SHENKAR —— שנקר

ENGINEERING. DESIGN. ART . הנדסה. עיצוב. אמנות

The Pernick Faculty of Engineering . הפקולטה להנדסה ע"ש פרניק

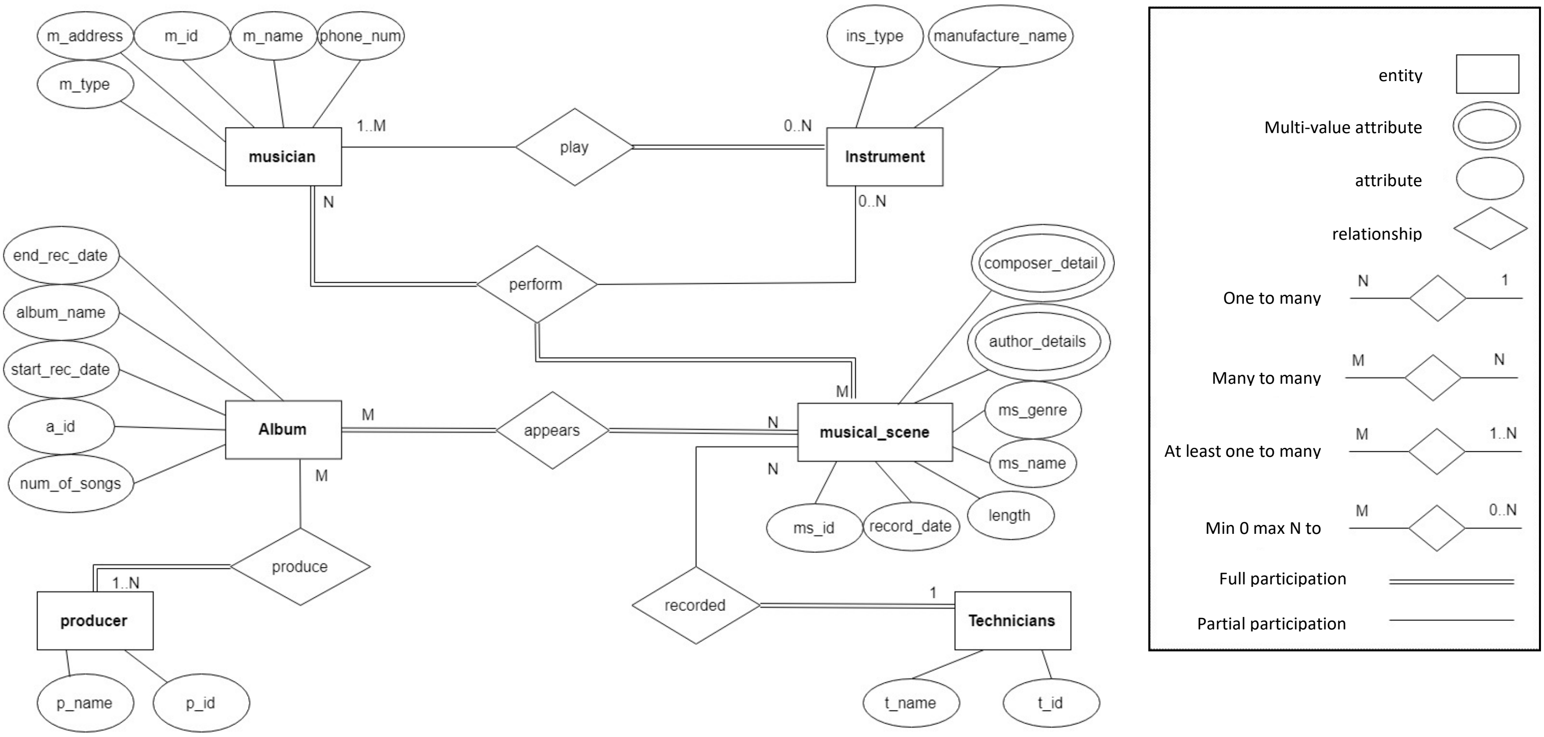
המחלקה להנדסת תוכנה
Software Engineering Dept.

Final project

file systems and databases

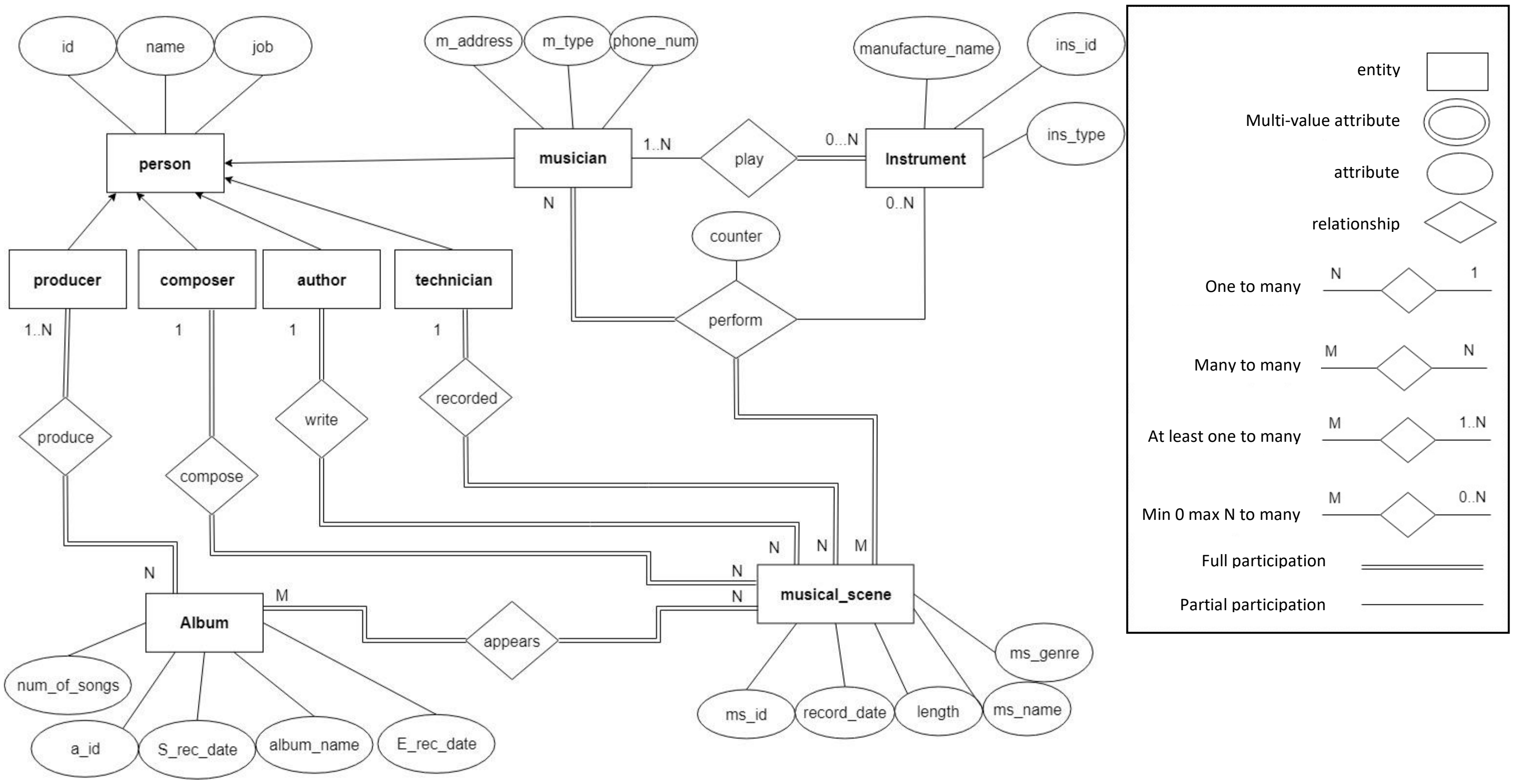
Initial database schema

ERD:



Final database schema

ERD:



Tables:

person_table	
PK	id (int)
	name (string) job (string)

instrumnet_details_table	
PK	ins_id (int)
	ins_type (string) manufacture_name (string) i_amount_in_ms (int)

album_details_table	
PK	a_id (int)
	album_name (string) num_of_songs (int) s_rec_date (date) e_rec_date (date)

musicaian_details_table	
PK FK	m_id (int)
	m_address (string) phone_number (string) m_type (int) song_amount (int)

music_scene_tedail_table	
PK	ms_id (int)
	ms_name (string) record_date (date) length (int) ms_genre (string)
FK	t_id (int)
FK	autor_id (int)
FK	composer_id (int)

ins_in_ms	
PK FK	part_id (int)
FK	instrument_id (int)

PK

produce_album	
FK	id (int)
FK	p_id (int)

PK

musical_scene_in_album	
FK	album_id (int)
FK	musical_scene_id

musician_in_ms	
PK	participant_id (int)
FK	musican_id (int)
FK	m_scene_id (int)

PK

musician_play_ins	
FK	musican_id (int)
FK	ins_type (string)

tables Explanation:

1. Person_table - represents all the people in the system (musicians, technicians, composers, poets, producers).
   The person type is identified by the string type job field (m - musician, t - technician, c - composer, a - poet, p - producer).
   The id field in the person_table table will be associated with all other relevant tables (in the musician_details_table table for the m_id field, in the music_scene_tedail_table table for t_id, autor_id, composer_id fields)
2. Album_details_table - Represents the list of albums in the system.
3. Instrument_details_table - Represents the list of musical instruments in the system.
4. Musician_details_table - Represents the list of musicians in the system.
5. Music_scene_tedail_table - Represents the list of musical scenes in the system.

Linking tables:

6. Musician_in_ms - Represents which musician participates in which musical scene.
7. Ins_in_ms - Represents the musical instrument in a musical scene and who plays it.
8. Produce_album - Represents which producer produced which album.
9. Musical_scene_in_album - Represents which musical scene appears in which album.
10. Musician_play_ins - Represents which musical instruments every musician knows how to play.

Table data:

music_scene_tedail_table

ms_id	ms_name	record_date	leght	ms_genre	t_id	autor_id	composer_id
1	wind	01/01/2018	180	rock	13	15	18
2	roots	01/02/2018	240	classic	13	16	17
3	late	01/03/2018	187	classic	14	15	18
4	all about	01/04/2018	200	hip hop	13	15	18
5	river	01/05/2018	202	rock	13	16	17
6	song2	01/06/2018	300	rock	14	16	18
7	help	01/07/2018	304	hip hop	14	15	17
8	goodbye	01/08/2018	189	rock	13	16	18
9	hello	01/09/2018	193	hip hop	14	15	17
10	osher	01/10/2018	215	rock	14	16	17

person_table

id	name	job
1	omer	m
2	eden	m
3	evri	m
4	aviv	m
5	itzik	m
6	zohar	m
7	sarit	m
8	eli	m
9	rotem	m
10	ofra	m
11	dana	p
12	odi	p
13	dany	t
14	noa	t
15	michael	a
16	nirit	a
17	adi	c
18	amos	c

instrument_details_table

i_amount_in_ms (number of times the instrument appears in a musical scene in the system)

ins_id	ins_type	manfecture_name	i_amount_in _ms
1	guitar	gibson	1
2	guitar	fender	1
3	piano	wind	1
4	piano	music X	1
5	drums	music X	0
6	drums	gibson	1
7	flute	fender	1
8	flute	wind	1
9	trumpet	wind	2
10	trumpet	music inc	1

album_details_table

a_id	album_name	num_of_songs	s_rec_date	e_rec_date
1	a	2	01/02/2018	05/04/2018
2	trees	1	01/04/2018	10/05/2018
3	love	2	01/01/2018	20/02/2018
4	banana	2	01/02/2018	20/06/2018
5	dog	1	23/06/2018	29/07/2018
6	cpp	1	13/07/2018	24/08/2018
7	master	1	01/08/2018	01/10/2018
8	new	1	20/09/2018	02/10/2018
9	phonix	1	01/01/2018	15/01/2018
10	lion	1	30/01/2018	25/02/2018

produce_album

a_id	p_id
1	11
1	12
2	11
3	12
3	11
4	11
4	12
5	12
6	12
6	11
7	11
8	12
8	11
9	11
9	12
10	11
10	12

musician_play_ins

musican_id	ins_type
1	guitar
1	drums
2	trumpet
3	flute
3	trumpet
4	flute
5	drums
5	guitar
7	piano
8	piano
9	guitar
9	piano
9	drums
9	flute
9	trumpet
10	guitar
10	piano
10	drums
10	flute
10	trumpet

musical_scene_in_album

album_id	musical_scene_id
1	4
1	3
3	1
3	2
2	5
4	6
4	2
5	7
6	8
7	9
10	2
8	10
9	1

musician_in_ms

participant_id	musican_id	m_scene_id
1	2	1
2	4	1
3	6	2
4	5	3
5	8	3
6	6	4
7	7	5
8	7	6
9	9	7
10	3	7
11	10	8
12	1	8
13	10	9
14	2	10
15	6	10

ins_in_ms

part_id	instrument_id
1	9
2	8
4	6
5	4
8	3
9	7
10	9
11	1
13	2
14	10

musician_details_table
m_type(musician type) , 0- singer only, 1-play instrument only, 2-both

m_id	m_address	phone_number	m_type	song_amount
1	holon	052-1234567	2	1
2	tel aviv	054-1234567	2	2
3	ramla	055-1234567	2	1
4	tel aviv	053-1234567	2	1
5	natanya	057-1234567	2	1
6	eilat	077-1234567	0	3
7	haifa	052-7654321	2	2
8	kiryat shmona	054-7654321	2	1
9	haifa	053-7654321	1	1
10	holon	050-7654321	2	2

Base assumptions for work:

In accordance with the description of the given world, we have made some basic assumptions about the system that we characterized in order to describe the relationships between the entities in the system and take the necessary actions:

1. There is a unique name for every person in the system (musician, producer, technician, poet, composer) and there cannot be two people in the system with the same name.

2. There is a unique name for each album in the system. There cannot be two albums in the system with the same name.

3. There is no song without an album (and conversely - there is no album without at least one song).

4. There is no musical scene without a musician.

5. There is a single poet for a musical scene.

6. There is a single composer for a musical scene.

7. We have assumed that the same manufacturer can produce different types of musical instruments.

8. We assumed that a musical scene could be just a song (i.e. only a vocalist) and / or a musical instrument (with the instrument and musician).

System operations in Relational algebra and SQL Queries:

(1 SQL:

```
SELECT count(a_id)

FROM album_details_table

WHERE s_rec_date>= y1 and e_rec_date <= y2;
```

(2 SQL:

```
SELECT count(ms_id)

FROM (select id from person_table where name=x and job='m')   p join musician_in_ms  a on
p.id =a.musican_id join music_scene_tedail_table  b on a.m_scene_id=b.ms_id

WHERE  b.record_date>= y1 and b.record_date <= y2;
```

(3 SQL:

```
SELECT count(distinct musical_scene_in_album.album_id)

FROM (SELECT id FROM person_table WHERE name=x AND job="m") P

join musician_in_ms on p.id=musician_in_ms.musican_id

join music_scene_tedail_table m on m.ms_id=musician_in_ms.m_scene_id

join musical_scene_in_album  on
musical_scene_in_album.musical_scene_id=musician_in_ms.m_scene_id

WHERE m.record_date>= y1 AND m.record_date<= y2;
```

(4 SQL:

```
select ins_type

from instrument_details_table

group by ins_type

order by sum(i_amount_in_ms) desc limit 1;
```

(5 SQL:

```
select ins_type, manufacture_name

from (select a_id

from album_details_table

where album_name=x) as a JOIN musical_scene_in_album on musical_scene_in_album.album_id=a.a_id

JOIN musician_in_ms on musical_scene_in_album.musical_scene_id=musician_in_ms.m_scene_id

JOIN ins_in_ms on musician_in_ms.participant_id=ins_in_ms.part_id

JOIN instrument_details_table on ins_in_ms.instrument_id=instrument_details_table.ins_id;
```

Relational algebra:

$$\rho(r1, \pi_{a\_id}(\sigma_{album\_name=x}(album\_details\_table))$$

$$\pi_{ins_{t}ype,manufacture_{n}ame}(\pi_{ins_{i}d}(\pi_{participant_{i}d}(\pi_{ms\ \ _{id}}(r1 \bowtie musical\_scene\_in\_album)\bowtie musician\_in\_ms) \bowtie ins\_in\_ms) \bowtie instrument\_details\_table)$$

(6 SQL:

```
SELECT name

from person_table join produce_album on person_table.id=produce_album.p_id

join (SELECT a_id from album_details_table WHERE s_rec_date>= y1 and e_rec_date<= y2) as a

on a.a_id=produce_album.a_id

GROUP BY produce_album.p_id

ORDER BY COUNT(produce_album.a_id) DESC limit 1;
```

(7 SQL:

```
select manufacture_name
from instrument_details_table
group by manufacture_name
order by sum(i_amount_in_ms) desc limit 1;
```

(8 SQL:

```
SELECT count(distinct musican_id)
FROM musician_in_ms;
```

(9 SQL:

```
select name
from musician_in_ms as a CROSS JOIN musician_in_ms as b JOIN person_table as p on a.musican_id=p.id
where a.musican_id<>b.musican_id and a.m_scene_id=b.m_scene_id
group by a.musican_id
order by count(id) desc limit 1;
```

(10 SQL:

```
select ms_genre
from instrument_details_table JOIN music_scene_tedail_table
group by ms_genre
order by sum(i_amount_in_ms) desc limit 1;
```

(11 SQL:

```
select name

from music_scene_tedail_table as m JOIN person_table as p on m.t_id=p.id

where m.record_date between y1 and y2

GROUP BY m.t_id

order by count(ms_id) desc limit 1;
```

(12 SQL:

```
select album_name

from album_details_table

where e_rec_date=(select min(e_rec_date) from album_details_table);
```

Relational algebra:

$$\rho(r1, album\_details\_table)$$

$$\rho(r2, album\_details\_table)$$

$$\pi_{r1.album\_name}(r1 \bowtie_{r1.e\_rec\_date<r2.e\_rec\_date} \mathrm{r2})$$

(13 SQL:

```
select ms_name

from musical_scene_in_album as a JOIN music_scene_tedail_table as b on
a.musical_scene_id=b.ms_id

where exists(select ms_name

from musical_scene_in_album as c JOIN music_scene_tedail_table
as d on c.musical_scene_id=d.ms_id

where c.musical_scene_id=a.musical_scene_id and
(c.album_id<>a.album_id

group by ms_name;
```

Relational algebra:

$$\rho(r1, ms\_in\_album)$$

$$\rho(r2, ms\_in\_album)$$

$$\pi_{ms\_name}(\pi_{ms\_id}(\ \sigma_{\ r1.ms\_id=r2.ms\_id\ \wedge\ r1.a\_id\ != \ r2.a\_id}(\mathrm{r1\ X\ r2}))\bowtie \mathrm{music\_scene\_detail\_table})$$

(14 SQL:

```
select name

from person_table as p JOIN music_scene_tedail_table as a on a.t_id=p.id JOIN
musical_scene_in_album as b on a.ms_id=b.musical_scene_id

where not exists(select p1.name

from person_table as p1 JOIN music_scene_tedail_table as
a1 on a1.t_id=p1.id JOIN musical_scene_in_album as b1 on a1.ms_id=b1.musical_scene_id

(where b1.album_id=b.album_id and a1.t_id<>a.t_id

group by name;
```

Relational algebra:

$$\rho(a, music\_scene\_detail\_table)$$

$$\rho(b, music\_scene\_detail\_table)$$

$$\rho(p, person\_table)$$

$$\rho(r1, ((\ a \bowtie_{p.id=a.t\_id} p) \bowtie_{\ b.ms\_id=a.ms\_id} \boldsymbol{b}))$$

$$\pi_{name}((r1)\ -\ (\sigma_{r1.t\_id\ !=\ a.t\_id\ \wedge\ r1.a\_id\ =\ b.a\_id}(\mathrm{r1})\ \mathrm{X}\ (\ \mathrm{a1} \bowtie_{\ b1.ms\_id=a.ms\_id}\ b)))$$

(15 SQL:

```
select p.name

from person_table p JOIN musician_in_ms  m on p.id=m.musican_id JOIN
music_scene_tedail_table  a on m.m_scene_id=a.ms_id

group by m.musican_id

order by count(distinct a.ms_genre)

desc limit 1;
```