SQL 面试用题

employees 表：

EMPLOYEE_ID	NUMBER(6)
FIRST_NAME	VARCHAR2(20)
LAST_NAME	VARCHAR2(25)
EMAIL	VARCHAR2(25)
PHONE_NUMBER	VARCHAR2(20)
HIRE_DATE	DATE
JOB_ID	VARCHAR2(10)
SALARY	NUMBER(8,2)
COMMISSION_PCT	NUMBER(2,2)
MANAGER_ID	NUMBER(6)
DEPARTMENT_ID	NUMBER(4)

departments 表：

DEPARTMENT_ID	NUMBER(4)
DEPARTMENT_NAME	VARCHAR2(30)
MANAGER_ID	NUMBER(6)
LOCATION_ID	NUMBER(4)

locations 表：

LOCATION_ID	NUMBER(4)
STREET_ADDRESS	VARCHAR2(40)
POSTAL_CODE	VARCHAR2(12)
CITY	VARCHAR2(30)
STATE_PROVINCE	VARCHAR2(25)
COUNTRY_ID	CHAR(2)

jobs 表：

JOB_ID	VARCHAR2(10)
JOB_TITLE	VARCHAR2(35)
MIN_SALARY	NUMBER(6)
MAX_SALARY	NUMBER(6)

job_grades 表：

GRADE_LEVEL	VARCHAR2(3)
LOWEST_SAL	NUMBER
HIGHEST_SAL	NUMBER

1．查询每个月倒数第 2 天入职的员工的信息.

2．查询出 last_name 为 'Chen' 的 manager 的信息.

3．查询平均工资高于 8000 的部门 id 和它的平均工资.

4．查询工资最低的员工信息: last_name, salary

5．查询平均工资最低的部门信息

6．查询平均工资最低的部门信息和该部门的平均工资

7．查询平均工资最高的 job 信息

8．查询平均工资高于公司平均工资的部门有哪些?

9．查询出公司中所有 manager 的详细信息.

10．各个部门中 最高工资中最低的那个部门的 最低工资是多少

11．查询平均工资最高的部门的 manager 的详细信息：last_name, department_id, email, salary

12．查询 1999 年来公司的人所有员工的最高工资的那个员工的信息.

13．返回其它部门中比 job_id 为‘IT_PROG’部门所有工资都低的员工的员工号、姓名、job_id 以及 salary

**************************answers*************************

1．查询每个月倒数第 2 天入职的员工的信息.

```
select last_name, hire_date
from employees
where hire_date = last_day(hire_date) - 1
```

2．查询出 last_name 为 'Chen' 的 manager 的信息.

1)．通过两条 sql 查询:

```
select manager_id
from employees
where lower(last_name) = 'chen' --返回的结果为 108

select *
from employees
where employee_id = 108
```

2)．通过一条 sql 查询(自连接):

```
select m.*
from employees e, employees m
where e.manager_id = m.employee_id and e.last_name
= 'Chen'
```

3). 通过一条 sql 查询(子查询):

```
select *
from employees
where employee_id = (
                    select manager_id
                    from employees
                    where last_name = 'Chen'
                   )
```

3. 查询平均工资高于 8000 的部门 id 和它的平均工资.

```
SELECT department_id, avg(salary)
FROM employees e
GROUP BY department_id
HAVING avg(salary) > 8000
```

4. 查询工资最低的员工信息: last_name, salary

```
SELECT last_name, salary
FROM employees
WHERE salary = (
   SELECT min(salary)
   FROM employees
)
```

5. 查询平均工资最低的部门信息

```
SELECT *
FROM departments
WHERE department_id = (
   SELECT department_id
   FROM employees
   GROUP BY department_id
   HAVING avg(salary) = (
      SELECT min(avg(salary))
      FROM employees
      GROUP BY department_id
   )
)
```

6．查询平均工资最低的部门信息和该部门的平均工资

```
select d.*, (select avg(salary) from employees where
department_id = d.department_id)
from departments d
where d.department_id = (
      SELECT department_id
      FROM employees
      GROUP BY department_id
      HAVING avg(salary) = (
            SELECT min(avg(salary))
            FROM employees
            GROUP BY department_id
             )
      )
```

7．查询平均工资最高的 job 信息

1)．按 job_id 分组，查询最高的平均工资

```
SELECT max(avg(salary))
FROM employees
GROUP BY job_id
```

2)．查询出平均工资等于 1) 的 job_id

```
SELECT job_id FROM
employees GROUP BY
job_id HAVING
avg(salary) = (
   SELECT max(avg(salary))
   FROM employees
   GROUP BY job_id
)
```

3)．查询出 2) 对应的 job 信息

```
SELECT *
FROM jobs
```

```
WHERE job_id = (
   SELECT job_id
   FROM employees
   GROUP BY job_id
   HAVING avg(salary) = (
      SELECT max(avg(salary))
      FROM employees
      GROUP BY job_id
   )
)
```

8．查询平均工资高于公司平均工资的部门有哪些?

1)．查询出公司的平均工资

```
SELECT avg(salary)
FROM employees
```

2)．查询平均工资高于 1) 的部门 ID

```
SELECT department_id
FROM employees
GROUP BY department_id
HAVING avg(salary) > (
SELECT avg(salary)
   FROM employees
)
```

9．查询出公司中所有 manager 的详细信息.

1)．查询出所有的 manager_id

```
SELECT distinct manager_id
FROM employeess
```

2)．查询出 employee_id 为 1) 查询结果的那些员工的信息

```
SELECT employee_id, last_name
FROM employees
WHERE employee_id in ( SELECT
   distinct manager_id
```

```
    FROM employees
)
```

10．各个部门中 最高工资中最低的那个部门的 最低工资是多少

1)．查询出各个部门的最高工资

```
SELECT max(salary)
FROM employees
GROUP BY department_id
```

2)．查询出 1) 对应的查询结果的最低值：各个部门中最低的最高工资(无法查询对应的 department_id)

```
SELECT min(max(salary))
FROM employees
GROUP BY department_id
```

3)．查询出 2) 所对应的部门 id 是多少：各个部门中最高工资等于 2) 的那个部门的 id

```
SELECT department_id
FROM employees
GROUP BY department_id
HAVING max(salary) = (
    SELECT min(max(salary))
    FROM employees
    GROUP BY department_id
)
```

4)．查询出 3) 所在部门的最低工资

```
SELECT min(salary)
FROM employees
WHERE department_id = (
    SELECT department_id
    FROM employees
    GROUP BY department_id
    HAVING max(salary) = (
```

```
        SELECT min(max(salary))
        FROM employees
        GROUP BY department_id
    )
)
```

11. 查询平均工资最高的部门的 manager 的详细信息: last_name, department_id, email, salary

1). 各个部门中, 查询平均工资最高的平均工资是多少

```
SELECT max(avg(salary))
FROM employees
GROUP BY department_id
```

2). 各个部门中, 平均工资等于 1) 的那个部门的部门号是多少

```
SELECT department_id
FROM employees
GROUP BY department_id
HAVING avg(salary) = (
    SELECT max(avg(salary))
    FROM employees
    GROUP BY department_id
)
```

3). 查询出 2) 对应的部门的 manager_id

```
SELECT manager_id
FROM departments
WHERE department_id = (
    SELECT department_id
    FROM employees
    GROUP BY department_id
    HAVING avg(salary) = (
        SELECT max(avg(salary))
        FROM employees
        GROUP BY department_id
    )
)
```

4). 查询出 employee_id 为 3) 查询的 manager_id 的员工的 last_name, department_id, email, salary

```
SELECT last_name, department_id, email,
salary FROM employees
WHERE employee_id = (
   SELECT manager_id
   FROM departments
   WHERE department_id = (
      SELECT department_id
      FROM employees
      GROUP BY department_id
      HAVING avg(salary) = (
         SELECT max(avg(salary))
         FROM employees
         GROUP BY department_id
      )
   )
)
```

12. 查询 1999 年来公司的人所有员工的最高工资的那个员工的信息.

1). 查询出 1999 年来公司的所有的员工的 salary

```
SELECT salary
FROM employees
WHERE to_char(hire_date, 'yyyy') = '1999'
```

2). 查询出 1) 对应的结果的最大值

```
SELECT max(salary)
FROM employees
WHERE to_char(hire_date, 'yyyy') = '1999'
```

3). 查询工资等于 2) 对应的结果且 1999 年入职的员工信息

```
SELECT *
FROM employees
WHERE to_char(hire_date, 'yyyy') = '1999' AND salary
= (
   SELECT max(salary)
```

```
            FROM employees
            WHERE to_char(hire_date, 'yyyy') = '1999'
        )
```

13．返回其它部门中比 job_id 为‘IT_PROG’部门所有工资都低的员工的员工号、姓名、job_id 以及 salary

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary < ALL
                (SELECT salary
                 FROM   employees
                 WHERE  job_id = 'IT_PROG')
AND    job_id <> 'IT_PROG';
```

**********高级子查询**********

- 书写多列子查询
- 在 FROM 子句中使用子查询
- 在 SQL 中使用单列子查询
- 书写相关子查询
- 使用 EXISTS 和 NOT EXISTS 操作符
- 使用子查询更新和删除数据
- 使用 WITH 子句

--多列子查询（不成对比较 & 成对比较）

1. 查询与 141 号或 174 号员工的 manager_id 和 department_id 相同的其他员工的 employee_id，manager_id，department_id

[方式一]

```
SELECT  employee_id, manager_id, department_id
FROM    employees
WHERE   manager_id IN
```

```
                (SELECT  manager_id
                 FROM    employees
                 WHERE   employee_id IN (174,141))
AND    department_id IN
                (SELECT  department_id
                 FROM    employees
                 WHERE   employee_id IN (174,141))
AND   employee_id NOT IN(174,141);
```

[方式二]

```
SELECT   employee_id, manager_id, department_id
FROM  employees
WHERE  (manager_id, department_id) IN
                   (SELECT manager_id, department_id
                    FROM   employees
                    WHERE  employee_id IN (141,174))
AND   employee_id NOT IN (141,174);
```

--在 FROM 子句中使用子查询

2. 返回比本部门平均工资高的员工的 last_name, department_id, salary 及平均工资

[方式一]

```
select last_name,department_id,salary,
(select avg(salary)from employees e3
where e1.department_id =
e3.department_id group by department_id)
avg_salary from employees e1
where salary >
        (select avg(salary)
         from employees e2
         where e1.department_id =
         e2.department_id --group by
         department_id )
```

[方式二]

```
SELECT  a.last_name, a.salary,
       a.department_id, b.salavg
```

```
FROM    employees a, (SELECT   department_id,
                    AVG(salary) salavg
                    FROM    employees
                    GROUP BY department_id)
b WHERE a.department_id = b.department_id
AND     a.salary > b.salavg;
```

--单列子查询表达式

- Oracle8*i* 只在下列情况下可以使用，例如：
  - SELECT 语句（FROM 和 WHERE 子句）
  - INSERT 语句中的 VALUES 列表中
- Oracle9*i* 中单列子查询表达式可在下列情况下使用：
  - DECODE 和 CASE
  - SELECT 中除 GROUP BY 子句以外的所有子句中

3. 显式员工的 employee_id,last_name 和 location。其中，若员工 department_id 与 location_id 为 1800 的 department_id 相同，则 location 为’Canada’,其余则为’USA’。

```
SELECT employee_id, last_name,
      (CASE department_id
     WHEN    (SELECT department_id FROM departments
             WHERE location_id = 1800)
       THEN 'Canada' ELSE 'USA' END) location
FROM employees;
```

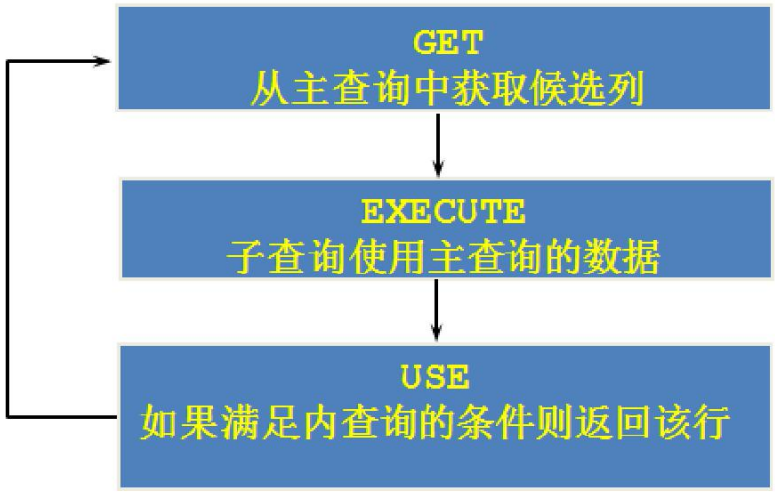
4. 查询员工的 employee_id,last_name，要求按照员工的 department_name 排序

```
SELECT   employee_id, last_name
FROM     employees e
```

```
ORDER BY (SELECT department_name
            FROM departments d
            WHERE e.department_id = d.department_id);
```

--相关子查询

相关子查询按照一行接一行的顺序执行，主查询的每一行都执行一次子查询




```
SELECT column1, column2, ...
FROM   table1  outer
WHERE  column1 operator
                    (SELECT  colum1, column2
                     FROM    table2
                     WHERE   expr1 =
                                 outer.expr2);
```

5.查询员工中工资大于本部门平均工资的员工的 last_name,

salary 和其 department_id

```
SELECT last_name, salary, department_id
FROM   employees outer
WHERE  salary >
```

```
                    (SELECT AVG(salary)
                     FROM   employees
                     WHERE  department_id =
                            outer.department_id) ;
```

6. 若 employees 表中 employee_id 与 job_history 表中 employee_id 相同的数目不小于 2，输出这些相同 id 的员工的 employee_id,last_name 和其 job_id

```
SELECT e.employee_id,
last_name,e.job_id FROM employees e
WHERE  2 <= (SELECT COUNT(*)
             FROM   job_history
             WHERE  employee_id = e.employee_id);
```

--EXISTS 操作符

- EXISTS 操作符检查在子查询中是否存在满足条件的行
- **如果在子查询中存在满足条件的行：**
  - 不在子查询中继续查找
  - 条件返回 TRUE

7. 查询公司管理者的 employee_id,last_name,job_id, department_id 信息

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees outer
WHERE  EXISTS ( SELECT 'X'
                FROM   employees
                WHERE  manager_id =
                       outer.employee_id);
```

8. 查询 departments 表中，不存在于 employees 表中的部门的

department_id 和 department_name

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'X'
                  FROM   employees
                  WHERE  department_id
                         = d.department_id);
```

--关于数据更新

9.修改表 employees,添加 department_name 列，赋予 department_id 相应的部门名称。

```
ALTER TABLE employees
ADD(department_name VARCHAR2(14));


UPDATE employees e
SET    department_name =
               (SELECT department_name
            FROM   departments d
            WHERE  e.department_id = d.department_id);
```

--关于数据删除

10.删除表 employees 中，其与 emp_history 表皆有的数据

```
DELETE FROM employees E
WHERE employee_id in
           (SELECT employee_id
            FROM   emp_history
          WHERE      employee_id = E.employee_id);
```

--WITH 子句

11. 查询公司中各部门的总工资大于公司中各部门的平均总工资的部门信息

```
WITH
```

```
dept_costs  AS (
   SELECT  d.department_name, SUM(e.salary) AS dept_total
   FROM    employees e, departments d
   WHERE   e.department_id = d.department_id
   GROUP BY d.department_name),
avg_cost    AS (
   SELECT SUM(dept_total)/COUNT(*) AS dept_avg
   FROM   dept_costs)
SELECT *
FROM   dept_costs
WHERE  dept_total >
       (SELECT dept_avg
        FROM avg_cost)
ORDER BY department_name;
```

附加题目：

12. 查询员工的 last_name, department_id, salary. 其中员工的 salary,department_id 与有奖金的任何一个员工的 salary, department_id 相同即可

```
select last_name, department_id,
salary from employees
where (salary,department_id) in (
select salary,department_id
from employees
where commission_pct is not null
                            )
```

13.选择工资大于所有 JOB_ID = 'SA_MAN' 的员工的工资的员工的 last_name, job_id, salary

```
select last_name, job_id, salary
from employees
where salary > all(
                select salary
```

```
                from employees
                where job_id = 'SA_MAN'
                )
```

14.选择所有没有管理者的员工的 last_name

```
select last_name
from employees e1
where not exists (
            select 'A'
            from employees e2
            where e1.manager_id = e2.employee_id
                )
```

15. 查询 10，50，20 号部门的 job_id，department_id 并且 department_id 按 10，50，20 的顺序排列

```
Column dummy noprint;
select job_id , department_id ,1 dummy
from employees
where department_id = 10
union
select job_id , department_id , 2
from employees
where department_id = 50
union
select job_id , department_id , 3
from employees
where department_id= 20
order by 3
```