

Normalization

Normalization is a process of organizing data in the database.

It is used to minimize the redundancy from relations or a set of relations.

➤ **1st Normal Form:**

It says that a relation is in 1NF if and only if:

- The domain of each attribute contains only atomic value.
- The value of each attribute contains only a single value from that domain.
- There should not be repeating groups.

➤ **2nd Normal Form:**

It says that a relation is in second normal form if and only if:

- The relation should be in 1NF.
- Every non key attribute should be fully functionally dependent on each candidate key.

➤ **3rd Normal Form:**

It says that a relation is in 3rd normal form if and only if:

- It is in 2NF.
- Transitive functional dependency of a non-prime attribute on any candidate key should be removed. Or

All Fields can be determined only by the key in the table and no other column.

Theatre (th_id, loc, city, state)

<u>Th_id</u>	loc	city	state
100	Akota	Vadodara	Gujarat
101	Sarkhej	Ahmedabad	Gujarat

Each cell is single valued

The table is in 1NF.

The table is in 2NF as there are no candidate keys.

$\text{th_id} \rightarrow \text{loc}$ and $\text{loc} \rightarrow \text{city}$ hence $\text{th_id} \rightarrow \text{city}$

Also, $\text{city} \rightarrow \text{state}$ so $\text{th_id} \rightarrow \text{state}$.

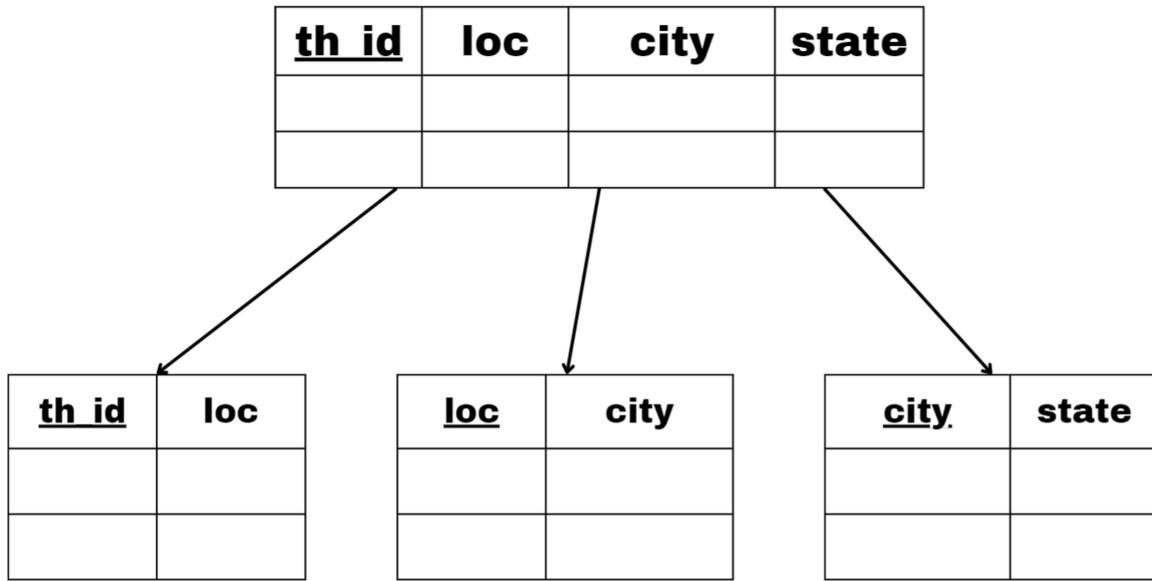
Since there is transitivity, the table is not in 3NF.

The table can be divided as:

Th_main (th_id, loc)

Th_loc (loc, city)

Th_st (city, state)



Screen (screen_id, th_id, silver_seats, gold_seats, screen_name)

<u>screen_id</u>	<u>th_id</u>	silver_seats	Gold_seats	screen_name

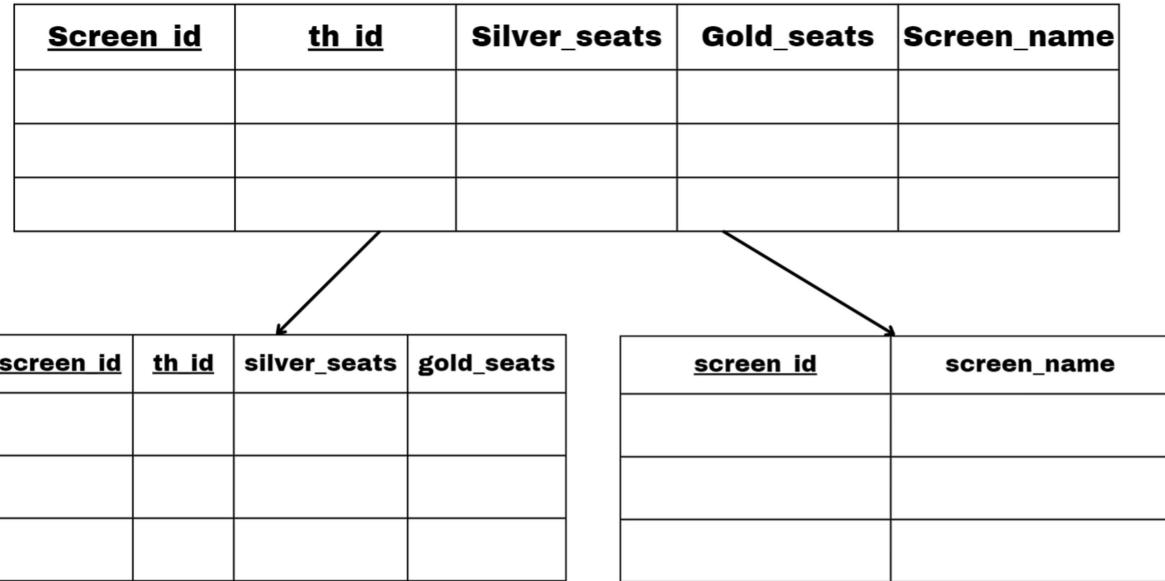
Each attribute is single valued. Therefore, it is in 1NF.

The number of silver seats and gold seats a screen will have is dependent on screen_id as well th_id. However, screen names depend on only screen_id.

So, table can be divided as:

Screen (screen_id, th_id, silver_seats, gold_seats)

Screen_name (screen_id, screen_name)



Both the tables are in 2NF.

Since Screen table has no transitivity, it is in 3NF.

Projection_Room (screen_id, th_id, server, type)

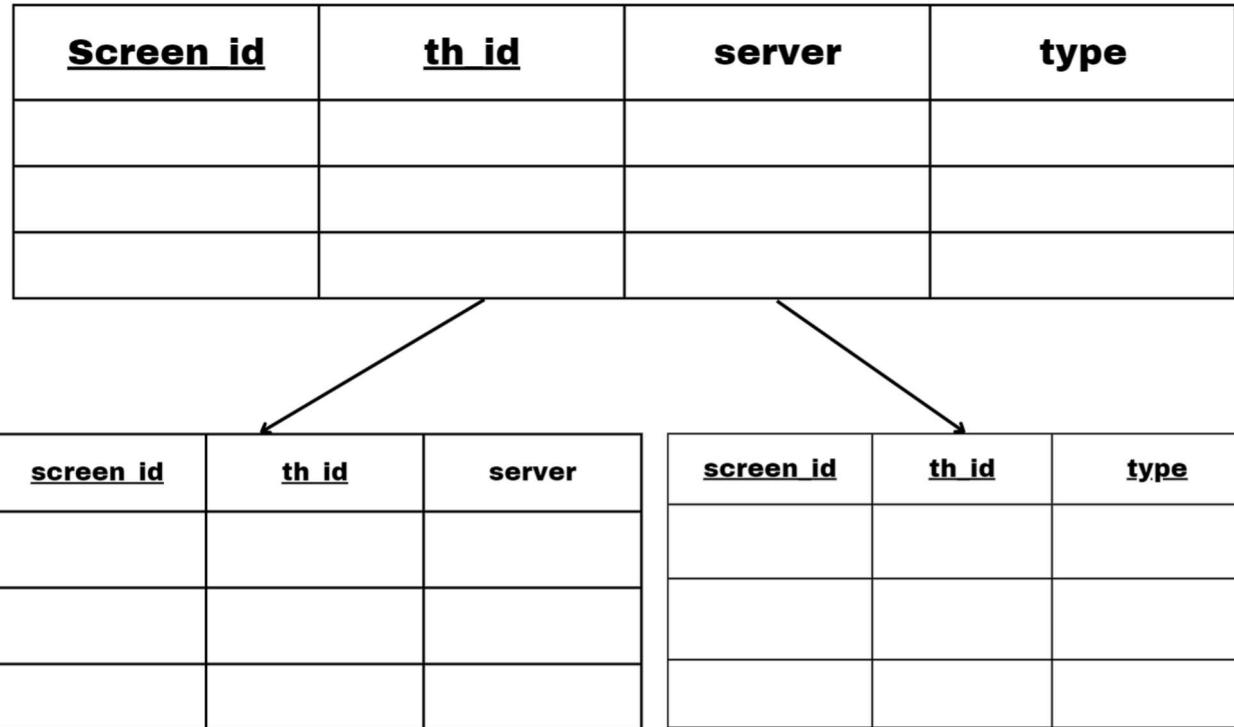
<u>Screen_id</u>	<u>th_id</u>	server	type

Since there can be multivalve for type attribute (2D, 3D), the table is not in 1NF.

The table can be divided as:

Projection_Room (screen_id, th_id, server)

Projection_type (screen_id, th_id, type)



As server depends on **screen_id** and **th_id** together the table **Pro_Room** is in 2NF. There is no more than 1 non-prime attribute So, it is also in 3NF.

Pr_type has no non-prime attribute, so it is also in 2NF and in 3NF.

Employee (eid, th_id, ename, email, phone_no, job, hiredate, sal)

<u>e_id</u>	<u>th_id</u>	ename	email	Phone_no	job	hiredate	sal	comm
-------------	--------------	-------	-------	----------	-----	----------	-----	------

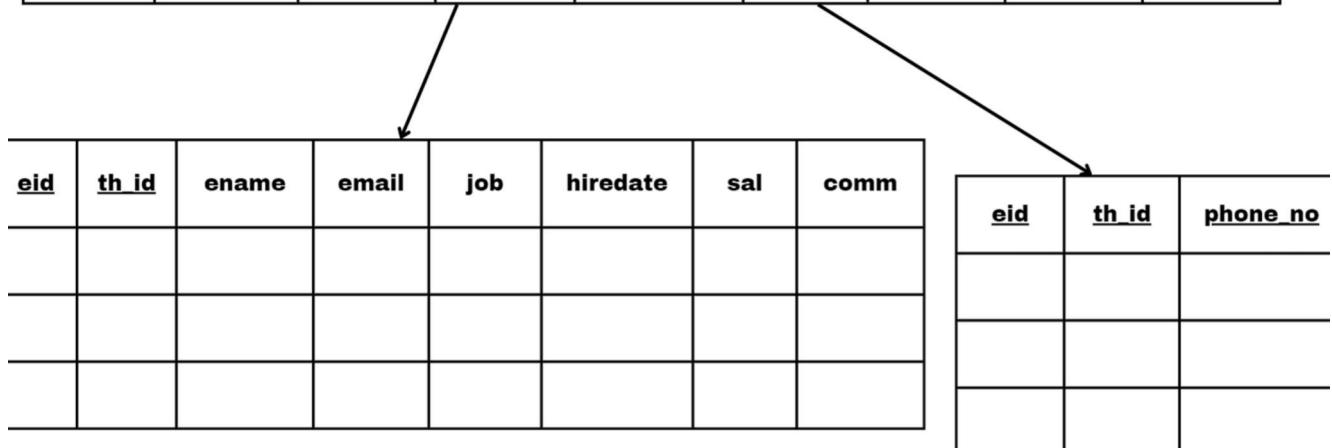
One employee can have more than one contact number, so the table is not in 1NF.

It can be divided as:

Employee (eid, th_id, ename, email, job, hiredate, sal, comm)

Em_no (eid, th_id, phone_no)

<u>eid</u>	<u>th_id</u>	ename	email	phone_no	job	hiredate	sal	comm



As ename depend only on eid and not on th_id so the table Employee is not in 2NF and sal depend on job and th_id not on eid. So table is divided as:

Employee_n (eid, ename)

Employee_jsal(th_id, job, sal, comm)

Employee (eid, th_id, email, job, hiredate)

<u>eid</u>	<u>th_id</u>	ename	email	job	hiredate	sal	comm

↓

<u>th_id</u>	job	sal	comm

↓

<u>eid</u>	<u>th_id</u>	job	hiredate

Projectionist (pid, th_id, shift)

<u>pid</u>	<u>th_id</u>	shift

Each projectionist will have only one shift, so it is in 1NF.

As shift depends on pid as well as th_id, it is in 2NF.

Since there is no more than 1 non-prime attribute, it is in 3NF.

Movie (mov_name, release_dt, duration, lang, genre, budget)

Mov_name	Release_dt	duration	language	genre	budget	M_share

As one movie has many genres and it can be in more than 1 language, the table is not in 1NF.

It can be divided as:

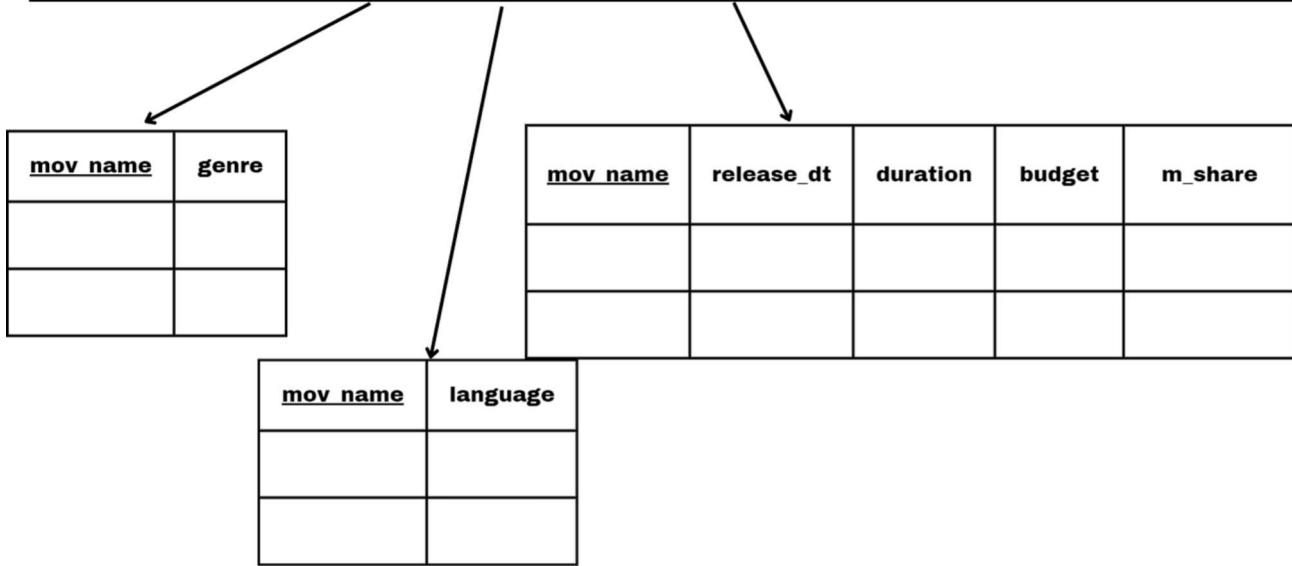
Movie (mov_name, release_dt, duration, budget, movie_share)

Movie_genre (mov_name, genre)

Movie_lang (mov_name, language)

As there is no non-prime attribute in Movie_genre and Movie_lang, they both are in 2NF and 3NF.

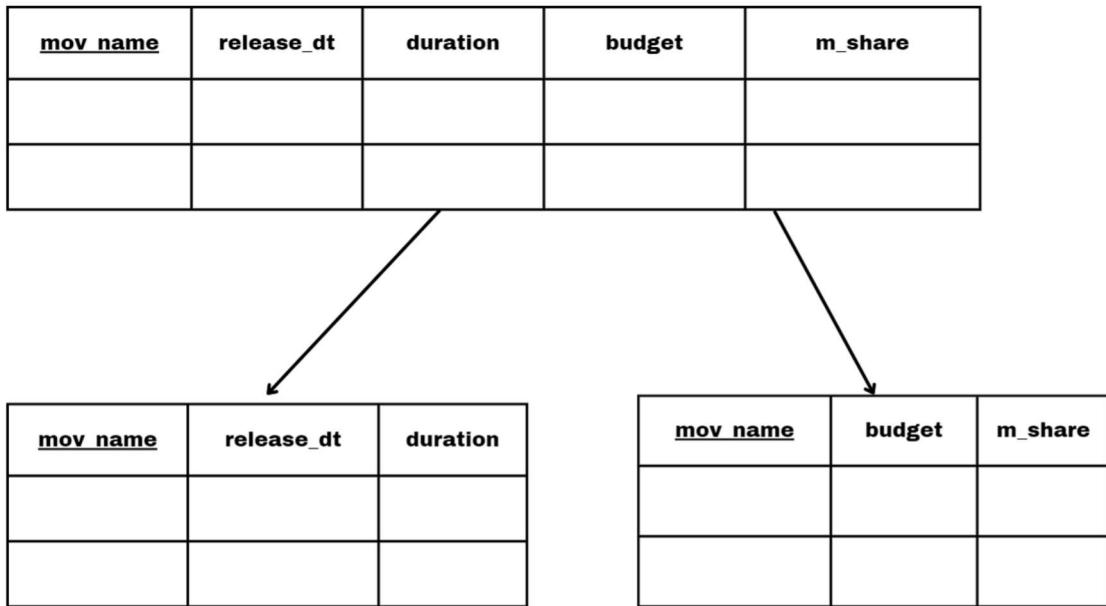
<u>mov_name</u>	release_dt	duration	language	genre	budget	movie_share



M_share depend on budget and budget depend on mov-name. So there is transitivity. So it is not in 3NF. The table can be divided as :

Movie (mov_name, release_dt, duration)

Movie_share (mov_name, budget, m_share)



Shows (screen_id, th_id, sdatetime, mov_name, runtime)

<u>Screen_id</u>	<u>Th_id</u>	<u>sdatetime</u>	<u>Mov_name</u>	<u>runtime</u>

Mov_name and runtime will be single valued for same screen_id, th_id and sdatetime together.

So, the table is in 1NF.

Mov_name and runtime both are totally dependent on the candidate key (screen_id, th_id, sdatetime). So, the table is in 2NF.

As there is no transitivity the table is in 3NF.

Silver_seat (silver_seatnum, screen_id, th_id, sdatetime, availability, price)

<u>Silver_seatnum</u>	<u>Screen_id</u>	<u>Th_id</u>	<u>sdatetime</u>	<u>availability</u>	<u>price</u>

Availability is a single value attribute, so the table is in 1NF.

Now price depends on screen_id, th_id and sdatetime only because each seat is silver. Therefore, the table is not in 2NF. The table is divided into:

Silver_seat (silver_seatnum, screen_id, th_id, sdatetime, availability)

Silver_price(th_id, screen_id, sdatetime, price)

<u>Silver_seatnum</u>	<u>Screen_id</u>	<u>Th_id</u>	<u>sdatetime</u>	availability	price

<u>Silver_seatnum</u>	<u>Screen_id</u>	<u>Th_id</u>	<u>sdatetime</u>	availability

<u>Th_id</u>	<u>Screen_id</u>	<u>sdatetime</u>	price

Availability depends on the candidate key totally so the silver_seat table is in 2NF. Price depends on th_id, screen_id and sdatetime together. So, table silver_price is also in 2NF.

As there is no more than 1 non-prime attribute the tables are in 3NF.

Gold_seat (gold_seatnum, screen_id, th_id, sdatetime, availability, price)

<u>Gold_seatnum</u>	<u>Screen_id</u>	<u>Th_id</u>	<u>sdatetime</u>	availability	price

Availability is a single value attribute, so the table is in 1NF.

Now price depends on screen_id, th_id and sdatetime only because each seat is gold. Therefore, the table is not in 2NF. The table is divided into:

Gold_seat (gold_seatnum, screen_id, th_id, sdatetime, availability)

Gold_price(th_id, screen_id, sdatetime, price)

<u>Gold_seatnum</u>	<u>Screen_id</u>	<u>Th_id</u>	<u>sdatetime</u>	availability	price

<u>Gold_seatnum</u>	<u>Screen_id</u>	<u>Th_id</u>	<u>sdatetime</u>	availability

<u>Th_id</u>	<u>Screen_id</u>	<u>sdatetime</u>	price

Availability depends on the candidate key totally so the gold_seat table is in 2NF. Price depends on th_id, screen_id and sdatetime together. So, table gold_price is also in 2NF.

As there is no more than 1 non-prime attribute the tables are in 3NF.

Customer (Cid, cname, phone_num, email, payment_method, th_id)

<u>cid</u>	Th_id	cname	Phone_num	email	Payment_method

As each attribute is a single valued the table is in 1NF.

It is in the 2NF as there are no candidate keys.

There is no transitivity, the table is in 3NF.

**SilverTicket (ticket_id, th_id, nprice, bk_dt_time,
bk_type, sdatetime, screen_id, silver_seatnum, cid,
webname)**

**GoldTicket (ticket_id, th_id, nprice, bk_dt_time,
bk_type, sdatetime, screen_id, gold_seatnum, cid,
webname)**

Boxoffice (th_id, sdatetime, screen_id, mov_name)

<u>Th_id</u>	<u>sdatetime</u>	<u>Screen_id</u>	<u>Mov_name</u>

Each attribute is single valued, so the table is in 1NF.

Screen_id depends on th_id, sdatetime and mov_name together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

Website (wid, sdatetime, screen_id, mov_name)

<u>wid</u>	<u>sdatetime</u>	<u>Screen_id</u>	<u>Mov_name</u>

Each attribute is single valued, so the table is in 1NF.

Screen_id depends on wid, sdatetime and mov_name together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

WebsiteDetails (th_id, webname, mov_name, webshare_in_per)

<u>Th_id</u>	<u>webname</u>	<u>Mov_name</u>	<u>Webshare_in_per</u>

Each attribute is single valued, so the table is in 1NF.

Webshare_in_per depends on th_id, mov_name and webname together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

AdvertisementDetails (th_id, ad_id, mov_name, per_show_price)

<u>Th_id</u>	<u>Ad_id</u>	<u>Mov_name</u>	<u>Per_show_price</u>

Each attribute is single valued, so the table is in 1NF.

Per_show_price depends on th_id, mov_name and ad_id together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

AdsShown (th_id, screen_id, sdatetime, ad_id)

<u>Th_id</u>	<u>Screen_id</u>	<u>sdatetime</u>	<u>Ad_id</u>

Each attribute is single valued, so the table is in 1NF.

As there are no non-prime attributes the table is in 2NF and 3NF.

AdvertisementRevenue (th_id, rdate, mov_name, ad_id, revenue)

<u>Th_id</u>	<u>rdate</u>	<u>Mov_name</u>	<u>Ad_id</u>	<u>revenue</u>

Each attribute is single valued, so the table is in 1NF.

revenue depends on th_id, mov_name, ad_id and rdate together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

TicketRevenue (th_id, mov_name, revenue_date, ticketrevenue)

<u>Th_id</u>	<u>mov_name</u>	<u>Revenue_date</u>	<u>ticketrevenue</u>

Each attribute is single valued, so the table is in 1NF.

ticketrevenue depends on th_id, mov_name and revenue_date together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

MovieRevenue (th_id, mov_name, revenue_date, movieRevenue)

<u>Th_id</u>	<u>mov_name</u>	<u>Revenue_date</u>	<u>movieRevenue</u>

Each attribute is single valued, so the table is in 1NF.

MovieRevenue depends on th_id, mov_name and revenue_date together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

WebRevenue (th_id, mov_name, revenue_date, wname, webrevenue)

<u>Th_id</u>	<u>Mov_name</u>	<u>Revenue_date</u>	<u>wname</u>	<u>webrevenue</u>

Each attribute is single valued, so the table is in 1NF.

Webrevenue depends on th_id, mov_name wname and revenue_date together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

ConcessionDetails(th_id, conc_name, conc_price)

<u>Th_id</u>	<u>Conc_name</u>	<u>Conc_price</u>

Each attribute is single valued, so the table is in 1NF.

Conc_price depends on th_id, conc_name together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

Concession (th_id, conc_name, cid, orderdttime, quantity)

<u>Th_id</u>	<u>Conc_name</u>	<u>cid</u>	<u>orderdttime</u>	<u>quantity</u>

Each attribute is single valued, so the table is in 1NF.

Orderdttime and quantity depends on th_id, conc_name and cid together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

ConcessionQuantity(th_id,conc_name, revenuedate, quantity)

<u>Th_id</u>	<u>Conc_name</u>	<u>revenuedate</u>	<u>quantity</u>

Each attribute is single valued, so the table is in 1NF.

quantity depends on th_id, conc_name and revenuedate together. So the is in 2NF.

As there is no more than 1 non-prime attribute the table is in 3NF.

ConcessionRevenue (th_id, conc_name, revenuedate, conc_rev)

<u>Th_id</u>	<u>Conc_name</u>	<u>revenuedate</u>	<u>Conc_rev</u>

Each attribute is single valued, so the table is in 1NF.

Conc_rev depends on th_id, conc_name and revenuedate together. So the is in 2NF.

As there are no more than 1 non-prime attribute the table is in 3NF.

Function

1. To find screen id for a movie show in a particular theatre.

```
create function f1(thid in shows.th_id%type,sdt in  
shows.sdatetime%type,mov in shows.mov_name%type) return number
```

```
is
sid number(2);
begin
select screen_id into sid from shows where sdatetime=sdt and
th_id=thid and mov_name=mov;
return sid;
end;
/
```

--calling program

```
declare
m shows.mov_name%type;
th shows.th_id%type;
sdt1 shows.sdatetime%type;
scid shows.screen_id%type;
begin
m:='Tu Jhuthi Main Makkar';
th:=100;
sdt1:=to_date('20-Apr-2023 12:00:00 pm','dd-mon-yyyy hh:mi:ss
pm');
scid:=f1(th,sdt1,m);
dbms_output.put_line(scid);
end;
```