insert into ansvara (namn, personnummer) values ('Riddarspelen', '400230-7810');
insert into ansvara (namn, personnummer) values ('Excaliburtävlingen', '731224-6532');

insert into boll (signatur, marke, personnummer, nummer) values ('Vinnarbollen',
'CarlLagerfeldt', '880101-8888', '123456789');
insert into boll (signatur, marke, personnummer, nummer) values ('Apollobollen',
'AthenaSports', '770101-7777', '987654321');
insert into boll (signatur, marke, personnummer, nummer) values ('Zeusbollen', 'AthenaSports',
'660101-6666', '13579');

insert into golfbag (marke, personnummer, typ) values ('CarlLagerfeldt', '880101-8888',
'Brun');
insert into golfbag (marke, personnummer, typ) values ('AthenaSports', '770101-7777', 'Guld');
insert into golfbag (marke, personnummer, typ) values ('AthenaSports', '660101-6666',
'Silver');

insert into klubba (namn, kommentar, marke, personnummer) values ('Putt', 'Vann tävlingen',
'CarlLagerfeldt', '880101-8888');
insert into klubba (namn, kommentar, marke, personnummer) values ('Slag', 'Hole in one',
'AthenaSports', '770101-7777');
insert into klubba (namn, kommentar, marke, personnummer) values ('Slag', 'Ok',
'AthenaSports', '660101-6666');

insert into caddy (personnummer, favorit_tips, marke, namn) values ('550123-4214', 'NEVER Give
UP', 'CarlLagerfeldt', 'Olaus');
insert into caddy (personnummer, favorit_tips, marke, namn) values ('400901-5598', 'Se saken
positivt', 'AthenaSports', 'Linn');
insert into caddy (personnummer, favorit_tips, marke, namn) values ('800420-5665', 'Kör
bara!', 'AthenaSports', 'Tony');

```

7.3 Källkod för frågeoperationer

```

/* Uppgift 1
select namn from domare
where personnummer = '790129-4444'; */

/* Uppgift 2
select signatur from boll
where personnummer = '560123-6666'; */

/* Uppgift 3
select typ from golfbag
where personnummer = '560123-6666'; */

```

```

/* Uppgift 4
select namn from spelare
where spelare.personnummer = (
    select boll.personnummer
    from boll
    where marke = 'Titleist'); */

```

```

/* Uppgift 5
select resultat from speltillfalle
where speltillfalle.personnummer =
    (select spelare.personnummer from spelare
    where spelare.personnummer =
        (select boll.personnummer from boll
        where marke='Nike')); */

```

```

/* Uppgift 6
insert into spelare (namn, personnummer) values ('sune', '888888-7777');

```

```

CREATE VIEW dubblettnamn AS
SELECT namn, count(*)
FROM spelare
group by namn
having COUNT(*) > 1;

```

```

create view alla_namn AS
select namn, personnummer from spelare;

select personnummer, namn as 'Dubblettnamn' from alla_namn
where alla_namn.namn = (select dubblettnamn.namn from dubblettnamn); */

```

```

/* Uppgift 7
select domare.personnummer from domare
where domare.personnummer not in (
    select ansvara.personnummer
    from ansvara) */

```

```

/* Uppgift 8 Inget svar ges eftersom det inte finns någon spelare med efterfrågat
personnummer som fick 72 poäng i tävlingen Ryder Cup.

```

```

select favorit_tips from caddy
where caddy.marke = (
    select golfbag.marke from golfbag

```

```

where golfbag.personnummer = (
    select spelare.personnummer from spelare
    where spelare.personnummer = (
        select speltillfalle.personnummer from speltillfalle
        where speltillfalle.resultat = 72 and speltillfalle.namn = 'Ryder Cup' and
        speltillfalle.personnummer = '660808-5555'))); */

```

/* Uppgift 9

```

select spelare.namn from spelare join speltillfalle
    on spelare.personnummer = speltillfalle.personnummer; */

```

/* Uppgift 10

```

insert into tavling (namn) value ('Supertävlingen');
insert into ansvara (namn, personnummer) values ('Supertävlingen', '550102-5589');

```

```

select domare.namn from domare
where domare.personnummer = (
    select ansvara.personnummer
    from ansvara
    group by personnummer
    having count(*) = 2); */

```

/* Uppgift 11

```

select marke from golfbag
order by marke DESC; */

```

/* Uppgift 12

```

select * from speltillfalle;
select AVG(resultat) from speltillfalle; */

```

/* Uppgift 13

```

select AVG(resultat) from speltillfalle
where resultat group by namn; */

```

/* Uppgift 14

```

select * from klubba
where namn LIKE 'J%'; */

```

/* Uppgift 15

```

insert into tavling (namn, datum) values ('Masters', '2016-01-23');
insert into spelare (namn, personnummer) values ('HermannBrant', '888888-5555');
insert into speltillfalle (starttid, resultat, namn, personnummer) values ('10:15', '14',
'Masters', '888888-5555');

```

```

insert into spelare (namn, personnummer) values ('Göstavarg', '123456-4444');
insert into speltillfalle (starttid, resultat, namn, personnummer) values ('11:15', '11',
'Masters', '123456-4444');

select namn from spelare
where spelare.personnummer =
    (select speltillfalle.personnummer from speltillfalle
    where resultat =
        (select MAX(resultat) from speltillfalle
        where namn = 'Masters')); */

/*Uppgift 16
select namn from spelare
where spelare.personnummer = ANY
    (select speltillfalle.personnummer from speltillfalle
    where resultat = 'Diskad'); */

/* Uppgift 17
select speltillfalle.namn from speltillfalle
where speltillfalle.namn = any (
select tavling.namn from tavling
where (datum between '2016-07-10' and '2016-07-17')); */

/* Uppgift 18
SET SQL_SAFE_UPDATES = 0;
update domare
set lon = lon * 1.03
where lon >= 10000 and lon <= 12000; */

/* Uppgift 19
insert into spelare (namn, personnummer) values ('Jesper', '440101-8989');
insert into golfbag (marke, typ, personnummer) values ('Decker', 'Snygg', '440101-8989');
insert into caddy (personnummer, favorit_tips, namn, marke) values ('990123-4214', 'Sluta
stirra', 'OlausP', 'Decker');
insert into klubba (namn, kommentar, personnummer, marke) values ('Driver', 'Svag', '440101-
8989', 'Decker');

delete from caddy
where favorit_tips = 'Sluta stirra'; */

```

```
/* Uppgift 20  
delete from golfbag  
where marke = 'titleist' and personnummer = '560123-6666'; */
```



Referenser

IBM. (u.å.). Advantages of XML.

https://www.ibm.com/support/knowledgecenter/ssw_i5_54/rzamj/rzamjintroadvantages.htm [2017-02-10]

Jajodia, S., Ng, P.A., Springsteel, F.N. (1983). Entity-relationship diagrams which are in BCNF. International Journal of Computer & Information Sciences, 12(4), ss. 269-283.

Kim, L. (2015). 10 Most Popular Programming Languages Today.

<http://www.inc.com/larry-kim/10-most-popular-programming-languages-today.html> [2017-02-12]

Oracle Corporation. (2017). Subquery Errors.

<https://dev.mysql.com/doc/refman/5.7/en/subquery-errors.html> [2017-02-15]

Padron-McCarthy, T., Risch, T. (2015). Databasteknik. Lund: Studentlitteratur AB.

Payam. (2014). ERROR 1451: 1451: Cannot delete or update a parent row: a foreign key constraint fails. Stack Overflow.

<http://stackoverflow.com/questions/26696534/error-1451-1451-cannot-delete-or-update-a-parent-row-a-foreign-key-constraint> [2017-02-15]

Zala, A. (2011). MySQL Safe Update Mode Explained.

<http://www.xpertdeveloper.com/2011/10/mysql-safe-update/> [2017-02-20]

W3 Schools. (2017). SQL BETWEEN operator.

https://www.w3schools.com/sql/sql_between.asp [2017-02-17]