**HR**

**NO.1**

 select employee\_id,first\_name,last\_name,salary,d.department\_name  
 from EMPLOYEES join departments d on EMPLOYEES.department\_id = d.department\_id  
 where d.department\_name like 'F%'  
 order by d.department\_name asc, salary desc ,employee\_id asc

**NO.2**

 select distinct State\_province,city,Street\_address  
 from LOCATIONS  
 where Location\_id not in (select Location\_id from DEPARTMENTS)

**NO.3**

 with aa as (  
 select distinct country\_name,employee\_id,l.country\_id,salary from countries  
 join locations l on COUNTRIES.country\_id = l.country\_id  
 join departments d2 on l.location\_id = d2.location\_id  
 join employees e2 on d2.department\_id = e2.department\_id  
 where state\_province in (select state\_province  
 from locations join DEPARTMENTS D on LOCATIONS.location\_id = D.location\_id  
                 join EMPLOYEES E on D.department\_id = E.department\_id  
 group by state\_province  
 having avg(salary) > 8000) )  
 -- SQLINES LICENSE FOR EVALUATION USE ONLY  
 select country\_name,count(employee\_id), min(salary), max(salary) from aa  
 group by aa.country\_name

**NO.4**

 WITH  
 a as(  
 SELECT EMPLOYEE\_ID,e.DEPARTMENT\_ID ,e.JOB\_ID  
 FROM EMPLOYEES e  
 UNION  
 -- SQLINES LICENSE FOR EVALUATION USE ONLY  
 SELECT EMPLOYEE\_ID,jh.DEPARTMENT\_ID,jh.JOB\_ID  
 FROM JOB\_HISTORY jh),  
 b as(  
 SELECT EMPLOYEE\_ID, count(\*) CNT  
 FROM A  
 GROUP BY employee\_id  
 )  
 -- SQLINES LICENSE FOR EVALUATION USE ONLY  
 SELECT e2.EMPLOYEE\_ID ,e2.FIRST\_NAME ,e2.LAST\_NAME, b.CNT  
 FROM EMPLOYEES e2 JOIN b ON e2.EMPLOYEE\_ID =b.employee\_id  
 WHERE b.CNT>1  
 ORDER BY b.CNT DESC,e2.SALARY ,e2.HIRE\_DATE

**NO.5**

 WITH dep\_to\_reg AS (  
  SELECT regions.region\_id , COUNT(regions.region\_id) AS dep\_count  
  FROM (select temp5.region\_id , Department\_id as dept\_id from departments  
    join (select temp6.region\_id, Location\_id from Locations   
        join (  
            select countries.region\_id, Country\_id from Countries  
            join Regions on Countries.region\_id = Regions.region\_id  
            ) temp6  
            on Locations.Country\_id = temp6.Country\_id  
        ) temp5  
        on Departments.Location\_id = temp5.Location\_id  
        ) temp4  
    left JOIN regions ON temp4.region\_id = regions.region\_id  
    GROUP BY regions.region\_id  
 ),  
 region\_info AS (  
  SELECT regions.region\_name, COUNT(emp.reg\_id) AS reg\_count , nvl(round(AVG(emp.sal), 2), 0) AS sal\_reg  
  FROM (SELECT employees.employee\_id AS emp\_id , employees.department\_id AS dep\_id , employees.salary AS sal , temp1.region\_id AS reg\_id  
  FROM Employees  
  join (select temp2.region\_id, Department\_id from Departments  
    join (select temp3.region\_id, Location\_id from Locations   
        join (  
            select countries.region\_id, Country\_id from Countries  
            join Regions on Countries.region\_id = Regions.region\_id  
            ) temp3  
            on Locations.Country\_id = temp3.Country\_id  
        ) temp2  
        on Departments.Location\_id = temp2.Location\_id  
    ) temp1  
  on Employees.Department\_id = temp1.Department\_id  
  ) emp  
  right JOIN regions ON regions.region\_id = emp.reg\_id  
  GROUP BY regions.region\_name  
  ORDER BY LENGTH(regions.region\_name)  
 )  
 -- SQLINES LICENSE FOR EVALUATION USE ONLY  
 SELECT region\_info.region\_name, region\_info.reg\_count, nvl(dep\_to\_reg.dep\_count, 0), region\_info.sal\_reg  
 FROM region\_info  
 left JOIN dep\_to\_reg ON region\_info.region\_name = (  
  SELECT region\_name FROM regions WHERE regions.region\_id = dep\_to\_reg.region\_id  
 )  
 ​

**NO.6**

 with aa as (  
     select manager\_id,count(employee\_id) empnum from  
 (select employee\_id,manager\_id from employees  
                         where salary>7000) aa  
 group by manager\_id  
 having count(employee\_id) > 4  
 )  
 ,bb as ( select employee\_id,job\_id,first\_name,last\_name,salary from employees  
 join (select avg(salary) avgsal,department\_id from employees  
 group by department\_id) aa on EMPLOYEES.department\_id=aa.department\_id  
 where salary>aa.avgsal\*1.3 )  
 -- SQLINES LICENSE FOR EVALUATION USE ONLY  
 select aa.manager\_id,aa.empnum,concat(concat(bb.first\_name ,' '),bb.last\_name) name,bb.salary,job\_title  
 from aa right join bb on aa.manager\_id=bb.employee\_id  
 join jobs on bb.job\_id = JOBS.job\_id  
 where aa.manager\_id is not null

**NO.7**

 WITH stuff AS (  
  SELECT manager\_id, COUNT(employee\_id) AS emp\_count, AVG(salary) AS avgsal  
  FROM employees  
  WHERE employees.salary > 3000  
  GROUP BY manager\_id  
  HAVING COUNT(employee\_id) > 4  
 ), avg\_sal AS (  
  SELECT department\_name, AVG(salary) AS sal\_avg  
  FROM employees  
  JOIN departments ON employees.department\_id = departments.department\_id  
  GROUP BY department\_name  
 ), stuff\_all AS (  
  SELECT manager\_id, COUNT(employee\_id) AS emp\_count, AVG(salary) AS avgsal  
  FROM employees  
  GROUP BY manager\_id  
  HAVING COUNT(employee\_id) > 4  
 )  
 SELECT employees.employee\_id, CONCAT(CONCAT(employees.first\_name, ' '), employees.last\_name),  
 (SELECT job\_title FROM jobs WHERE employees.job\_id = jobs.job\_id),   
 (SELECT emp\_count FROM stuff WHERE stuff.manager\_id = employees.employee\_id),  
 ROUND((SELECT avgsal FROM stuff\_all WHERE stuff\_all.manager\_id = employees.employee\_id), 2),   
 employees.salary  
 FROM employees  
 WHERE employees.employee\_id IN (  
  SELECT manager\_id FROM stuff  
 )  
 ORDER BY (  
  SELECT sal\_avg FROM avg\_sal WHERE (  
  SELECT department\_name FROM departments WHERE departments.department\_id = employees.department\_id  
  ) = avg\_sal.department\_name  
 ) ASC , employees.salary DESC, employees.employee\_id ASC

**NO.8**

 WITH avg\_sal\_dep AS (  
  SELECT department\_name, AVG(salary) AS sal\_avg  
  FROM employees  
  JOIN departments ON employees.department\_id = departments.department\_id  
  GROUP BY department\_name  
 ), all\_set AS (  
  select REGIONS.region\_id, d.department\_id  
     from regions left join COUNTRIES C on REGIONS.region\_id = C.region\_id  
       left join LOCATIONS L on C.country\_id = L.country\_id  
          left join departments d on L.location\_id = d.location\_id  
             left join employees e on d.department\_id = e.department\_id  
 ), reg\_to\_dep AS (  
  SELECT departments.department\_id, avg\_sal\_dep.sal\_avg, all\_set.region\_id  
  FROM avg\_sal\_dep  
  left JOIN departments ON avg\_sal\_dep.department\_name = departments.department\_name  
  LEFT JOIN all\_set ON all\_set.department\_id = departments.department\_id  
 ), reg\_map AS (  
  SELECT region\_id, MAX(sal\_avg) AS max\_sal  
  FROM reg\_to\_dep  
  GROUP BY region\_id  
 )   
 -- SQLINES LICENSE FOR EVALUATION USE ONLY  
 SELECT employees.employee\_id, CONCAT(CONCAT(employees.first\_name, ' '), employees.last\_name),  
 (SELECT department\_name FROM departments WHERE employees.department\_id = departments.department\_id),   
 (SELECT job\_title FROM jobs WHERE employees.job\_id = jobs.job\_id),  
 employees.salary  
 FROM employees  
 WHERE employees.salary > (SELECT sal\_avg FROM avg\_sal\_dep   
  WHERE (SELECT department\_name FROM departments   
  WHERE employees.department\_id = departments.department\_id  
  ) = avg\_sal\_dep.department\_name  
 ) AND (SELECT sal\_avg FROM avg\_sal\_dep   
  WHERE (SELECT department\_name FROM departments   
  WHERE employees.department\_id = departments.department\_id  
  ) = avg\_sal\_dep.department\_name ) = (  
  SELECT max\_sal FROM reg\_map  
  WHERE reg\_map.region\_id = (  
  SELECT distinct region\_id FROM all\_set  
  WHERE all\_set.department\_id = employees.department\_id  
  )  
  )

**NO.9**

 with aa as (  
 select m.manager\_id, empcount1 from employees e  
     join (  
     select manager\_id, count(employee\_id) empcount1 from employees  
 group by manager\_id  
 having manager\_id is not null  
     ) m on m.manager\_id=e.employee\_id)  
 ,bb as (  
 select e.manager\_id,sum(empcount1) empcount2 from employees e  
     join aa on aa.manager\_id=e.employee\_id  
 group by e.manager\_id  
 having e.manager\_id is not null)  
 ,cc as (  
 select e.manager\_id,sum(empcount2) empcount3 from employees e  
     join bb on bb.manager\_id=e.employee\_id  
 group by e.manager\_id  
 having e.manager\_id is not null)  
 ,dd as (  
 select aa.manager\_id,empcount1+ nvl(empcount2,0)+nvl(empcount3,0)+1 empcount from aa  
 left join bb on aa.manager\_id = bb.manager\_id  
 left join cc on bb.manager\_id = cc.manager\_id)  
 select first\_name,last\_name,department\_name,salary,dd.empcount empcount  
 from dd  
 join employees e on e.employee\_id = dd.manager\_id  
 join departments d on e.department\_id = d.department\_id  
 order by empcount desc ,salary asc  
 ​

**NO.9+**

 with emp as(  
 select e.Employee\_id,e.First\_name,e.Last\_name,Salary,(select count(\*)  
 from EMPLOYEES ee  
 start with ee.Employee\_id=e.Employee\_id  
 connect by prior Employee\_id = ee.Manager\_id) c,  
 (((select sum(ee.Salary)  
 from EMPLOYEES ee  
 start with ee.Employee\_id=e.Employee\_id  
 connect by prior Employee\_id = Manager\_id)-e.Salary)/((select count(\*)  
 from EMPLOYEES ee  
 start with ee.Employee\_id=e.Employee\_id  
 connect by prior Employee\_id = ee.Manager\_id)-1)) avg,e.Department\_id  
 from EMPLOYEES e join DEPARTMENTS on e.Department\_id=DEPARTMENTS.Department\_id  
 where (  
 select count(\*)  
 from EMPLOYEES ee  
 start with ee.Employee\_id=e.Employee\_id  
 connect by prior Employee\_id = ee.Manager\_id)>1  
 ),  
 dep as(  
 select emp.Department\_id,max(Salary) mxs,max(c) mxc  
 from emp join DEPARTMENTS on DEPARTMENTS.Department\_id=emp.Department\_id  
 group by emp.Department\_id  
 )  
 select Employee\_id,First\_name,Last\_name,emp.Department\_id,avg  
 from emp join dep on emp.Department\_id=dep.Department\_id  
 where emp.c=dep.mxc  
 minus  
 select Employee\_id,First\_name,Last\_name,emp.Department\_id,avg  
 from emp join dep on emp.Department\_id=dep.Department\_id  
 where emp.Salary<dep.mxs  
 order by avg desc,Employee\_id

**UNIVERSITY**

**NO.2.1**

 select a.COURSE\_ID,a.TITLE,count(SEC\_ID)  
 from COURSE a left join SECTION b on a.COURSE\_ID=b.COURSE\_ID  
 where a.COURSE\_ID not in (select c.COURSE\_ID from TAKES c where c.GRADE='F')  
 group by a.COURSE\_ID,a.TITLE

**NO.2.2**

 select COURSE\_ID,YEAR,SEMESTER,SEC\_ID,  
 (select count(ID) from TAKES where TAKES.SEC\_ID=s.SEC\_ID and TAKES.COURSE\_ID=s.COURSE\_ID and TAKES.YEAR=s.YEAR and TAKES.SEMESTER=s.SEMESTER and GRADE like '%A%'),  
 (select count(ID) from TAKES where TAKES.SEC\_ID=s.SEC\_ID and TAKES.COURSE\_ID=s.COURSE\_ID and TAKES.YEAR=s.YEAR and TAKES.SEMESTER=s.SEMESTER and GRADE like '%B%'),  
 (select count(ID) from TAKES where TAKES.SEC\_ID=s.SEC\_ID and TAKES.COURSE\_ID=s.COURSE\_ID and TAKES.YEAR=s.YEAR and TAKES.SEMESTER=s.SEMESTER and GRADE like '%C%'),  
 (select count(ID) from TAKES where TAKES.SEC\_ID=s.SEC\_ID and TAKES.COURSE\_ID=s.COURSE\_ID and TAKES.YEAR=s.YEAR and TAKES.SEMESTER=s.SEMESTER and GRADE like '%D%'),  
 (select count(ID) from TAKES where TAKES.SEC\_ID=s.SEC\_ID and TAKES.COURSE\_ID=s.COURSE\_ID and TAKES.YEAR=s.YEAR and TAKES.SEMESTER=s.SEMESTER and GRADE like '%F%'),  
 (select count(ID) from TAKES where TAKES.SEC\_ID=s.SEC\_ID and TAKES.COURSE\_ID=s.COURSE\_ID and TAKES.YEAR=s.YEAR and TAKES.SEMESTER=s.SEMESTER and GRADE is null)  
 from TAKES s  
 group by COURSE\_ID,YEAR,SEMESTER,SEC\_ID  
 order by COURSE\_ID

**NO.1.1**

 select s.NAME,TITLE,count(t.COURSE\_ID)  
 from student s left join TAKES t on s.ID=t.ID left join COURSE on t.COURSE\_ID=COURSE.COURSE\_ID  
 group by NAME,TITLE  
 having count(t.COURSE\_ID)>1

**NO.1.2**

 select a.ID,NAME  
 from (select ID  
 from STUDENT  
 minus  
 select ID  
 from TAKES) a left join STUDENT on a.ID=STUDENT.ID

**NO.1.3**

 select TITLE,SEC\_ID,YEAR,SEMESTER  
 from TAKES join COURSE on TAKES.COURSE\_ID=COURSE.COURSE\_ID  
 where GRADE is null  
 group by TITLE,SEC\_ID,YEAR,SEMESTER

**NO.1.4**

 with t as(  
 select d.DEPT\_NAME  
 from DEPARTMENT d join INSTRUCTOR i on d.DEPT\_NAME=i.DEPT\_NAME  
 )  
 select ID,NAME,t.DEPT\_NAME  
 from STUDENT s join t on s.DEPT\_NAME=t.DEPT\_NAME  
 group by ID,NAME,t.DEPT\_NAME  
 having count(ID)>=4

**NO.1.5**

 select d.DEPT\_NAME,BUILDING,count(\*)  
 from DEPARTMENT d join INSTRUCTOR i on d.DEPT\_NAME=i.DEPT\_NAME  
 group by d.DEPT\_NAME,BUILDING  
 having d.DEPT\_NAME in (select d.DEPT\_NAME  
 from DEPARTMENT d join INSTRUCTOR i on d.DEPT\_NAME=i.DEPT\_NAME  
 where i.NAME like '%世%')

**NO.2.3**

/\*  
"Get the total credit hours in a week for every course which has has corresponding sections in every semester .   
For each course as such , following information should be listed:  
\* ID of the course  
\* Year of the course ’s sections  
\* Semester of the course ’s sections  
\* Count of the Count of the course ’s sections in the corresponding year and semester  
\* Total Credit hours in a week.  
Note: 1 credit hour = 50 minutes"  
/\*

 select COURSE\_ID, YEAR, SEMESTER, count(distinct s.TIME\_SLOT\_ID),sum(((END\_HR - START\_HR) \* 60 + END\_MIN - START\_MIN)/50)  
 from SECTION s  
          join TIME\_SLOT t on s.TIME\_SLOT\_ID = t.TIME\_SLOT\_ID  
 group by COURSE\_ID, YEAR, SEMESTER

**NO.2.4**

 select distinct e.NAME,i.NAME  
 from STUDENT e  
          join ADVISOR a on e.ID = a.S\_ID  
          join TAKES t on e.ID = t.ID  
          join TEACHES s  
               on s.COURSE\_ID = t.COURSE\_ID and s.SEC\_ID = t.SEC\_ID and s.SEMESTER = t.SEMESTER and s.YEAR = t.YEAR  
          join INSTRUCTOR i on i.ID = s.ID  
 where a.I\_ID=i.ID

**NO.3.2**

 select NAME,DEPT\_NAME  
 from STUDENT  
 where not exists(  
 select COURSE\_ID  
 from COURSE  
 where COURSE.DEPT\_NAME = STUDENT.DEPT\_NAME  
 minus  
 select COURSE\_ID  
 from takes  
 where STUDENT.ID =takes.ID and GRADE not like '%F%'  
 )

**NO.3.3**

 with a as(  
     select ID,NAME,row\_number() over (order by ID) r  
 from STUDENT  
 )  
 select ID,NAME  
 from a  
 where r>=6 and r<=10

**NO.3.4**

 select STUDENT.ID,  
        STUDENT.NAME,  
        (select count(\*)  
         from (  
                  select distinct COURSE\_ID  
                  from TAKES  
                  WHERE STUDENT.id = TAKES.id  
                    and TAKES.GRADE like '%F%'  
                  minus  
                  select distinct COURSE\_ID  
                  from TAKES  
                  WHERE STUDENT.id = TAKES.id  
                    and TAKES.GRADE not like '%F%'  
              )  
        )  
 from STUDENT  
 where exists(  
               select distinct COURSE\_ID  
               from TAKES  
               WHERE STUDENT.id = TAKES.id  
                 and TAKES.GRADE like '%F%'  
              minus  
               select distinct COURSE\_ID  
               from TAKES  
               WHERE STUDENT.id = TAKES.id  
                 and TAKES.GRADE not like '%F%'  
           )

**NO.3.5**

 with t1 as (  
     select ID, COURSE\_ID, count(\*) cnt  
     from TAKES  
     group by ID, COURSE\_ID  
     having count(\*) > 1  
 )  
 select t1.ID,  
        NAME,  
        COURSE\_ID,  
        cnt,  
        (select count(\*)  
        from TAKES t  
        where t.ID = t1.ID and t.COURSE\_ID=t1.COURSE\_ID and GRADE like '%F%')  
 from t1  
          join STUDENT on t1.ID = STUDENT.ID

select STUDENT.ID,

STUDENT.NAME,

(select count(\*)

from (

select distinct COURSE\_ID

from TAKES

WHERE STUDENT.id = TAKES.id

and TAKES.GRADE like '%F%'

minus

select distinct COURSE\_ID

from TAKES

WHERE STUDENT.id = TAKES.id

and TAKES.GRADE not like '%F%'

)

)

from STUDENT

where exists(

select distinct COURSE\_ID

from TAKES

WHERE STUDENT.id = TAKES.id

and TAKES.GRADE like '%F%'

minus

select distinct COURSE\_ID

from TAKES

WHERE STUDENT.id = TAKES.id

and TAKES.GRADE not like '%F%'

)

with t1 as (

select ID, COURSE\_ID, count(\*) cnt

from TAKES

group by ID, COURSE\_ID

having count(\*) > 1

)

select t1.ID,

NAME,

COURSE\_ID,

cnt,

(select count(\*)

from TAKES t

where t.ID = t1.ID and t.COURSE\_ID=t1.COURSE\_ID and GRADE like '%F%')

from t1

join STUDENT on t1.ID = STUDENT.ID