深　圳　大　学

实　　验　　报　　告

课程名称：数据库系统

实验序号：实验 1

实验名称：SQL的DDL语言和单表查询

学　　号：2022155028

姓　　名：黄亮铭

实验完成日期：2024 年 9 月 21 日

一、实验目的：

1、 了解 DBMS 系统的功能、软件组成；

2、掌握利用 SQL 语句定义、和简单操纵数据库的方法。

二、实验要求：

1、在课外安装相关软件并浏览软件自带的帮助文件和功能菜单，了解 DBMS 的功能、结构；

2、数据库、关系表定义；

3、学习定义关系表的约束(主键、外键、自定义)；

4、了解 SQL 的数据定义功能；

5、了解 SQL 的操纵基本功能；

6、了解视图的概念；

三、实验设备：

计算机、数据库管理系统如 MYSQL,DB2，Oracle 等软件。

四、实验内容

1、使用 SQL DDL 语句**建立关系数据库模式，并用 DML 数据**如下；

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-Dec-90	13750		20
7499	ALLEN	SALESMAN	7698	20-FEB-89	19000	6400	30
7521	WARD	SALESMAN	7698	22-FEB-93	18500	4250	30
7566	JONES	MANAGER	7839	02-APR-89	26850		20
7654	MARTIN	SALESMAN	7698	28-SEP-97	15675	3500	30
7698	BLAKE	MANAGER	7839	01-MAY-90	24000		30
7782	CLARK	MANAGER	7839	09-JUN-88	27500		10
7788	SCOTT	ANALYST	7566	19-APR-87	19500		20
7839	KING	PRESIDENT		17-NOV-83	82500		10
7844	TURNER	SALESMAN	7698	08-SEP-92	18500	6250	30
7876	ADAMS	CLERK	7788	23-MAY-96	11900		20
7900	JAMES	CLERK	7698	03-DEC-95	12500		30
7902	FORD	ANALYST	7566	03-DEC-91	21500		20
7934	MILLER	CLERK	7782	23-JAN-95	13250		10
3258	GREEN	SALESMAN	4422	24-Jul-95	18500	2750	50
4422	STEVENS	MANAGER	7839	14-Jan-94	24750		50
6548	BARNES	CLERK	4422	16-Jan-95	11950		50

DEPT+(学生自己的学号):

DEPTNO	DNAME	LOC
10	ACCOUNTING	LONDON
20	RESEARCH	PRESTON
30	SALES	LIVERPOOL
40	OPERATIONS	STAFFORD
50	MARKETING	LUTON

以下为学生实验填写部分：

1. 参考课件约束方式，创建 emp 和 dept 的 DDL 语句 （要有语句和运行结果截屏）

① 使用命令*create datasbase first_ex*创建数据库，然后使用命令*show databases*查看数据是否创建成功，如果成功，结果如下图所示。

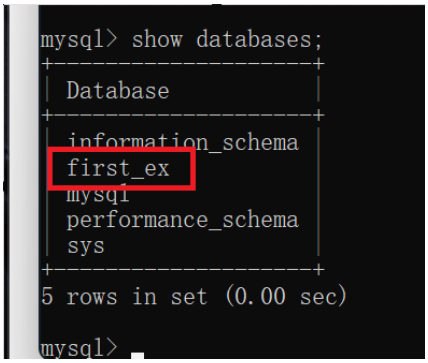

图 1：数据库创建成功

② 查看两个表中的键，发现 EMP 表中有 DEPT 表中的键，即：EMP 表中有外键约束。因此，我们需要先创建 DEPT 表，然后再创建 EMP 表。

i. 使用如下命令即可创建 DEPT 表。

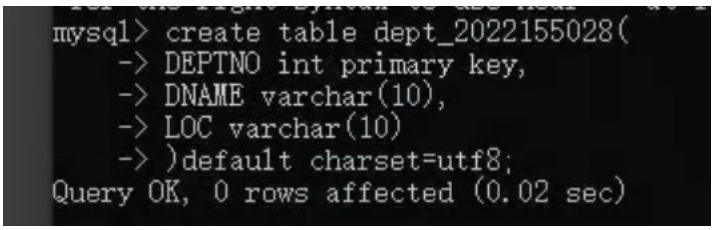

图 2：创建表 dept

ii. 检查 DEPT 表是否创建成功（使用命令*desc dept_2022155028*），如果成功则结果如下图所示，否则显示为空。

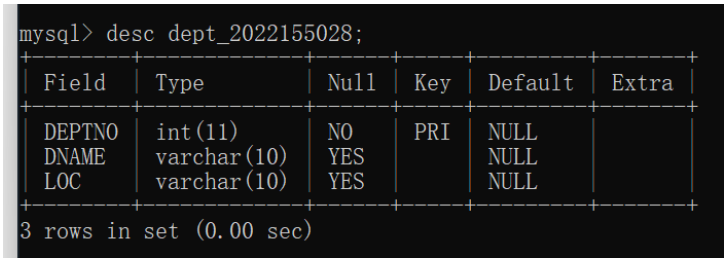

图 3：创建表 dept 成功

iii. 使用如下命令创建 EMP 表。

```
mysql> create table emp_2022155028(
    -> EMPNO int primary key,
    -> ENAME varchar(10),
    -> JOB varchar(10),
    -> MGR int,
    -> HIREDATE DATE,
    -> SAL decimal(10,2),
    -> COMM decimal(9,0),
    -> DEPTNO int,
    -> CONSTRAINT FK_DEPTNO FOREIGN KEY (DEPTNO) REFERENCES dept_2022155028(DEPTNO)
    -> )default charset=utf8;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

图 4：创建表 emp

iv. 检查 EMP 表是否创建成功（使用命令*desc emp_2022155028*），如果成功则结果如下图所示，否则显示为空。

```
mysql> desc emp_2022155028;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| EMPNO    | int(11)       | NO   | PRI | NULL    |       |
| ENAME    | varchar(10)   | YES  |     | NULL    |       |
| JOB      | varchar(10)   | YES  |     | NULL    |       |
| MGR      | int(11)       | YES  |     | NULL    |       |
| HIREDATE | date          | YES  |     | NULL    |       |
| SAL      | decimal(10,2) | YES  |     | NULL    |       |
| COMM     | decimal(9,0)  | YES  |     | NULL    |       |
| DEPTNO   | int(11)       | YES  | MUL | NULL    |       |
+----------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

图 5：创建表 emp 成功

v. 发现在创建表时忘记将外键 DEPTNO 设置为非空，因此在这里使用命令将该键设置为非空。

```
mysql> alter table emp_2022155028 modify DEPTNO int not null;
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

图 6：修改字段

vi. 然后再次使用命令*desc emp_2022155028*，发现字段已经设置为非空。

```
mysql> desc emp_2022155028;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| EMPNO    | int(11)       | NO   | PRI | NULL    |       |
| ENAME    | varchar(10)   | YES  |     | NULL    |       |
| JOB      | varchar(10)   | YES  |     | NULL    |       |
| MGR      | int(11)       | YES  |     | NULL    |       |
| HIREDATE | date          | YES  |     | NULL    |       |
| SAL      | decimal(10,2) | YES  |     | NULL    |       |
| COMM     | decimal(9,0)  | YES  |     | NULL    |       |
| DEPTNO   | int(11)       | NO   | MUL | NULL    |       |
+----------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

图 7：修改字段结果

vii. 创建表的时候忘记将 MGR 设置为 EMPNO 的外键，因此这里使用命令 *ALTER TABLE EMP_2022155028 ADD FOREIGN KEY (MGR) REFERENCES EMP_2022155028(EMPNO)*。

```
mysql> ALTER TABLE EMP_2022155028 ADD FOREIGN KEY (MGR) REFERENCES EMP_2022155028(EMPNO);
Query OK, 17 rows affected (0.04 sec)
Records: 17  Duplicates: 0  Warnings: 0

mysql>
```

图 8：修改字段

viii. 最终效果展示

```
mysql> DESC DEPT_2022155028;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| DEPTNO | int(11)     | NO   | PRI | NULL    |       |
| DNAME  | varchar(10) | YES  |     | NULL    |       |
| LOC    | varchar(10) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql> DESC EMP_2022155028;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| EMPNO    | int(11)       | NO   | PRI | NULL    |       |
| ENAME    | varchar(10)   | YES  |     | NULL    |       |
| JOB      | varchar(10)   | YES  |     | NULL    |       |
| MGR      | int(11)       | YES  | MUL | NULL    |       |
| HIREDATE | date          | YES  |     | NULL    |       |
| SAL      | decimal(10,2) | YES  |     | NULL    |       |
| COMM     | decimal(9,0)  | YES  |     | NULL    |       |
| DEPTNO   | int(11)       | NO   | MUL | NULL    |       |
+----------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

图 9：最终效果

2. 插入 emp 和 dept 数据的 DML 语句 （要有语句和运行结果截屏）

① 使用命令 *INSERT INTO dept_2022155028 VALUES (10, 'ACCOUNTING', 'LONDON'), (20, 'RESEARCH', 'PRESTON'), (30, 'SALES', 'LIVERPOOL'), (40, 'OPERATIONS', 'STAFFORD'), (50, 'MARKETING', 'LUTON');*向表中插入数据，然后输入命令 *SELECT * FROM dept_2022155028*查看插入结果（见下图）。

```
mysql> select * from dept_2022155028;
+--------+------------+-----------+
| DEPTNO | DNAME      | LOC       |
+--------+------------+-----------+
|     10 | ACCOUNTING | LONDON    |
|     20 | RESEARCH   | PRESTON   |
|     30 | SALES      | LIVERPOOL |
|     40 | OPERATIONS | STAFFORD  |
|     50 | MARKETING  | LUTON     |
+--------+------------+-----------+
5 rows in set (0.00 sec)

mysql>
```

图 10：表 dept_2022155028 插入结果

② 使用命令 INSERT INTO emp_2022155028 VALUE (7369, 'SMITH', 'CLERK', 7902, '1990-12-17', 13750, NULL, 20), (XX,XXX,……,XX);即可向表中插入数据（因为数据太长，这里只以第一条数据的插入作为命令演示，实际上可以像①中多条插入），然后输入命令$SELECT * FROM\ emp_2022155028$查看插入结果（见下图）。

```
mysql> select * from emp_2022155028;
+-------+--------+----------+------+------------+----------+------+--------+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL      | COMM | DEPTNO |
+-------+--------+----------+------+------------+----------+------+--------+
|  3258 | GREEN  | SALESMAN | 4422 | 1995-07-24 | 18500.00 | 2750 |     50 |
|  4422 | STEVENS| MANAGER  | 7839 | 1994-01-14 | 24750.00 | NULL |     50 |
|  6548 | BARNES | CLERK    | 4422 | 1995-01-16 | 11950.00 | NULL |     50 |
|  7369 | SMITH  | CLERK    | 7902 | 1990-12-17 | 13750.00 | NULL |     20 |
|  7499 | ALLEN  | SALESMAN | 7698 | 1989-02-20 | 19000.00 | 6400 |     30 |
|  7521 | WARD   | SALESMAN | 7698 | 1993-02-22 | 18500.00 | 4250 |     30 |
|  7566 | JONES  | MANAGER  | 7839 | 1989-04-02 | 26850.00 | NULL |     20 |
|  7654 | MARTIN | SALESMAN | 7698 | 1997-09-28 | 15675.00 | 3500 |     30 |
|  7698 | BLAKE  | MANAGER  | 7839 | 1990-05-01 | 24000.00 | NULL |     30 |
|  7782 | CLARK  | MANAGER  | 7839 | 1988-06-09 | 27500.00 | NULL |     10 |
|  7788 | SCOTT  | ANALYST  | 7566 | 1987-04-19 | 19500.00 | NULL |     20 |
|  7839 | KING   | PRESIDENT| NULL | 1983-11-17 | 82500.00 | NULL |     10 |
|  7844 | TURNER | SALESMAN | 7698 | 1992-09-08 | 18500.00 | 6250 |     30 |
|  7876 | ADAMS  | CLERK    | 7788 | 1996-05-23 | 11900.00 | NULL |     20 |
|  7900 | JAMES  | CLERK    | 7698 | 1995-12-03 | 12500.00 | NULL |     30 |
|  7902 | FORD   | ANALYST  | 7566 | 1991-12-03 | 21500.00 | NULL |     20 |
|  7934 | MILLER | CLERK    | 7782 | 1995-01-23 | 13250.00 | NULL |     10 |
+-------+--------+----------+------+------------+----------+------+--------+
17 rows in set (0.00 sec)

mysql>
```

图 11：emp_2022155028 插入结果

3．完成实验指导书中对应老师指定的题目.（要有题目语句和运行结果截屏）

EX1

7. What is the name, job title and employee number of the person in department 20 who earns more than £25000.

输入命令：SELECT ENAME, JOB, EMPNO FROM EMP_2022155028 WHERE DEPNO = 20 AND SAL > 25000;

```
mysql> SELECT ENAME, JOB, EMPNO FROM EMP_2022155028 WHERE DEPTNO = 20 AND SAL > 25000;
+-------+---------+-------+
| ENAME | JOB     | EMPNO |
+-------+---------+-------+
| JONES | MANAGER |  7566 |
+-------+---------+-------+
1 row in set (0.01 sec)
```

图 12：运行结果

9. Find any Clerk who is not in department 10.

输入命令：SELECT * FROM EMP_2022155028 WHERE DEPTNO != 10 AND JOB = “CLERK”;

```
mysql> SELECT * FROM EMP_2022155028 WHERE DEPTNO != 10 AND JOB = "CLERK";
+-------+--------+-------+------+------------+----------+------+--------+
| EMPNO | ENAME  | JOB   | MGR  | HIREDATE   | SAL      | COMM | DEPTNO |
+-------+--------+-------+------+------------+----------+------+--------+
|  6548 | BARNES | CLERK | 4422 | 1995-01-16 | 11950.00 | NULL |     50 |
|  7369 | SMITH  | CLERK | 7902 | 1990-12-17 | 13750.00 | NULL |     20 |
|  7876 | ADAMS  | CLERK | 7788 | 1996-05-23 | 11900.00 | NULL |     20 |
|  7900 | JAMES  | CLERK | 7698 | 1995-12-03 | 12500.00 | NULL |     30 |
+-------+--------+-------+------+------------+----------+------+--------+
4 rows in set (0.01 sec)
```

图 13：运行结果

12 Find the name of the President.

输入命令：SELECT ENAME FROM EMP_2022155028 WHERE JOB = “PRESIDENT”;

```
mysql> SELECT ENAME FROM EMP_2022155028 WHERE JOB = "PRESIDENT";
+--------+
| ENAME  |
+--------+
| KING   |
+--------+
1 row in set (0.00 sec)
```

图 14：运行结果

14 List the employees whose names have TH or LL in them

输入命令：SELECT * FROM EMP_2022155028 WHERE ENAME LIKE “%TH%” OR ENAME LIKE “%LL%”;

```
mysql> SELECT * FROM EMP_2022155028 WHERE ENAME LIKE "%TH%" OR ENAME LIKE "%LL%";
+-------+--------+---------+------+------------+----------+------+--------+
| EMPNO | ENAME  | JOB     | MGR  | HIREDATE   | SAL      | COMM | DEPTNO |
+-------+--------+---------+------+------------+----------+------+--------+
|  7369 | SMITH  | CLERK   | 7902 | 1990-12-17 | 13750.00 | NULL |     20 |
|  7499 | ALLEN  | SALESMAN| 7698 | 1989-02-20 | 19000.00 | 6400 |     30 |
|  7934 | MILLER | CLERK   | 7782 | 1995-01-23 | 13250.00 | NULL |     10 |
+-------+--------+---------+------+------------+----------+------+--------+
3 rows in set (0.00 sec)
```

图 15：运行结果

17. Find the name, job, salary, hiredate and department number of all employees in ascending order by their salaries.

输入命令：SELECT ENAME, JOB, SAL, HIREDATE, DEPTNO FROM EMP_2022155028 ORDER BY SAL;

```
mysql> SELECT ENAME, JOB, SAL, HIREDATE, DEPTNO FROM EMP_2022155028 ORDER BY SAL;
+---------+-----------+----------+------------+--------+
| ENAME   | JOB       | SAL      | HIREDATE   | DEPTNO |
+---------+-----------+----------+------------+--------+
| ADAMS   | CLERK     | 11900.00 | 1996-05-23 |     20 |
| BARNES  | CLERK     | 11950.00 | 1995-01-16 |     50 |
| JAMES   | CLERK     | 12500.00 | 1995-12-03 |     30 |
| MILLER  | CLERK     | 13250.00 | 1995-01-23 |     10 |
| SMITH   | CLERK     | 13750.00 | 1990-12-17 |     20 |
| MARTIN  | SALESMAN  | 15675.00 | 1997-09-28 |     30 |
| GREEN   | SALESMAN  | 18500.00 | 1995-07-24 |     50 |
| WARD    | SALESMAN  | 18500.00 | 1993-02-22 |     30 |
| TURNER  | SALESMAN  | 18500.00 | 1992-09-08 |     30 |
| ALLEN   | SALESMAN  | 19000.00 | 1989-02-20 |     30 |
| SCOTT   | ANALYST   | 19500.00 | 1987-04-19 |     20 |
| FORD    | ANALYST   | 21500.00 | 1991-12-03 |     20 |
| BLAKE   | MANAGER   | 24000.00 | 1990-05-01 |     30 |
| STEVENS | MANAGER   | 24750.00 | 1994-01-14 |     50 |
| JONES   | MANAGER   | 26850.00 | 1989-04-02 |     20 |
| CLARK   | MANAGER   | 27500.00 | 1988-06-09 |     10 |
| KING    | PRESIDENT | 82500.00 | 1983-11-17 |     10 |
+---------+-----------+----------+------------+--------+
17 rows in set (0.01 sec)
```

图 16：运行结果

18. List all salesmen in descending order by commission divided by their salary.

输入命令：SELECT * FROM EMP_2022155028 WHERE JOB = "SALESMAN" ORDER BY (COMM / SAL) DESC;

```
mysql> SELECT * FROM EMP_2022155028 WHERE JOB = "SALESMAN" ORDER BY (COMM / SAL) DESC;
+-------+--------+----------+------+------------+----------+------+--------+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL      | COMM | DEPTNO |
+-------+--------+----------+------+------------+----------+------+--------+
|  7844 | TURNER | SALESMAN | 7698 | 1992-09-08 | 18500.00 | 6250 |     30 |
|  7499 | ALLEN  | SALESMAN | 7698 | 1989-02-20 | 19000.00 | 6400 |     30 |
|  7521 | WARD   | SALESMAN | 7698 | 1993-02-22 | 18500.00 | 4250 |     30 |
|  7654 | MARTIN | SALESMAN | 7698 | 1997-09-28 | 15675.00 | 3500 |     30 |
|  3258 | GREEN  | SALESMAN | 4422 | 1995-07-24 | 18500.00 | 2750 |     50 |
+-------+--------+----------+------+------------+----------+------+--------+
5 rows in set (0.00 sec)
```

图 17：运行结果

19. Order employees in department 30 who receive commision, in ascending order by commission

输入命令：SELECT * FROM EMP_2022155028 WHERE DEPTNO = 30 AND COMM IS NOT NULL ORDER BY COALESCE(COMM, 0);

```
mysql> SELECT * FROM EMP_2022155028 WHERE DEPTNO = 30 AND COMM IS NOT NULL ORDER BY COALESCE(COMM, 0);
+-------+--------+----------+------+------------+----------+------+--------+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL      | COMM | DEPTNO |
+-------+--------+----------+------+------------+----------+------+--------+
|  7654 | MARTIN | SALESMAN | 7698 | 1997-09-28 | 15675.00 | 3500 |     30 |
|  7521 | WARD   | SALESMAN | 7698 | 1993-02-22 | 18500.00 | 4250 |     30 |
|  7844 | TURNER | SALESMAN | 7698 | 1992-09-08 | 18500.00 | 6250 |     30 |
|  7499 | ALLEN  | SALESMAN | 7698 | 1989-02-20 | 19000.00 | 6400 |     30 |
+-------+--------+----------+------+------------+----------+------+--------+
4 rows in set (0.00 sec)
```

图 18：运行结果

20 Find the names, jobs, salaries and commissions of all employees who do not have managers.

输入命令：SELECT ENAME, JOB, SAL, COMM FROM EMP_2022155028 WHERE MGR IS NULL;

```
mysql> SELECT ENAME, JOB, SAL, COMM FROM EMP_2022155028 WHERE MGR IS NULL;
+-------+-----------+----------+------+
| ENAME | JOB       | SAL      | COMM |
+-------+-----------+----------+------+
| KING  | PRESIDENT | 82500.00 | NULL |
+-------+-----------+----------+------+
1 row in set (0.01 sec)
```

图 19：运行结果

EX2

3. List the names of all salesmen who work in SALES

输入命令：SELECT ENAME FROM EMP_2022155028, DEPT_2022155028 WHERE EMP_2022155028.DEPTNO = DEPT_2022155028.DEPTNO AND DEPT_2022155028.DNAME = “SALES”;

```
mysql> SELECT ENAME FROM EMP_
022155028.DNAME = "SALES";
+--------+
| ENAME  |
+--------+
| ALLEN  |
| WARD   |
| MARTIN |
| BLAKE  |
| TURNER |
| JAMES  |
+--------+
```

图 20：运行结果

4. List all departments that do not have any employees.

输入命令：SELECT DNAME FROM DEPT_2022155028 WHERE DEPT_2022155028.DEMPTNO NOT IN (SELECT DEPTNO FROM EMP_2022155028);

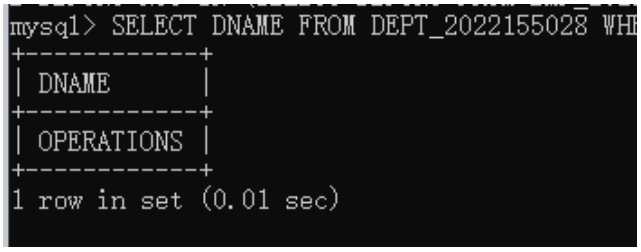

图 21：运行结果

5 For each employee whose salary exceeds his manager's salary, list the employee's name and salary and the manager's name and salary.

输入命令：SELECT WORKER.ENAME, WORKER.SAL, MANAGER.ENAME, MANAGER.SAL FROM EMP_2022155028 WORKER, EMP_2022155028 MANAGER WHERE WORKER.MGR = MANAGER.EMPNO AND WORKER.SAL > MANAGER.SAL;

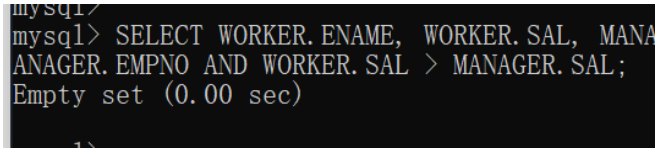

图 22：运行结果

6. List the employees who have BLAKE as their manager.

输入命令：SELECT WORKER.* FROM EMP_2022155028 WORKER, EMP_2022155028 MANAGER WHERE WORKER.MGR = MANAGER.EMPNO AND MANAGER.ENAME = "BLAKE";

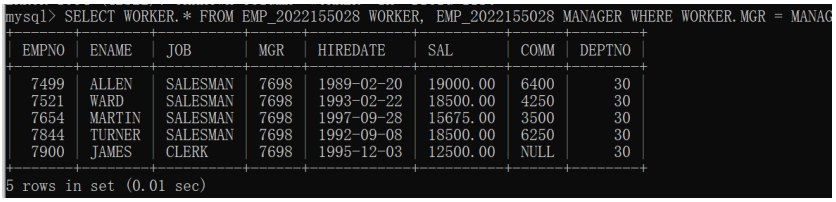

图 23：运行结果

开放性题目：索引的创建与优化分析

1. 你需要设计一个优化查询来找到在指定年份（比如 2000 年）之后入职的所有员工的姓名和工资。请根据查询的需求设计一个索引，来优化如下查询的执行效率。使用 EXPLAIN 分析输出显示索引的有效性。

使用命令：*EXPLAIN SELECT ENAME, SAL FROM EMP_2022155028 WHERE HIREDATE > "2000-01-01";*以及使用命令：*EXPLAIN SELECT ENAME, SAL FROM EMP_2022155028 WHERE HIREDATE > "1990-01-01";*进行测试。

不建立索引的测试结果如下：可以看到，无论查询的数据是否在表中存在，在没有索引的情况下均会遍历整个表的 17 行数据。

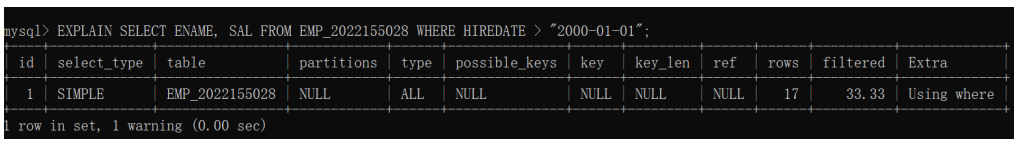

图 24：测试结果（1）

```
mysql> EXPLAIN SELECT ENAME, SAL FROM EMP_2022155028 WHERE HIREDATE > "1990-01-01";
+----+-------------+-----------------+------------+------+---------------+------+---------+------+------+----------+-------------+
| id | select_type | table           | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra       |
+----+-------------+-----------------+------------+------+---------------+------+---------+------+------+----------+-------------+
|  1 | SIMPLE      | EMP_2022155028 | NULL       | ALL  | NULL          | NULL | NULL    | NULL |   17 |    33.33 | Using where |
+----+-------------+-----------------+------------+------+---------------+------+---------+------+------+----------+-------------+
1 row in set, 1 warning (0.00 sec)
```

图 25：测试结果（2）

因为题目要求涉及查询年份、员工姓名和工资，所以使用命令：*CREATE INDEX YEAR ON EMP_2022155028(HIREDATE, ENAME, SAL);*建立索引。

建立索引的测试结果如下：可以看到在使用第一条命令的情况下只需要探测一行数据即可，相比于建立索引前的 17 行，探测量大大减少，由此证明索引的有效性；即使是第二条命令也只需要探测 12 行数据，相比于 17 行，探测量也下降了不少，由此也可以证明索引的有效性。

```
mysql> EXPLAIN SELECT ENAME, SAL FROM EMP_2022155028 WHERE HIREDATE > "2000-01-01";
+----+-------------+-----------------+------------+-------+---------------+------+---------+------+------+----------+-----------------------+
| id | select_type | table           | partitions | type  | possible_keys | key  | key_len | ref  | rows | filtered | Extra                 |
+----+-------------+-----------------+------------+-------+---------------+------+---------+------+------+----------+-----------------------+
|  1 | SIMPLE      | EMP_2022155028 | NULL       | range | YEAR          | YEAR | 4       | NULL |    1 |   100.00 | Using where; Using index |
+----+-------------+-----------------+------------+-------+---------------+------+---------+------+------+----------+-----------------------+
1 row in set, 1 warning (0.00 sec)
```

图 26：测试结果（1）

```
mysql> EXPLAIN SELECT ENAME, SAL FROM EMP_2022155028 WHERE HIREDATE > "1990-01-01";
+----+-------------+-----------------+------------+-------+---------------+------+---------+------+------+----------+-----------------------+
| id | select_type | table           | partitions | type  | possible_keys | key  | key_len | ref  | rows | filtered | Extra                 |
+----+-------------+-----------------+------------+-------+---------------+------+---------+------+------+----------+-----------------------+
|  1 | SIMPLE      | EMP_2022155028 | NULL       | range | YEAR          | YEAR | 4       | NULL |   12 |   100.00 | Using where; Using index |
+----+-------------+-----------------+------------+-------+---------------+------+---------+------+------+----------+-----------------------+
1 row in set, 1 warning (0.00 sec)

mysql>
```

图 27：测试结果（2）

2. 索引回表（Index Back to Table）是 MySQL 中常见的一个概念，通常发生在使用非聚簇索引（即辅助索引）的查询过程中。当查询通过索引检索到相关的索引列数据后，还需要回到数据表中查找非索引列的数据，这一过程就被称为回表。

这种回表的操作增加了 IO 操作，尤其是在大量查询数据时，会对性能产生一定的影响。因此，设计合理的索引，可以避免或减少回表操作，提升查询效率。请对此问题，设计相关索引并说明解决方案。

① 设置覆盖索引。设置覆盖索引即是指索引包含所有查询所需的列，这样查询可以直接通过索引来获取数据，而不需要回表。

解决方案：如果查询经常访问某些列，而这些列已经存在于某个索引中，那么可以创建一个包含这些列的复合索引。

② 根据索引的选择性排列索引列顺序。在复合索引中，MySQL 会从左到右使用索引中的列，所以最左边的列应该具有最高的选择性。

解决方案：根据查询模式调整复合索引中列的顺序，使得选择性高的列排在前面。

③ 利用索引合并策略：MySQL 可以使用多个索引来优化查。

解决方案：设计索引时，考虑可能的查询模式，以便 MySQL 可以有效地使用多个索引。

④ 利用索引的选择性：选择性高的索引可以更有效地过滤数据。选择性是指索引列中唯一值的数量与表中行数的比例。

解决方案：选择那些具有高选择性的列来创建索引，这样可以减少查询时需要检查的行数，从而减少回表的可能性。

⑤ 避免使用函数，因为函数会阻止 MySQL 使用索引。

解决方案：1）避免在 WHERE 子句中使用函数或表达式，或者创建计算列并为其创建索引。2）将 MySQL 版本升级为 8.x，因为最新版本的 MySQL 支持使用函数索引。

五.实验心得

1 通过这次数据库系统的实验，我对 SQL 的 DDL 语言 DML 语言有了更深入的了解和实践。

2 在数据插入和查询的过程中,我体会到了 SQL 语言的强大和灵活性。通过编写 INSERT INTO 语句，我成功地向数据库表中插入了数据，而 SELECT 语句则让我能够根据需要检索数据。此外，我还学习了如何使用 ORDER BY 子句对查询结果进行排序，以及如何使用 LIKE 子句进行模式匹配查询。

3 通过 EXPLAIN 命令，我观察到了索引对查询效率的影响，这让我认识到了索引在数据库性能优化中的关键作用。此外，我也学习了如何设计合理的索引，以减少查询过程中的回表操作，提高查询效率。

指导教师批阅意见：

成绩评定：

指导教师签字：

年　　月　　日

备注：