**로고학습일지**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 07. 31 (화) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | UI/UX | | | | |
| 작 성 자 | 장영선 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | 예외 처리, DB 설치 및 Setting, DB Select 구문 |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| 1. **예외 처리** 프로그램이 동작하는 과정 중 잘못된 입력값이 전달되거나, 읽어야 하는 파일이 존재하지 않는 경우 많이 뜨는 Exception 중요함!!!!! 1) IOException : In, Out 관련 Error : 파일을 부를 때에는 기본적으로 예외처리가 필요로 한다. 2) NullPointerException : Null 값을 참조하지 못 할 때 발생하는 Error : 가장 많이 발생하며, 개발자가 무서워하는 에러 중 하나. 주로 Null값. 어떤 행동을 했을 때 많이 나타나며, 해결 방법은 if문을 이용하여 예외 처리 하도록 한다. null값과 비교하는 조건문을 사용하면 된다. 3) NumberFormatException : 문자를 숫자로 바꿀 때 발생하는 Error 4) IndexOutOfBoundsException : List 또는 ArrayList 에 잘못된 값이 들어있을 때 발생하는 에러 5) ArrayIndexOutOfBoundsException : 배열에서 잘못된 값이 들어 갔을 때 발생하는 에러 2. **사후 처리** 1) Exception : IOException 같은 경우 사후 처리 -> 복구가 불가능 하기 때문에 public 과 함께 throws로 선언해야 하며, 복구가 불가능 한 것을 Exception 이라고 한다. 2) RuntimeException : 나머지는 이미 선언이 되어있으며 사후처리로 복구가 가능하기 때문에 따로 선언하지 않는다. 3. **Try ~ catch ~ finally** 주로 app이 죽지 않도록 복구가 불가능한 Exception에서 사용하도록 하고, 프로그램을 무겁게만들기 때문에 되도록 피하는게 좋다. 4. **Data base 설치 및 Setting** 1) Oracle 설치  Developers -> download - > database -> Database 11g Express Edition 다운로드 2) 톰켓 아파치 https://tomcat.apache.org/download-80.cgi#8.0.53 -> zip 파일 다운로드 3) Toad for oracle 가서 Free 버전  <https://www.toadworld.com/products/toad-for-oracle>   Apache Tomcat 실행 후 Oracle 설치  1. 아파치 bin 파일 가서 startup 실행으로 설치 (자바 홈을 먼저 잡아야함.)  2. Oracle Setup 파일 실행 -> Next 후 ->  TNS Port (Oracle로 접속하는 Port)  MTS Port  HTTP Port 9999로 변경 -> Password 꼭 잊어버리지 않도록 적어두도록 한다. -> 확인 후  Install  만일, 설치가 되있다면 Google -> Oracle Modify HTTP Port 관련하여 수정 하도록 한다.  3. Toadfororacle 설치  Toadfororacle Setting  첫화면에서 User SYSTEM  PASSWORD 설치시 설정한 Password jys  TNS XE 선택  SAVE PASSWORDS 체크 후 CONNECT  연결을 끊고 싶을 때 Session -> end connection  연결하고싶을 때 Session -> new connection  Schema Browser 클릭시 목록왼쪽에 보면 Tables -> Users로 들어가면 빨간 자물쇠 (HR) 잠겨있는 상태  HR 실습용으로 쓸 DB -> HR 우클릭 후 Unlock 하면 X 표시가 됨.-> Alter User -> Password 다시 설정 -> password expired 체크 해제 후 ok  -> Session end connection -> new connection -> SYSTEM을 HR로 변경 후 CONNECT -> 다시 Schema Browser  Font 크기 설정 방법 VIEW -> EDITOR -> FONT AND STYLING  주석 -- 또는 /\* \*/ 를 사용하면된다.   1. **Data base** MySQL 또는 Oracle을 RDB (관계형 데이터베이스) 라고 불린다. 1) Data Base : 정형화 된 데이터를 저장하는 공간 2) Table (테이블) : 데이터가 저장 될 형태를 가지고 있음.  ex) 고객리스트, 제품 카탈로그, 특정한 종류의 정보 목록을 저장  테이블이 모여서 만들어 진 것이 RDB이다. 테이블에서는 글자수 제약이나 이 정보가 필수 입력 정보인지를 설정 할 수 있다. 3) Column (컬럼) : 테이블 안에서의 목록 하나하나를 컬럼이라고 부른다. 컬럼의 타입 == 변수의 타입 하나의 테이블에는 하나 이상의 컬럼으로 구성되어 있다. 4) Row (행) : 테이블의 행은 데이터라고 표현한다.  컬럼이 모인 하나의 특정 데이터를 Row 라고 한다. 5) Primary Key (주 키) P.K  Column에서 지정 하도록 한다. 절대 중복되어서는 안되며 이것을 무결성의 원칙이라고 한다. 무결성의 원칙 : 중복 되는 것이 없는 것을 무결성의 원칙이라고 한다. 만일, 중복이 된다면 무결성의 원칙이 깨졌다고 말한다. 2개이상 지정될 때에는 멀티키라고도 부른다. PK를 참조한 Col은 foreign Key(외래 키) 라고 불린다. 2. **C R U D** 1) Create : Insert into  2) Read : Select 3) Update : Update set 4) Delete 3. **Select 구문** 사용 방법  Select Column From Table 1) 중복 제거 DISTINCT   추출된 ROW의 중복을 제거한다. 2) ORDER BY 절  ORDER BY는 가장 문장 마지막에 사용한다. 기본적으로 오름차순 정렬로 되어 있다.  SELECT 하지 않아도 TABLE에 들어있는 무엇이든 정렬하는데 사용 할 수 있다. 기본이 오름차순 이지만 설정을 할 수 있다. 오름차순 ORDER BY NAME ASC 내림차순 ORDER BY NAME DESC  PK는 기본적으로 정렬되기 때문에 따로 정렬을 할 필요는 없다. 이차 정렬 시에는 ORDER BY 1차 정렬, 2차정렬 이런식으로 사용하면 된다. ORDER BY가 들어가는 순간 프로그램이 느려지기 때문에 PK로 해결이 안되는 정말 필요 할 때에만 사용하도록 한다.  3) 데이터 필터링 WHERE 절  SELECT에서 WHERE 절로 검색 조건을 지정하여 필터링 할 수 있다 FROM 뒤에 나와야 한다. 사용 방법은 WHERE TABLE 명 여러 개를 비교 할 때에는 WHERE TABLE 명 AND TABLE 명 형식으로 AND 를 사용하면 된다.  4) 비교연산의 종류 = 같다 > 크다 < 작다 >= 크거나 같다 <= 작거나 같다 != 다르다 IS NULL = NULL 값이다. 문자를 비교 할 때에는 ‘ ‘ 안에 문자를 넣어주도록 한다. 4. **오라클과 자바 연동** 오라클과 자바를 연동하기 위해서는 OJBC 라이브러리가 꼭 필요하다. 오라클에서 Oracle JDBC 드라이버 다운로드 11.2.0.4 버전  ojdbc6.jar   연동 방법 1) 프로젝트에 jar 파일을 붙여넣는다. 2) 프로젝트 설정 Build Path -> configure Build Path -> Libraries -> ClassPath 선택 -> add JARs -> 해당 프로젝트 선택 -> ojdbc6 파일 선택 5. **중요하므로 꼭 외어야 한다. 연동 프로그램 작성 법** \* DB에 접근해 EMPLOYEES 테이블의 정보를 출력한다.   \* 1. DB에 접근(HR)  \* 2. Query 준비 및 실행  \* 3. Query 결과 출력    //1.DB에 접근(HR)  //오라클에 접근하기 위한 객체 Load  try {  Class.forName("oracle.jdbc.driver.OracleDriver");  } catch (ClassNotFoundException e) {  System.out.println( e.getMessage() );  }      // Oracle(HR) 접속  Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;  try {  conn = DriverManager.getConnection(  "jdbc:oracle:thin:@localhost:1521:XE", "HR", "jys");  //2. Query 준비 및 실행  pstmt = conn.prepareStatement("SELECT \* FROM EMPLOYEES");    //3. Query 결과 출력  rs = pstmt.executeQuery();    // 결과 출력  while ( rs.next() ) {  int employeeId = rs.getInt("EMPLOYEE\_ID");  String firstName = rs.getString("FIRST\_NAME");  String lastName = rs.getString("LAST\_NAME");  String hireDate = rs.getString("HIRE\_DATE");  double commissionPct = rs.getDouble("COMMISSION\_PCT");    System.out.printf("%d, %s, %s, %s, %f\n",  employeeId,  firstName,  lastName,  hireDate,  commissionPct);  }  } catch (SQLException e) {  System.out.println(e.getMessage());  }  finally {  if ( rs != null ) {  try {  rs.close();  } catch (SQLException e) {  }  }  if ( pstmt != null ) {  try {  pstmt.close();  } catch (SQLException e) {  }  }  if ( conn != null ) {  try {  conn.close();  } catch (SQLException e) {  }  }    } |
|  |  |

|  |  |
| --- | --- |
| 특이사항 |  |