SQL update data query

USER:

Experts in the authentication of Qiaopi

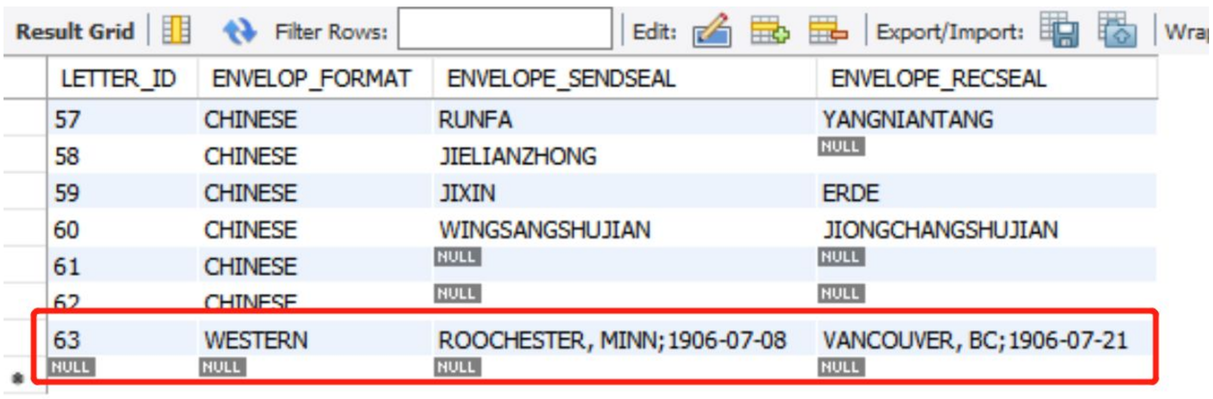
PURPOSE:

This query enables the experts to change the existing records that contain errors. For example, they can change the attributes in ENVELOP by using the following query.

SQL STATEMENT:

```
UPDATE ENVELOPE
SET ENVELOPE_SENDSEAL = 'ROOCHESTER, MINN;1906-07-08',
ENVELOPE_RECSEAL = 'VANCOUVER, BC;1906-07-21'
WHERE LETTER_ID = 63
```

RESULT SET:



LETTER_ID	ENVELOP_FORMAT	ENVELOPE_SENDSEAL	ENVELOPE_RECSEAL
57	CHINESE	RUNFA	YANGNIANTANG
58	CHINESE	JIELIANZHONG	NULL
59	CHINESE	JIXIN	ERDE
60	CHINESE	WINGSANGSHUJIAN	JIONGCHANGSHUJIAN
61	CHINESE	NULL	NULL
62	CHINESE	NULL	NULL
63	WESTERN	ROOCHESTER, MINN;1906-07-08	VANCOUVER, BC;1906-07-21
NULL	NULL	NULL	NULL

Other useful queries:

1.

USER:

Researchers who are interested in the subjects of Qiaopi

PURPOSE:

This query enables the users to identify the popular themes in Qiaopi

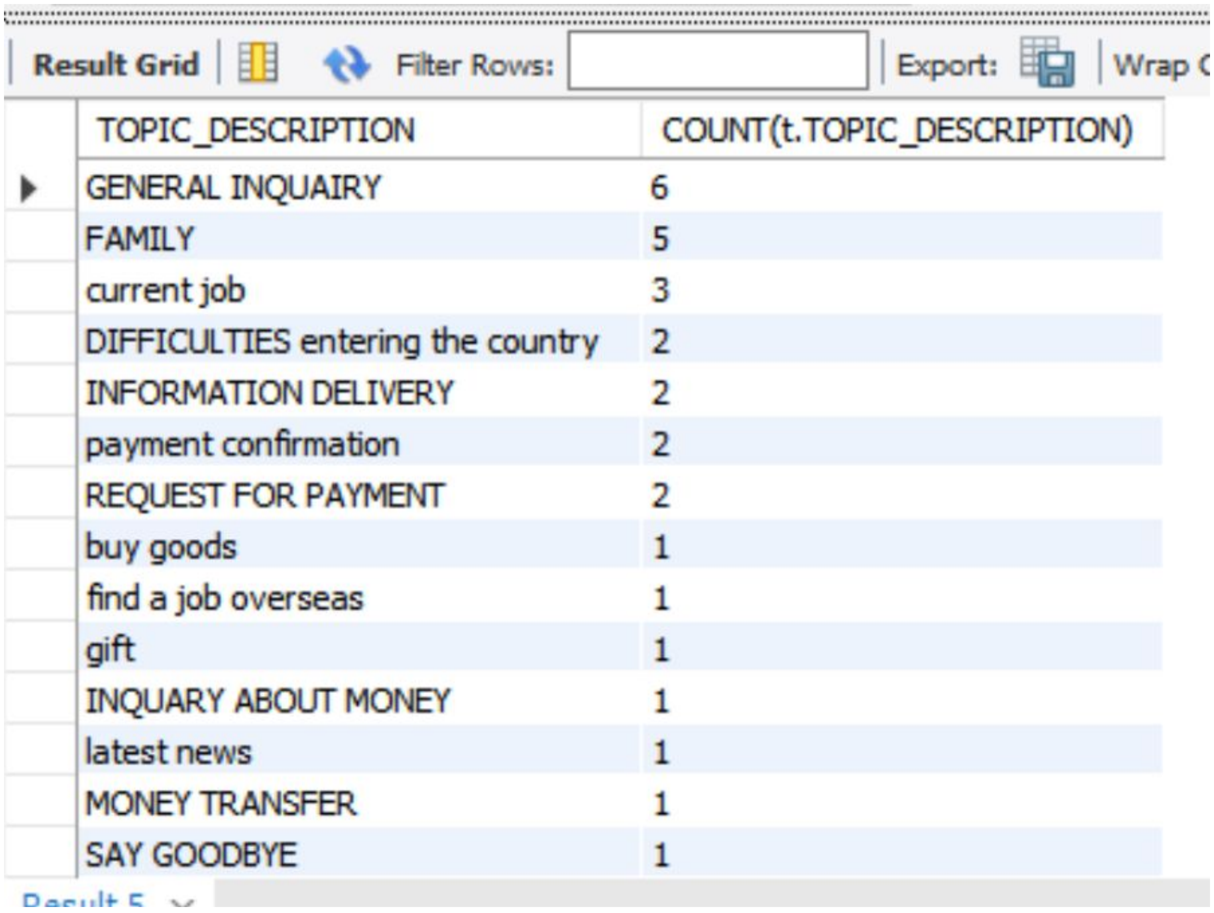
REQUIRED LAYOUT:

display the topics of each letter and the times they appear in letters, and list all the topics in descending order

SQL STATEMENT:

```
SELECT t.TOPIC_DESCRIPTION, COUNT(t.TOPIC_DESCRIPTION)
FROM EXPRESS e INNER JOIN TOPIC t ON t.TOPIC_ID = e.TOPIC_ID
GROUP BY t.TOPIC_DESCRIPTION
ORDER BY COUNT(t.TOPIC_DESCRIPTION) DESC
```

RESULT SET:



TOPIC_DESCRIPTION	COUNT(t.TOPIC_DESCRIPTION)
GENERAL INQUAIRY	6
FAMILY	5
current job	3
DIFFICULTIES entering the country	2
INFORMATION DELIVERY	2
payment confirmation	2
REQUEST FOR PAYMENT	2
buy goods	1
find a job overseas	1
gift	1
INQUARY ABOUT MONEY	1
latest news	1
MONEY TRANSFER	1
SAY GOODBYE	1

2.

USER:

Researchers who are interested in doing research about the Qiaopi NOT sent from BC and written in 1902

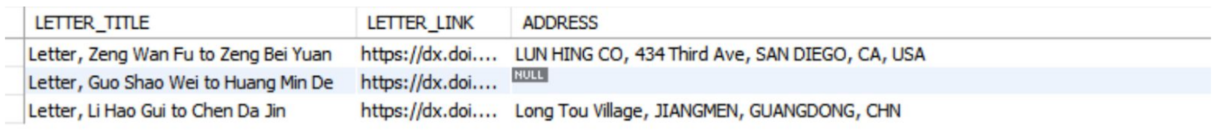
PURPOSE:

find all the Qiaopi NOT sent from BC in 1902

REQUIRED LAYOUT:
list the letter title and link and their location address

SQL STATEMENT:
SELECT l.LETTER_TITLE, l.LETTER_LINK, CONCAT(loc.LOCATION_DESCRIPTION, ', ', loc.LOCATION_CITYORCOUNTY,', ',loc.LOCATION_PROVINCE, ', ', loc.LOCATION_COUNTRY) AS ADDRESS
FROM LETTER l INNER JOIN LOCATION loc ON l.SENDERLOC_ID = loc.LOCATION_ID
WHERE ((loc.LOCATION_PROVINCE NOT LIKE 'BC') OR (loc.LOCATION_PROVINCE IS NULL)) AND (l.LETTER_CREATEDYEAR=1902)

RESULT SET:



LETTER_TITLE	LETTER_LINK	ADDRESS
Letter, Zeng Wan Fu to Zeng Bei Yuan	https://dx.doi....	LUN HING CO, 434 Third Ave, SAN DIEGO, CA, USA
Letter, Guo Shao Wei to Huang Min De	https://dx.doi....	NULL
Letter, Li Hao Gui to Chen Da Jin	https://dx.doi....	Long Tou Village, JIANGMEN, GUANGDONG, CHN

3.
USER:
Researchers who are interested in pairing the original letters with their responses

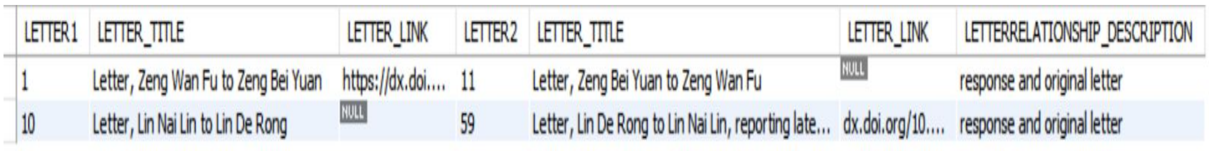
PURPOSE:
find all the letters that have responses

REQUIRED LAYOUT:
find all the letters that have responses, list their ID, TITLE and LINK and their relationship with each other

SQL STATEMENT:
SELECT r.LETTER1_ID AS LETTER1, l1.LETTER_TITLE, l1.LETTER_LINK, r.LETTER2_ID AS LETTER2, l2.LETTER_TITLE, l2.LETTER_LINK, r.LETTERRELATIONSHIP_DESCRIPTION

FROM LETTERRELATIONSHIP r INNER JOIN LETTER l1 ON (l1.LETTER_ID =
r.LETTER1_ID)
INNER JOIN LETTER l2 ON (l2.LETTER_ID = r.LETTER2_ID)

RESULT SET:



LETTER1	LETTER_TITLE	LETTER_LINK	LETTER2	LETTER_TITLE	LETTER_LINK	LETTERRELATIONSHIP_DESCRIPTION
1	Letter, Zeng Wan Fu to Zeng Bei Yuan	https://dx.doi....	11	Letter, Zeng Bei Yuan to Zeng Wan Fu	NULL	response and original letter
10	Letter, Lin Nai Lin to Lin De Rong	NULL	59	Letter, Lin De Rong to Lin Nai Lin, reporting late...	dx.doi.org/10....	response and original letter

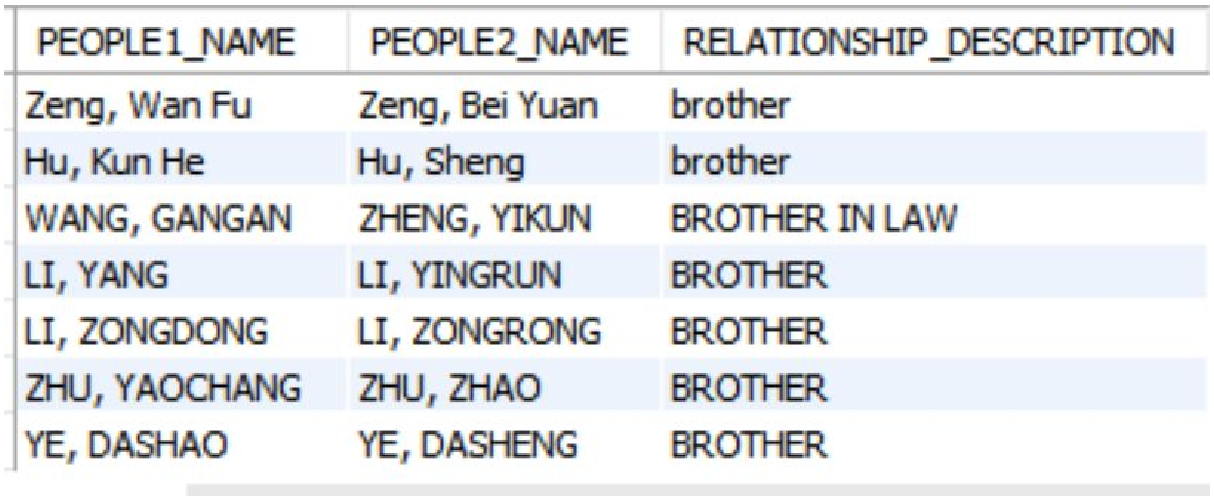
4.
USER:
Researchers who are interested in the personal network of the authors

PURPOSE:
find all the names of senders and receivers who are addressed as “brother” or anything like “brother”

REQUIRED LAYOUT:
full name of both the senders and receivers and their relationship

SQL STATEMENT:
SELECT CONCAT(p1.PEOPLE_LNAME, ', ',p1.PEOPLE_FNAME) AS PEOPLE1_NAME,
CONCAT(p2.PEOPLE_LNAME, ', ',p2.PEOPLE_FNAME) AS PEOPLE2_NAME,
r.RELATIONSHIP_DESCRIPTION
FROM RELATIONSHIP r INNER JOIN PEOPLE p1 ON (p1.PEOPLE_ID = r.PEOPLE1_ID)
INNER JOIN PEOPLE p2 ON (p2.PEOPLE_ID = r.PEOPLE2_ID)
WHERE r.RELATIONSHIP_DESCRIPTION LIKE '%BROTHER%'

RESULT SET:



PEOPLE1_NAME	PEOPLE2_NAME	RELATIONSHIP_DESCRIPTION
Zeng, Wan Fu	Zeng, Bei Yuan	brother
Hu, Kun He	Hu, Sheng	brother
WANG, GANGAN	ZHENG, YIKUN	BROTHER IN LAW
LI, YANG	LI, YINGRUN	BROTHER
LI, ZONGDONG	LI, ZONGRONG	BROTHER
ZHU, YAOCHANG	ZHU, ZHAO	BROTHER
YE, DASHAO	YE, DASHENG	BROTHER

INDEX QUERY #1:

PURPOSE:
Although we do not currently have a very large number of data, as this database may get much larger later. We can create an index to increase the efficiency of queries. In letter queries, the senders and receivers are very important and the quantity of people can be very large. So our solution is to add an index to the first name and last name, so that researchers can finish these queries through the names much faster.

SQL STATEMENT:

```
CREATE INDEX idx_firstname
ON people (PEOPLE_FNAME);

CREATE INDEX idx_lastname
ON people (PEOPLE_LNAME);
```

INDEX QUERY #2:
PURPOSE:
Queries about topic descriptions will be commonly executed by users. To increase the efficiency of queries, we created a unique index of all the topic descriptions in our database.
SQL STATEMENT:

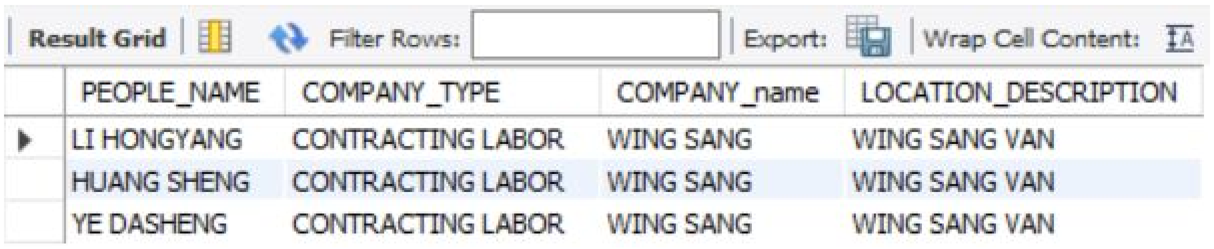
```
CREATE UNIQUE INDEX SUBJECTS
ON TOPIC (TOPIC_DESCRIPTION);
```

VIEW QUERIES:

In our database, there is not any private information related to security, so we are trying to create two views that can facilitate the query process.

QUERY #1
PURPOSE:
This view can give simpler access to the employment details of the people. The result set needs to include information about people, company and location, which is both comprehensive and concise. It will help researchers or people who are interested in this area quickly find the job information they want.

SQL STATEMENT:
```
create view jobsdetail as
SELECT  CONCAT(pl.PEOPLE_LNAME, ' ',pl.PEOPLE_FNAME) AS PEOPLE_NAME,
            cp.COMPANY_TYPE,
      cp.COMPANY_name,
      lc.LOCATION_DESCRIPTION
FROM employment ep
   INNER JOIN PEOPLE pl ON (pl.PEOPLE_ID = ep.PEOPLE_ID)
   INNER JOIN company cp ON (cp.company_ID = ep.company_ID)
         inner join location lc on cp.LOCATION_ID = lc.location_id
```



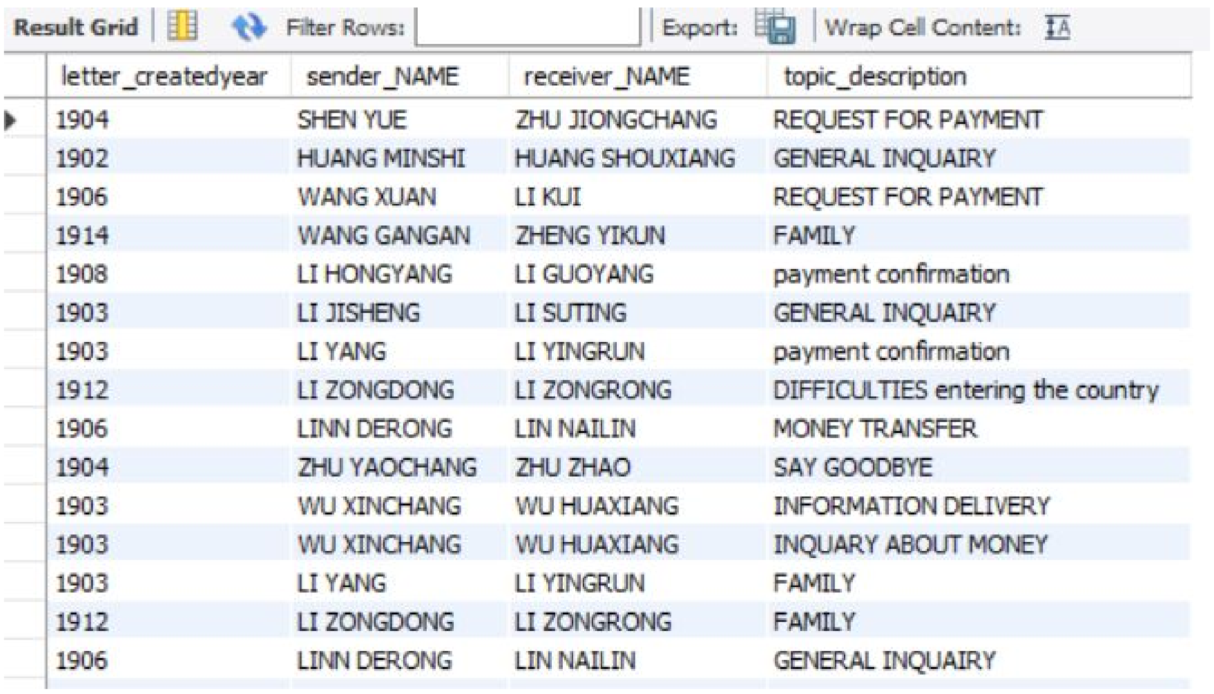
PEOPLE_NAME	COMPANY_TYPE	COMPANY_name	LOCATION_DESCRIPTION
LI HONGYANG	CONTRACTING LABOR	WING SANG	WING SANG VAN
HUANG SHENG	CONTRACTING LABOR	WING SANG	WING SANG VAN
YE DASHENG	CONTRACTING LABOR	WING SANG	WING SANG VAN

QUERY #2
PURPOSE:
This view can give a simpler access to the letters' important information, which includes people, letter and topic. We tried to keep the information both comprehensive and concise. It

will help researchers or people who are interested in this area quickly have a general idea about the information about the letters.

SQL STATEMENT:

```
create view letterbrief as
SELECT  l.letter_createdyear, CONCAT(pls.PEOPLE_LNAME, ' ',pls.PEOPLE_FNAME) AS
sender_NAME,
      CONCAT(plr.PEOPLE_LNAME, ' ',plr.PEOPLE_FNAME) AS receiver_NAME,
            tp.topic_description
FROM express ex
   INNER JOIN letter l ON (l.letter_id = ex.letter_ID)
   INNER JOIN topic tp ON (ex.topic_ID = tp.TOPIC_ID)
         inner join people pls on pls.PEOPLE_ID = l.SENDER_ID
   inner join people plr on plr.PEOPLE_ID = l.RECEIVER_ID;
```



letter_createdyear	sender_NAME	receiver_NAME	topic_description
1904	SHEN YUE	ZHU JIONGCHANG	REQUEST FOR PAYMENT
1902	HUANG MINSHI	HUANG SHOUXIANG	GENERAL INQUAIRY
1906	WANG XUAN	LI KUI	REQUEST FOR PAYMENT
1914	WANG GANGAN	ZHENG YIKUN	FAMILY
1908	LI HONGYANG	LI GUOYANG	payment confirmation
1903	LI JISHENG	LI SUTING	GENERAL INQUAIRY
1903	LI YANG	LI YINGRUN	payment confirmation
1912	LI ZONGDONG	LI ZONGRONG	DIFFICULTIES entering the country
1906	LINN DERONG	LIN NAILIN	MONEY TRANSFER
1904	ZHU YAOCHANG	ZHU ZHAO	SAY GOODBYE
1903	WU XINCHANG	WU HUAXIANG	INFORMATION DELIVERY
1903	WU XINCHANG	WU HUAXIANG	INQUARY ABOUT MONEY
1903	LI YANG	LI YINGRUN	FAMILY
1912	LI ZONGDONG	LI ZONGRONG	FAMILY
1906	LINN DERONG	LIN NAILIN	GENERAL INQUAIRY

STORED PROCEDURES:

PROCEDURE #1

USERS:

end-users who do not have access to all the tables but need to look up information about the letters written in a certain year

PURPOSE:

To enable users to retrieve letter information by their created years; display the most important information in the table of LETTER, including letter_title, letter_createdyear and letter_link, through which they can view the content of letters

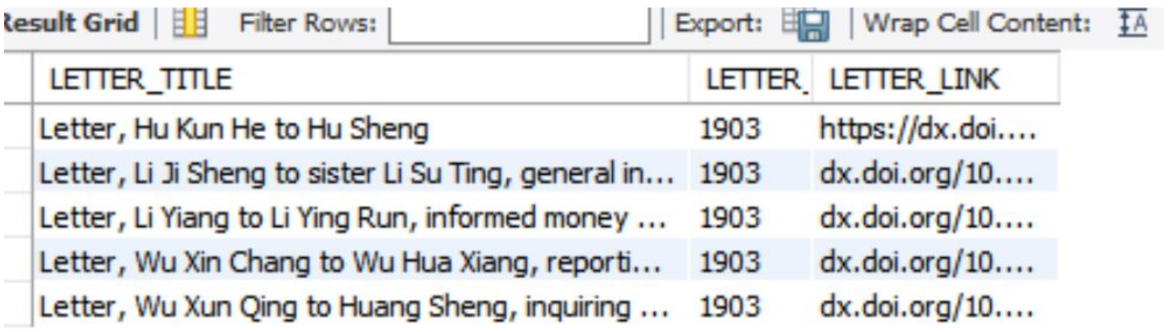
SQL STATEMENT:

```
DELIMITER $$
CREATE PROCEDURE GetLetterByCreatedtime (
IN Createdtime int)
BEGIN
SELECT LETTER_TITLE, LETTER_CREATEDYEAR, LETTER_LINK
FROM LETTER
WHERE Createdtime = LETTER_CREATEDYEAR;
END $$
DELIMITER ;
```

For example, users may want to look up information about letters written in the year of 1903. Then they can execute the following statement:

CALL GetLetterByCreatedtime (1903);

They can get the following RESULT SET:



LETTER_TITLE	LETTER_	LETTER_LINK
Letter, Hu Kun He to Hu Sheng	1903	https://dx.doi....
Letter, Li Ji Sheng to sister Li Su Ting, general in...	1903	dx.doi.org/10....
Letter, Li Yiang to Li Ying Run, informed money ...	1903	dx.doi.org/10....
Letter, Wu Xin Chang to Wu Hua Xiang, reporti...	1903	dx.doi.org/10....
Letter, Wu Xun Qing to Huang Sheng, inquiring ...	1903	dx.doi.org/10....

PROCEDURE #2

USERS:

end-users want to know search the information about people through the company they worked in.

PURPOSE:
reduce the complexity of retrieving information about people through the name of the company they were employed. It also allows searching with incomplete information about company name.

SQL STATEMENT:

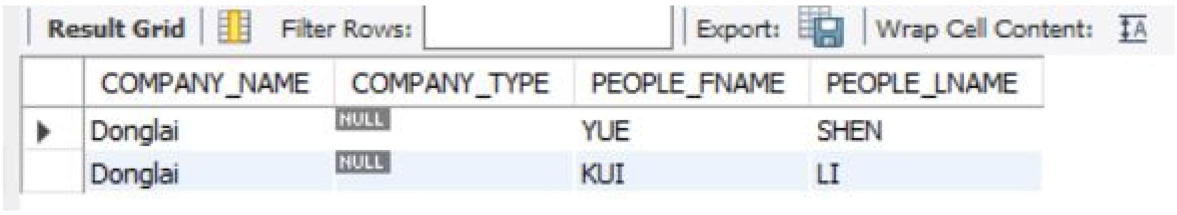
```
DROP PROCEDURE GETCOMPANY;
DELIMITER //
CREATE procedure GETCOMPANY(
        IN COMPANYNAME VARCHAR(32)
)
BEGIN
        SELECT CP.COMPANY_NAME, CP.COMPANY_TYPE, PL.PEOPLE_FNAME,
PL.PEOPLE_LNAME
   FROM COMPANY CP
                INNER JOIN employment EMP ON EMP.COMPANY_ID = CP.COMPANY_ID
      INNER JOIN PEOPLE PL ON PL.PEOPLE_ID = EMP.PEOPLE_ID
   WHERE COMPANY_NAME LIKE COMPANYNAME;
END //

DELIMITER ;
```

For example, users may want to look up information about workers in Donglai company. Then they can execute the following statement:

```
CALL GETCOMPANY('%Donglai%');
```

They can get the following RESULT SET:



COMPANY_NAME	COMPANY_TYPE	PEOPLE_FNAME	PEOPLE_LNAME
Donglai	NULL	YUE	SHEN
Donglai	NULL	KUI	LI