**로고학습일지**

**kt ds University 자바 기반의 데이터 사이언티스트 양성과정**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 08. 01 (수) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | 손에 잡히는 10분 SQL 6-8장 | | | | |
| 작 성 자 | 신예리 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | Oracle과 연결하여 STS에서 데이터 출력  와일드카드, 계산 필드 생성, 데이터 조작 함수 |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| HumanResource  >new HRDao().getCities(List.of(“JP”, “US”, “CA”, “UK”)); <- List에 이런 식으로 데이터 삽입할 수 있음  >동일하게 쓰이는 부분 : 추상 클래스 & 인터페이스 활용  >상속(extends)는 다중 상속 불가능 / 인터페이스(implements)는 다중 가능  >추상 : 이름은 있지만 바디(구현 내용)이 없는 것 -3단계 (0: 클래스, 1:Abstract, 2: 인터페이스)  >Template Callback : 한 프로세스에서 변경되는 부분만 분리하여 즉시 객체화 시킴.  Sql sql = new Sql() {  @Override  public PreparedStatement preparedStatement(Connection conn) throws SQLException {  PreparedStatement pstmt = conn.prepareStatement("");  return pstmt;  }  @Override  public void printRow(ResultSet rs) throws SQLException {  };  🡺 Sql()은 클래스가 아니라 객체화 시킬 수 없지만 Template Callback을 사용하여 객체화시킴.  <DB>  대문자가 Default  와일드 카드 문자를 이용한 필터링 : 쓸 수 있으나 속도에 많은 지장. 사용하기 전에 여부 확인 必  >LIKE : 문자열이 포함된 제품을 찾을 때.  >>% : 가장 많이 사용됨. = LIKE ‘Fish%’ / LIKE ‘%bean bag%’ / LIKE ‘\_\_ev%’  : 검색할 때 사용하기 좋아  계산  >문자열 더하기 : ;’ ‘+ ‘ ‘(X) 🡺 ‘ ‘ || ‘ ‘ (concat)  >별칭 : \_\_\_ (AS) 별칭 <- AS 생략 가능/ 한글 가능  >ROUND ( SALARY/12, 0 ) <- 소수점 0째까지  데이터 조작함수  >문자열 조작하기 위한 문자 함수  -UPPER : 대문자로 / LOWER : 소문자로 (거의 사용되지 않아)  >날짜와 시간을 조작하기 위한 날짜 함수  -SYSDATE : 현재 시간을 보여주는 FUNCTION  > DUAL : 더미 테이블 (단 하나만 나오게 / 내부 데이터 : ‘X’)   * SELECT SYSDATE   FROM DUAL  ; // 현재 시간을 보여주는   * SELECT HIRE\_DATE   FROM EMPLOYEES  WHERE HIRE\_DATE = ‘2006/03/15’  ; <- 이렇게 사용해 줄 수 있지만 안전하지 않다  🡺 TO\_DATE(‘2006/03/15’)를 사용해 String을 DATE 타입으로 변환   * SELECT TO\_CHAR(SYSDATE, ‘YYYY-MM-DD HH:MI; SS’)   FROM DUAL  ; // 2018-08-01 15:45:30   * SELECT TO\_CHAR(SYSDATE, ‘YYYY-MM-DD HH24:MI:SS’)   FROM DUAL  ; // 2018-08-01 15:45:30  => TO\_CHAR는 DATE 타입을 CHAR로  TO\_DATE는 CHAR 타입을 DATE로 (날짜 검색할 때 많이 사용)  > 외에는 잘 쓰이지 않는다. |

|  |  |
| --- | --- |
| 특이사항 |  |

■ORACLE

-- 100번 부서에서 일을 하는 사원들의 모든 정보를 조회하라.

-- (사원번호: EMPLOYEE\_ID로 오름차순 정렬)

SELECT \*

FROM EMPLOYEES

WHERE EMPLOYEE\_ID = 100

ORDER BY EMPLOYEE\_ID

;

SELECT \*

FROM EMPLOYEES

WHERE DEPARTMENT\_ID IN ( 90, 60, 100)

ORDER BY EMPLOYEE\_ID

;

SELECT \*

FROM EMPLOYEES

WHERE DEPARTMENT\_ID NOT IN ( 90, 60, 100)

ORDER BY EMPLOYEE\_ID

;

SELECT \*

FROM LOCATIONS

WHERE COUNTRY\_ID IN ( 'JP' )

ORDER BY COUNTRY\_ID

;

SELECT \*

FROM LOCATIONS

WHERE COUNTRY\_ID IN ( 'JP', 'CA' )

ORDER BY COUNTRY\_ID

;

-- 90, 60, 100번 부서에서 일을 하는 사원들 중 직무명(JOB\_ID)이 IT\_PROG인 사원들의

-- 모든 정보를 출력 (오름차순 EMPLOYEE\_ID)

SELECT \*

FROM EMPLOYEES

WHERE DEPARTMENT\_ID IN ( 90, 60, 100 ) -- 여러개일때는 IN

AND JOB\_ID = 'IT\_PROG' -- 조건이 하나일때는 =을 쓴다

ORDER BY EMPLOYEE\_ID

;

SELECT \*

FROM EMPLOYEES

WHERE MANAGER\_ID IS NULL

;

/\*

STATE\_PROVINCE의 값이 없는

CITY만 조회

(CITY로 내림차순 정렬)

\*/

SELECT CITY

FROM LOCATIONS

WHERE STATE\_PROVINCE IS NULL

ORDER BY CITY DESC

;

/\*

국가 코드가 JP, US, CA, UK인 도시 중

STATE\_PROVINCE의 값이 있는 도시명만 조회

(국가 코드 오름차순, 도시명 내림차순 정렬)

\*/

SELECT CITY

FROM LOCATIONS

WHERE COUNTRY\_ID IN ('JP', 'US', 'CA', 'UK')

AND STATE\_PROVINCE IS NOT NULL

ORDER BY COUNTRY\_ID

,CITY DESC

;

-- FIRST\_NAME 과 LAST\_NAME의 값이 'S'로 시작하는 모든 사원들 조회

SELECT \*

FROM EMPLOYEES

WHERE FIRST\_NAME LIKE 'S%'

AND LAST\_NAME LIKE 'S%'

;

-- FIRST\_NAME 혹은 LAST\_NAME의 값이 'S'로 시작하는 모든 사원들 조회

SELECT \*

FROM EMPLOYEES

WHERE FIRST\_NAME LIKE '%ev%'

OR LAST\_NAME LIKE '%ev%'

;

SELECT FIRST\_NAME || ' ' || LAST\_NAME NAME

FROM EMPLOYEES

;

SELECT ROUND(SALARY / 12, 0) MONTHLY\_PAY

FROM EMPLOYEES

;

/\*

인센티브(COMMISSION\_PCT)를 받는 사람의

총 연봉 구하기

\*/

SELECT EMPLOYEE\_ID, SALARY + (SALARY\*COMMISSION\_PCT) SALARY

FROM EMPLOYEES

WHERE COMMISSION\_PCT IS NOT NULL

;

-- 현재 시간 출력하기

SELECT SYSDATE

FROM DUAL

;

SELECT \*

FROM DUAL

;

SELECT HIRE\_DATE

FROM EMPLOYEES

;

SELECT HIRE\_DATE

FROM EMPLOYEES

WHERE HIRE\_DATE = TO\_DATE('2006/03/15')

;

SELECT TO\_CHAR(SYSDATE, 'YYYY-MM-DD HH24:MI:SS') --2018-08-01 15:45:30

FROM DUAL

;

SELECT TO\_DATE('2006/03/15 01:00:00', 'YYYY-MM-DD HH24:MI:SS')

FROM DUAL

;

SELECT FIRST\_NAME || LAST\_NAME NAME

, CASE

WHEN COMMISSION\_PCT IS NOT NULL THEN

SALARY + SALARY \* COMMISSION\_PCT

ELSE

SALARY

END SALARY

, TO\_CHAR(HIRE\_DATE, 'YYYY.MM.DD.') HIRE\_DATE

FROM EMPLOYEES

;