Projektrapport- Logisk och fysisk modell

Datalagring, IV1351

Mattias Sandberg matsandb@kth.se

2022-11-24

1 Introduktion

Syftet med detta projekt är att designa en applikation och databas för Soundgood Music School. Applikationen kommer att hantera transaktionerna som uthyrningen av instrument medan databasen kommer hantera all information som musikskolan innehar. I detta projekt så kommer jag att jobba nära Viktor Björken för att underlätta och förbättra min inlärning. Projektet kommer att vara indelat i fyra delar vilket är följande (Konceptuell modell, Logisk och fysisk modell, SQL, Programmatisk åtkomst). Astah är det dataprogrammet som jag använder för att konstruera den konceptuella modellen över musikskolan. I Astah så visas alla relationer, enheter och attribut som är väsentliga för musikskolan ska kunna fungera.

2 Litteraturstudier & förberedelser

2.1 Uppgift 1 – Logisk och fysisk modell

Den första saken som jag gjorde för att förbereda mig inför denna uppgift var att kolla på samtliga föreläsningar. Jag tittade dessutom på det inspelade videorna från canvas där Leif Lindbäck systematiskt och tydligt förklarade hur man på ett elegant sätt går från en konceptuell modell till en logisk och fysisk modell. Då jag inte var bekant med en logisk och fysisk modell och Astah så var jag tvungen att söka ytterligare information om hur man bygger dessa modeller i Astah. Denna information hittade jag på Youtube och kursens hemsida på canvas. Ett exempel på detta var att det fanns många inställningar som jag behövde ändra då jag nu konstruerade min modell med ett ER-Diagram i stället för ett Class-Diagram. Hemsidan ”geeksforgeeks” hjälpte även mig att bättre förstå vad nycklarna användes till och skillnaderna mellan det olika nycklarna.

3 Metod

3.1 Uppgift 1 – Logisk och fysisk modell

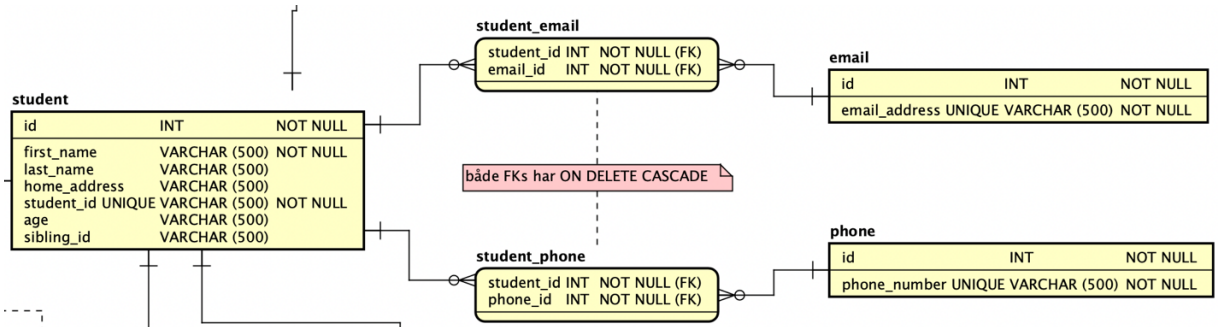
I föregående uppgift så konstruerade jag en konceptuell modell över ”Soundgood Music School” vilket jag använde som grund i denna uppgift genom att jag konverterade den till en logisk och fysisk modell. Den stora skillnaden mellan en konceptuell modell och en logisk och fysisk modell som jag nu har skapat är att den tillhandahåller betydligt mer information till databasen som exempelvis primära nycklar och datatyper för attribut.

Den logiska delen av modellen definierar innehållet och anpassas inte för någon speciell DBSM (database managementsystem) för vilket vi använder PostgreSQL. Den logiska modellen hanterar inte heller fysisk lagring som till exempel ”views”. Anledningen till att vår modell är en blandning av en logisk och fysisk modell är eftersom den fysiska modellen är anpassad för ett specifikt DBMS.

Det första steget var att skapa alla enheter från mitt föregående program vars attribut hade en kardinalitet på antingen 0..1 eller 1..1. När detta var klart så skapade jag nya enheter för det attribut som hade en högre kardinalitet som tillexempel ”Phone och Email”. När jag skapade dessa så följde jag en namnkonvention med små bokstäver och understreck mellan ord. Nästa steg var att specificera datatyp för varje attribut. Jag valde ”VarChar” med en gräns på 500 karaktärer och för det attribut med tidbaserad information valdes ”TimeStamp”. Nästa steg var att införa kolumnrestriktioner där det infördes ”UNIQUE och NOT NULL” beroende på om attributen är unik och eller om den får lämnas tom.

Därefter så tilldelade jag primära nycklar till alla starka enheter, dvs det enheter som enligt mig hade en elementär betydelse för skolans struktur. Det enheter som dessa relaterade till fick främmande nycklar (FK) som primära nycklar (PK) om det ej ansågs triviala utan den ordinarie primära nyckeln. Om den ansågs väsentlig utan relationen infördes en surrogat nyckel som primär nyckel.

Till sist så skapades det ”cross-reference table” för ”many-to-many” relationer vilket i mitt fall är från student till email och telefonnummer. Detta visualiseras enligt bilden nedan vilket är ett urklipp från Astah när jag grovt konstruerade den logiska och fysiska modellen:



När modellen i Astah var klar skrevs det två olika sql-filer. Det ena programmet skulle konstruera databasen och den andra skulle föra in data till databasen. Programmet för att skapa en databas från Astah skrev jag för hand.

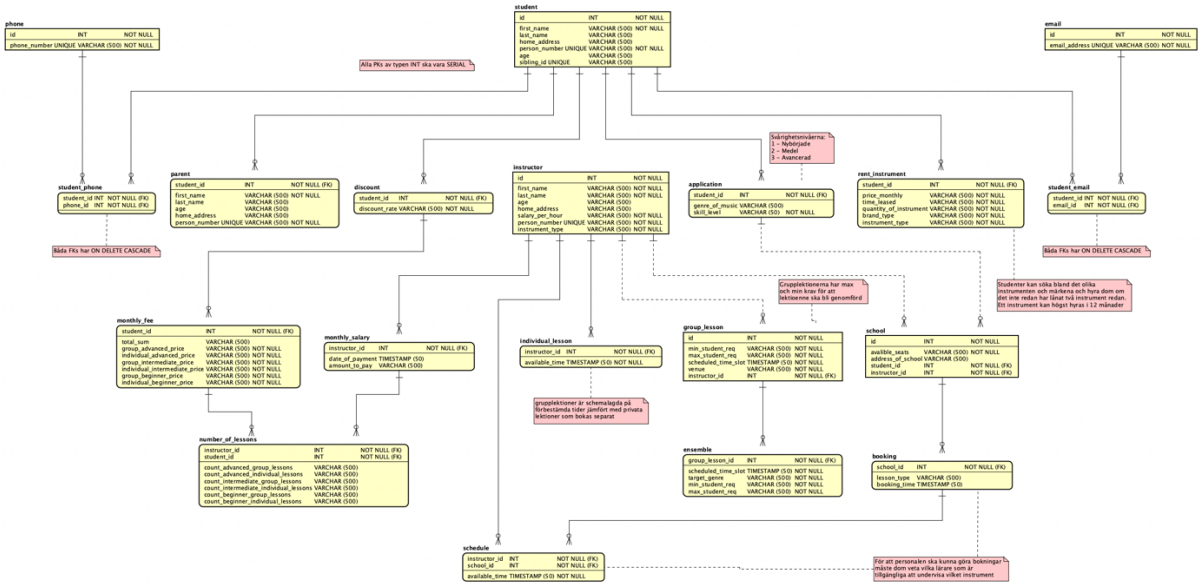
Dock använde jag en slumpmässig generator från hemsidan ” https://www.bestrandoms.com/random-address-in-se” och ” https://randommer.io/random-email-address” när jag skrev programmet för att infoga data till databasen då det sparade mig tid att slippa behöva komma på egna adresser och email adresser. Resten av datan var manuellt införd med viss pression som tex att ”brand_type” attributet för ”rental_instrument” enheten faktisk är ett märke som säljer instrument som studenterna kan hyra. Detta visualiseras genom den infogade bilden nedan:

```
INSERT INTO rent_instrument (price_monthly,time_leased,quantity_of_instrument,brand_type,instrument_type,student_id) VALUES
('200', '6 months','1','Gibson','Guitar','0'),
('400', '8 months','2','Yamaha','Guitar,Piano','1'),
('600', '5 months','3','Yamaha','Trumpet,Drums,Guitar','2');
```

4 Resultat

4.1 uppgift 1 – Logisk och fysisk modell

Den logiska och fysiska modellen presenteras nedan:



Länk till GitHub: https://gits-15.sys.kth.se/matsandb/IV1351

I denna uppgift har det skett förhållandevis mycket i min modell i Astah. Exempelvis så har enheter som ”room” tagits bort och lagts till som ett attribut i enheten ”school” vilket jag blev tipsad om från ledningen från den förra inlämningsuppgiften. Det enheterna som jag tyckte var extra elementära till skolans modell och därmed skulle få en egen id som primär nyckel var följande enheter ”studet”, ”instructor”, ”school” och ”group_lesson” såsom på email och phone vilket konstruerades enligt ”many-to-many” principen.

Det enheter som fick främmande nycklar som primära nycklar var exempelvis ”parent” som fick ”student_id” som sin primära nyckel vilket jag tyckte var logiskt korrekt då det lätt i databasen kommer kunna matcha varje elev med sin förälder (kontaktperson). Ett annat exempel på dessa främmande nycklar som är enheters primära nycklar är enheten ”application” vilket fick ”student_id” som primär nyckel eftersom när studenten fyller i önskvärd svårighetsgrad samt genre av musik kommer databasen enkelt kunna para ihop rätt ansökan med rätt student med hjälp av studentens id med all information som den innehåller. På detta vis fortsatte jag att skapa relationer mellan det olika enheterna på det mest logiska sättet jag kunde tänka mig. I exemplet där det går relationer från” instructor” och ”student” till ”school” vilket har sin egen primära nyckel så läggs det två främmande nycklarna ”instructor_id” och ”student_id” till som attribut i ”school”.

Jag började att konstruera mitt sql -program för att skapa en databas som skulle spegla den logiska och fysiska modellen i Astah manuellt innan jag insåg att det fanns en funktion som gjorde det automatiskt. Även fast det var tidskrävande så fortsatte jag manuellt för att jag

kände att jag lärde mig språket och implementationen mycket bättre på detta sätt. Efter att jag hade skapat mitt ”TABLE” som skulle spegla enheten och dess attribut i Astah så införde jag samma begränsningar och datatyper vilket var ”INT” för alla nycklar och ”VARCHAR” och ”TIMESTAMP” för resten av innehållet. Efter varje ”TABLE” så tilldelade jag en eller flera primära nycklar genom kommandot ”ALTER TABLE”. I slutet av programmet så kommer en lång lista med kod där jag kortfattat beskriver för databasen vilka nycklar i respektive attribut som är främmande nycklar ”FK”. Det beskrevs även vad som skulle hända om ”FK” raderades. Det två relevanta alternativ som då fanns var antingen ”ON DELETE CASCADE” vilket betyder att hela enheten försvinner eller kommandot ”ON DELETE SETT NULL” vilket helt enkelt sätter attributet till noll. Sedan utfördes några tester med delvis slumpmässiga data för väsentliga enheter från databasen.

5 Diskussion

5.1 Uppgift 1 – Logisk och fysisk modell

Are naming conventions followed? Are all names sufficiently explaining?

Namnkonventionen är annorlunda i den logiska och fysiska modellen än vad den var i den konceptuella modellen. Denna namnkonvention kallas för ”snake case” och används vanligtvis för variabler, filnamn och subrutinnamn. ”snake case” namnkonvention betyder att alla bokstäver är små och att orden har ett understreck emellan sig om det förekommer flera ord.

Is the crow foot notation correctly followed?

Eftersom jag skapade ett ”Class Diagram” i den första uppgiften så blev det mycket nytt med det nya sättet att dra relationer i ett ER Diagram men försökte att efterlikna mitt ”Class Diagram” samt tänka logiskt. Jag valde att det huvudsakliga enheter som student och ”instructor” skulle ha så många ”one-to-many” relationer som möjligt medan en viss logik fortfarande skulle kunna följas i databasen. På grund av denna anledning ser relationerna ut på det vis det gör i min logiska och fysiska modell.

Are all tables relevant? Is some table missing?

Vissa enheter som exempelvis det som rör betalningssystemen kan man motivera att man kan utvinna den informationen på andra sätt utan att skapa nya enheter med nya attribut men jag tycker att detta är det tydligaste sättet att symbolisera denna process. Efter noggrann läsning så ser jag ingen trivial information som saknas och med denna nya modell som innehållsmässigt är väldigt lik ser jag ingen information som skulle ha försvunnit eller blivit otydlig. Som jag lyfte i tidigare rapport så skulle jag troligtvis kunna förenkla modellen med mer komplexa enheter, attribut och relationer men bestämmer mig att hellre ta med lite för mycket information och sedan anpassa beroende på vad kommande uppgifter anses vara relevant.

Are there columns for all data that shall be stored? Are all relevant column constraints and foreign key constraints specified? Can all column types be motivated?

Jag valde att låta mycket information vara frivillig det vill säga att inte döpa data till ”NOT NULL” då jag ville ge en stor frihet till kundernas önskemål och försökte tänka på vilka verkliga fall där tex en student inte skulle vilja ange sin ålder mm. Det jag valde att döpa till ”NOT NULL” var bland annat elementär kontaktuppgift, tilltalsnamn, olika id samt uppgifter från skolan som rör elever som prissättning på olika lektioner. Det enda som jag tyckte skulle vara speciellt för varje deltagande aktör i denna datastruktur var dess id och dom är därmed döpta till ”UNIQUE”, plus personens personnummer givetvis. Jag valde ”VARCHAR” för varje attribut som inte hade med tid att göra enligt motivation från Leif Lindbäck vilket jag höll med om. Jag ville inte behöva gå in i efterhand och ändra min logiska och fysiska modell så jag valde ett högt max tak för antalet tecken på 500 för majoriteten av attributen. Och sedan använde jag ”TIMESTAMP” för det attribut som var tidsrelaterade då gör det enkelt och smidigt för visa datum och tid. Alla nycklar fick datatypen INT som ska vara SERIAL.