|  |  |
| --- | --- |
| **머싱러닝지식기반의 데이터 사이언티스 양성과정** | Report |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **과정명** | **정형데이터처리 이해 : SQL활용** | | | **팀명** | 아프지말조 | **점수** |
| **성명** | 신동준 |  |
| **시행일** | 2019년 01월 29일 | **강사** | 김수환 | **직위** | 팀원 |

|  |
| --- |
| 실습문제를 레포트로 제출합니다. |

1. DML, DDL의 차이를 작성하시오.

DML : 데이터 조작어.

SELECT, INSERT, UPDATE, DELETE 등이 있다.

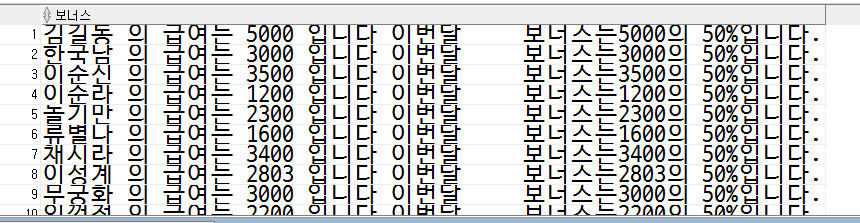
DDL : 데이터 정의어

CREATE, ALTER, DROP, RENAME, COMMENT 등이 있다.

1. 김길동의 급여는 10000 입니다. 이번달 보너스는 급여의 50% 입니다. 와 같은 형태로 || 연산자를 사용해서 출력하시오.

select saname || ' 의 급여는 '||sapay||' 입니다 이번달 보너스는'||sapay||'의 50%입니다.' as 보너스

from sawon;

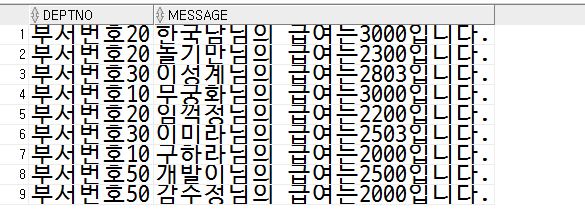


문제)" 부서번호 20 김길동 님의 급여는 3000 입니다. "  
단 ) 2000 ~ 3000 사이의 급여를 받는 20번 부서사원의 정보를 출력하되  
컬럼은 deptno, message로 출력하시오.  
  
select '부서번호'||deptno as deptno

, saname||'님의 급여는'||sapay||'입니다.' as message

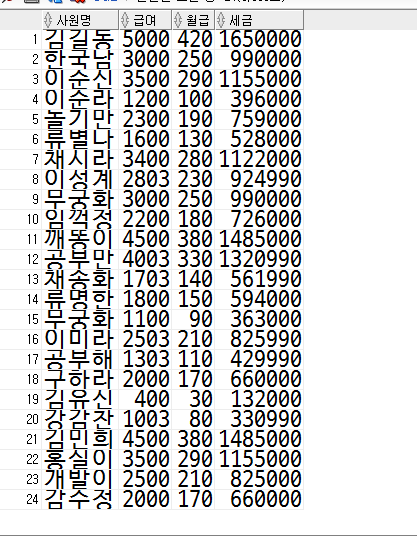
from sawon

where sapay>=2000 and sapay<=3000;



문제) 사원명, 급여, 월급(급여/12), 세금(급여의 3.3%)를 출력하시오.  
단) 월급은 십단위로 반올림하고, 세금은 원단위로 절삭하시오.  
select saname as 사원명, sapay as 급여, round(sapay/12,-1) as 월급, sapay/ 1000 \* 33\*10000 as 세금

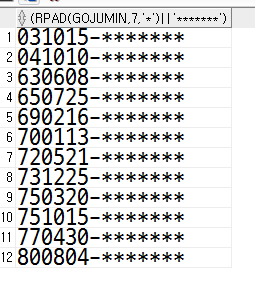
from sawon;



문제) 고객테이블에서 주민번호 뒷자리를 \* 로 보안처리해서 출력하시오.  
851113-\*\*\*\*\*\*\*

select (RPAD(gojumin,7,'\*') ||'\*\*\*\*\*\*\*')

from gogek;

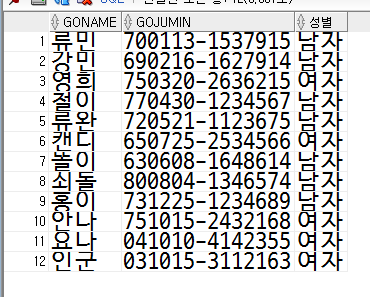


문제) 고객테이블에서 고객명, 주민번호, 성별을 출력 하시오,  
select goname,gojumin,

case when substr(gojumin, (instr(gojumin,'-',1,1)+1),1) =1 then '남자'

else '여자'

end "성별"

from gogek;  


문제) 사원명, 급여, 보너스를 출력   
-- 단) 보너스는 급여가 1000 미만 --> 급여의 10%  
--- 1000 ~ 2000 --> 급여의 15%  
--- 2000 초과 --> 급여의 20%  
--- null ---> 0

select saname, sapay,

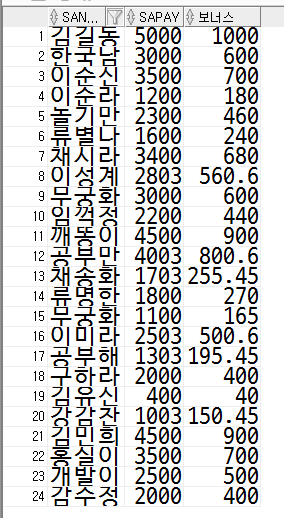
case when sapay < 1000 then sapay\*0.1

when sapay >=1000 and sapay < 2000 then sapay\*0.15

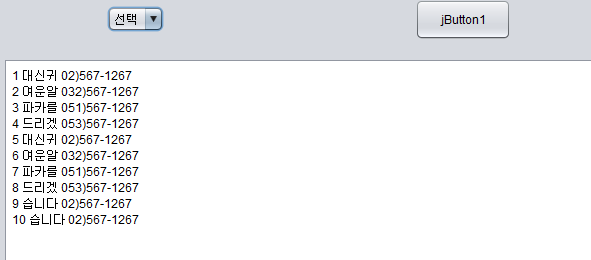
else sapay\*0.2

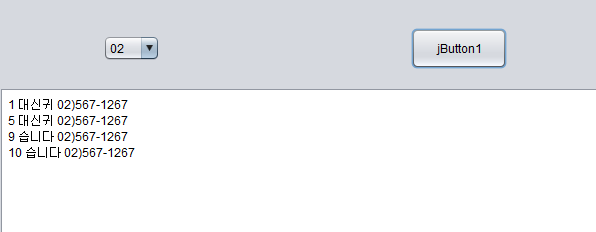
end "보너스"

from sawon;



문제)지역번호 코드 – gui로 만든 것 실행화면 캡쳐와 주요코드





* POJO

private int num;

private String name;

private String phone;

private String front;

* EXAM

private static NumSearchExam nse;

public synchronized static NumSearchExam getNse(){

if(nse == null){

nse = new NumSearchExam();

}

return nse;}

private NumSearchExam(){}

public List<MemphoneVO> getSearchList(String dname) throws SQLException{

List<MemphoneVO> list = new ArrayList<>();

List<MemphoneVO> resultList=new ArrayList<>();

Connection con = conn.MyConn.getConn();

String sql = "select num,name,pnum,substr(pnum,1,instr(pnum,')',1,1)-1) AS front from memphone";

PreparedStatement pstmt = con.prepareStatement(sql);

// pstmt.setString(1, sql);

ResultSet rs = pstmt.executeQuery();

while(rs.next()){

MemphoneVO mv=new MemphoneVO();

mv.setNum(rs.getInt("num"));

mv.setName(rs.getString("name"));

mv.setPhone(rs.getString("pnum"));

mv.setFront(rs.getString("front"));

list.add(mv);

}

if(dname.equals("선택")){

con.close();

return list;

}

else{

for(MemphoneVO e : list){

if(dname.equals(e.getFront())){

resultList.add(e);

}

}

}

con.close();

return resultList;

}

* GUI

public class NumSearchGUI extends javax.swing.JFrame {

List<MemphoneVO> list=null;

/\*\*

\* Creates new form NumSearchExam

\*/

public NumSearchGUI() {

initComponents();

try {

list=NumSearchExam.getNse().getSearchList((String)jComboBox1.getSelectedItem());

} catch (SQLException ex) {

Logger.getLogger(NumSearchGUI.class.getName()).log(Level.SEVERE, null, ex);

}

for(MemphoneVO e : list){

testArea.append(e.getNum()+" "+e.getName()+" "+e.getPhone()+"\n");

}

}

* 버튼

try {

list=NumSearchExam.getNse().getSearchList((String)jComboBox1.getSelectedItem());

testArea.selectAll();

testArea.setText("");

if(list!=null){

for(MemphoneVO e : list){

testArea.append(e.getNum()+" "+e.getName()+" "+e.getPhone()+"\n");

}

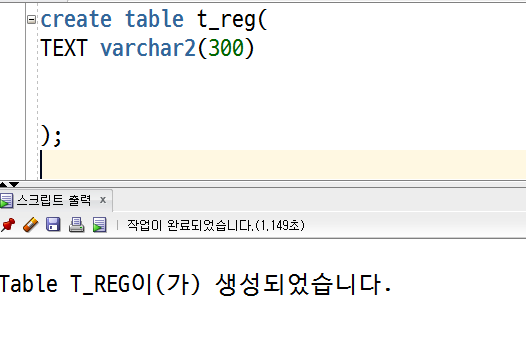
}

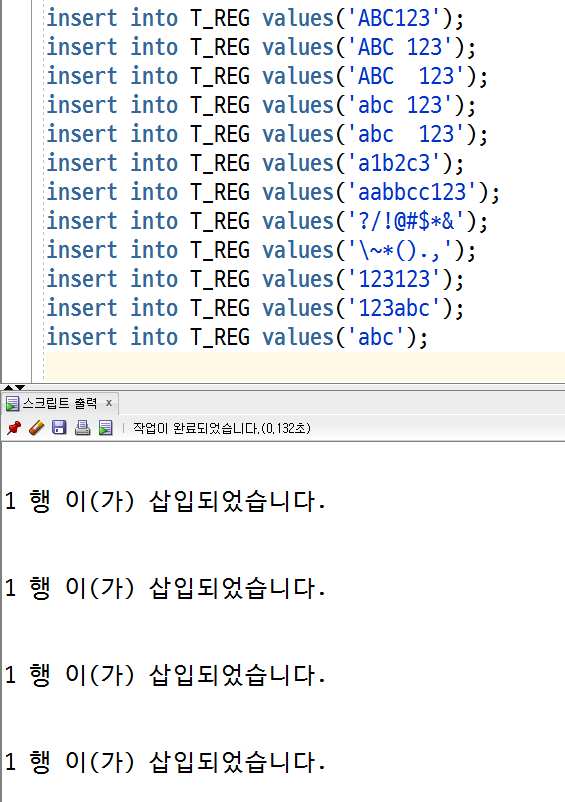
}catch(Exception e){

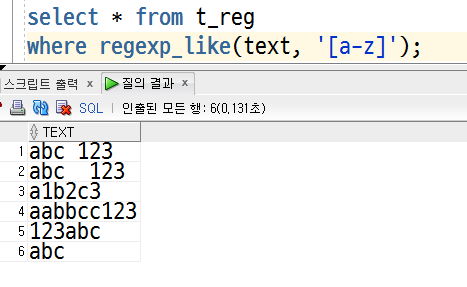
}

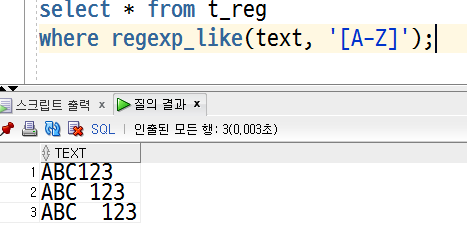
문제)오라클 책 p125-151 예제 공부!

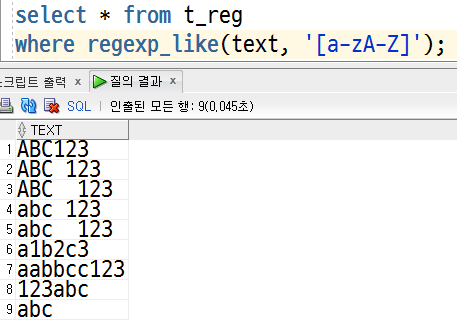
* 테이블 생성



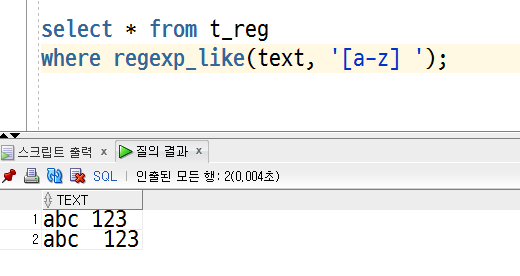


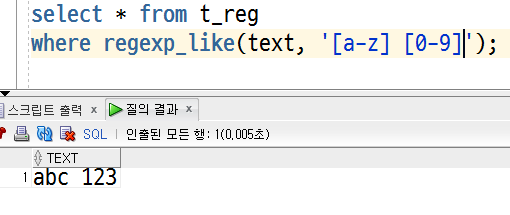


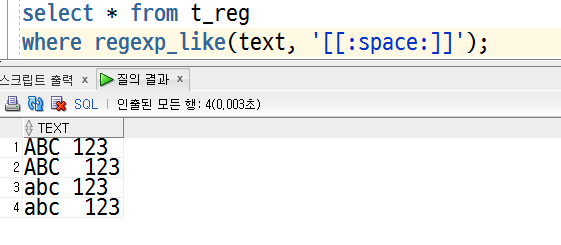


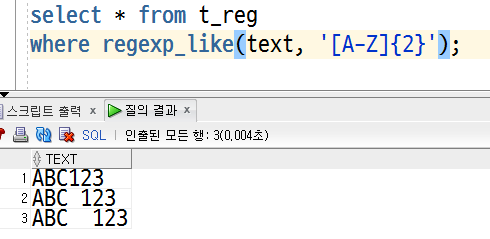


사용예제2. 소문자로 시작하고 공백을 포함하는 경우 찾기

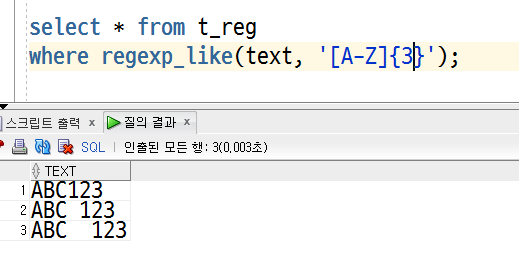


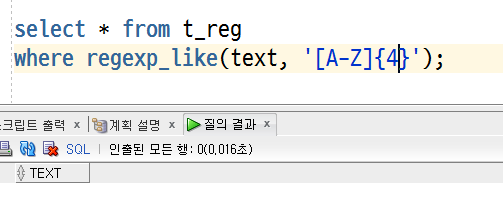


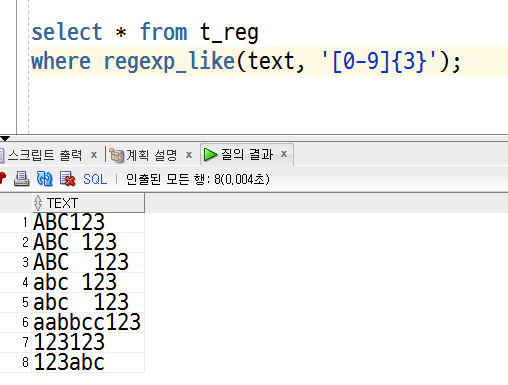




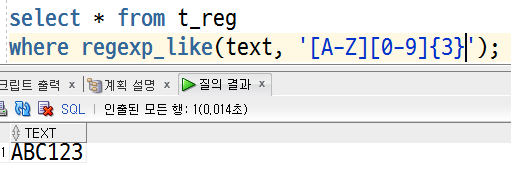
-대문자가 연속으로 2번오는 경우를 출력함.

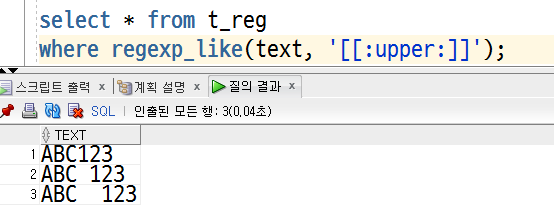




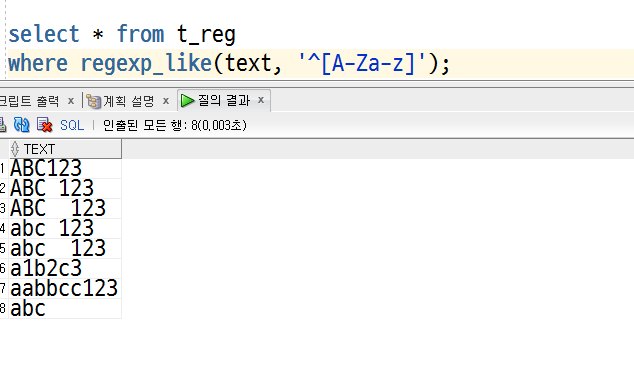


-숫자가 연속적으로 3글자 오는 행을 출력하는 예

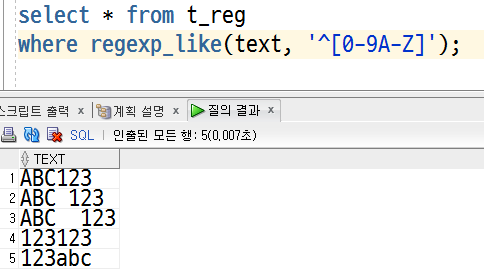




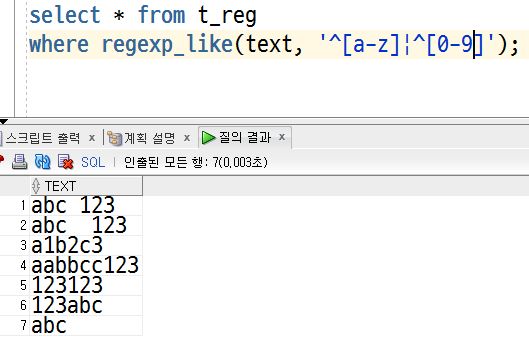
* 대문자만 오는 경우



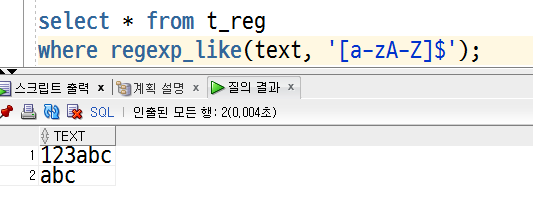
* 첫 시작을 대문자나 소문자로 시작하는 것을 모두 출력



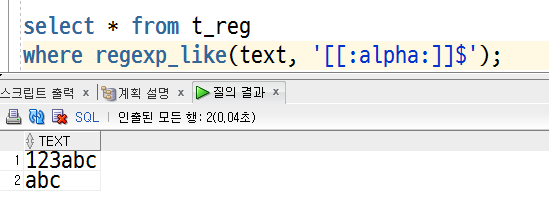
첫 시작을 숫자나 대문자로 시작하는 행 모두를 가져온다

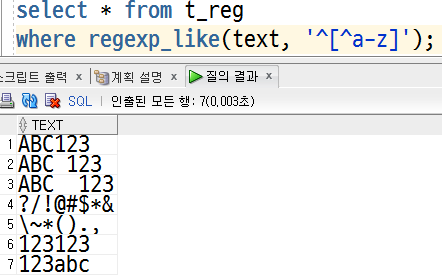


* 이 화면은 소문자로 시작하거나 숫자로 시작하는 화면을 출력

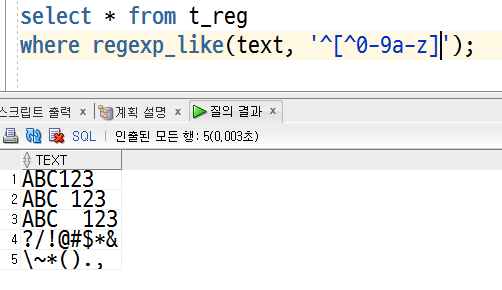


$문자ㄹㄹ 사용하여 소문자로 끝나는 행

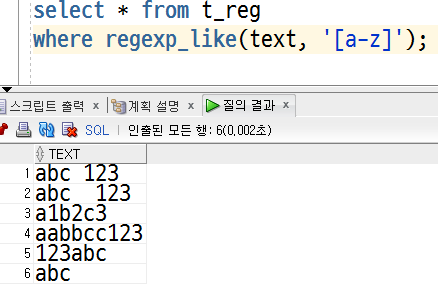




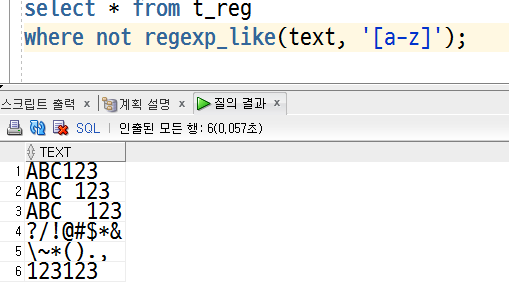
* 이 예제는 소문자로 시작하지 않는 행을 모두 출력한다.



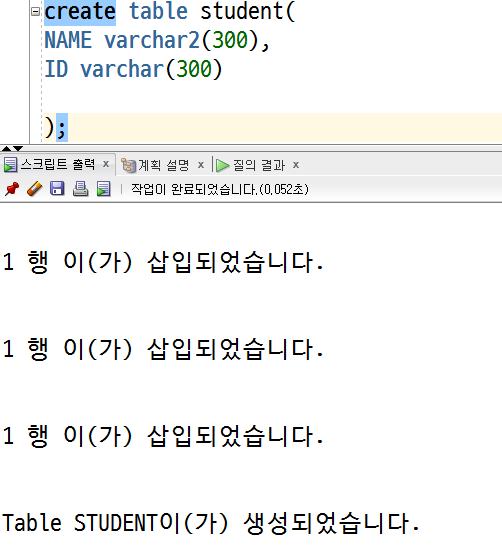
* 소문자나 숫자로 시작하지 않는 행을 출력하는 화면

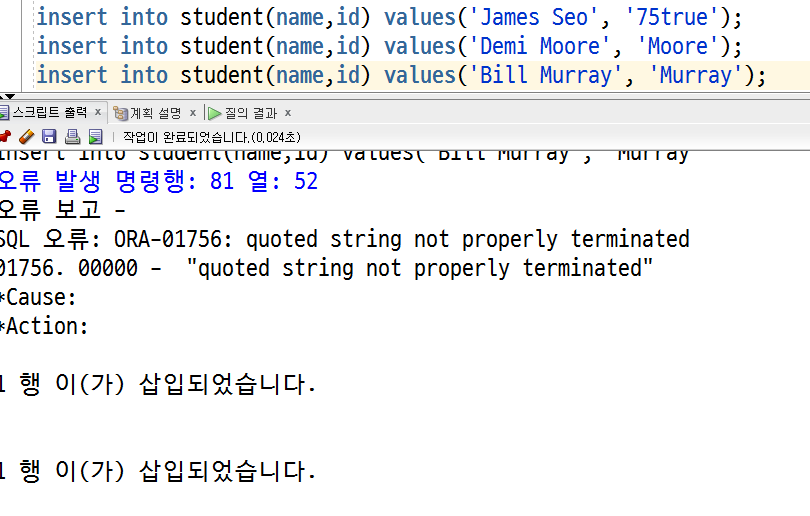


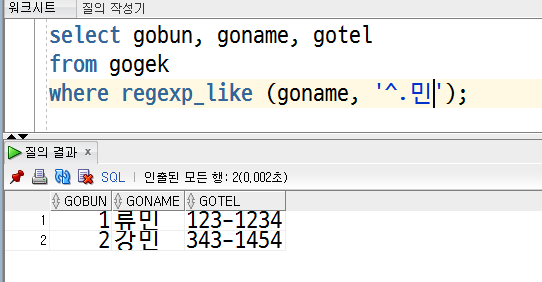
-기존

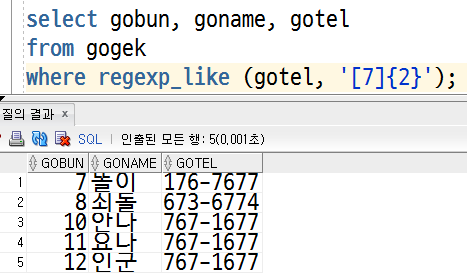


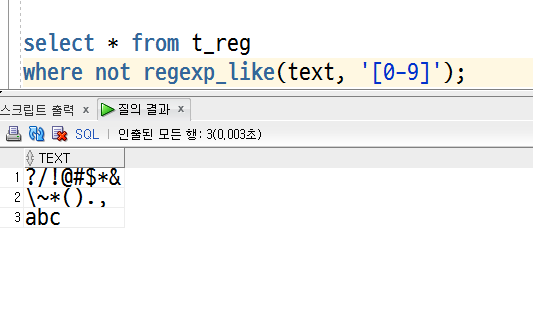
* Not을 사용할 때

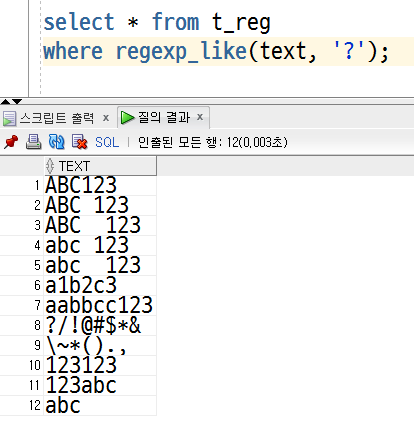


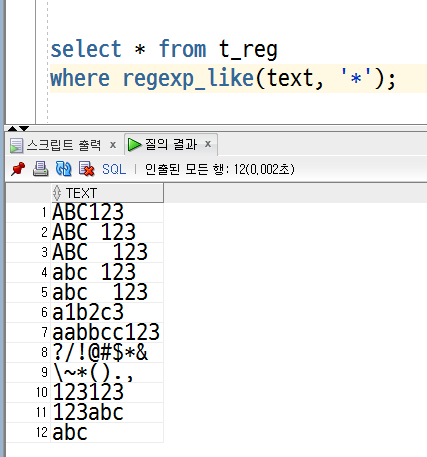


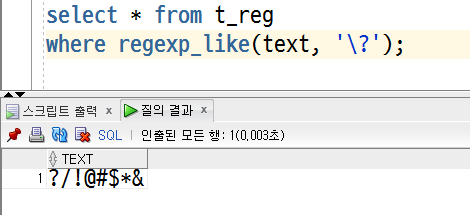


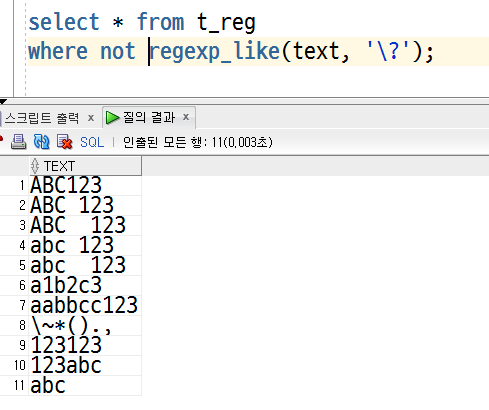


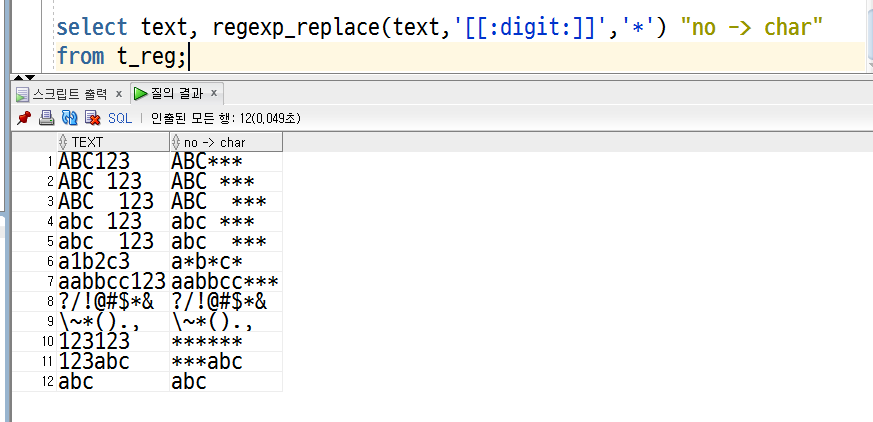


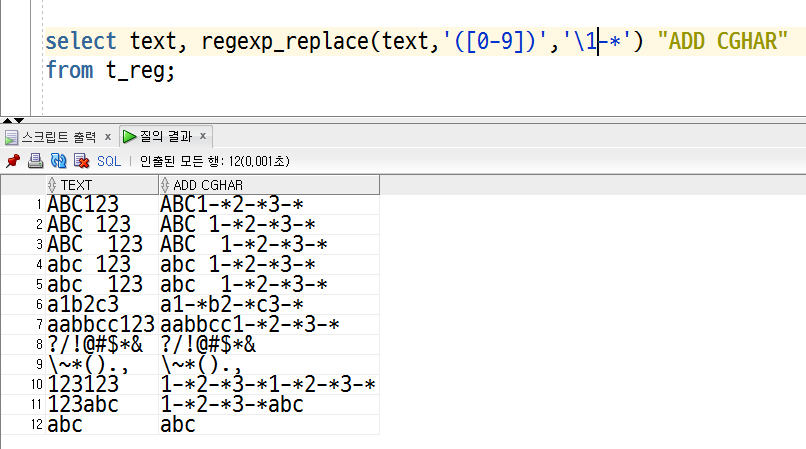




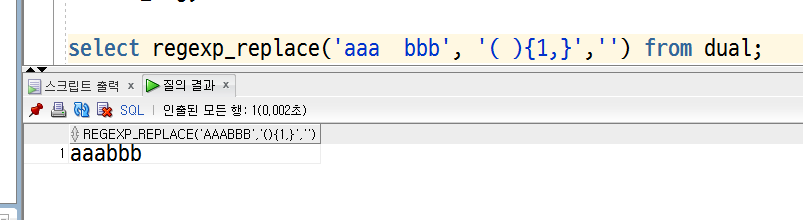




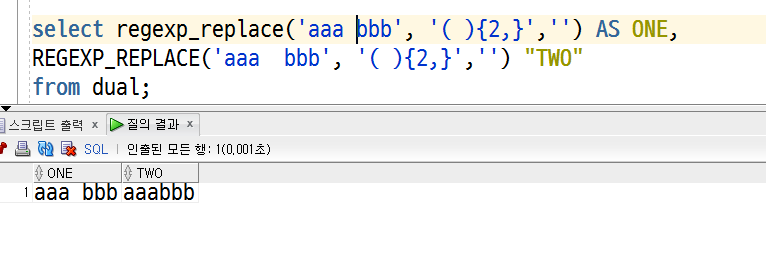




* 여기서 \ 는 한 개씩 더해주는 역할도 하고 있음.

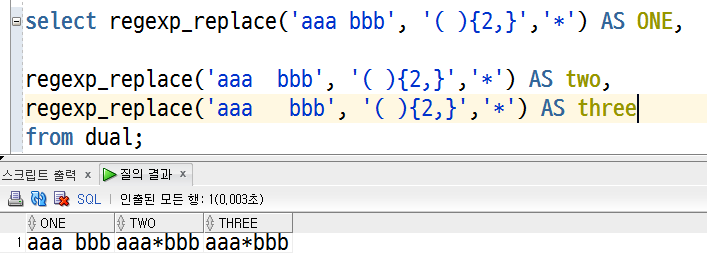


* Aaa와 bbb사이의 공백을 없에고 출력하는 방법이다.

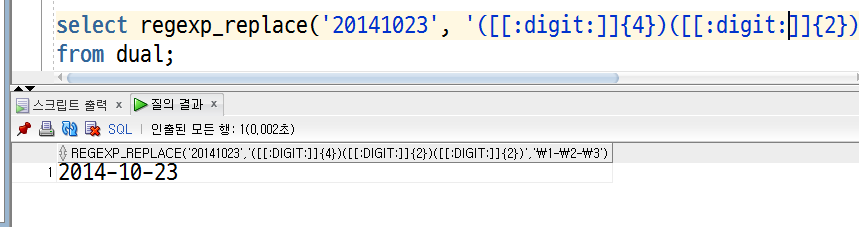


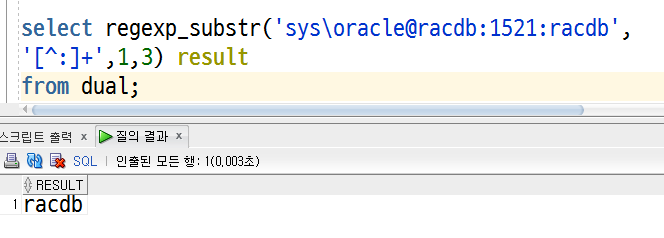
One은 두번째것을 스페이스된걸 붙이는 역할을 하는데 스페이스바는 1개 -> 끝

Two는 두칸뜨어져있고 그걸 합쳐지는 역할이다.

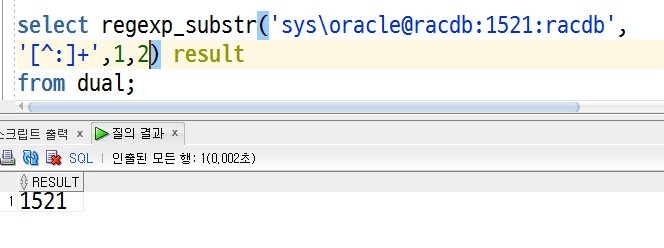


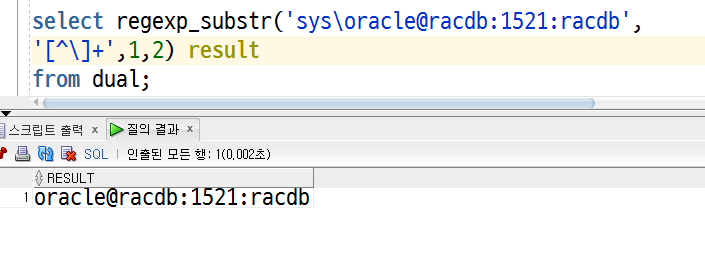
위랑 비슷한 예제이다.

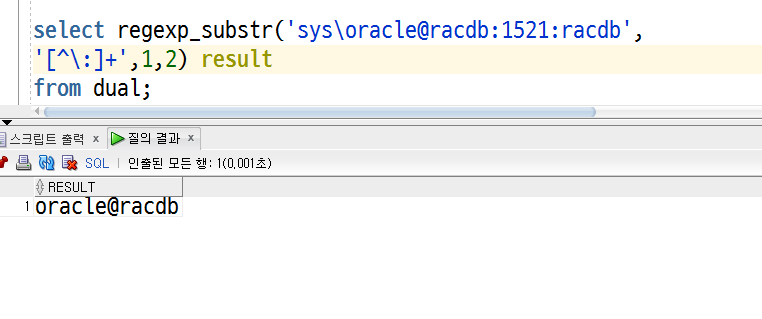




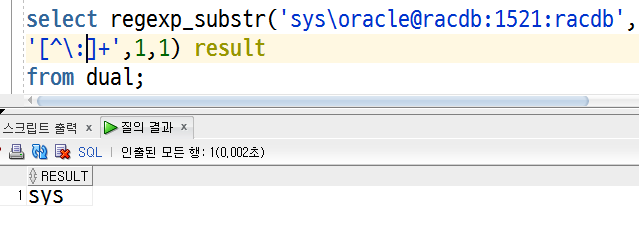
* 구분자를 사용하였다.
* 여기서는 : 기호를 기준으로 3번째를 찾아서 출력해주는 기능이다.
* 그래서 출력값이 racdb

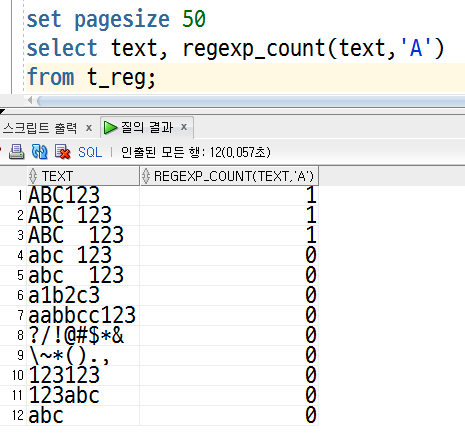


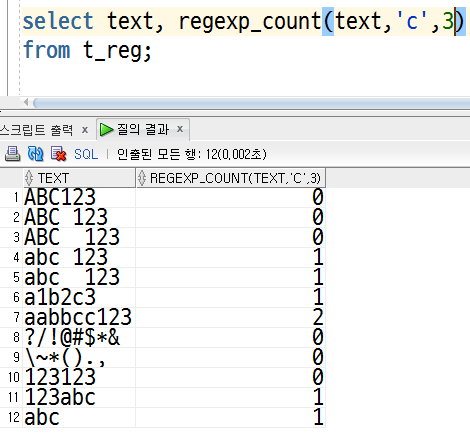




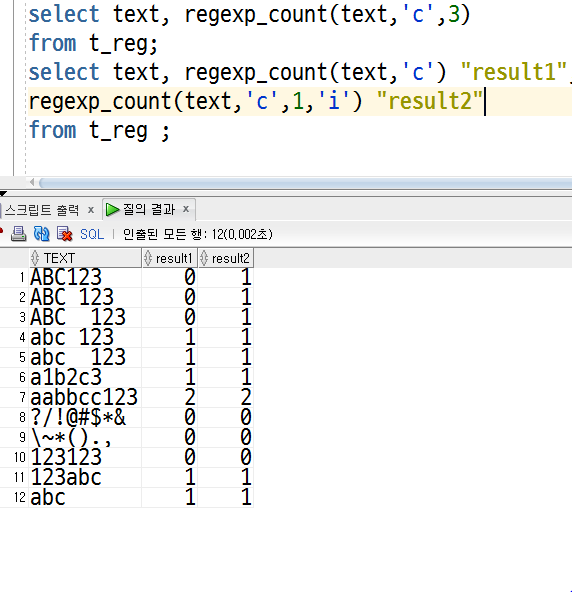
:를 넣었더니 구분자가 생겼다. /을 먼저 파악하고, :는 나중이다. :이 2번째이고 /d이 1이다. 즉, / : 1 : 2번째

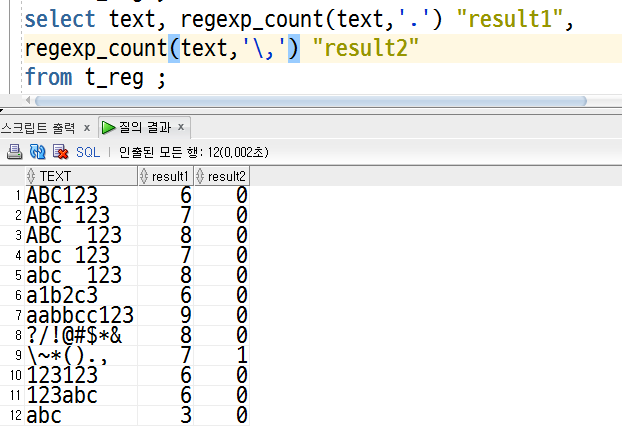


S



3번째 문자 이후부터 셈. 소문자 c이다.





* 이거 같은 경우에 탈출문자를 표시하는것이다.
* 리절트1에서는 . 이란 것은 모든 것이란 뜻으로 작동했지만 (length)
* 리절트2에는 \를 탈출문자로 썼기 때문에 \이 나온 9번이 1로 나왔다.