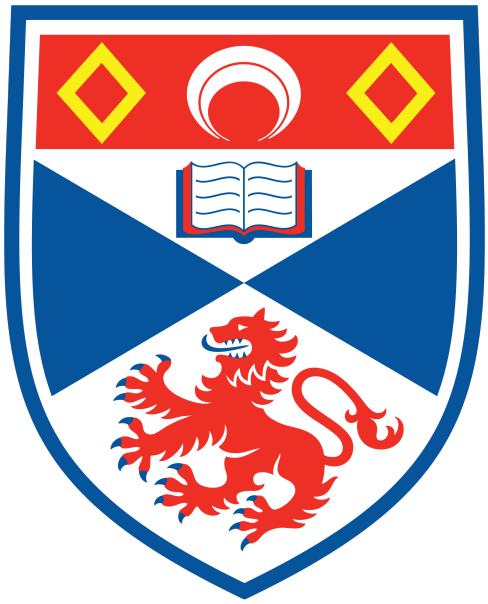
Personal Report

Database management with SQLite

210016568



University of St Andrews

IS5102 - Database Management Systems
Academic Year 2021/2, Semester 1 - Assessment 2

Lecturer: Alexander Konovalov

Word Count: 2103 words

Upload date: November 3, 2021

Task 1: Translation

For Staff entity set, it can be represented by Staff relational schemas.

Staff (staff_id, first_name, middle_name, family_name, email, street, city, postcode)

- staff_id is the primary key of this relational schema
  - I used a fixed length (9) string to illustrate it, the format is "YYMMDDXXX", XXX is the order this staff registered at that day.
  - Because INT(9) is not fixed length, so I chose to use CHAR(9).
- In E-R diagram, name is a composite attribute, so it should be divided into several simple attribute first_name, middle_name, and family_name.
  - For first_name and family_name, it should be added NOT NULL constraint because all people have first name and family name.
- Email is a string not up to 50 length, so I used VARCHAR(50).
- Address is a composite attribute, so it should be divided into street, city, postcode.
  - For the street and city, they are variable length character string, so used VARCHAR(n).
  - Especially, for the postcode, in UK, the max length of it is 10 (because there is a space between two parts e.g., KY16 9LY)
- Because phone is a multivalued attribute, so it should be created as other relational schema i.e., table.

For Phone, it can be seemed a weak entity set (identifying entity set is Staff), so both the primary key of Staff (staff_id) and one or more attribute of Phone should be primary key of this relational schemas, the staff_id should be added foreign key.

Phone (*staff_id*, phone_number, type)

- Because in UK, the length of phone number is 10 (e.g., 7579012049), so I used CHAR(10) in phone_number.
- The attribute type may be "family", "work" or something else, so I used VARCHAR(10) as its domain.

The Staff entity set can be divided into Manager and Driver, so there also need to have Manager and Driver relational schemas.

Manager (*staff_id*, annual_salary) and Driver (*staff_id*, hourly_salary)

- The PK from higher entity set staff_id should be the primary key for both two schemas as well as the foreign key from Staff.
- For annual_salary, I used NUMERIC(8, 2) because generally the annual salary cannot up to million pounds for a manger.
  - It should be added NOT NULL constraint, because every Manager has the salary.
- For hourly_salary, I used DOUBLE, it is suitable for hourly salary.
  - It should be added NOT NULL constraint, because every Driver has the salary.

Here is a one-to-many relation between Manager and Station, so it should add the primary key (staff_id) of the one-side (Manager) to the many-side (Station).

Station (station_name, town, *manager_id*)

- The attribute station_name is the primary key. I gave VARCHAR(50) as its domain.
- Usually, the length of town is not up to 10, so I used VARCHAR(10).
  - It should be added NOT NULL constraint.
- The attribute manager_id is the staff_id from Manager, so here I added a foreign key.

Here is two one-to-many relations between Station and Service, so it should add the primary key (station_name) of the one-side (Station) to the many-side (Service) twice.

Service (service_number, *origin_station_name*, *destination_station_name*)

- According to the scenario, the primary key service_number seems like “99A”, “42”or “X59”, so I gave the VARCHAR(3) as domain.
- Both origin_station_name and destination_station_name are the station_name from Station, so I added foreign key.

ServiceTime is a weak entity, its identifying entity is Service, according to slides of lecture 6, A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set. So, service_number was added as foreign key, start_time is the discriminator. Together two attributes are the primary key of ServiceTime.

ServiceTime (*service_number*, start_time)

- The ServiceTime records the general departure time for a certain service_number, it is unrelated to the date, so I chose to use TIME instead of DATE or TIMESTAMP.

The Stop entity set is used to record the name of different bus stop.

Stop (stop_name)

- The stop_name is primary key, the name of stop seem like “South Street”, So I set the domain as VARCHAR(50).

Here is a many-to-many relation between ServiceTime and Stop. According to the slides of lecture 6, “A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set”. So I create a schema called TimeOfServiceOnStop, service_number and start_time from ServiceTime as well as stop_name from stop are together the primary key, and they are foreign key.

TimeOfServiceOnStop (*service_number*, *start_time*, *stop_name*, arrival_time,

fare_from_origin)

- The arrival_time is a TIME type, the reason is same as the start_time in ServiceTime.
  - It should be added NOT NULL constraint.
- The fare_from_origin is DOUBLE type, because the fare seems like £1.40.
  - It should be added NOT NULL constraint.

Here is a many-to-many relation between Driver and Service, so the staff_id from Driver and service_number are the foreign key and primary key.

Drives (*staff_id*, *service_number*, hours_driven)

- The hours_driven should be INT type, because when the company calculate the salary, the usually use the integer hours.
  - It should be added NOT NULL constraint.

Task 2: SQL Data Definition

To enforce the integrity of database, I used CASCADE. Generally, all foreign keys should be applied “ON DELETE CASCADE ON UPDATE CASCADE”, it can be saw in “buses.sql”.

But **Specifically**, For the table **Station**, the foreign key manager_id should be applied “ON DELETE SET NULL ON UPDATE CASCADE”, because whether the station has a manager or not, the station building itself exists objectively. (The different result of two kind of cascade will be tested following)

I established tables according to these schemas as well as adding some tuples, the results can be seen following:

Staff:

	staff_id	first_name	middle_name	family_name	email	street	city	postcode
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	210901001	Andy	A	Torpy	at1@bus.co.uk	North Haugh	Fife	KY16 9XW
2	210901002	Ben	B	Young	by1@bus.co.uk	Buchanan Gardens	Fife	KY16 9LY
3	210913001	Cindy	NULL	Russell	cr1@bus.co.uk	South Bridge	Edinburgh	EH8 9YL
4	210913002	Kiki	NULL	Chapel	kc1@bus.co.uk	Fountainbridge	Edinburgh	EH3 9QA
5	211002001	Tammy	NULL	Lee	tl1@bus.co.uk	Annfield Rd	Inverness	IV2 3HX
6	211002002	Mendoza	K	Henderson	mh1@bus.co.uk	Howard St	Glasgow	G1 4EE
7	211002003	Ammy	NULL	Taylor	at2@bus.co.uk	Maizey Rd	Swindon	SN25 2RP
8	211002004	Edvin	NULL	Brown	eb1@bus.co.uk	Pall Mall E	London	SW1Y 5AU
9	211002005	Tessa	NULL	Timber	tt1@bus.co.uk	Wapping Ln	London	E1W 2RL
10	210003001	Judy	Q	Kernel	jk1@bus.co.uk	Redriff Rd	London	SE16 7LL

Phone:

	staff_id	phone_number	type
	Filter	Filter	Filter
1	210901001	7579012049	family
2	210901001	7578112323	work
3	210901002	7655434335	NULL
4	210913001	7654565370	NULL
5	210913002	7334537312	NULL
6	211002001	7124645543	NULL
7	211002002	7154546743	NULL
8	211002003	7176844454	NULL
9	211002004	7756558713	NULL
10	211002005	7543761173	NULL
11	210003001	7175788813	NULL

Manager:

	staff_id	annual_salary
	Filter	Filter
1	210901001	34500.5
2	210901002	76550
3	210913001	41230
4	210913002	35000
5	211002001	35050

Driver:

	staff_id	hourly_salary
	Filter	Filter
1	211002002	40.0
2	211002003	50.0
3	211002004	45.0
4	211002005	70.0
5	210003001	60.0

Station:

	station_name	town	manager_id
	Filter	Filter	Filter
1	St Andrews Bus ...	St Andrews	210901001
2	Bogward	St Andrews	210901002
3	Seagate	Dundee	210913001
4	Edinburgh Bus ...	Edinburgh	210913002
5	Union Square	Aberdeen	211002001

Service:

	service_number	origin_station_name	tination_station_na
	Filter	Filter	Filter
1	99A	St Andrews Bus ...	Bogward
2	92B	St Andrews Bus ...	Seagate
3	62	St Andrews Bus ...	Edinburgh Bus ...
4	97A	Union Square	St Andrews Bus ...
5	32R	Union Square	Seagate

ServiceTime:

	service_number	start_time
	Filter	Filter
1	99A	09:00:00
2	99A	15:00:00
3	99A	21:00:00
4	92B	09:30:00
5	92B	21:30:00
6	62	10:30:00
7	62	22:30:00
8	97A	11:00:00
9	97A	23:00:00
10	32R	09:15:00
11	32R	16:00:00

Drives:

	staff_id	service_number	hours_driven
	Filter	Filter	Filter
1	211002002	99A	6
2	211002003	92B	2
3	211002004	62	2
4	211002002	97A	2
5	211002005	32R	2

TimeOfServiceOnStop:

	service_number	start_time	stop_name	arrival_time	fare_from_origin
	Filter	Filter	Filter	Filter	Filter
1	99A	09:00:00	Church Square	09:30:00	1.1
2	99A	09:00:00	South Street	10:00:00	1.3
3	99A	09:00:00	James Robb	11:00:00	1.5
4	99A	15:00:00	Church Square	15:30:00	1.1
5	99A	15:00:00	South Street	16:00:00	1.3
6	99A	15:00:00	James Robb	17:00:00	1.5
7	99A	21:00:00	Church Square	21:30:00	1.1
8	99A	21:00:00	South Street	22:00:00	1.3
9	99A	21:00:00	James Robb	23:00:00	1.5
10	92B	09:30:00	South Street	09:50:00	2.3
11	92B	09:30:00	Horseleys Park	10:30:00	2.8
12	92B	21:30:00	South Street	21:50:00	2.3
13	92B	21:30:00	Horseleys Park	22:30:00	2.8
14	62	10:30:00	Queens Hotel	10:50:00	1.7
15	62	10:30:00	Airlie Place	11:30:00	1.9
16	62	22:30:00	Queens Hotel	22:50:00	1.7
17	62	22:30:00	Airlie Place	23:30:00	1.9
18	97A	11:00:00	South Street	11:25:00	2.7
19	97A	11:00:00	Mcvicars Lane	12:00:00	3.0
20	97A	23:00:00	South Street	23:25:00	2.7
21	97A	23:00:00	Mcvicars Lane	23:59:00	3.0
22	32R	09:15:00	South Street	10:00:00	3.7
23	32R	09:15:00	Patons Lane	10:15:00	4.0
24	32R	16:00:00	South Street	16:45:00	3.7
25	32R	16:00:00	Patons Lane	17:00:00	4.0

When I delete the staff 210901001 in the table Staff, the result is that:

	staff_id	annual_salary
	Filter	Filter
1	210901002	76550
2	210913001	41230
3	210913002	35000
4	211002001	35050

	station_name	town	manager_id
	Filter	Filter	Filter
1	St Andrews Bus ...	St Andrews	*NULL*
2	Bogward	St Andrews	210901002
3	Seagate	Dundee	210913001
4	Edinburgh Bus ...	Edinburgh	210913002
5	Union Square	Aberdeen	211002001

For the Manager table (applied the ON DELETE CASCADE ON UPDATE CASCADE), the tuple which staff_id is 210901001 is deleted.

For the Station table (applied the ON DELETE SET NULL ON UPDATE CASCADE), the tuple which manager_id is 210901001 still exist, but its manager_id become NULL.

Besides, I also try to delete a random tuple in every table in database browser, there is no error. I guess that means I use the “CASCADE” in the right way. But for enforce the integrity, I do not write the delete code in “buses.sql”.

Then I put this sql file into the server to run “sqlite3 –init buses.sql”, there is no error.

```
zg34@trenco:~ $ sqlite3 --init buses.sql
-- Loading resources from buses.sql
service_number
--------------
92B
32R
first_name  middle_name  family_name
----------  -----------  -----------
Mendoza     K            Henderson
Ammy                     Taylor
Edvin                    Brown
first_name  middle_name  family_name
----------  -----------  -----------
Andy        A            Torpy
arrival_time  origin_station_name     destination_station_name  service_number
------------  ----------------------  ------------------------  --------------
09:50:00      St Andrews Bus Station  Seagate                   92B
10:00:00      St Andrews Bus Station  Bogward                   99A
10:00:00      Union Square            Seagate                   32R
11:25:00      Union Square            St Andrews Bus Station    97A
16:00:00      St Andrews Bus Station  Bogward                   99A
16:45:00      Union Square            Seagate                   32R
staff_id    first_name  middle_name  family_name  email          type        phone_number  street      city        postcode
----------  ----------  -----------  -----------  -------------  ----------  ------------  ----------  ----------  ----------
211002002   Mendoza     K            Henderson    mh1@bus.co.uk              7154546743    Howard St   Glasgow     G1 4EE
211002003   Ammy                     Taylor       at2@bus.co.uk              7176844454    Maizey Rd   Swindon     SN25 2RP
211002004   Edvin                    Brown        eb1@bus.co.uk              7756558713    Pall Mall   London      SW1Y 5AU
first_name  middle_name  family_name  station_name  annual_salary
----------  -----------  -----------  ------------  -------------
Ben         B            Young        Bogward       76550
Cindy                    Russell      Seagate       41230
Tammy                    Lee          Union Square  35050
Kiki                     Chapel       Edinburgh Bu  35000
Andy        A            Torpy        St Andrews B  34500.5
first_name  middle_name  family_name  daily_salary
----------  -----------  -----------  ------------
Mendoza     K            Henderson    320.0
Tessa                    Timber       140.0
Ammy                     Taylor       100.0
Edvin                    Brown        90.0
Judy        Q            Kernel
SQLite version 3.26.0 2018-12-01 12:34:55
Enter ".help" for usage hints.
sqlite>
```

Task 3: SQL Data Manipulation

Queries:

1. List all services which have Seagate Bus Station in Dundee as their destination.

Code:

```
SELECT
	service_number
FROM
	Service
WHERE
	destination_station_name = 'Seagate';
```

Result:

	service_number
1	92B
2	32R

2. List the names of all drivers of services which have St Andrews Bus Station in St Andrews as their origin or destination, in decreasing order of total hours driven.

Code:

```
SELECT
	a.first_name,
	a.middle_name,
	a.family_name
FROM
	Staff a
	LEFT JOIN Drives d ON d.staff_id = a.staff_id
	LEFT JOIN Service s ON s.service_number = d.service_number
WHERE
	origin_station_name = 'St Andrews Bus Station'
	or destination_station_name = 'St Andrews Bus Station'
GROUP BY
	a.first_name,
	a.middle_name,
	a.family_name
ORDER BY
	SUM(d.hours_driven) DESC;
```

Result:

	first_name	middle_name	family_name
1	Mendoza	K	Henderson
2	Ammy	*NULL*	Taylor
3	Edvin	*NULL*	Brown

3. List the manager of the most connected station, measured by the number of services which have that station as their origin or destination.

Code:

```
SELECT
    s2.first_name,
    s2.middle_name,
    s2.family_name
FROM
    (
        SELECT
            origin_station_name station_name,
            COUNT(origin_station_name) AS connected_num
        FROM
            Service
        GROUP BY
            origin_station_name
        UNION
        all
        SELECT
            destination_station_name station_name,
            COUNT(destination_station_name) AS connected_num
        FROM
            Service
        GROUP BY
            origin_station_name
    )   a
    LEFT JOIN Station s ON s.station_name = a.station_name
    LEFT JOIN Staff s2 ON s2.staff_id = s.manager_id
GROUP BY
    s2.first_name,
    s2.middle_name,
    s2.family_name
ORDER BY
    sum(a.connected_num) DESC
LIMIT
    1;
```

Result:

	first_name	middle_name	family_name
1	Andy	A	Torpy

4. For the bus stop ”South Street, St Andrews” list in the chronological order arrival times at this stop, origins, destinations, and service numbers of all bus services passing this stop between 8 am and 6 pm.

Code:

```
SELECT
	t.arrival_time,
	s.origin_station_name,
	s.destination_station_name,
	t.service_number
FROM
	TimeOfServiceOnStop t
	LEFT JOIN Service s ON s.service_number = t.service_number
WHERE
	t.arrival_time BETWEEN '08:00:00'
	AND '18:00:00'
	AND t.stop_name = 'South Street'
ORDER BY
	t.arrival_time;
```

Result:

	arrival_time	origin_station_name	destination_station_name	service_number
1	09:50:00	St Andrews Bus Station	Seagate	92B
2	10:00:00	St Andrews Bus Station	Bogward	99A
3	10:00:00	Union Square	Seagate	32R
4	11:25:00	Union Square	St Andrews Bus Station	97A
5	16:00:00	St Andrews Bus Station	Bogward	99A
6	16:45:00	Union Square	Seagate	32R

5. List the contact details of all drivers of services which have St Andrews Bus Station in St Andrews as their origin or destination.

Code:

```
SELECT DISTINCT
    s.staff_id,
    s.first_name,
    s.middle_name,
    s.family_name,
    s.email,
    s.type,
    s.phone_number,
    s.street,
    s.city,
    s.postcode
FROM
    (
        Driver
        LEFT JOIN Staff ON Driver.staff_id = Staff.staff_id
        LEFT JOIN Phone ON Driver.staff_id = Phone.staff_id
    ) s,
    (
        Drives
        LEFT JOIN Service ON Drives.service_number = Service.service_number
    ) d
WHERE
    s.staff_id = d.staff_id
    and (
        d.origin_station_name = 'St Andrews Bus Station'
        or destination_station_name = 'St Andrews Bus Station'
    );
```

Result:

	staff_id	first_name	middle_name	family_name	email	type	phone_number	street	city	postcode
1	211002002	Mendoza	K	Henderson	mh1@bus.co.uk	NULL	7154546743	Howard St	Glasgow	G1 4EE
2	211002003	Ammy	NULL	Taylor	at2@bus.co.uk	NULL	7176844454	Maizey Rd	Swindon	SN25 2RP
3	211002004	Edvin	NULL	Brown	eb1@bus.co.uk	NULL	7756558713	Pall Mall E	London	SW1Y 5AU

6. List the name, station and annual salary of managers, in decreasing order of annual salary

Code:

```
SELECT
    first_name,
    middle_name,
    family_name,
    station_name,
    annual_salary
FROM
    Manager NATURAL
    JOIN Staff
    LEFT JOIN Station ON Manager.staff_id = Station.manager_id
ORDER BY
    annual_salary DESC;
```

Result:

	first_name	middle_name	family_name	station_name	annual_salary
1	Ben	B	Young	Bogward	76550
2	Cindy	*NULL*	Russell	Seagate	41230
3	Tammy	*NULL*	Lee	Union Square	35050
4	Kiki	*NULL*	Chapel	Edinburgh Bus ...	35000
5	Andy	A	Torpy	St Andrews Bus ...	34500.5

7. List the name, daily salary of drivers, in decreasing order of daily salary

Code:

```
SELECT
    first_name,
    middle_name,
    family_name,
    (sum(d.hours_driven) * d.hourly_salary) AS daily_salary
FROM
    (
        Driver
        LEFT JOIN Staff ON Driver.staff_id = Staff.staff_id
        LEFT JOIN Drives ON Drives.staff_id = Driver.staff_id
        LEFT JOIN Service ON Service.service_number = Drives.service_number
    ) d
GROUP BY
    first_name,
    middle_name,
    family_name
ORDER BY
    daily_salary DESC;
```

Result:

	first_name	middle_name	family_name	daily_salary
1	Mendoza	K	Henderson	320.0
2	Tessa	*NULL*	Timber	140.0
3	Ammy	*NULL*	Taylor	100.0
4	Edvin	*NULL*	Brown	90.0
5	Judy	Q	Kernel	*NULL*

(Judy do not drive any service, can be seen in table **Drives**)

Views:

1. Create a view which contains the station name, the service number of buses which this station is origin or destination, full name and contact details of its manager.

   Statement:

   I want to use this

   Code:

```
CREATE VIEW station_manager_service AS
SELECT
    station_name,
    service_numbers,
    manager_name,
    email,
    group_concat(type_number) AS phone_numbers,
    street,
    city,
    postcode
FROM
    (
        SELECT
            *,
            group_concat(service_number) AS service_numbers,
            first_name || ' ' || ifnull(middle_name, '') || ' ' || family_name AS manager_name,
            ifnull(type, 'general') || ':' || phone_number AS type_number
        FROM
            Station
            LEFT JOIN Manager ON Station.manager_id = Manager.staff_id
            LEFT JOIN Staff ON Station.manager_id = Staff.staff_id
            LEFT JOIN Phone ON Staff.staff_id = Phone.staff_id
            LEFT JOIN Service ON Station.station_name = Service.origin_station_name
            or Station.station_name = Service.destination_station_name
        GROUP BY
            Phone.phone_number
    )
GROUP BY
    station_name;
```

   Result:

	station_name	service_numbers	manager_name	email	phone_numbers	street	city	postcode
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Bogward	99A	Ben B Young	by1@bus.co.uk	general:7655434335	Buchanan Gardens	Fife	KY16 9LY
2	Edinburgh Bus Station	62	Kiki Chapel	kc1@bus.co.uk	general:7334537312	Fountainbridge	Edinburgh	EH3 9QA
3	Seagate	92B,32R	Cindy Russell	cr1@bus.co.uk	general:7654565370	South Bridge	Edinburgh	EH8 9YL
4	St Andrews Bus Station	99A,92B,62,97A	Andy A Torpy	at1@bus.co.uk	work:7578112323,family:7579012049	North Haugh	Fife	KY16 9XW
5	Union Square	97A,32R	Tammy Lee	tl1@bus.co.uk	general:7124645543	Annfield Rd	Inverness	IV2 3HX

2. Create a view which cotains the name of driver, stops this driver passed.

Code:

```
CREATE VIEW stop_service_time AS
SELECT
    first_name || ' ' || ifnull(middle_name, '') || ' ' || family_name AS driver_name,
    group_concat(stop_name) AS stops
FROM
(
        SELECT DISTINCT
            first_name,
            middle_name,
            family_name,
            stop_name
        FROM
            Driver
            NATURAL JOIN Drives
            NATURAL JOIN TimeOfServiceOnStop
            NATURAL JOIN Staff
    )
GROUP BY
    driver_name;
```

Result:

	driver_name	stops
	Filter	Filter
1	Ammy Taylor	Horseleys Park,South Street
2	Edvin Brown	Airlie Place,Queens Hotel
3	Mendoza K Henderson	Mcvicars Lane,South Street,Church Square,James Robb
4	Tessa Timber	Patons Lane,South Street

Task 4: Reflection

Problem I encountered and solution

In this assignment, the first two tasks went well. Only when I translate the Service entity set, at the beginning I was confused how to deal with the origin and destination, because they are all from the same entity set. By reviewing slides and recordings, I found that I need to add the foreign key twice although for those two attributes.

When I tried to list the hours_driven of all the drivers in Task 3, I found that after using LEFT JOIN, there were two drivers, their hours_driven is NULL, which made me very confused, because I only let one driver do not join service. After repeated attempts I found that I had mistakenly written the staff_id of a manager in the Drives table.

But I clearly used a foreign key constraint to restrict the value of staff_id, and by looking at the table structure further, I found that I had added a foreign key from the Staff table instead of the Driver table when I created the Drives table. After modifying it, everything is well done.

In creating views, I used "||" to combine the first name, middle name, and family name. (first_name || middle_name || family_name) But a lot of combined result is NULL. I was really confused about that. After a long while, I found that, some staff may do not have middle name, and if I combine a string with NULL by using "||", the result is NULL. Then I use the function "ifnull" to determine whether it is NULL, and replace the NULL by empty string, such like ifnull(middle_name, '').

The good job I did

I want to combine a series of CHAR string, and where between every two strings need a comma, so I searched it on Google, it showed that here is a function called "group_concat", it can be used with GROUP BY, then I learnt its usage. Finally I used it to achieve my request (ca be seen in both two views).

Summary

I practiced how to translate the E-R diagram to relational schemas as well as creating tables according to them. Besides, I also searched some new functions on internet to achieve excepted result.