NTNU

Kunnskap for en bedre verden

Institutt for datateknologi og informatikk

TDT4145 - Datamodellering og databasesystemer

# Delivery, part 1: ER-model and relational schema

*Forfatter:*
Ivar Haugland

Spring, 2024

# Table of Contents

# 1 ER-Model

## 1.1 ER-Model design assumptions

1. A ticket is only for one seat.
2. Email is not unique for employee and actor.
3. A role belongs to only one play.
4. Roles can exist without being registered to an act, vice versa.
5. Design choice to register area directly in a ticket, rather than having to join ticket with chair to get the area. Leads to increased query performance, but also increased storage costs.
6. We want email, (employee) status and description for actors.
7. Assume it should be possible to add more tasks for employees in plays than the ones mentioned in project description, instead of creating constraints for only the mentioned tasks.

## 1.2 ER-model constraints that cannot be expressed through the model

1. For group of customers where price of minimal 10 like ”Group 10 and Group honours 10”, it is impossible to create the constraint where the price for these groups get determined by a minimal of 10 tickets in the ER-model.
2. Unique constraint on attributes for some entities. See section 1.3 and section 2 for which entities and attributes this applies.

## 1.3 ER-Model design choices:

Some entities can be considered ”weak” in the ER model, though it does not have the weak property. This applies to the entities:

- ”Area” w/ weak relation ”AreaInHall”
- ”Chair” w/ weak relation ”ChairInArea”
- ”TicketPrice” w/ weak relation ”DeterminesPrice”
- ”Act” w/ weak relation ”InPlay”

Some entity relations are many-to-many and requires its own relational entity. These are:

- ”ActorRole”, a relation entity between ”Actor” and ”Role”
- ”ActorPlay”, a relation entity between ”Actor” and ”Play”
- ”EmployeePlay”, a relation entity between ”Employee” and ”Play”
- ”ActRole”, a relation entity between ”Act” and ”Role”

I’m mentioning this because the second criterion for evaluating the project is about ”Using keys: Natural vs. generated keys.” All my keys in the model are generated, including the ”weak” and relational entities I talked about before. I went for this approach not because it’s some superior ER-model design choice, but mainly because it fits my project goals and tech stack. I wanted to do more than the basics and get into creating an API for the database, so I adjusted the database setup to match the application stack I picked. Bringing up the application stack is just to explain why the database ended up designed this way without composite and natural primary keys. The next sections get into the application stack and how I dealt with the lack of composite primary keys.

For a straightforward setup of database models and schemas for the API, I picked Tortoise ORM. It's handy because it can automatically serialize Pydantic models from Tortoise models, making it easier to validate user input for API requests like GET, PUT, POST, and DELETE. Given the project's size, I didn't want to manually build extensive schema models for each object, so I went with Tortoise ORM for its simplicity. SQLAlchemy ORM was another option, but it would've meant creating hefty Pydantic schemas for each API operation since it doesn't support auto-serialization of Pydantic models.

One bummer with Tortoise ORM is the lack of composite key functionality. The primary keys in Tortoise models only play well with single and non-composite primary keys, which clashes with the second criterion. It gets tricky to create natural keys with weak and relational entities when there's no support for composite primary keys. Take the "Area" entity, for example; having a composite primary key with attributes like "name" and foreign key "hall_id" would be a nice. Sadly, that's a no-go with the current release of Tortoise ORM. To work around this, I added unique and not null constraints on the attributes that should've been composite primary keys. This workaround applies to all the "weak" and relational entities I brought up earlier, even if it means a bit of a hit on storage efficiency.

### 1.4 ER-model diagram

Figure 1 shows the ER-model diagram from the project description.

## 2 Relational schemas

**TheaterHall**(hall_id, name, capacity)

- hall_id is primary key in the "TheaterHall" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on hall_id, and hall_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Hall_id can't have more values for attributes name and capacity. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Season**(season_id, season, year)

- season_id is primary key in the "Season" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on season_id, and season_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Season_id can't have more values for attributes season and year. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Play**(play_id, title, author, description, season_id, hall_id)

- play_id is primary key in the "Play" table.
- season_id is foreign key to the "Season" table.
- hall_id is foreign key to the "TheaterHall" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on play_id, and play_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Play_id can't have more values for attributes title, author, description, season_id and hall_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Performance**(performance_id, datetime, play_id, hall_id)

- performance_id is primary key in the "Performance" table.
- play_id is foreign key to the "Play" table.
- hall_id is foreign key to the "TheaterHall" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on performance_id, and performance_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Performance_id can't have more values for attributes datetime, play_id and hall_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Area**(area_id, name, hall_id)

- area_id is primary key in the "Area" table.
- hall_id is foreign key to the "TheaterHall" table.
- name and hall_id are composite unique.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on area_id, and area_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Area_id can't have more values for attributes name and hall_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Chair**(chair_id, number, row, area_id, hall_id)

- chair_id is primary key in the ”Chair” table.
- area_id is foreign key to the ”Area” table.
- hall_id is foreign key to the ”TheaterHall” table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on chair_id, and chair_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Chair_id can't have more values for attributes number, row, area_id and hall_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**CustomerProfile**(customer_id, userName, name, address, phone)

- customer_id is primary key in the ”CustomerProfile” table.
- userName is unique field

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on customer_id, and customer_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Customer_id can't have more values for attributes name, address and phone. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**CustomerGroup**(group_id, name)

- group_id is primary key in the ”CustomerGroup” table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on group_id, and group_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Group_id can't have more values for attributes name. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**TicketPurchase**(purchase_id, datetime, customer_id)

- purchase_id is primary key in the "TicketPurchase" table.
- customer_id is foreign key to the "CustomerProfile" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on purchase_id, and purchase_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Purchase_id can't have more values for attributes datetime and customer_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**TicketPrice**(ticketPrice_id, price, group_id, play_id)

- ticketPrice_id is primary key in the "TicketPrice" table.
- group_id is foreign key to the "CustomerGroup" table.
- play_id is foreign key to the "Play" table.
- group_id and play_id are composite unique.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on ticketPrice_id, and ticketPrice_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

TicketPrice_id can't have more values for attributes price, group_id and play_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Ticket**(ticket_id, purchase_id, performance_id, chair_id, area_id, ticketPrice_id)

- ticket_id is primary key in the "Ticket" table.
- purchase_id is foreign key to the "TicketPurchase" table.
- performance_id is foreign key to the "Performance" table.
- chair_id is foreign key to the "Chair" table.
- area_id is foreign key to the "Area" table.
- ticketPrice_id is foreign key to the "TicketPrice" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on ticket_id, and ticket_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Ticket_id can't have more values for attributes purchase_id, performance_id, chair_id, area_id and ticketPrice_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Act**(act_id, number, name, play_id)

- act_id is primary key in the "Act" table.
- play_id is foreign key to the "Play" table.
- number and play_id are composite unique.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on act_id, and act_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Act_id can't have more values for attributes number, name and play_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Role**(role_id, name, play_id)

- role_id is primary key in the "Role" table.
- play_id is foreign key to the "Play" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on role_id, and role_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Role_id can't have more values for attributes name and play_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Actor**(actor_id, name, email, status, description)

- actor_id is primary key in the "Actor" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on actor_id, and actor_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Actor_id can't have more values for attributes name, email, status and description. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**Employee**(employee_id, name, email, status, description, task)

- employee_id is primary key in the "Employee" table.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on employee_id, and employee_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

Employee_id can't have more values for attributes name, email, status, description and task. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**ActorRole**(actorRole_id, actor_id, role_id)

- actorRole_id is primary key in the "ActorRole" table.
- actor_id is foreign key to the "Actor" table.
- role_id is foreign key to the "Role" table.
- actor_id and role_id are composite unique.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on actorRole_id, and actorRole_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

ActorRole_id can't have more values for attributes actor_id and role_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**ActorPlay**(actorPlay_id, actor_id, play_id)

- actorPlay_id is primary key in the "ActorPlay" table.
- actor_id is foreign key to the "Actor" table.
- play_id is foreign key to the "Play" table.
- actor_id and play_id are composite unique.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on actorPlay_id, and actorPlay_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

ActorPlay_id can't have more values for attributes actor_id and play_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**EmployeePlay**(employeePlay_id, employee_id, play_id)

- employeePlay_id is primary key in the "EmployeePlay" table.
- employee_id is foreign key to the "Employee" table.
- play_id is foreign key to the "Play" table.
- employee_id and play_id are composite unique.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on employeePlay_id, and employeePlay_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

EmployeePlay_id can't have more values for attributes employee_id and play_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

**ActRole**(actorRole_id, act_id, role_id)

- actorRole_id is primary key in the "ActRole" table.
- act_id is foreign key to the "Act" table.
- role_id is foreign key to the "Role" table.
- act_id and role_id are composite unique.

Candidate key consists of only one key attribute. This means that there won't be any problem with other non-key attributes being partly dependent on the candidate key. *The table is therefore in second normal form.*

All of the non-key attributes are functionally dependent on actorRole_id, and actorRole_id is super key in the table. *The table is therefore in third normal form.*

Because the candidate key only consists of one key attribute, there won't occur overlapping candidate keys caused by functional dependencies between key attributes. *The table is therefore in Boyce-Codd normal form.*

ActorRole_id can't have more values for attributes act_id and role_id. There is no MVD's in the table. Since the table is in third normal form without MVD's, *the table is therefore in fourth normal form.*

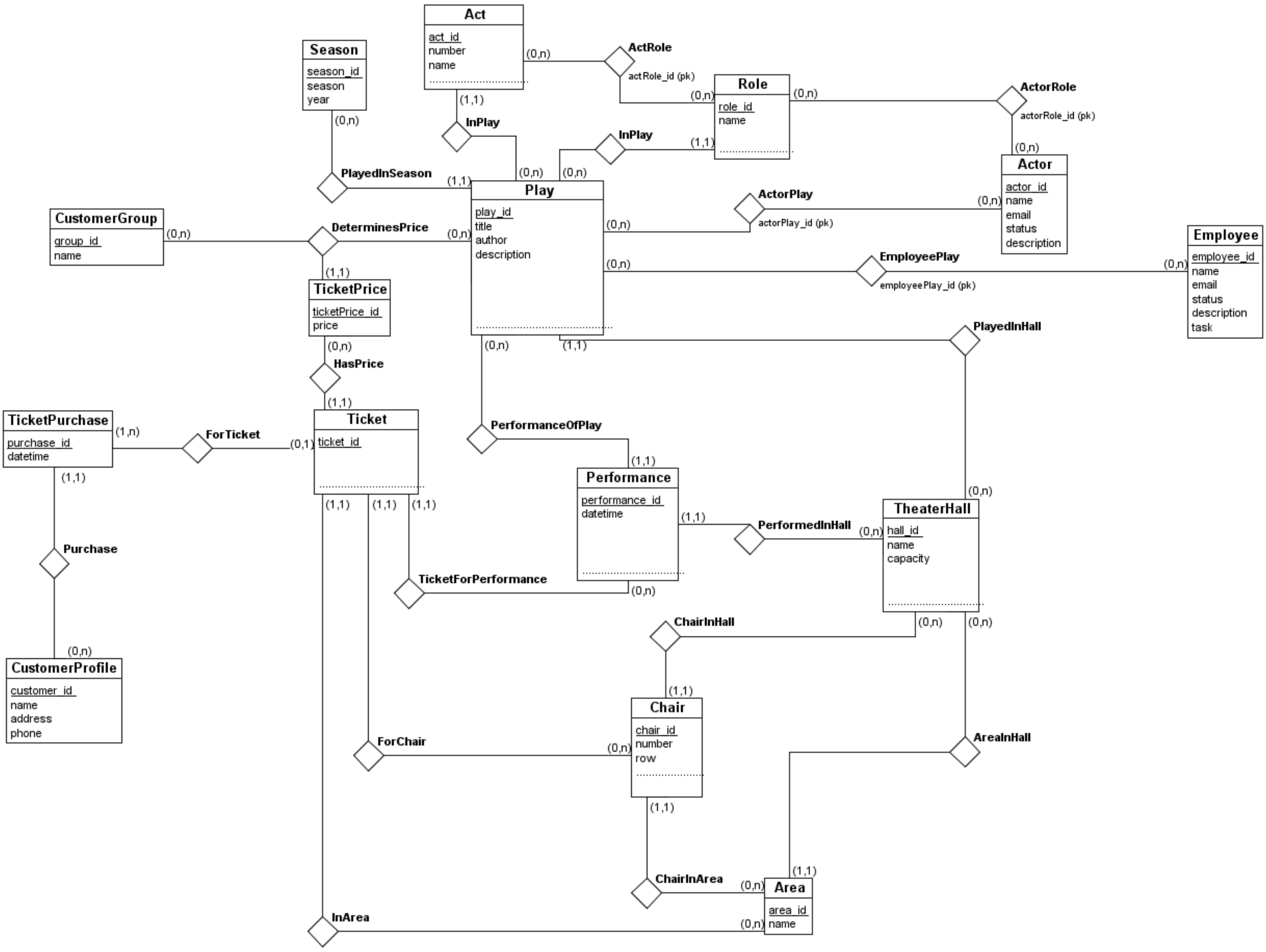

Figure 1: ER-model diagram from the project description