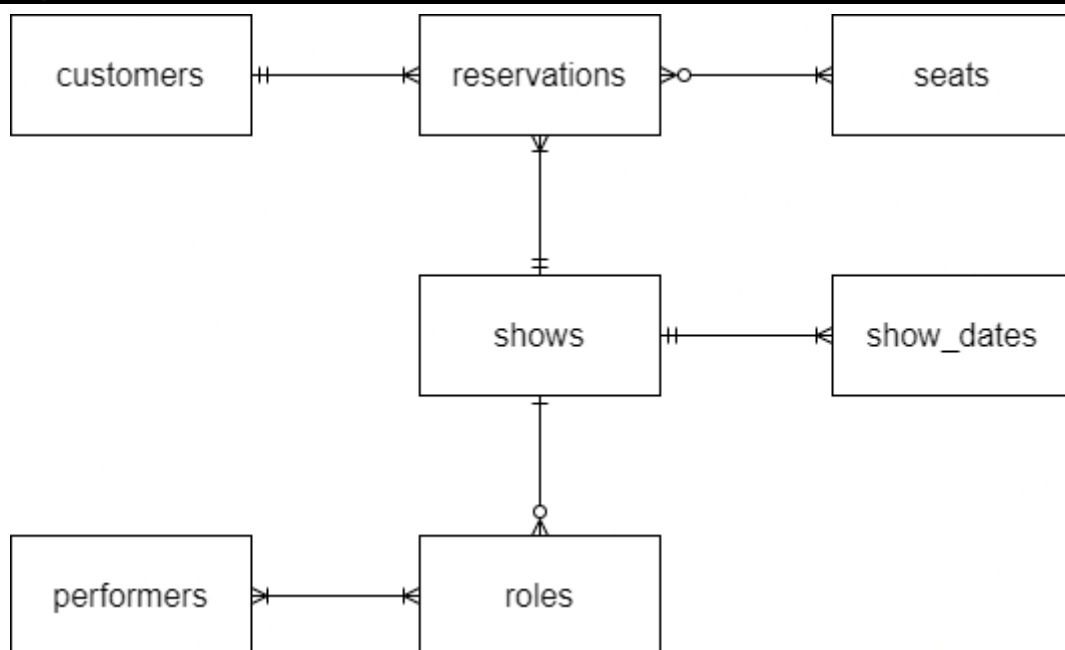




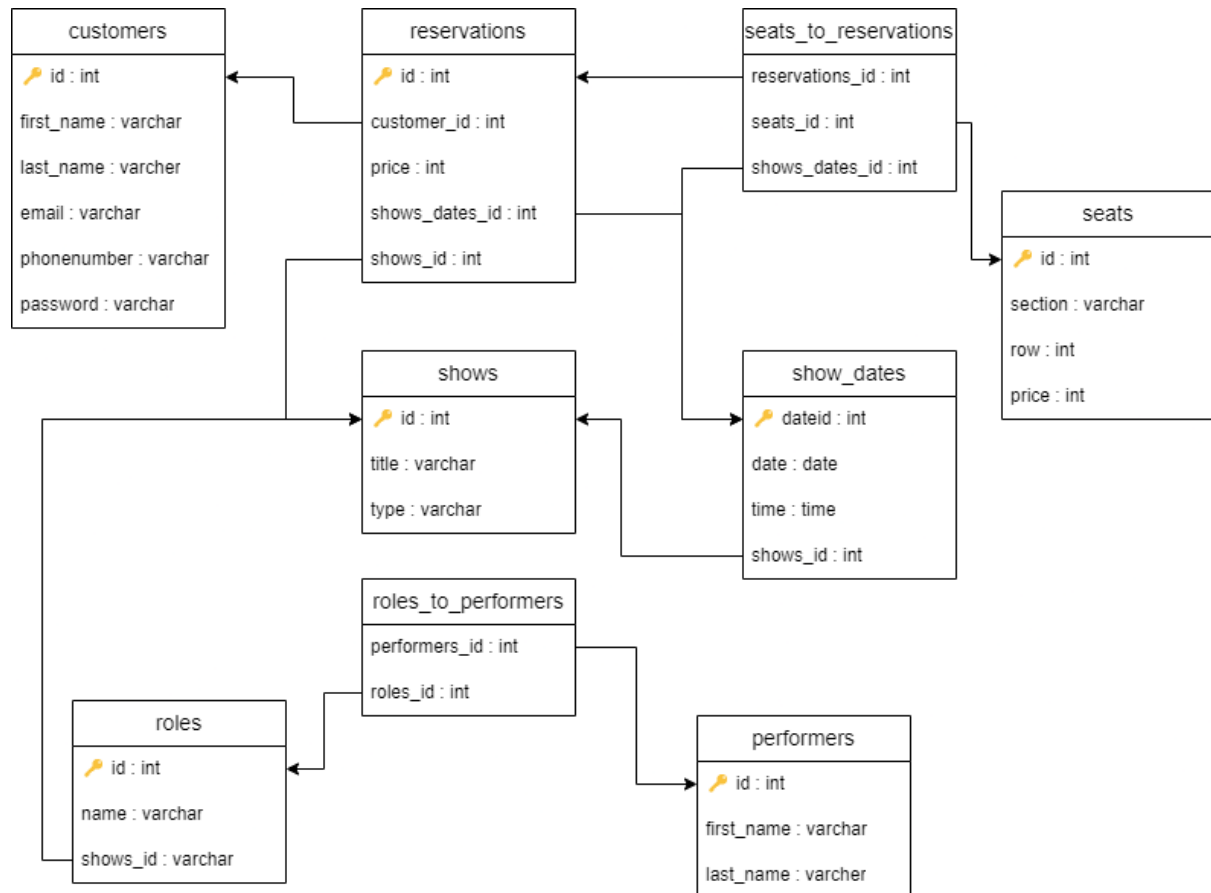
Göteborgs nya kultursatsning är Göteborgs **MINI**opera som kommer hålla föreställningar på en mindre scen med större innovativa satsningar inom musik och teknik. Databasen kommer hantera information från de olika föreställningarna, rollkaraktärer, artister, datumhantering, sittplatser, reservationer och kundinformation.

ER-diagram



Vilka entiteter är nödvändiga för att kunna skapa ett mindre bokningssystem var min första fråga, detta resulterade i tabellerna : customers, reservations, seats, shows, roles och performers.

Kopplingsdiagram



Databasen består av nio tabeller varav två tabeller är kopplingstabeller. För att lättare avgöra huruvida en seat är bokad eller inte har jag även kopplat show_dates till kopplingstabellen seats_to_reservations. Från första början låg datum och tid i tabellen shows men insåg snabbt att det var bättre att tidshanteringen var en egen tabell med unika id:n.

Databasen ska ha kommandon utifrån CRUD - create, read, update & delete. Jag har begränsat arbetet med CRUD att enbart behandla användarens version av programmet. Detta innebär i korthet att datan som finns i databasen som showerna, dess roller, artisterna samt säten och datum och tid inte kan skapas, tas bort eller uppdateras från applikationen.

Svårigheter under arbetets gång

Jag hade svårigheter vid borttagandet av reservationer och hur sätena skulle uppdateras så att de skulle visas som lediga igen. Jag testade med olika SQL-kommandon innan jag kom fram till att jag kunde använda relationerna mellan tabellerna; reservations och seats_to_reservations då den senare har foreign key som pekar mot reservations.id som är tabellens primary key. Vid DELETE eller UPDATE av reservations.id påverkas även "barntabellen"¹ vid användandet av CASCADE vilket fungerade perfekt för min databas och funktionalitet!

Den största utmaningen jag hade under projektets gång var att skriva SQL queries som hämtar ut alla id:n i tabellen seats samt de tillgängliga sätena till en viss show.

```
1 reference
public List<Seat> SelectAvailableSeats(int dateId)
{
    // Gets all seats that are available
    return _sqlconnection.Query<Seat>($"SELECT seats.id
    FROM seats
    WHERE seats.id NOT IN
    (SELECT seats_to_reservations.seats_id FROM seats_to_reservations
    WHERE seats_to_reservations.shows_dates_id = '{dateId}')").ToList();
}
```

En annan utmaning jag hade rent kunskapsmässigt var skillnaden på sparad data i databasen och den tillfälliga datalagringen som är i programmet. Genom att använda INNER JOIN kan jag hämta data från utvalda kolumner från flera olika tabeller och lagra datan i ett klassobjekt. Det vill säga en tabells attribut behöver inte överensstämma med en klass variabler eller properties. Jag väljer vilken data som är aktuellt för programmet att hämta!

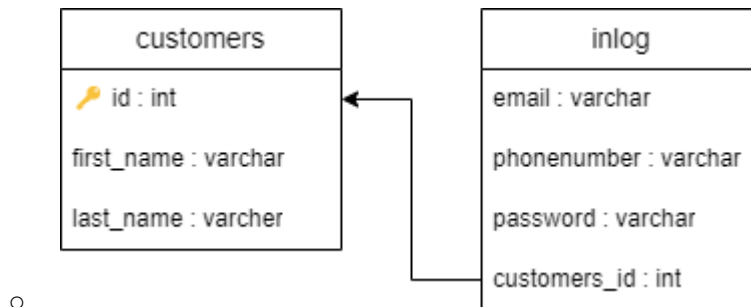
¹ <https://www.javatpoint.com/mysql-on-delete-cascade>

Vidareutveckling

Då programmet är designat utifrån användarens perspektiv, behövs vidareutveckling för en administrativ version av programmet. Sedan är det inte logiskt att informationen om de olika showerna hämtas direkt från databasen exempelvis vid menyhantering som hämtar ut alla shows titlar. Detta var ett val från min sida enbart för att arbeta mer åt databasen, det vill säga att skicka datan sinsemellan databasen och applikationen.

Sammanställning för vidare utvecklingsmöjligheter :

- Skapa en tabell som tillåter flera inloggningsmöjligheter per användare,



- Komma på en design som möjliggör bokning av säten på balkongen (se bilaga) samt olika prisnivåer av säten,
- Lägga till kod i programmet som ser över datum så inte tidigare datum för shower kan bokas samt en funktion som visar kommande shower och när denna informationen ska visas,
- Skapa administrativ version av programmet,
- Skapa en transaktion vid bokningar,
- Se över SQL-kommandon och låta databasen utföra stored procedures samt triggerfunktion när en show är slutsåld och att detta visas i applikationen.

Bilaga

Design över sätena i operan :



PRISER	ORDINARIE	BARN	STUDENT	PENSIONÄR
PARKETT	450	225	338	315
BALKONG R 1	380	190	285	266
BALKONG R 2	350	175	263	245