INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

BASE DE DATOS

PRÁCTICA 7

PROFESOR: HERNÁNDEZ CONTRERAS EULER

2CM10

PEREZ RAYA ALEJANDRO ADOLFO

Índice

1. Marco Teórico

Comandos	Descripción
SELECT * FROM nom_Relación;	Sintaxis general para consultar todos los datos de la relación.

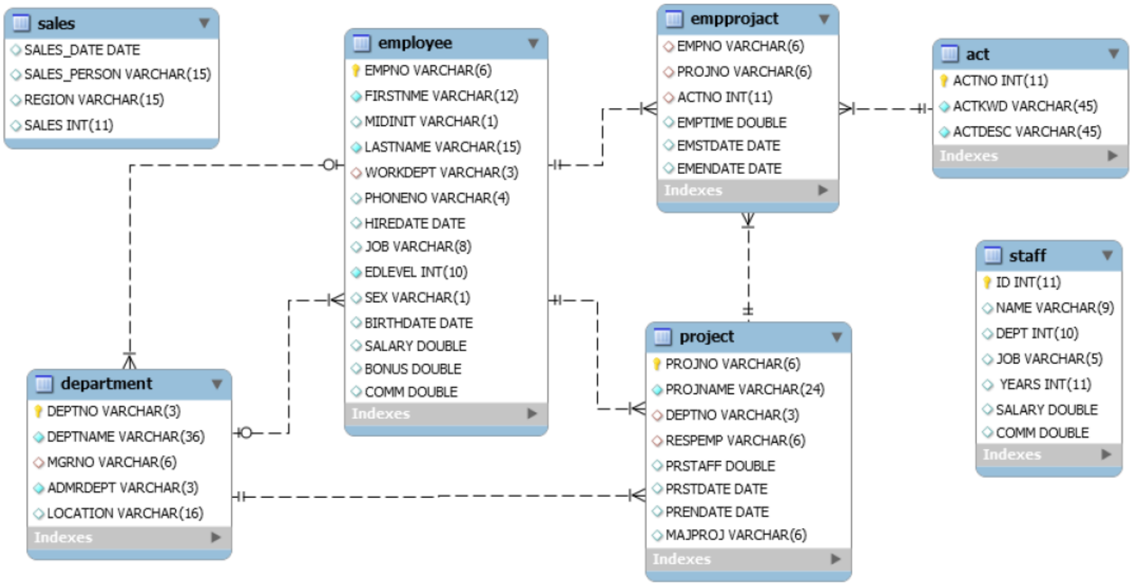
SELECT campo AS newNameCampo FROM nom_Relación;
SELECT campo FROM nom_Relación;
	Comando para consultar solo el campo indicado, la palabra reservada AS renombra el nombre del campo solo para la salida a pantalla, el uso de AS es opcional.

SELECT campo1, campo2 FROM nom_Relación;	Si se desea consultar más de un campo de la relación, se deben de poner separados por una coma.
SELECT campo FROM nom_Relación WHERE campo="Chihuahua";	La palabra clave WHERE seguida por una expresión que indica la condición o condiciones que deben satisfacer los registros para ser seleccionados. Todas las consultas pueden usar la palabra clave WHERE y si se desea más de una condición basta con poner la palabra AND entre cada una de las condiciones. OPCION: Puede ocupar las siguientes condiciones. • Campo=”Valor” • Campo>Valor • Campo<Valor • Campo like “Valor%” El % indica que no importa lo que este adelante después del valor que buscamos.
SELECT a.campo, b.campo2 FROM nom_Relación a, nomRelación2 b;	Para consultar diferentes valores de diferentes relaciones desde una sola línea de comando, usamos alias (letras en rojo) al principio del campo a consultar ponemos alias.campo y después del nombre de la relación ponemos el nombre del alias que ocupamos en el campo (sin el punto).
SELECT COUNT(*) FROM nom_Relación;	Este comando cuenta el total de tuplas existentes en la relación.

SELECT * FROM nom_Relación ORDER BY campo DESC;	La palabra reservada ORDER BY ordena la salida con respecto al campo que se le indique y la palabra DESC o ASC se refieren a orden descendente o ascendente respectivamente.
SELECT DISTINCT campo FROM nom_Relación	La palabra reservada DISTINCT especifica que los registros duplicados en el conjunto de resultados no deben de ser mostrados.
SELECT campo FROM nom_Relación GROUP BY campo;	La palabra reservada GROUP BY realiza la tarea del ORDER BY y de DISTINCT a la vez, quita los valores repetidos y los ordena.
INSERT INTO nom_Relación (campo1,campo2,…,campoN) VALUES (val1,val2,…,valN);	Este comando nos permite instanciar los campos del registro, si se van a llenar todos los campos se puede omitir el parte de (campo1,campo2,..,campon).
SELECT * FROM nomRelación WHERE campo BETWEEN valor1 and valor2 ;	El comando BETWEEN nos ayuda en las condiciones para buscar los datos donde su campo este entre los valores indicados.
SELECT * FROM nomRelación WHERE campo IN (valor1,valor2,…,valorn);	EL comando IN es una condición del WHERE el cual indica que se mostrara los datos donde el valor sea igual a valor1 o valor2 o … o valor n.

2. Instrucciones

Resuelve las siguientes consultas usando el siguiente modelo relacional.



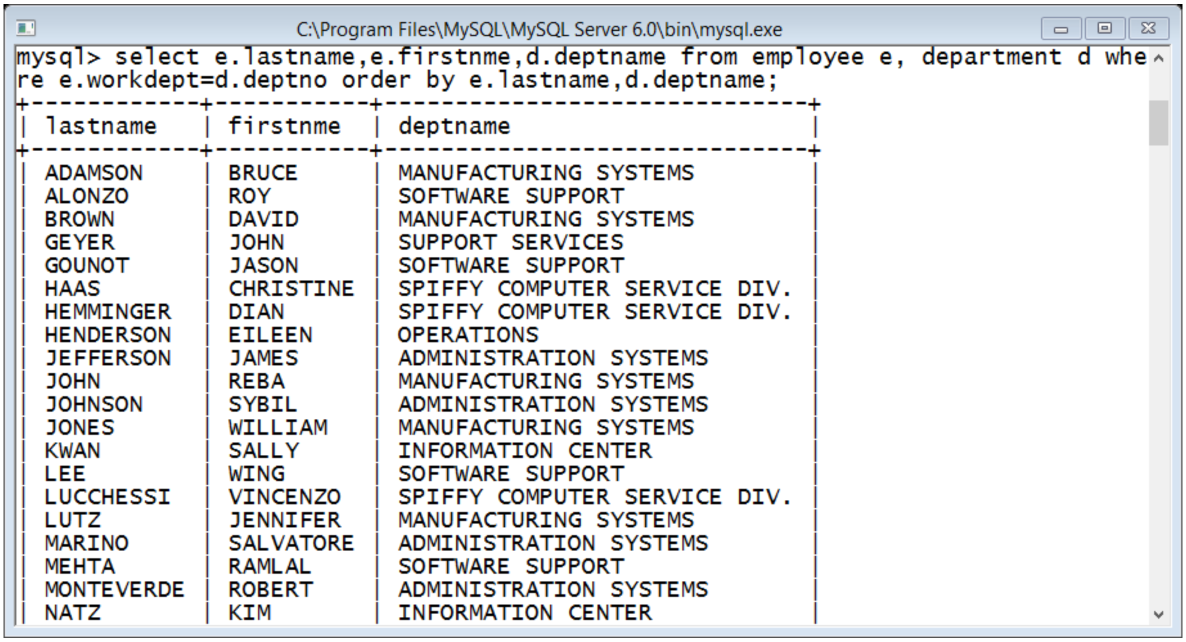
1. Produce a report that lists employees' last names, first names, and department names. Sequence the report on first name within last name, within department name.
2. Modify the previous query to include job. Also, list data for only departments between A02 and D22, and exclude managers from the list.
3. List the name of each department and the lastname and first name of its manager. Sequence the list by department name. Use the EMPNO and MGRNO columns to relate the two tables. Sequence the result rows by department name.
4. Try the following: modify the previous query using WORKDEPT and DEPTNO as the join predicate. Include a local predicate that looks for people whose job is manager.
5. For all projects that have a project number beginning with AD, list project number, project name, and activity number. List identical rows once. Order the list by project number and then by activity number.
6. Which employees are assigned to project number AD3113? List employee number, last name, and project number. Order the list by employee number and then by project number. List only one occurrence of duplicate result rows.
7. Which activities began on October 1, 2002? For each of these activities, list the employee number of the person performing the activity, the project

number, project name, activity number, and starting date of the activity. Order the list by project number, then by employee number, and then by activity number.

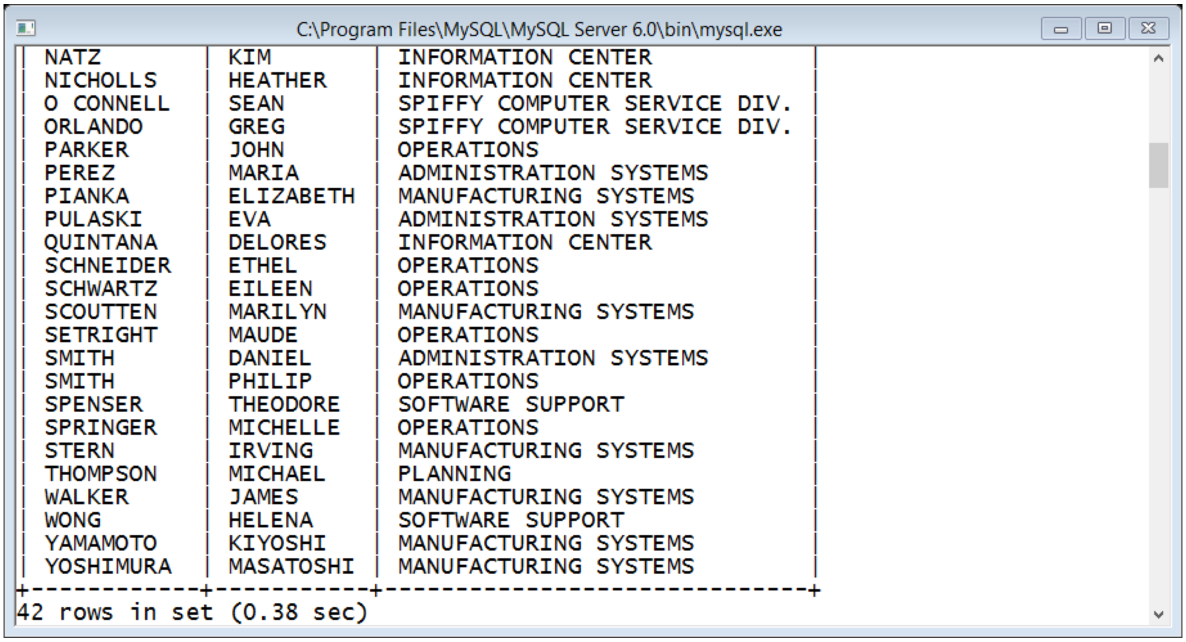
8. Display department number, last name, project name, and activity number for activities performed by the employees in department A00. Sequence the results first by project name and then by activity number.
9. List department number, last name, project name, and activity number for those employees in work departments A00 through C01. Suppress identical rows. Sort the list by department number, last name, and activity number.
10. The second line manager needs a list of activities which began on October 15, 1982 or thereafter. For these activities, list the activity number, the manager number of the manager of the department assigned to the project, the starting date for the activity, the project number, and the last name of the employee performing the activity. The list should be ordered by the activity number and then by the activity start date.

3. Desarrollo

1. select e.lastname,e.firstnme,d.deptname from employee e, department d where e.workdept=d.deptno order by e.lastname,d.deptname

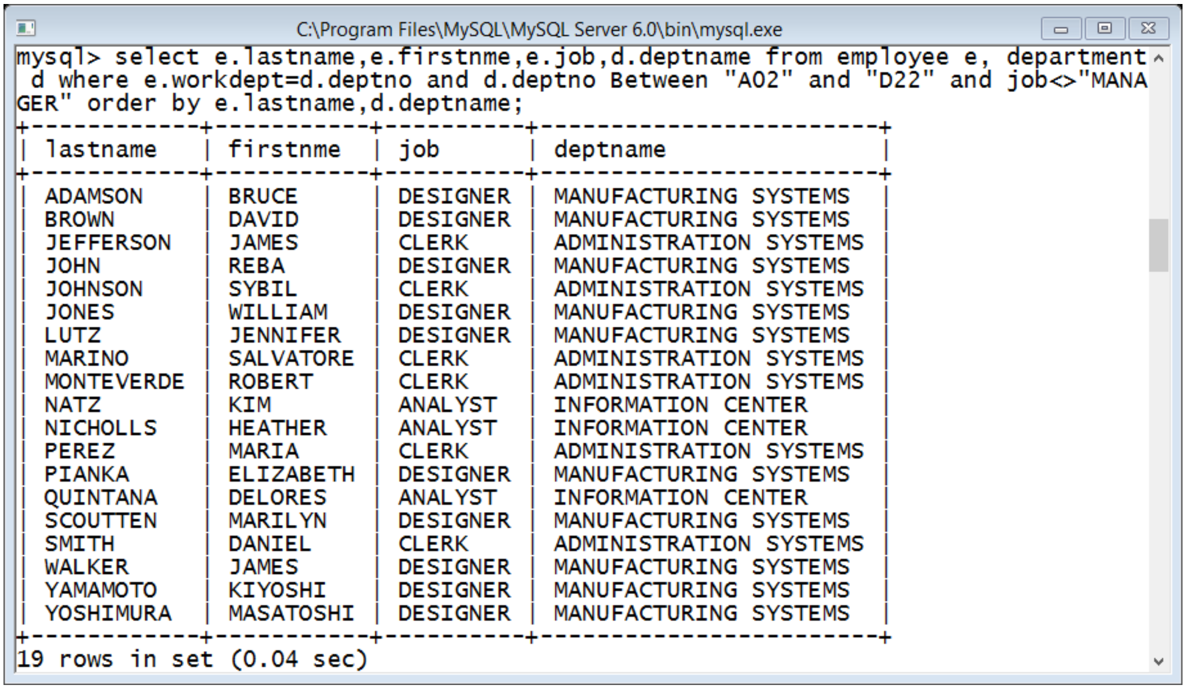


```
mysql> select e.lastname,e.firstnme,d.deptname from employee e, department d whe
re e.workdept=d.deptno order by e.lastname,d.deptname;
+------------+-----------+------------------------------+
| lastname   | firstnme  | deptname                     |
+------------+-----------+------------------------------+
| ADAMSON    | BRUCE     | MANUFACTURING SYSTEMS        |
| ALONZO     | ROY       | SOFTWARE SUPPORT             |
| BROWN      | DAVID     | MANUFACTURING SYSTEMS        |
| GEYER      | JOHN      | SUPPORT SERVICES             |
| GOUNOT     | JASON     | SOFTWARE SUPPORT             |
| HAAS       | CHRISTINE | SPIFFY COMPUTER SERVICE DIV. |
| HEMMINGER  | DIAN      | SPIFFY COMPUTER SERVICE DIV. |
| HENDERSON  | EILEEN    | OPERATIONS                   |
| JEFFERSON  | JAMES     | ADMINISTRATION SYSTEMS       |
| JOHN       | REBA      | MANUFACTURING SYSTEMS        |
| JOHNSON    | SYBIL     | ADMINISTRATION SYSTEMS       |
| JONES      | WILLIAM   | MANUFACTURING SYSTEMS        |
| KWAN       | SALLY     | INFORMATION CENTER           |
| LEE        | WING      | SOFTWARE SUPPORT             |
| LUCCHESSI  | VINCENZO  | SPIFFY COMPUTER SERVICE DIV. |
| LUTZ       | JENNIFER  | MANUFACTURING SYSTEMS        |
| MARINO     | SALVATORE | ADMINISTRATION SYSTEMS       |
| MEHTA      | RAMLAL    | SOFTWARE SUPPORT             |
| MONTEVERDE | ROBERT    | ADMINISTRATION SYSTEMS       |
| NATZ       | KIM       | INFORMATION CENTER           |
```



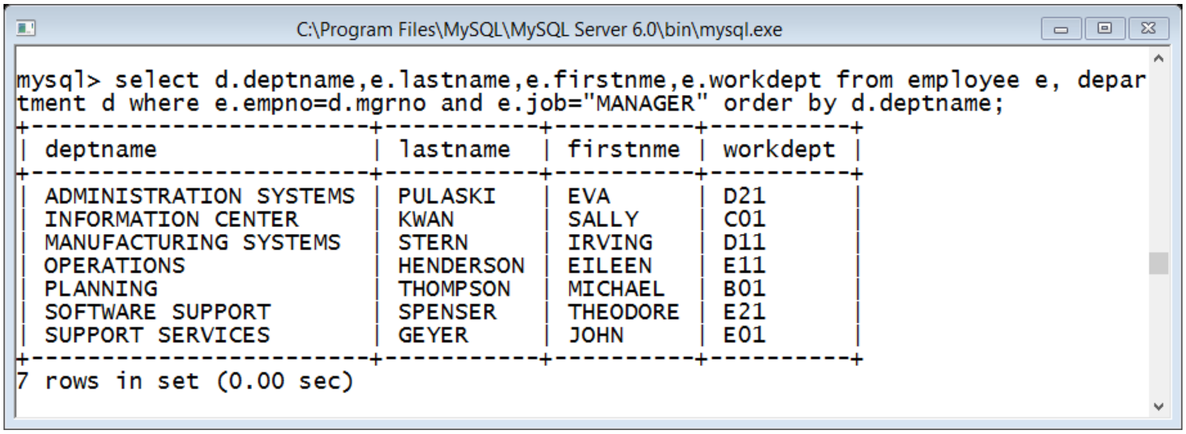
```
| NATZ       | KIM       | INFORMATION CENTER           |
| NICHOLLS   | HEATHER   | INFORMATION CENTER           |
| O CONNELL  | SEAN      | SPIFFY COMPUTER SERVICE DIV. |
| ORLANDO    | GREG      | SPIFFY COMPUTER SERVICE DIV. |
| PARKER     | JOHN      | OPERATIONS                   |
| PEREZ      | MARIA     | ADMINISTRATION SYSTEMS       |
| PIANKA     | ELIZABETH | MANUFACTURING SYSTEMS        |
| PULASKI    | EVA       | ADMINISTRATION SYSTEMS       |
| QUINTANA   | DELORES   | INFORMATION CENTER           |
| SCHNEIDER  | ETHEL     | OPERATIONS                   |
| SCHWARTZ   | EILEEN    | OPERATIONS                   |
| SCOUTTEN   | MARILYN   | MANUFACTURING SYSTEMS        |
| SETRIGHT   | MAUDE     | OPERATIONS                   |
| SMITH      | DANIEL    | ADMINISTRATION SYSTEMS       |
| SMITH      | PHILIP    | OPERATIONS                   |
| SPENSER    | THEODORE  | SOFTWARE SUPPORT             |
| SPRINGER   | MICHELLE  | OPERATIONS                   |
| STERN      | IRVING    | MANUFACTURING SYSTEMS        |
| THOMPSON   | MICHAEL   | PLANNING                     |
| WALKER     | JAMES     | MANUFACTURING SYSTEMS        |
| WONG       | HELENA    | SOFTWARE SUPPORT             |
| YAMAMOTO   | KIYOSHI   | MANUFACTURING SYSTEMS        |
| YOSHIMURA  | MASATOSHI | MANUFACTURING SYSTEMS        |
+------------+-----------+------------------------------+
42 rows in set (0.38 sec)
```

2. mysql> select e.lastname,e.firstnme,e.job,d.deptname from employee e, department d where e.workdept=d.deptno and d.deptno Between "A02" and "D22" and job<>"MANAGER" order by e.lastname,d.deptname;



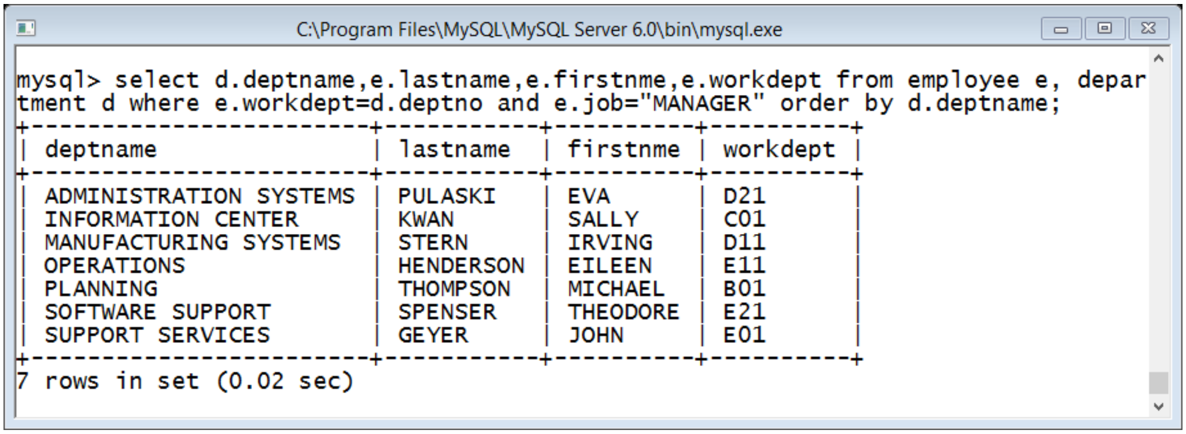
```
mysql> select e.lastname,e.firstnme,e.job,d.deptname from employee e, department
 d where e.workdept=d.deptno and d.deptno Between "A02" and "D22" and job<>"MANA
GER" order by e.lastname,d.deptname;
+------------+-----------+----------+------------------------+
| lastname   | firstnme  | job      | deptname               |
+------------+-----------+----------+------------------------+
| ADAMSON    | BRUCE     | DESIGNER | MANUFACTURING SYSTEMS  |
| BROWN      | DAVID     | DESIGNER | MANUFACTURING SYSTEMS  |
| JEFFERSON  | JAMES     | CLERK    | ADMINISTRATION SYSTEMS |
| JOHN       | REBA      | DESIGNER | MANUFACTURING SYSTEMS  |
| JOHNSON    | SYBIL     | CLERK    | ADMINISTRATION SYSTEMS |
| JONES      | WILLIAM   | DESIGNER | MANUFACTURING SYSTEMS  |
| LUTZ       | JENNIFER  | DESIGNER | MANUFACTURING SYSTEMS  |
| MARINO     | SALVATORE | CLERK    | ADMINISTRATION SYSTEMS |
| MONTEVERDE | ROBERT    | CLERK    | ADMINISTRATION SYSTEMS |
| NATZ       | KIM       | ANALYST  | INFORMATION CENTER     |
| NICHOLLS   | HEATHER   | ANALYST  | INFORMATION CENTER     |
| PEREZ      | MARIA     | CLERK    | ADMINISTRATION SYSTEMS |
| PIANKA     | ELIZABETH | DESIGNER | MANUFACTURING SYSTEMS  |
| QUINTANA   | DELORES   | ANALYST  | INFORMATION CENTER     |
| SCOUTTEN   | MARILYN   | DESIGNER | MANUFACTURING SYSTEMS  |
| SMITH      | DANIEL    | CLERK    | ADMINISTRATION SYSTEMS |
| WALKER     | JAMES     | DESIGNER | MANUFACTURING SYSTEMS  |
| YAMAMOTO   | KIYOSHI   | DESIGNER | MANUFACTURING SYSTEMS  |
| YOSHIMURA  | MASATOSHI | DESIGNER | MANUFACTURING SYSTEMS  |
+------------+-----------+----------+------------------------+
19 rows in set (0.04 sec)
```

3. select d.deptname,e.lastname,e.firstnme,e.workdept from employee e, department d where e.empno=d.mgrno and e.job="MANAGER" order by d.deptname;



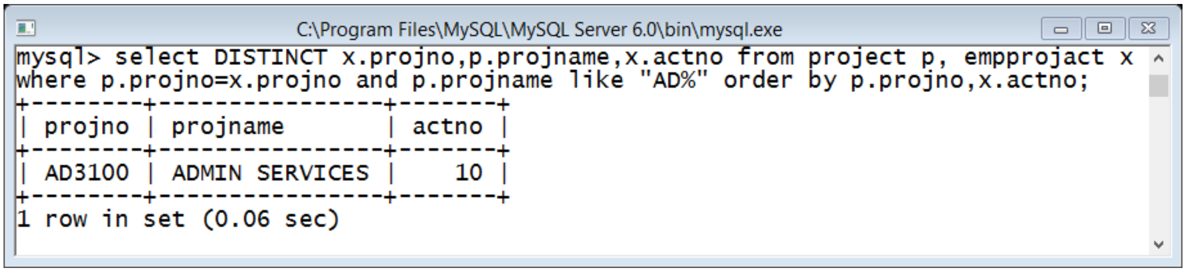
```

mysql> select d.deptname,e.lastname,e.firstnme,e.workdept from employee e, depar
tment d where e.empno=d.mgrno and e.job="MANAGER" order by d.deptname;
+------------------------+-----------+----------+----------+
| deptname               | lastname  | firstnme | workdept |
+------------------------+-----------+----------+----------+
| ADMINISTRATION SYSTEMS | PULASKI   | EVA      | D21      |
| INFORMATION CENTER     | KWAN      | SALLY    | C01      |
| MANUFACTURING SYSTEMS  | STERN     | IRVING   | D11      |
| OPERATIONS             | HENDERSON | EILEEN   | E11      |
| PLANNING               | THOMPSON  | MICHAEL  | B01      |
| SOFTWARE SUPPORT       | SPENSER   | THEODORE | E21      |
| SUPPORT SERVICES       | GEYER     | JOHN     | E01      |
+------------------------+-----------+----------+----------+
7 rows in set (0.00 sec)
```

4. select d.deptname,e.lastname,e.firstnme,e.workdept from employee e, department d where e.workdept=d.deptno and e.job="MANAGER" order by d.deptname;



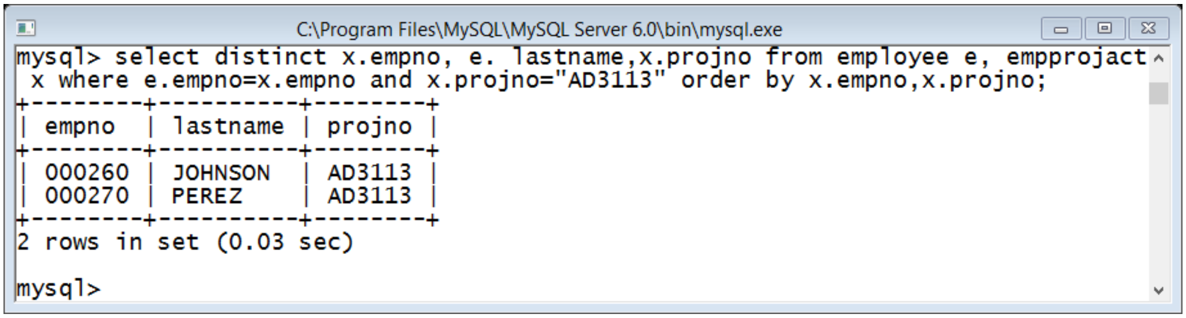
```
mysql> select d.deptname,e.lastname,e.firstnme,e.workdept from employee e, depar
tment d where e.workdept=d.deptno and e.job="MANAGER" order by d.deptname;
+------------------------+-----------+----------+----------+
| deptname               | lastname  | firstnme | workdept |
+------------------------+-----------+----------+----------+
| ADMINISTRATION SYSTEMS | PULASKI   | EVA      | D21      |
| INFORMATION CENTER     | KWAN      | SALLY    | C01      |
| MANUFACTURING SYSTEMS  | STERN     | IRVING   | D11      |
| OPERATIONS             | HENDERSON | EILEEN   | E11      |
| PLANNING               | THOMPSON  | MICHAEL  | B01      |
| SOFTWARE SUPPORT       | SPENSER   | THEODORE | E21      |
| SUPPORT SERVICES       | GEYER     | JOHN     | E01      |
+------------------------+-----------+----------+----------+
7 rows in set (0.02 sec)
```

5. select DISTINCT x.projno,p.projname,x.actno from project p, empprojact x where p.projno=x.projno and p.projname like "AD%" order by p.projno,x.actno;



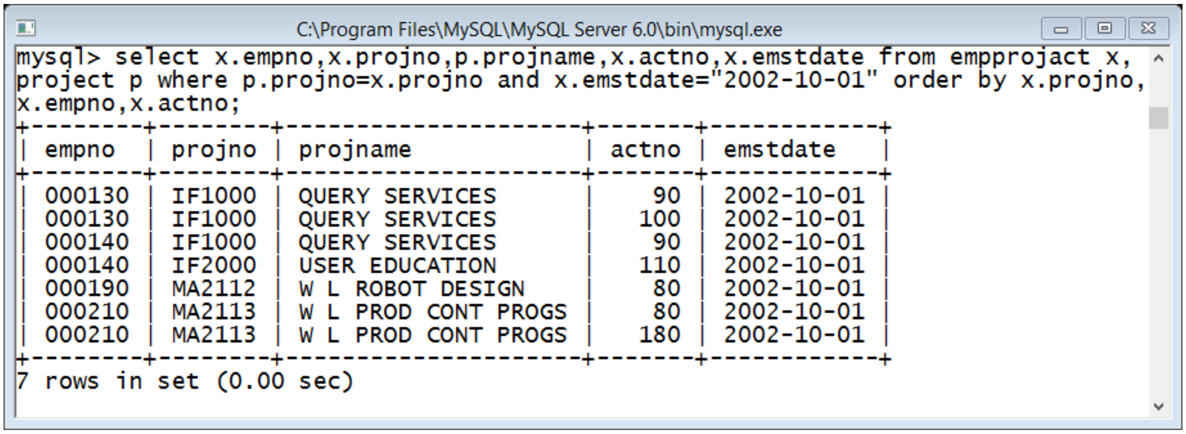
```
mysql> select DISTINCT x.projno,p.projname,x.actno from project p, empprojact x
where p.projno=x.projno and p.projname like "AD%" order by p.projno,x.actno;
+--------+----------------+-------+
| projno | projname       | actno |
+--------+----------------+-------+
| AD3100 | ADMIN SERVICES |    10 |
+--------+----------------+-------+
1 row in set (0.06 sec)
```

6. select distinct x.empno, e. lastname,x.projno from employee e, empprojact x where e.empno=x.empno and x.projno="AD3113" order by x.empno,x.projno;



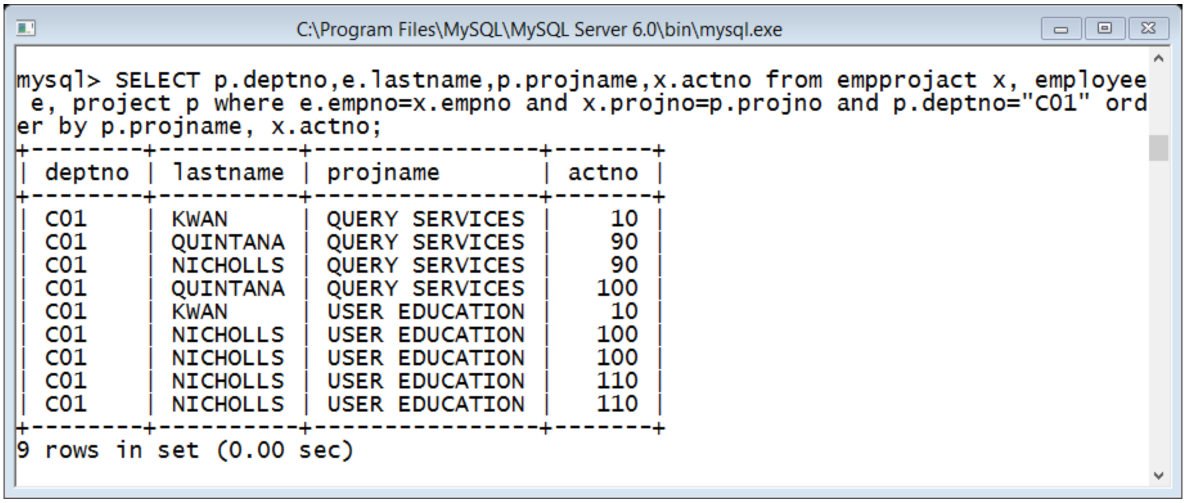
```
mysql> select distinct x.empno, e. lastname,x.projno from employee e, empprojact
 x where e.empno=x.empno and x.projno="AD3113" order by x.empno,x.projno;
+--------+----------+--------+
| empno  | lastname | projno |
+--------+----------+--------+
| 000260 | JOHNSON  | AD3113 |
| 000270 | PEREZ    | AD3113 |
+--------+----------+--------+
2 rows in set (0.03 sec)

mysql>
```

7. select x.empno,x.projno,p.projname,x.actno,x.emstdate from empprojact x, project p where p.projno=x.projno and x.emstdate="2002-10-01" order by x.projno,x.empno,x.actno;



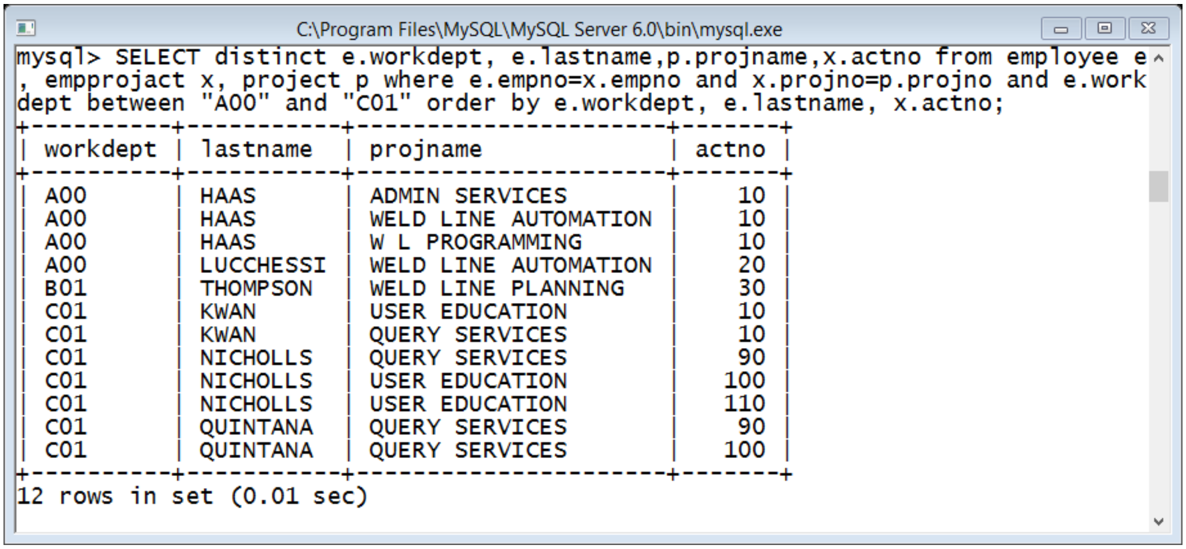
```
mysql> select x.empno,x.projno,p.projname,x.actno,x.emstdate from empprojact x,
project p where p.projno=x.projno and x.emstdate="2002-10-01" order by x.projno,
x.empno,x.actno;
+--------+--------+---------------------+-------+------------+
| empno  | projno | projname            | actno | emstdate   |
+--------+--------+---------------------+-------+------------+
| 000130 | IF1000 | QUERY SERVICES      |    90 | 2002-10-01 |
| 000130 | IF1000 | QUERY SERVICES      |   100 | 2002-10-01 |
| 000140 | IF1000 | QUERY SERVICES      |    90 | 2002-10-01 |
| 000140 | IF2000 | USER EDUCATION      |   110 | 2002-10-01 |
| 000190 | MA2112 | W L ROBOT DESIGN    |    80 | 2002-10-01 |
| 000210 | MA2113 | W L PROD CONT PROGS |    80 | 2002-10-01 |
| 000210 | MA2113 | W L PROD CONT PROGS |   180 | 2002-10-01 |
+--------+--------+---------------------+-------+------------+
7 rows in set (0.00 sec)
```

8. select p.deptno,e.lastname,p.projname,x.actno from empprojact x, employee e, project p where e.empno=x.empno and x.projno=p.projno and p.deptno="C01" order by p.projname, x.actno;



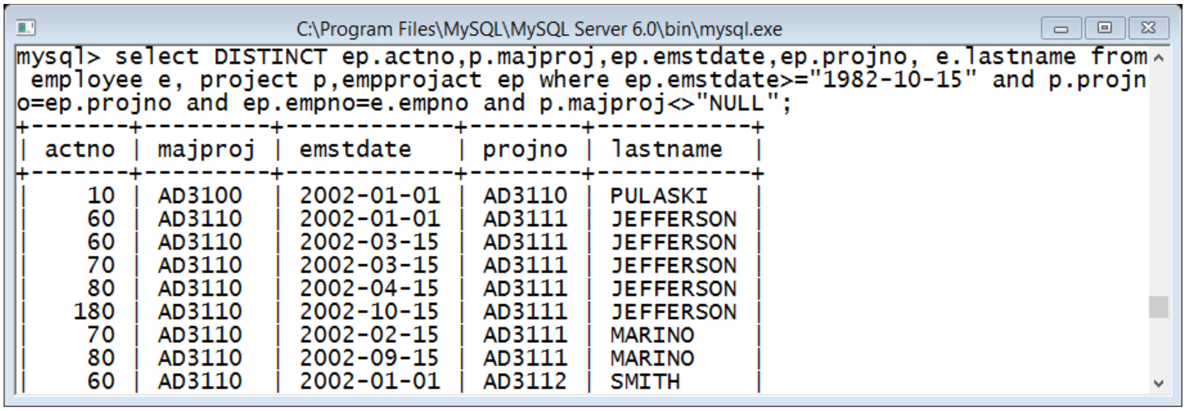
```
mysql> SELECT p.deptno,e.lastname,p.projname,x.actno from empprojact x, employee
 e, project p where e.empno=x.empno and x.projno=p.projno and p.deptno="C01" ord
er by p.projname, x.actno;
+--------+----------+----------------+-------+
| deptno | lastname | projname       | actno |
+--------+----------+----------------+-------+
| C01    | KWAN     | QUERY SERVICES |    10 |
| C01    | QUINTANA | QUERY SERVICES |    90 |
| C01    | NICHOLLS | QUERY SERVICES |    90 |
| C01    | QUINTANA | QUERY SERVICES |   100 |
| C01    | KWAN     | USER EDUCATION |    10 |
| C01    | NICHOLLS | USER EDUCATION |   100 |
| C01    | NICHOLLS | USER EDUCATION |   100 |
| C01    | NICHOLLS | USER EDUCATION |   110 |
| C01    | NICHOLLS | USER EDUCATION |   110 |
+--------+----------+----------------+-------+
9 rows in set (0.00 sec)
```

9. SELECT distinct e.workdept, e.lastname,p.projname,x.actno from employee e, empprojact x, project p where e.empno=x.empno and x.projno=p.projno and e.workdept between "A00" and "C01" order by e.workdept, e.lastname, x.actno;



```
mysql> SELECT distinct e.workdept, e.lastname,p.projname,x.actno from employee e
, empprojact x, project p where e.empno=x.empno and x.projno=p.projno and e.work
dept between "A00" and "C01" order by e.workdept, e.lastname, x.actno;
+----------+-----------+----------------------+-------+
| workdept | lastname  | projname             | actno |
+----------+-----------+----------------------+-------+
| A00      | HAAS      | ADMIN SERVICES       |    10 |
| A00      | HAAS      | WELD LINE AUTOMATION |    10 |
| A00      | HAAS      | W L PROGRAMMING      |    10 |
| A00      | LUCCHESSI | WELD LINE AUTOMATION |    20 |
| B01      | THOMPSON  | WELD LINE PLANNING   |    30 |
| C01      | KWAN      | USER EDUCATION       |    10 |
| C01      | KWAN      | QUERY SERVICES       |    10 |
| C01      | NICHOLLS  | QUERY SERVICES       |    90 |
| C01      | NICHOLLS  | USER EDUCATION       |   100 |
| C01      | NICHOLLS  | USER EDUCATION       |   110 |
| C01      | QUINTANA  | QUERY SERVICES       |    90 |
| C01      | QUINTANA  | QUERY SERVICES       |   100 |
+----------+-----------+----------------------+-------+
12 rows in set (0.01 sec)
```

10. select DISTINCT ep.actno,p.majproj,ep.emstdate,ep.projno, e.lastname from employee e, project p,empprojact ep where ep.emstdate>="1982-10-15" and p.projno=ep.projno and ep.empno=e.empno and p.majproj<>"NULL";



```
mysql> select DISTINCT ep.actno,p.majproj,ep.emstdate,ep.projno, e.lastname from
 employee e, project p,empprojact ep where ep.emstdate>="1982-10-15" and p.projn
o=ep.projno and ep.empno=e.empno and p.majproj<>"NULL";
+-------+---------+------------+--------+-----------+
| actno | majproj | emstdate   | projno | lastname  |
+-------+---------+------------+--------+-----------+
|    10 | AD3100  | 2002-01-01 | AD3110 | PULASKI   |
|    60 | AD3110  | 2002-01-01 | AD3111 | JEFFERSON |
|    60 | AD3110  | 2002-03-15 | AD3111 | JEFFERSON |
|    70 | AD3110  | 2002-03-15 | AD3111 | JEFFERSON |
|    80 | AD3110  | 2002-04-15 | AD3111 | JEFFERSON |
|   180 | AD3110  | 2002-10-15 | AD3111 | JEFFERSON |
|    70 | AD3110  | 2002-02-15 | AD3111 | MARINO    |
|    80 | AD3110  | 2002-09-15 | AD3111 | MARINO    |
|    60 | AD3110  | 2002-01-01 | AD3112 | SMITH     |
```

4. Conclusión

Esta práctica fue más difícil que las anteriores ya que las consultas debían de ser más elaboradas, otro factor es que no conocía la base de datos y fue difícil saber qué campo hacía referencia a los datos que se solicitaban, pero esta práctica fue muy buena porque no hizo practicar más los comandos y a analizar las consultas que nos piden lo cual es de gran importancia para obtener la información adecuada y no estar cometiendo errores y obtener información errónea.

5. Bibliografía

1. *13.2.7. Sintaxis de SELECT.* Recuperado 01 de septiembre de 2013. [En línea]. Disponible en: http://dev.mysql.com/doc/refman/5.0/es/select.html.
2. *13.2.4. Sintaxis de INSERT.* Recuperado 01 de septiembre de 2013. [En línea]. Disponible en: http://dev.mysql.com/doc/refman/5.0/es/insert.html.