**list all the consultants who are still active (note, in a where clause, where a string match is used this is case sensitive)**sql> select \* from consultant  
where consultant\_status = ‘active’;

**list all the employers**  
sql> select \* from employer

**list all the jobs**  
sql> select \* from job

**display the name, value and expense description for all expenses claimed for contract number 10**  
sql> select expense\_type, expense\_value, expense\_consultant  
from job\_expenses  
where expense\_job = 10;

**add in the expense\_cat table to change the expense type to a word**  
sql> select cat\_desc, expense\_value, expense\_consultant  
from job\_expenses join expense\_cat on cat\_id = expense\_type  
where expense\_job = 10;

**add in the consultant table to swap consultant id for the name**sql> select cat\_desc, expense\_value, consultant\_fname, consultant\_snamefrom job\_expenses join expense\_cat on cat\_id = expense\_typejoin consultant on expense\_consultant = consultant\_idwhere expense\_job = 10;

**which consultants are still leading active contracts (hint: how do you know a contract is still active and not closed?)**  
sql> select consultant\_fname, consultant\_sname  
from contract join consultant on contract\_lead = consultant\_id  
where contract\_end is null;

**display the names of all consultants and their job descriptions**sql> select consultant\_fname, consultant\_sname, job\_desc  
from consultant join job on job\_id = consultant\_job;

**What is the value of the highest expense submitted (does not required grouping but will require a function in select clause)**SQL> select max(expense\_value)  
from job\_expenses;

**Which contract had the lowest value of expenses submitted and approved?**Work out the lowest value of the expensesUse that SQL as a subquery in the where clause (doesn’t need a group by or having because you are not clustering the data, simply finding out what the lowest value is for an expense and then filtering on that)SQL> select expense\_job  
from job\_expenses  
where expense\_value = (select min(expense\_value)from job\_expenses);

**Who was the contract lead on the contract**This needs to expand the previous query by adding in the consultant table and the contract table, add contract then consultant. You need to use the in clause in the where filter to list the sub query. We use in because it may return (and indeed does return more than one value)

**Find the contract ids for the contracts which have had the lowest expenses**SQL> select contract\_id from contract  
where contract\_id in (select expense\_job from job\_expenses  
where expense\_value = (select min(expense\_value)  
from job\_expenses));

**Add in the consultant table and double check the join** (result should be the same contract id but pull in the contract lead, we add the consultant table even through we do nothing with the data to check the join is working correctly)SQL> select contract\_lead, contract\_id  
from contract join consultant on contract\_lead = consultant\_id  
where contract\_id in (select expense\_job from job\_expenses  
where expense\_value = (select min(expense\_value) from job\_expenses));

**Replace the original select with the name attributes**SQL> select consultant\_fname, consultant\_snamefrom contract join consultant on contract\_lead = consultant\_idwhere contract\_id in (select expense\_jobfrom job\_expenseswhere expense\_value = (select min(expense\_value)from job\_expenses));

**Which client has placed the most contracts (may return more than one client name)**SQL> select emp\_name   
from contract join employer on emp\_id = contract\_client  
having count(\*) = (select max(count(\*))  
from contract group by contract\_client)  
group by emp\_name, emp\_id;

**Which client has been charged the most for contracts**First look at the charge\_sheet table this is where the full client bill is heldSQL> desc charge\_sheet;

**Add the charge\_fee and charge\_expenses together to work out the total cost of the charge and group by the job which is also the contract**SQL> select max(sum(charge\_fee + charge\_expenses))  
from charge\_sheet group by charge\_job;

**Add in the join but don’t use the added table, this will just double check the data is the same when the new table is added**SQL> select max(sum(charge\_fee + charge\_expenses))  
from charge\_sheet join contract on charge\_job = contract\_id  
group by charge\_job;

**Change the group by clause to put the data in groups of client rather the job**SQL> select max(sum(charge\_fee + charge\_expenses))from charge\_sheet join contract on charge\_job = contract\_idgroup by contract\_client;

**Add in the employer table to pull back the name of the client**SQL> select emp\_name  
from charge\_sheet join contract on charge\_job = contract\_id  
join employer on contract\_client = emp\_id  
having sum(charge\_fee + charge\_expenses) = (  
select max(sum(charge\_fee + charge\_expenses))  
from charge\_sheet join contract on charge\_job = contract\_id  
group by contract\_client)  
group by emp\_name, emp\_id;

**Identify which active consultants are not leading on a contract**SQL> select count(\*)  
from consultant join contract on consultant\_id = contract\_lead  
where consultant\_status = ‘ACTIVE’;

Count: 21

SQL> select count(\*)  
from consultant left join contract on consultant\_id = contract\_lead  
where consultant\_status = ‘ACTIVE’;

Count: 41

SQL> select count(\*)  
from consultant left join contract on consultant\_id = contract\_lead  
where consultant\_status = ‘ACTIVE’  
and contract\_id is NULL;

Count: 20  
Once the query has been run with a count so you can be sure the data value is correct and filters are working as expected, replace the count(\*) with the name of the consultant (consultant\_fname, consultant\_sname).  
  
**Are there any currently active consultants who have never placed an expense claim? If so how many?**  
SQL> select count(\*) from job\_expenses   
right join consultant on consultant\_id = expense\_consultant  
where expense\_id is null;  
  
COUNT: 22  
**Which contract that have been closed have yet to have a charge sheet generated for them?**select contract\_id, charge\_id  
from contract left join charge\_sheet on contract\_job = charge\_job  
where contract\_end is not null  
and charge\_id is null;  
CONTRACT\_ID CHARGE\_ID  
----------- ----------  
 30  
 3  
 2  
 24