**로고학습일지**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 08. 08 (수) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 |  | | | | |
| 작 성 자 | 신예리 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 |  |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| **<ERwin>**  실무에서 컬럼 너무 길면 모음을 지워서 줄이는 편임. (26자를 넘어가면 안댐)  -> DELIVERY 🡺 DLVRY / FROM\_PHONE 🡺 FRM\_PHN  >Physical에서 Datebase  > Choose Database > Oracle 선택 > Default Oracle Datatype : VARCHAR2(0)  (Oracle 문자타입 /. 0: byte 크기 설정)  > 컬럼 더블 클릭 > Oracle > Oracle Datatype 선택 / NULL OR NOT NULL 선택  > 마우스 오른쪽 클릭 > Table Display > Column Datatype & Primary Key Designator 켜기  -NUMBER : 소수점까지 / INTEGER : 숫자  -VARCHAR2(2000) <- 가변길이 : 2000바이트까지 자유롭게 / CHAR(1) <- 고정길이 : 1바이트만 꼭 쓰기  -시간 : TIMESTAMP 많이 쓰임 (DATE말고)    **<TOAD>** 새로운 Schema, User, Table 만드는 법  System으로 접속 > Schema Browser > Users > 오른쪽 클릭 > Create User  > User Info > User Name : DLVRY > password : DLVRY > Profile : DEFAULT  > Tablespace > Default tablespace : USERS > Temporary tablespace : TEMP  > Roles > Copy From… > HR 선택 > OK (HR의 설정 그대로 가져옴)  > System Privileges > HR 선택 > OK (HR의 설정 그대로 가져옴)  **<ERwin>**  >**ERwin** 그대로 가져와 테이블 만들 때  Tools > Forward Engineer > Schema Generation > Table 클릭 후 Procedure, Funciton 체크 해제  > Priview 클릭 > CREATE TRIGGER 전까지 복사 >>> **<TOAD>**에 복붙  **<TOAD>**  데이터 넣어줄 때 참조되는 것 먼저 데이터 넣기. (1쪽을 먼저 쓰고 다쪽을 나중에)  /\*  다음에는 아래 코드 복사하지 말 것.  데이터 관리 어려워짐.  논리적으로 관계를 만들었으나 물리적인 관계를 만들지 않는 것!!!  ALTER TABLE DELIVERY  ADD (CONSTRAINT R\_4 FOREIGN KEY (FROM\_PHONE) REFERENCES CUSTOMER (PHONE));  ALTER TABLE DELIVERY  ADD (CONSTRAINT R\_8 FOREIGN KEY (TO\_PHONE) REFERENCES CUSTOMER (PHONE));  ALTER TABLE DELIVERY  ADD (CONSTRAINT R\_9 FOREIGN KEY (DELIVERY\_MAN\_PHONE) REFERENCES DELIVERY\_MAN (PHONE));  \*/  <DB>  INSERT INTO 테이블명 (컬럼들)  VALUES (컬럼에 맞는 값들)  ;  INSERT INTO DELIVERY\_MAN (  PHONE  , NAME  )  VALUES (  ‘010-1234-5678’  ‘강재현’  )  ;  COMMIT; -- 저장 데이터를 영구히 보관하겠다.  ROLLBACK; -- 영구히 보관되지 않는 데이터가 제거됨.  일대다 관계가 N개 있다면 조인할 때 N번 써줘야 함.  SELECT TO\_PHONE  , "TO".NAME  , FROM\_PHONE  , "FROM".NAME  , DELIVERY\_MAN\_PHONE  , M.NAME  FROM DELIVERY D  , CUSTOMER "TO" -- TO랑 FROM은 예약어라서 "" 씀  , CUSTOMER "FROM"  , DELIVERY\_MAN M  WHERE D.TO\_PHONE = "TO".PHONE  AND D.FROM\_PHONE = "FROM".PHONE  AND M.PHONE = D.DELIVERY\_MAN\_PHONE  ;  <STS>  SELECT 할 떄 rs = pstmt.executeQuery();  INSERT UPDATE DELETE 할 때 pstmt.executeUpdate()  Object는 모든 클래스의 부모클래스이다. 즉, 모든것을 리턴할 수 있다. |

|  |  |
| --- | --- |
| 특이사항 |  |

■ORACLE

※ERwin에서 참조해 온 Table CREATE, ALTER

CREATE TABLE CUSTOMER

(

PHONE VARCHAR2(13) NOT NULL ,

NAME VARCHAR2(100) NOT NULL ,

ADDRESS VARCHAR2(2000) NOT NULL

);

CREATE UNIQUE INDEX XPK고객 ON CUSTOMER

(PHONE ASC);

ALTER TABLE CUSTOMER

ADD CONSTRAINT XPK고객 PRIMARY KEY (PHONE);

CREATE TABLE DELIVERY

(

DELIVERY\_ID VARCHAR2(100) NOT NULL ,

BOX\_SIZE INTEGER NOT NULL ,

WEIGHT NUMBER NOT NULL ,

PRICE NUMBER NOT NULL ,

NAME VARCHAR2(2000) NOT NULL ,

TYPE VARCHAR2(100) NOT NULL ,

FRAGIL CHAR(1) NOT NULL ,

DISTANCE NUMBER NOT NULL ,

QUICK CHAR(1) NOT NULL ,

PAY\_TYPE CHAR(1) NOT NULL ,

SHIPPING\_FEE INTEGER NOT NULL ,

RETURN\_YN CHAR(1) NOT NULL ,

MESSAGE VARCHAR2(3000) NOT NULL ,

RECEIPT\_DATE TIMESTAMP NOT NULL ,

FROM\_PHONE VARCHAR2(13) NOT NULL ,

TO\_PHONE VARCHAR2(13) NOT NULL ,

DELIVERY\_MAN\_PHONE VARCHAR2(13) NOT NULL

);

CREATE UNIQUE INDEX XPK배송 ON DELIVERY

(DELIVERY\_ID ASC);

ALTER TABLE DELIVERY

ADD CONSTRAINT XPK배송 PRIMARY KEY (DELIVERY\_ID);

CREATE TABLE DELIVERY\_MAN

(

PHONE VARCHAR2(13) NOT NULL ,

NAME VARCHAR2(100) NOT NULL

);

CREATE UNIQUE INDEX XPK기사 ON DELIVERY\_MAN

(PHONE ASC);

ALTER TABLE DELIVERY\_MAN

ADD CONSTRAINT XPK기사 PRIMARY KEY (PHONE);

/\*

다음에는 아래 코드 복사하지 말 것.

데이터 관리 어려워짐.

논리적으로 관계를 만들었으나 물리적인 관계를 만들지 않는 것!!!

ALTER TABLE DELIVERY

ADD (CONSTRAINT R\_4 FOREIGN KEY (FROM\_PHONE) REFERENCES CUSTOMER (PHONE));

ALTER TABLE DELIVERY

ADD (CONSTRAINT R\_8 FOREIGN KEY (TO\_PHONE) REFERENCES CUSTOMER (PHONE));

ALTER TABLE DELIVERY

ADD (CONSTRAINT R\_9 FOREIGN KEY (DELIVERY\_MAN\_PHONE) REFERENCES DELIVERY\_MAN (PHONE));

\*/

※ INSERT

INSERT INTO DELIVERY\_MAN (

PHONE

, NAME

)

VALUES (

'010-1234-5678'

, '강재현'

)

;

COMMIT; -- 저장 데이터를 영구히 보관하겠다

INSERT INTO DELIVERY\_MAN (

PHONE

, NAME

)

VALUES (

'010-1234-5679'

, '류복기'

)

;

ROLLBACK; -- 영구히 보관되지 않은 데이터가 제거됨.

INSERT INTO DELIVERY\_MAN (

PHONE

, NAME

)

VALUES (

'010-1234-5680'

, '김종현'

)

;

INSERT INTO CUSTOMER (

PHONE

, NAME

, ADDRESS

)

VALUES (

'010-9876-5432'

, '손동식'

, '인천'

)

;

INSERT INTO CUSTOMER (

PHONE

, NAME

, ADDRESS

)

VALUES (

'010-9876-2345'

, '신예리'

, '인천'

)

;

INSERT INTO CUSTOMER (

PHONE

, NAME

, ADDRESS

)

VALUES (

'010-6789-5432'

, '김세현'

, '광진구'

)

;

INSERT INTO CUSTOMER (

PHONE

, NAME

, ADDRESS

)

VALUES (

'010-6789-2345'

, '전상문'

, '은평구'

)

;

INSERT INTO CUSTOMER (

PHONE

, NAME

, ADDRESS

)

VALUES (

'010-1234-5432'

, '강준영'

, '강릉'

)

;

COMMIT;

INSERT INTO DELIVERY (

DELIVERY\_ID

, BOX\_SIZE

, WEIGHT

, PRICE

, NAME

, TYPE

, FRAGIL

, DISTANCE

, QUICK

, PAY\_TYPE

, SHIPPING\_FEE

, RETURN\_YN

, MESSAGE

, RECEIPT\_DATE

, FROM\_PHONE

, TO\_PHONE

, DELIVERY\_MAN\_PHONE

)

VALUES (

'1'

, 3

, 5

, 1000

, '돌'

, '잡화'

, 'N'

, 10

, 'N'

, 'N' -- 착불

, 2500

, 'N'

, '던져도 됩니다.'

, SYSDATE

, '010-3333-2222'

, '010-9876-5432'

, '010-1234-5680'

)

;

COMMIT;

SELECT TO\_PHONE

, FROM\_PHONE

, DELIVERY\_MAN\_PHONE

FROM DELIVERY D

, CUSTOMER C

, DELIVERY\_MAN M

WHERE D.TO\_PHONE = C.PHONE

AND D.FROM\_PHONE = C.PHONE

AND M.PHONE = D.DELIVERY\_MAN\_PHONE

; -- 값이 나오지 않음. (TO\_PHONE과 FROM\_PHONE이 같을 수 없어)

SELECT TO\_PHONE

, "TO".NAME

, FROM\_PHONE

, "FROM".NAME

, DELIVERY\_MAN\_PHONE

, M.NAME

FROM DELIVERY D

, CUSTOMER "TO" -- TO랑 FROM은 예약어라서 "" 씀

, CUSTOMER "FROM" -- 1대다가 2개이기 때문에 2번 조인

, DELIVERY\_MAN M

WHERE D.TO\_PHONE = "TO".PHONE

AND D.FROM\_PHONE = "FROM".PHONE

AND M.PHONE = D.DELIVERY\_MAN\_PHONE

;