**ЗАПРОСЫ**

**https://learntutorials.net/ru/mongodb/topic/9271/%d0%b7%d0%b0%d0%bf%d1%80%d0%be%d1%81-%d0%b4%d0%b0%d0%bd%d0%bd%d1%8b%d1%85--%d0%bd%d0%b0%d1%87%d0%b0%d0%bb%d0%be-%d1%80%d0%b0%d0%b1%d0%be%d1%82%d1%8b-**

**ЗАДАНИЕ: выполнить запросы в среде SQL и в среде MongoDB.**

**(см. файлы:**

**4\_5sql запросы по EMP\_1.pdf**

**4\_6sql запросы по EMP\_2.pdf )**

|  |  |
| --- | --- |
|  | **SQL** |
| 1. Напишите запрос, который выводит номера сотрудников, их оклады и даты приёма на работу | select empno, sal, hiredate from emp; |
| 2. Напишите запрос, который выводит номера и названия отделов | select deptno, dname from dept; |
| 3. Перечислите имена сотрудников и их оклад, увеличенный на 15% и округленный до целого числа долларов | select ename, sal\*1.15 from emp; |
| 4. Известно, что в некоторые из продавцов — женщины (поэтому для них название SALESMAN записанное в таблице EMP не уместно). Поменяйте в результате вывода окончания MAN на PERSON, так чтобы получилось SALESPERSON | select empno, ename, replace (job, 'MAN', 'PERSON') as job from emp; |
| 5. Выведите фамилии сотрудников, принятых на работу в сентябре. (Смотрите функцию TO\_CHAR, которая преобразует дату в строковый формат) | select ename from emp where to\_char(hiredate, 'mm') = 9; |
| 6. Выдайте номера (DEPTNO) и названия (DNAME) всех отделов в алфавитном порядке  Найдите всех сотрудников с окладом в интервале от 1000 до 2000, включительно | select deptno, dname, ename, sal from emp join dept using (deptno) where sal >= 1000 AND sal <=2000 order by dname; |
| 7. Извлеките полную информацию о всех сотрудниках в отделах 10 и 20 так чтобы их фамилии шли в алфавитном порядке | select ename, deptno from emp where deptno = 10 or deptno = 20 order by ename; |
| 8. Выведите фамилии и должности всех клерков в отделе 20. | select ename from emp where job = 'CLERK' and deptno = 20; |
| 9. Найдите всех сотрудников, чьи фамилии содержат сочетания 'ТН' или 'LL'. | SELECT ENAME FROM EMP WHERE ENAME LIKE '%TH%' OR ENAME LIKE '%LL%'; |
| 10. Выведите фамилию, должность и оклад сотрудников, у которых есть руководитель. | select ename,job,sal from emp where mgr is not null; |
| 11. В таблице EMP найдите фамилию с двумя буквами «А» | SELECT ENAME FROM EMP WHERE ENAME LIKE '%A%A%'; |
| 12. Выдайте фамилии, годовой доход и комиссионное вознаграждение всех продавцов, у которых месячный оклад превышает их комиссионные. Расположите их в порядке убывания окладов, если встретится несколько сотрудников с одинаковым окладом они должны быть перечислены в алфавитном порядке. | select ename, sal\*12, comm from emp where sal > comm AND job = 'SALESMAN' order by sal\*12 DESC, ename ASC; |
| ………………… | ……………. |
| N. | N. |
| **Задание (2 пара)**  1). Напишите запрос, который выводит фамилии сотрудников, работающих в городе Нью-Йорке. Для этого вам пригодится таблица DEPT  2). Напишите запрос, который выводит фамилии сотрудников, которые получают больше, чем их непосредственные руководители  3). Напишите запрос, который выводит фамилии сотрудников из отделов, где работают больше трёх человек | |

|  |  |
| --- | --- |
| **SQL** | **MongoDb** |
| select \* from emp; | db.emp.find({}); |
| select \* from dept; | db.dept.find({}); |
| select ename, sal from emp; | db.emp.find({}, {ename: 1, sal: 1}); |
| Select ename from emp; | Db.emp.find({}, {ename: 1, \_id: 0}); |
| Select \* from emp where ename = ‘SMITH’; | db.emp.find({ename: ‘SMITH’}); |
| Select \* from emp where empno in (7369, 15, 46, 16); | db.emp.find({empno: { $in: [7369, 15, 46, 16]}}); |
| Select dept.deptno, emp.empno from dept join emp on dept.deptno = emp.deptno where dname = ‘ACCOUNTING’; | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $match: { dname: "ACCOUNTING" } },  { $project: { deptno: 1, "emp\_details.empno": 1, \_id: 0 } }  ]); |
| Select dept.deptno, emp.empno from dept join emp on dept.deptno = emp.deptno where dname = ‘WORKS’; | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $match: { dname: "WORKS" } },  { $project: { deptno: 1, "emp\_details.empno": 1, \_id: 0 } }  ]); |
| Select \* from dept join emp on dept.deptno = emp.deptno where dname = ‘ACCOUNTING’; | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $match: { dname: "ACCOUNTING" } }  ]); |
| Select \* from dept join emp on dept.deptno = emp.deptno where dname = ‘RESEARCH; | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $match: { dname: "RESEARCH" } }  ]); |
| Select \* from emp join dept on emp.empno = 7369 and dept.deptno = 20; | db.emp.aggregate([  {  $lookup: {  from: "dept",  localField: "deptno",  foreignField: "deptno",  as: "dept\_details"  }  },  { $match: { empno: 7369, "dept\_details.deptno": 20 } }  ]); |
| select a.deptno, b.ename from dept a, emp b | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $project: { deptno: 1, "emp\_details.ename": 1, \_id: 0 } }  ]); |
| Select a, b from dept a, emp b; | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  }  ]); |
| select a.deptno, b.ename from dept a, emp b | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $project: { deptno: 1, "emp\_details.ename": 1, \_id: 0 } }  ]); |
| Select \* from dept join emp on dept.deptno = emp.deptno where emp.ename = ‘SMITH’; | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $match: { "emp\_details.ename": "SMITH" } }  ]); |
| **ВЫБОРКА(SELECT) (σ):**  SELECT \* from EMP WHERE SAL>2500; | db.emp.find({ sal: { $gt: 2500 } }); |
| **ПРОЕКЦИЯ(PROJECT) (π):**  SELECT DISTINCT ENAME,DEPTNO from EMP;  **ПРОИЗВЕДЕНИЕ(PRODUCT) (×):**  SELECT \* from EMP,DEPT; | db.emp.aggregate([  { $group: { \_id: { ename: "$ename", deptno: "$deptno" } } },  { $project: { ename: "$\_id.ename", deptno: "$\_id.deptno", \_id: 0 } }  ]);  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  db.emp.aggregate([  {  $lookup: {  from: "dept",  localField: "deptno",  foreignField: "deptno",  as: "dept\_details"  }  }  ]); |
| **ОБЪЕДИНЕНИЕ(UNION) (∪):**  SELECT DEPTNO,DNAME, LOC from DEPT;  SELECT ENAME,DEPTNO from EMP UNION SELECT DEPTNO,DNAME, LOC from DEPT;  SELECT DEPTNO from EMP UNION SELECT DEPTNO from DEPT; | db.dept.find({}, { deptno: 1, dname: 1, loc: 1, \_id: 0 });  db.emp.aggregate([  { $project: { ename: 1, deptno: 1, \_id: 0 } },  { $unionWith: {  coll: "dept",  pipeline: [  { $project: { deptno: 1, dname: 1, loc: 1, \_id: 0 } }  ]  }  }  ]);  db.emp.aggregate([  { $group: { \_id: "$deptno" } },  { $project: { deptno: "$\_id", \_id: 0 } },  { $unionWith: {  coll: "dept",  pipeline: [  { $group: { \_id: "$deptno" } },  { $project: { deptno: "$\_id", \_id: 0 } }  ]  }  },  { $group: { \_id: "$deptno" } }, // Убираем дубликаты  { $project: { deptno: "$\_id", \_id: 0 } }  ]); |
| **ПЕРЕСЕЧЕНИЕ(INTERSECT) (∩):**  SELECT DEPTNO from EMP INTERSECT SELECT DEPTNO from DEPT; | db.emp.aggregate([  { $group: { \_id: "$deptno" } },  { $project: { deptno: "$\_id", \_id: 0 } },  { $lookup: {  from: "dept",  localField: "deptno",  foreignField: "deptno",  as: "dept\_details"  }  },  { $match: { "dept\_details": { $ne: [] } } }  ]); |
| **СОЕДИНЕНИЕ(JOIN) (∏):**  SELECT \* from DEPT, EMP WHERE SAL>3000; | db.dept.aggregate([  {  $lookup: {  from: "emp",  localField: "deptno",  foreignField: "deptno",  as: "emp\_details"  }  },  { $match: { "emp\_details.sal": { $gt: 3000 } } }  ]); |