Check the following database:

DEPARTMENTS:

num	name	town_code
10	ACCOUNTING	SVQ
20	RESEARCH	MAD
30	SALES	BCN
40	PRODUCTION	BIO

EMPLOYEES:

num	surname	name	manager	start_date	salary	commission	dept_num	occu_code
800	BANDERAS	ANTONIO	7839	1991-01-09	2885	NULL	20	MAN
7369	SÁNCHEZ	SERGIO	7902	1990-12-17	1040	NULL	20	EMP
7499	ARROYO	MARTA	7698	1990-02-20	1500	390	30	SAL
7521	SALA	RAUL	7698	1991-02-22	1625	650	30	SAL
7566	JIMÉNEZ	JUDIT	7839	1991-04-02	2900	NULL	20	MAN
7654	MARTÍN	MONICA	7698	1991-09-29	1600	1020	30	SAL
7698	NEGRO	BARTOLOME	7839	1991-05-01	3005	NULL	30	MAN
7782	CEREZO	ENRIQUE	7839	1991-06-09	2885	NULL	10	MAN
7788	GIL	JESUS	7566	1991-11-09	3000	NULL	20	ANA
7844	TOVAR	LUIS	7698	1991-09-08	1350	0	30	SAL
7876	ALONSO	FERNANDO	7788	1991-09-23	1430	NULL	20	EMP
7900	JIMENO	XAVIER	7698	1991-12-03	1335	NULL	30	EMP
7902	FERNÁNDEZ	ANA	7566	1991-12-03	3000	NULL	20	ANA
7934	MUÑOZ	ANTONIA	7782	1992-01-23	1690	NULL	10	EMP
8001	RUIZ	FERNANDA	7839	1992-06-10	2885	NULL	20	MAN

OCCUPATIONS:

code	name
ANA	ANALYST
EMP	EMPLOYEE
MAN	MANAGER
PRE	PRESIDENT
SAL	SALESMAN

TOWNS:

code	name
BCN	BARCELONA
BIO	BILBAO
MAD	MADRID
SVQ	SEVILLA

Import the next database:

```
CREATE DATABASE IF NOT EXISTS `EMPLOYEESDBNORMAL`;
USE `EMPLOYEESDBNORMAL`;

CREATE TABLE IF NOT EXISTS `DEPARTMENTS` (
  `num` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
  `town_code` varchar(3) DEFAULT NULL,
  PRIMARY KEY (`num`),
  KEY `town_code` (`town_code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `DEPARTMENTS` (`num`, `name`, `town_code`) VALUES
(10, 'ACCOUNTING', 'SVQ'),
(20, 'RESEARCH', 'MAD'),
(30, 'SALES', 'BCN'),
(40, 'PRODUCTION', 'BIO');

CREATE TABLE IF NOT EXISTS `EMPLOYEES` (
  `num` int(11) NOT NULL,
  `surname` varchar(50) NOT NULL,
  `name` varchar(50) NOT NULL,
  `manager` int(11) DEFAULT NULL,
  `start_date` date DEFAULT NULL,
  `salary` int(11) DEFAULT NULL,
  `commission` int(11) DEFAULT NULL,
  `dept_num` int(11) DEFAULT NULL,
  `occu_code` varchar(3) DEFAULT NULL,
  PRIMARY KEY (`num`),
  KEY `dept_num` (`dept_num`),
  KEY `occu_code` (`occu_code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `EMPLOYEES` (`num`, `surname`, `name`, `manager`, `start_date`, `salary`,
`commission`, `dept_num`, `occu_code`) VALUES
(800, 'BANDERAS', 'ANTONIO', 7839, '1991-01-09', 2885, NULL, 20, 'MAN'),
(7369, 'SÁNCHEZ', 'SERGIO', 7902, '1990-12-17', 1040, NULL, 20, 'EMP'),
(7499, 'ARROYO', 'MARTA', 7698, '1990-02-20', 1500, 390, 30, 'SAL'),
(7521, 'SALA', 'RAUL', 7698, '1991-02-22', 1625, 650, 30, 'SAL'),
(7566, 'JIMÉNEZ', 'JUDIT', 7839, '1991-04-02', 2900, NULL, 20, 'MAN'),
(7654, 'MARTÍN', 'MONICA', 7698, '1991-09-29', 1600, 1020, 30, 'SAL'),
(7698, 'NEGRO', 'BARTOLOME', 7839, '1991-05-01', 3005, NULL, 30, 'MAN'),
(7782, 'CEREZO', 'ENRIQUE', 7839, '1991-06-09', 2885, NULL, 10, 'MAN'),
(7788, 'GIL', 'JESUS', 7566, '1991-11-09', 3000, NULL, 20, 'ANA'),
(7844, 'TOVAR', 'LUIS', 7698, '1991-09-08', 1350, 0, 30, 'SAL'),
(7876, 'ALONSO', 'FERNANDO', 7788, '1991-09-23', 1430, NULL, 20, 'EMP'),
(7900, 'JIMENO', 'XAVIER', 7698, '1991-12-03', 1335, NULL, 30, 'EMP'),
(7902, 'FERNÁNDEZ', 'ANA', 7566, '1991-12-03', 3000, NULL, 20, 'ANA'),
(7934, 'MUÑOZ', 'ANTONIA', 7782, '1992-01-23', 1690, NULL, 10, 'EMP'),
(8001, 'RUIZ', 'FERNANDA', 7839, '1992-06-10', 2885, NULL, 20, 'MAN');

CREATE TABLE IF NOT EXISTS `OCCUPATIONS` (
  `code` varchar(3) NOT NULL,
  `name` varchar(30) NOT NULL,
  PRIMARY KEY (`code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `OCCUPATIONS` (`code`, `name`) VALUES
('ANA', 'ANALYST'),
('EMP', 'EMPLOYEE'),
('MAN', 'MANAGER'),
('PRE', 'PRESIDENT'),
('SAL', 'SALESMAN');
```

```
CREATE TABLE IF NOT EXISTS `TOWNS` (
  `code` varchar(3) NOT NULL,
  `name` varchar(30) NOT NULL,
  PRIMARY KEY (`code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `TOWNS` (`code`, `name`) VALUES
('BCN', 'BARCELONA'),
('BIO', 'BILBAO'),
('MAD', 'MADRID'),
('SVQ', 'SEVILLA');

ALTER TABLE `DEPARTMENTS`
  ADD CONSTRAINT `DEPARTMENTS_ibfk_1` FOREIGN KEY (`town_code`) REFERENCES `TOWNS`
(`code`);

ALTER TABLE `EMPLOYEES`
  ADD CONSTRAINT `EMPLOYEES_ibfk_1` FOREIGN KEY (`dept_num`) REFERENCES `DEPARTMENTS`
(`num`),
  ADD CONSTRAINT `EMPLOYEES_ibfk_2` FOREIGN KEY (`occu_code`) REFERENCES `OCCUPATIONS`
(`code`);
```

Do the following queries with that database:

1. Display the number of employees in each department. Use GROUP BY to group by department.

```
+----------+-------------+
| dept_num | N_employees |
+----------+-------------+
|       10 |           2 |
|       20 |           7 |
|       30 |           6 |
+----------+-------------+
3 rows in set (0.001 sec)
```

2. For each occupation obtain the average of salary.

```
+----------+----------------+
| name     | average_salary |
+----------+----------------+
| ANALYST  |      3000.0000 |
| EMPLOYEE |      1373.7500 |
| MANAGER  |      2912.0000 |
| SALESMAN |      1518.7500 |
+----------+----------------+
4 rows in set (0.001 sec)
```

3. Display the departments with more than 5 employees. Use GROUP BY to group by department and HAVING to establish the condition on the groups.

```
+----------+---------------+
| dept_num | num_employees |
+----------+---------------+
|       20 |             7 |
|       30 |             6 |
+----------+---------------+
2 rows in set (0.001 sec)
```

4. Find the average wages (=”*media de los salarios*”) of each department (use the function avg and GROUP BY).

```
+----------+---------------+
| dept_num | average_wages |
+----------+---------------+
|       10 |     2287.5000 |
|       20 |     2448.5714 |
|       30 |     1735.8333 |
+----------+---------------+
3 rows in set (0.002 sec)
```

5. Display the surname of the salesmen of the 'SALES' department.

```
+---------+
| surname |
+---------+
| ARROYO  |
| SALA    |
| MARTÍN  |
| TOVAR   |
+---------+
4 rows in set (0.001 sec)
```

6. Display the sum of salaries of the 'SALES' department.

```
+-------+-------+
| name  | total |
+-------+-------+
| SALES | 10415 |
+-------+-------+
1 row in set (0.001 sec)
```

7. Display the count of employees with occupation “EMPLOYEE” in every department (show the name of the department).

```
+------------+-----+
| name       | num |
+------------+-----+
| ACCOUNTING |   1 |
| RESEARCH   |   2 |
| SALES      |   1 |
+------------+-----+
3 rows in set (0.001 sec)
```

8. Show the number of different occupations in each department.

```
+------------+------------+---------------------+
| Department | Occupation | Number_of_employees |
+------------+------------+---------------------+
| ACCOUNTING | EMPLOYEE   |                   1 |
| ACCOUNTING | MANAGER    |                   1 |
| RESEARCH   | ANALYST    |                   2 |
| RESEARCH   | EMPLOYEE   |                   2 |
| RESEARCH   | MANAGER    |                   3 |
| SALES      | EMPLOYEE   |                   1 |
| SALES      | MANAGER    |                   1 |
| SALES      | SALESMAN   |                   4 |
+------------+------------+---------------------+
8 rows in set (0.004 sec)
```

9. Show departments that have more than two people working in the same occupation.

```
+------------+
| Department |
+------------+
| RESEARCH   |
| SALES      |
+------------+
2 rows in set (0.002 sec)
```

10. Displays a query that is the union between the table OCCUPATIONS and TOWNS.

```
+------+-----------+
| code | name      |
+------+-----------+
| ANA  | ANALYST   |
| EMP  | EMPLOYEE  |
| MAN  | MANAGER   |
| PRE  | PRESIDENT |
| SAL  | SALESMAN  |
| BCN  | BARCELONA |
| BIO  | BILBAO    |
| MAD  | MADRID    |
| SVQ  | SEVILLA   |
+------+-----------+
9 rows in set (0.001 sec)
```

11. Do the same query than in exercise 10 but order the results by name.

```
+------+-----------+
| code | name      |
+------+-----------+
| ANA  | ANALYST   |
| BCN  | BARCELONA |
| BIO  | BILBAO    |
| EMP  | EMPLOYEE  |
| MAD  | MADRID    |
| MAN  | MANAGER   |
| PRE  | PRESIDENT |
| SAL  | SALESMAN  |
| SVQ  | SEVILLA   |
+------+-----------+
9 rows in set (0.001 sec)
```

12. Select the occupation names of all the employees of the department with name ‘RESEARCH’ and do the union of this query with the selection of the occupation names of the employees of the department with name ‘SALES’. Use union operator.

```
+----------+
| name     |
+----------+
| ANALYST  |
| EMPLOYEE |
| MANAGER  |
| SALESMAN |
+----------+
4 rows in set (0.001 sec)
```

13. Repeat the last query showing the repeated results (union all).

```
+----------+
| name     |
+----------+
| ANALYST  |
| ANALYST  |
| EMPLOYEE |
| EMPLOYEE |
| MANAGER  |
| MANAGER  |
| MANAGER  |
| EMPLOYEE |
| MANAGER  |
| SALESMAN |
| SALESMAN |
| SALESMAN |
| SALESMAN |
+----------+
13 rows in set (0.001 sec)
```

14. Display the number of sellers in the 'SALES' department.

```
+--------------------+
| number_of_sellers |
+--------------------+
|                  4 |
+--------------------+
1 row in set (0.001 sec)
```

15. Display the surnames and occupations of the employees of the 'SALES' department.

```
+---------+----------+
| surname | name     |
+---------+----------+
| JIMENO  | EMPLOYEE |
| NEGRO   | MANAGER  |
| ARROYO  | SALESMAN |
| SALA    | SALESMAN |
| MARTÍN  | SALESMAN |
| TOVAR   | SALESMAN |
+---------+----------+
6 rows in set (0.001 sec)
```

16. Display the number of employees and occupations of the employees of the 'SALES' department.

```
+----------+--------------------+
| name     | number_of_employees |
+----------+--------------------+
| EMPLOYEE |                  1 |
| MANAGER  |                  1 |
| SALESMAN |                  4 |
+----------+--------------------+
3 rows in set (0.001 sec)
```

17. Display the number of employees of each department whose profession is "EMPLOYEE".

```
+------------+--------------------+
| name       | number_of_employees |
+------------+--------------------+
| ACCOUNTING |                  1 |
| RESEARCH   |                  2 |
| SALES      |                  1 |
+------------+--------------------+
3 rows in set (0.001 sec)
```

18. Display de department names and the count of employees working into them.

```
+------------+----------------------+
| name       | number_of_employees |
+------------+----------------------+
| ACCOUNTING |                    2 |
| RESEARCH   |                    7 |
| SALES      |                    6 |
+------------+----------------------+
3 rows in set (0.001 sec)
```

19. Display the maximum number of employees of all the departments (clue: you need exercise 18 as a subquery and you should use MAX function).

```
+------------+
| max_number |
+------------+
|          7 |
+------------+
1 row in set (0.001 sec)
```

20. Show the departments whose average salary is greater than the average of salaries of all employees.

```
+----------+----------------+
| dept_num | average_salary |
+----------+----------------+
|       10 |      2287.5000 |
|       20 |      2448.5714 |
+----------+----------------+
2 rows in set (0.001 sec)
```

21. DANGER, this is for PROS: Display the name of the department with more employees and its number of employees (clue you must use HAVING with a subselect inside).

```
+----------+---------------+
| name     | num_employees |
+----------+---------------+
| RESEARCH |             7 |
+----------+---------------+
1 row in set (0.001 sec)
```

22. Repeat 12 changing “union” for “intersect”.

```
+----------+
| name     |
+----------+
| EMPLOYEE |
| MANAGER  |
+----------+
2 rows in set (0.002 sec)
```

23. Repeat 22 but do not use intersect operator to query the same data (clue: IN operator).

```
+----------+
| name     |
+----------+
| EMPLOYEE |
| MANAGER  |
+----------+
2 rows in set (0.002 sec)
```