NAME :: IBRAR BABAR

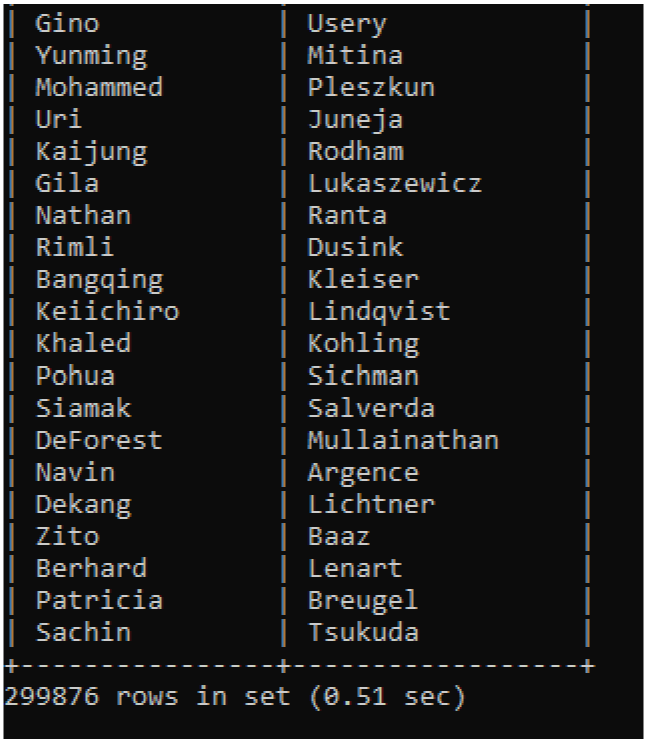
ROLL NO. 19P-0104

DATABASE LAB

ASSIGNMENT#1

TASK#1

```
select First_name,Last_name
   from employees_V2
    where last_name like '%a%' or
    last_name like  '%e%' or
    last_name like '%i%' or
    last_name like  '%o%' or
    last_name like  '%u%';
```

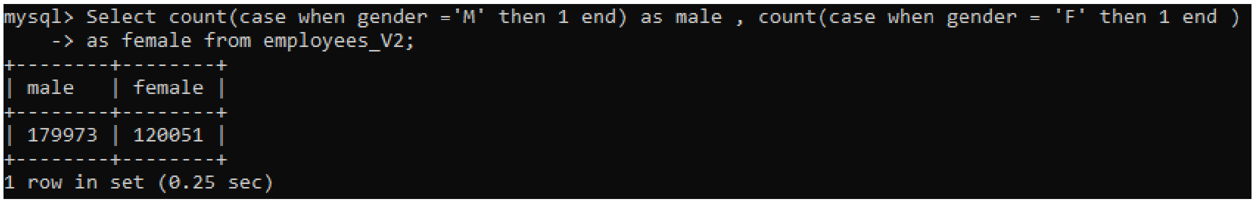


```
| Gino           | Usery            |
| Yunming        | Mitina           |
| Mohammed       | Pleszkun         |
| Uri            | Juneja           |
| Kaijung        | Rodham           |
| Gila           | Lukaszewicz      |
| Nathan         | Ranta            |
| Rimli          | Dusink           |
| Bangqing       | Kleiser          |
| Keiichiro      | Lindqvist        |
| Khaled         | Kohling          |
| Pohua          | Sichman          |
| Siamak         | Salverda         |
| DeForest       | Mullainathan     |
| Navin          | Argence          |
| Dekang         | Lichtner         |
| Zito           | Baaz             |
| Berhard        | Lenart           |
| Patricia       | Breugel          |
| Sachin         | Tsukuda          |
+----------------+------------------+
299876 rows in set (0.51 sec)
```

TASK#2

Select count(case when gender ='M' then 1 end) as male , count(case when gender = 'F' then 1 end)

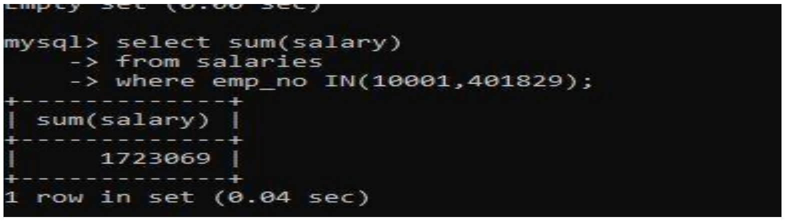
as female from employees;



```
mysql> Select count(case when gender ='M' then 1 end) as male , count(case when gender = 'F' then 1 end )
    -> as female from employees_V2;
+--------+--------+
| male   | female |
+--------+--------+
| 179973 | 120051 |
+--------+--------+
1 row in set (0.25 sec)
```

TASK#3

Select sum (salary) from salaries where emp_no IN (10001,401829);

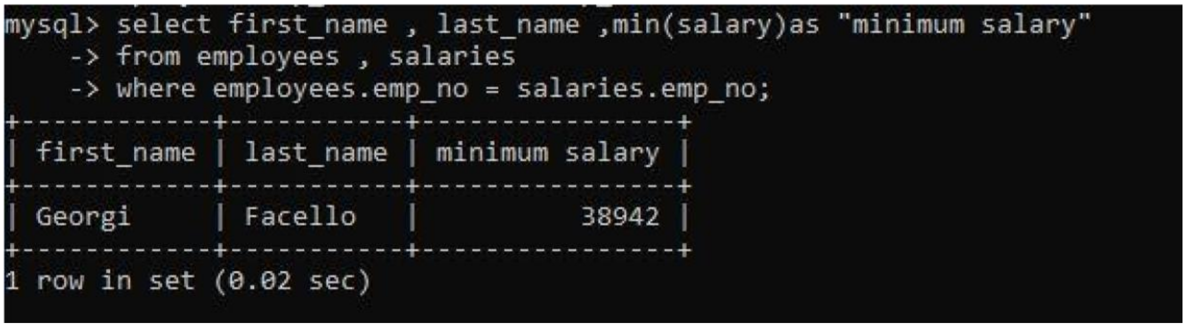


```
Empty set (0.00 sec)

mysql> select sum(salary)
    -> from salaries
    -> where emp_no IN(10001,401829);
+-------------+
| sum(salary) |
+-------------+
|     1723069 |
+-------------+
1 row in set (0.04 sec)
```

TASK#4

Select first_name , last_name , min(salary) as “minimum salary” from employees , salaries Where employees.emp_no = salaries.emp_no;

Output:



```
mysql> select first_name , last_name ,min(salary)as "minimum salary"
    -> from employees , salaries
    -> where employees.emp_no = salaries.emp_no;
+------------+-----------+----------------+
| first_name | last_name | minimum salary |
+------------+-----------+----------------+
| Georgi     | Facello   |          38942 |
+------------+-----------+----------------+
1 row in set (0.02 sec)
```

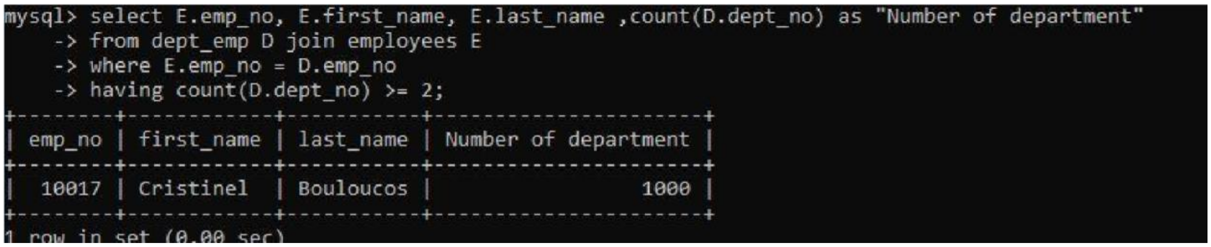
TASK#5

Select E.emp_no , E.first_name ,E.last_name ,count(D.dept_no) as “Number of department”

From dept_emp D join employees E

Where E.emp_no = D.emp_no

Having count(D.dept_no) >= 2;



```
mysql> select E.emp_no, E.first_name, E.last_name ,count(D.dept_no) as "Number of department"
    -> from dept_emp D join employees E
    -> where E.emp_no = D.emp_no
    -> having count(D.dept_no) >= 2;
+--------+------------+-----------+----------------------+
| emp_no | first_name | last_name | Number of department |
+--------+------------+-----------+----------------------+
|  10017 | Cristinel  | Bouloucos |                 1000 |
+--------+------------+-----------+----------------------+
1 row in set (0.00 sec)
```

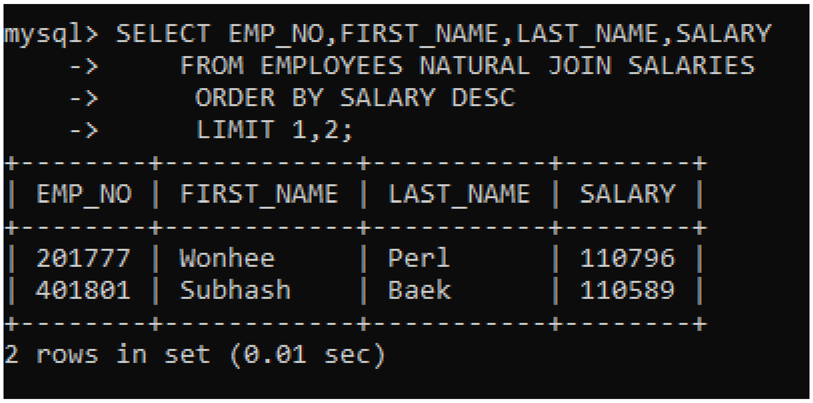
TASK#6

SELECT EMP_NO,FIRST_NAME,LAST_NAME,SALARY

FROM EMPLOYEES NATURAL JOIN SALARIES

ORDER BY SALARY DESC

LIMIT 1,2;



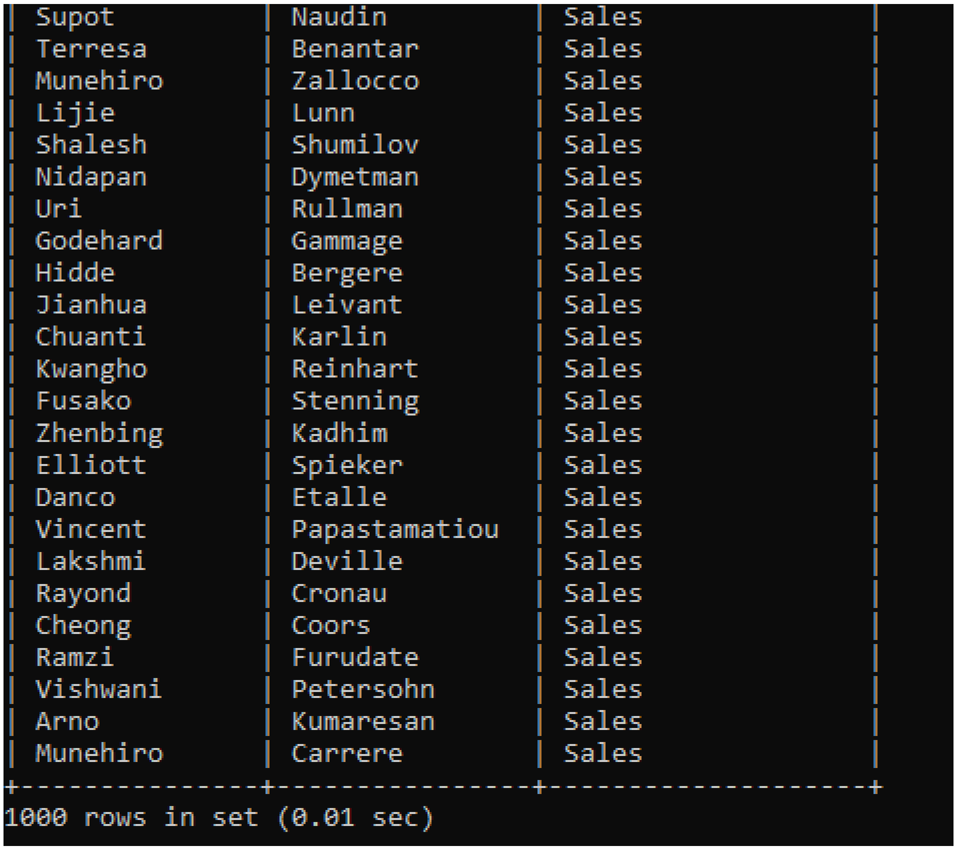
```
mysql> SELECT EMP_NO,FIRST_NAME,LAST_NAME,SALARY
    ->     FROM EMPLOYEES NATURAL JOIN SALARIES
    ->      ORDER BY SALARY DESC
    ->      LIMIT 1,2;
+--------+------------+-----------+--------+
| EMP_NO | FIRST_NAME | LAST_NAME | SALARY |
+--------+------------+-----------+--------+
| 201777 | Wonhee     | Perl      | 110796 |
| 401801 | Subhash    | Baek      | 110589 |
+--------+------------+-----------+--------+
2 rows in set (0.01 sec)
```

TASK#7

SELECT FIRST_NAME,LAST_NAME,DEPT_NAME

FROM EMPLOYEES JOIN DEPT_EMP USING (EMP_NO)

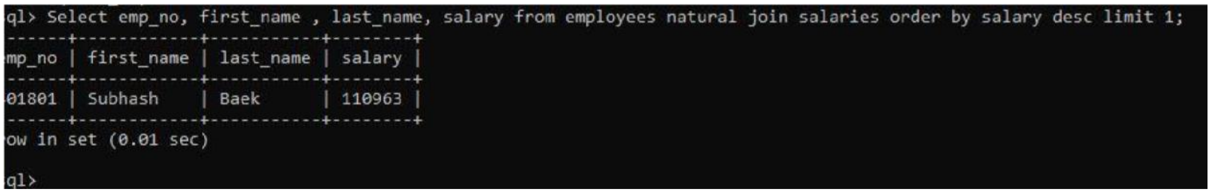
JOIN DEPARTMENTS USING (DEPT_NO);



```
| Supot          | Naudin          | Sales                |
| Terresa        | Benantar        | Sales                |
| Munehiro       | Zallocco        | Sales                |
| Lijie          | Lunn            | Sales                |
| Shalesh        | Shumilov        | Sales                |
| Nidapan        | Dymetman        | Sales                |
| Uri            | Rullman         | Sales                |
| Godehard       | Gammage         | Sales                |
| Hidde          | Bergere         | Sales                |
| Jianhua        | Leivant         | Sales                |
| Chuanti        | Karlin          | Sales                |
| Kwangho        | Reinhart        | Sales                |
| Fusako         | Stenning        | Sales                |
| Zhenbing       | Kadhim          | Sales                |
| Elliott        | Spieker         | Sales                |
| Danco          | Etalle          | Sales                |
| Vincent        | Papastamatiou   | Sales                |
| Lakshmi        | Deville         | Sales                |
| Rayond         | Cronau          | Sales                |
| Cheong         | Coors           | Sales                |
| Ramzi          | Furudate        | Sales                |
| Vishwani       | Petersohn       | Sales                |
| Arno           | Kumaresan       | Sales                |
| Munehiro       | Carrere         | Sales                |
+----------------+-----------------+----------------------+
1000 rows in set (0.01 sec)
```

TASK#8

Select emp_no, first_name , last_name, salary from employees natural join salaries order by salary desc limit 1;

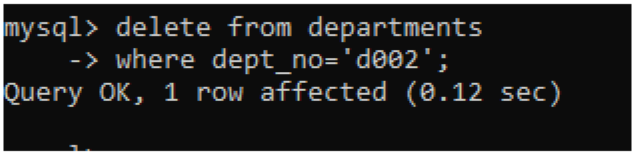


```
ql> Select emp_no, first_name , last_name, salary from employees natural join salaries order by salary desc limit 1;
------+------------+-----------+--------+
mp_no | first_name | last_name | salary |
------+------------+-----------+--------+
01801 | Subhash    | Baek      | 110963 |
------+------------+-----------+--------+
ow in set (0.01 sec)

ql>
```

TASK#9

ON DELETE CASCADE clause in MySQL is used to automatically remove the matching records from the child table when we delete the rows from the parent table. If the ON DELETE CASCADE is defined for one FOREIGN KEY clause only, then cascading operations will throw an error.

```
delete from departments

     where dept_no='d002';
```



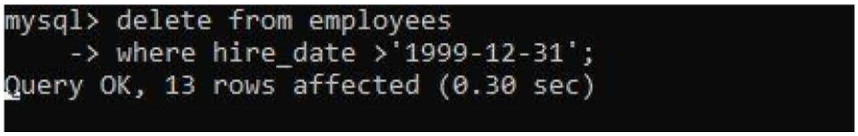
```
mysql> delete from departments
    -> where dept_no='d002';
Query OK, 1 row affected (0.12 sec)
```

As department is parent and dept_emp is child tale.

As I remove dempt_no=d002, it will automatically remove all dept_no from the child tale(dept_emp) having dept_no=d002.

TASK#10

Delete from employees Where
hire_date > ‘1999-12-31’;

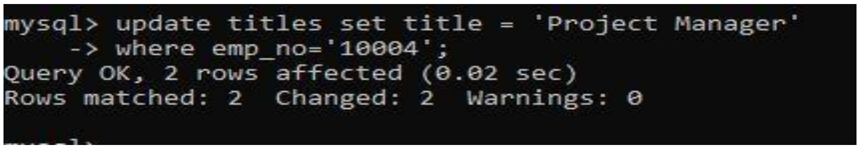


```
mysql> delete from employees
    -> where hire_date >'1999-12-31';
Query OK, 13 rows affected (0.30 sec)
```

TASK#11

Update titles set title = ‘Project Manager’

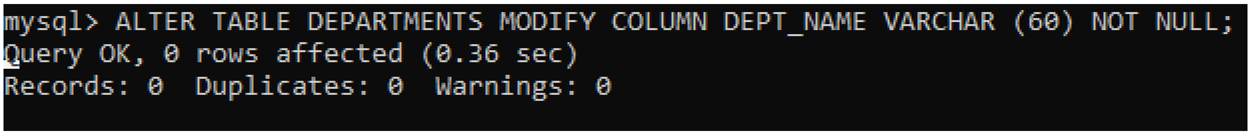
Where emp_no = ‘10004’;



```
mysql> update titles set title = 'Project Manager'
    -> where emp_no='10004';
Query OK, 2 rows affected (0.02 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

TASK#12

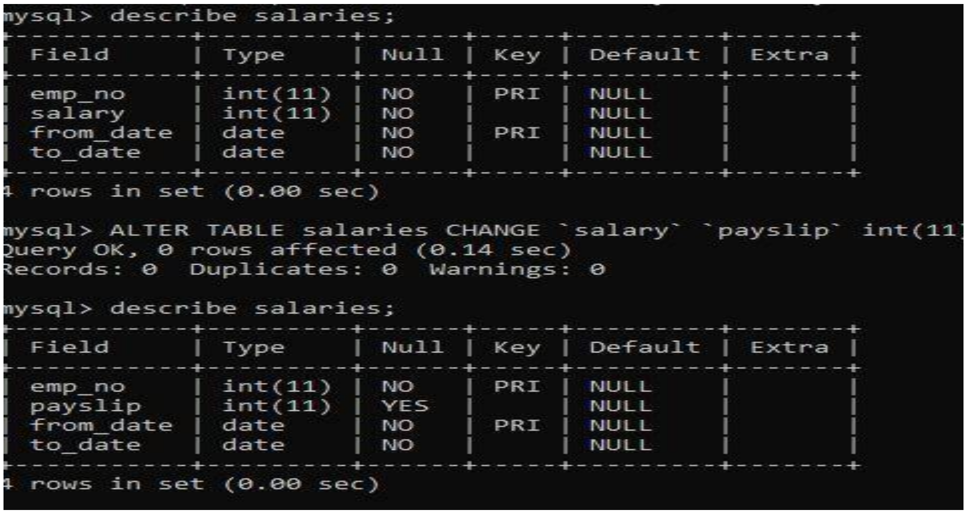
ALTER TABLE DEPARTMENTS MODIFY COLUMN DEPT_NAME VARCHAR (60) NOT NULL;



```
mysql> ALTER TABLE DEPARTMENTS MODIFY COLUMN DEPT_NAME VARCHAR (60) NOT NULL;
Query OK, 0 rows affected (0.36 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

TASK#13

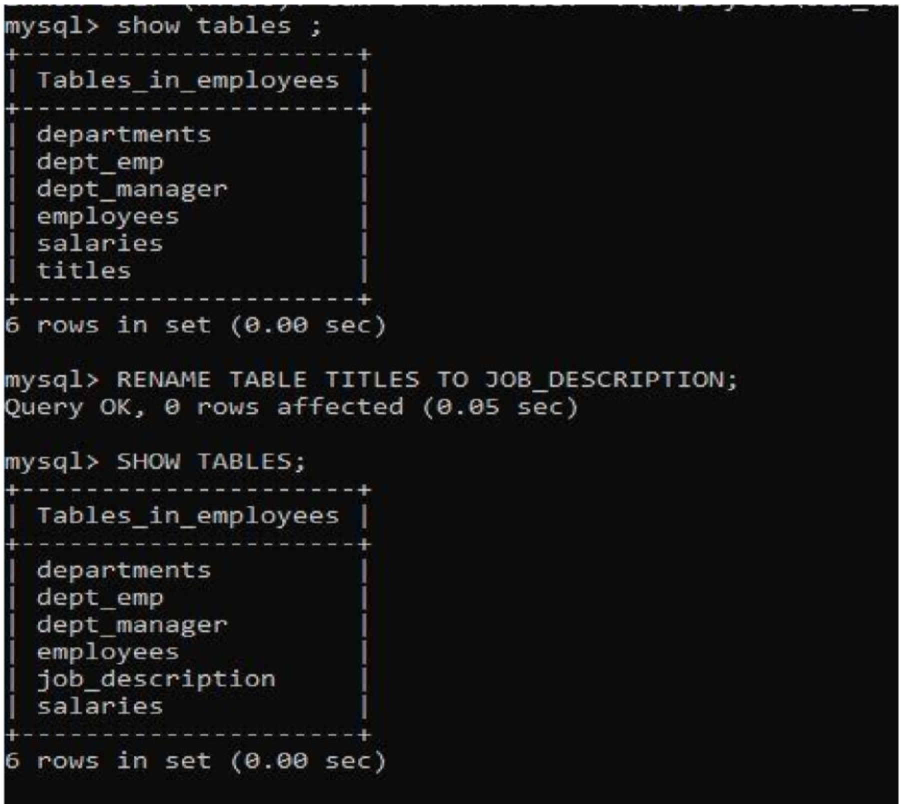
Alter table salaries change ‘salary ’ ‘Payslip’ int(11);



```
mysql> describe salaries;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| emp_no    | int(11) | NO   | PRI | NULL    |       |
| salary    | int(11) | NO   |     | NULL    |       |
| from_date | date    | NO   | PRI | NULL    |       |
| to_date   | date    | NO   |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> ALTER TABLE salaries CHANGE `salary` `payslip` int(11)
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe salaries;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| emp_no    | int(11) | NO   | PRI | NULL    |       |
| payslip   | int(11) | YES  |     | NULL    |       |
| from_date | date    | NO   | PRI | NULL    |       |
| to_date   | date    | NO   |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

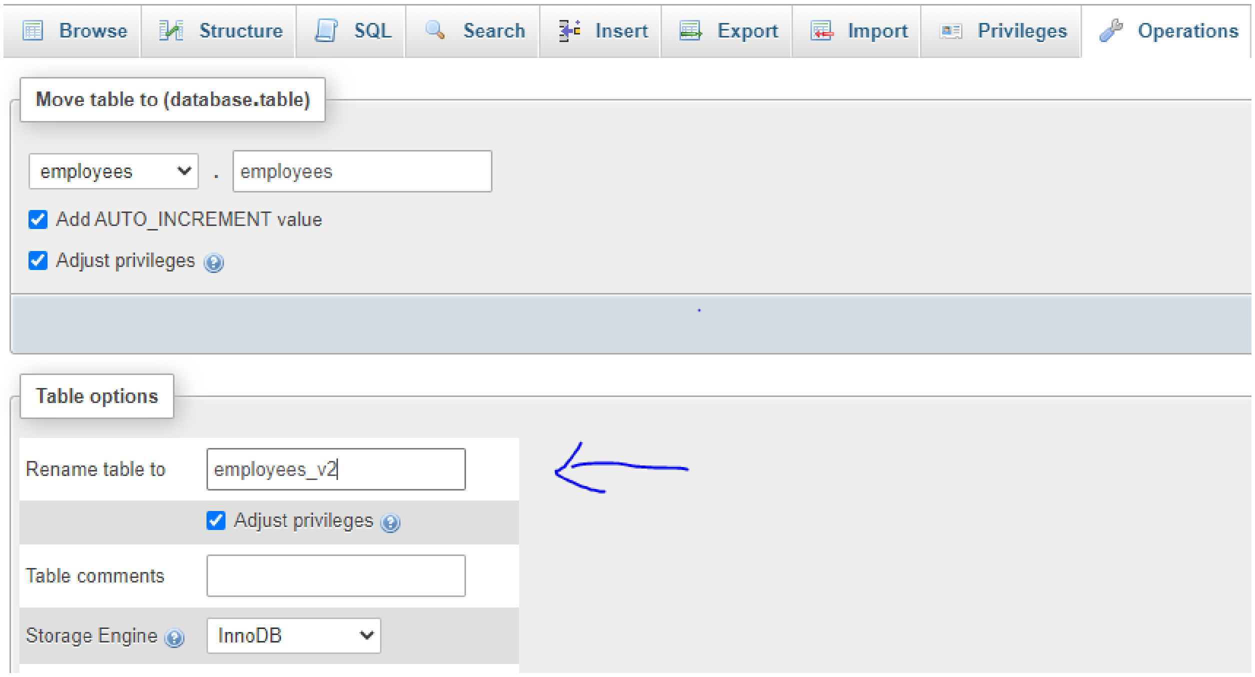
TASK#14

Rename table titles to Job_description;



```
mysql> show tables ;
+----------------------+
| Tables_in_employees |
+----------------------+
| departments          |
| dept_emp             |
| dept_manager         |
| employees            |
| salaries             |
| titles               |
+----------------------+
6 rows in set (0.00 sec)

mysql> RENAME TABLE TITLES TO JOB_DESCRIPTION;
Query OK, 0 rows affected (0.05 sec)

mysql> SHOW TABLES;
+----------------------+
| Tables_in_employees |
+----------------------+
| departments          |
| dept_emp             |
| dept_manager         |
| employees            |
| job_description      |
| salaries             |
+----------------------+
6 rows in set (0.00 sec)
```

TASK#15



Move table to (database.table)

employees . employees

☑ Add AUTO_INCREMENT value

☑ Adjust privileges

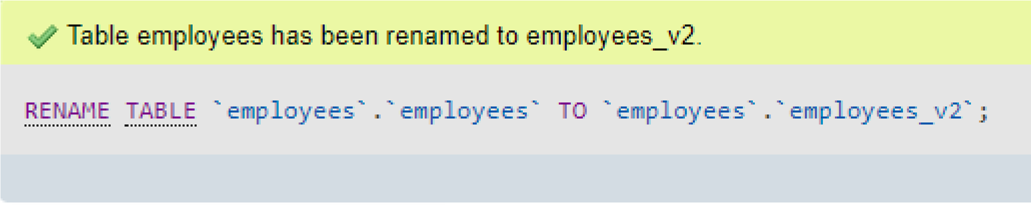
Table options

Rename table to: employees_v2

☑ Adjust privileges

Table comments:

Storage Engine: InnoDB



✓ Table employees has been renamed to employees_v2.

```
RENAME TABLE `employees`.`employees` TO `employees`.`employees_v2`;
```

QUESTION#2

```
CREATE DATABASE SCHOOL_SYSTEM;


CREATE TABLE MARKS (
   MARK_id      INT          NOT NULL,
   STUDENT_ID   INT          NOT NULL,
   SUBJECT_ID   INT          NOT NULL,
   DATE         DATE         NOT NULL,
   MARK         INT          NOT NULL,
   FOREIGN KEY (SUBJECT_ID)  REFERENCES SUBJECTS (SUBJECT_ID)  ON DELETE CASCADE,
   FOREIGN KEY (STUDENT_ID)  REFERENCES STUDENTS (STUDENT_ID) ON DELETE CASCADE,
   PRIMARY KEY (MARK_ID)
);


CREATE TABLE STUDENTS (
   STUDENT_ID   INT            NOT NULL,
   FIRST_NAME   VARCHAR(14)    NOT NULL,
   LAST_NAME    VARCHAR(16)    NOT NULL,
   GROUP_ID     INT            NOT NULL,
   FOREIGN KEY (GROUP_ID) REFERENCES GROUPS (GROUP_ID) ON DELETE CASCADE,
   PRIMARY KEY (STUDENT_ID)
);

CREATE TABLE GROUPS (
   GROUP_ID      INT            NOT NULL,
   NAME         VATCHAR(20)     NOT NULL,
   PRIMARY KEY (GROUP_ID)
);

CREATE TABLE SUBJECTS (
   SUBJECT_ID      INT            NOT NULL,
   TITLE         VARCHAR(20)     NOT NULL,
   PRIMARY KEY(SUBJECT_ID)
);

CREATE TABLE SUBJECT_TEACHER (
   TEACHER_ID   INT            NOT NULL,
   SUBJECT_ID    INT            NOT NULL,
   GROUP_ID     INT            NOT NULL,
  FOREIGN KEY (SUBJECT_ID)  REFERENCES SUBJECTS (SUBJECT_ID) ON DELETE CASCADE,
  FOREIGN KEY (GROUP_ID)  REFERENCES GROUPS (GROUP_ID) ON DELETE CASCADE,
  FOREIGN KEY (TEACHER_ID)  REFERENCES TEACHERS (TEACHER_ID)   ON DELETE CASCADE,
  PRIMARY KEY (SUBJECT_ID,TEACHER_ID,GROUP_ID)

);
```

```
CREATE TABLE TEACHERS (
   TEACHER_ID   INT            NOT NULL,
   FIRST_NAME   VARCHAR(14)     NOT NULL,
   LAST_NAME    VARCHAR(16)     NOT NULL,
   PRIMARY KEY (TEACHER_ID)
);
```